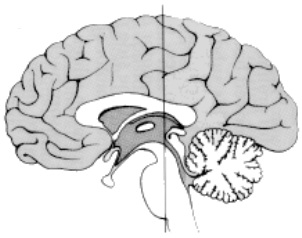
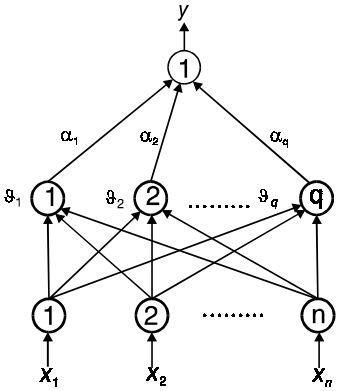
Úvod do teórie neurónových sietí

Vladimír Kvasnička
Ľubica Beňušková
Jiří Pospíchal
Igor Farkaš
Peter Tiňo
Andrej Kráľ





OBSAH

1. Neurónové siete a nervový systém

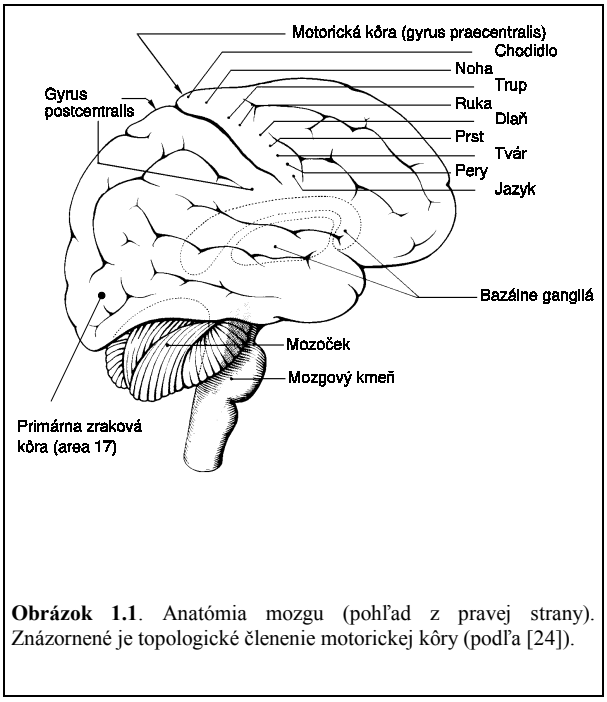
Teória neurónových sietí vychádza z neurofyziologických poznatkov. Snaží sa vysvetliť správanie sa na princípe spracovania informácií v nervových bunkách. Niekedy sa umelé neurónové siete označujú aj ako modely mozgu bez mysle (angl. *brain without mind*, Clark a spol. [5]), keďže sa snažia pochopiť nervový systém, ale nezaoberajú sa psychikou. Znalosti získané v oblasti výskumu umelých neurónových sietí majú veľký význam pre neurofyziológiu. Momentálne totiž nemáme k dispozícii experimentálne metodiky, ktoré by umožňovali sledovať aktivitu v prirodzených neurónových sieťach (existujú len metódy na sledovanie aktivity jednej, popr. niekoľkých nervových buniek súčasne). V oblasti spracovania informácií nervovým systémom sa preto k niektorým poznatkom možno dopracovať len myšlienkovým experimentom (napr. počítačovým modelom).

1.1 Anatómia nervového systému

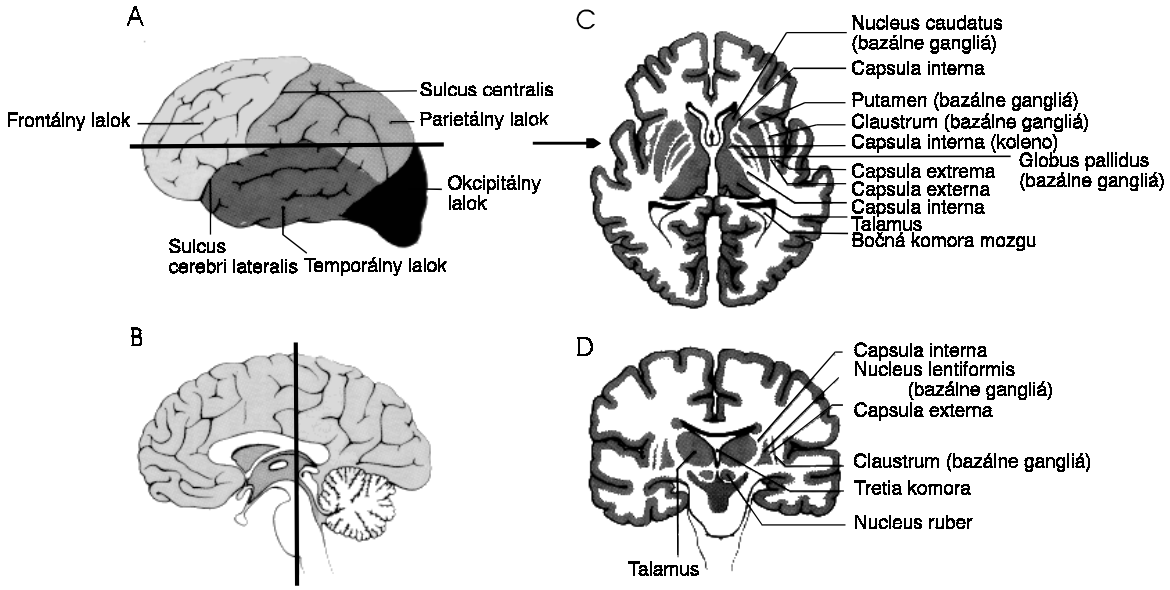
Nervový systém obsahuje obrovské množstvo buniek (len mozog ich obsahuje viac ako 100 miliárd). V princípe sa rozdeľuje na *centrálny nervový systém* (CNS, t.j. mozog a miecha) a *periférny nervový systém* (PNS, t.j. periférne nervy a zhluky nervových buniek, tzv. *gangliá*). Všetky informácie, ktoré CNS dostáva o svete naokolo, sa zakladajú na signáloch vznikajúcich v zmyslových (senzorických) bunkách. Tieto signály sa šíria do príslušných ústredí v CNS cestou oddelených (tzv. *aferentných,* dostredivých) nervových dráh. Podobne jediným spôsobom, ako môže organizmus cielene ovplyvňovať okolité prostredie, je cestou *eferentných* (odstredivých) dráh inervujúcich (riadiacich) svalstvo (aj reč je zabezpečovaná svalstvom). Svet, tak ako ho vnímame, je svet virtuálny, existujúci len v našej mysli vo forme nervových vzruchov. Nevidíme elektromagnetické vlnenie v istom rozsahu frekvencií (predstavujúcom adekvátny podnet pre oko), "vidíme" virtuálny svet vytvorený neurálnou aktivitou v zrakových centrách CNS (podrobnosti v [17]).

Miecha vykonáva elementárnu analýzu hmatových podnetov, vnímania teploty a bolesti. Ústredia pre tieto činnosti sa nachádzajú aj na vyšších úrovniach CNS. Okrem toho vykonáva reguláciu niektorých reflexných funkcií pohybového aparátu (*motorika*) a niektorých vegetatívnych reflexov. Obsahuje aferentné a eferentné dráhy. Najvyššou úrovňou miechy je *predĺžená miecha*, ktorá pokračuje štruktúrami nazývanými *Varolov most* a *mesencephalon*. Tieto tri štruktúry sa súborne označujú ako *mozgový kmeň* (obr. 1.1). Zaisťujú *vegetatívne funkcie* (t.j. vedomím priamo neovládané funkcie zabezpečujúce stabilitu vnútorného prostredia organizmu). Ide napr. o reguláciu srdcovej frekvencie, dýchania, kyslosť vnútorného prostredia, koncentrácie iónov v krvi, atď. Okrem toho mozgový kmeň zabezpečuje základnú analýzu sluchových, chuťových ako aj hmatových podnetov z oblasti tváre. Podobne riadi motoriku tváre. Je centrom niektorých reflexov (napr. zabezpečujúcich rovnováhu, pohyby očí, rozširovanie a zužovanie zreníc a pod.).

Mozoček hrá veľmi významnú úlohu pri modifikácii starých a učení nových pohybových stereotypov. Súčasne porovnáva vykonávaný pohyb s jeho plánom a prípadné odchýlky koriguje. *Mozog* sa skladá z dvoch polovíc, tzv. *hemisfér*. Väčšina štruktúr patriacich k mozgu je zastúpená dvakrát, v oboch hemisférach (obr. 1.2). *Talamus* predstavuje veľmi zložitú štruktúru, ktorou prechádzajú takmer všetky aferentné dráhy. Ide o nahromadenie nervových buniek v centrálnych častiach mozgu; takéto zhluky sú typické tým, že na reze majú sivú farbu. Miesta, kde sa nenachádzajú telá nervových buniek, majú bielu farbu. Okolo talamu sa nachádza niekoľko zhlukov neurónov, ktoré sa súborne označujú ako *bazálne gangliá*. Hrajú dôležitú úlohu pri selekcii programu pre pohybový vzorec zvoleného pohybu. Pri ich poruchách vznikajú zložité poruchy motoriky (napr. Parkinsonova choroba). Emotívne podfarbené správanie ovplyvňuje systém skupín nervových buniek nazvaný *limbický systém*. Jeho jedna časť, tzv. *hippocampus*, má zrejme centrálnu úlohu vo fyziológii pamäti.



Obrázok 1.1. Anatómia mozgu (pohľad z pravej strany). Znázornené je topologické členenie motorickej kôry (podľa [24]).



Obrázok 1.2. Anatómia mozgu. A. Pohľad na ľavú hemisféru zboku. Znázornené je členenie kôry na laloky. Vodorovná línia znázorňuje miesto rezu C. B. Pohľad na strednú plochu pravej hemisféry. Zvislá čiara znázorňuje miesto rezu D. C. Horizontálny rez mozgom. Viditeľné sú štruktúry uložené v hĺbke mozgu (majú prirodzene sivú farbu kvôli telám nervových buniek, ktoré obsahujú). D. Vertikálny rez mozgom (podľa [24]).

Mozgová kôra zohrala v evolúcii človeka rozhodujúcu funkciu. Najväčší rozdiel medzi mozgom človeka a k nemu vývinovo najbližších opíc je práve v štruktúre mozgovej kôry. Je na povrchu rozdelená na vyvýšenia a brázdy medzi nimi (*gyrus* je označenie vyvýšenia a *sulcus* brázdy oddeľujúcej dve susedné vyvýšenia). V smere kolmom na povrch mozgu je kôra vzhľadom na nehomogénnu bunkovú štruktúru rozdelená na 6 vrstiev. Okrem toho ju Korbinian Brodmann podľa funkcie rozdelil v pozdĺžnom smere na tzv. *Brodmannove arey* (napr. primárne zrakové centrum reprezentuje area 17). V princípe sa na kôre nachádzajú oblasti zodpovedajúce analýze vstupu zo zmyslových orgánov, oblasti zodpovedné za riadenie motoriky a tzv. asociačné oblasti, kde sa zbiehajú informácie z ostatných oblastí kôry. Vzhľadom na to, že väčšina aferentných aj eferentných dráh sa kríži, sú v ľavej hemisfére reprezentované informácie z pravého zorného poľa, hmat z pravej polovice tela, motorika pravej polovice tela a pod. Pochopiteľne obe hemisféry nepracujú autonómne. Sú spojené

veľkým množstvom nervových dráh. Spojené sú korešpondujúce oblasti jednej hemisféry s korešpondujúcimi oblasťami druhej hemisféry a naopak. Štruktúra, kde tieto spojenia prebiehajú, sa nazýva *corpus callosum* (existujú aj iné spojenia medzi hemisférami, *corpus callosum* je najväčšie z nich).

Funkcia jednotlivých hemisfér je mierne odlišná. Dokázali to Roger Sperry a Michael Gazzaniga v sérii experimentov na pacientoch, ktorým boli z terapeutických dôvodov prerušené spojenia medzi hemisférami (pacienti po komisurotómii, *"split-brain"* pacienti; prvé poznatky boli získané v experimentoch na zvieratách [21], prehľad pozri v [9]). Zistilo sa, že v jednej hemisfére sú lokalizované funkcie súvisiace s rečou (tzv. *dominantná hemisféra*, zvyčajne ide o ľavú hemisféru). Okrem toho je dominantná hemisféra špecializovaná na racionálne spracovanie reality a je spojená s vedomím. Druhá hemisféra (*subdominantná*) je špecializovaná na vnímanie melódie, funkcie spojené s vnímaním estetiky, pomyselné manipulácie s objektmi v priestore a skôr celostné, holistické spracovanie reality. Vstup do subdominantnej hemisféry je u pacientov po komisurotómii neprístupný vedomiu. Napríklad čo existuje len v jej odpovedajúcej časti zorného poľa, si pacient neuvedomuje (pacient nevie, čo vidí; predpokladom je, že zrak fixuje jeden bod, v dôsledku čoho sa oči neobracajú smerom k prezentovanému objektu - tak sa zabezpečí, aby bol objekt premietnutý len na tú časť sietnice, ktorá predstavuje vstup subdominantnej hemisféry). Sperry, Gazzaniga a spolupracovníci ukázali, že ak je napr. obraz pílky krátko prezentovaný v ľavom zornom poli (teda premietnutý do pravej hemisféry) a pacient má dominantnú hemisféru ľavú, nevie povedať, čo videl. Ak má však nakresliť obrázok toho, čo videl, ľavou rukou (riadenou subdominantnou hemisférou) nakreslí pílku (hoci si neuvedomuje, čo kreslí).[1]

Na tomto mieste je potrebné upozorniť, že u veľkej väčšiny vyšších psychických funkcií sú nervové mechanizmy zodpovedné za tieto funkcie úplne neznáme (podrobnosti pozri v [22]).

Aby sme si mohli priblížiť, aké funkcie musí nervový systém zabezpečiť, predstavme si, čo musí vykonať, keď chceme prejsť cez frekventovanú ulicu. Aby sme mohli stáť na dvoch končatinách s telom vzpriameným (a teda s ťažiskom vysoko od zeme, v pomerne labilnej polohe), musí nervový systém spracovať informáciu z vestibulárneho orgánu, registrujúceho vplyv gravitácie a zrýchlenie nášho tela v niektorom z troch na seba kolmých smeroch (vestibulárny orgán je uložený v blízkosti vnútorného ucha). Ten je komplikovanou neurónovou sieťou spojený s nervovými bunkami, stimulujúcimi kontrakcie istých svalov (tzv. posturálne, postojové svalstvo). Primeraná kontrakcia jednotlivých posturálnych svalov udržuje človeka vo vzpriamenej polohe. Ide však o stále sa meniaci proces, rozsah ich kontrakcie totiž závisí od konkrétnej situácie: na ako naklonenom, ako mäkkom a ako pohyblivom povrchu stojíme, odkiaľ a ako silno fúka vietor, čo nesieme v rukách a podobne. Naše oči sledujú ulicu, zatiaľ čo ich oslepuje letné slnko. Informácia o svetelnej intenzite sa tvorí už v oku, ďalej sa spracuje v aferentnej zrakovej dráhe a pomocou ústredí v mozgovom kmeni sa určí, aký má byť prierez zreničky. Jej ovplyvnenie sa deje cestou príslušných eferentných dráh. Zrenička sa potom ako clona na fotoaparáte pri silnom svetle zužuje. Tým zabezpečí primeranú svetelnú intenzitu na spracovanie obrazu sietnicou[2] (na tomto mechanizme sa zúčastňuje aj samotná sietnica). Iné centrá v mozgovom kmeni riadia natočenie očných gúľ tak, aby "pozerali" na ten istý bod (konvergencia). Tak sa môžu obrazy premietnuté na sietnice oboch očí zložiť v zrakovej kôre dohromady, čím na základe ich disparity (t.j. malého posunu medzi obrazom v pravom a ľavom oku) môže vzniknúť "hĺbka" obrazu (jeho trojdimenzionálnosť). Našu vzdialenosť od objektov v zornom poli odhaduje nervový systém aj porovnaním veľkosti obrazu so skúsenosťou (vyžaduje to analýzu celej situácie). Aby sme preorientovali svoj pohľad na iný objekt, je potrebné natočiť očné gule príslušným smerom a upraviť aj ich konvergenciu podľa vzdialenosti objektu od nás. Zrak má okrem toho schopnosť autonómne kompenzovať pohyby hlavy komplexom zložitých reflexov. Pasívne pohyby hlavou sú registrované vestibulárnym aparátom, ktorý potom ovplyvňuje centrá otáčajúce očné gule opačným smerom, ako sa pohybuje hlava. Podobne, ak sa pohybuje obraz dopadajúci na sietnicu (čo sa v našom prípade naozaj deje - autá sa pomerne veľkou rýchlosťou približujú a vzďaľujú), očné gule tento pohyb dokážu kompenzovať. Reflexný oblúk je v tomto prípade ešte zložitejší. Zvyčajne nastávajú všetky tieto procesy súčasne, takže dochádza k ich komplexnej superpozícii. Náš sluch registruje hluk ulice. Je v ňom zvuk motora štartujúceho motocykla z blízkeho parkoviska, krik dieťaťa ťahajúceho mamu na opačnej strane cesty k hračkárstvu, brechot psa kdesi zďaleka, slová priateľky popisujúcej zážitky z práce a hluk motorov prechádzajúcich áut. Sluch vníma len jedno spektrum, vzniknuté zložením všetkých týchto zvukov. Napriek tomu vieme veľmi ľahko tieto zvuky odlíšiť, prisúdiť im miesto, odkiaľ prichádzajú, hlasitosť, výšku a farbu. Rozumieme tomu, čo naša priateľka hovorí, zvuky motorov nám pomáhajú odhadnúť, kedy možno prejsť cez cestu. Na to je potrebné, aby náš sluch vyhodnotil spektrum sluchového podnetu, intenzitné a časové charakteristiky (aj vzájomný vzťah týchto

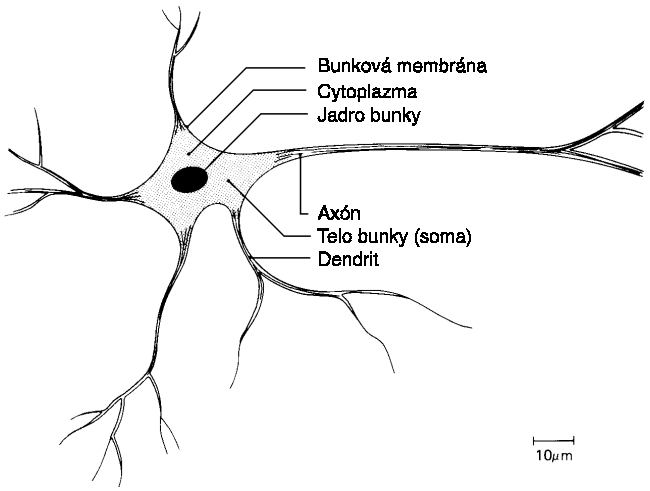
[1] Všetky popísané efekty predpokladajú dôslednú separáciu vstupu pre jednotlivé hemisféry. Za normálnych okolností má však pacient tendenciu zabezpečiť, aby vstupy boli do oboch hemisfér (pre zrak napr. otáča hlavou a očnými guľami). Takisto špecializácia hemisfér je skutočnosť čiastočne zjednodušená, v opísanej podobe prítomná len po komisurotómii. Za normálnych okolností sú hemisféry spojené a vytvárajú spolu jeden funkčný celok.

[2] Neprítomnosť tohto reflexu svedčí o ťažkom poškodení mozgu zasahujúcom až mozgový kmeň.

parametrov medzi jedným a druhým uchom). Tieto sa musia porovnať so zvukmi uloženými v pamäti. Nervový systém je súčasne nútený riadiť vnútorné prostredie organizmu - t.j. parametre ako teplota tela, krvný tlak, frekvencia srdca, dýchania, koncentrácia rôznych látok v krvi. Syntézou všetkých podnetov, ktoré pôsobia na naše zmysly, si vytvárame akýsi "vnútorný" obraz o premávke na ulici a vnímame aj slová priateľky. Vnímame zmeny veľkosti obrazu áut na našej sietnici, vnímame zmeny ich polohy, zmeny zvuku ich motorov. Situácia v druhom jazdnom smere zatiaľ zostáva v pamäti, dokonca sa podľa odhadu aktualizuje. V momente, keď vypočítame časovú medzeru na prechod ulicou, spustí náš nervový systém natrénovanú sekvenciu procesov v mieche, vedúcich veľmi komplikovaným spôsobom k chôdzi. Dochádza k aktivácii obrovského množstva motorických jednotiek (základných funkčných zložiek svalu) v rôznych svaloch v istej časovej sekvencii. Svaly umožňujúce opačné pohyby však musia relaxovať. Začatý pohyb zmení polohu ťažiska. To zas vedie k reflexnej zmene tonusu posturálneho svalstva. Náš pohyb musí byť pri odhade premávky vzatý do úvahy. Cielene presúvame ťažisko z jednej nohy na druhú, kráčame. Všetko toto v zlomkoch sekundy. Náhle zahliadneme rýchlo sa blížiace červené auto. Prudko zastavíme. Bleskovo prebehne sekvencia kontrakcií svalov zabezpečujúca rýchle zastavenie aj udržanie rovnováhy. Všetky tieto procesy (a enormné množstvo ďalších) prebiehajú v našom organizme každý zlomok sekundy, a všetky sú ovládané nervovým systémom. Konštrukcia robota, vykonávajúceho tieto činnosti naraz (napriek skutočnosti, že k tomu nie je nevyhnutnou podmienkou existencia vedomia), zostáva v nedohľadne.

1.2 Neurón

Hlavnou funkciou nervového systému je riadiť organizmus. Podkladom tejto funkcie je schopnosť nervového systému spracovávať informácie. Informácie sa v nervovom systéme prenášajú vo forme zmien membránového potenciálu nervových buniek, **neurónov**.



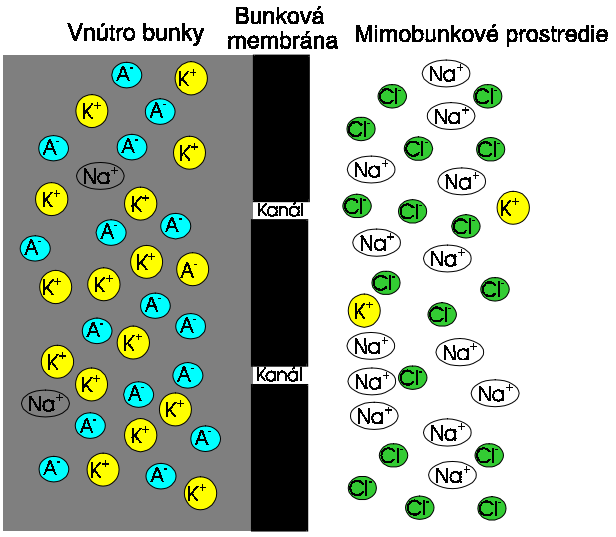
Obrázok 1.3. Nervová bunka (neurón). V spodnej časti obrázku je pre orientáciu znázornená mierka. Tvar neurónu je značne variabilný, na rôznych miestach v nervovom systéme sa nachádzajú bunky veľmi rozdielneho tvaru (podľa [24]).

Neuróny (podobne ako iné bunky v organizme) sú ohraničené bunkovou membránou (obr. 1.3). Je tvorená fosfolipidovou dvojvrstvou. Ide o membránu polopriepustnú. Môžu cez ňu prechádzať niektoré molekuly s malou molekulovou hmotnosťou. Molekuly bielkovín za normálnych okolností membránou neprechádzajú. Medzi bunkou a jej okolím existuje elektrický gradient. Nie je spôsobený len molekulami bielkovín, ale aj preskupením iónov, ktoré v istom rozsahu dokážu prechádzať bunkovou membránou a vytvárajú rovnovážny stav medzi tokom do bunky a z bunky von. Vnútrobunkové prostredie je typické vyššou koncentráciou draslíka, mimobunkové prostredie vyššou koncentráciou sodíka, kalcia a chloridov. V pokojových podmienkach je na vnútornej strane bunkovej membrány negatívny náboj (pokojový potenciál je okolo -90 mV, obr. 1.4).

Ióny prechádzajú cez membránu iónovými kanálmi, ktoré sú tvorené molekulami bielkovín. Medzi niekoľkými takýmito molekulami zapadajúcimi do seba (podobne ako tehličky v skladačke Lego; hovoríme o tzv. podjednotkách kanála) vznikajú otvory, cez ktoré prechádzajú ióny a molekuly vody z medzibunkového prostredia do bunky a opačne. Vzhľadom na to, že spomínané ióny majú rôznu veľkosť, elektrický náboj a pod.,

je zvyčajne iónový kanál špecifický pre niektorý ión (hovoríme o kanáloch sodíkových, draslíkových, kalciových a chloridových). V princípe možno iónové kanály rozdeliť na 4 podtypy:

- "*leakage*" kanály (za normálnych okolností otvorené kanály, cez ktoré aj v pokojových podmienkach prechádzajú ióny v závislosti od koncentračného a elektrického gradientu);
- *napätím ovládané* kanály (angl. *voltage-operated* alebo nazývané aj *voltage-gated,* otvárajúce sa pri istej hodnote membránového potenciálu). Majú rozhodujúcu funkciu pri šírení zmien elektrického potenciálu pozdĺž bunkovej membrány;
- *receptorom ovládané* kanály (angl. *receptor-operated,* otvárajúce sa pri naviazaní istej látky - tzv. *ligandu* - na *receptor*; nazývajú sa aj *ligand-gated*). Ligand zapadá do príslušnej časti receptorovej molekuly ako ruka do rukavice. Podobne ako rukavica sa aj molekula receptora (zvyčajne tvoria podjednotku kanálu) prispôsobí ligandu. To vyvolá zmenu priestorovej štruktúry celého kanálu. Rozšírením najužšieho miesta kanála (tzv. brána) sa kanál stáva priechodným. Receptorom ovládané iónové kanály majú rozhodujúcu funkciu pri prenose elektrického vzruchu z jednej nervovej bunky na druhú;
- *mechanicky ovládané* kanály (otvárajúce sa pri natiahnutí alebo stlačení okolitej membrány). Takéto kanály sú zodpovedné za premenu mechanickej energie na elektrickú aktivitu v zmyslových orgánoch (napr. vnútorné ucho);
- *dvojito-otvárané* kanály (angl. *double-gated*). Takýmto kanálom je napr. NMDA[3] receptorom ovládaný kalciový kanál. Na jeho otvorenie je potrebné jednak naviazanie istej molekuly (neuromediátora glutamát) na jeho receptorovú časť, jednak istá depolarizácia okolitej membrány. Ide teda o kombináciu receptorom a napätím ovládaného kanála.



Obrázok 1.4. Rozdelenie elektricky nabitých molekúl na bunkovej membráne.

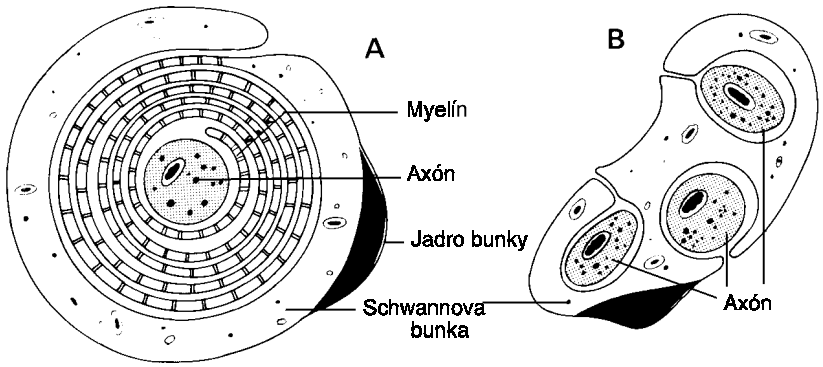
1.3 Stavba neurónu

Nervová bunka sa skladá z tela (lat. *soma*) a niekoľkých výbežkov. Tieto možno rozdeliť na dva typy: *dendrity*, ktoré predstavujú z informatického hľadiska vstupnú časť (predovšetkým na ne prechádza vzruch z iných buniek) a jeden *axón*, po ktorom sa vzruch šíri k iným bunkám (obr. 1.3). V mieste odstupu axónu z tela neurónu sa nachádza tzv. *iniciálny segment*, ktorý má významnú funkciu pri šírení vzruchu. Axón býva často obalený *myelínovou pošvou*. Ide o bunkovú membránu tzv. *gliovej bunky* [4], ktorá tesne obtáča axón. Na hraniciach medzi dvoma susednými gliovými bunkami sa na axóne nachádzajú neobalené úseky, tzv. *Ranvierove zárezy* (obr. 1.5,

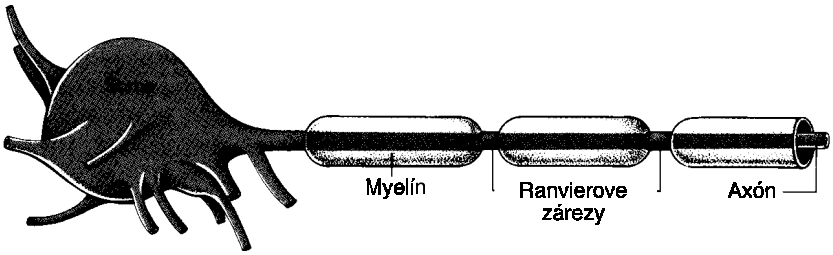
[3] NMDA je skratka pre N-metyl-D-aspartát, čo je synteticky pripravený (umelý) ligand vedúci k otvoreniu kanála.

[4] Medzibunkové priestory medzi neurónmi sú vyplnené spojivom a spojivovými bunkami. V centrálnom nervovom systéme sa tieto bunky nazývajú *gliové bunky*, na periférii (v nervoch) *Schwannove bunky*. Majú množstvo funkcií, okrem iného vyživovaciu, podpornú a obrannú.

1.6). Zmeny membránového napätia (elektrické vzruchy) potom preskakujú myelínovou pošvou izolované úseky a šíria sa priamo od jedného Ranvierovho zárezu k druhému. Takéto tzv. *saltatórne (skokovité) vedenie* je výhodné vzhľadom na vysokú rýchlosť šírenia. (Podrobnosti o štruktúre nervovej bunky možno nájsť napr. v [15]).



Obrázok 1.5. Poloha axónu vo vzťahu k spojivovým bunkám. A. Myelinizovaný axón. Membrána Schwannovej bunky sa mnohokrát obtáča okolo axónu. B. Tri nemyelinizované axóny a ich poloha vo vzťahu k Schwannovej bunke. V tomto prípade sa membrána Schwannovej bunky neobtáča okolo axónu (podľa [24]).

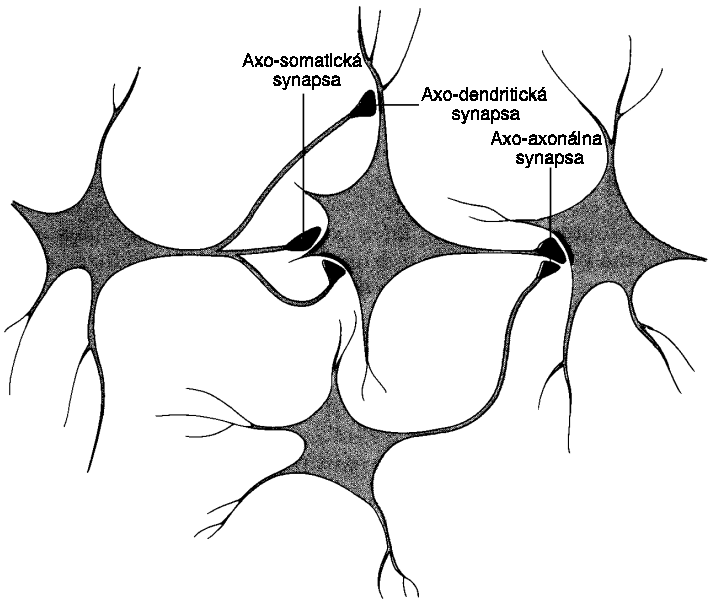


Obrázok 1.6. Myelínová pošva je na mnohých miestach prerušená. Tieto tzv. Ranvierove zárezy umožňujú saltatórne vedenie vzruchu (podľa [24]).

1.4 Synapsa

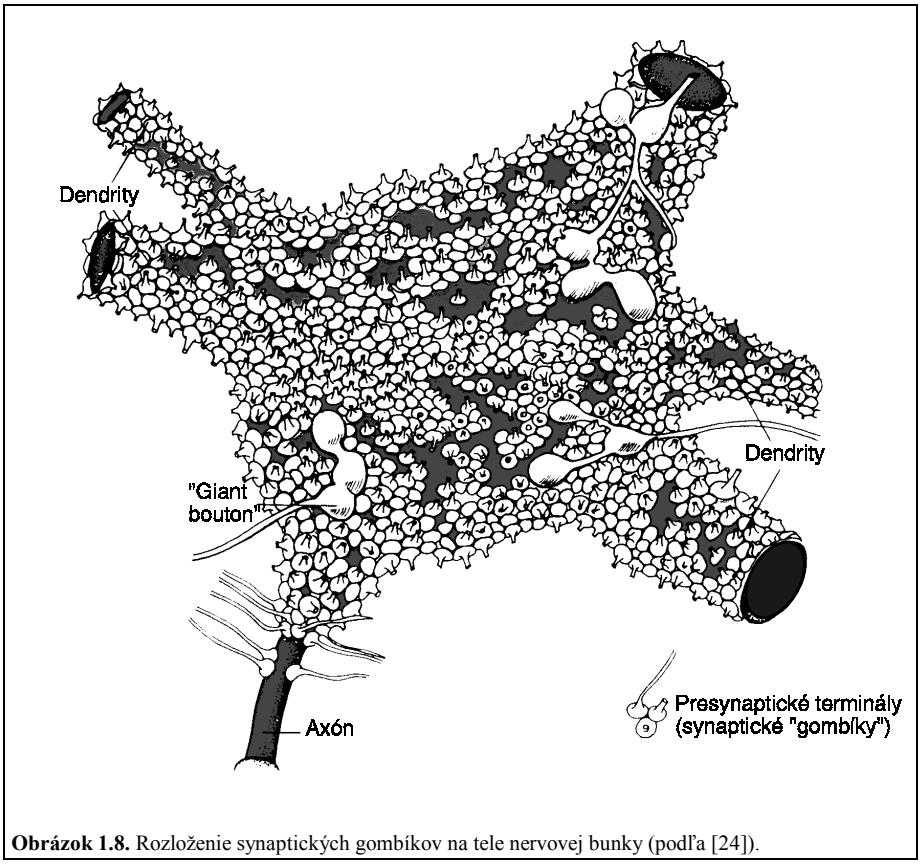
Santiago Ramón y Cajal[5] ako prvý dokázal, že nervový systém sa skladá z dobre ohraničených buniek (neurónová teória). Neuróny vytvárajú funkčné spojenia v mieste tesného kontaktu axónu jednej bunky s membránou inej bunky. Tieto spojenia sa nazývajú *synapsy* (termín synapsa ako prvý zaviedol Charles Sherrington od slova *synapto* - lat. tesne sa objímať). Synapsy, podľa toho ktoré časti nervových buniek ich tvoria, existujú *axo-dendritické*, *axo-somatické* a výnimočne aj *axo-axonálne* (obr. 1.7; axo-axonálne synapsy majú zvláštnu funkciu, o podrobnostiach pozri napr. [24]). Na jednom neuróne sú stovky, tisícky a na niektorých dokonca mnoho desiatok tisíc synáps (obr. 1.8).

[5] Cajal objavil spôsob, ako na histologickom preparáte zafarbi (a tým zvidite ni) nervové bunky. Vykonal doteraz najrozsiahlejšiu štúdiu na nervovom systéme. Polo il základy všetkých neurovied.



Obrázok 1.7. Typy spojení nervových buniek.

Synapsa sa skladá z troch častí: z *membrány presynaptického terminálu* (rozšírené zakončenie axónu), *synaptickej štrbiny* a *postsynaptickej membrány*. Po prechode vzruchu cez presynaptický terminál sa otvárajú napätím riadené kalciové kanály. Kalcium prúdi pozdĺž koncentračného gradientu do bunky. Naviaže sa na bielkoviny, ktoré spájajú *synaptické vezikuly* (membránou ohraničené telieska transportujúce látky z tela neurónu, kde sú syntetizované, k presynaptickým terminálom) s membránou presynaptického terminálu. Po naviazaní kalcia sa tieto bielkoviny kontrahujú, čím spôsobia priblíženie a nakoniec spojenie membrány vezikuly s bunkovou membránou. Tak sa vyprázdňuje obsah vezikuly do synaptickej štrbiny. Vezikuly obsahujú látku, nazvanú *neuromediátor* (angl. *neurotransmitter*). Táto má schopnosť naviazať sa na receptor iónových kanálov postsynaptickej membrány. Po ich otvorení (podľa charakteru kanálu) sa uskutočňuje buď *depolarizácia* (posun membránového potenciálu k pozitívnym hodnotám) alebo *hyperpolarizácia* (posun membránového potenciálu k negatívnym hodnotám v porovnaní s pokojovým potenciálom).

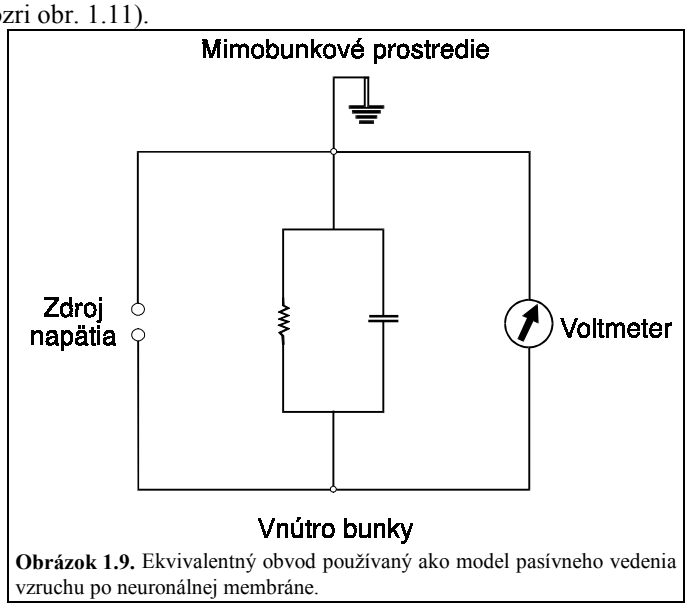


Obrázok 1.8. Rozloženie synaptických gombíkov na tele nervovej bunky (podľa [24]).

1.5 Fyziológia neurónu

Depolarizácia sa uskutočňuje pri otvorení kanálov priepustných pre sodík a kalcium (prúdia pozdĺž elektrického a koncentračného gradientu do bunky, a keďže ide o kladné ióny, znižujú záporný vnútrobunkový potenciál). Hyperpolarizácia nastáva pri otvorení draslíkových kanálov (ide o kladne nabitý ión prúdiaci pozdĺž svojho koncentračného gradientu z bunky von, čím zvyšuje zápornosť membránového potenciálu). Otvorenie chloridových kanálov vedie k fixovaniu membránového potenciálu na hodnote pokojového potenciálu, preto je dôsledok podobný hyperpolarizácii, hoci hyperpolarizácia fakticky nenastáva (podrobnosti v [14], str. 160). Existujú podobné efekty spôsobené zatvorením "leakage" kanálov.

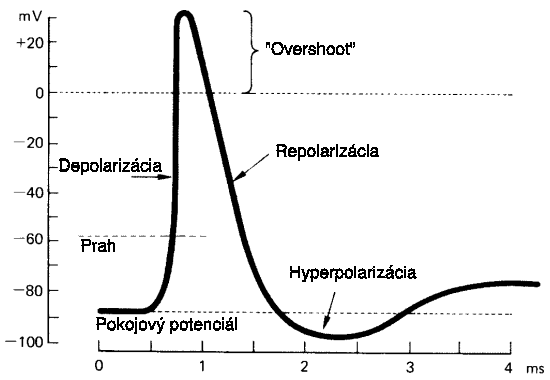
Zmena membránového potenciálu počas depolarizácie postsynaptickej membrány sa označuje ako *excitačný postsynaptický potenciál* (EPSP). Táto zmena sa ďalej po membráne dendritov a tela neurónu šíri "pasívne" (pasívne elektrické vlastnosti bunkovej membrány možno modelovať pomocou tzv. *ekvivalentného obvodu* znázorneného na obr. 1.9). Pri pasívnom šírení vzruchu je totiž množstvo napätím riadených kanálov otvorených takouto depolarizáciou (pomerne malej amplitúdy) malé. Amplitúda EPSP sa preto počas postupu po dendritoch a some zmenšuje (vedenie vzruchu s *útlmom* (*dekrementom)*, podobne ako vedenie inými vodičmi s nenulovým odporom, pozri obr. 1.11).



Obrázok 1.9. Ekvivalentný obvod používaný ako model pasívneho vedenia vzruchu po neuronálnej membráne.

Na iniciálnom segmente (ako aj na samotnom axóne) sa nachádza veľké množstvo napätím ovládaných sodíkových kanálov. Preto je prúd iónov do bunky vznikajúci otvorením týchto kanálov najväčší práve v týchto miestach nervovej bunky. Ak zmena membránového potenciálu na iniciálnom segmente dosiahne tzv. *kritický potenciál* (*prah excitácie*, zvyčajne okolo -60 mV), prevýši prúd sodíka do bunky zvýšené prúdy v draslíkových "leakage" kanáloch z bunky von. Vzniká ďalšia depolarizácia, ďalší vtok sodíka do bunky. Vzniká samo sa šíriaca depolarizácia. Membránový potenciál dosiahne až pozitívne hodnoty. Dva faktory limitujú ďalšie zosilňovanie depolarizácie: Tzv. *inaktivácia* sodíkových kanálov (ich zatvorenie; v tomto stave sa však na rozdiel od pôvodného zatvoreného stavu nemôžu na krátky čas znovu otvoriť) daná veľkosťou depolarizácie a masívne otvorenie napätím ovládaných draslíkových kanálov (ktoré nastáva s istým oneskorením). Draslík, prúdiaci z bunky von, pôsobí na membránový potenciál opačne ako sodík (keďže pozdĺž koncentračného gradientu opúšťa bunku a nesie kladný náboj, má tendenciu vyvolať hyperpolarizáciu). Membránový potenciál sa v dôsledku týchto dvoch faktorov vracia na pôvodnú hodnotu (*repolarizácia*). Súhrou opísaných dejov vzniká svojím časovým priebehom typický *akčný potenciál* (*nervový vzruch,* angl. *spike*) (obr. 1.10)[6]. Pôvodné koncentrácie sodíka a draslíka sa obnovia činnosťou sodíkovo/draslíkových púmp (ktoré presúvajú sodík z bunky von výmenou za molekuly draslíka smerom dnu; ide o aktívny dej, pri ktorom sa spotrebúva veľa energie).

[6] Za opísanie mechanizmu vzniku akčného potenciálu dostali Alan Hodgkin a Andrew Huxley Nobelovu cenu.



Obrázok 1.10. Tvar akčného potenciálu (popis v texte). Podľa [24].

Akčný potenciál postupuje od somy smerom k periférii axónu. Späť sa nešíri, lebo bunková membrána je po prechode akčného potenciálu v tzv. *refraktérnej fáze* (nevedie akčný potenciál). Jej iónové kanály sú vo fáze inaktivácie, ktorá síce trvá iba veľmi krátko, ale dostatočne dlho na to, aby znemožnila opačné (retrográdne) vedenie akčného potenciálu. Keď akčný potenciál dosiahne presynaptický terminál, aktivujú sa vyššie opísané mechanizmy vylúčenia neuromediátora[7] [16].

Jediný excitačný postsynaptický potenciál (EPSP) zvyčajne nestačí na dosiahnutie prahu na iniciálnom segmente. Akčný potenciál sa "nespustí", axón zostáva v pokoji. Treba, aby sa viacero EPSP *stretlo* na iniciálnom segmente súčasne a aby sa ich efekt sčítal. Takéto EPSP môžu buď pochádzať z rôznych synáps alebo z tej istej synapsy, ak je aktivovaná rýchlo za sebou. V prvom prípade hovoríme o *priestorovej sumácii*, v druhom o *časovej sumácii*. Z toho vyplýva, že ak vznikne hyperpolarizácia postsynaptickej membrány, zmenší sa pravdepodobnosť dosiahnutia prahu na iniciálnom segmente, pretože sa táto hyperpolarizácia odčíta od depolarizácie spôsobenej EPSP. Takéto hyperpolarizačné potenciály sa preto nazývajú *inhibičné postsynaptické potenciály* (IPSP).

1.6 Kódovanie v nervovom systéme

Ak by sme zostali pri tomto opise, vznikol by chybný dojem, že neurón je binárny výpočtový element. Tak to však nie je. Na some neurónu sa sumujú mnohé EPSP a IPSP. Podľa veľkosti dosiahnutej sumárnej depolarizácie na tele neurónu iniciálny segment spúšťa *balík (sériu)* akčných potenciálov. Procesy na iniciálnom segmente spojené s generovaním akčného potenciálu neovplyvnia procesy na tele neurónu. Na some je depolarizácia prevažne daná receptorom ovládanými kanálmi na postsynaptickej membráne, ktoré nie sú citlivé na zmenu membránového napätia. Depolarizácia na some preto pretrváva a spúšťa ďalšie a ďalšie akčné potenciály. Podobne aj zmyslové analyzátory (zrakový, sluchový, hmatový, chuťový a čuchový analyzátor) kódujú intenzity podnetov *frekvenčne* (čím vyššia intenzita senzorického stimulu, tým vyššia frekvencia generovania akčných potenciálov - angl. *firing rate* - v príslušnom neuróne).

Okrem frekvenčného kódu existuje aj tzv. *anatomický kód*. Ak sa uskutoční dotyk v *istej* oblasti kože, podráždia sa *isté* hmatové receptory[8], a tie spôsobia aktivitu v *istých* neurónoch. Tá sa potom šíri *istou* dráhou až do mozgovej kôry, kde opäť *istým* miestam povrchu tela zodpovedajú *isté* oblasti kôry. Vzniká akési *topologické* členenie (*mapovanie*) aktivity v nervovom systéme. Aktivita *daného neurónu* zodpovedá intenzite podráždenia *jemu korešpondujúcej oblasti kože*. Charakter podnetu je v tomto prípade kódovaný miestom, ktoré je v nervovom systéme podráždené. Anatomický princíp kódovania platí aj pre ostatné senzorické systémy. V niektorých prípadoch sa aj frekvenčný kód zrejme mení na anatomický (podrobnosti pozri v [17]).

---

[7] Okrem tejto klasickej neurotransmisie poznáme aj *neuromoduláciu*. Pri tomto procese je receptorom aktivovaný nie iónový kanál, ale bielkovina vedúca k aktivácii enzýmov, ktoré nakoniec dlhodobo zmenia excitabilitu neurónu cez zmenu štruktúry bielkovín tvoriacich iónové kanály.

[8] Termín receptor sa ako vidno používa v dvoch významoch. V tu použitom význame je receptor bunka, ktorá kóduje podnet z prostredia a dáva vznik nervovému signálu.

Nervové bunky majú aj v pokojových podmienkach istú aktivitu (tzv. *spontánna aktivita*). Neuróny vtedy generujú akčné potenciály zriedkavo, ale nie sú elektricky "nemé". Na spontánnej aktivite sa v rozhodujúcej miere zúčastňujú zmyslové orgány (receptorové bunky), je však čiastočne aj vlastnosťou samotných nervových buniek. Spontánna aktivita predstavuje vnútorný nervový šum, ktorý môže mať význam aj v neuroinformatike (vyhýbanie sa lokálnym minimám, napr. v Hopfieldovom modeli alebo v Boltzmannovom stroji [23]).

1.7 Formálny neurón

Modely neurónu sú zväčša abstrakciou mechanizmu, ako nervové bunky spracúvajú informácie. Nie je možné vytvoriť presnú analógiu "výpočtových" schopností skutočného neurónu, a to hneď z viacerých dôvodov:

(a) Výpočtové procesy na neurónoch sú enormne komplexné, zahŕňajú asynchrónne procesy a dva principiálne odlišné mechanizmy prenosu vzruchu: neurotransmisiu s krátkodobým ovplyvnením postsynaptického neurónu a neuromoduláciu s dlhodobým ovplyvnením neurónu. Jeden neurón obsahuje aj desiatky tisíc synáps, pričom pre ich efekt je významná ich lokalizácia. Synapsy uložené bližšie k iniciálnemu segmentu majú väčší vplyv na dosiahnutie prahu excitácie ako vzdialenejšie. Pri aktivácii susedných synáps vznikajú na membránach zložité málo prebádané fenomény ("skraty"). Väčšina týchto procesov má nelineárny charakter. Vzhľadom na zložitosť neurónu existuje “trade-off“ medzi zložitosťou modelu neurónu a veľkosťou siete. Veľkosť siete je z výpočtového hľadiska veľmi významný parameter. Zložitosť formálneho (modelového) neurónu komplikuje aj matematickú analýzu správania sa siete.

(b) Fyziológia neurónu zostáva v mnohých otázkach predmetom výskumu. Zatiaľ nie je možné jednoznačne opísať funkciu neurónu ako výpočtového elementu.

(c) Otázky učenia neurónových sietí nie sú v nervovom systéme uspokojivo vysvetlené.

Východiskom pre možnú koreláciu výsledkov medzi neuroinformatikou (angl. *computational neuroscience*) a neurofyziológiou je existencia analógie medzi modelom a skutočným neurónom.

Predstavme si neurónovú sieť vo všeobecnosti podobnú tým, ktoré uvádzame v ďalších kapitolách. Nech ide o synchrónnu sieť, zloženú z K vrstiev po N formálnych neurónoch. Nech o_j^k reprezentuje výstup z j-teho formálneho neurónu v k-tej vrstve ($o \in \langle 0, u \rangle$, kde u je analógia hornej hranice frekvencie akčných potenciálov - fyziologicky opodstatnenou hranicou je menej ako 1000 Hz). Nech w_{ij}^k označuje váhu spojenia medzi j-tym formálnym neurónom k-tej vrstvy a i-tym formálnym neurónom k+1-vej vrstvy ($w \in \mathbf{R}$). Potom je *výstup i-teho formálneho neurónu* k+1-vej vrstvy daný vzťahom

$$o_i^{k+1} = f\left(\sum_{j=1}^{N} w_{ij}^k \cdot o_j^k - \vartheta_i^{k+1} \right) \tag{1.1}$$

pričom ϑ predstavuje *prah excitácie* formálneho neurónu ($\vartheta \in \mathbf{R}$) a $f(x)$ je *vstupno-výstupná aktivačná funkcia* (napr. sigmoidálna funkcia, jej saturačná hodnota odpovedá u, pozri napr. obr. 5.5 alebo 8.6). Pripomíname, že každý formálny neurón k-tej vrstvy nemusí byť nevyhnutne spojený s i-tym prvkom k+1-vej vrstvy (v tom prípade je príslušná váha rovná nule). Pri takomto zjednodušení funkcie neurónu na jednoduchý nelineárny výpočtový element (angl. *processing element*) je analógia k realite založená na nasledujúcej korešpondencii:

Neurón	***Formálny neurón***
synaptická účinnosť	váha spojenia (w)
frekvencia excitácie neurónu	výstupná hodnota (o)
postsynaptický potenciál	$w \times o$
celkový potenciál na tele neurónu	$\sum w \times o$
prah excitácie	ϑ

Ak ide o zložitejší výpočtový element, možno nájsť viac korešpondencií. Fyziológii najbližším modelom, ktorý bol publikovaný, je *MacGregorov výpočtový element* [20].

Existujú aj zložitejšie modely, ktoré priamo simulujú kinetiku zmien na membráne neurónu podľa rovníc Hodgkina a Huxleyho [12,11]. Pokiaľ chceme simulovať väčšiu skupinu neurónov, je výpočtová náročnosť týchto modelov veľká, preto sú prevažne realizované na superpočítačoch a ich cieľom je vysvetlenie biofyzikálnych skutočností vo fyziológii neurónu a v jej ovplyvnení.

1.8 Synaptická plasticita

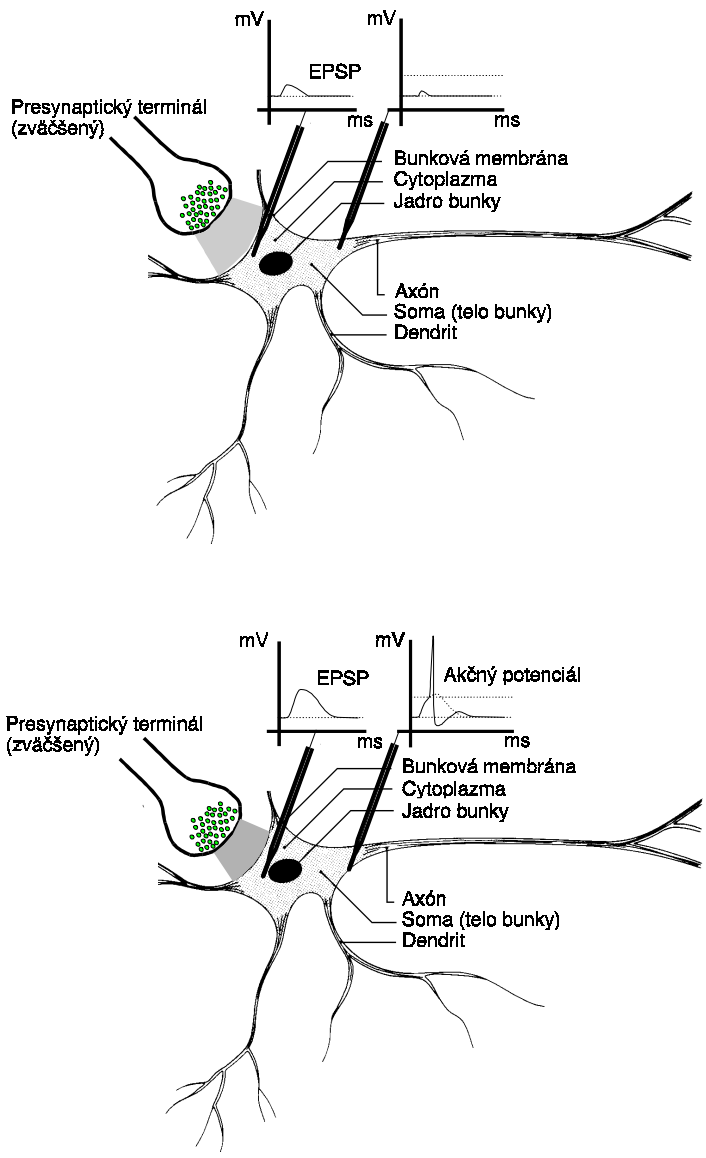
Zmeny v správaní sa živočíchov sú založené na zmenách v nervovom systéme. Donald Olding Hebb v roku 1949 [10] formalizoval hypotézu Cajala z r. 1911 [4], že plasticita v nervovom systéme je založená na zmenách efektívnosti prenosu vzruchu cez synapsu (čiže v zmenách tzv. *synaptických účinností*). Hebb tvrdil, že zmena synaptickej účinnosti je závislá od súčasnej aktivity presynaptického a postsynaptického neurónu (ich súčin, vynásobený tzv. *faktorom rýchlosti učenia*, udáva hodnotu, o ktorú je potrebné pôvodnú účinnosť zmeniť). Formálne možno *Hebbovo pravidlo* vyjadriť v tvare

$$\Delta w_{ij} = \eta \, o_i^{k+1} o_j^k \tag{1.2}$$

kde η je tzv. *rýchlosť učenia* (angl. *learning rate)* a významy ostatných symbolov sú také isté ako v rovnici (1.1). Toto Hebbovo pravidlo, mnohokrát modifikované (či vo forme s učiteľom - Widrowov-Hoffov variant, alebo bez neho - napr. Grossbergov variant [23], sa stále pokladá za všeobecné pravidlo učenia v nervovom systéme [7].

Dostávame sa k otázke, ktoré neurofyziologické procesy zodpovedajú za zmeny synaptickej účinnosti, a teda za *synaptickú plasticitu*. Ako prví takéto zmeny dokázali Tim Bliss a Terje L∅mo v roku 1973 [3]. Dokázali dlhodobé zvýšenie synaptickej účinnosti po opakovanej aktivácii synapsy (*dlhodobá potenciácia*). Až takmer o 10 rokov neskôr sa podarilo jednoznačne dokázať opačný proces, dlhodobé zníženie synaptickej účinnosti (*dlhodobý útlm*, angl. *long-term depression*), a to Masaom Itom a spolupracovníkmi [13] (prehľad v [18]). Je dosť málo objasnené, aké mechanizmy sú zodpovedné za dlhodobý útlm. Viac sa vie o procesoch potenciácie, ktoré sa pokúsime objasniť v ďalšom texte.

Zvýšenie synaptickej účinnosti sa dosahuje dvoma principiálne odlišnými mechanizmami: presynaptickými a postsynaptickými (obr. 1.10). Pravdepodobne je potrebná istá časová následnosť týchto procesov, aby sa vzájomne potenciovali (umocňovali) a dosiahli dlhodobú zmenu synaptickej účinnosti. V opačnom prípade je zvýšenie synaptickej účinnosti len krátkodobé, vzniká tzv. *krátkodobá potenciácia* (angl. *short-term potentiation*).



Obrázok 1.11. Efekt dlhodobej potenciácie (angl. *long-term potentiation*). A. Pred potenciáciou vedie vylúčenie neuromediátora len k podprahovej depolarizácii na iniciálnom segmente, akčný potenciál sa nespúšťa. B. Po potenciácii sa do synaptickej štrbiny vylučuje viac neuromediátora. Postsynaptická membrána sa stáva "senzitívnejšou" na neuromediátor. EPSP má preto väčšiu amplitúdu a vedie (napriek dekrementu) k nadprahovej depolarizácii na iniciálnom segmente (na to však zvyčajne nestačí aktivácia jedinej synapsy). Vzniká akčný potenciál. Proporcie obrázku sú z dôvodu ilustratívnosti zmenené a nezachytávajú skutočné pomery (predovšetkým presynaptický terminál so znázornenými vezikulami je zväčšený).

1.8.1 Presynaptické mechanizmy

V ostatných rokoch bolo objavených viacero spôsobov, ako môže byť zvýšená synaptická účinnosť. Jeden z nich je zvýšenie množstva neuromediátora, uvoľneného do synaptickej štrbiny po príchode akčného potenciálu na presynaptickú membránu. Tento proces je zrejme spôsobený zvýšením koncentrácie tzv. cyklického guanozínmonofosfátu (cGMP) v presynaptickom termináli. Jeho syntéza sa môže zvýšiť po aktivácii synapsy nasledujúcim spôsobom:

Pri opakovanej aktivácii synapsy sa v postsynaptickej membráne otvárajú tzv. NMDA receptorom ovládané kalciové kanály. Kalcium má funkciu tzv. druhého posla (angl. *second messenger*), ktorý odovzdá signál z mimobunkového prostredia dovnútra bunky. Aktivuje množstvo enzýmov, jedným z nich je enzým tvoriaci oxid dusnatý (NO). NO ľahko prechádza membránami, a dostáva sa do presynaptického terminálu, kde

zvyšuje produkciu cGMP. Tým sa zvýši množstvo uvoľneného neurotransmitera po príchode jedného akčného potenciálu. Takýchto tzv. *retrográdnych poslov* je viacej, princíp ich funkcie je podobný ako pre NO [6,8].

1.8.2 Postsynaptické mechanizmy

Pri vtoku kalcia do postsynaptického neurónu sa aktivujú aj *kinázy*, čo sú enzýmy, ktoré menia molekuly neuromediátorom otváraných iónových kanálov, predovšetkým sodíkových kanálov (napr. tzv. non-NMDA kanály). Dôsledkom toho je väčší prúd sodíka vznikajúci po otvorení takéhoto kanálu. Tým sa zvýši amplitúda EPSP, a teda aj odpoveď postsynaptického neurónu na vylúčenie neuromediátora do synaptickej štrbiny. Kinázy sú pravdepodobne zodpovedné za dlhodobosť dlhodobej potenciácie synaptickej účinnosti. Sú to totiž molekuly, ktoré okrem zmeny iónových kanálov dokážu aktivovať aj samy seba zo svojich neaktívnych predchodcov (prekurzorov; hovoríme o *autokatalýze*). Všetky molekuly buniek sú súčasťou metabolickej premeny (angl. *metabolic turnover*). Molekuly sa enzymaticky degradujú a nové sa syntetizujú. Každá molekula má teda istú dĺžku života (meranú biologickým polčasom). Táto doba života by predstavovala limit pre dĺžku efektu aktívnych kináz a teda aj trvanie "pamäti". Avšak po ich prepnutí do aktívneho stavu, tieto kinázy autokatalyzujú novosyntetizované kinázy, a tak aktívny stav pretrváva dlhšie ako by dovoľovala ich individuálna životnosť. Autokatalýza predchádza zániku účinku dlhodobej potenciácie synaptickej účinnosti v dlhodobom meradle (prehľad v [19]).

Túto stať môžeme uzavrieť konštatovaním, že aktivita presynaptického terminálu (vedúca k vylúčeniu neuromediátora) spojená s depolarizáciou postsynaptickej membrány vedie k otvoreniu NMDA receptorom ovládaných kanálov. V konečnom dôsledku sú práve NMDA receptory zodpovedné za zvýšenie synaptickej účinnosti. V tom vidíme súlad s hebbovským učením (musí dochádzať k synchronizácii presynaptickej a postsynaptickej aktivity). Veľmi dobrú korešpondenciu medzi známymi experimentálnymi údajmi (dokonca aj mechanizmami dlhodobého útlmu) a správaním modelu vykazuje tzv. BCM model (nazvaný podľa autorov *Bienenstocka, Coopera a Munroa*[9] [1,2]).

Na záver je potrebné zdôrazniť, že uvedené mechanizmy dlhodobej potenciácie synaptickej účinnosti sú stále len hypotetické. Boli navrhnuté rôzne iné alternatívy. Neuróny nie sú uniformné výpočtové elementy: na rôznych miestach v nervovom systéme majú iné vlastnosti. Okrem rozdielnosti v morfológii (tvare), sú rôzne aj čo sa týka distribúcie iónových kanálov (ako napätím ovládaných tak aj receptormi ovládaných), vylučujú rôzne neuromediátory, a pravdepodobne existujú aj rozdielne mechanizmy synaptickej plasticity. Ide teda o veľmi rôznorodú populáciu buniek. Pochopenie ako mozog pracuje ešte stále nie je na dosah ruky.

Literatúra

[1] A. Artola, W. Singer. Long-term depression of excitatory synaptic transmission and its relationship to long-term potentiation. *Trends Neurosci*., 16(11): 480-487, 1993.

[2] E.L. Bienenstock, L.N Cooper, P.W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci*., 2: 32-48, 1982.

[3] T.V.P. Bliss, T. LØmo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *J. Physiol.* (London), 232: 331-356, 1973.

[4] S.R. Cajal. *Histologie du Systeme Nerveux de l´Homme et des Vertebrés.* Paris: Malone, Vol. 2, 1911.

[5] J.W. Clark, J. Rafelski, J.V. Winston. *Brain without mind: computer simulations of neural networks with modifiable neuronal interactions*. Physics Report 1985, University of Cape Town.

[6] A.A. Farooqui, L.A. Horrocks. Involvement of glutamate receptors, lipases and phospholipases in long-term potentiation and neurodegeneration. *J. Neurosci. Res*., 38: 6-11, 1994.

[7] Y. Fregnac. Les mille et une vies de la synapse de Hebb. *La Recherche*, 25: 788-790, 1994.

[8] J. Garthwaite. Glutamate, nitric oxide and cell-cell signaling in the nervous system. *Trends Neurosci*., 14(2): 60-67, 1991.

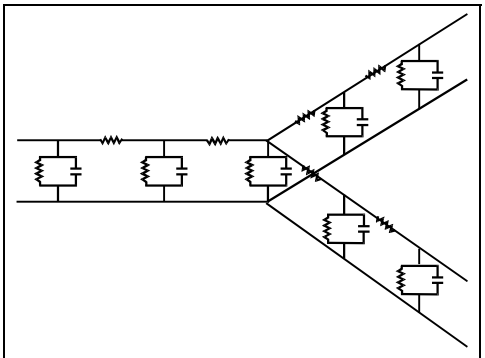
[9] Leon N Cooper je nositeľom Nobelovej ceny za fyziku za práce týkajúce sa supravodivosti.

[9] M.S. Gazzaniga. Principles of the human brain organization derived from split-brain patients. *Neuron*, 14: 217-228, 1995.
[10] D.O. Hebb. *The Organization of Behavior*. J. Wiley and Sons, 1949.
[11] B. Hille. *Ionic Channels of Excitable Membranes.* Sinauer Associates, Inc., 1984.
[12] A.L. Hodgkin, A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol*., 117: 500-544, 1952.
[13] M. Ito, M. Sakurai, P. Tongroach. Climbing fibre induced depression of both mossy fibre responsiveness and glutamate senzitivity of cerebellar Purkinje cells. *J. Physiol.* (Lond.), 324: 113-127, 1982
[14] E.R. Kandel , J.H. Schwartz , T.M. Jessell. *Principles of Neural Science*. Appleton & Lange, 1991.
[15] E. Klika. Histologie, nervová tkáň. V knihe E. Klika, editor: *Histologie*. Avicenum Praha, 183-220, 1985.
[16] J. Koester. Voltage-gated ion channels and the generation of the action potential. V knihe E.R. Kandel, J.H. Schwartz a T.M. Jessell, editori: *Principles of Neural Science*. Appleton and Lange, 104-118, 1991.
[17] A. Kráľ. Úvod do fyziológie zmyslových orgánov. V knihe I. Hulín, editor: *Patofyziológia II*. Slovak Academic Press, 457-459, 1994.
[18] D.J. Linden, J.A. Connor. Long-term synaptic depression. *Annu. Rev. Neurosci.,* 18: 319-357, 1995.
[19] J. Lisman. The CaM kinase II hypothesis for the storage of synaptic memory. *Trends Neurosci*., 17(10): 406-412, 1994.
[20] R.J. MacGregor. *Neural and Brain Modeling*. New York Academic Press, 1987.
[21] R. E. Meyers. Function of corpus callosum in interocular transfer. *Brain*, 79: 358-363, 1956.
[22] K.R. Popper, J.C. Eccles. *Das Ich und sein Gehirn*. R. Pipper & Co. Verlag, Muenchen, 1982.
[23] E.D. Rumelhart, J.L. McClelland. *Parallel Distributed Processing*. MIT Press, 1989.
[24] R.F. Schmidt, editor. *Fundamentals of Neurophysiology.* Springer Verlag, 1985.

2. História neurónových sietí

2.1 Modelovanie nervovej bunky

Jedna z tém v histórii neurónových sietí súvisí s modelovaním samotnej nervovej bunky. V roku 1952 Hodgkin a Huxley na základe meraní generovania a šírenia sa akčného potenciálu v axóne sépie navrhli sústavu diferenciálnych rovníc, ktoré popisujú dynamiku iónových prúdov na aktívnej excitabilnej membráne [23]. Základom týchto rovníc je popis kinetiky membránových iónových kanálov riadených membránovým napätím a výsledkom je veľmi presná výpočtová reprodukcia generovania a šírenia sa akčných potenciálov. Významnou postavou v teoretickom a experimentálnom skúmaní nervového systému je Sir John Eccles [10]. Venoval sa hlavne synaptickému prenosu a popisu iónových prúdov vyvolaných na postsynaptickej membráne pôsobením neuromediátorov. Na popis použil terminológiu elektrických obvodov zložených z rezistorov a kondenzátorov. Matematické štúdie šírenia sa elektrických impulzov v nervových bunkách a ich porovnávanie s elektrickými meraniami na skutočných neurónoch pokračovali v prácach Wilfrida Ralla [47, 48]. Rall aplikoval káblovú teóriu lorda Kelvina na neurón veľmi úspešne, t.j. riešenia rovníc sa zhodovali s meraniami na skutočných neurónoch. V jeho lineárnej káblovej teórii neurónu je soma a každý segment výbežkov bunky (t.j. axónov alebo dendritov) reprezentovaný jednodimenzionálnou RC-linkou s distribuovanými parametrami (obr. 2.1). Analytické riešenie takéhoto systému reprezentujúceho reálny neurón s tisíckami dendritov v dendritickom strome nie je možné a numerické riešenie je veľmi náročné. Pre isté geometrie dendritických stromov navrhol Rall výpočtovo ekvivalentnú reprezentáciu pomocou jedného zužujúceho sa kábla. Butz a Cowan [5] vyvinuli prístup k integrovaniu takéhoto veľkého systému diferenciálnych rovníc s komplikovanými okrajovými podmienkami pre dendritický strom ľubovoľnej geometrie. Využitie tohto prístupu v praxi je vlastne obmedzené len neschopnosťou experimentálne presne zistiť hodnoty odporov a kapacít na všetkých miestach dendritického stromu. Preto sa v praktickom použití káblovej teórie na modelovanie neurónu používajú rozličné aproximácie, v ktorých vystupujú merateľné alebo odvoditeľné parametre. De Schutter [9] podáva prehľad dostupného softwaru na modelovanie jednej nervovej bunky, ako aj malých sietí z nich zložených, na základe káblovej teórie. Okrem iného, práce Ralla a jeho kolegov sú významné v tom, že ukázali, že geometria dendritického stromu determinuje odpoveď neurónu na signály, ktoré prichádzajú od ostatných neurónov. Ukázali, že odpoveď neurónu je vždy iná pre iný priestorovo-časový vzorec aktivácie jeho synáps. Koch a Poggio študovali pomocou káblovej teórie interakcie medzi excitačnými synapsami umiestnenými na dendritických tŕňoch a vypočítali, že účinnosť synapsy je veľmi citlivá na zmenu rozmerov tŕňov [28, 29]. K takýmto zmenám môže dochádzať ako pri indukcii dlhodobej potenciácie synaptickej účinnosti (LTP), tak aj pri prirodzenej stimulácii neurónov [11]. U nás sa káblovej teórii neurónu venovali Dr. Peter Fedor so spolupracovníkmi [14], ktorí navrhli hypotézu



Obrázok 2.1. Schéma káblového modelu časti rozvetvujúceho sa dendritu pomocou navzájom pospájaných jednodimenzionál-nych RC-liniek.

vysvetľujúcu mechanizmus vedúci od stimulácie excitačnej synapsy umiestnenej na dendritickom tŕni ku zmene rozmerov tŕňa.

2.2 Hebbovo pravidlo

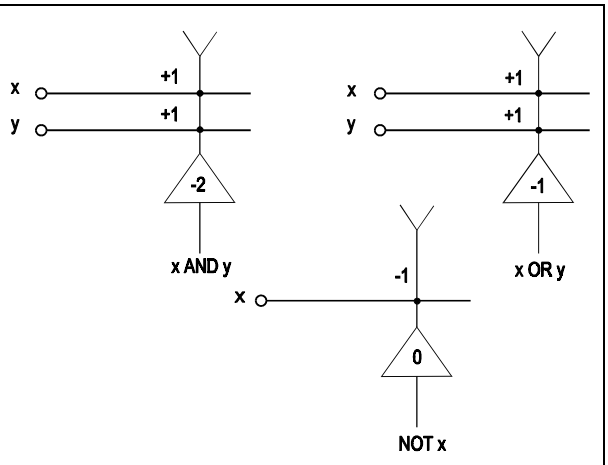
V r. 1949 vyšla kniha kanadského psychológa Donalda Hebba "The Organization of Behavior", jedna z najcitovanejších prác v obore [19]. Hebb v nej navrhol, že účinnosti spojení medzi neurónmi v mozgu sa permanentne menia ako sa jedinec adaptuje a učí nové veci, a to podľa nasledovného pravidla: *"When an axon of cell A ... excite(s) cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells so that A´ s efficiency as one of the cells firing B is increased"* (Keď má axón bunky A excitačný účinok na bunku B, a opakovane alebo stále sa zúčastňuje na jej aktivácii, v jednej alebo v oboch bunkách prebehne nejaký rastový proces alebo metabolická zmena, takže účinnosť bunky A ako jednej z buniek, ktoré aktivujú B, vzrastie). To má za následok, že skupina neurónov, ktoré sú navzájom pospájané, a ktoré sú synchrónne aktivované, bude mať tendenciu vytvárať tzv. bunečné zoskupenie (angl. *cell assembly*), v ktorom sú neuróny pospájané silnejšie. Podstatná invencia Hebbovej predpovede, odvtedy experimentálne nespočetne veľakrát potvrdenej je, že nová informácia je reprezentovaná distribuovane prostredníctvom zmeny účinností mnohých synaptických spojení.

2.3 McCullochov-Pittsov model

V r. 1943 McCulloch a Pitts predstavili prvý model neuronóvej siete [41]. Aplikovali sieť zloženú z tzv. formálnych neurónov na symbolickú logiku, na výroky zložené z elementárnych logických operácií (x AND y, x OR y, NOT x). Obr. 2.2 ilustruje niekoľko príkladov McCullochovych-Pittsovych formálnych neurónov. Tieto formálne neuróny sú vlastne jednoduché logické prepínače, a ich prepínanie prebieha synchrónne a v diskrétnych

časových intervaloch. Hodnoty synaptických váh a prahov sú fixné. McCulloch a Pitts ukázali, že všetky procesy, ktoré sa dajú popísať konečným počtom symbolických výrazov, teda jednoduchá aritmetika, klasifikácia, záznam konečnej množiny dát, rekurzívna aplikácia logických pravidiel a pod., sa dajú realizovať sieťou zloženou z takýchto elementov. Ak je táto sieť nekonečne veľká, tak je výpočtovo ekvivalentná univerzálnemu Turingovmu stroju [41, 52].

V r. 1956 John von Neumann, matematik a významná postava vo vývoji digitálnych počítačov, vyriešil problém spoľahlivosti McCullochovych-Pittsovych sietí v prípade hardwarového poškodenia [44]. Zaviedol redundanciu ~ viacero formálnych neurónov vykonáva tú istú prácu. Tak napr. jeden bit informácie (výber medzi "1" a "0") nie je signalizovaný výstupom jedného logického prepínača, ale synchrónnou aktiváciou mnohých neurónov: "1" dostávame, keď viac ako polovica neurónov je aktívnych a "0", keď je aktívnych menej ako polovica neurónov. Winograd a Cowan [55] zaviedli navyše do neurónovej siete aj distribuovanú reprezentáciu informácie. To znamená, že 1 bit informácie je reprezentovaný redundantne (ako u von Neumanna) a navyše, každý formálny neurón čiastočne reprezentuje mnoho bitov.



Obrázok 2.2. McCullochove-Pittsove formálne neuróny.

2.4 Inšpirácia z teórie spinových skiel

V r. 1954 neuroanatóm Cragg a fyzik Temperley navrhli, že pamäťové stavy v mozgu sú reprezentované oblasťami aktívnych a neaktívnych neurónov [8]. Viedla ich k tomu analógia s tuhými látkami, kde sa môžu striedať domény "dolu" alebo "hore" orientovaných spinov a kde sa vďaka vzájomným interakciám medzi prvkami takéto domény vytvoria a udržia. O 20 rokov neskôr, v r. 1974, prišiel k tejto myšlienkovej konštrukcii aj Little, a to

na základe matematickej analýzy mriežkového spinového systému [34]. Táto línia myslenia bola zavŕšená článkom Johna Hopfielda "Neurónové siete a fyzikálne systémy s emergentnými kolektívnymi výpočtovými schopnosťami", v ktorom reinterpretoval Kirkpatrickovu, Sherringtonovu a Isingovu teóriu spinových skiel do jazyka neurónových sietí a vytvoril tak model robustnej autoasociatívnej pamäti, ktorá spoľahlivo rozpoznáva zašumené vzory aj keď je 80% spojení medzi neurónmi zničených [24, 26, 27]. Tým sa inicioval mohutný prúd prác skúmajúcich neurónové siete pomocou exaktného fyzikálneho aparátu, hlavne z oblasti magnetických systémov [2]. V Hopfieldovych neurónových sieťach je každý neurón spojený s každým symetrickou väzbou. Všetky neuróny sú zároveň vstupom a výstupom siete (viď obr. 8.1). Tieto siete sa neučia, ale ich pamäťové stavy sú predprogramované v matici synaptických spojení, ktorá je skonštruovaná tak, aby pamäťové stavy tvorili atraktory v stavovom priestore. V týchto neurónových sieťach je informácia reprezentovaná distribuovane a redundantne. Neaktívne neuróny sú rovnako dôležité ako aktívne, lebo vlastne konkrétna distribúcia aktivity a "neaktivity" v celej sieti je kódom informácie reprezentovanej v danom okamihu (angl. *coarse coding*). Hopfieldova neurónová sieť funguje ako obsahom adresovaná pamäť (angl. *content addressable memory*), t.j. na základe prítomnosti časti informácie na vstupe dokáže obnoviť celú informáciu.

2.5 Perceptróny

V r. 1958 Frank Rosenblatt ukázal, že McCullochove-Pittsove siete s modifikovateľnými synaptickými váhami sa dajú natrénovať tak, aby vedeli rozpoznávať a klasifikovať objekty [49]. Vymyslel pre ne názov "perceptróny" (viď obr. 5.8). Hlavná myšlienka jeho trénovacej procedúry je takáto: najskôr zaznamenajme odpoveď každého formálneho neurónu na daný podnet. Ak je odpoveď správna, nemodifikujme váhy. Ak je odpoveď daného neurónu nesprávna, potom modifikujme váhy všetkých aktivovaných vstupných synáps, a to nasledovným spôsobom: ak má byť neurón aktívny a nie je, zväčšime ich, a naopak, ak má byť na výstupe neurónu 0 a nie je, zmenšime ich. Rosenblattova trénovacia procedúra je založená na znalosti toho, ktoré vzory (reprezentujúce objekty) patria do ktorej triedy. Jeho idea modifikácie váh spojení na základe korekcie chýb tvorí základ mnohých algoritmov učenia s pomocou učiteľa, ktoré sa používajú dodnes. V r. 1960 Widrow a Hoff použili podobné pravidlo na učenie pre ich model neurónového klasifikátora nazvaného ADALINE (ADAptive LInear NEuron) [54]. Prvýkrát ukázali, že počas učenia sa minimalizuje suma štvorcov chýb, čiže počas učenia sa minimalizuje nejaká globálna funkcia systému.

V r. 1969 Minsky a Papert vo svojej knihe "Perceptróny: úvod do výpočtovej geometrie" poukázali na obmedzenia perceptrónov [42]. Ukázali, že tieto siete vôbec nie sú výpočtovo univerzálne a nedokážu riešiť všetky triedy problémov. Hlavne však išlo o to, že perceptróny nedokážu riešiť tzv. lineárne neseparovateľné problémy. Klasickým najjednoduchším príkladom zlyhania je logická funkcia XOR (vylučujúce alebo). V tom čase nebolo známe žiadne pravidlo učenia (t.j. modifikácie synaptických váh) v umelých neurónových sieťach, nevyhnutné na implementáciu výpočtov tohoto typu. Minsky a Papert usúdili, že bude užitočnejšie, keď sa výskum zameria iným smerom.

Tento problém bol vyriešený až skoro o 20 rokov neskôr, v r. 1986, keď autori Rumelhart, Hinton a Williams zaviedli pravidlo učenia metódou spätného šírenia sa chýb pre viacvrstvové perceptróny (angl. *error back-propagation learning*) [50]. Formálne neuróny v ich perceptrónoch však už nie sú jednoduché logické prepínače McCullochovho-Pittsovho typu, ale analógové elementy so spojitou vstupno-výstupnou funkciou (najčastejšie sigmoidálneho tvaru, pozri napr. obr. 8.6). Hoci to nie je absolútne univerzálny algoritmus schopný naučiť sieť riešiť ľubovoľnú výpočtovú úlohu, predsa len dokáže vyriešiť mnoho lineárne neseparovateľných problémov (vrátane XOR). Veľa úsilia v súčasnej výskumnej a aplikačnej aktivite sa sústreďuje práve na "backprop" a jeho modifikácie. Metóda učenia pomocou spätného šírenia sa chýb je aplikovateľná len na dopredné troj- a viacvrstvové perceptróny, v ktorých neuróny nie sú spojené každý s každým, ale vzájomne nespojené neuróny v rámci jednej vrstvy posielajú spojenia dopredným spôsobom (jednosmerne) ku všetkým neurónom v ďalšej vrstve (pozri obr. 5.3 a obr. 5.17). Stavy neurónov v prvej vrstve predstavujú vstup do siete a stavy neurónov v poslednej vrstve predstavujú výstup siete. Jedno- a viacvrstvové perceptróny tiež reprezentujú informáciu redundantne a distribuovane, a fungujú ako obsahom adresovaná pamäť. Sú schopné učiť sa na príkladoch, a samy nájsť príznaky spoločné prvkom (vzorom) patriacim do tej istej triedy. Extrakcia príznakov prebieha vo vnútornej, tzv. skrytej vrstve neurónov (skrytých vrstiev môže byť aj viac ako jedna). Avšak pri učení je potrebná znalosť toho, ktoré prvky (vzory) patria do ktorej triedy. Až po natrénovaní sú tieto siete schopné zovšeobecňovať, t.j. správne klasifikovať nové vzory. Viacvrstvové siete sú veľmi dobré aj na aproximáciu spojitých funkcií.

Teoretické odvodenie učenia pomocou algoritmu spätného šírenia sa chýb je založené na minimalizácii objektívnej funkcie, ktorá vyjadruje celkovú chybu, t.j. rozdiel medzi želaným a skutočným výstupom siete. Ronald Williams a David Zipser použili metódu gradientovej minimalizácie chybovej funkcie na odvodenie algoritmu spätného šírenia sa chýb v čase na trénovanie rekurentných sietí [53]. V rekurentnej sieti posielajú neuróny v každej vrstve spojenia nielen k nasledujúcim vrstvám, ale aj k predchádzajúcim, a môžu byť pospájané aj medzi sebou (pozri obr. 6.6). Rekurentné spojenia umožňujú uchovať a vyvolať informáciu, ktorá sa vyskytla v minulosti a použiť ju pre súčasný výpočet. To znamená, že rekurentné siete sú dobré na zapamätávanie a predikciu časových postupností vzorov.

Hinton a Sejnowski [22] vymysleli algoritmus učenia pre ľubovoľné, úplne alebo čiastočne rekurentné stochastické siete, ktoré majú symetrické synaptické spojenia. Takéto siete sa dajú považovať za zovšeobecnenie Hopfieldovej neurónovej siete, tak aby táto sieť obsahovala aj skryté neuróny na extrakciu príznakov. Takisto ako v dopredných sieťach, aj tu sa adaptujú hodnoty váh synaptických spojení v dôsledku pôsobenia vstupov a znalosti o tom, ktoré vstupné vzory sú asociované s ktorými výstupnými konfiguráciami aktivity. Keďže takáto sieť má tú vlastnosť, že pravdepodobnostná distribúcia jej možných stavov (konfigurácií aktivity) je totožná s Boltzmannovym rozdelením, nazvali ju jej autori Boltzmannov stroj (angl. *Boltzmann machine*).

V súčasnosti sa skúmajú dopredné a rekurentné siete, ktorých prvky nemajú sigmoidálnu vstupno-výstupnú prechodovú funkciu, ale tzv. radiálnu bázovú funkciu (angl. *radial basis function*, RBF). Najčastejšie je to aktivačná funkcia v tvare gaussiánu [43]. Takáto prechodová funkcia sa veľmi podobá na tvar recepčných polí v reálnom neurónovom

systéme a v mnohých prípadoch RBF siete dosahujú lepšie výsledky ako siete s prvkami, ktoré majú sigmoidálnu prechodovú funkciu.

2.6 Samoorganizácia

Článok Minského a Paperta znamenal veľkú ranu teoretickému výskumu perceptrónov ako univerzálnych výpočtových systémov. Pred objavením algoritmu učenia metódou spätného šírenia sa chýb sa preto výskum v neurónových sieťach začal zameriavať hlavne na učenie bez učiteľa a na systémy, ktoré sú schopné samoorganizácie, t.j. ktoré sú schopné naučiť sa klasifikovať vzory aj bez explicitnej informácie o tom, ktoré vzory do ktorej triedy patria. Neurónová sieť musí objaviť sama prototypy, príznaky, korelácie, kategórie alebo "štatistické pravidelnosti a nepravidelnosti" vo vstupných dátach a zakódovať ich na svojom výstupe. Je zaujímavé, že učenie v týchto typoch neurónových sietí prebieha podľa Hebbovho pravidla a jeho rôznych modifikácií, ktoré však nemenia jeho princíp. Tu spomeňme najmä mená ako von der Malsburg [35], Grossberg [17, 18], Bienenstock so spolupracovníkmi [3], Oja [45, 46], Sanger [51], Linsker [33].

Ojove a Sangerove neurónové siete sú schopné nájsť vo vstupných dátach hlavné komponenty, t.j. smery, v ktorých majú dáta najväčšiu varianciu, a redukovať tak dimenzionalitu mnohorozmerných dát. Sú efektívnymi nástrojmi na štatistickú analýzu hlavných komponent (angl. *principal component analysis*). Grossberg je okrem iného známy aj svojou teóriou adaptívnej rezonancie (angl. *adaptive resonance theory*) a jej implementáciou pomocou neurónovej siete ART1 a ART2 [6, 7]. Von der Malsburg [35], Bienenstock, Cooper a Munro (BCM) [3] a Linsker [33] sa venovali modelovaniu samoorganizácie v skutočných neurónových sieťach zrakovej dráhy v mozgu. Na základe jednoduchých princípov ukázali, ako sa môžu vytvárať recepčné polia neurónov v zrakovej kôre. BCM neuróny a siete z nich zložené sú schopné vyhľadávať projekcie (angl. *projection pursuit*). V tomto procese sa synaptické váhy adjustujú na také hodnoty, ktoré určujú smery (projekcie), v ktorých majú vstupné dáta viacmodálne rozdelenie [25].

Významným príspevkom k teórii samoorganizujúcich sa neurónových sietí sú práce Teuvo Kohonena [30, 31, 32]. Jeho samoorganizujúce sa neurónové siete sa učia bez učiteľa a súťažia medzi sebou o vstupy. Algoritmus učenia je navrhnutý tak, že zobrazenie vstupného priestoru na diskrétnu geometricky usporiadanú množinu umelých neurónov zachováva topológiu (pozri obr.7.8 a obr. 7.26). To znamená, že susedné neuróny odpovedajú najlepšie na susedné vstupy, t.j. vytvorí sa topografická mapa vstupného priestoru. Topografický princíp je jedným z hlavných princípov organizácie spojení v každom senzorickom systéme v skutočnom mozgu.

Zlúčenie dvoch princípov učenia, bez učiteľa a s učiteľom, viedlo k návrhu neurónových sietí s hybridným učením, kde sa prvé vrstvy učia bez učiteľa a vyššie vrstvy sa učia s učiteľom [20, 21]. Takéto siete využívajú prednosti oboch typov učenia a predstavujú systémy vykonávajúce klasifikáciu na základe hierarchických máp príznakov vstupných dát.

U nás sa samoorganizácii a učeniu neurónových sietí bez učiteľa venoval Dr. Peter Fedor [12, 13]. Navrhol vlastný originálny algoritmus na učenie formálnych neurónov založený na Hebbovom pravidle. Svoj model nazval diskriminačný neurón, tzv. D-neurón, pretože v priebehu pôsobenia vstupných vzorov sa jednotlivé D-neuróny "naladia" na diskrimináciu

(rozlíšenie) jednotlivých vzorov, vstupných vektorov. Siete zložené z týchto formálnych neurónov sú schopné rozpoznávať a generovať časové postupnosti vzorov, a to invariantne vzhľadom k trvaniu jednotlivých vzorov a dĺžke prestávok medzi nimi. Okrem toho, dynamicky menia svoju architektúru. Neskôr tieto siete aplikoval na komponovanie jednoduchých melódií [15, 16].

2.7 Späť k mozgu

Modelovanie činnosti nervového systému sa neskončilo pri modelovaní spracovania signálov v jednej nervovej bunke. Z prác, ktoré sa pokúsili biologicky realisticky modelovať a vysvetľovať činnosť mozgu a jeho jednotlivých častí sú najznámejšie a stále rešpektované práce Davida Marra. V roku 1969 publikoval teóriu mozočka (lat., *cerebellum*), v ktorej na základe výpočtovej analýzy neurónových obvodov mozočka vysvetlil, že mozoček funguje ako asociatívna obsahom adresovaná pamäť, ktorú mozog "trénuje" na riadenie a vykonávanie sekvencií zložitých vôľou riadených (voluntárnych) pohybov, takých ako napr. plávanie, hranie na klavír, šoférovanie, a pod. [36, 4]. V Marrovej teórii je každému z piatich hlavných typov neurónov, ktoré sa nachádzajú v mozočku, priradená špecifická funkcia v učení a vyvolávaní vzorcov aktivity. Jeho hypotéza asociatívneho učenia sa výstupných cerebellárnych buniek definovala presne, kedy a ako sa tieto bunky učia, teda kedy a ako sa menia účinnosti synaptických spojení na týchto bunkách. Zatiaľ jeho teória nebola experimentálne ani potvrdená ani vyvrátená, hoci sa zdá, že v skutočnosti pravdepodobne bude správna jej malá modifikácia, ktorú navrhol Albus [1]. Marr aplikoval podobnú analýzu aj na neokortex [37] a na hipokampus (evolučne staršia časť hlavného mozgu, o ktorej sa vie, že hrá veľmi dôležitú úlohu v mechanizmoch pamäti) [38]. O týchto teóriách sa ešte nedá rozhodnúť, či sú správne alebo nie, ale experimenty ich zatiaľ skôr potvrdzujú ako vyvracajú. Týmto sa však Marrov príspevok k modelovaniu činnosti mozgu nekončí. Neskôr sa sústredil na videnie, ako napr. na detekciu hrán objektov a na stereo-videnie (schopnosť vnímať hĺbku, t.j. tretí rozmer, na základe binokulárnej disparity, tak ako to robíme stále a tiež keď sa bavíme pri prezeraní 3D obrázkov, ktoré sa vynárajú z farebných škvŕn) [39, 40]. Marr definoval tri úrovne popisu činnosti centrálneho nervového systému: výpočtovú, algoritmickú a implementačnú. Napríklad vypracovanie výpočtovej úrovne popisu videnia znamená nájdenie optimálnych reprezentácií popisu obrazu v jednotlivých štádiách jeho spracovania, a nájdenie operátorov, pomocou ktorých možno takéto spracovávanie obrazu uskutočniť. Algoritmická úroveň popisu zahŕňa špecifikáciu konkrétnych algoritmov, ktoré by tieto výpočty realizovali. Implementačná úroveň zase špecifikuje "hardware" (biologický alebo iný), ktorý by tieto algoritmy realizoval. Marrov prístup predstavoval, a s malými modifikáciami dodnes predstavuje, akúsi metodickú paradigmu modelovania mozgovej činnosti.

Literatúra

[1] J.S. Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10: 25-61, 1971.

[2] D.J. Amit. *Modeling Brain Function. The World of Attractor Neural Networks*. Cambridge University Press, Cambridge, 1989.

[3] E.L. Bienenstock, L.N Cooper and P.W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2: 32-48, 1982.

[4] S. Blomfield and D. Marr. How the cerebellum may be used. *Nature*, 227: 1224-1228, 1970.

[5] E. Butz and J.D. Cowan. Transient potentials in systems of arbitrary dendritic geometry. *Biophysical Journal*, 14: 661-689, 1974.

[6] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37: 54-115, 1987a.

[7] G.A. Carpenter and S. Grossberg. ART2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26: 4919-4930, 1987b.

[8] B.G. Cragg and H.N.V. Temperley. The organization of neurons: A cooperative analogy. *EEG Clinical Neurophysiology*, 6: 85-92, 1954.

[9] E. De Schutter. A consumer guide to neuronal modeling software. *Trends in Neurosciences*, 15: 462-464, 1992.

[10] J.C. Eccles. *The Physiology of Synapses*. Springer-Verlag, Berlin, 1964.

[11] J.C. Eccles. Synaptic plasticity. *Naturwissenschaften*, 66: 147-153, 1979.

[12] P. Fedor and V. Majerník. A neuron model as an universal element of self-learning networks for pattern recognition. *Biological Cybernetics*, 26: 25-34, 1977.

[13] P. Fedor. Principles of the design of D-neuronal networks. I. A neural model for pragmatic analysis of simple melodies. *Biological Cybernetics*, 27: 129-146, 1977.

[14] P. Fedor, Ľ. Beňušková, H. Jakeš and V. Majerník. An electrophoretic coupling mechanism between efficiency modification of spine synapses and their stimulation. *Studia Biophysica*, 92: 141-146, 1982.

[15] P. Fedor. Principles of the design of D-neuronal networks. I. Net representation for computer simulation of a melody compositional process. *International Journal of Neural Systems*, 3: 65-73, 1992a.

[16] P. Fedor. Principles of the design of D-neuronal networks. II. Composing simple melodies. *International Journal of Neural Systems*, 3: 75-82, 1992b.

[17] S. Grossberg. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23: 121-134, 1976a.

[18] S. Grossberg. Adaptive pattern classification and universal recoding: II. Feedback, Expectation, Olfaction, Illusions. *Biological Cybernetics*, 23: 187-202, 1976b.

[19] D. Hebb. *The Organization of Behavior*. J. Wiley and Sons, New York, 1949.

[20] R. Hecht-Nielsen. Counterpropagation networks, *Applied Optics*, 26: 4979-4984, 1987.

[21] R. Hecht-Nielsen. Applications of counterpropagation networks. *Neural Networks*, 1: 131-139, 1988.

[22] G.E. Hinton and T.J. Sejnowski. Learning and relearning in Boltzmann machines. In: D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. Foundations*, chapter 7, MIT Press/Bradford Books, Cambridge, MA, 1986.

[23] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117: 500-544, 1952.

[24] J.J. Hopfield. Neural systems and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79: 2554-2558, 1982.

[25] N. Intrator and L.N Cooper. Objective function formulation of the BCM theory of visual cortical plasticity: statistical connections, stability conditions. *Neural Networks*, 5: 3-17, 1992.

[26] E. Ising. Bertrag zur Theory des Ferromagnetism. *Zeitschrift fur Physik*, 31:253-287, 1925.

[27] S. Kirkpatrick and D. Sherrington. Infinite-ranged models of spin glasses. *Physical Review B,* 17: 4384-4403, 1978.

[28] C. Koch and T. Poggio. A theoretical analysis of electrical properties of spines. *Proceedings of the Royal Society of London B*, 218: 455-477, 1983.

[29] C. Koch, T. Poggio and V. Torre. Nonlinear interactions in a dendritic tree: Localization, timing, and role in information processing. *Proceedings of the National Academy of Sciences of the USA*, 80: 2799-2802, 1983.

[30] T. Kohonen. An adaptive associative memory principle. *IEEE Transactions on Computers*, C-23: 444-445, 1974.

[31] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43: 59-69, 1982.

[32] T. Kohonen. *Self-Organization and Associative Memory (3rd ed.),* Springer-Verlag, Berlin, 1989.

[33] R. Linsker. From basic network principles to neural architecture. *Proceedings of the National Academy of Sciences of the USA,* 83: 7508-7512, 1986.

[34] W.A. Little. The existence of persistent states in the brain. *Mathematical Biosciences*, 19: 101-120, 1974.

[35] C. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14: 85-100, 1973.

[36] D. Marr. A theory of cerebellar cortex. *Journal of Physiology*, London, 202: 437-470, 1969.

[37] D. Marr. A theory for cerebral neocortex. *Proceedings of the Royal Society of London B*, 176: 161-234, 1970.

[38] D. Marr Simple memory: a theory for archicortex. *Philosophical Transactions of the Royal Society of London B*, 262: 23-81, 1971.

[39] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194: 283-287, 1976.

[40] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B*, 207: 187-217, 1980.

[41] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5: 115-133, 1943.

[42] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, MA, 1969.

[43] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1: 281-294, 1989.

[44] J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: C.E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 43-98, Princeton University Press, Princeton, NJ, 1956.

[45] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15: 267-273, 1982.

[46] E. Oja. Neural networks, principal components, and subspaces. *International Journal of Neural Systems*, 1: 61-68, 1989.

[47] J. Rinzel and W. Rall. Transient response in a dendritic neuron model for current injected at one branch. *Biophysical Journal*, 14: 759-790, 1974.

[48] W. Rall. Core conductor theory and cable properties of neurons. In: E.R. Kandel and Geiger S., editors, *Handbook of Physiology - The Nervous System I: Cellular Biology of Neurons*, volume 3, pages 39-97, American Physiological Society, Bethesda, MA, 1977.

[49] F. Rosenblatt. The Perceptron, a probabilistic model for information storage and organization in the brain. *Psychological Review*, 62: 386-408.

[50] D.E. Rumelhart, G.E. Hinton and R.J. Williams. Learning internal representations by error propagation. In: D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. Foundations*, pages 318-363, MIT Press/Bradford Books, Cambridge, MA, 1986.

[51] T.D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2: 459-473, 1989.

[52] A.M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proceedings of London Mathematical Society*, XLII: 230-265, 1937.

[53] R.J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1: 268-278, 1989.

[54] B. Widrow and M.E. Hoff. Adaptive switching circuits. *WESCON convention record*, IV: 96-104, 1960.

[55] S. Winograd and J.D. Cowan. *Reliable Computation in the Presence of Noise*. MIT Press, Cambridge, MA, 1963.

3. Neurónové siete a umelá inteligencia[♣]

3.1 Symbolický verzus subsymbolický prístup k spracovaniu informácií

Podobne ako v umelej inteligencii, tak aj pri neurónových sieťach vzišla historická inšpirácia zo snahy vytvoriť umelý systém schopný vykonávať komplexné, snáď aj "inteligentné" výpočty podobné rutinným operáciám v ľudskom mozgu. Medzi umelou inteligenciou a neurónovými sieťami je teda správne prepojenie v mysli neprofesionálov, no u ľudí zaoberajúcich sa neurónovými sieťami alebo "klasickými systémami umelej inteligencie" je namiesto prepojenia niekedy aj priepasť. Zásadný problém je v tom, že v začiatkoch umelej inteligencie bolo vnútorné presvedčenie (aj keď iba v podvedomí väčšiny programátorov), že človek je predsa len inteligentnejší ako počítač, a počítač môže prekonať človeka, iba ak sa mu presne povie ako na to. Týmto pohľadom je ťažko otriasť, lebo napriek všetkému úsiliu zatiaľ neexistujú programy (ani neurónové siete!) schopné inteligentne pracovať na viacerých problémoch súčasne a učiť sa nie v umelom prostredí, ale v reálnom svete. Na druhej strane majú biologické neurónové siete o niekoľko rádov viac neurónov. No aj tak je otázny predpoklad, že stačí vytvoriť dostatočne veľký počítač, zaviesť neurónovú sieť, a počítač nadobudne atribúty ľudskej inteligencie.

Tieto, ako aj každé iné tvrdenie spojené s umelou inteligenciou budú najskôr kontroverzné, pretože umelá ako aj prirodzená inteligencia nie je presne definovaným pojmom. Jej základné atribúty by mali zahŕňať schopnosť rozpoznávať vzory, nejaké prvky správania sa metódou pokusu a omylu, a schopnosť učiť sa. Z jedného pohľadu sa ľudia snažia vyvinúť programy riadiace stroje, ktoré by vykonávali to, čo normálne človek nazýva inteligentným správaním sa. Iní išli ešte ďalej, prečo rovno nevymyslieť program, ktorý by sa sám ďalej vyvíjal, aby sa už človek nemusel zaoberať ani tým programovaním. Z druhého pohľadu sa snažia psychológovia, humanisti a neurofyziológovia zistiť, čo je to vlastne tá inteligencia. K tomuto cieľu je najlepšou cestou vytvoriť si umelú inteligenciu a zistiť, či sa "správa" rovnako ako "prirodzená", alebo čo jej chýba, a aké princípy sa zle použili, a na čo sa pozabudlo.

Programy spojené s počiatočným rozvojom umelej inteligencie boli do detailov vymyslené človekom - programátorom, a náhoda v ďalšom vývoji alebo upresňovaní programu nehrala takmer žiadnu úlohu. Človek začal tým, že si predstavil, ako by problém riešil on, a keď sa mu to nedarilo, rozdrobil si problém na časti, ktoré už jednotlivo zvládol. Postup potom naprogramoval do počítača. Automaticky sa predpokladalo, že človek potom dokáže vysledovať, čo vlastne počítač robí, a rozhodnúť, či nejaké dané rozhodnutie počítača bolo správne. Toto presvedčenie sa samozrejme s rastúcou komplikovanosťou programov oslabovalo, no principiálne presvedčenie v podvedomí pretrvávalo. Programy

---

♣ "All things are artificial, for nature is the art of God." Sir Thomas Browne (Všetko je umelo vytvorené, preto□e príroda je Bo□í výtvor.)

boli komplikované z hľadiska množstva námahy vynaloženej programátorom na riešenie konkrétneho typu problému. Výsledný program sa síce správal inteligentne, no bola to inteligencia explicitne vložená programátorom, a program bol schopný riešiť iba veľmi obmedzený typ problémov pri presne definovaných podmienkach.

Postupne sa začala preformulovávať aj samotná definícia toho, čo sa považuje za umelú inteligenciu. V počiatkoch to bol každý program, ktorý sa správal tak, že si laik myslel, že na riešenie takéhoto problému je potrebná inteligencia. Typickým príkladom môžu byť šachové programy, ktoré sú síce na veľmajsterskej úrovni, no ich správanie sa je do detailov určené človekom, a keď urobia chybu, tím analytikov je schopný túto chybu napraviť analýzou rozhodovania programu a pridaním nejakého pravidla alebo zmenou nejakej váhy. Takéto programy sa zo začiatku považovali za umelú inteligenciu, no v súčasnej dobe sa vzhľadom k svojej špecializácii na jeden typ problému už väčšinou za umelú inteligenciu nepovažujú.

Ďalším medzníkom v tvorbe inteligentných programov bola teda tvorba takých systémov, ktoré by boli úspešne schopné riešiť väčšiu triedu problémov. Príkladom sú expertné systémy, ktoré sú tvorené "prázdnym" balíkom programov, v ktorých je zabudovaný logický postup, ako zaobchádzať s pravidlami, no vlastné zadefinovanie konkrétnych pravidiel pre určitý problém je už úlohou užívateľa expertného systému. Expertný systém je teda schopný nielen odpovedať na otázku, no aj dodať užívateľovi prístupnou formou podrobné vysvetlenie, ako sa k odpovedi na nejakú otázku prišlo. Pokiaľ je odpoveď zlá, užívateľ by mal byť schopný "ručne" pozmeniť pravidlá tak, aby nabudúce dali správnu odpoveď. Vo svojej špecializovanej oblasti môžu pravidlá fungovať perfektne, no akonáhle túto oblasť rozšírime, nájdu sa výnimky. Tieto systémy sú ale neobratné pri zahrnutí heuristík alebo výnimiek z pravidiel a neurčitosti zadanej informácie.

Problémom je to, že užívateľ by sa nemal lopotiť s ručným konfigurovaním pravidiel, ale mal by to robiť počítač. Čím viac práce sa ale necháva na počítač, tým je vo všeobecnosti menšia šanca, že by človek bol schopný prekontrolovať postup nejakého rozhodnutia tak, aby rozumel, ako k nemu počítač "dospel". Tento posun od zrozumiteľnosti k nezrozumiteľnosti však niekedy nemusí byť pozvoľný, ako je tomu napríklad v rozdiele medzi sčítaním desiatich čísel a desiatich tisíc čísel, kedy sa hranica možnosti kontroly počítača človekom stáva nezreteľnou. Skokový posun je v rozdiele medzi exaktným "logickým" prístupom a len ťažko formálne popísateľným "analógovým" prístupom, alebo inak povedané, medzi prístupom založenom na

symbolickej logike a medzi subsymbolickým prístupom, kde je výsledok spoluvytváraný príspevkami mnohých hodnôt parametrov neurónovej siete spracovávajúcej informáciu. Prírodovedecká a počítačová komunita bola privyknutá veriť, že všetko, vrátane inteligencie, musí mať jednoduché, kompaktné vyjadrenie v reči symbolov a matematických zákonov. To však nevyzerá byť pravdou pri subsymbolickom prístupe.

Začala sa vynárať nová myšlienka, a to tá, že počítače už fakticky môžu byť lepšie nielen pri riešení čisto numerických problémov alebo pri zreťazení logických úsudkov, no aj v komplikovaných systémoch pri iných problémoch, ktoré doteraz boli doménou človeka. Tu už nestačí narábať so symbolmi alebo celými číslami — reálne čísla a obrazová či akustická informácia je nevyhnutná. Takéto problémy sa typicky skladajú z veľkého množstva drobných interakcií vytvárajúcich celkový pojem či obraz [63]. Vyžaduje sa flexibilita, použitie "podobnosti", kompromisov, analógií, metafor a výnimiek. Človek pritom už nie je schopný preniknúť do činnosti počítača a rozhodnúť, či sa v danom prípade počítač rozhodol správne, alebo nie, a tu problém spočíva nie v množstve dát, ale v ich reprezentácii. Súčasne však takáto architektúra riešenia problému najskôr lepšie zodpovedá procesom v mozgu ako súbor dopredu daných pravidiel a znalostí spojených s rigidným systémom, ktorý určuje, ako s týmito znalosťami zaobchádzať. Zatiaľ ale zďaleka nejde o to, žeby počítač sám od seba vymyslel niečo, čo mu človek dopredu nepovedal. Architektúra v súčasnosti používaných neurónových sietí je zvyčajne presne definovaná človekom pre ten ktorý typ problému. No hranice toho, čo je treba počítaču explicitne povedať, a toho, čo je schopný sám si vypočítať sa pomaly posúvajú. Dá sa iba dúfať, že schopnosť počítačových systémov vytvárať, modifikovať a vysvetľovať hypotézy bude postupne narastať.

Taktiež prístup k tvorbe logického a analógového systému je iný. Pri neurónových sieťach človek ťažko dopredu povie: synaptická váha má byť taká a taká. Je potrebné vytvoriť automatizovanú procedúru učenia sa. Takéto učiace procedúry ale zaberajú veľa strojového času počítača. Aj vzhľadom k tomu, že v dobe začiatkov umelej inteligencie bol čas počítača drahší vo vzťahu k času programátora, ako je tomu dnes, neurónové siete sa vo vývoji oneskorili za heuristickým programovaním založeným na logike. No s rozvojom počítačov tento nepomer začína prevažovať na druhú stranu.

Príklad rozdielu medzi klasickým symbolickým a subsymbolickým spracovaním informácie je prístup k rozpoznávaniu vzorov (angl. *pattern recognition*). Jeden spôsob je jednoducho skladovať všetky možné prípady a každý nový porovnávať so všetkými predchádzajúcimi. Klasické symbolické spracovanie sa pokúša skomprimovať prípady do malého súboru pravidiel, a potom skladované prípady vyhodiť. Neurónové siete sú niekde medzi tým, vo všeobecnosti u nich nemožno jednoznačne rozdeliť naučené pravidlá od informácií vo vzoroch, oboje sú totiž kódované synaptickými váhami a aktiváciami neurónov.

Neurónové siete sú trocha extrémnym príkladom posunu v tvorbe systémov. Namiesto symbolických informácií prenášajú cez komunikačné kanály a spracovávajú vo svojich základných jednotkách numerické informácie. Sú schopné aspoň čiastočne sa učiť, no zatiaľ žiaden systém nevykazuje normálne myslenie, ani na úrovni dieťaťa. Sily spojení medzi neurónmi, tzv. synaptické váhy, predstavujú distribuovanú formu uchovanej informácie. Pokiaľ je neurónová sieť správne natrénovaná, dáva vynikajúce výsledky, no ani dosť podrobnou analýzou nie je človek väčšinou schopný pochopiť, na základe čoho neurónová sieť k výsledku dospela. To považuje komunita klasickej umelej inteligencie za najväčší nedostatok, pretože jej systémy sú schopné zjednodušiť vysvetlenie natoľko, že človek je ho ešte stále schopný sledovať. Neurónové siete fungujú ako čierna skrinka, do ktorej vložíme vstup, vypadne z nej výsledok, no nedostaneme vysvetlenie, prečo akurát takýto výsledok. Jedným zo súčasných trendov výskumu v oblasti neurónových sieti je snaha túto čiernu skrinku "vykradnúť", extrahovať z nej pravidlá a znalosť zakódovanú v numerických koeficientoch siete, a to tak, aby sa tieto pravidlá dali použiť napríklad v expertnom systéme. Pri vyššom počte parametrov je pre expertný systém veľmi ťažké vyextrahovať z distribuovaných príspevkov znalosť, pretože zo štruktúry, ktorá nie je rozčlenená (alebo má veľmi veľa častí, ktoré navyše nie sú rozlíšené), je ťažké vyberať oddelené "kusy" informácií.

Nedá sa povedať, ktorý prístup je lepší, či ten založený na prísnej logike, alebo subsymbolický prístup neurónových sietí. Obidva prístupy majú svoje uplatnenie. Keď vieme, že sa niečo správa na základe zákonov a princípov, ktoré poznáme, a máme dostatok informácií ako určiť súčasný stav, môžeme predpovedať budúce správanie sa a budúce stavy, a vzhľadom na to, že prírodné zákony sú väčšinou priehľadné, dá sa presne sledovať logický reťazec od vstupov do výstupov, od príčin k následkom. Vtedy nemá veľký zmysel používať neurónové siete. Tie vzhľadom k svojej "voľne" definovanej architektúre (v porovnaní so symbolickými systémami) niekedy dosť ťažko hľadajú veľmi zložité vzťahy, ktoré keď už dopredu poznáme, ľahko ich zabudujeme do klasického symbolického systému. Keď sú vstupné informácie neisté alebo nepoznáme presne zákonitosti systému, ktorého správanie sa snažíme predpovedať, vtedy sú neurónové siete vhodnou voľbou. Príkladom takých problémov môže byť klastrovanie, klasifikácia, či rozpoznávanie vzorov. No pritom ani neurónové siete nie sú vo všeobecnosti schopné rozpoznať jednotlivé objekty figurujúce na zložitej scéne (zatiaľ čo v prípade, že je objekt "odseparovaný", ho rozpoznajú). Podobne je tomu pri analýze zložených štruktúr, ako sa nachádzajú vo frázach prirodzeného jazyka.

Minsky [46, 47] navrhuje hybridný systém, ktorý by bol schopný využiť prednosti obidvoch typov systémov, a to tak, že by sa rozhodovalo podľa typu problému, ktorý z prístupov skôr použiť. No aj keď sa pokusy o tvorbu takýchto hybridných systémov objavujú [29], zatiaľ to vždy ešte funguje najlepšie, keď rozhodne človek, ktorým zo špecializovaných prístupov sa bude ten ktorý problém riešiť.

3.2 Oblasti použitia neurónových sietí

Čo sa dá realizovať pomocou neurónových sietí, a čo sa nedá?

V princípe môžu neurónové siete spočítať akúkoľvek spočitateľnú funkciu, t.j. môžu robiť čokoľvek, čo dokáže číslicový počítač. V praxi sa neurónové siete najviac používajú na klasifikáciu a funkčnú aproximáciu alebo mapovanie funkcií pri použití množstva trénovacích dát, kde systémy s jasne stanovenými pravidlami (ako sú expertné systémy) zlyhávajú. Neurónové siete sú pritom tolerantné k neurčitostiam v trénovacích dátach.

Na druhej strane neurónové siete ťažko zvládajú manipuláciu so symbolmi.

Neurónové siete by mali byť teoreticky lepšie a rýchlejšie v porovnaní s väčšinou výpočtov založených na symbolickej logike, pretože sa dajú ľahko rozparalelniť tak, aby bol každý neurón "implementovaný" na jednom procesore. Vzhľadom k tomu, že výpočty jednotlivých prvkov sú na sebe do značnej miery nezávislé, bude celý výpočet o mnoho rýchlejší. Na druhej strane, zatiaľ neexistuje veľa počítačov vybavených veľkým počtom procesorov, takže väčšina aplikácií beží na jednoprocesorových strojoch. Výhoda rozparalelnenia je v súčasnosti teda skôr teoretická.

Medzi najčastejšie oblasti použitia neurónových sietí patria:

- Počítačová informatika — skúmanie vlastností nesymbolického spraco-vania informácie [12, 41, 42, 62];
- Inžinierstvo — automatické riadenie, spracovanie signálov a veľa ďalších aplikácií [9, 10, 26];

  Jednou z prvých komerčných aplikácií bolo (a ešte stále je) tlmenie šumu na telefónnych linkách [61, ADALINE v kap. 2.5]. V oblasti kybernetiky je známa aplikácia v riadení výroby [14, 45], alebo napríklad na cúvanie dlhých trajlerov [50]. Ďalším príkladom je automatické rozpoznávanie ručne písaných poštových smerovacích čísel [37] alebo analýza otlačkov prstov v kriminalistike.
- Finančníctvo — modelovanie vývoja trhu, rozhodovanie pri prideľovaní pôžičiek alebo pri určovaní veľkosti splátok alebo aj overovanie podpisov na šekoch [4, 51, 57];
- Fyzika — modelovanie javov v štatistickej mechanike [11];
- Chémia — predikcia fyzikálno-chemických vlastností zlúčenín [64], riadenie chemickej výroby [9], analýza dát z analytických meracích prístrojov [15], analýza spektroskopických dát — klasifikácia zlúčenín, predikcia spektier, molekulárne modelovanie — predikcia sekundárnej a terciárnej štruktúry proteínov [64];
- Jadrové inžinierstvo [34], jadrová fyzika a spektroskopia [21].
- Biológia — interpretácia nukleových sekvencií [20];
- Medicína — návrh diagnózy na základe príznakov a výsledkov laboratórnych vyšetrení (napr. diagnóza elektrokardiogramov, diagnóza testov na rakovinu, diagnóza sonogramov a röntgenových snímok apod.) [2, 3];
- Štatistika — flexibilná lineárna a nelineárna regresia a klasifikácia [44];
- Neurofyziológia — skúmanie senzorických systémov, motoriky, rozpozná-vanie a produkcia reči [59], modelovanie neurofyziológie mozgu [6, 24, 36, 43, 58];
- Neuropsychológia — modelovanie psychických funkcií [17, 31, 32, 53, 54];
- Ďalšie oblasti ako poľnohospodárstvo (ohodnocovanie a triedenie ovocia), meteorológia (predpoveď počasia), astronómia (klasifikácia galaxií), atď.

3.3 Možné smery vývoja

Medzi nielen možné, ale isté (aj keď neveľmi vzrušujúce) smery rozvoja neurónových sietí patrí väčšia **spolupráca so štatistikou**. Najtypickejší nedostatok, ktorý vytýkajú štatistici mnohým aplikáciám neurónových sietí, je príliš malý počet trénovacích príkladov v pomere k veľkému počtu optimalizovaných parametrov. Čo sa týka súvislosti so štatistikou, väčšina typov neurónových sietí je schopná učiť sa a generalizovať zo zašumených dát, v čom sú podobné štatistike. Napríklad perceptróny sú príbuzné niektorým lineárnym modelom. Neurónové siete s dopredným šírením a jednou skrytou vrstvou sú blízke regresii projekcií. Veľa výsledkov zo štatistickej teórie nelineárnych modelov môže byť priamo aplikovaných na tieto siete. Zatiaľ čo na učenie menších neurónových sietí je výhodnejšia napr. Newtonova metóda [27] alebo metóda konjugovaných gradientov [5], pri väčšom množstve synaptických váh je rozumnejšie použiť Levenberg-Marquardovu metódu [38, 39]. Pravdepodobnostné neurónové siete (PNN) vykonávajú jadrovú (angl. *kernel*) diskriminačnú analýzu. Neurónové siete realizujúce vektorovú kvantizáciu dávajú podobné výsledky ako "k-spriemernená" (angl. *k-means*) klastrová analýza. A neurónové siete učiace sa pomocou hebbovského pravidla sú schopné nájsť smery s maximálnou varianciou v dátach, čo odpovedá analýze hlavných komponent (angl. *principal component analysis*). Kohonenove samoorganizujúce sa mapy sú vzdialene príbuzné "k-spriemer-nenej" alebo skôr "1-spriemernenej" klasterovej analýze.

Problém, ktorý často zabraňuje efektívnej spolupráci so štatistikmi, je v terminológii, ktorú si ľudia zaoberajúci sa profesionálne neurónovými sieťami vyvinuli vlastnú, nezávisle na terminológii v štatistike. Okrem toho ľudia pracujúci s neurónovými sieťami často ignorujú predpoklady týkajúce sa distribúcie dát, s ktorými majú čo do činenia. Pritom sa dajú štatistické výsledky úspešne použiť pri neurónových sieťach. Napríklad, keď niektoré trénovacie vzorky sú viac zašumené, je výhodnejšie použiť váhovanú metódu najmenších štvorcov namiesto klasickej chybovej funkcie [7, 44, 52].

Aj keď nasledujúce dve pravidlá vyzerajú v súvislosti so štatistikou triviálne, treba ich zdôrazniť, aby nedošlo k zbytočným chybám a objavovaniu už objaveného. Pri klasifikácii do viac ako dvoch nezoradených kategórií je treba dať pozor, aby každá kategória mala vlastný výstupný neurón. Napríklad keď sú výstupom tri farby (ktoré pre náš účel nemá zmysel zoraďovať podľa frekvencie v spektre), potom by výsledok siete mal vyzerať takto

Červená	1 0 0
Zelená	0 1 0
Modrá	0 0 1

namiesto

Červená	0
Zelená	0,5
Modrá	1

Keď nám vychádza suma väčšia ako jedna a my chceme sumu výsledkov rovnú jednej, aby jednotlivé výstupy boli interpretovateľné ako "pravdepodob-nosti", potrebujeme, aby

suma pravdepodobností bola rovná jednej. To sa bežne dosahuje použitím tzv. "softmax" aktivačnej funkcie: teda keď výstup bez použitia aktivačnej funkcie by bol q_i pre i-tu kategóriu z celkového počtu c kategórií, potom pravdepodobnosť priradenia výsledku i-tej kategórii bude p_i

$$p_i = \frac{\exp(q_i)}{\sum_{j=1}^{c} \exp(q_j)} \tag{3.1}$$

Keď máme iba dve kategórie, táto funkcia sa redukuje na jednoduchú logistickú funkciu. Podobne je niekedy výhodné normovať alebo štandardizovať vstupné dáta.

V prípade, že potrebujeme klasifikačnú metódu, ktorá nám zistí prípady, v ktorých je klasifikácia neistá, je vhodné použiť PNN (angl. *Probabilistic Neural Network*), čo je termín zavedený Donaldom Spechtom pre jadrovú diskriminačnú analýzu (angl. *kernel discriminant analysis*). Môže sa o nej uvažovať ako o normalizovanej RBF (angl. *Radial Basis Function*) sieti [25, 40, 41, 44, 56].

Podobne funguje aj Spechtom navrhnutá GRNN (angl. *General Regression Neural Network*) [40, 48, 55, 60], čo je alternatívny termín pre Nadaraya-Watsonovu jadrovú regresiu (angl. *kernel regression*). Môže sa považovať za normalizovanú RBF sieť s jednotlivými vnútornými uzlami špecializovanými pre jeden tréningový prípad.

V prípade "preučenia" (angl. *overfitting*) alebo nedostatočnej konvergencie (angl. *underfitting*, bližšie vysvetlenie v podkap. 5.5.3) je niekoľko metód, ako sa týmto extrémom vyhnúť. Výber modelu (angl. *model selection*) sa týka počtu váh, trénovanie "s šumom" (angl. *jittering*) [7], zoslabovanie váh (angl. *weight decay*) [7, 52], včasné zastavenie učenia (angl. *early stopping*) [49] a bayesovský odhad (angl. *Bayesian estimation*) sa týkajú veľkostí váh [7, 22].

Ďalším typickým problémom je odhad chyby pri zovšeobecnení. Najčastejšie používané je rozdelenie dát na trénovaciu a testovaciu množinu (viď podkap. 5.5.1). Často používanou možnosťou je *cross-validation* [7, 22], teda vynechanie podmnožiny vzorov (zvyčajne jedného) zo všetkých prístupných trénovacích dát. Takto môžeme použiť na trénovanie všetky dáta, ktoré máme, no trénovanie musí prebiehať pre všetky možnosti vynechania vzoru (vzorov), teda veľakrát. Ešte lepší odhad za cenu ešte väčšej spotreby počítačového času poskytuje tzv. *boot-strapping* [16, 28, 40].

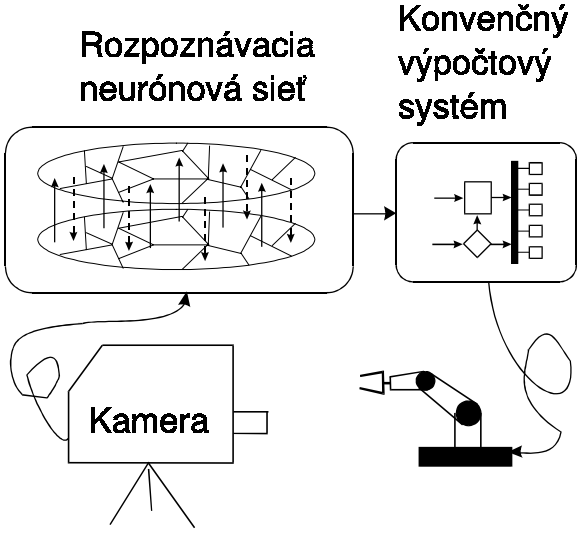
"**Fuzzy" logika**, zavedená v šesťdesiatych rokoch Lotfi A. Zadehom z Berkeley, je zásadným rozšírením klasickej dvojhodnotovej logiky a teórie množín. Používa neostro definované lingvistické premenné (ako "veľký", "horúci", "vysoký") a kontinuálny interval pravdivostných hodnôt [0,1] namiesto striktne binárnych rozhodnutí alebo priradení ("true" alebo "false"). Táto logika sa používa v prípadoch, keď je obtiažne vytvoriť exaktný model skúmaného systému, ale máme k dispozícii aspoň nejaký hrubý model. Takéto systémy boli doteraz riadené ľuďmi — expertmi. Typický fuzzy systém, ktorý má experta nahradiť, pozostáva z databázy pravidiel, funkcie priradenia (angl. *membership function*) a inferenčnej procedúry.

Neurónové siete možno využiť pri konštrukcii priraďovacích funkcií fuzzy množín a pre riešenie fuzzy relačných rovníc. Keď je už zostavený fuzzy riadiaci prvok alebo fuzzy expertný systém, klasická dopredná viacvrstvová neurónová sieť so spätným šírením ho môže nahradiť, využívajúc pritom paralelné spracovanie informácií.

Okrem nahradenia už hotových fuzzy systémov klasickými neurónovými sieťami sú tu pokusy vytvoriť alternatívne neurónové siete, kde vstupy, výstupy a váhy sú fuzzy čísla. Naviac váhované vstupy do každého neurónu nie sú sumované, ale je použitá iná, fuzzy operácia [1, 8, 30, 33, 35].

V súčasnej dobe je veľká časť pozornosti zameraná na to, akým spôsobom by mohli byť symbolický a subsymbolický prístup spojené do jedného celku, ktorý by zahŕňal najlepšie rysy obidvoch prístupov. Základnými témami sú napríklad obraz a symbol — spojité výpočty v reálnych číslach a vynáranie sa diskrétnych hodnôt, extrakcia a vkladanie symbolickej informácie do neurónových sietí a tvorba klasických systémov symbolického spracovania informácie na subsymbolických základoch. Jedným z najpopulárnejších príkladov takýchto snáh je prepojenie neurónovej siete s expertným systémom do **hybridného systému.**

Neurónové siete v súčasnosti trpia rovnakým nedostatkom ako systémy založené na symbolickej logike. Totiž tým, že sú naučené iba na jeden typ problémov. Aby dokázali riešiť väčšie spektrum problémov, je predsa len nevyhnutné nejaké vnútorné rozdelenie pôvodne homogénnej siete. Je niekoľko prístupov, ako spojiť subsymbolický a symbolický prístup [29]. Môžu sa prekrývať, čiastočne prekrývať, bežať paralelne alebo za sebou, alebo byť vnorené jeden do druhého. Zatiaľ najčastejšie spojenie je najskôr sériové, kedy výstupy z jedného systému (najčastejšie neurónovej siete) tvoria vstupy do druhého systému (založeného na symbolickej logike). "Znalostne" založené neurónové siete sú postavené na teórii alebo apriórnych znalostiach o tej ktorej oblasti. Neurónové siete sa starajú o také problémy ako je šum alebo neistá informácia, s ktorou klasické znalostné systémy nie sú schopné zaobchádzať.



Obrázok 3.1. Spojenie neurónovej siete a klasického systému založeného na symbolickej logike.

Inou možnosťou je hybridný systém založený na symbolickej logike, ktorá sa postupne dopĺňa adaptáciou.

Ďalším prístupom je **extrakcia znalostí z neurónovej siete** do expertného systému. To je základ pre vysvetlenie správania sa neurónovej siete a objavovanie znalostí zakrytých šumom a neistou informáciou. Tieto algoritmy sú však exponenciálne náročné vzhľadom k počtu neurónov vo vrstvách [18, 19]. Táto myšlienka môže byť tiež použitá pre zabránenie preučenia siete. Z klasických neurónových sietí s dopredným šírením sa extrahujú pravidlá typu IF-THEN-ELSE a pravidlá formálnych gramatík je možné extrahovať z rekurentných neurónových sietí [13, 23].

Literatúra

[1] H. Adeli and S.-L. Hung. *Machine Learning. Neural Networks, Genetic Algorithms, and Fuzzy Systems*. J. Wiley, NY, 1995.

[2] J.A. Anderson. Cognitive capabilities of a parallel system. In: E. Bienenstock, F. Fogelman-Souli, and G. Weisbuch, editors. *Disordered Systems and Biological Organization*. NATO ASI Series, F20, Springer-Verlag, Berlin, 1986.

[3] časopis Artificial Intelligence in Medicine, Elsevier Press.

[4] M.E. Azoff. *Neural Network Time Series Forecasting of Financial Markets*. J. Wiley, NY, 1994.

[5] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

[6] E. Bienenstock. A model of neocortex. *Network: Computation in Neural Systems*, 6: 179-224, 1995.

[7] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[8] M. Brown and Ch. Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall, NY, 1994.

[9] A.B. Bulsari, editor. *Neural Networks for Chemical Engineers*. Elsevier Press, Amsterdam, 1995.

[10] A.B. Bulsari and S. Kallio, editors. *Engineering Applications of Artificial Neural Networks*. Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN '95). Finnish Artificial Intelligence Society, 1995.

[11] C. Campbell, D. Sherington, and K.Y. Wong. Statistical mechanics and neural networks. In: I. Aleksander, editor. *Neural Computing Architectures*. The MIT Press, Cambridge, MA, 1989, pp. 239-257.

[12] A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing*. J. Wiley, NY, 1993

[13] A. Cleremans, D.S. Ervan-Schreiber, and J. McClelland. Finite state automata and simple recurrent neural networks. *Neural Computation* 1(3): 372, 1989.

[14] I.F. Croall and J.P. Mason, editors. *Industrial Applications of Neural Networks*. Springer-Verlag, Vol. 1 (Project 2092 ANNIE), 1992.

[15] D. Dong and T.J. McAvoy. Sensor data analysis using autoassociative neural nets. *Proceedings of the 1994 World Congress on Neural Networks*, (5-9 June 1994). San Diego, CA, Vol. 1, 1994, pp. 161-166.

[16] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, London, 1993.
[17] M. Enquist and A. Arak. Symmetry, beauty and evolution. *Nature*, 372: 169-172, 1994.
[18] L.M. Fu. Rule learning by searching on adapted nets. In: *Proceedings of AAAI-91* (Anaheim, CA), 1991, pp. 590-595.
[19] L.M. Fu. Rule generation from neural networks. *IEEE Transactions on System, Man and Cybernetics*, 28 (8): 1114-1124, 1994.
[20] L.M. Fu. *Neural Networks in Computer Intelligence*. McGraw-Hill, Singapore, 1994, chapter 16.
[21] S. Gazula, J.W. Clark, and H. Bohr. Learning and prediction of nuclear stability by neural networks. *Nuclear Physics A*, Vol. A540, pp. 1-26, 1992.
[22] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, London, 1995.
[23] C.L. Giles, C.B. Miller, D. Chen, G.Z. Sun, H.H. Chen, and Y.C. Lee. Extracting and learning an unknown grammar with recurrent neural networks. In: *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann, San Mateo, CA, 1992.
[24] M.A. Gluck and D.E. Rumelhart, editors. *Neuroscience and Connectionist Theory*. Lawrence Erlbaum Associates, Hillsdale, 1990.
[25] D.J. Hand. *Kernel Discriminant Analysis*. Research Studies Press, 1982.
[26] S. Haykin. *Neural Networks, A Comprehensive Foundation*. Macmillan, Englewood Cliffs, NJ, 1994.
[27] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA, 1991.
[28] J.S.U. Hjorth. *Computer Intensive Statistical Methods Validation, Model Selection, and Bootstrap*. Chapman & Hall, London, 1994.
[29] V. Honavar and L. Uhr, editors. *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration*. Academic Press, Boston, 1994.
[30] C.H. Chen, editor. *Fuzzy Logic and Neural Network Handbook*. McGraw-Hill, NY, 1996.
[31] R.A. Johntone. Female preferences for symmetrical male as a by-product of selection for mate recognition. *Nature*, 372: 172-175, 1994.
[32] M. Juhola, A. Vauhkonen, and M. Laine. Simulation of aphasic naming errors in Finish language with neural networks. *Neural Networks*, 8: 1-9, 1995.
[33] S. V. Kartalopoulos. *Understanding Neural Networks and Fuzzy Logic: Concepts and Applications*. IEEE Press, CA, 1996.
[34] W.J. Kim, S.H. Chang, and B.H. Lee. Application of neural networks to signal prediction in nuclear power plant. *IEEE Transactions on Nuclear Science*, Vol. 40, pp. 1337-1341, 1993.
[35] B. Kosko. Neural Networks and Fuzzy Systems. Prentice-Hall, Englewood Cliffs, NJ, 1992.
[36] A. Koster, A. Zippelius, and R. Kree. Modelling of the Bonhoeffer-effect during LTP learning. *Network: Computation in Neural Systems*, 5: 259-275, 1994.
[37] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Handwritten digit recognition with a backpropagation network. In: D.S.

Touretsky, editor. *Advances in Neural Information Processing Systems 2*. Morgan Kaufman, San Mateo, CA, pp. 396-404, 598-605.
[38] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math*., 2: 164-168, 1944.
[39] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math*., 11: 431-441, 1963.
[40] T. Masters. *Advanced Algorithms for Neural Networks: A C++ Sourcebook*. J. Wiley, NY, 1995.
[41] T. Masters. *Practical Neural Network Recipes in C++*. Academic Press, Boston, 1994
[42] T. Masters. *Signal and Image Processing with Neural Networks: A C++ Sourcebook*. J. Wiley, NY, 1994
[43] M. Matsugu and A.L. Yuille. Spatiotemporal information storage in a content addressable memory using realistic neurons. *Neural Networks*, 7: 419-439, 1994.
[44] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Chichester, 1994.
[45] W.T. Miller, R.S. Sutton, P.J. Werbos. *Neural Networks for Control*. MIT Press, Cambridge, MA, 1990.
[46] M. Minsky. Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy. In: P. H. Winston, editor. *Artificial Intelligence at MIT, Expanding Frontiers*, Vol 1, MIT Press, Cambridge, MA, 1990. Reprinted in AI Magazine, 1991.
[47] M. Minsky. *The Society of Mind*. Simon and Schuster, NY, 1987.
[48] E.A. Nadaraya. On estimating regression. *Theory Probab. Applic*. 10: 186-90, 1964.
[49] M.C. Nelson and W.T. Illingworth. *A Practical Guide to Neural Nets*. Addison-Wesley, Reading, MA, 1991.
[50] D. Nguyen and B. Widrow. The truck backer upper: An example of self-learning in neural networks. *International Joint Conference on Neural Networks*. Washington, D.C.: II: 357-363, 1989.
[51] A.-P. Refenes, editor. *Neural Networks in the Capital Markets*. J. Wiley, NY, 1995.
[52] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.
[53] D.E. Rumelhart and J.L. McClelland. Learning the past tenses of English verbs. In: D.E. Rumelhart and J.L. McClelland, editors. *Parallel Distributed Procesing*, vol 2, Bradford Books/MIT Press, Cambridge, MA, 1986.
[54] T.J. Sejnowski and C.R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1: 145-168, 1987.
[55] D.F. Specht. A generalized regression neural network. *IEEE Transactions on Neural Networks*, 2, Nov. 1991, 568-576, 1991.
[56] D.F. Specht. Probabilistic neural networks. *Neural Networks*, 3: 110-118, 1990.
[57] R.R. Trippi and E. Turban. *Neural Networks in Finance and Investing*. Irwin Professional Publishing, Chicago, 1993 (1996 2nd eddition).
[58] M. Usher, M. Stemmler, C. Koch, and Z. Olami. Network amplification of local fluctuations causes high spike rate variability, fractal firing patterns and oscillatory local field potentials. *Neural Computation*, 6: 795-836, 1994.
[59] A. Waibel and K. Lee. *Readings in Speech Recognition*. Morgan Kaufmann, CA, 1990.

[60] G.S. Watson. Smooth regression analysis, *Sankhya*, Series A, 26: 359-72, 1964.
[61] B. Widrow and S.D. Stearns. *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1985.
[62] A.S. Weigend and N.A. Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, Reading, MA, 1994.
[63] M. Zeidenberg. *Neural Networks in Artificial Intelligence*. Ellis Horwood, Ltd., Chichester, 1990.
[64] J. Zupan and J. Gasteiger. *Neural Networks for Chemists. An Introduction*. VCH Verlagsgesellschaft, Weinheim, 1993.

Niektoré časopisy týkajúce sa neurónových sietí a umelej inteligencie:

Neural Networks, Pergamon Press
Neural Computation, MIT Press
IEEE Transactions on Neural Networks, Institute of Electrical and Electronics Engineers (IEEE)
Network: Computations in Neural Systems, Institute of Physics (Bristol, UK)
International Journal of Neural Systems, World Scientific Publishing
International Journal of Neurocomputing, Elsevier Science Publishers
Neural Network World, IDG Czechoslovakia
Computer Simulations in Brain Science, Springer Verlag
International Journal of Neuroscience, Springer Verlag
Neural Network Computation, Springer Verlag
Neural Computing and Applications, Springer Verlag
Complex Systems, Complex Systems Publications
Biological Cybernetics (Kybernetik), Springer Verlag
The Behavioral and Brain Sciences, Cambridge University Press
Journal of Complex System, USA: World Scientific Publishing Co.
INTELLIGENCE - The Future of Computing, Intelligence.

4. Lineárne modely neurónových sietí

V tejto kapitole sa budeme zaoberať najjednoduchšími modelmi neurónových sietí — modelmi ktorých základnou stavebnou jednotkou je formálny neurón s lineárnou aktivačnou funkciou *f*. Označme výstup formálneho neurónu ako *y*. Nech aktivity prichádzajúce cez *n* vstupných kanálov tvoria vektor $\boldsymbol{x}=(x_1,x_2,...,x_n)^{\mathrm{T}}$, kde horný index T označuje transponovaný vektor. Nech sú vstupné kanály váhované pomocou vektora váh $\boldsymbol{w}=(w_1,w_2,...,w_n)^{\mathrm{T}}$. Potom bude výstupná aktivita lineárneho neurónu rovná

$$y = f(net) = f\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + \vartheta\right), \tag{4.1}$$

kde ϑ je prah aktivácie (excitácie). Najčastejšie je aktivačná funkcia identické zobrazenie a formálny neurón realizuje skalárny súčin vstupného vektora $\boldsymbol{x}=(x_1,x_2,...,x_n)^{\mathrm{T}}$ s váhovým vektorom $\boldsymbol{w}=(w_1,w_2,...,w_n)^{\mathrm{T}}$:

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}. \tag{4.2}$$

Samozrejme, takýto model nezohľadňuje saturačné vlastnosti reálnych neurónov (aktivita neurónu sa nemôže zvyšovať priamo úmerne s nárastom postsynaptického potenciálu $\boldsymbol{w}^{T}\boldsymbol{x}$, ale pre veľké hodnoty sa saturuje). Na druhej strane, lineárna povaha modelu nám umožní nájsť analytické riešenia problémov, ktoré v nelineárnych modeloch zohľadňujúcich saturácie neurónov nie je možné získať. Interpretáciou výsledkov aparátom lineárnej algebry možno presne pochopiť, čo sa skrýva za pojmami ako je zovšeobecnenie na základe trénovacej množiny, mechanizmus asociácie vstupov so "zapamätanými" prototypovými vzormi, atď.

Označme váhu *j*-teho vstupného kanála do *i*-teho neurónu w_{ij}. Ak neurónová sieť pozostáva z *m* neurónov reagujúcich na *n* vstupných kanálov (obr. 4.1), potom váhy všetkých prepojení možno prehľadne usporiadať do váhovej matice

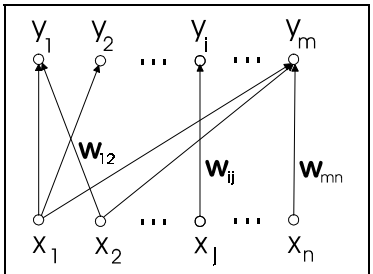
$$\boldsymbol{W} = \begin{bmatrix} w_{11}w_{12}\ldots w_{1n} \\ w_{21}w_{22}\ldots w_{2n} \\ \ldots\ldots\ldots\ldots \\ w_{m1}w_{m2}\ldots w_{mn} \end{bmatrix}. \tag{4.3}$$

Po priložení vektora $\boldsymbol{x}$ na vstup siete dostávame ako výstup vektor $\boldsymbol{y}=(y_1,y_2,...,y_m)^{\mathrm{T}}$, kde $y_1,y_2,...,y_m$ sú aktivácie výstupných neurónov, pričom $\boldsymbol{y}=\boldsymbol{W}\boldsymbol{x}$. Jednovrstvová lineárna neurónová sieť teda realizuje lineárne zobrazenie φ: $R^n \rightarrow R^m$. Ak by sme pridali ďalšiu vrstvu, ktorej vstupy by boli aktivácie $\boldsymbol{y}$, výstup siete $\boldsymbol{z}$ by bol lineárny obraz vektora $\boldsymbol{y}$, $\boldsymbol{z}=\boldsymbol{V}\boldsymbol{y}$. Zrejme $\boldsymbol{z}=\boldsymbol{A}\boldsymbol{x}$, kde matica $\boldsymbol{A}=\boldsymbol{V}\boldsymbol{W}$. Keďže kompozícia lineárnych zobrazení je opäť lineárne zobrazenie, pridávanie ďalších vrstiev v lineárnych sieťach neprináša žiadny nový kvalitatívny efekt, a preto sa v ďalšom budeme zaoberať iba jednovrstvovými sieťami.

Majme trénovaciu množinu $T=\{(\boldsymbol{x}^{(1)},\boldsymbol{y}^{(1)}),(\boldsymbol{x}^{(2)},\boldsymbol{y}^{(2)}),...,(\boldsymbol{x}^{(N)},\boldsymbol{y}^{(N)})\}$, kde $\boldsymbol{x}^{(i)}\in R^n$ a $\boldsymbol{y}^{(i)}\in R^m$, $\forall i\in\{1,2,...N\}$. Sieť bude vykonávať činnosť predpísanú trénovacou množinou *T,* ak nájdeme takú maticu váh $\boldsymbol{W}$, že

$$\boldsymbol{y}^{(i)} = \boldsymbol{W}\boldsymbol{x}^{(i)},\ \forall i \in \{1,2,\ldots,N\}. \tag{4.4}$$

Nech $\boldsymbol{X}$ je matica, ktorej stĺpce tvoria trénovacie vstupy $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$,..., $\boldsymbol{x}^{(N)}$, a $\boldsymbol{Y}$ matica,



Obrázok 4.1. Lineárna neurónová sieť s *n* vstupmi a *m* lineárnymi neurónmi. Váha *j*-teho vstupného kanála do *i*-teho neurónu w_{ij} .

ktorej stĺpce sú zodpovedajúce "požadované" výstupy $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$,..., $\boldsymbol{y}^{(N)}$. Potom podmienka (4.4) sa dá prepísať v maticovom tvare

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X}. \tag{4.5}$$

Pokiaľ je matica $\boldsymbol{X}$ regulárna[1], požadované váhy prepojení dostávame priamo z (4.5)

$$\boldsymbol{W} = \boldsymbol{Y}\boldsymbol{X}^{-1}. \tag{4.6}$$

Vo všeobecnosti však nemožno predpokladať ani len to, že matica $\boldsymbol{X}$ je štvorcová (počet vzoriek v trénovacej množine je rovný dimenzii vstupných vzorov), nieto ešte lineárnu nezávislosť vstupov $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$,..., $\boldsymbol{x}^{(N)}$. V takom prípade je (4.6) modifikované na [1, 2]

$$\boldsymbol{W} = \boldsymbol{Y}\boldsymbol{X}^{+} \tag{4.7}$$

kde $\boldsymbol{X}^{+}=\boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{X}^{T})^{-1}$, ak počet riadkov matice $\boldsymbol{X}$ je menší ako počet jej stĺpcov (počet trénovacích vzoriek *N* je väčší ako dimenzia vstupov *n*) a hodnosť matice $\boldsymbol{X}$ je rovná *n*. V opačnom prípade $\boldsymbol{X}^{+}=(\boldsymbol{X}^{T}\boldsymbol{X})^{-1}\boldsymbol{X}^{T}$, ak pravda, hodnosť matice $\boldsymbol{X}$ je *N*. Predpokladáme, že trénovacie príklady korešpondujúce s lineárne závislými vstupmi sme z trénovacej množiny vylúčili. Matica $\boldsymbol{X}^{+}$ sa nazýva *pseudoinverzná matica* matice $\boldsymbol{X}$ a platí

$$\boldsymbol{X}\boldsymbol{X}^{+}\boldsymbol{X}=\boldsymbol{X}. \tag{4.8}$$

V ďalšom sa budeme zaoberať prípadom, keď počet trénovacích vzoriek *N* je menší ako dimenzia vstupov *n*. Predstavme si, že máme vyriešiť úlohu autoasociácie, teda zapamätať

[1] Štvorcová matica $\boldsymbol{X}$ dimenzie *n* sa nazýva regulárna, ak všetky jej riadky (resp. stĺpce) sú lineárne nezávislé. Jej determinant je rôzny od nuly.

si určité "prototypové" vzory $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$,..., $\boldsymbol{x}^{(N)}$, ktoré tvoria stĺpce matice $\boldsymbol{X}$. Trénovacia množina má tvar

$$T = \left\{\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}\right)\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(2)}\right),\ldots,\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}\right)\right\} \tag{4.9}$$

a matica požadovaných výstupov $\boldsymbol{Y}=\boldsymbol{X}$. Z (4.7) dostávame pre váhovú maticu

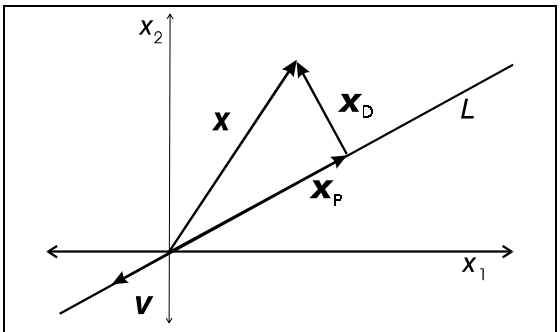
$$\boldsymbol{W} = \boldsymbol{X}\boldsymbol{X}^{+} \tag{4.10}$$

Samozrejme, jedným z možných riešení nášho problému by mohlo byť $\boldsymbol{W}=\boldsymbol{I}$, kde $\boldsymbol{I}$ je jednotková matica dimenzie n. V takomto prípade by sieť len slepo kopírovala na výstup to, čo vidí na vstupe. Takéto riešenie by sme dostali pomocou (4.9), ak by sme mali $N=n$ lineárne nezávislých vstupných vzorov. Ak je počet vstupných vzorov menší ako ich dimenzia, lineárny obal vzorov (množina vektorov, ktoré možno dostať lineárnou kombináciou vzorov) je lineárny podpriestor L lineárneho priestoru R^n. Označme ortogonálny doplnok priestoru L v R^n ako $L^{\perp}$. $L^{\perp}$ je lineárny podpriestor priestoru R^n obsahujúci všetky vektory kolmé na priestor L. Zopakujme si, že vektor je kolmý na priestor L, ak je kolmý na všetky vektory obsiahnuté v L. Zrejme platí, že ak je vektor kolmý na každý vektor z bázy priestoru L, potom je kolmý na priestor L. Každý vektor $\boldsymbol{x}\in R^n$ sa dá jednoznačne rozložiť $\boldsymbol{x}=\boldsymbol{x}_P+\boldsymbol{x}_D$ na zložky $\boldsymbol{x}_P\in L$ a $\boldsymbol{x}_D\in L^{\perp}$ (obr. 4.2).
Tento fakt sa ľahko dokáže sporom. Nech existujú dva rôzne rozklady

$$\boldsymbol{x}=\boldsymbol{x}_{P1}+\boldsymbol{x}_{D1} \text{ a } \boldsymbol{x}=\boldsymbol{x}_{P2}+\boldsymbol{x}_{D2},$$

kde $\boldsymbol{x}_{P1},\boldsymbol{x}_{P2}\in L$ a $\boldsymbol{x}_{D1},\boldsymbol{x}_{D2}\in L^{\perp}$. Zrejme $\boldsymbol{x}_{P1}-\boldsymbol{x}_{P2}=\boldsymbol{x}_{D2}-\boldsymbol{x}_{D1}$. Ľavá strana poslednej rovnosti je lineárna kombinácia vektorov z priestoru L, a teda je to vektor z priestoru L, kým pravá strana z analogických dôvodov je vektor z priestoru $L^{\perp}$. Toto je možné, len ak pravá aj ľavá strana sú nulové vektory a teda $\boldsymbol{x}_{P1}=\boldsymbol{x}_{P2}$ a $\boldsymbol{x}_{D1}=\boldsymbol{x}_{D2}$, čo je v spore s naším predpokladom dvoch rôznych rozkladov.

Dopredu prezradíme, že stratégia, ktorou sa lineárna sieť s váhami $\boldsymbol{W}=\boldsymbol{X}\boldsymbol{X}^{+}$ zhostí autoasociačnej úlohy je nasledujúca [2, 3]: Vstupné "prototypové" vzorky definujú podpriestor L v celom vstupnom priestore R^n. L je lineárny obal prototypových vstupných vektorov. Sieť považuje každú odchýlku od priestoru L za "pridaný šum", ktorý vyfiltruje. Pri príchode neznámej vstupnej vzorky $\boldsymbol{x}$ sieť "predpokladá", že ide o málo deformovanú podobu niektorého zo vstupných prototypov (uložených ako stĺpce matice $\boldsymbol{X}$) a urobí ortogonálny priemet $\boldsymbol{x}_P$ vektora $\boldsymbol{x}$ do podpriestoru L, pričom $\boldsymbol{x}_P$ prehlási za "vyčistenú" verziu vstupu $\boldsymbol{x}$. Vektor $\boldsymbol{x}_D=(\boldsymbol{x}-\boldsymbol{x}_P)\in L^{\perp}$ predstavuje časť, ktorú sieť považuje za deformáciu vstupu vzhľadom na uložené prototypy. Je možný aj iný pohľad, interpretujúci vektor $\boldsymbol{x}_D$ ako "novátorský aspekt" vstupu $\boldsymbol{x}$, vo svetle vzorov predstavených v trénovacej množine. Aby sme dokázali, že hore uvedená stratégia zodpovedá skutočnosti, musíme ukázať, že $\boldsymbol{W}\boldsymbol{x}=\boldsymbol{X}\boldsymbol{X}^{+}\boldsymbol{x}=\boldsymbol{x}_P$. Inými slovami, je treba dokázať, že matica $\boldsymbol{X}\boldsymbol{X}^{+}$ je operátor vykonávajúci ortogonálnu projekciu na lineárny obal stĺpcov matice $\boldsymbol{X}$. Nájdime rozklad $\boldsymbol{x}=\boldsymbol{x}_1+\boldsymbol{x}_2$ vektora $\boldsymbol{x}$, že $\boldsymbol{x}_1\in L$ a $\boldsymbol{x}_2\in L^{\perp}$. Vieme, že $\in L^{\perp}$ práve vtedy, keď $\boldsymbol{x}_2 \perp \mathrm{x}^{(i)}$, i=1, 2,..., N, čo v maticovej formulácii znamená



Obrázok 4.2. L je lineárny obal vektora $\boldsymbol{v}$ (lineárny priestor, ktorého báza je vektor $\boldsymbol{v}$), vektor $\boldsymbol{x}$ sa dá jednoznačne rozložiť na dve zložky $\boldsymbol{x}_P \in L$ a $\boldsymbol{x}_D \in L^{\perp}$.

$$\boldsymbol{X}^T \boldsymbol{x}_2 = \boldsymbol{o}\,, \tag{4.11}$$

kde **o** je nulový vektor dimenzie n.

V ďalšom budeme využívať fakt, že riešením sústavy lineárnych rovníc $\boldsymbol{Au} = \boldsymbol{b}$, kde $\boldsymbol{A}$ je matica sústavy s väčším počtom stĺpcov ako riadkov, je

$$\boldsymbol{u}=\boldsymbol{A}^{+}\boldsymbol{b}+(\boldsymbol{I}-\boldsymbol{A}^{+}\boldsymbol{A})\boldsymbol{v}\,. \tag{4.12}$$

$\boldsymbol{v}$ je ľubovoľný vektor rovnakej dimenzie ako vektor $\boldsymbol{x}$. To sa dá ľahko vidieť, pretože

$$\boldsymbol{Au}=\boldsymbol{AA}^{+}\boldsymbol{b}+(\boldsymbol{A}-\boldsymbol{AA}^{+}\boldsymbol{A})\boldsymbol{v}=\boldsymbol{AA}^{+}\boldsymbol{b}=\boldsymbol{b}\,, \tag{4.13}$$

keďže $\boldsymbol{AA}^{+}=\boldsymbol{AA}^{T}(\boldsymbol{AA}^{T})^{-1}=\boldsymbol{I}$. Matica $\boldsymbol{X}^T$ je požadovaného typu a teda z (4.11) a (4.12) dostávame

$$\boldsymbol{x}_2=(\boldsymbol{X}^T)^{+}\boldsymbol{o} + (\boldsymbol{I}-(\boldsymbol{X}^T)^{+}\boldsymbol{X}^T)\boldsymbol{v} \tag{4.14}$$

Ľahko sa presvedčíme, že

$$(\boldsymbol{X}^T)^{+}\boldsymbol{X}^T = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T = \boldsymbol{XX}^{+} \tag{4.15}$$

takže

$$\boldsymbol{x}_2=(\boldsymbol{I}-\boldsymbol{XX}^{+})\boldsymbol{v}\,. \tag{4.16}$$

Zostáva nájsť "ten pravý" vektor $\boldsymbol{v}$ definujúci ortogonálny rozklad vektora $\boldsymbol{x}$. Presvedčíme sa, že taký rozklad dostávame práve vtedy, keď $\boldsymbol{v}=\boldsymbol{x}$. Skutočne,

$$\boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 = (\boldsymbol{x} - \boldsymbol{XX}^{+}\boldsymbol{x})^{\mathrm{T}}\,\boldsymbol{XX}^{+}\boldsymbol{x} = (\boldsymbol{x}^{\mathrm{T}} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{XX}^{+})\,\boldsymbol{XX}^{+}\boldsymbol{x} = 0\,, \tag{4.17}$$

lebo $\boldsymbol{XX}^{+}$ je symetrická[2] matica a $\boldsymbol{XX}^{+}\,\boldsymbol{XX}^{+} = \boldsymbol{XX}^{+}$.

---

[2] Matica $\boldsymbol{X}$ sa nazýva symetrická, ak $\boldsymbol{X}=\boldsymbol{X}^{\mathrm{T}}$, kde $\boldsymbol{X}^{\mathrm{T}}$ je transponovaná matica k matici $\boldsymbol{X}$, ktorá vznikne z $\boldsymbol{X}$ zámenou riadkov za stĺpce.

Ukázali sme, že $\boldsymbol{x}_1=\boldsymbol{X}\boldsymbol{X}^+\boldsymbol{x}=\boldsymbol{x}_P$, čo znamená, že $\boldsymbol{X}\boldsymbol{X}^+$ je matica vykonávajúca ortogonálnu projekciu na lineárny obal stĺpcov matice $\boldsymbol{X}$ (obr. 4.3).
Podotýkame, že ak by sme namiesto $\boldsymbol{X}\boldsymbol{X}^+$ použili váhovú maticu $\boldsymbol{W}=\boldsymbol{I}-\boldsymbol{X}\boldsymbol{X}^+$, dostali by sme ako odozvu na vstup $\boldsymbol{x}$ vektor $\boldsymbol{x}_2=\boldsymbol{W}\boldsymbol{x}=(\boldsymbol{I}-\boldsymbol{X}\boldsymbol{X}^+)\boldsymbol{x}=\boldsymbol{x}_D$, ktorý sa dá voľne interpretovať ako reprezentant nového a originálneho vo vstupe $\boldsymbol{x}$, vzhľadom na trénovacie prototypy uložené ako stĺpce matice $\boldsymbol{X}$. Takáto neurónová sieť dostala meno detektor novosti (angl. *novelty detector*) [2, 3].

Zatiaľ sme si ukázali, ako pre danú trénovaciu množinu T vytvoriť váhovú maticu, ktorá realizuje zobrazenie určené asociačnými pármi z T. Okrem iného, je potrebné vypočítať pseudoinverznú maticu k matici $\boldsymbol{X}$, čo pri úlohách reálnejšieho rozsahu môže byť výpočtovo veľmi náročné. Navyše, ak by sme po natrénovaní siete na určitej trénovacej množine T chceli pribrať ešte jeden nový trénovací príklad, museli by sme odznova prerátať celú váhovú maticu. Ukážeme si, že v prípade autoasociačnej úlohy sa dá robiť výpočet váhovej matice iteračne a príchod nového trénovacieho príkladu predstavuje len pomerne jednoduchý výpočtový úkon príslušne modifikujúci maticu váh $\boldsymbol{W}$.

Predstavme si, že stojíme pred nasledujúcou úlohou. K danej báze $\boldsymbol{u}^{(1)},\boldsymbol{u}^{(2)},...,\boldsymbol{u}^{(k)}$ vektorového priestoru L máme vytvoriť ortogonálnu bázu $\boldsymbol{v}^{(1)},\boldsymbol{v}^{(2)},...,\boldsymbol{v}^{(k)}$ definujúcu ten istý priestor L. Algoritmus nazvaný Gramov a Schmidtov (G-S) ortogonalizačný proces realizuje presne takúto úlohu a jeho popis je nasledovný [2]:

1. Položme $\boldsymbol{v}^{(1)}=\boldsymbol{u}^{(1)}$
2. V priestore určenom bázou $\boldsymbol{v}^{(1)}$, $\boldsymbol{u}^{(2)}$ nájdime vektor $\boldsymbol{v}^{(2)}$, tak aby $\boldsymbol{v}^{(1)}\perp\boldsymbol{v}^{(2)}$. To sa dá dosiahnuť napríklad takto

$$\boldsymbol{v}^{(2)}=\boldsymbol{u}^{(2)}-\frac{\boldsymbol{v}^{(1)T}\boldsymbol{u}^{(2)}}{\left\|\boldsymbol{v}^{(1)}\right\|^2}\boldsymbol{v}^{(1)}. \tag{4.18}$$

Čitateľ sa môže ľahko presvedčiť, že vektor $\boldsymbol{v}^{(2)}$ je lineárnou kombináciou vektorov $\boldsymbol{v}^{(1)},\boldsymbol{u}^{(2)}$ a skalárny súčin vektorov $\boldsymbol{v}^{(1)},\boldsymbol{v}^{(2)}$ je rovný nule.
3. Bod 2. sa dá rekurentne rozšíriť nasledovne: Nech $\tilde{k}1$ vektorov $\boldsymbol{v}^{(1)},\boldsymbol{v}^{(2)},...,\boldsymbol{v}^{(\tilde{k}1)}$ novej ortogonálnej bázy už bolo určených. Potom k-ty vektor $\boldsymbol{v}^{(k)}$, od ktorého požadujeme aby ležal v priestore určenom bázou $\boldsymbol{v}^{(1)},\boldsymbol{v}^{(2)},...,\boldsymbol{v}^{(\tilde{k}1)},\boldsymbol{u}^{(k)}$ a bol ortogonálny na všetky vektory $\boldsymbol{v}^{(1)},\boldsymbol{v}^{(2)},...,\boldsymbol{v}^{(\tilde{k}1)}$, je určený takto:

$$\boldsymbol{v}^{(k)}=\boldsymbol{u}^{(k)}-\sum_{i=1}^{k-1}\frac{\boldsymbol{v}^{(i)T}\boldsymbol{u}^{(i)}}{\left\|\boldsymbol{v}^{(i)}\right\|^2}\boldsymbol{v}^{(i)}. \tag{4.19}$$

Opäť sa dá ľahko overiť, že $\boldsymbol{v}^{(k)}$ vyhovuje našim požiadavkám.

Predpokladajme, že trénovacia množina má N vzoriek $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},...,\boldsymbol{x}^{(N)}$ a $\boldsymbol{x}$ je vektor neležiaci v ich lineárnom obale. Nech $\tilde{\boldsymbol{x}}^{(1)},\tilde{\boldsymbol{x}}^{(2)},...,\tilde{\boldsymbol{x}}^{(N)}$ sú vektory určené zo vstupov $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},...,\boldsymbol{x}^{(N)}$ pomocou G-S ortogonalizačného procesu. Zrejme vektor

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)T}\boldsymbol{x}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\tilde{\boldsymbol{x}}^{(i)} \tag{4.20}$$

je kolmý na lineárny obal vstupov $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}$. Navyše ukážeme, že vektory $\tilde{\boldsymbol{x}}$ a $\boldsymbol{x} - \tilde{\boldsymbol{x}}$ sú na seba kolmé, čo znamená, že $\boldsymbol{x} - \tilde{\boldsymbol{x}} = \boldsymbol{x}_P = \boldsymbol{X}\boldsymbol{X}^{+}\boldsymbol{x}$ a $\tilde{\boldsymbol{x}} = \boldsymbol{x}_D = \left(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^{+}\right)\boldsymbol{x}$, kde $\boldsymbol{X}$ je matica, ktorej stĺpce tvoria vektory $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}$. Vypočítajme skalárny súčin $\tilde{\boldsymbol{x}}^{T}(\boldsymbol{x} - \tilde{\boldsymbol{x}})$. Keďže súčin matíc je asociatívny, dostaneme

$$\left(\tilde{\boldsymbol{x}}^{(j)T}\boldsymbol{x}\right)\tilde{\boldsymbol{x}}^{(j)} = \tilde{\boldsymbol{x}}^{(j)}\left(\tilde{\boldsymbol{x}}^{(j)T}\boldsymbol{x}\right) = \left(\tilde{\boldsymbol{x}}^{(j)}\tilde{\boldsymbol{x}}^{(j)T}\right)\boldsymbol{x}, \tag{4.21}$$

a preto

$$\tilde{\boldsymbol{x}}^{T}\left(\boldsymbol{x} - \tilde{\boldsymbol{x}}\right) = \left(\left[\boldsymbol{I} - \sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\right]\boldsymbol{x}\right)^{T}\left(\sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\right)\boldsymbol{x}. \tag{4.22}$$

Keď označíme maticu $\sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}$ symbolom $\boldsymbol{A}$, potom z (4.22) $\boldsymbol{x}^{T}\left(\boldsymbol{I} - \boldsymbol{A}^{T}\right)\boldsymbol{A}\boldsymbol{x} =$

$= \boldsymbol{x}^{T}\left(\boldsymbol{A} - \boldsymbol{A}^{T}\boldsymbol{A}\right)\boldsymbol{x} = \boldsymbol{0}$, lebo $\boldsymbol{A}^{T}\boldsymbol{A} = \boldsymbol{A}$. To sa dá ľahko dokázať. Stačí si uvedomiť, že matica $\boldsymbol{A}$ je symetrická (a teda $\boldsymbol{A}^{T}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}$) a $\frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\frac{\tilde{\boldsymbol{x}}^{(j)}\tilde{\boldsymbol{x}}^{(j)T}}{\left\|\tilde{\boldsymbol{x}}^{(j)}\right\|^2} = \boldsymbol{0}$, pre $i \neq j$, zatiaľ čo

$\frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2} = \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^4}\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2 = \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}$, pre $i=j$. Ukázali sme, že priemet vektora $\boldsymbol{x} = \boldsymbol{x}^{(N+1)}$ do lineárneho obalu vektorov $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}$ možno určiť ako

$$\tilde{\boldsymbol{x}}^{(N+1)} = \sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)T}\boldsymbol{x}^{(N+1)}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\tilde{\boldsymbol{x}}^{(i)} = \sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}\boldsymbol{x}^{(N+1)} = \boldsymbol{A}_N\boldsymbol{x}^{(N+1)}, \tag{4.23}$$

kde $\boldsymbol{A}_N = \sum_{i=1}^{N} \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}$. Ak by sme teraz uvažovali ďalší vstupný vzor $\boldsymbol{x}^{(N+2)}$, neležiaci v lineárnom obale vektorov $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N+1)}$, jeho priemet do tohoto lineárneho obalu dostaneme pomocou $\tilde{\boldsymbol{x}}^{(N+2)} = \boldsymbol{A}_{N+1}\boldsymbol{x}^{(N+2)}$, kde $\boldsymbol{A}_{N+1} = \sum_{i=1}^{N+1} \frac{\tilde{\boldsymbol{x}}^{(i)}\tilde{\boldsymbol{x}}^{(i)T}}{\left\|\tilde{\boldsymbol{x}}^{(i)}\right\|^2}$ a $\tilde{\boldsymbol{x}}^{(N+1)}$ dostaneme z predošlého kroku ortogonalizačného procesu $\tilde{\boldsymbol{x}}^{(N+1)} = \boldsymbol{x}^{(N+1)} - \boldsymbol{A}_N\boldsymbol{x}^{(N+1)}$.

Ukázali sme si, že maticu váh prepojení $\boldsymbol{W}=\boldsymbol{X}\boldsymbol{X}^{+}$ lineárneho autoasociátora môžeme rátať iteračne, vždy pri príchode novej prototypovej vstupnej vzorky, na základe rekurentnej schémy:

1. Inicializuj $\boldsymbol{W}_0$ ako nulovú maticu, $\boldsymbol{W}_0 = \boldsymbol{0}$.
2. Pre $N>0$,

$$\boldsymbol{W}_{N+1} = \boldsymbol{W}_N + \frac{\tilde{\boldsymbol{x}}^{(N+1)}\tilde{\boldsymbol{x}}^{(N+1)^T}}{\left\|\tilde{\boldsymbol{x}}^{(N+1)}\right\|^2}, \tag{4.24}$$

kde $\tilde{\boldsymbol{x}}^{(N+1)} = \boldsymbol{x}^{(N+1)} - \boldsymbol{W}_N\boldsymbol{x}^{(N+1)}$. Model, ktorým sme sa doteraz zaoberali, dostal názov GI (angl. *General Inverse*) [2, 3] podľa pseudoinverznej matice $\boldsymbol{X}^{+}$ použitej pri výpočte matice synaptických váh. V ďalšom si ukážeme iný model lineárnych sietí, ktorý sa líši od predošlého maticou $\boldsymbol{W}$ synaptických váh.

4.1 Realizácia pamäti pomocou korelačnej matice

V roku 1949 sformuloval kanadský psychológ Hebb hypotézu o plasticite zmeny synaptických váh [4]. Priepustnosť synaptického spojenia vstupného kanála s neurónovou bunkou je priamo úmerná pred- a postsynaptickej aktivite neurónu. Inými slovami, ak je veľká korelácia medzi aktivitou na kanáli z *j*-teho neurónu do neurónu *i* a výstupnou aktivitou neurónu *i*, potom nárast synaptickej váhy kanála je priamo úmerný tejto korelácii. Formálne, ak je trénovacia množina

$$T = \left\{\left(\boldsymbol{x}^{(1)},\boldsymbol{y}^{(1)}\right),\left(\boldsymbol{x}^{(2)},\boldsymbol{y}^{(2)}\right),\ldots,\left(\boldsymbol{x}^{(N)},\boldsymbol{y}^{(N)}\right)\right\}, \tag{4.25}$$

potom váha w_{ij} bude priamo úmerná korelácii

$$w_{ij} \approx \sum_{k=1}^{N} x_j^{(k)} y_i^{(k)} \quad , \tag{4.26}$$

čomu v maticovom zápise zodpovedá

$$\boldsymbol{W} \approx \sum_{k=1}^{N} \boldsymbol{y}^{(k)}\boldsymbol{x}^{(k)^T} = \boldsymbol{Y}\boldsymbol{X}^T. \tag{4.27}$$

V našich ďalších úvahách budeme uvažovať $\boldsymbol{W}=\boldsymbol{Y}\boldsymbol{X}^{\mathrm{T}}$. Okamžite vidíme, že ak sú vstupné trénovacie vektory $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(N)}$ ortonormálne, dostávame $\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X}^{-1} = \boldsymbol{X}^{+}$ a náš "biologicky motivovaný" model je totožný s modelom GI, ktorým sme sa zaoberali v predošlom odseku. Požiadavka ortonormality vstupov však v praxi ťažko obstojí. Preto je namieste otázka, aká je funkcia siete, ak vstupné vektory sú len lineárne nezávislé. Opäť sa pokúsime ilustrovať funkciu siete na autoasociačnej úlohe.

Máme si zapamätať N lineárne nezávislých vstupov $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(N)}$ uložených ako stĺpce matice $\boldsymbol{X}$. Zrejme $\boldsymbol{W}=\sum_{k=1}^{N}\boldsymbol{x}^{(k)}\boldsymbol{x}^{(k)^T}=\boldsymbol{X}\boldsymbol{X}^T$. Po priložení p-teho vektora $\boldsymbol{x}^{(p)}$, $p\in\{1,2,\ldots,N\}$ na vstup takejto siete dostávame ako odozvu vektor

$$\begin{aligned}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{x}^{(p)} &= \left(\sum_{k=1}^{N}\boldsymbol{x}^{(k)}\boldsymbol{x}^{(k)^T}\right)\boldsymbol{x}^{(p)}\\ &= \sum_{k=1}^{N}\boldsymbol{x}^{(k)}\boldsymbol{x}^{(k)^T}\boldsymbol{x}^{(p)} = \boldsymbol{x}^{(p)}\boldsymbol{x}^{(p)^T}\boldsymbol{x}^{(p)} + \sum_{k=1,k\neq p}^{N}\boldsymbol{x}^{(k)}\boldsymbol{x}^{(k)^T}\boldsymbol{x}^{(p)}\end{aligned} \tag{4.28}$$

a teda

$$\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{x}^{(p)} = \boldsymbol{x}^{(p)}\left\|\boldsymbol{x}^{(p)}\right\|^2 + C(p), \tag{4.29}$$

kde $C(p)=\sum_{k=1,k\neq p}^{N}\boldsymbol{x}^{(k)}\boldsymbol{x}^{(k)^T}\boldsymbol{x}^{(p)}$ je tzv. "presluch" (angl. *crosstalk*) od iných vstupných vzoriek. Ak je presluch nenulový, vstupy $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(N)}$ nie sú ortogonálne. Vo všeobecnosti teda na výstupe siete dostávame súčet vektora kolineárneho s daným uloženým vstupom a "presluchového" vektora.

V prípade modelu GI sme si vysvetlili činnosť autoasociátora aparátom lineárnej algebry. Videli sme, že takáto sieť zovšeobecní vstupy trénovacej množiny tak, že vytvorí ich lineárny obal. Odozva siete na nový, dosiaľ nepredložený vstup (ktorý môže predstavovať poškodenú, alebo zašumenú verziu niektorého z trénovacích vzorových vstupov) je ortogonálny priemet vstupu do lineárneho obalu trénovacích vstupov. Činnosť autoasociátora založeného na modeli uvedenom v tejto sekcii si vysvetlíme štatistickými úvahami.

Predstavme si, že trénovacie vstupy $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(N)}$ sú realizácie vektorovej náhodnej premennej $X=(X_1,X_2,\ldots,X_n)^T$, ktorej každá zložka má strednú hodnotu nula ($E(X_i)=0,\ i=1,2,\ldots,n$).

Prvok w_{ij} váhovej matice $\boldsymbol{W}=\boldsymbol{X}\boldsymbol{X}^T$ je rovný $w_{ij}=\sum_{k=1}^{N}x_i^{(k)}x_j^{(k)}$, čo je priamo úmerné neodchýlenému odhadu $\langle\mathrm{cov}(X_i,X_j)\rangle$ kovariancie $\mathrm{cov}(X_i,X_j)=E[(X_i-E(X_i))(X_j-E(X_j))]=E[X_iX_j]$ i-tej a j-tej zložky náhodného vektora X. Váha w_{ij} nesie informáciu o sile vzájomnej (lineárnej) previazanosti náhodných premenných X_i,X_j. Priložme na vstup siete vektor $\boldsymbol{x}=(x_1,x_2,\ldots,x_n)^T$. i-ta zložka výstupu je daná súčtom príspevkov od všetkých súradníc ováhovaným príslušnými koreláciami a je priamo úmerná $\sum_{j=1}^{n}\langle\mathrm{cov}(X_i,X_j)\rangle\,x_j$. Ak by bola i-ta zložka vstupu $\boldsymbol{x}$

vplyvom šumu silne zredukovaná a na základe trénovacích vstupov je možné sa domnievať, že napríklad existuje silná korelácia medzi *i*-tou a *r*-tou, ako aj medzi *i*-tou a *p*-tou zložkou vstupov (váhy w_{ir} a w_{ip} sú dominantné), potom ak zložky $\boldsymbol{x}_r$, $\boldsymbol{x}_p$ vstupu $\boldsymbol{x}$ sú silne aktivované, spôsobia aj silnú aktiváciu *i*-tej zložky výstupu. Autokorekcia *i*-tej zložky vstupu sa udiala na základe pozorovania, že súčasne s *r*-tou a *p*-tou zložkou je obyčajne aktivovaná aj zložka *i* a zložky *r* a *p* mali vysokú aktivitu. Podobná úvaha platí pre štatisticky slabo prepojené zložky vstupov. Tu zasa možno stiahnuť vysokú aktivitu nejakej zložky vstupného vektora, ak na trénovacích vstupoch nebola pozorovaná výrazná previazanosť tejto zložky na momentálne aktívne komponenty vstupov.

Model autoasociátora, ktorý sme popísali, sa nazýva pamäť korelačnej matice (angl. *Correlation Matrix Memory*, CMM) [2, 3]. Treba poznamenať, že tento model sa opiera o štatistiky budované nad dvojicami zložiek vstupných vektorov. To môže byť pre zachytenie podstatných korelácií málo, pretože výrazným spôsobom môžu byť prepojené napr. trojice, alebo štvorice vstupných zložiek a na zachytenie tejto skutočnosti by sme potrebovali odhadovať štatistiky vyšších rádov.

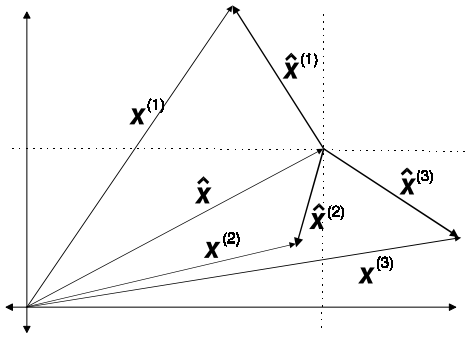
Predpoklad, že stredné hodnoty zložiek náhodného vektora *X* (ktorého realizácie sú prototypové vstupy) sú nulové nie je až taký nerealistický, ako by sa mohlo zdať na prvý pohľad. Vždy je totiž možné urobiť odhad strednej hodnoty vstupov

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{x}^{(k)} \quad , \tag{4.30}$$

a posunúť do nej počiatok súradnicovej sústavy

$$\hat{\boldsymbol{x}} = \boldsymbol{x} - \overline{\boldsymbol{x}} \, . \tag{4.31}$$

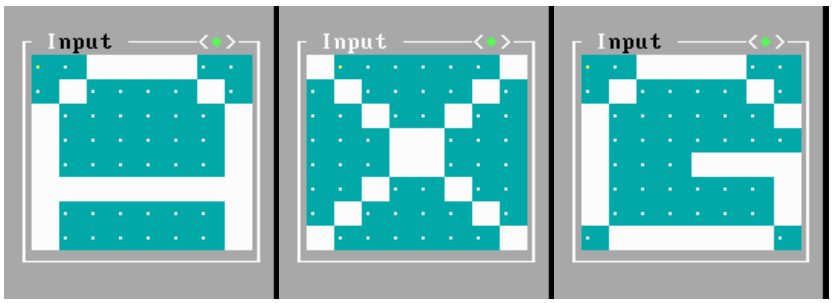
Maticu $\boldsymbol{X}$ potom skonštruujeme z transformovaných prototypových vstupov $\hat{\boldsymbol{x}}^{(i)}, \, i = 1,2,\ldots,N$. Odozvu siete na transformovaný vstup opäť transformujeme do pôvodnej súradnicovej sústavy spätnou transformáciou $\boldsymbol{x} = \hat{\boldsymbol{x}} + \overline{\boldsymbol{x}}$.
Transformácia (4.31) pomáha zmenšiť presluch medzi prototypovými vstupmi (ako naznačuje obr. 4.4), pretože ak sú vstupné vektory "primknuté k sebe", ich transformované verzie budú navzájom zvierať väčšie uhly.



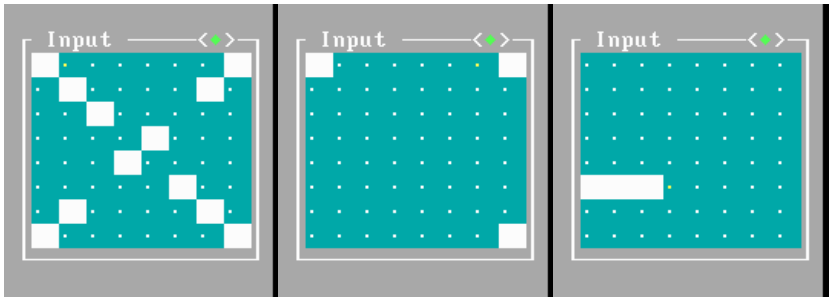
Obrázok 4.4. Prototypové vstupy $x^{(1)}, x^{(2)}, x^{(3)}$ a ich transformované obrazy $\hat{x}^{(1)}, \hat{x}^{(2)}, \hat{x}^{(3)}$ vyjadre-né v súradnicovej sústave s počiatkom v ťažisku $\hat{x}$ vektorov $x^{(1)}, x^{(2)}, x^{(3)}$.

4.2 Príklady lineárnej autoasociácie

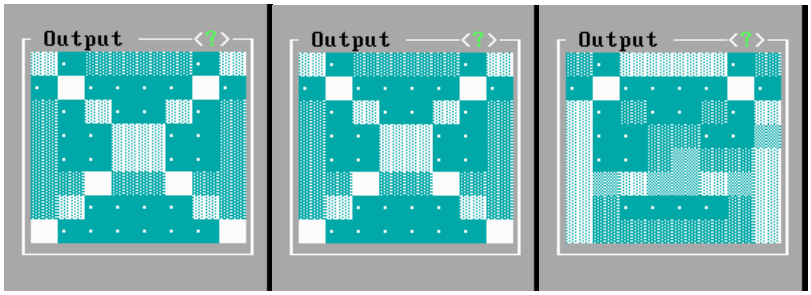
Na obr. 4.5a vidíme 3 prototypové vzory priložené na vstup lineárnej autoasociačnej siete so 64 neurónmi. Sieť bola testovaná na 3 porušených vstupoch zobrazených na obrázku 4.5b. Ide o dve "porušené" písmená X a jedno narušené písmeno A. Odozvy siete trénovanej stratégiou CMM sú prezentované na obrázku 4.5c. Obrázok 4.5d zobrazuje odozvu siete trénovanej stratégiou GI. Keďže prototypové vzory nie sú ortogonálne, odozvy siete GI sú kvalitnejšie než odozvy siete CMM.



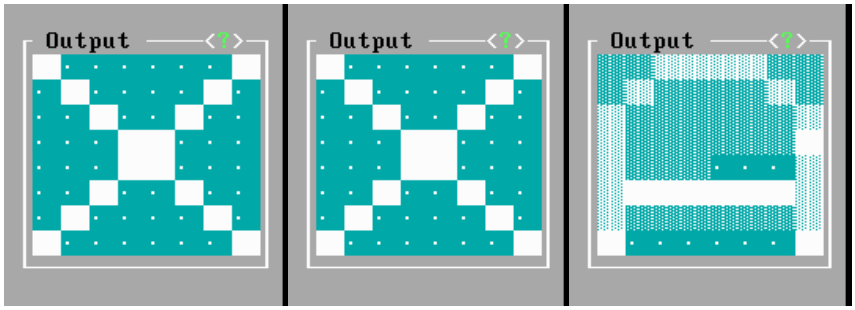
Obrázok 4.5a. Prototypové vstupy použité na trénovanie lineárneho autoasociátora.



Obrázok 4.5b. Testovacie vstupy pre lineárny asociátor trénovaný na vstupoch zobrazených na obrázku 4.5a.



Obrázok 4.5c. Odozvy siete typu CMM na testovacie vstupy (obrázok 4.5b).



Obrázok 4.5d. Odozvy siete typu GI na testovacie vstupy (obrázok 4.5b).

Literatúra

[1] F. Štulajter. *Odhady v náhodných procesoch*. Alfa, Bratislava, 1990.

[2] G.E. Hinton and J.A. Anderson, editors. *Parallel Models of Associative Memory*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1989.

[3] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1984.

[4] D.O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

5. Viacvrstvové neurónové siete

Jednoduchá neurónová sieť (tzv. lineárny model neurónovej siete, ako bol uvedený v kapitole 4), či už jednovrstvová alebo viacvrstvová, je schopná korektne riešiť len obmedzenú triedu problémov — tzv. lineárne separovateľné problémy (viď ďalej podkapitola 5.3.1). Lineárne neurónové siete môžu byť jednoducho zovšeobecnené tak, že aktivity výstupných neurónov sú určené pomocou nelineárnej prechodovej funkcie (ktorá v najjednoduchšom prípade odpovedá tzv. tvrdej nelinearite, napr. skokovej alebo znamienkovej funkcii). Avšak takéto zovšeobecnenie lineárnej neurónovej siete je tiež schopné klasifikovať len lineárne separovateľné problémy. Toto ohraničenie bolo považované za vážny nedostatok neurónových sietí [1]. Teoreticky sa uvažovalo o možnosti zavedenia ďalších (skrytých) vrstiev nelineárnych neurónov do sietí. Žiaľ, nebolo jasné ako adaptovať váhové koeficienty, ktoré sú priradené neurónom zo skrytej vrstvy. Až Rumelhart so spolupracovníkmi [2] navrhli jednoduchý gradientový algoritmus (nazývaný metóda spätného šírenia — angl. *back propagation*) adaptácie viacvrstvových neurónových sietí s dopredným šírením. Týmto sa viacvrstvové neurónové siete stali veľmi populárne a patria medzi univerzálne prístupy teórie neurónových sietí so širokou paletou aplikácií v rôznych oblastiach informatiky a prírodných vied. Naviac bolo dokázané, že neurónové siete tohto typu sú univerzálnym aproximátorom, t.j. sú schopné aproximovať s požadovanou presnosťou ľubovoľnú spojitú funkciu, čiže môžu byť chápané ako univerzálny prostriedok pre regresnú analýzu, kde tvar modelovej funkcie je určený architektúrou neurónovej siete. Pod architektúrou máme na mysli nielen "topológiu" prepojenia neurónov, ale tiež aj nastavenie váhových a prahových koeficientov na určité hodnoty.

5.1 Všeobecný klasifikačný problém

Zavedieme všeobecnú formuláciu klasifikačného problému pomocou pojmu zobrazenia — funkcie definovanej nad dvomi množinami A a B. Tento prístup bude užitočný pre interpretáciu neurónových sietí ako *klasifikátora* alebo *prediktora*. Nech $F(\mathbf{x})$ je funkcia definovaná nad množinou A, ktorá priradí každému elementu $\boldsymbol{x} \in A$ obraz — funkčnú hodnotu z množiny B, $\hat{\boldsymbol{x}} = F(\boldsymbol{x}) \in B$,

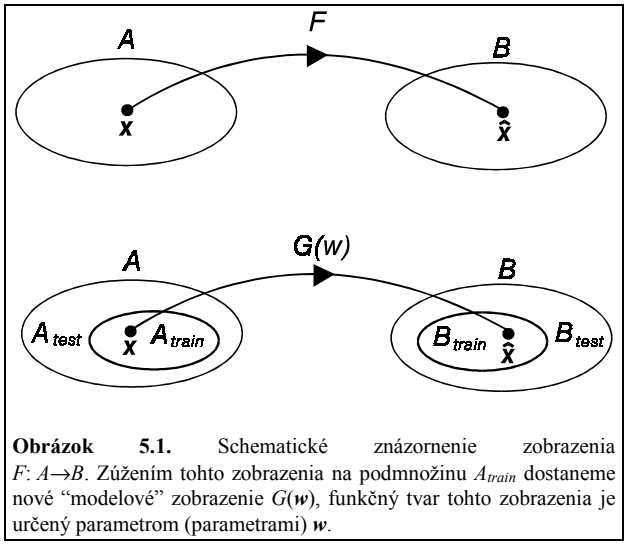
$$F: A \rightarrow B \tag{5.1}$$

Nech $G(\boldsymbol{x},\boldsymbol{w})$ je funkcia, ktorej argumenty sú z konečnej podmnožiny $A_{train} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r\} \subset A$ (nazývanej *tréningová množina*) a $\boldsymbol{w}$ je parameter (alebo parametre) zobrazenia G, potom $\hat{\boldsymbol{x}} = G(\boldsymbol{x},\boldsymbol{w}) \in B_{train} \subset B$ (pozri obr. 5.1)

$$G(\boldsymbol{w}): A_{train} \rightarrow B_{train} \tag{5.2}$$

Formálne môžeme povedať, že zobrazenie $G(\boldsymbol{w})$ je reštrikcia zobrazenia $F(\boldsymbol{x})$ nad množinou $A_{train} \subset A$. Komplement A_{train} vzhľadom k množine A je označený A_{test} (nazývaný *testovacia množina*), $A_{test}=A \setminus A_{train}$. Predpokladajme, že pre každé $\boldsymbol{x}_i \in A_{train}$ poznáme požadovaný obraz — funkčnú hodnotu $\hat{\boldsymbol{x}}_i$,

$$\boldsymbol{x}_1 \,/\, \hat{\boldsymbol{x}}_1, \boldsymbol{x}_2 \,/\, \hat{\boldsymbol{x}}_2, \ldots, \boldsymbol{x}_r \,/\, \hat{\boldsymbol{x}}_r \tag{5.3}$$



Obrázok 5.1. Schematické znázornenie zobrazenia F: $A \rightarrow B$. Zúžením tohto zobrazenia na podmnožinu A_{train} dostaneme nové "modelové" zobrazenie $G(\boldsymbol{w})$, funkčný tvar tohto zobrazenia je určený parametrom (parametrami) $\boldsymbol{w}$.

Požadované funkčné hodnoty $\hat{\boldsymbol{x}}_i$ sú interpretované ako obrazy funkcie F

$$\hat{\boldsymbol{x}}_i = F(\boldsymbol{x}_i) \qquad (i = 1,2,\ldots,r) \tag{5.4}$$

Cieľom našich úvah je nájsť taký parameter (alebo parametre) $\boldsymbol{w}$ funkcie $G(\boldsymbol{x},\boldsymbol{w})$, aby funkčné hodnoty argumentov z tréningovej množiny A_{train} boli čo najbližšie obrazom funkcie $F(\boldsymbol{x})$ (t.j. požadovaným hodnotám). Definujme *účelovú funkciu*

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{r}\left(G(\boldsymbol{x}_i,\boldsymbol{w}) - \hat{\boldsymbol{x}}_i\right)^2 \tag{5.5}$$

Táto funkcia vyjadruje sumu kvadrátov odchýlok funkcie $G(\boldsymbol{x},\boldsymbol{w})$ od požadovaných hodnôt $\hat{\boldsymbol{x}}$ braných z tréningovej množiny. Požiadavka, aby vypočítané hodnoty $G(\boldsymbol{x},\boldsymbol{w})$ boli "čo najbližšie" požadovaným hodnotám $\hat{\boldsymbol{x}}$ je realizovaná pomocou požiadavky minimálnosti účelovej funkcie $E(\boldsymbol{w})$ vzhľadom k parametru $\boldsymbol{w}$. Hovoríme, že funkcia $G(\boldsymbol{x},\boldsymbol{w})$ je *adaptovaná*, ak jej parameter $\boldsymbol{w}$ je vybraný tak, aby sa rovnal svojej optimálnej hodnote (t.j. v ktorom má účelová funkcia globálne minimum). Nech $\overline{\boldsymbol{w}}$ je optimálna hodnota parametru $\boldsymbol{w}$ určená nasledujúcim minimalizačným problémom

$$\overline{\boldsymbol{w}} = \arg\min_{\boldsymbol{w} \in W} E(\boldsymbol{w}) \tag{5.6}$$

kde W je množina (priestor) prípustných hodnôt parametra $\boldsymbol{w}$. Adaptovaná funkcia $G(\boldsymbol{x}, \overline{\boldsymbol{w}})$ simuluje pôvodnú funkciu $F(\boldsymbol{x})$ pre hodnoty argumentov z tréningovej množiny A_{train} na základe minimalizačného kritéria (5.6). Naviac, adaptovaná funkcia $G(\boldsymbol{x}, \overline{\boldsymbol{w}})$ sa používa aj pre predpoveď funkčných hodnôt odpovedajúcich argumentom z testovacej množiny A_{test}, t.j. predpokladá sa, že adaptovaná funkcia dobre aproximuje pôvodnú funkciu $F(\boldsymbol{x})$ tiež mimo tréningovej množiny. Naše úvahy môžu byť jednoducho chápané ako klasický regresný problém, v ktorom parametre modelovej funkcie G sú optimalizované (adaptované) tak, aby vypočítané funkčné hodnoty boli blízke požadovaným (experimentálnym) funkčným hodnotám.

Jeden zo základných problémov v každej oblasti prírodných vied je *hľadanie vzťahu — funkcie medzi štruktúrou jej objektov a ich vlastnosťami*. Ideálom je konštrukcia tejto funkcie v analytickom tvare, ktorá vzťahuje vlastnosti objektov k ich štruktúre. V mnohých prípadoch je tento cieľ buď vôbec nerealizovateľný alebo len s veľkými obtiažami. Tento meta-teoretický prístup ku konštrukcii analytických vzťahov pre koreláciu štruktúra verzus vlastnosť v mnohých prípadoch naráža na principiálne problémy tak teoretického, ako aj numerického charakteru. Preto sa pomerne často používa prístup "regresnej analýzy". Modelová funkcia G sa zostrojí na základe určitých úvah a jej voľne adjustovateľné parametre sa určia pomocou minimalizácie účelovej funkcie $E(\boldsymbol{w})$ (vzťah (5.6)). Takto adaptovaná modelová funkcia G sa potom berie ako analytický vzťah medzi štruktúrou prírodovedných objektov a ich vlastnosťami.

Nech O je množina *objektov*, $O=\{o_1,o_2,...\}$. Každý objekt $o \in O$ je popísaný deskriptorom $\boldsymbol{x}$, ktorý charakterizuje jeho *štruktúru*, a klasifikátorom $\hat{\boldsymbol{x}}$, ktorý popisuje jeho *vlastnosti*. Vzťah medzi deskriptorom a klasifikátorom je formálne vyjadrený pomocou hypotetickej funkcie $\hat{\boldsymbol{x}}=F(\boldsymbol{x})$. Ako už bolo spomenuté vyššie, konštrukcia tejto funkcie patrí medzi základné problémy každej vednej oblasti. Alternatívne riešenie tohto problému môže byť uskutočnené pomocou ad-hoc postulovania modelovej funkcie $G(\boldsymbol{x},\boldsymbol{w})$ v analytickom tvare. Parameter (alebo parametre) $\boldsymbol{w}$ je určený podmienkou, aby funkčné hodnoty pre deskriptory z tréningovej množiny boli blízke požadovaným hodnotám klasifikátora. Hlavným cieľom tohto postupu je, že adaptovaná modelová funkcia $G(\boldsymbol{x}, \overline{\boldsymbol{w}})$ je použitá pre klasifikáciu objektov mimo tréningovej množiny. To znamená, že adaptovaná modelová funkcia $G(\boldsymbol{x}, \overline{\boldsymbol{w}})$ je extrapolovaná mimo tréningovej množiny "v dobrej viere", že aj v tomto prípade bude dobre aproximovať funkciu $F(\boldsymbol{x})$, ktorá presne klasifikuje objekty z celej množiny O.

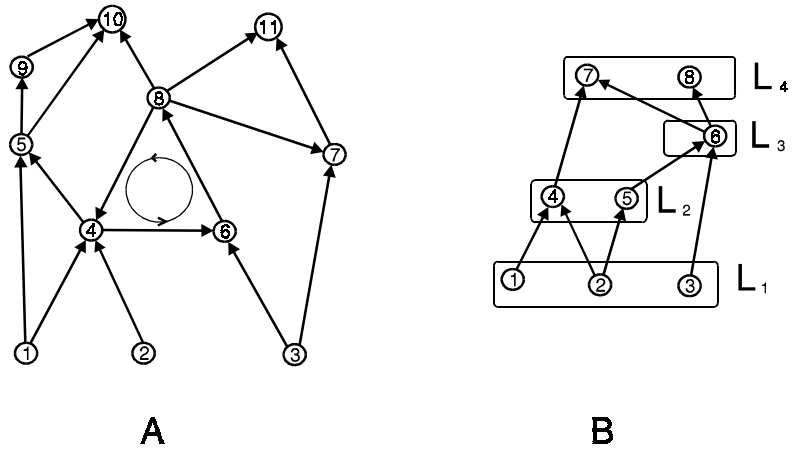
5.2 Definícia neurónovej siete

Paradigma neurónovej siete bude formulovaná pomocou grafovo-teoretického prístupu [3]. Pritom sa vychádza sa z analógie s ľudským mozgom (pozri kapitolu 1) a koncept neurónovej siete bude použitý na konštrukciu modelovej funkcie $G(\boldsymbol{x},\boldsymbol{w})$. Formálne je neurónová sieť určená ako orientovaný graf G=(V,E), pozri obr. 5.2. Výrazy $V=\{v_1,v_2,...,v_N\}$ a $E=\{e_1,e_2,...e_M\}$ označujú neprázdnu vrcholovú množinu resp. hranovú množinu grafu G obsahujúceho N vrcholov (neurónov) a M hrán (spojov). Každý spoj $e \in E$ sa interpretuje ako usporiadaná dvojica dvoch neurónov z množiny V, $e=(v,v')$. Hovoríme, že spoj e začína v neuróne v a končí v neuróne v'. Množina neurónov V je rozložená na disjunktné podmnožiny (pozri obr. 5. 2)

$$V = V_I \cup V_H \cup V_O \tag{5.7}$$

kde V_I obsahuje N_I vstupných neurónov, ktoré sú susedné len s vychádzajúcimi hranami, V_H obsahuje N_H skrytých (angl. *hidden*) neurónov, ktoré sú susedné súčasne s vychádzajúcimi ako aj s vchádzajúcimi hranami, a konečne V_O obsahuje N_O výstupných neurónov, ktoré sú susedné len s vchádzajúcimi hranami. V našich nasledujúcich úvahách budeme vždy predpokladať, že množiny V_I a V_O sú neprázdne, t.j. neurónová sieť obsahuje vždy aspoň jeden vstupný a jeden výstupný neurón.

Pre acyklické neurónové siete (ktoré neobsahujú orientované cykly (pozri graf A na obr. 5.2)) neuróny môžu byť usporiadané do vrstiev (pozri graf B na obr. 5.2)



Obrázok 5.2. Neurónová sieť je definovaná ako orientovaný súvislý graf. Diagram A obsahuje orientovaný graf s jedným cyklom a teda nemôže byť použitý pre definíciu neurónovej siete s dopredným šírením. Diagram B ilustruje možnosť rozkladu vrcholov (neurónov) acyklického orientovaného grafu na vrstvy $L_1,..,L_4$.

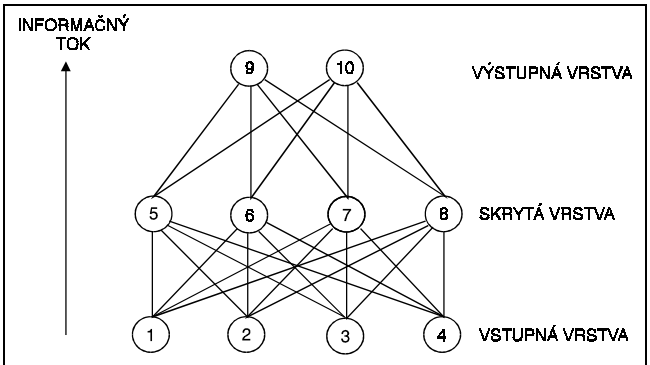
$$V = L_1 \cup L_2 \cup L_3 \cup ... \cup L_t \tag{5.8}$$

kde $L_1=V_I$ je *vstupná vrstva* (obsahuje len vstupné neuróny), L_2, L_3,..., L_{t-1} sú *skryté vrstvy* a L_t je *výstupná vrstva*. Vrstva L_i (pre $1 \le i \le t$) je určená nasledujúcim jednoduchým spôsobom

$$L_i = \{v \in V; d(v) = i + 1\} \tag{5.9}$$

kde vzdialenosť $d(v)$ sa rovná dĺžke maximálnej cesty, ktorá spája daný neurón so vstupným neurónom, potom musí platiť $d(v)=0$, pre $v \in V_I$. Neurónová sieť určená acyklickým grafom je obvykle volená tak, že neuróny z dvoch susedných vrstiev sú poprepájané všetkými možnými spojmi (pozri obr. 5.3). Žiaľ, takýto rozklad množiny neurónov na vrstvy je

možný len pre neurónové siete reprezentované acyklickými grafmi, pre cyklické grafy vzdialenosť $d(v)$ môže nadobúdať ľubovolnú kladnú celočíselnú hodnotu.



Obrázok 5.3. Znázornenie trojvrstvovej neurónovej siete s dopredným šírením.

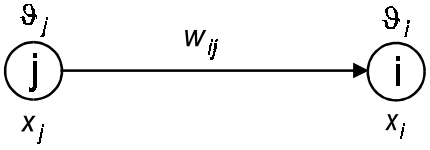
Iný, alternatívny spôsob [3], ako definovať orientovaný graf G je použitie zobrazenia Γ, ktoré priradí každému vrcholu $v \in V$ podmnožinu $\Gamma(v) \subset V$ obsahujúcu tie neuróny, ktoré sú koncovými na spojoch vychádzajúcich z vrcholu v. Neuróny z podmnožiny $\Gamma(v)$ sa nazývajú *nasledovníci* vrcholu v v grafe G. "Inverzné" zobrazenie Γ^{-1} priradí každému vrcholu $v \in V$ podmnožinu $\Gamma^{-1}(v) \subset V$ zloženú z "predchodcov" vrcholu v v grafe G.

Neuróny a spoje sú ohodnotené reálnymi číslami, pozri obr. 5.4. Každý neurón v_i je ohodnotený *prahovým koeficientom* ϑ_i a *aktivitou* x_i. Podobne, každý spoj (v_j,v_i) je ohodnotený váhovým koeficientom (alebo jednoducho, váhou) w_{ij}. Postulujeme, že aktivity skrytých a výstupných neurónov sú určené vzťahom

$$x_i = t(\xi_i) \tag{5.10a}$$

$$\xi_i = \sum_{j \in \Gamma_i^{-1}} w_{ij} x_j + \vartheta_i \tag{5.10b}$$

kde sumácia beží cez neuróny, ktoré sú predchodcami neurónu v_i.



Obrázok 5.4. Neuróny a spoje neurónovej siete sú ohodnotené reálnymi číslami: neuróny sú ohodnotené dvoma rôznymi parametrami, a to prahovými faktormi ϑ_i a aktivitami x_i. Spoje neurónovej siete sú ohodnotené váhovými koeficientmi w_{ij}.

Veličina ξ_i sa nazýva *potenciál* neurónu v_i (analógia tzv. postsynaptického potenciálu, pozri kapitolu 1). *Prechodová (aktivačná) funkcia* $t(\xi)$ z pravej strany (5.10a) je monotónne rastúca funkcia, ktorá vyhovuje nasledujúcim dvom asymptotickým podmienkam: $t(\xi)\to A$, pre $\xi\to-\infty$ a $t(\xi)\to B$, pre $\xi\to\infty$, kde $-\infty<A<B<\infty$. V teórii neurónových sietí sa často využíva nasledujúca "sigmoidálna" funkcia

$$t(\xi)=\frac{B+Ae^{-\xi}}{1+e^{-\xi}} \tag{5.11a}$$

s prvou deriváciou určenou

$$t'(\xi)=\frac{\left[-A+t(\xi)\right]\left[B-t(\xi)\right]}{A+B} \tag{5.11b}$$

Táto prechodová funkcia zobrazuje celú množinu reálnych čísel R na otvorený interval (A,B), formálne t: $R\to(A,B)$. Najčastejšie sa prechodová funkcia (5.11a) využíva pre hodnoty parametrov $A=0$, $B=1$ alebo $A=-1$, $B=1$ (pozri obr. 5.5). Prvý graf odpovedá klasickej sigmoidálnej prechodovej funkcii, zatiaľ čo druhý graf je analógiou hyperbolického tangentu.

Aktivity neurónov tvoria vektor $\boldsymbol{x}=(x_1,x_2,...x_N)$. Tento vektor možno formálne rozložiť na tri podvektory obsahujúce vstupné, skryté a výstupné aktivity

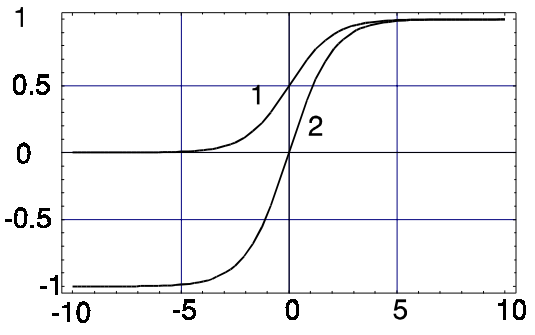
$$\boldsymbol{x}=\boldsymbol{x}_I\oplus\boldsymbol{x}_H\oplus\boldsymbol{x}_O \tag{5.12}$$

Neurónovú sieť s fixovanými váhami a prahovými koeficientmi možno formálne chápať ako funkciu

$$G:R^{N_I}\to(A,B)^{N_O} \tag{5.13}$$

Táto funkcia G priradí vstupnej aktivite $\boldsymbol{x}_I$ (deskriptor) výstupný vektor $\boldsymbol{x}_O$ (klasifikátor) s hodnotami svojich zložiek z otvoreného intervalu (A,B)

$$G(\boldsymbol{x}_I)=\boldsymbol{x}_O \tag{5.14}$$



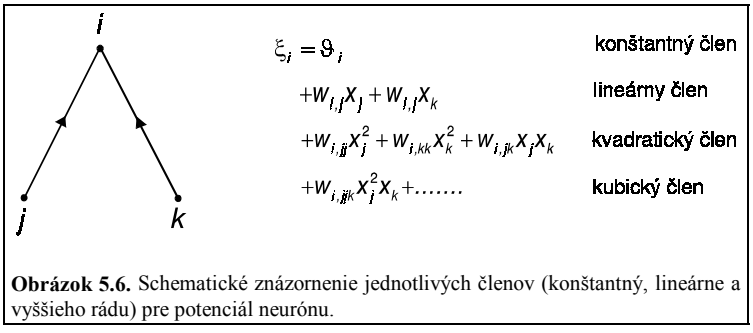
Obrázok 5.5. Priebeh aktivačnej funkcie definovanej (5.11a). Graf 1 odpovedá štandardnej sigmoide ($A=0$, $B=1$), graf 2 je podobný funkcii hyperbolický tangent ($A=-1$,$B=1$).

Skryté aktivity nie sú explicitne uvedené, hrajú len úlohu medzivýsledkov. Ešte niekoľko poznámok k výpočtu aktivít podľa (5.10a-b). Vstupné aktivity sú určené deskriptorom, preto ich pokladáme za fixované. Aktivity skrytých neurónov z druhej vrstvy L_2 môžeme teraz spočítať len použitím vstupných aktivít z vrstvy L_1. Vo všeobecnosti pre výpočet aktivít z vrstvy L_i (kde i>1) musíme poznať len aktivity z nižších vrstiev L_1, L_2,..., L_{i-1} Týmto rekurentným spôsobom môžeme postupne spočítať aktivity všetkých neurónov, ako posledné sa počítajú aktivity výstupných neurónov. Vďaka tomu sa pre neurónové siete reprezentované acyklickým grafom zaužíval názov *neurónové siete* s *dopredným šírením* (angl. *feed-forward neural networks*). Žiaľ, spomínaný jednoduchý postup výpočtu aktivít neurónov je aplikovateľný len na neurónové siete reprezentované acyklickým orientovaným grafom. V prípade, že graf obsahuje orientované cykly, tento postup nie je použiteľný. Rovnice (5.10a-b) sú v tomto prípade spriahnuté a nelineárne. Preto ich riešenie (t.j. skryté a výstupné aktivity) môžeme dosiahnuť len použitím iteračného postupu, a to tak, že štartujeme z počiatočných aktivít, pomocou týchto spočítame nové aktivity a tieto sa v nasledujúcom iteračnom kroku použijú ako vstup pre výpočet nových aktivít. Tento iteračný postup sa opakuje tak dlho, až rozdiel medzi starými a novými aktivitami je menší ako predpísaná presnosť.

V literatúre [4,5] je študovaných ešte mnoho ďalších modifikácií neurónových sietí s dopredným šírením. V ďalšej časti tejto kapitoly uvedieme dva typy, a to neurónovú sieť vyššieho rádu a adaptívnu kombináciu lokálnych neurónových sietí.

5.2.1 Neurónová sieť vyššieho rádu

Jednoduchou možnosťou, ako zovšeobecniť pojem neurónovej siete s dopredným šírením je zovšeobecnenie formuly (5.10b) pre potenciál ξ_i tak, aby obsahovala nielen konštantný člen (prahový faktor) a lineárne členy (vážené aktivity predchádzajúcich neurónov), ale aj členy vyššieho rádu [6] (kvadratické, kubické, ...), pozri obr. 5.6.



Obrázok 5.6. Schematické znázornenie jednotlivých členov (konštantný, lineárne a vyššieho rádu) pre potenciál neurónu.

Potom

$$x_i = t(\xi_i) \qquad (\text{pre } i = 1,2,...,N) \tag{5.15a}$$

$$\xi_i = \vartheta_i + \sum_j w_{i,j} x_j + \sum_{j \le k} w_{i,jk} x_j x_k + ... \tag{5.15b}$$

kde prvá sumácia beží cez všetky $j \in \Gamma_i^{-1}$, druhá sumácia beží cez všetky $j,k \in \Gamma_i^{-1}$, ktoré sú ohraničené podmienkou $j \le k$. Ak formula (5.15b) obsahuje nanajvýš kvadratické členy, potom neurónová sieť sa nazýva sieť druhého rádu. Vo všeobecnosti, členy najvyššieho rádu v (5.15b) určujú rád neurónovej siete. Neurónové siete vyššieho rádu dosahujú podobné výsledky ako neurónové siete prvého rádu, avšak s podstatne menším počtom skrytých neurónov. Je potrebné zdôrazniť, že táto vlastnosť neurónových sietí vyššieho rádu je získaná za "cenu" podstatne horších konvergentných vlastností adaptačného procesu (obrazne povedané, neurónové siete vyššieho rádu sú "viac nelineárne" ako neurónové siete prvého rádu).

5.2.2 Adaptívna kombinácia lokálnych neurónových sietí

Nech $\boldsymbol{N}$=(G,$\boldsymbol{w}$,$\boldsymbol{\vartheta}$) je neurónová sieť s dopredným šírením určená acyklickým orientovaným grafom G, pričom spoje a neuróny sú ohodnotené váhovými koeficientmi $\boldsymbol{w}$ resp. prahovými koeficientmi $\boldsymbol{\vartheta}$. Uvažujme *t lokálnych* neurónových sietí [7,8]

$$\boldsymbol{N}_i = \left(\mathrm{G}^{(i)}, \boldsymbol{w}^{(i)}, \boldsymbol{\vartheta}^{(i)}\right) \qquad (i = 1,2,...,t) \tag{5.16a}$$

a jednu tzv. bránovú (angl. *gating*) neurónovú sieť

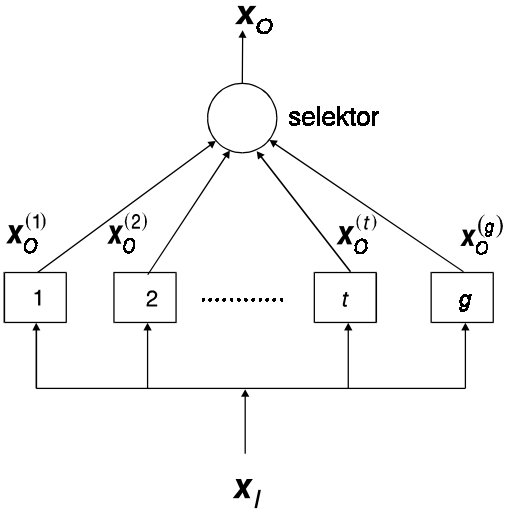
$$\boldsymbol{N}_g = \left(\mathrm{G}^{(g)}, \boldsymbol{w}^{(g)}, \boldsymbol{\vartheta}^{(g)}\right) \tag{5.16b}$$

Lokálne neurónové siete sú určené grafmi $G^{(1)}$, $G^{(2)}$, ..., $G^{(t)}$, ktoré sú ohraničené tak, že všetky obsahujú rovnaký počet vstupných a výstupných neurónov. Orientovaný graf $G^{(g)}$, priradený bránovej neurónovej sieti, má tiež rovnaký počet vstupných neurónov ako lokálne siete, ale počet jeho výstupných neurónov sa rovná počtu t lokálnych sietí (pozri obr. 5.7). Naviac sa predpokladá, že výstupné aktivity bránovej neurónovej siete sú z otvoreného intervalu (0,1), t.j. prechodová funkcia (5.11a) je špecifikovaná parametrami A=0, B=1. Označme ako $\boldsymbol{x}_O^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_t^{(i)}\right)$ resp. $\boldsymbol{x}_O^{(g)} = \left(x_1^{(g)}, x_2^{(g)}, \ldots, x_t^{(g)}\right)$ výstupné aktivity jednotlivých lokálnych sietí resp. bránovej siete ako odozvu na rovnaký vektor vstupných aktivít $\boldsymbol{x}_I$. Poznamenajme, že všetky zložky týchto vektorov výstupných aktivít sú z otvoreného intervalu (0,1). *Koeficienty proporcionality*, produkované bránovou sieťou, sa určia ako odozva na vstupný vektor $\boldsymbol{x}_I$

$$p_i = \frac{x_i^{(g)}}{\sum_{(j=1)}^{(t)} x_j^{(g)}} \qquad (i = 1,2,\ldots,t) \tag{5.17}$$

Tieto koeficienty nadobúdajú hodnoty z otvoreného intervalu (0,1) a ich suma sa rovná jednej

$$p_1 + p_2 + \ldots + p_t = 1 \tag{5.18}$$



Obrázok 5.7. Znázornenie adaptívnej kombinácie lokálnych neurónových sietí. Selektor realizuje výber výstupnej aktivity $\boldsymbol{x}_O$ na základe výstupných aktivít lokálnych neurónových sietí.

V našich ďalších úvahách tieto koeficienty proporcionality budeme interpretovať ako "pravdepodobnosti" toho, že príslušná lokálna neurónová sieť je použitá na klasifikáciu objektu popísaného deskriptorom — vstupnou aktivitou $\boldsymbol{x}_I$.

Ako sa určí výstupná aktivita $\boldsymbol{x}_O$ adaptívnej kombinácie lokálnych neurónových sietí? Pre vstupnú aktivitu spočítame výstupné aktivity všetkých lokálnych sietí a bránovej siete, $\boldsymbol{x}_O^{(1)}, \boldsymbol{x}_O^{(2)}, \ldots, \boldsymbol{x}_O^{(t)}, \boldsymbol{x}_O^{(g)}$. Výstupná aktivita celej siete môže byť určená ako konvexná[1] kombinácia výstupných aktivít jednotlivých lokálnych sietí, pričom koeficienty konvexnej kombinácie sú určené pomocou koeficientov proporcionality

$$\boldsymbol{x}_O = p_1 \boldsymbol{x}_O^{(1)} + p_2 \boldsymbol{x}_O^{(2)} + \ldots + p_t \boldsymbol{x}_O^{(t)} \tag{5.19}$$

Výstupný vektor $\boldsymbol{x}_O$ považujeme za odozvu adaptívnej kombinácie lokálnych sietí na vektor vstupných aktivít $\boldsymbol{x}_I$. V limitnom prípade — ak len jeden koeficient proporcionality p_i je blízky jednotke a ostatné sú skoro nulové — hovoríme, že i-ta lokálna sieť interpretuje objekt so vstupnou aktivitou $\boldsymbol{x}_I$, a táto sieť je teda "expert" na klasifikáciu daného objektu. Spôsob konštrukcie (výberu) môže byť realizovaný aj tak, že sa vyberie lokálna sieť s maximálnou hodnotou koeficientu proporcionality

$$j = \arg \max_{1 \le i \le t} p_i \tag{5.20}$$

potom $\boldsymbol{x}_O = \boldsymbol{x}_O^{(j)}$. To znamená, že adaptívna kombinácia lokálnych sietí poskytuje ako odozvu na vstupnú aktivitu $\boldsymbol{x}_I$ výstupnú aktivitu tej lokálnej siete, ktorá má maximálny koeficient proporcionality. V našich ďalších úvahách budeme používať formulu (5.19) pre určenie výstupnej aktivity adaptívnej kombinácie lokálnych sietí. Jej hlavnou výhodou pred ostatnými prístupmi (napr. pred prístupom založenom na maximálnej hodnote koeficienta proporcionality (5.20)) je jej "spojitosť" a "diferencovateľnosť".

5.3 Adaptácia neurónovej siete

Adaptácia neurónovej siete spočíva v hľadaní takých prahových a váhových koeficientov, ktoré pre danú dvojicu vstupného a požadovaného výstupného vektora $\boldsymbol{x}_I / \hat{\boldsymbol{x}}_O$ a vypočítaného výstupného vektora $\boldsymbol{x}_O$, určeného vzťahom (5.14), minimalizujú rozdiel medzi výstupnými aktivitami $\boldsymbol{x}_O$ a $\hat{\boldsymbol{x}}_O$. Zostrojme účelovú funkciu

$$E = \frac{1}{2}\left(\boldsymbol{x}_O - \hat{\boldsymbol{x}}_O\right)^2 = \frac{1}{2}\sum_k g_k^2 \tag{5.21a}$$

[1] Konvexná kombinácia vektorov $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ je taká lineárna kombinácia $\alpha_1\boldsymbol{x}_1+\alpha_2\boldsymbol{x}_2+\ldots+\alpha_n\boldsymbol{x}_n$, kde lineárne koeficienty α_i sú nezáporné a ich suma sa rovná jednej.

$$g_k = \begin{cases} (x_k - \hat{x}_k) & (\text{pre } k \in V_O) \\ 0 & (\text{pre } k \notin V_O) \end{cases} \tag{5.21b}$$

kde x_k a $\hat{x}_k$ sú komponenty vektorov $\boldsymbol{x}_O$ resp. $\hat{\boldsymbol{x}}_O$, a $\boldsymbol{a}^2$ je skalárny súčin $\boldsymbol{a}\cdot\boldsymbol{a}=\sum a_i^2$. Cieľom adaptačného procesu je nájdenie takých prahových a váhových koeficientov, ktoré minimalizujú účelovú funkciu E. Pre viac párov vstupných a výstupných vektorov

$$\boldsymbol{x}_I^{(1)} / \hat{\boldsymbol{x}}_O^{(1)}, \boldsymbol{x}_I^{(2)} / \hat{\boldsymbol{x}}_O^{(2)}, \ldots, \boldsymbol{x}_I^{(r)} / \hat{\boldsymbol{x}}_O^{(r)} \tag{5.22}$$

(ktoré tvoria tréningovú množinu), má účelová funkcia (5.21) tvar

$$E = \sum_{i=1}^{r} E^{(i)} \tag{5.23a}$$

$$E^{(i)} = \frac{1}{2}\left(\boldsymbol{x}_O^{(i)} - \hat{\boldsymbol{x}}_O^{(i)}\right)^2 \tag{5.23b}$$

kde $\boldsymbol{x}_O^{(i)}$ je výstupný vektor neurónovej siete, určený vzťahom (5.14) ako odozva na vstupný vektor $\boldsymbol{x}_I^{(i)}$ a $\hat{\boldsymbol{x}}_O^{(i)}$ je požadovaný výstupný vektor priradený vstupu $\boldsymbol{x}_I^{(i)}$.

5.3.1 Adaptačný proces perceptrónu

Perceptrón je najjednoduchšia forma neurónovej siete, ktorá obsahuje len dve vrstvy [1]. Spodná vrstva obsahuje p vstupných neurónov a horná vrstva obsahuje len jeden výstupný neurón (to znamená, že perceptrón neobsahuje skryté neuróny), pozri obr. 5.8. Orientované spoje sú ohodnotené váhovými koeficientmi w_i (kde index i vyjadruje index vstupného neurónu, z ktorého daná hrana vychádza) a výstupný neurón je ohodnotený prahovým koeficientom ϑ. Výstupná aktivita y je určená vzťahom (pozri (5.10a-b))

$$y = t\left(\vartheta + w_1 x_1 + w_2 x_2 + \ldots w_p x_p\right) \tag{5.24}$$

Predpokladajme, že tréningová množina obsahuje r párov $\mathbf{x}_1 / y_1$, $\mathbf{x}_2 / y_2, \ldots, \mathbf{x}_r / y_r$ kde $\boldsymbol{x}_i = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_p^{(i)}\right)$, pre i=1, 2,..., r, sú vektory vstupných aktivít. Budeme predpokladať, že tréningová množina je *neprotirečivá*, t.j. ak pre rôzne dva indexy i a j platí $\boldsymbol{x}_i=\boldsymbol{x}_j$, potom požadované príslušné aktivity tiež musia byť rovnaké, $y_i=y_j$. Rovnica (5.24) pre i-ty pár z tréningovej množiny má tvar

$$y_i = t\left(\vartheta + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \ldots + w_p x_p^{(i)}\right) \tag{5.25}$$

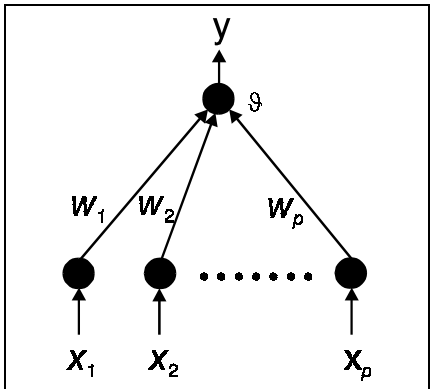
Pretože prechodová funkcia t je monotónne rastúca, musí k nej existovať inverzná funkcia t^{-1}. Potom

$$\vartheta + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \ldots + w_p x_p^{(i)} = t^{-1}(y_i) = \chi_i \tag{5.26}$$

Tieto rovnice tvoria systém lineárnych rovníc pre neznáme $\vartheta, w_1, \ldots, w_p$. Ich maticový tvar je

$$\boldsymbol{A}\boldsymbol{w} = \boldsymbol{\chi} \tag{5.27a}$$

kde $\boldsymbol{A}$ je obdĺžniková matica typu $r \times (p+1)$



Obrázok 5.8. Perceptrón, neurónová sieť s dopredným šírením, obsahujúca len dve vrstvy neurónov (vstupnú a výstupnú) .

$$\boldsymbol{A} = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \ldots & x_p^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \ldots & x_p^{(2)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_1^{(r)} & x_2^{(r)} & \ldots & x_p^{(r)} \end{pmatrix} \tag{5.27b}$$

a $\boldsymbol{w}$ resp. $\boldsymbol{\chi}$ sú stĺpcové vektory určené ako $\boldsymbol{w}=(\vartheta, w_1, \ldots, w_p)^{\mathrm{T}}$ resp. $\boldsymbol{\chi}=(\chi_1, \chi_2, \ldots, \chi_r)^{\mathrm{T}}$. To znamená, že adaptačný proces perceptrónu možno pretransformovať na riešenie systému lineárnych rovníc (5.26). Ak tento systém má riešenie, potom povieme, že tréningová množina (obsahujúca r párov tréningových vzorov $\boldsymbol{x}_i/y_i$) je *lineárne separovateľná*. V opačnom prípade, ak systém (5.26) nemá riešenie, perceptrón nemôže byť korektne adaptovaný (hovoríme, že tréningová množina *je lineárne neseparovateľná*).

V prípade, že systém (5.26) nemá riešenie (t.j. podľa Frobeniovej vety [9] vtedy a len vtedy, ak platí hodnosť($\boldsymbol{A}$)≠hodnosť($\boldsymbol{A}|\boldsymbol{\chi}$)), môžeme zostrojiť približné riešenie (v zmysle metódy najmenších štvorcov) použitím prístupu pseudoinverznej matice [9]. Násobme zľava rovnicu (5.27a) transponovanou maticou $\boldsymbol{A}^{\mathrm{T}}$, a dostaneme $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\chi}$, kde $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ je

pozitívne semidefinitná matica typu $r \times r$. Potom, formálnym aplikovaním matice $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}$ zľava, dostaneme konečné riešenie (5.27a) v tvare

$$\boldsymbol{w} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\chi \tag{5.28}$$

kde matica $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}$ sa nazýva pseudoinverzná matica. Riešenie (5.28) sa nazýva *zovšeobecnené riešenie* rovnice (5.27a) a minimalizuje euklidovskú normu $|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{\chi}|$. Ak $|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{\chi}|=0$, potom $\boldsymbol{w}$ je presné riešenie (5.27a). Poznamenajme, že aj keď sme v (5.28) použili explicitný výraz pre pseudoinverznú maticu, tento je platný len za predpokladu, že matica $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ je regulárna (t.j. existuje k nej inverzná matica). V opačnom prípade, ak matica $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ je singulárna (t.j. neexistuje k nej inverzná matica), existujú metódy jej konštrukcie, ktoré nepožadujú znalosť inverznej matice $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}$. Konštrukcia pseudoinverznej matice $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}$ patrí medzi štandardné numerické problémy, program pre jej implementáciu je uvedený napr. v monografii [10].

Vyššie uvedený adaptačný proces je ľahko zovšeobecniteľný aj pre perceptróny obsahujúce viac ako jeden výstupný neurón. V tomto prípade pre každý výstupný neurón zostrojíme nezávislé zovšeobecnené riešenie (5.28), ktoré je najlepšie pre daný výstupný neurón. Žiaľ, tento jednoduchý algebraický prístup k adaptácii perceptrónu je neaplikovateľný pre neurónovú sieť obsahujúcu skryté neuróny, pretože nie je možné linearizovať analógiu rovnice (5.25) pomocou inverznej prechodovej funkcie do tvaru systému lineárnych rovníc.

Ilustračný príklad (logická funkcia XOR)

Logická funkcia XOR zohrala v histórii neurónových sietí dôležitú úlohu. Koncom 60-tych rokov Minsky a Papert v známej knihe [1] kritizovali perceptrón ako prístup, ktorý nie je schopný realizovať ľubovoľnú výpočtovú úlohu (napr. XOR funkciu). Na základe tohto pozorovania dospeli k záveru, že neurónové siete (perceptróny patria medzi ne) nie sú univerzálnym výpočtovým prostriedkom.

Tabuľka 5.1. Funkcia XOR

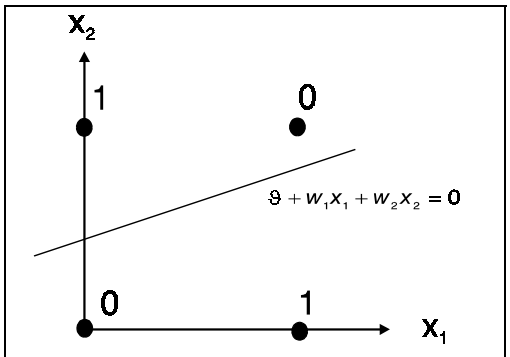
x_1	x_2	y(XOR)
0	0	0
0	1	1
1	0	1
1	1	0

Logická funkcia XOR (vylučujúce alebo) je určená pomocou tab. 5.1. Ide o boolovskú funkciu $y=F(x_1,x_2)$, kde x_1 a x_2 sú nezávislé dvojhodnotové premenné a y je závislá dvojhodnotová premenná. Pokúsime sa realizovať túto funkciu pomocou perceptrónu, ktorý obsahuje dva vstupné neuróny (s aktivitami x_1 a x_2) a jeden výstupný neurón s aktivitou y. Tréningová množina obsahuje 4 objekty z tab. 5.1. Systém (5.27a) má potom tvar

$$\begin{aligned} \vartheta \qquad\qquad &= -Q \\ \vartheta \qquad + w_2 &= Q \\ \vartheta + w_1 \qquad &= Q \\ \vartheta + w_1 + w_2 &= -Q \end{aligned} \tag{5.29}$$

kde Q je dostatočne veľké kladné číslo (určené tak, aby pre funkčné hodnoty prechodovej funkcie platilo $t(Q)=1-\varepsilon$ a $t(-Q)=\varepsilon$, pre malé kladné číslo ε). Systém (5.29) nemá riešenie (o čom sa môžeme jednoducho presvedčiť tak, že od štvrtej rovnice odpočítame druhú a tretiu rovnicu, dostaneme $\vartheta=3Q$, čo je v protirečení s prvou rovnicou).

Ako interpretovať tento výsledok? Argument v prechodovej funkcii vo vzťahu (5.24), ktorý špecifikuje aktivitu výstupného neurónu, možno formálne chápať ako rovnicu roviny $\vartheta+w_1x_1+w_2x_2+...+w_px_p=0$ v p-rozmernom priestore vstupných aktivít. Táto rovina rozdeľuje priestor na dva polpriestory, ktoré sú buď nad rovinou ($\vartheta+w_1x_1+w_2x_2+...+w_px_p>0$) alebo pod rovinou ($\vartheta+w_1x_1+w_2x_2+...+w_px_p<0$). Zhruba povedané, bod ležiaci nad (pod) rovinou má jednotkovú (nulovú) výstupnú aktivitu. Pretože systém (5.29) nemá riešenie, neexistuje rovina, ktorá korektne rozdeľuje 2-rozmerný priestor so súradnicami x_1 a x_2 na dva podpriestory tak, aby body (0,0) a (1,1) ležali pod rovinou a body (0,1) a (1,0) nad rovinou (v tomto prípade ide o priamkou určenú rovnicou $\vartheta+w_1x_1+w_2x_2=0$), pozri obr. 5.9. Môžeme teda povedať, že dve triedy objektov XOR problému sú lineárne neseparovateľné.

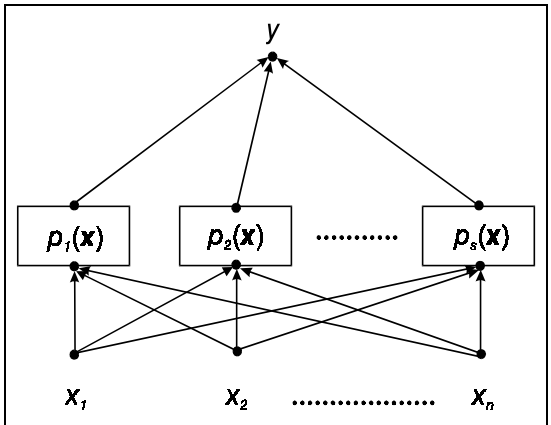


Obrázok 5.9. Znázornenie "nadroviny" (určenej rovnicou $\vartheta+w_1x_1+w_2x_2=0$), ktorá rozdeľuje priestor vstupných aktivít na dva podpriestory.

5.3.2 Adaptačný proces perceptrónu vyššieho rádu

Perceptrón vyššieho rádu je zovšeobecnením obyčajného perceptrónu, a to takým spôsobom, že jeho výstupná aktivita je určená nielen lineárnymi členmi, ale aj kvadratickými, kubickými, atď., členmi (pozri definíciu neurónovej siete vyššieho rádu v podkapitole 5.2.1). Aktivita výstupného neurónu je určená vzťahom

$$y = t\left(\vartheta + w_1 p_1(\boldsymbol{x}) + w_2 p_2(\boldsymbol{x}) + \ldots + w_s p_s(\boldsymbol{x})\right) \tag{5.30}$$



Obrázok 5.10. Schematické znázornenie perceptrónu vyššieho rádu. Obdĺžnikové bloky sú priradené multilineárnym funkciám $p_i(\boldsymbol{x})$.

kde $p_i(\boldsymbol{x}) = x^{\alpha_1} x^{\alpha_2} \ldots x^{\alpha_n}$ sú rôzne multilineárne členy určené exponentmi α_1, α_2, ..., α_p. Diagramatická interpretácia perceptrónu vyššieho rádu je znázornená na obr. 5.10, kde obdĺžnikové bloky reprezentujú multilineárne funkcie $p_i(\boldsymbol{x})$. Ak študujeme perceptrón druhého rádu, potom exponenty vyhovujú buď podmienke $\alpha_1+\alpha_2+...+\alpha_p=1$ (lineárny člen) alebo $\alpha_1+\alpha_2+...+\alpha_p=2$ (kvadratický člen).

Výstupná aktivita je určená vzťahom

$$y = t\left(\vartheta + w_1 x_1 + w_2 x_2 + w_{12} x_1 x_2 + w_{11} x_1^2 + w_{22} x_2^2 + \ldots\right) \tag{5.31}$$

kde sme pre jednoduchosť uviedli len niekoľko prvých členov. Použitím rovnakej linearizačnej procedúry ako v podkapitole 5.3.1 dostaneme rovnaký systém lineárnych rovníc ako (5.27a), matica $\boldsymbol{A}$ má tvar

$$\boldsymbol{A} = \begin{pmatrix} 1 & p_1\left(\boldsymbol{x}^{(1)}\right) & p_2\left(\boldsymbol{x}^{(1)}\right) & \ldots & p_s\left(\boldsymbol{x}^{(1)}\right) \\ 1 & p_1\left(\boldsymbol{x}^{(2)}\right) & p_2\left(\boldsymbol{x}^{(2)}\right) & \ldots & p_s\left(\boldsymbol{x}^{(2)}\right) \\ & & \ldots\ldots\ldots\ldots & & \\ 1 & p_1\left(\boldsymbol{x}^{(r)}\right) & p_2\left(\boldsymbol{x}^{(r)}\right) & \ldots & p_s\left(\boldsymbol{x}^{(r)}\right) \end{pmatrix} \tag{5.32}$$

Prahové a váhové koeficienty sú určené systémom lineárnych rovníc (5.27a), alebo explicitne vzťahom (5.28).

Predpokladajme, že rovnica (5.27a) nemá riešenie, t.j. platí hodnosť($\boldsymbol{A}$)≠hodnosť($\boldsymbol{A}|\boldsymbol{\chi}$). Zavedením nového multilineárneho člena v (5.30) dostaneme novú maticu koeficientov $\boldsymbol{A}'$

(matica $\boldsymbol{A}$ je rozšírená sprava o nový stĺpec), a potom platí jeden z nasledujúcich dvoch prípadov:

(1) Nová matica $\boldsymbol{A}'$ už vyhovuje podmienke hodnosť($\boldsymbol{A}'$)=hodnosť($\boldsymbol{A}'|\boldsymbol{\chi}$), a potom systém (5.28) má riešenie. To znamená, že prahové a váhové koeficienty perceptrónu sú určené riešením systému lineárnych rovníc (5.27a).

(2) Nová matica $\boldsymbol{A}'$ stále nevyhovuje podmienke hodnosť($\boldsymbol{A}'$)= =hodnosť($\boldsymbol{A}'|\boldsymbol{\chi}$), a potom riešenie $\boldsymbol{w}'$ zostrojené pomocou pseudoinverznej matice (5.28) vyhovuje podmienke $|\boldsymbol{A}'\boldsymbol{w}' - \boldsymbol{\chi}| < |\boldsymbol{A}\boldsymbol{w} - \boldsymbol{\chi}|$, t.j. nové riešenie $\boldsymbol{w}'$ je zostrojené s menšou chybou (v zmysle euklidovskej vzdialenosti) ako pôvodné riešenie $\boldsymbol{w}$. To znamená, že ak zavedieme nový multilineárny člen v (5.30), potom prahové a váhové koeficienty určené pomocou pseudoinverznej matice (5.28) poskytujú lepšiu klasifikáciu objektov z tréningovej množiny. V limitnom prípade, ak sme zaviedli dostatočný počet multilineárnych členov môže nastať situácia, že buď podmienka riešiteľnosti hodnosť($\boldsymbol{A}$)=hodnosť($\boldsymbol{A}|\boldsymbol{\chi}$) začne platiť, alebo stále hodnosť($\boldsymbol{A}$)≠hodnosť($\boldsymbol{A}|\boldsymbol{\chi}$), avšak "nepresnosť" $|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{\chi}|$ je už menšia ako predpísané malé kladné číslo ε (riešenie $\boldsymbol{w}$ je "správne" s presnosťou ε). Táto jednoduchá procedúra rozširovania perceptrónu novými multilineárnymi členmi vyššieho rádu je umožnená tým, že použité multilineárne členy $p_1(\boldsymbol{x})$, $p_2(\boldsymbol{x})$, ..., $p_s(\boldsymbol{x})$ sú lineárne nezávislé.

Možno teda povedať, *že perceptrón dostatočne vysokého rádu je schopný klasifikovať korektne s predpísanou presnosťou ε ľubovoľnú neprotirečivú tréningovú množinu*. Toto je veľmi dôležitá vlastnosť perceptrónov vyššieho rádu, ináč povedané, tieto perceptróny sú *univerzálne aproximátory funkcií*, t.j. sú schopné ich aproximovať s ľubovoľnou predpísanou presnosťou.

Ilustračný príklad (logická funkcia XOR)

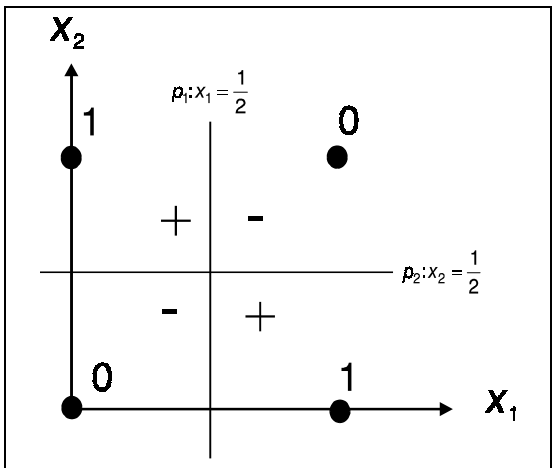
V predchádzajúcom ilustračnom príklade sme ukázali, že obyčajný perceptrón (t.j. perceptrón prvého rádu) nie je schopný klasifikovať logickú funkciu XOR. Toto obmedzenie obyčajného perceptrónu len na lineárne separovateľné vstupné aktivity objektov z tréningovej množiny sa jednoducho odstráni použitím perceptrónu vyššieho rádu. Na klasifikáciu XOR funkcie použijeme perceptrón 2. rádu. Nech potenciál výstupného neurónu obsahuje člen $w_{12}x_1x_2$ (ďalšie členy 2. rádu $w_{11}x_1^2$ a $w_{22}x_2^2$ nie sú uvažované, pretože v dôsledku binárneho charakteru vstupných aktivít x_1 a x_2 sú totožné s členmi 1. rádu). Analógia systému lineárnych rovníc (5.29) má tvar

$$\begin{aligned} \vartheta &= -Q \\ \vartheta + w_2 &= Q \\ \vartheta + w_1 &= Q \\ \vartheta + w_1 + w_2 + w_{12} &= -Q \end{aligned} \qquad (5.33)$$

Tento systém rovníc (pre neznáme ϑ, w_1, w_2 a w_{12}) má jednoznačné riešenie (hodnosť($\boldsymbol{A}$)=hodnosť($\boldsymbol{A}|\boldsymbol{\chi}$)=4), a platí $\vartheta=-Q$, $w_1=w_2=2Q$, $w_{12}=-4Q$. Uvažuj-me funkciu 2. rádu (potenciál výstupného neurónu) určenú implicitne vzťahom $-Q+2Qx_1+2Qx_2-4Qx_1x_2=0$, alebo v zjednodušenom tvare

$$x_1 + x_2 - 2x_1x_2 = \frac{1}{2} \tag{5.34}$$

Táto rovnica má dve nezávislé riešenia, ktoré určujú dve priamky kolmé na osy x_1 alebo x_2



Obrázok 5.11. Funkcia 2. rádu (5.34) rozdeľuje priestor vstupných aktivít (pomocou priamok p_1 a p_2 určených vzťahmi (5.35)) na 4 podpriestory, pričom je dosiahnutá správna interpretácia XOR problému.

$$p_1: x_1 = \frac{1}{2}, \qquad p_2: x_2 = \frac{1}{2} \tag{5.35}$$

Tieto dve priamky určujú hraničné oblasti v rovine x_1-x_2, kde argument (potenciál) prechodovej funkcie v (5.31) mení znamienko (pozri obr. 5.11). Perceptrón druhého rádu je schopný korektne klasifikovať XOR funkciu, priestor vstupných aktivít je rozdelený na 4 podpriestory, kde potenciál nadobúda správne znamienko. Tento jednoduchý príklad môže byť chápaný ako ilustrácia toho, že perceptrón vyššieho rádu je schopný klasifikovať objekty neprotirečivej tréningovej množiny.

5.3.3 Adaptácia neurónovej siete s dopredným šírením

Adaptačný proces neurónovej siete s dopredným šírením, ktorá obsahuje skryté neuróny, nemôže byť uskutočnený takým jednoduchým spôsobom ako pre perceptrón, kde sa adaptačný proces redukuje na riešenie systému lineárnych rovníc (použitím prístupu pseudoinverznej matice). Pre neurónové siete, ktoré obsahujú skryté neuróny, nie je možné linearizovať systém rovníc, ktoré popisujú aktivity skrytých a výstupných neurónov. Preto musíme obrátiť našu pozornosť na takú adaptáciu neurónovej siete, ktorá minimalizuje účelovú funkciu (5.21) alebo (5.23). Túto minimalizáciu nelineárnej účelovej funkcie možno uskutočniť mnohými optimalizačnými metódami, ktoré sú známe v numerickej

matematike [11]. Medzi najefektívnejšie patria tzv. gradientové metódy, založené na použití gradientu účelovej funkcie pre iteratívnu konštrukciu optimálneho riešenia. Pri ich použití musíme poznať gradient účelovej funkcie (t.j. parciálne derivácie $\partial E/\partial \vartheta_i$ a $\partial E/\partial w_{ij}$) a jeho výpočet bude náplňou tejto podkapitoly.

Parciálne derivácie účelovej funkcie E (5.21) vzhľadom k prahovým a váhovým faktorom sú určené vzťahmi (jednoduchá aplikácia formuly pre parciálnu deriváciu zloženej funkcie [12])

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial w_{ij}} = \frac{\partial E}{\partial x_i} t'(\xi_i) x_j \\ \frac{\partial E}{\partial \vartheta_i} &= \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial \vartheta_i} = \frac{\partial E}{\partial x_i} t'(\xi_i) \end{aligned} \tag{5.36}$$

kde parciálna derivácia $\partial x_i/\partial w_{ij}$ je jednoducho spočítaná pomocou $x_i=t(\xi_i)$ (pozri (5.10)). Potom dostaneme $\partial x_i/\partial w_{ij} = t'(\xi_i)\cdot\partial\xi_i/\partial w_{ij} = t'(\xi_i)x_j$. Podobné úvahy sú aplikovateľné aj pre výpočet parciálnej derivácie $\partial x_i/\partial \vartheta_i$. Porovnaním rovníc (5.36) dostaneme jednoduchý vzťah medzi parciálnymi deriváciami $\partial x_i/\partial w_{ij}$ a $\partial x_i/\partial \vartheta_i$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \vartheta_i} x_j \tag{5.37}$$

Výpočet gradientu sa redukuje na výpočet parciálnych derivácií účelovej funkcie vzhľadom k prahovým faktorom, podľa (5.37) parciálne derivácie vzhľadom k váhovým koeficientom sú určené pomocou jednoduchších parciálnych derivácií vzhľadom k prahovým koeficientom. Upriamime našu pozornosť na výpočet parciálnych derivácií $\partial E/\partial x_i$. Ich výpočet závisí od toho, či index i popisuje výstupný neurón alebo skrytý neurón,

$$\begin{aligned} \frac{\partial E}{\partial x_i} &= g_i \qquad (\text{pre } i \in V_O) \\ \frac{\partial E}{\partial x_i} &= \sum_k \frac{\partial E}{\partial x_k}\frac{\partial x_k}{\partial x_i} \quad (pre\ i \in V_H) \end{aligned} \tag{5.38}$$

kde v druhom výraze sumácia beží cez všetky neuróny, ktoré nasledujú za i-tym neurónom. Výraz g_i je určený vzťahom (5.21b), ktorý je nulový pre iné neuróny ako výstupné. Formuly z (5.38) môžeme zjednotiť do jedného vzťahu

$$\frac{\partial E}{\partial x_i} = g_i + \sum_k \frac{\partial E}{\partial x_k}\frac{\partial x_k}{\partial x_i} \tag{5.39}$$

kde je potrebné si uvedomiť, že člen na pravej strane obsahujúci sumáciu, ako už bolo uvedené vyššie, je nulový pre index i popisujúci výstupný neurón. Podobne, ako v (5.36), pre parciálne derivácie z pravej strany (5.39) platí $\partial x_k/\partial x_i = t'(\xi_k)w_{ki}$, dosadením (5.39)

do (5.36) dostaneme konečnú formulu pre výpočet parciálnych derivácií účelovej funkcie E vzhľadom k prahovým faktorom

$$\frac{\partial E}{\partial \vartheta_i} = t'(\xi_i)\left(g_i + \sum_k \frac{\partial E}{\partial \vartheta_k} w_{ki} \right) \qquad \left(\text{pre } i \in \mathsf{V}_H \cup \mathsf{V}_O\right) \tag{5.40}$$

kde derivácia prechodovej funkcie $t'(\xi_i)$ je určená vzťahom (5.11b) a sumácia beží cez všetky neuróny, ktoré sú nasledovníkmi i-teho neurónu.

Vo všeobecnosti možno charakterizovať vzťah (5.40) ako systém lineárnych rovníc, ktorého riešenie určujú parciálne derivácie $\partial E/\partial \vartheta_i$. Pre neurónovú sieť typu dopredného šírenia je možné zostrojiť riešenie tohto systému jednoduchým rekurentným postupom. V prvom kroku vypočítame parciálne derivácie $\partial E/\partial \vartheta_i$ pre výstupné neuróny (výstupná vrstva L_t), pre ktoré platí $\partial E/\partial \vartheta_i = t'(\xi_i)\, g_i$. Pomocou (5.40) potom môžeme spočítať parciálne derivácie $\partial E/\partial \vartheta_i$ pre neuróny z predposlednej vrstvy L_{t-1}. Pri výpočte parciálnych derivácií $\partial E/\partial \vartheta_i$ pre neuróny z vrstvy L_j musíme poznať tieto derivácie z nasledujúcich vrstiev L_{j+1}, L_{j+2}, ..., L_t. Výpočet končí, keď zostrojíme parciálne derivácie pre neuróny z druhej vrstvy L_2. Poznajúc všetky parciálne derivácie $\partial E/\partial \vartheta_i$ pre celú neurónovú sieť, určíme parciálne derivácie $\partial E/\partial w_{ij}$ jednoducho pomocou vzťahu (5.37). Spôsob výpočtu parciálnych derivácií pre neurónovú sieť s dopredným šírením popísaný vyššie, prebiehajúci rekurentne od najvyššej k najnižšej vrstve (t.j. proti smeru šírenia informácie v neurónovej sieti, ktorá prebieha od najnižšej k najvyššej vrstve — tzv. dopredné šírenie) je aj hlavným dôvodom toho, prečo sa tento postup v literatúre často nazýva *spätné šírenie* (angl. *back propagation*). Vzťah podobný formule (5.40) pre výpočet parciálnych derivácií účelovej funkcie vzhľadom k prahovým a váhovým koeficientom bol odvodený v r. 1986 Rumelhartom so spolupracovníkmi [2]. Táto práca je pokladaná za jeden z historických medzníkov rozvoja teórie neurónových sietí, pretože v nej bolo ukázané na mnohých príkladoch (ktoré boli evidentne lineárne neseparovateľné), že zovšeobecnenie perceptrónu tak, aby obsahoval skryté neuróny, spolu s metódou spätného šírenia pre výpočet gradientu účelovej funkcie, je schopné prekonať limity stanovené Minskym a Papertom [1] pre jednoduchý perceptrón neobsahujúci skryté neuróny.

Pri odvodení formuly (5.40) nebol použitý predpoklad, že graf reprezentujúci neurónovú sieť je acyklický (t.j. neurónová sieť je typu dopredného šírenia). Preto táto formula platí pre ľubovoľnú neurónovú sieť, ktorá môže obsahovať aj orientované cykly (tzv. rekurentné siete, pozri kapitolu 6). Avšak v tomto prípade už nie je použiteľný postup spätného šírenia pre výpočet parciálnych derivácií, tieto sú teraz určené ako riešenie systému lineárnych rovníc (5.40) pre parciálne derivácie $\partial E/\partial \vartheta_i$. Totiž sumácia v (5.40) môže obsahovať vo všeobecnosti aj parciálne derivácie $\partial E/\partial \vartheta_i$, ktoré ešte neboli spočítané v predchádzajúcich krokoch rekurentného výpočtu.

Vyššie uvedený postup pre výpočet gradientu účelovej funkcie je ľahko zovšeobecniteľný aj pre účelovú funkciu, ktorá obsahuje viac ako jeden pár vstupno-výstupných vektorov $\boldsymbol{x}_l / \hat{\boldsymbol{x}}_O$ (pozri (5.23a-b)). Potom celkový gradient účelovej funkcie sa jednoducho určí ako suma gradientov spočítaných pomocou (5.40) pre všetky dvojice $\boldsymbol{x}_l / \hat{\boldsymbol{x}}_O$ z tréningovej množiny (5.22)

$$\text{grad } E = \sum_{i=1}^{r} \text{grad } E^{(i)} \tag{5.41}$$

kde účelová funkcia $E^{(i)}$ je definovaná (5.23b) pre *i*-tu dvojicu $\boldsymbol{x}_I / \hat{\boldsymbol{x}}_O$ z tréningovej množiny.

Ak poznáme gradient účelovej funkcie *E*, potom adaptačný proces neurónovej siete je realizovaný minimalizáciou účelovej funkcie *E* vzhľadom k prahovým a váhovým koeficientom. Formálne, adaptovaná neurónová sieť je popísaná koeficientmi, ktoré sú určené ako

$$\left(\overline{\boldsymbol{w}}, \overline{\boldsymbol{\vartheta}}\right) = \arg \min_{(\boldsymbol{w}, \boldsymbol{\vartheta})} E(\boldsymbol{w}, \boldsymbol{\vartheta}) \tag{5.42}$$

Jedným z najjednoduchších spôsobov (súčasne aj najviac používaným), ako realizovať túto minimalizáciu v rámci gradientových optimalizačných metód je metóda *najprudšieho spádu* (angl. *steepest descent*) [11], v ktorej váhové a prahové koeficienty sú rekurentne obnovované pomocou vzťahov

$$\begin{aligned} w_{ij}^{(k+1)} &= w_{ij}^{(k)} - \lambda \frac{\partial E}{\partial w_{ij}} + \mu \Delta w_{ij}^{(k)} \\ \vartheta_j^{(k+1)} &= \vartheta_j^{(k)} - \lambda \frac{\partial E}{\partial \vartheta_j} + \mu \Delta \vartheta_j^{(k)} \end{aligned} \tag{5.43}$$

kde parameter $\lambda>0$ musí byť dostatočne malý (obvykle $\lambda=0{,}01$-$0{,}1$), aby bola zabezpečená monotónna konvergentnosť optimalizačného algoritmu a súčasne dostatočne veľký pre zabezpečenie dostatočne vysokej rýchlosti konvergentnosti. Počiatočné hodnoty prahových a váhových koeficientov $\vartheta_j^{(0)}$ a $w_{ij}^{(0)}$ sú náhodne generované z malého intervalu so stredom v nule, napr. z otvoreného intervalu $(-1,1)$. Posledný člen v (5.43) odpovedá tzv. *momentovému členu*, ktorý je určený pomocou rozdielu koeficientov z posledných dvoch iterácií, $\Delta w_{ij}^{(k)} = w_{ij}^{(k)} - w_{ij}^{(k-1)}$ a $\Delta \vartheta_j^{(k)} = \vartheta_j^{(k)} - \vartheta_j^{(k-1)}$. Momen-tový člen (momentum) je dôležitý pre "obskočenie" lokálnych miním v počia-točnej fáze optimalizácie, hodnota parametru μ sa obvykle volí z intervalu $0{,}5 \le \mu \le 0{,}7$.

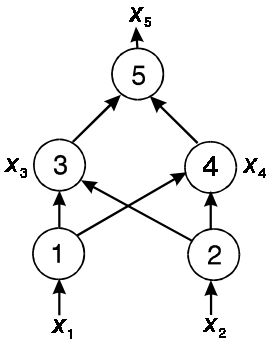
Ilustračný príklad (logická funkcia XOR)

Efektívnosť neurónovej siete obsahujúcej skryté neuróny ilustrujeme na príklade logickej funkcie XOR, ktorá pre obyčajný perceptrón nie je korektne klasifikovateľná. Použitá neurónová sieť bude obsahovať tri vrstvy, prvá vrstva obsahuje dva vstupné neuróny, druhá vrstva dva skryté neuróny a posledná tretia vrstva obsahuje jeden výstupný neurón (pozri obr. 5.12).

Tabuľka 5.2 Aktivity neurónovej siete z obr. 5.12
pre interpretáciu XOR problému

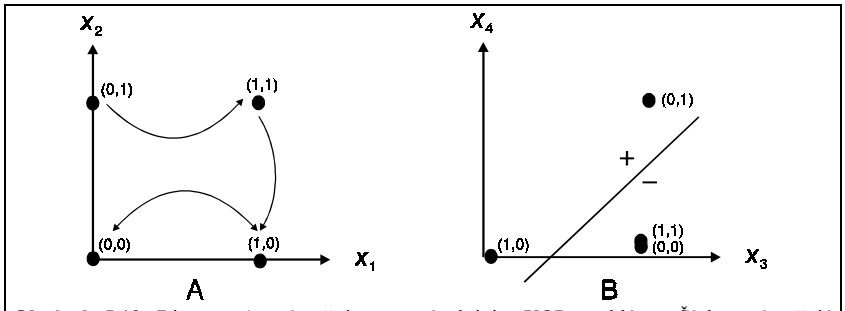
Čís.	x_1	x_2	x_3	x_4	x_5	$\hat{x}_5$
1	0,00	0,00	0,96	0,08	0,06	0,00
2	0,00	1,00	1,00	0,89	0,95	1,00
3	1,00	0,00	0,06	0,00	0,94	1,00
4	1,00	1,00	0,96	0,07	0,05	0,00

Skryté neuróny vytvárajú tzv. vnútornú reprezentáciu funkcie XOR, ktorá už je lineárne separovateľná. Táto skutočnosť, že skryté neuróny sú schopné zaviesť reprezentáciu, v ktorej sú už objekty správne interpretované, je hlavným dôvodom širokého používania neurónových sietí. Parametre adaptačného procesu boli tieto: λ=0,1, μ=0,5; po 400 iteráciach účelová funkcia mala hodnotu E=0,031. Výsledné aktivity skrytých a výstupných neurónov sú uvedené v tab. 5.2. Ak nakreslíme výsledné aktivity skrytých neurónov do roviny x_3-x_4 , vidíme, že tieto poskytujú vnútornú reprezentáciu, ktorá je lineárne separovateľná (pozri obr. 5.13).



Obrázok 5.12. Znázornenie jednoduchej neurónovej siete s dvoma skrytými neurónmi, ktorá je schopná správne interpretovať XOR problém.

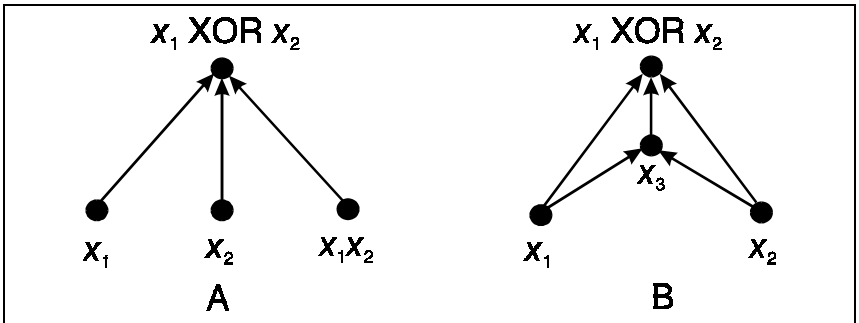
Perceptrón 2. rádu pre XOR funkciu (pozri podkapitolu 5.3.2, ilustračný príklad) je jednoducho realizovateľný pomocou neurónovej siete s jedným skrytým neurónom tak, že zavedieme novú (*funkcionálnu*) vstupnú aktivitu $x_3=x_1 \wedge x_2$ (t.j. nová vstupná aktivita je vytvorením funkcie AND aplikovanej na pôvodné vstupné aktivity x_1 a x_2), pozri obr. 5.14. Perceptrón A môže byť jednoducho pretransformovaný na neurónovú sieť B, ktorá obsahuje jeden skrytý neurón, pričom tento neurón vykonáva logickú funkciu AND.



Obrázok 5.13. Diagram A znázorňuje vstupné aktivity XOR problému. Šípky znázorňujú presun objektov v dôsledku adaptácie siete. Diagram B znázorňuje skryté aktivity neurónovej siete, ktoré sú priradené jednotlivým "obrazcom" XOR problému. Čo je dôležité, skryté aktivity už sú lineárne separovateľné "nadrovinou" určenou $\vartheta_5+w_{45}x_4+w_{35}x_3=0$.

5.3.4 Adaptácia neurónovej siete vyššieho rádu

Výpočet parciálnych derivácií $\partial E/\partial \vartheta_i$ a $\partial E/\partial w_{ji}$ pre neurónové siete vyššieho rádu je analógiou postupu pre neurónové siete s dopredným šírením prvého rádu (pozri podkapitolu 5.3.3). Pre jednoduchosť predpokladajme, že neurónová sieť je druhého rádu, t.j. aktivity (5.15a-b) sú určené lineárnymi a kvadratickými členmi. Zovšeobecnenie pre neurónové siete tretieho alebo vyššieho rádu je jednoduché. Parciálne derivácie účelovej funkcie vzhľadom k váhovým koeficientom prvého a druhého rádu sú určené takto (pozri (5.37))



Obrázok 5.14. Diagram A znázorňuje jednoduchú realizáciu perceptrónu 2. rádu zavedením nového vstupného neurónu, ktorému je priradené vstupná aktivita vyjadrená ako súčin (logická funkcia AND) aktivít prvého a druhého vstupného neurónu. Diagram B znázorňuje najjednoduchšiu neurónovú sieť s jedným skrytým neurónom, ktorého aktivita je realizovaná ako AND funkcia. Pripomeňme, že problém AND je lineárne separovateľný, čiže je možné zostrojiť pre tento problém jednoduchý perceptrón (ktorého výstupný neurón odpovedá skrytému neurónu 3).

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial \vartheta_i} x_j$$
$$\frac{\partial E}{\partial w_{i,jk}} = \frac{\partial E}{\partial \vartheta_i} x_j x_k \tag{5.44}$$

Parciálne derivácie $\partial E/\partial \vartheta_i$ sú určené systémom lineárnych rovníc (pozri (5.40))

$$\frac{\partial E}{\partial \vartheta_i} = t'(\xi_i)\left[g_i + \sum_k \frac{\partial E}{\partial \vartheta_k}\left(w_{k,i} + 2w_{k,ii}\, x_i + \sum_{\substack{j\neq k \\ i<j}} w_{k,ij} x_j \right)\right] \tag{5.45}$$

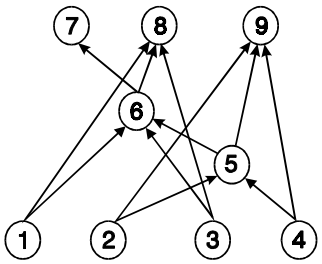
Druhý člen na pravej strane $2w_{k,ii}\, x_i$ odpovedá "čistému" kvadratickému členu $w_{k,ii}x_i^2$ v (5.15b), zatiaľ čo tretí člen $\sum_{(j\neq k, i<j)} w_{k,ij} x_j$ odpovedá "krížovým" členom v (5.15b). Rekurentná formula pre obnovu prahových a váhových koeficientov je analogická formulám (5.43) pre obyčajnú neurónovú sieť prvého rádu.

Ilustračný príklad (sumácia dvoch 2-bitových čísel)

Študujme sumáciu dvoch 2-bitových čísel

$$\begin{array}{r} \alpha_1\ \alpha_2 \\ +\ \alpha_3\ \alpha_4 \\ \hline \alpha_5\ \alpha_6\ \alpha_7 \end{array} \tag{5.46}$$

kde α_i sú binárne čísla. Tab. 5.3 obsahuje všetkých 16 možných realizácií súčtu (5.46), v poslednom stĺpci je uvedená dekadická interpretácia daného súčtu. Rumelhart so spolupracovníkmi [2] navrhli neurónovú sieť špeciálneho tvaru, ktorá obsahuje dva skryté neuróny a je schopná korektne interpretovať súčet (5.46). Skryté neuróny majú význam dvoch medzisúčtov, ktoré sú potrebné pre realizáciu celkového súčtu (medzisúčty pre druhý a prvý stĺpec v (5.46)), pozri obr. 5.15. Avšak s nimi navrhnutou neurónovou sieťou mali vážne problémy pri jej adaptačnom procese.
Z týchto dôvodov Rumelhart so spolupracovníkmi použili na klasifikáciu súčtu (5.46) neurónovú sieť s 3 alebo 4 skrytými neurónmi, ktoré už nemali konvergentné problémy. Tieto problémy adaptačného procesu neurónovej siete znázornenej na obr. 5.15 sa odstránia, ak sa tá bude interpretovať ako neurónová sieť druhého rádu. Po 3000 iteráciách bola adaptácia úspešná, hodnota účelovej funkcie je E=0,03, pre parametre λ=0,1 a μ=0,5. Výsledné aktivity skrytých a výstupných neurónov sú uvedené v tab. 5.4. Posledné tri stĺpce v tejto tabuľke odpovedajú výstupným aktivitám, ktoré sa priamo vzťahujú k požadovaným aktivitám (bitovým premenným) v sumácii (5.46). V prípade neurónovej siete 2. rádu aktivity skrytých neurónov už nemajú význam medzisumácií v (5.46).



Obrázok 5.15. Štruktúra neuróno-vej siete, ktorá je schopná riešiť sumáciu dvoch 2-bitových čísel.

Tabuľka 5.3. Hodnoty binárnych premenných sumácie dvoch 2-bitových čísel z (5.46)

Čís.	α_1	α_2	α_3	α_4	α_5	α_6	α_7	Význam
1	0	0	0	0	0	0	0	0+0=0
2	0	0	0	1	0	0	1	0+1=1
3	0	0	1	0	0	1	0	0+2=2
4	0	0	1	1	0	1	1	0+3=3
5	0	1	0	0	0	0	1	1+0=1
6	0	1	0	1	0	1	0	1+1=2
7	0	1	1	0	0	1	1	1+2=3
8	0	1	1	1	1	0	0	1+3=4
9	1	0	0	0	0	1	0	2+0=2
10	1	0	0	1	0	1	1	2+1=3
11	1	0	1	0	1	0	0	2+2=4
12	1	0	1	1	1	0	1	2+3=5
13	1	1	0	0	0	1	1	3+0=3
14	1	1	0	1	1	0	0	3+1=4
15	1	1	1	0	1	0	1	3+2=5
16	1	1	1	1	1	1	0	3+3=6

Tabuľka 5.4. Hodnoty aktivít skrytých a výstupných neurónov siete určenej obr. 5.15

Čís.	x_5	x_6	x_7	x_8	x_9
1	1,00	1,00	0,00	0,00	0,00
2	0,29	1,00	0,00	0,11	0,95
3	1,00	1,00	0,00	0,93	0,04
4	0,29	0,81	0,01	1,00	0,95
5	0,29	1,00	0,00	0,11	0,95
6	0,00	1,00	0,00	0,88	0,05
7	0,29	0,81	0,01	1,00	0,95
8	0,00	0,99	0,99	0,00	0,05
9	1,00	1,00	0,00	0,93	0,04
10	0,29	0,81	0,00	1,00	0,95
11	1,00	0,13	0,99	0,00	0,04
12	0,29	0,00	1,00	0,10	0,95
13	0,29	0,81	0,01	1,00	0,95
14	0,00	0,16	0,99	0,00	0,05
15	0,29	0,00	1,00	0,10	0,95
16	0,00	0,00	1,00	1,00	0,05

5.3.5 Adaptácia kombinácie lokálnych neurónových sietí

Účelová funkcia pre kombináciu *t* lokálnych sietí má tvar

$$E = \frac{1}{2}\sum_{i=1}^{t} p_i\left(x_O^{(i)} - \hat{x}_O\right)^2 = \frac{1}{2}\sum_{i=1}^{t}\sum_{k} g_k^{(i)2} \tag{5.47a}$$

$$g_k^{(i)} = \begin{cases} p_i\left(x_k^{(i)} - \hat{x}_k\right) & \left(\text{pre } k \in V_O\right) \\ 0 & \left(\text{pre } k \notin V_O\right) \end{cases} \tag{5.47b}$$

kde $x_k^{(i)}$ a $\hat{x}_k$ je vypočítaná resp. požadovaná výstupná aktivita k-teho neurónu i-tej lokálnej neurónovej siete. Tento vzťah pre účelovú funkciu je zovšeobecnením vzťahu (5.21), obsahuje príspevky od každej lokálnej neurónovej siete, pričom tieto sú vážené koeficientmi proporcionality p_i definovanými pomocou výstupných aktivít bránovej siete (pozri podkapitolu 5.2.2). Adaptácia kombinácie lokálnych neurónových sietí spočíva v hľadaní takých prahových a váhových koeficientov lokálnych sietí a bránovej siete, ktoré minimalizujú účelovú funkciu (5.47). Adaptačný proces pre lokálne siete je úplne analogický s adaptačným procesom obyčajnej neurónovej siete popísaným v podkapitole 5.3.3. Výrazy pre výpočet gradientu sú platné aj pre lokálnu sieť s malou modifikáciou, že výrazy g_i v (5.40) sú rozšírené o koeficienty proporcionality. Pre každú lokálnu neurónovú sieť spočítame zvlášť gradient účelovej funkcie a jej prahové a váhové koeficienty sú obnovené pomocou formuly (5.43).

Adaptačný proces kombinácie lokálnych sietí vzhľadom k prahovým a váhovým koeficientom bránovej siete je uskutočniteľný pomocou malej modifikácie adaptačného procesu lokálnych sietí. Parciálne derivácie účelovej funkcie vzhľadom k váhovým koeficientom bránovej siete sú určené vzťahom (pozri (5.37))

$$\frac{\partial E}{\partial w_{ij}^{(g)}} = \frac{\partial E}{\partial \vartheta_i^{(g)}} x_j^{(g)} \tag{5.48a}$$

Parciálne derivácie $\partial E / \partial \vartheta_i^{(g)}$ sú rekurentne určené vzťahom, ktorý má rovnakú formálnu štruktúru ako (5.40)

$$\frac{\partial E}{\partial \vartheta_i^{(g)}} = t'\left(\xi_i^{(g)}\right)\left(g_i^{(g)} + \sum_k \frac{\partial E}{\partial \vartheta_k^{(g)}} w_{ki}^{(g)}\right) \tag{5.48b}$$

kde veličiny $g_k^{(g)}$ určené vzťahom

$$g_k^{(g)} = \begin{cases} \left(\frac{1}{2}\left(\boldsymbol{x}_O^{(k)} - \hat{\boldsymbol{x}}_O\right)^2 - E\right)\left(\sum_j x_j^{(g)}\right)^{-1} & \left(\text{pre } k \in \mathsf{V}_O\right) \\ 0 & \left(\text{pre } k \notin \mathsf{V}_O\right) \end{cases} \tag{5.48c}$$

Jednoduchá diskusia vzťahov (5.48a-c) vedie k nasledujúcim dôležitým záverom: Ak lokálne siete a bránová sieť sú súčasne adaptované pre daný pár $\boldsymbol{x}_I / \hat{\boldsymbol{x}}_O$, potom kombinácia lokálnych sietí smeruje k použitiu len jednej lokálnej siete ku klasifikácii objektu popísaného vstupnými aktivitami $\boldsymbol{x}_I$, pričom výstupná aktivita danej lokálnej siete je blízka požadovanej klasifikácii určenej výstupným vektorom $\hat{\boldsymbol{x}}_O$. To znamená, že koeficienty proporcionality v priebehu adaptačného procesu sa upravia na "binárnu" hodnotu

$$p_j = \begin{cases} 1 & \left(\text{pre } j = i\right) \\ 0 & \left(\text{pre } j \neq i\right) \end{cases} \tag{5.49}$$

pre j=1,2,...,t, kde i je index lokálnej siete poskytujúcej výstupný vektor blízky požadovanému $\hat{\boldsymbol{x}}_O$. Bránová sieť teda rozhoduje tak, že si vyberie jednu lokálnu sieť, ktorá bude použitá pre klasifikáciu daného objektu. Ostatné lokálne siete v dôsledku malosti ich koeficientu proporcionality sa zúčastňujú na tejto klasifikácii zanedbateľnou mierou.

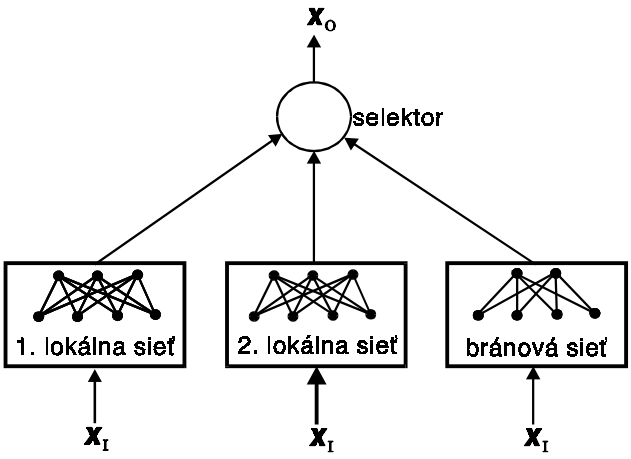
Účelová funkcia (5.47) pre aktuálny adaptačný proces je zovšeobecnená tak, že je sumovaná cez všetky objekty tréningovej množiny (pozri (5.23)). Gradient tejto účelovej funkcie sa potom rovná sume gradientov spočítaných pre jednotlivé objekty (pozri (5.41)).

Ilustračný príklad (sumácia dvoch 2-bitových čísel)

Efektívnosť teórie kombinácie lokálnych neurónových sietí ilustrujeme pomocou klasifikácie sumácie dvoch 2-bitových čísel; tento príklad už bol použitý v predchádzajúcej podkapitole 5.3.4. Ako už bolo zdôraznené, tento príklad obsahujúci 16 objektov je správne interpretovaný neurónovou sieťou, ktorá obsahuje aspoň dva skryté neuróny. Ako ilustračný príklad použijeme kombináciu dvoch lokálnych perceptrónov (neurónových sietí bez skrytých neurónov), ktoré obsahujú štyri vstupné neuróny a tri výstupné neuróny, a bránová sieť tiež predstavuje jednoduchý perceptrón obsahujúci štyri vstupné neuróny a dva (rovnaký počet ako lokálnych sietí) výstupné neuróny (pozri obr. 5.16).

Tréningová množina obsahuje všetkých 16 objektov uvedených v tab. 5.3. Adaptačný proces bol uskutočnený pomocou formúl (5.43) aplikovaných na prahové a váhové koeficienty jednotlivých lokálnych sietí a prahovej siete s parametrami λ=0,1 a μ=0,7. Adaptačný proces bol ukončený po 500 cykloch s hodnotou účelovej funkcie E=0,005. Všetkých 16 objektov bolo korektne klasifikovaných buď prvou alebo druhou lokálnou sieťou. To znamená, že bránová sieť rozdelila tréningovú množinu na dve podmnožiny,

ktoré už sú lineárne separovateľné, čiže korektne interpretovateľné lokálnymi sieťami — perceptrónmi. Dva ilustračné výsledky sú uvedené v tab. 5.5a-b. Tak napríklad, objekt č. 3 (odpovedajúci súčtu 0+2=2) je korektne klasifikovaný prvou lokálnou sieťou (s koeficientom proporcionality p_1=1). Druhá lokálna sieť poskytuje nesprávnu klasifikáciu, ale jej koeficient proporcionality je nulový, p_2=0.



Obrázok 5.16. Schematické znázornenie adaptívnej kombinácie dvoch lokálnych neurónových sietí a jednej bránovej neurónovej siete. Lokálne a bránové siete sú realizované ako dvojvrstvové neurónové siete bez skrytých neurónov (perceptróny).

Tabuľka 5.5a. Výstupné aktivity jednotlivých lokálnych sietí pre interpretáciu objektu č. 3 (pozri tab.5.3)

Objekt č. 3 (0+2=2)				
vstupné aktivity	0	0	1	0
požadované aktivity		0	1	0
vypočítané aktivity				
1. lokálna sieť		0,02	0,99	0,02
2. lokálna sieť		0,91	0,00	0,83
koef. proporcionality	1,00	0,00		

Tabuľka 5.5b. Výstupné aktivity jednotlivých lokálnych sietí pre interpretáciu objektu č. 14 (pozri tab. 5.3)

Objekt č. 14 (3+1=4)				
vstupné aktivity	1	1	0	1
požadované aktivity		1	0	0
vypočítané aktivity				
1. lokálna sieť		0,99	0,01	0,95
2. lokálna sieť		0,99	0,01	0,01
koef. proporcionality	0,00	1,00		

5.4 Neurónová sieť ako univerzálny aproximátor

V počiatkoch histórie neurónových sietí s dopredným šírením [2] bolo venované veľké úsilie tomu, aby sa ukázalo, že tieto neurónové siete s dostatočným počtom skrytých neurónov sú vždy schopné simulovať (aproximovať) zložité binárne alebo spojité funkcie s požadovanou presnosťou. Z pohľadu súčasnosti tieto snahy sú ľahko vysvetliteľné, jednalo sa o prekonanie šoku vyvolaného názorom Minského a Paperta [1], že perceptróny nemajú univerzálny výpočtový charakter. V predchádzajúcej časti tejto kapitoly sme ukázali na rôznych ilustračných príkladoch, že zovšeobecnenie perceptrónu zavedením skrytých neurónov alebo "interakcií" vyššieho rádu medzi neurónmi poskytuje dostatočne flexibilný výpočtový aparát, ktorý je schopný korektne simulovať rôzne zložité binárne funkcie. Hecht-Nielsen v r. 1987 prvý ukázal [13], že trojvrstvové neurónové siete s dopredným šírením a s dostatočným počtom skrytých neurónov sú schopné aproximovať s požadovanou presnosťou každé spojité zobrazenie. V súčasnosti k tomuto problému existuje už pomerne rozsiahla literatúra. Žiaľ, nejedná sa o jednoducho formulovateľný problém. Používajú sa pomerne zložité prostriedky funkcionálnej analýzy a preto sa obmedzíme len na formuláciu základných myšlienok tejto teórie [14].

Študujme spojitú funkciu F, ktorá zobrazuje n-rozmerný priestor R^n na otvorený interval $(0,1)$

$$F: R^n \rightarrow (0,1) \tag{5.50}$$

kde $y=F(\boldsymbol{x})=f(x_1,x_2,...,x_n)$. Tréningová množina A_{train} obsahuje r bodov (aktivít) z n-rozmerného priestoru R^n, $A_{train}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_r\}$. Nech funkcia

$$t: R \rightarrow (0,1) \tag{5.51}$$

je tzv. *prechodová funkcia*, ktorá v tejto súvislosti je len veľmi všeobecne špecifikovaná ako spojitá a monotónne rastúca, vyhovujúca asymptotickým podmienkam $t(-\infty)=0$ a $t(\infty)=1$. Jednoduchá realizácia týchto všeobecných podmienok sa dá dosiahnuť použitím sigmoidy $t(x)=1/(1+e^{-x})$ (porovnaj s (5.11a) pre $A=0$ a $B=1$). Pretože sme predpokladali, že prechodová funkcia je monotónne rastúca, musí k nej existovať inverzná funkcia $t^{-1}:(0,1)\rightarrow R$. Pre sigmoidu je táto inverzná funkcia určená vzťahom $x=\ln(y/(1-y))$. Podľa Hornika [14] platí nasledujúca veta.

Veta. Pre každé $\varepsilon > 0$ existuje taká funkcia

$$G(\boldsymbol{x}) = \sum_{i=1}^{q} \alpha_i t(\vartheta_i + \boldsymbol{w}_i \cdot \boldsymbol{x}) \tag{5.52a}$$

kde α_i a ϑ_i sú reálne koeficienty, $\boldsymbol{w}_i = \left(w_1^{(i)}, w_2^{(i)}, ..., w_n^{(i)}\right)$ sú vektory obsahujúce n reálnych komponent a $\boldsymbol{x} \cdot \boldsymbol{w}_i = x_1 w_1^{(i)} + x_2 w_2^{(i)} + ... + x_n w_n^{(n)}$ je skalárny súčin vektorov $\boldsymbol{x}$ a $\boldsymbol{w}_i$, že

$$\sum_{k=1}^{r} \left| F(\boldsymbol{x}_k) - G(\boldsymbol{x}_k) \right| < \varepsilon \tag{5.52b}$$

Na základe tejto vety môžeme hovoriť, že funkcia $F(\boldsymbol{x})$ je aproximovaná s presnosťou ε nad tréningovou množinou A_{train} pomocou funkcie $G(\boldsymbol{x})$ určenej (5.52a) pomocou všeobecnej prechodovej funkcie $t(x)$ (realizovanej napr. sigmoidou). Žiaľ, táto veta je len existenčného charakteru, nešpecifikuje parametre funkcie G (napr. koeficienty α_i a ϑ_i a vektory $\boldsymbol{w}_i$), tvrdí len, že táto funkcia existuje a aproximuje funkciu $F(\boldsymbol{x})$ nad tréningovou množinou A_{train} .

Funkcia $G(\boldsymbol{x})$ je jednoducho interpretovateľná neurónovou sieťou s dopredným šírením, ktorá obsahuje jednu vrstvu q skrytých neurónov (pozri obr. 5.17). Koeficienty α_i sú váhy spojov medzi skrytými neurónmi a výstupným neurónom, ϑ_i sú prahové koeficienty skrytých neurónov a vektor $\boldsymbol{w}_i$ obsahuje zložky, ktoré tvoria váhové koeficienty hrán medzi i-tym skrytým neurónom a vstupnými neurónmi. Aktivita výstupného neurónu je v tomto prípade rovná potenciálu výstupného neurónu, zatiaľ čo v štandardnej neurónovej sieti aktivita výstupného neurónu je funkčná hodnota prechodovej funkcie pre jeho potenciál (pozri (5.10a)). Táto reštrikcia je ľahko odstrániteľná použitím predpokladu, že k prechodovej funkcii $y=t(x)$ existuje spojitá inverzná funkcia $x=t^{-1}(y)$. Potom podmienka (5.52b) má tvar

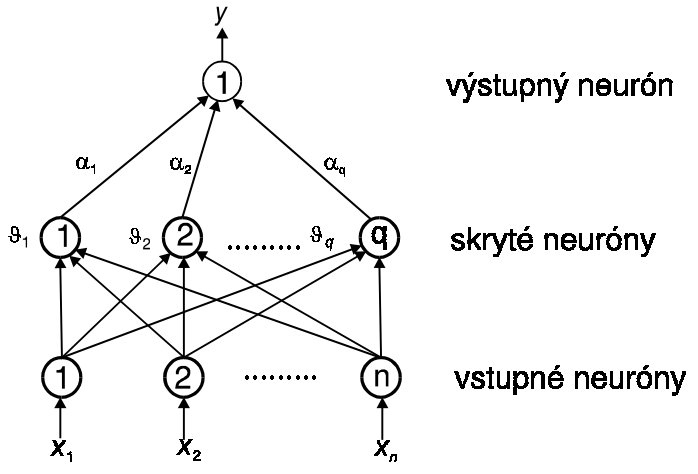
$$\sum_{k=1}^{r} \left| F(\boldsymbol{x}_k) - \tilde{G}(\boldsymbol{x}_k) \right| < \varepsilon' \tag{5.53}$$

kde nová funkcia $\tilde{G}(\boldsymbol{x})$ má tvar

$$\tilde{G}(\boldsymbol{x}) = t\left(\sum_{i=1}^{q} \alpha_i t(\vartheta_i + \boldsymbol{w}_i \cdot \boldsymbol{x})\right) \tag{5.54}$$

Touto jednoduchou modifikáciou vyššie uvedenej vety sme ukázali, že ľubovoľná spojitá funkcia $F(\boldsymbol{x})$, definovaná nad tréningovou množinou A_{train} a s funkčnými hodnotami z otvoreného intervalu (0,1), je aproximovateľná funkciou $\tilde{G}(\boldsymbol{x})$ s požadovanou presnosťou ε'. Čo je dôležité, funkcia $\tilde{G}(\boldsymbol{x})$ je už interpretovateľná neurónovou sieťou s dopredným šírením, ktorá obsahuje jednu vrstvu q skrytých neurónov.

Uvedená veta má principiálnu dôležitosť pre neurónové siete. Zabezpečuje nám, že trojvrstvová neurónová sieť (obsahujúca jednu vrstvu skrytých neurónov) je schopná simulovať s požadovanou presnosťou ľubovolné zobrazenie typu (5.50) definované nad konečnou tréningovou množinou. Týmto máme teda k dispozícii všeobecný prostriedok pre regresnú analýzu funkcií definovaných pomocou "regresnej tabuľky", kde pre nezávislé argumenty sú predpísané funkčné hodnoty (t.j. tréningová množina v zmysle úvodnej podkapitoly 5.1, pozri (5.3)). Teória neurónových sietí poskytuje univerzálny prostriedok pre návrh "modelovej funkcie" v tvare (5.54), kde počet skrytých neurónov a prahové a váhové koeficienty sú regresné parametre. Avšak musíme poznamenať, že hlavný cieľ teórie neurónových sietí s dopredným šírením nie je regresná analýza funkcií definovaných tréningovou množinou (aj keď tento moment v mnohých prípadoch je veľmi dôležitý), ale extrapolácia funkčných hodnôt mimo tréningovej množiny, čiže problém zovšeobecnenia (predikcia a klasifikácia).



Obrázok 5.17. Znázornenie štandardnej neurónovej siete s dopredným šírením, ktorá obsahuje jednu vrstvu skrytých neurónov. Aktivita výstupného neurónu y "simuluje" zobrazenie $F(x)$ pre požadované body z tréningovej množiny.

5.5 Praktické skúsenosti s aplikáciami neurónových sietí na klasifikáciu a predikciu

Viacvrstvové neurónové siete s dopredným šírením patria medzi tie neurónové siete, ktoré sú najčastejšie používané ako univerzálny prostriedok pre klasifikáciu a predikciu. Uvedieme niekoľko praktických skúseností, ako realizovať tieto aplikácie. Najprv budeme študovať problém, ako rozložiť množinu klasifikovaných objektov A na tréningovú a testovaciu množinu, $A=A_{train}\cup A_{test}$. Realizácia tohto rozkladu patrí medzi prvé základné problémy pri aplikáciách neurónových sietí, tréningová množina by mala obsahovať tie objekty z A, ktoré dobre "reprezentujú" ostatné podobné objekty (zahrnuté v testovacej množine A_{test}). Problém rovnakej dôležitosti ako rozklad množiny objektov na tréningovú a testovaciu množinu je aj problém výberu deskriptorov, ktoré sú podstatné pre klasifikáciu objektov. Obvykle sa deskriptory objektov navrhujú "ad-hoc" spôsobom — vyberajú sa také deskriptory, ktoré sú (alebo môžu byť) dôležité pre popis objektov. Z tohto pohľadu vystupuje do popredia problém výberu len tých deskriptorov, ktoré poskytujú rovnakú (alebo o málo horšiu) klasifikáciu objektov ako pôvodná sada deskriptorov. Tento problém budeme riešiť pomocou jednoduchej metódy využívajúcej najbližších susedov v okolí klasifikovaného objektu.

Ďalším dôležitým problémom pri aplikáciách neurónových sietí je navrhnúť vhodnú architektúru neurónovej siete. V našich úvahách sa pre jednoduchosť obmedzíme len na neurónové siete s jednou vrstvou skrytých neurónov (pozri obr. 5.17), pričom pod architektúrou budeme rozumieť počet skrytých neurónov. Na základe vety (uvedenej v podkapitole 5.4), ktorá charakterizuje 3-vrstvovú neurónovú sieť ako univerzálny aproximátor, môžeme očakávať, že s rastom počtu skrytých neurónov bude adaptačný proces neurónovej siete poskytovať lepšie a lepšie výsledky (t.j. hodnota účelovej funkcie (5.23) sa bude asymptoticky blížiť k nule). Tento záver je správny, avšak ak budeme porovnávať predikčné (alebo klasifikačné) schopnosti týchto neurónových sietí, spozorujeme, že od určitého počtu skrytých neurónov sa predikcia neurónovej siete pre objekty z testovacej množiny začne zhoršovať. To znamená, že z hľadiska správnej klasifikácie objektov z testovacej množiny je ďalšie zvyšovanie počtu skrytých neurónov už zbytočné (alebo až nežiaduce, z pohľadu adaptačného procesu neurónovej siete).

Podobný problém je aj s počtom iteračných krokov pri adaptácii neurónových sietí (pre daný počet skrytých neurónov). Ak súčasne sledujeme znižovanie účelovej funkcie (5.23) v priebehu adaptácie, od určitého počtu iteračných krokov spozorujeme, že predikčná schopnosť neurónovej siete sa začne zhoršovať. Podobne ako v predchádzajúcom prípade (zvyšovanie počtu skrytých neurónov), ďalšia adaptácia neurónovej siete je zbytočná, už len zhoršuje jej predikčnú schopnosť.

Problém optimálneho počtu adaptačných krokov úzko súvisí tiež s výberom adaptačnej (minimalizačnej) metódy. V podkapitole 5.3.3 bola diskutovaná jednoduchá modifikácia gradientovej metódy najprudšieho spádu (pozri (5.43)). Aj keď tento prístup je najčastejšie používaný pre adaptáciu viacvrstvových neurónových sietí s dopredným šírením, obvykle je kritizovaný ako veľmi pomalý, vyžadujúci mnoho tisíc iteračných krokov. Z týchto dôvodov sa venuje pozornosť aj iným, efektívnejším optimalizačným metódam numerickej matematiky (napr. Newtonova metóda, metóda združených gradientov alebo metóda premennej metriky, pozri [11]). Ich použitím v teórii neurónových sietí sa dosiahne podstatne rýchlejší proces adaptácie, avšak obvykle za cenu zhoršenia predikčných

schopností neurónovej siete. Obrazne povedané, neurónová sieť je "vynikajúco" adaptovaná na objekty tréningovej množiny (váhové a prahové koeficienty neurónovej siete majú hodnoty odpovedajúce presným hodnotám daného lokálneho minima účelovej funkcie (5.23)), avšak za cenu "preučenia" neurónovej siete s následnou slabou predikčnou schopnosťou.

Na záver tejto podkapitoly uvedieme jednoduchú algoritmizáciu v pascalovskom pseudokóde neurónovej siete s dopredným šírením, ktorá obsahuje jednu vrstvu skrytých neurónov.

5.5.1 Rozklad množiny objektov na tréningovú a testovaciu množinu

Popíšeme jednoduchý spôsob rozkladu množiny objektov A na tréningovú a testovaciu množinu, $A=A_{train}\cup A_{test}$. Predpokladajme, že poznáme nejakú klastrovaciu metódu [15], ktorá nám rozloží množinu A na disjunktné podmnožiny — klastre, ktoré obsahujú "podobné" objekty (z hľadiska metriky použitej v klastrovacej metóde)

$$A = C_1 \cup C_2 \cup ... \cup C_p \tag{5.55}$$

kde i-ty klaster C_i obsahuje n_i objektov z A

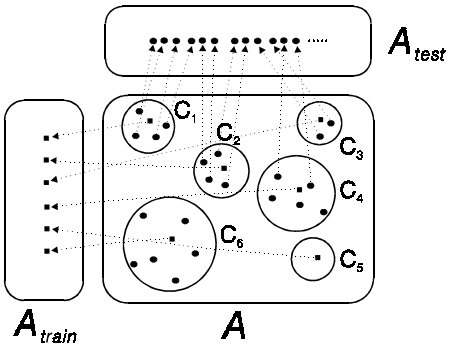
$$C_i = \left\{ o_1^{(i)}, o_2^{(i)}, ..., o_{n_i}^{(i)} \right\} \subset A \tag{5.56}$$

pričom predpokladáme, že objekt $o_1^{(i)} \in C_i$ je ten objekt z i-teho klastra C_i , ktorý leží "najbližšie" k jeho centru. Tento objekt nám v nasledujúcich úvahách bude slúžiť ako "reprezentant" objektov z klastra C_i. Potom tréningová a testovacia množina je určená objektmi

$$\begin{aligned} A_{train} &= \left\{ o_1^{(1)}, o_1^{(2)}, ..., o_1^{(p)} \right\} \\ A_{test} &= \left(C_1 - \left\{ o_1^{(1)} \right\} \right) \cup \left(C_2 - \left\{ o_1^{(2)} \right\} \right) \cup ... \cup \left(C_p - \left\{ o_1^{(p)} \right\} \right) \end{aligned} \tag{5.57}$$

To znamená, že tréningová množina je zložená zo všetkých reprezentantov klastrov a testovacia množina obsahuje zostávajúce objekty (pozri obr. 5.18)). Počet objektov v tréningovej množine je totožný s počtom klastrov, $|A_{train}|=p$ a $|A_{test}|= |A|-p$.

Teória neurónových sietí poskytuje výborný klastrovací prostriedok pomocou *Kohonenovej neurónovej siete* (pozri kapitolu 7), ktorý je ľahko použiteľný aj pre diskutovanú problematiku rozkladu množiny objektov na tréningovú a testovaciu množinu [16]. Obvykle sú výstupné neuróny tejto siete priestorovo uložené na ortogonálnej mriežke typu $N \times N$ (t.j. sieť obsahuje N^2 výstupných neurónov). Adaptačný proces tejto siete spočíva v tom, že objekty množiny A aktivujú len jeden výstupný neurón. Objekty, ktoré aktivujú rovnaký výstupný neurón, môžeme považovať za "podobné". Z teórie Kohonenových neurónových sietí tiež vyplýva, že tieto objekty môžeme tiež ešte podrobnejšie klasifikovať z hľadiska ich "blízkosti" k určitému centru daného výstupného neurónu (napr. minimálnosťou normy rozdielu deskriptorov objektu a váhových koeficientov výstupného neurónu). Tieto "centrálne" objekty nám slúžia ako reprezentanti objektov, ktoré aktivujú dané výstupné neuróny (klastre) a teda tvoria tréningovú množinu. V tejto súvislosti je potrebné bližšie špecifikovať tvar deskriptorov objektov, ktoré sa použijú pre "klastrovanie" množiny A na tréningovú a testovaciu množinu. Podobnosť objektov je v tomto prípade určená nielen ich deskriptormi ale aj ich vlastnosťami. Z týchto dôvodov, pre potreby klastrovania objektov pomocou Kohonenovej siete deskriptory objektov sú rozšírené ešte o vlastnosť (alebo vlastnosti) objektov. Jednoduchá realizácia tohto prístupu je znázornená na obr. 5.19.



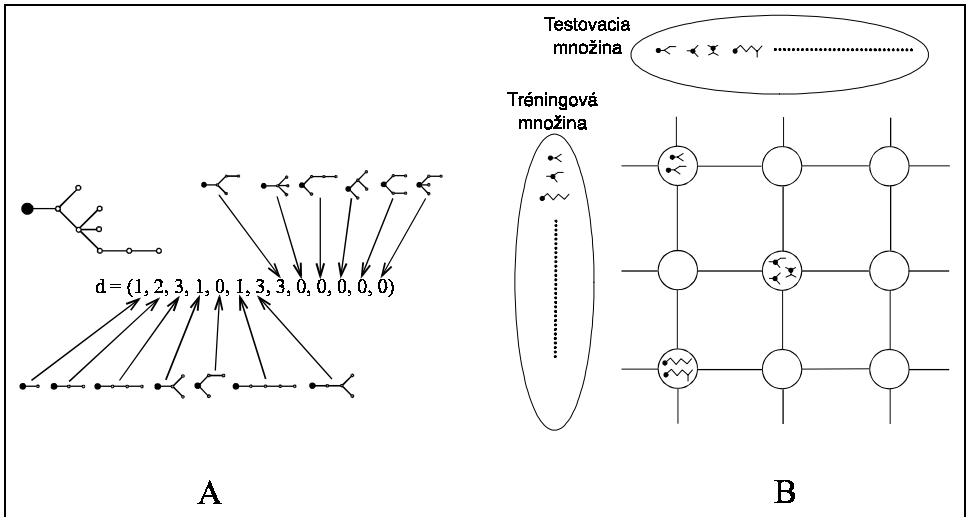
Obrázok 5.18. Schematické znázornenie rozkladu množiny objektov A na tréningovú a testovaciu množinu pomocou rozkladu A na klastre C_1, C_2, ..., C_p . Objekty reprezentované zaplnenými štvorcami odpovedajú tým objektom, ktoré ležia najbližšie k stredom príslušných klastrov, ostatné objekty sú reprezentované zaplnenými krúžkami. Poznamenajme, že klaster C_5 obsahuje len jeden objekt, ktorý patrí do tréningovej množiny.

5.5.2 Optimálny výber deskriptorov

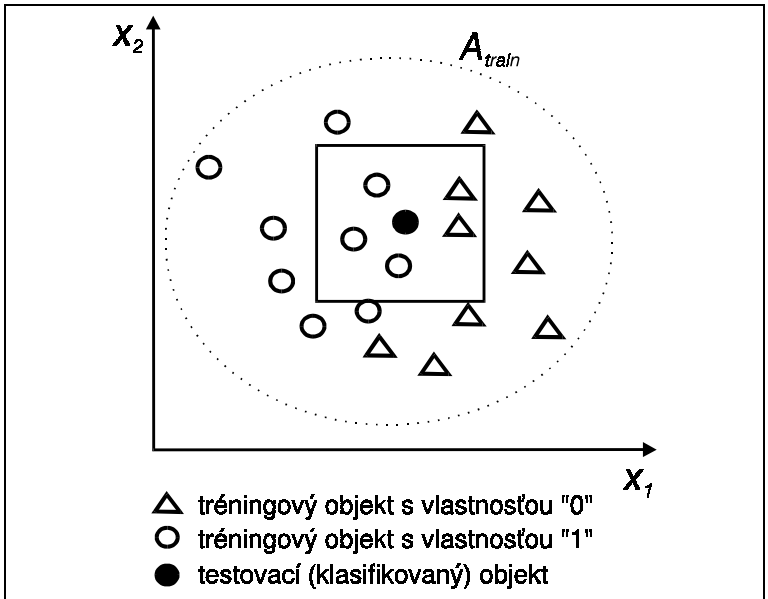
Výber deskriptorov — príznakov, ktoré popisujú vhodným spôsobom objekty, patrí medzi základné úlohy predspracovania dát pre potreby neurónových sietí. V mnohých prípadoch sú deskriptory navrhnuté "ad-hoc", bez podrobnejšieho štúdia ich vzájomnej závislosti a významnosti pre popis objektov. Z týchto dôvodov je dôležité analyzovať použité

deskriptory z pohľadu ich významnosti pre klasifikáciu objektov, z množiny navrhnutých deskriptorov určiť tie, ktoré sú významné pre klasifikáciu testovacích objektov. Pre aplikácie neurónových sietí ako klasifikátorov a prediktorov je určenie optimálneho výberu deskriptorov významné nielen z pohľadu efektívnosti adaptačného procesu, ale tiež aj ako významný medzikrok pre uľahčenie interpretácie výsledkov poskytovaných adaptovanou neurónovou sieťou.

V tejto podkapitole popíšeme klasifikačnú metódu KNN (angl. *K Nearest Neighbor*) [17], ktorej jednoduchá modifikácia je vhodná pre optimálny výber deskriptorov. Základný princíp metódy KNN je znázornený na obr. 5.20. V tejto metóde hrá základnú úlohu vzdialenosť medzi tréningovým objektom a testovacím objektom, táto vzdialenosť môže byť definovaná ako L_1 metrika



Obrázok 5.19. Ilustračný príklad použitia Kohonenovej siete na rozklad množiny objektov *A* na tréningovú a testovaciu množinu prevzatý z publikácie [16], v ktorej sa autori zaoberali použitím neurónových sietí k predikcii ^{13}C NMR (jadrová magnetická rezonancia) chemických posunov nasýtených acyklických uhlovodíkov — alkánov (napr. náplň v zapaľovačoch je tvorená zmesou dvoch alkánov, a to propánu a butánu). Chemický posun je lokálna vlastnosť uhlíkových atómov, preto daný uhlíkový atóm v alkáne je popísaný jeho najbližším okolím (diagram A), t.j. koľkokrát sa daný fragment vyskytuje v alkáne so špecifikovaným uhlíkom. Deskriptor ***d*** obsahuje 13 nezáporných celých čísel, ktoré odpovedajú "frekvenciám výskytu" fragmentov. Pre potreby klastrovania množiny alkánov z hľadiska ^{13}C NMR chemických posunov, tento deskriptor je rozšírený o experimentálne nameraný chemický posun. Takto modifikované deskriptory sú použité pre adaptáciu Kohonenovej neurónovej siete. Po skončení tohto procesu daný deskriptor (alkán) aktivuje práve jeden výstupný neurón. Rôzne alkány, ktoré aktivujú rovnaký výstupný neurón, sú považované za "klaster podobných" alkánov (B).



Obrázok 5.20. Schematické znázornenie metódy KNN. V rovine deskriptorov x_1 a x_2 sú umiestnené objekty tréningovej množiny. Okolo klasifikovaného objektu z testovacej množiny vytvoríme takú sféru (reprezentovanú v dôsledku použitia L_1 metriky štvorcom), aby obsahovala K objektov (v našom prípade K=5). Testovací objekt je klasifikovaný podľa tých tréningových objektov, ktoré sa najviac vyskytujú vo sfére, v tomto prípade má testovací objekt vlastnosť 1.

$$D(\boldsymbol{d}_{train}, \boldsymbol{d}_{test}) = \sum_{i=1}^{n} \left| d_{train}^{(i)} - d_{test}^{(i)} \right| \tag{5.58}$$

kde $\boldsymbol{d}_{train} = \left(d_{train}^{(1)}, d_{train}^{(2)}, \ldots, d_{train}^{(n)}\right)$ a $\boldsymbol{d}_{train} = \left(d_{test}^{(1)}, d_{test}^{(2)}, \ldots, d_{test}^{(n)}\right)$ sú vektory deskriptorov priradené objektu z tréningovej resp. testovacej množiny.
Predpokladajme, že tréningové objekty sú usporiadané tak, že pre testovací objekt s vektorom deskriptorov $\boldsymbol{d}_{train}$ platí

$$D\left(\boldsymbol{d}_{train,1}, \boldsymbol{d}_{test}\right) \le D\left(\boldsymbol{d}_{train,2}, \boldsymbol{d}_{test}\right) \le \ldots \le D\left(\boldsymbol{d}_{train,K}, \boldsymbol{d}_{test}\right) \le D\left(\boldsymbol{d}_{train,K+1}, \boldsymbol{d}_{test}\right) \le \ldots \tag{5.59}$$

K prvých tréningových objektov z tejto postupnosti tvorí okolie (K-rozmernú sféru) testovacieho objektu $\boldsymbol{d}_{test}$. Testovací objekt je klasifikovaný podľa tých objektov z K sféry, ktoré sa v nej najviac vyskytujú. Týmto spôsobom sme schopní klasifikovať každý objekt z testovacej množiny. Formálne, $y_{test} = KNN(\boldsymbol{d}_{test})$, kde y_{test} je vlastnosť priradená testovaciemu objektu s deskriptorom $\boldsymbol{d}_{test}$.

Výraz (5.58) pre vzdialenosť dvoch objektov možno zovšeobecniť tak, že sa zavedú binárne váhy $w_i \in \{0,1\}$, ktoré popisujú, či sa i-ty deskriptor uvažuje (w_i=1) alebo neuvažuje (w_i=0)

$$D(\boldsymbol{d}_{train}, \boldsymbol{d}_{test}) = \sum_{i=1}^{n} w_i \left| d_{train}^{(i)} - d_{test}^{(i)} \right| \tag{5.60}$$

Modifikovaný KNN klasifikátor s binárnymi váhami označíme $KNN_{\boldsymbol{w}}$, alebo $y_{test}^{(\boldsymbol{w})} = KNN_{\boldsymbol{w}}(\boldsymbol{d}_{test})$. Pre binárny váhový vektor $\boldsymbol{w}$ sú výsledky poskytované metódou KNN_w totožné s výsledkami poskytovanými KNN, ak všetky komponenty $\boldsymbol{w}$ sú jednotkové.

Úspešnosť klasifikátora $KNN_{\boldsymbol{w}}$ pri interpretácii objektov z testovacej množiny môže byť popísaná účelovou funkciou

$$f(\boldsymbol{w}) = \frac{1}{|A_{test}|} \sum_{\boldsymbol{d}_{test}} \delta\left(y_{test}^{(req)}, KNN_{\boldsymbol{w}}(\boldsymbol{d}_{test})\right) \tag{5.61}$$

kde $\delta(i,j)$=1 pre $i=j$, $\delta(i,j)$=0 pre $i \neq j$ a $y_{test}^{(req)}$ vyjadruje požadovanú vlastnosť. V prípade, že klasifikátor KNN_w interpretuje správne všetky objekty testovacej množiny, hodnota účelovej funkcie $f(\boldsymbol{w})$ je maximálna (jednotková), jej menšie hodnoty ($0 \leq f(\boldsymbol{w}) < 1$) indikujú, že klasifikátor KNN_w poskytuje nesprávnu interpretáciu. Výraz $1 - f(\boldsymbol{w})$ určuje frakciu objektov testovacej množiny, ktoré sú nesprávne interpretované. Optimálny výber deskriptorov je určený riešením diskrétneho optimalizačného problému

$$\boldsymbol{w}_{opt} = \arg \max_{\boldsymbol{w} \in \{0,1\}^n} f(\boldsymbol{w}) \tag{5.62}$$

kde hľadáme globálne minimum v priestore všetkých binárnych vektorov dĺžky n. Pre malé hodnoty n je problém (5.62) riešiteľný systematickým prehľadávaním celého priestoru riešení (dimenzia priestoru riešení je 2^n), napríklad metódou spätného prehľadávania [18], ktorá môže byť podstatne urýchlená metódou vetiev a hrán (angl. *branch and bound*). Pre väčšie hodnoty n (n>15) už nemožno riešiť optimalizačný problém (5.62) systematickými prehľadávacími algoritmami v dôsledku exponenciálneho rastu CPU času potrebného na riešenie problému. Z týchto dôvodov musíme obrátiť našu pozornosť na také metódy riešenia problému (5.62) ktoré, aj keď sú približné, poskytujú obvykle suboptimálne riešenia blízke optimálnym. V súčasnej informatike sú veľmi populárne tzv. evolučné algoritmy, založené na heuristikách prevzatých z biológie alebo z fyziky, ktoré poskytujú pomerne rýchlo suboptimálne riešenie zložitých optimalizačných problémov spojitého alebo diskrétneho charakteru (pozri kapitolu 9).

5.5.3 Architektúra neurónovej siete a počet adaptačných krokov

Návrh vhodnej architektúry (t.j. topológie grafu určujúceho neurónovú sieť) je zložitý a hlavne numericky náročný problém. Preto sa obmedzíme len na neurónové siete s jednou vrstvou skrytých neurónov (pozri obr. 5.17). Hlavným kritériom pre optimálny návrh

neurónovej siete bude optimálnosť jej klasifikačnej schopnosti, a realizácia tohto návrhu sa bude vykonávať súbežne s určením optimálneho počtu adaptačných krokov. Definujme si dve nasledujúce účelové funkcie (pozri (5.23))

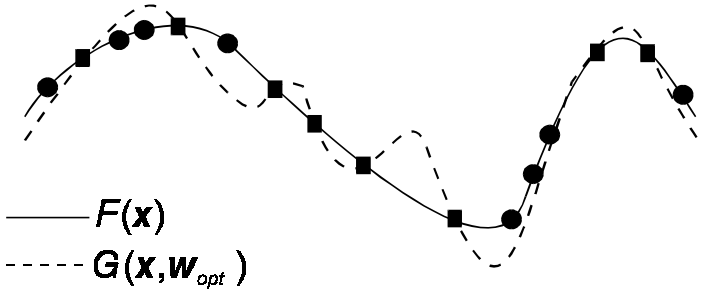
$$E_{train} = \frac{1}{2}\sum_{i}^{A_{train}} \left(G(\boldsymbol{x}_i, \boldsymbol{w}) - \hat{\boldsymbol{x}}_i\right)^2$$
$$E_{test} = \frac{1}{2}\sum_{i}^{A_{test}} \left(G(\boldsymbol{x}_i, \boldsymbol{w}) - \hat{\boldsymbol{x}}_i\right)^2 \tag{5.63}$$

kde E_{train} (E_{test}) je účelová funkcia definovaná pre objekty z tréningovej (testovacej) množiny pre dané hodnoty váhových a prahových koeficientov $\boldsymbol{w}$. Na základe vety o neurónovej sieti ako univerzálnom aproximátore (pozri podkapitolu 5.4) vieme, že pre rastúci počet skrytých neurónov účelová funkcia E_{train} (s adaptovanými váhovými a prahovými koeficientmi) konverguje k nule

$$\lim_{q\to\infty} E_{train} = 0 \tag{5.64}$$

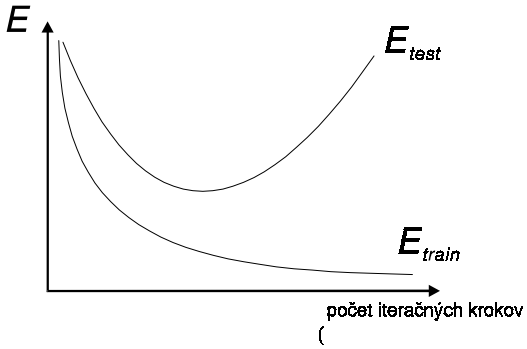
kde q je počet skrytých neurónov. Ako môžeme interpretovať tento dôležitý výsledok teórie neurónových sietí s dopredným šírením a s jednou vrstvou skrytých neurónov? Interpretácia je veľmi podobná analogickej situácii v regresnej analýze s polynomiálnou modelovou funkciou. Zvyšovaním rádu polynómu dostávame stále menšiu a menšiu hodnotu minimalizovanej účelovej funkcie. Flexibilita polynómu rastie s jeho stupňom (pozri obr. 5.21).

Podobný obrázok by sme dostali aj pri štúdiu závislosti schopnosti korektne klasifikovať objekty z testovacej množiny od počtu iteračných krokov pre neurónovú sieť s daným počtom skrytých neurónov. Hodnota účelovej funkcie E_{train} bude klesať s rastom počtu iteračných krokov. Žiaľ, hodnota účelovej funkcie E_{test} bude od určitého počtu iteračných krokov rásť, t.j. zhoršuje sa predikčná schopnosť neurónovej siete s pokračovaním adaptácie neurónovej siete (hovoríme, že neurónová sieť je preučená, pozri obr. 5.22).

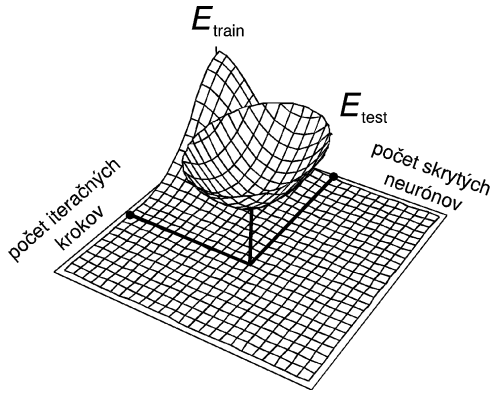


Obrázok 5.21. Schematické znázornenie simulácie "presnej" funkcie *F*, ktorá interpretuje objekty tak z tréningovej (plné štvorce), ako aj z testovacej množiny (plné krúžky) (pozri podkapitolu 5.1). Funkcia *G* reprezentuje adaptovanú neurónovú sieť s dostatočným počtom skrytých neurónov, ktorej adaptácia bola vykonaná vzhľadom k objektom z tréningovej množiny. Funkcia *G* nemusí už presne postihovať objekty z testovacej množiny. Počet "inflexných bodov" funkcie *G* rastie s počtom skrytých neurónov (analógia s polynomiálnou regresiou). Môžeme teda očakávať, že interpretácia objektov z testovacej množiny sa bude zhoršovať s rastom počtu skrytých neurónov.

Z vyššie uvedených úvah vyplýva, že stanovenie optimálneho počtu skrytých neurónov a počtu iteračných krokov vzhľadom pre dané rozdelenie objektov na tréningovú a testovaciu množinu môže byť realizované súčasne. Pre daný počet skrytých neurónov nájdeme optimálny počet iteračných krokov adaptačného procesu. Tento prístup je založený na poznatku, že zatiaľ čo tréningová účelová funkcia E_{train} klesá s rastúcim počtom skrytých neurónov a/alebo rastúcim počtom iteračných krokov, testovacia účelová funkcia E_{test} vykazuje minimum pre určitý počet skrytých neurónov a počet iteračných krokov (pozri obr. 5.23). Tieto hodnoty, v ktorých má E_{test} minimum, sú optimálne pre použitie 3-vrstvovej neurónovej siete pre klasifikáciu objektov z testovacej množiny A_{test}.



Obrázok 5.22. Priebeh účelových funkcií pre tréningovú a testovaciu množinu v závislosti na počte iteračných krokov adaptácie (resp. na počte skrytých neurónov) neurónovej siete. Účelová funkcia E_{train} klesá s rastom počtu iteračných krokov (resp. skrytých neurónov), zatiaľ čo účelová funkcia E_{test} vykazuje minimum.



Obrázok 5.23. Schematické znázornenie priebehu účelových funkcií E_{train} a E_{test} vzhľadom k počtu skrytých neurónov a počtu iteračných krokov. Na rozdiel od účelovej funkcie E_{train}, ktorá monotónne klesá s rastúcim počtom skrytých neurónov a s rastúcim počtom iteračných krokov, účelová funkcia E_{test} má minimum pre optimálny počet skrytých neurónov a iteračných krokov (pozri hrubé čiary na grafe).

5.5.4 Algoritmizácia neurónovej siete s dopredným šírením

Účelom tejto podkapitoly je naznačiť základné princípy algoritmizácie neurónových sietí s dopredným šírením, ktoré obsahujú skryté neuróny. Pre jednoduchosť budeme uvažovať 3-vrstvovú neurónovú sieť, ktorá obsahuje jednu vrstvu skrytých neurónov, pričom neuróny zo susedných vrstiev sú prepojené všetkými možnými spôsobmi, pozri obr. 5.24 .

Aktivity skrytých a výstupných neurónov sú určené vzťahmi (pozri vzťahy (5.10a-b))

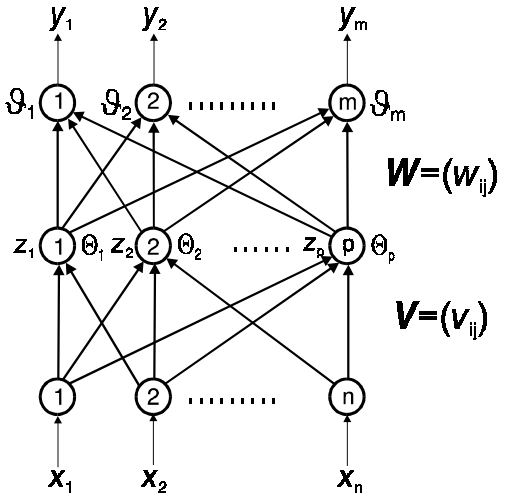
$$z_i = t\left(\sum_{j=1}^{n} v_{ij} x_j + \Theta_i\right) \qquad (\text{pre } i = 1,2,\ldots,p) \tag{5.65a}$$

$$y_i = t\left(\sum_{j=1}^{p} w_{ij} z_j + \vartheta_i\right) \qquad (\text{pre } i = 1,2,\ldots,m) \tag{5.65a}$$

kde $t(\xi)$ je sigmoida určená pomocou (5.11a-b) (s parametrami A=0 a B=1)

$$t(\xi) = \frac{1}{1+e^{-\xi}} \tag{5.66}$$

Grafický priebeh tejto funkcie je znázornený na obr. 5.5, graf 1.



Obrázok 5.24. Trojvrstvová neurónová sieť s dopredným šírením. Vstupná vrstva obsahuje n vstupných neurónov s aktivitami x_1, x_2, ..., x_n. Skrytá vrstva obsahuje p skrytých neurónov, ich aktivity resp. prahové koeficienty sú označené z_1, z_2, ..., z_p resp. Θ_1, Θ_2, ..., Θ_p . Výstupná vrstva obsahuje m výstupných neurónov, ich aktivity resp. prahové koeficienty sú označené y_1, y_2, ..., y_m resp. ϑ_1, ϑ_2, ..., ϑ_m. Váhové koeficienty medzi vstupnou a skrytou vrstvou resp. skrytou a výstupnou vrstvou tvoria maticu $\boldsymbol{V}=(v_{ij})$ resp. maticu $\boldsymbol{W}=(w_{ij})$.

Výpočet aktivít neurónov pre dané váhové a prahové koeficienty sa nazýva aktívna fáza neurónovej siete. Tieto aktivity pre danú neurónovú sieť sa vypočítajú jednoduchým rekurentným postupom: Predpokladajme, že vstupné aktivity x_1, x_2, ..., x_n (deskriptory klasifikovaného objektu) sú známe, potom pomocou (5.65a) zostrojíme aktivity skrytých neurónov z_1, z_2, ..., z_p. Následne, pomocou (5.65b) zostrojíme aktivity výstupných neurónov y_1, y_2, ..., y_m. Uvedený rekurentný spôsob výpočtu aktivít postupuje zdola nahor neurónovou sieťou. Táto skutočnosť sa odráža v názve týchto sietí, ako neurónových sietí s dopredným šírením signálu. Algoritmizácia tohto postupu je uvedená formou pascalovského pseudokódu na obr. 5.25.

```
procedure activities(input :Θ,v,ϑ,w,x;
                         output:z,y);
begin for i:=1 to p do
      begin ξ:=Θ[i];
            for j:=1 to n do ξ:=ξ+v[i,j]*x[j];
            z[i]:=t(ξ);
      end;
      for i:=1 to m do
      begin ξ:=ϑ[i];
            for j:=1 to p do ξ:=ξ+w[i,j]*z[j];
            y[i]:=t(ξ);
      end;
end;
```

Obrázok 5.25. Algoritmizácia v pascalovskom pseudokóde aktívnej fáze neurónovej siete s dopredným šírením, ktorá obsahuje jednu vrstvu skrytých neurónov. Vstupnými parametrami procedúry activities sú vstupné aktivity a váhové a prahové koeficienty, výstupnými parametrami sú skryté a výstupné aktivity. Reálna funkcia t(ξ) je prechodová funkcia definovaná (5.66).

Teraz upriamime našu pozornosť na tzv. adaptačnú fázu neurónovej siete, v ktorej sú upravované (pozri (5.43)) váhové a prahové koeficienty pomocou gradientu účelovej funkcie *E* definovanej (5.41). V prvom kroku budeme študovať konštrukciu gradientu účelovej funkcie (5.21), ktorá je priradená len jednému objektu z tréningovej množiny. Komponenty gradientu sú určené formulami (5.37) a (5.40), ktoré sa jednoducho prepíšu zvlášť pre výstupné neuróny

$$\frac{\partial E}{\partial \vartheta_i} = y_i(1-y_i)\left(y_i - y_{i,req}\right) \qquad \left(\text{pre } i = 1,2,\ldots,m\right) \tag{5.67a}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \vartheta_i} z_j \qquad \begin{pmatrix} \text{pre } i = 1,2,\ldots,m \\ j = 1,2,\ldots p \end{pmatrix} \tag{5.67b}$$

a zvlášť pre skryté neuróny

$$\frac{\partial E}{\partial \Theta_i} = z_i(1-z_i)\sum_{j=1}^{m}\frac{\partial E}{\partial \vartheta_j} w_{ji} \qquad \left(\text{pre } i = 1,2,\ldots,p\right) \tag{5.68a}$$

$$\frac{\partial E}{\partial v_{ij}} = \frac{\partial E}{\partial \Theta_i} x_j \qquad \begin{pmatrix} \text{pre } i = 1,2,\ldots,p \\ j = 1,2,\ldots n \end{pmatrix} \tag{5.68b}$$

V týchto formulách derivácia prechodovej funkcie $t'(\xi)$ je určená jednoduchým vzťahom $t'(\xi) = t(\xi)(1-t(\xi))$ (pozri vzťah (5.11b)). Formuly (5.67-68) pre výpočet parciálnych derivácií možno realizovať rekurentne tak, že sa postupuje cez neurónovú sieť zhora nadol (t.j. v opačnom smere ako pri výpočte aktivít neurónovej siete). V prvom kroku sa vypočítajú parciálne derivácie účelovej funkcie vzhľadom k prahovým koeficientom výstupných neurónov pomocou (5.67a). Potom sa jednoducho vypočítajú aj parciálne derivácie účelovej funkcie vzhľadom k váhovým koeficientom spojov medzi skrytými a výstupnými neurónmi pomocou (5.67b). Poznajúc túto časť gradientu účelovej funkcie, môžeme pristúpiť k výpočtu tej jeho časti, ktorá odpovedá skrytým neurónom. Podobne ako v predchádzajúcom prípade, pomocou (5.68a) vypočítame parciálne derivácie účelovej funkcie vzhľadom k prahovým koeficientom skrytých neurónov, a potom pomocou (5.68b) vypočítame parciálne derivácie účelovej funkcie vzhľadom k váhovým koeficientom medzi vstupnými a skrytými neurónmi. Algoritmizácia tohto postupu (metódy spätného šírenia) je znázornená v pascalovskom pseudokóde na obr. 5.26.

```
procedure gradient(input :Θ,V,ϑ,W,x,y_req;
                   output:grad_ϑ,grad_w,grad_Θ,grad_v);
begin activities(Θ,V,ϑ,W,x,z,y);
      for i:=1 to m do
      grad_ϑ[i]:=y[i]*(1-y[i])*(y[i]-y_req[i]);
      for i:=1 to m do
      for j:=1 to p do
            grad_w[i,j]:=grad_ϑ[i]*z[j];
      for i:=1 to p do
      begin aux:=0;
            for j:=1 to m do
            aux:=aux+grad_ϑ[j]*w[j,i]
            grad_Θ[i]:=z[i]*(1-z[i])*aux;
      end;
      for i:=1 to p do
      for j:=1 to n do
            grad_v[i,j]:=grad_Θ[i]*x[j];
end;
```

Obrázok 5.26. Výpočet gradientu účelovej funkcie pre dané vektory vstupných aktivít $\boldsymbol{x}$ požadovaných výstupných aktivít $\boldsymbol{y}_{req}$. Proces je inicializovaný výpočtom skrytých a výstupných aktivít pomocou procedúry activities. Prahové a váhové koeficienty sú vstupnými parametrami procedúry. Vypočítaný gradient je výstupným parametrom procedúry.

Parciálne derivácie účelovej funkcie (5.21), ktorá je definovaná nad celou tréningovou množinou, sú určené vzťahom (5.41), t.j. celkový gradient sa rovná sume gradientov pre jednotlivé elementy tréningovej množiny. Pascalovský pseudokód výpočtu celkového gradientu účelovej funkcie je znázornený na obr. 5.27.

```
procedure gradient_total(input :Θ,V,ϑ,W,A_train;
                         output:
                          grad_total_ϑ,, grad_total_w,
                          grad_total_Θ,grad_total_v);
begin for i:=1 to m do grad_total_ϑ[i]:=0;
      for i:=1 to m do
      for j:=1 to p do grad_total_w[i,j]:=0;
      for i:=1 to p do grad_total_Θ[i]:=0;
      for i:=1 to p do
      for j:=1 to n do grad_total_v[i,j]:=0;

      for each pair x/y_req of A_train do
      begin gradient(Θ,V,ϑ,W,x,y_req;
                     grad_ϑ,grad_w,grad_Θ,grad_v);
            for i:=1 to m do
            grad_total_ϑ[i]:=grad_total_ϑ[i]+grad_ϑ[i];
            for i:=1 to m do
            for j:=1 to p do
            grad_total_w[i,j]:=grad_total_w[i,j]
                               +grad_w[i,j];
            for i:=1 to p do
            grad_total_Θ[i]:=grad_total_Θ[i]
                             +grad_Θ[i];
            for i:=1 to p do
            for j:=1 to n do
            grad_total_v[i,j]:=grad_total_v[i,j]
                               +grad_v[i,j];
      end;
end;
```

Obrázok 5.27. Výpočet celkového gradientu účelovej funkcie pre celú tréningovú množinu. Algoritmus je inicializovaný vynulovaním jednotlivých zložiek celkového gradientu. Vlastný výpočet je vnorený do vonkajšieho for-cyklu, ktorý sa opakuje pre všetky páry $\boldsymbol{x}/\boldsymbol{y}_{\text{req}}$ tréningovej množiny A_{train}.

Na záver našich úvah o algoritmizácii neurónových sietí s dopredným šírením, pristúpime k ich adaptačnej fáze, ktorá spočíva v iteračnej úprave prahových a váhových

koeficientov tak, aby účelová funkcia (5.21) bola minimálna (prahové a váhové koeficienty sú určené ako riešenie minimalizačného problému (5.42)). Gradientová minimalizačná metóda najprudšieho spádu je vyjadrená vzťahmi (5.43), ich jednoduchou modifikáciou pre neurónovú sieť s troma vrstvami dostaneme tieto vzťahy

$$\begin{aligned} w_{ij}^{(k+1)} &= w_{ij}^{(k)} - \lambda \frac{\partial E}{\partial w_{ij}} + \mu \Delta w_{ij}^{(k)} \\ \vartheta_i^{(k+1)} &= \vartheta_i^{(k)} - \lambda \frac{\partial E}{\partial \vartheta_i} + \mu \Delta \vartheta_i^{(k)} \end{aligned} \tag{5.69}$$

pre i=1,2,...,m a j=1,2,...,p

$$\begin{aligned} v_{ij}^{(k+1)} &= v_{ij}^{(k)} - \lambda \frac{\partial E}{\partial v_{ij}} + \mu \Delta v_{ij}^{(k)} \\ \Theta_i^{(k+1)} &= \Theta_i^{(k)} - \lambda \frac{\partial E}{\partial \Theta_i} + \mu \Delta \Theta_i^{(k)} \end{aligned} \tag{5.70}$$

pre i=1,2,...,p a j=1,2,...,n, index k popisuje iteračný krok. Symboly Δ sú určené ako rozdiel koeficientov z predchádzajúcich dvoch krokov, tak napr. $\Delta w_{ij}^{(k)} = w_{ij}^{(k)} - w_{ij}^{(k-1)}$. Adaptačný proces je inicializovaný náhodne generovanými prahovými a váhovými koeficientmi, napr. z intervalu (–1,1). Ako je obvyklé v gradientových optimalizačných metódach, adaptačný proces je ukončený, keď hodnota celkového gradientu je menšia ako predpísané malé kladné číslo ε, $|\text{grad } E_{\text{tot}}| < \varepsilon$. Iná alternatíva ukončenia adaptačného procesu je, keď počet iterácií k dosiahne predpísaný počet $k_{\max}$. Pacalovský pseudokód adaptačného procesu je znázornený na obr. 5.28.

Principiálnu dôležitosť v adaptačnom procese neurónovej siete hrá rýchlosť učenia (parameter λ). Tento parameter sa obvykle položí rovný malému kladnému číslu, napr. λ=0,1). V mnohých prípadoch je vhodné tento parameter dynamicky meniť v závislosti na rýchlosti adaptácie neurónovej siete. V prípade, že hodnota účelovej funkcie sa zväčší, potom je potrebné parameter zmenšiť, napr. $\lambda \leftarrow \lambda/10$. V opačnom prípade, ak sa účelová funkcia monotónne zmenšuje, je vhodné zväčšiť parameter λ, napr. $\lambda \leftarrow 2\lambda$. Týmto jednoduchým spôsobom máme zabezpečenú približne optimálnu hodnotu rýchlosti učenia λ.

```
procedure adaptation(input :A_train,k_max,ε,λ,μ;
                     output:Θ,V,ϑ,W);
begin for i:=1 to m do
      begin ϑ[i]:=2*random-1; Δϑ[i]:=0 end;
      for i:=1 to m do
      for j:=1 to p do
      begin w[i,j]:=2*random-1; Δw[i,j]:=0 end;
      for i:=1 to p do
      begin Θ[i]:=2*random-1; ΔΘ[i]:=0 end;
      for i:=1 to p do
      for j:=1 to n do
      begin v[i,j]:=2*random-1; Δv[i,j]:=0 end;
      k:=0; length_grad_E=∞;
      while (k<k_max) and (length_grad_E>ε) do
      begin gradient_total(Θ,V,ϑ,W,A_train;
                           grad_total_ϑ,grad_total_w,
                           grad_total_Θ,grad_total_v);
            for i:=1 to m do
            begin Δ:=-λ*grad_total_ϑ[i]+μ*Δϑ[i];
                  ϑ[i]:=ϑ[i]+Δ; Δϑ[i]:=Δ
            end;
            for i:=1 to m do
            for j:=1 to p do
            begin Δ:=-λ*grad_total_w[i,j]+μ*Δw[i,j];
                  w[i,j]:=w[i,j]+Δ; Δw[i,j]:=Δ
            end;
            for i:=1 to p do
            begin Δ:=-λ*grad_total_Θ[i]+μ*ΔΘ[i];
                  Θ[i]:=Θ[i]+Δ; ΔΘ[i]:=Δ
            end;
            for i:=1 to p do
            for j:=1 to n do
            begin Δ:=-λ*grad_total_v[i,j]+μ*Δv[i,j];
                  v[i,j]:=v[i,j]+Δ; Δv[i,j]:=Δ
            end;
            length_grad_E:=|grad_total_ϑ|+|grad_total_w|+
                           |grad_total_Θ|+|grad_total_v|;
      end;
end;
```

Obrázok 5.28. Algoritmizácia adaptačného procesu neurónovej siete. Procedúra je inicializovaná vynulovaním momentových Δ-členov a náhodnou generáciou prahových a váhových koeficientov z intervalu $(-1,1)$ (premenná random je generátor náhodných čísel s rovnomernou distribúciou z intervalu $\langle 0,1)$). Vonkajší while-cyklus sa opakuje tak dlho, až buď počet iterácií k je väčší ako predpísaný počet k_{max} alebo norma (dĺžka) gradientu length_grad_E je menšia ako požadovaná presnosť ε. Premenné λ resp. μ označujú rýchlosť učenia (malé kladné číslo, napr. λ=0,1) resp. momentový člen (obvykle μ=0,5-0,7). Výstupné parametre procedúry sú prahové a váhové koeficienty adaptovanej neurónovej siete.

Literatúra

[1] M. Minsky and S. Papert. *Perceptrons. An Introduction to Computational Geometry*. The MIT Press, Cambridge, MA, 1969.

[2] D.E. Rumelhart, G.E. Hinton, and R.J. Wiliams. Learning internal representation by error propagation. In: D.E. Rumelhart, J.L. McClelland, and PDP Research Group. *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Vol 1: Foundation.* The MIT Press, Cambridge, MA, 1987, pp. 318-362.

[3] J. Sedláček. *Úvod do teorie grafů*. Academia, Praha, 1981.

[4] M. Novák. *Neuronové sítě a neuropočítače*. Edice Výber, SENZO a.s., Praha, 1992.

[5] M. Novák, J. Faber a O. Kufudaki. *Neuronové sítě a informační systémy živých organizmů*. Grada, Praha, 1992.

[6] C.L. Giles and T. Maxwell. Learning, invariance, and generalization in high-order neural networks. *Applied Optics* 26:4972-4978, 1987.

[7] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixture of local experts. *Neural Computation* 3:79-87,1991.

[8] V. Kvasnička. Adaptive mixture of local neural networks. *Neural Network World* 3:161-174, 1993.

[9] M. Demlová a J. Nagy. *Algebra*. Edice Matematika pro vysoké školy technické, sešit III. SNTL, Praha, 1982.

[10] S. Mika. *Numerické metody algebry*. Edice Matematika pro vysoké školy technické, sešit IV. SNTL, Praha, 1982.

[11] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Wetterling. *Numerical Recipes in Pascal. The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, 1992.

[12] I. Kluvánek, L. Mišík a M. Švec. *Matematika I*, Alfa, Bratislava, 1971.

[13] R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. *1st IEEE International Conference on Neural Networks*, San Diego, CA, Vol. 3, pp. 11-14, 1987.

[14] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks* 2:259-366, 1989.

[15] A. Lukasová a J. Šormanová. *Metody shlukové analýzy*. SNTL, Praha, 1985.

[16] D. Svozil, V. Kvasnička, and J. Pospíchal. Neural network prediction of carbon-13 NMR chemical shifts of alkanes. *Journal of Chemical Information and Computer Sciences* 35:924-928, 1995.

[17] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, London, 1982.

[18] L. Kučera. *Kombinatorické algoritmy*. SNTL, Praha, 1983.

6. Rekurentné neurónové siete

6.1 Prečo rekurentné siete?

V kapitole 5 venovanej viacvrstvovým neurónových sieťam a metódam ich trénovania sme videli, že tieto siete sú za určitých podmienok schopné naučiť sa asociovať vstupné vektory s požadovanými výstupnými vektormi. Zvyčajne požadujeme, aby sieť nefungovala len ako akási "look-up table" (kde sú dvojice vstup - požadovaný výstup "natvrdo" memorované), ale aby "inteligentne" reagovala aj na vstupy, ktoré jej pri trénovaní neboli ukázané. Inými slovami, boli by sme radi, ak by sieť správne "zovšeobecnila" trénovacie príklady. Rigoróznym úvahám o náročnosti trénovania, zovšeobecňovacím vlastnostiam sietí, kvalite trénovacej množiny (koľko trénovacích príkladov, aká je distribúcia trénovacích vzoriek, atď.) sa venuje disciplína nazvaná *teória učenia* (angl. *learning theory* [1]). Rozsah tejto práce neumožňuje podrobnejšie poznámky o tejto disciplíne. Obmedzíme sa len na konštatovanie, že viacvrstvová neurónová sieť zovšeobecní trénovacie vzorky tak, že nimi preloží nadplochu (pri lineárnych sieťach nadrovinu), ktorá je čo najmenej "zvlnená" (pozri obr. 5.21).

Táto požiadavka intuitívne reprezentuje staré pravidlo modelovania dát (tzv. *Occam's Razor* [2-3]), podľa ktorého by model nemal demonštrovať "štruktúry", ktoré nie sú obsiahnuté v dátach. Ak by napríklad trénovacia množina pozostávala z dvojíc (vstup, požadovaný výstup) ležiacich na nejakej nadrovine Π, intuitívne očakávame, že tam budú ležať aj dosiaľ nevidené asociačné dvojice (vstup, výstup). Lineárna sieť realizujúca zobrazenia vstupov na výstupy tak, že dvojice (vstup, výstup) ležia na Π, zrejme nevnáša do modelovania tie štruktúry, ktoré nemožno vystopovať v trénovacích dátach. Aj nelineárna viacvrstvová sieť zobrazujúca vstupy na výstupy, pričom dvojice (vstup, výstup) ležia na nadploche Ψ, ktorá je "mierne zvlnenou" verziou nadroviny Π, bude zrejme vyhovujúca. Naproti tomu, ak by nelineárna viacvrstvová sieť *presne* preložila trénovacími vzormi nadplochu Φ, ktorá by bola vysoko nelineárna (veľmi "zvlnená"), ťažko by sme mohli uveriť, že odpovede siete na vstupy neobsiahnuté v trénovacej množine budú mať niečo spoločné s tendenciou dát ležať na nadrovine Π. Voľne možno povedať, že takýto model vidí v dátach viac štruktúr, než v nich v skutočnosti je - fenomén známy pod menom *premodelovanie dát* (angl. *overfitting*, *overlearning*). V literatúre sú rozpracované metódy umožňujúce, aspoň do určitej miery, vyhnúť sa nástrahám premodelovania dát. Prípadných záujemcov odkazujeme na knihu [4].

Existujú však typy úloh, kde nelineárne viacvrstvové siete zlyhávajú, no nie v dôsledku nedostatku optimálnych trénovacích procedúr (pri nelineárnych sieťach dosiahneme len lokálne minimum na chybovom povrchu nad priestorom synaptických váh), či premodelovania dát. Inými slovami, nelineárne viacvrstvové siete by na tomto type úloh zlyhávali, aj keby sme dokázali spomenuté problémy spoľahlivo vyriešiť. Príčina tkvie v

samotnej podstate úlohy, ako je to napríklad pri úlohách, kde sa popri priestorových štruktúrach objavujú ešte aj časové štruktúry. Uvedieme si jednoduchý ilustračný príklad.

6.1.1 Príklad časovej štruktúry v dátach

Viacvrstvová sieť je schopná "naučiť sa" trénovaciu množinu pozostávajúcu napríklad z párov (vstup, požadovaný výstup)

$$A \rightarrow \alpha,\ B \rightarrow \beta,\ C \rightarrow \gamma,\ D \rightarrow \alpha$$

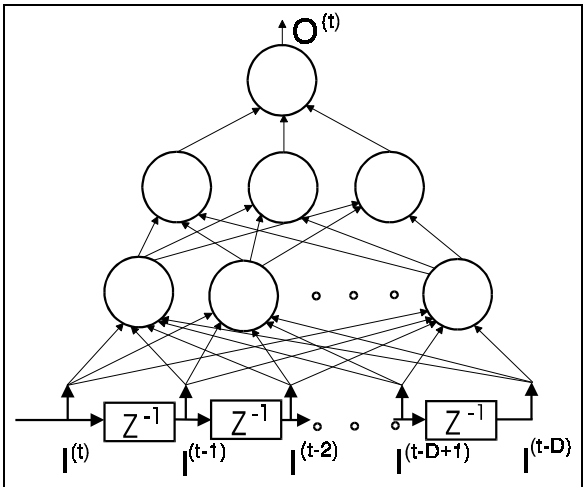
kde A,B,C,D sú vektory zo vstupného priestoru R^N a α,β,γ sú vektory z výstupného priestoru R^M. Sieť bude realizovať zobrazenie G: $R^N \rightarrow R^M$ tak, že $G(A) \cong \alpha$, $G(B) \cong \beta$, $G(C) \cong \gamma$ a $G(D) \cong \alpha$.

Trénovacie páry definujú určitú "priestorovú" štruktúru na priestore $R^N \times R^M$ párov (vstup, výstup) a táto štruktúra môže byť vystihnutá natrénovanou sieťou ako sme si spomenuli v úvode kapitoly. Predstavme si však, že trénovacia množina by mala nasledujúci tvar

$$A \rightarrow \alpha,\ B \rightarrow \beta,\ B \rightarrow \alpha,\ B \rightarrow \gamma,\ C \rightarrow \alpha,\ C \rightarrow \gamma, D \rightarrow \alpha$$

Vidíme, že k jednému vstupu môžeme mať viacero výstupov, v závislosti od *časového kontextu* tej-ktorej asociácie. Inými slovami, o výstupe siete by nemal rozhodovať len vstup siete, ale aj informácia o doterajšej histórii predkladaných vzoriek. Viacvrstvová sieť by mala byť rozšírená o možnosť reprezentovať časový kontext, aby tak mohla na základe predloženého vstupu lepšie rozhodnúť o výstupe. Architektonicky najjednoduchšie riešenie ponúka tzv. *neurónová sieť s časovým posunom* (angl. *Time Delay Neural Network,* TDNN) (obr. 6.1).

TDNN v podstate poskytuje viacvrstvovej sieti "okno do minulosti" - okrem momentálneho vstupu (v čase t) "vidí" sieť ešte aj vstupy z minulých D krokov (v časoch t-1, t-2, ..., t-D). Takúto sieť je možné trénovať klasickou procedúrou spätného šírenia (angl. *Back Propagation*, BP, pozri kapitolu 5), pričom je dôležité *zachovať poradie trénovacích vzoriek* v trénovacej množine.



Obrázok 6.1. Architektúra TDNN. Okrem aktuálneho prvku časovej postupnosti vstupov (v čase t), sú viacvrstvovej sieti prezentované aj vstupy z predošlých D krokov diskrétneho času. Bloky Z^{-1} sú oneskorovacie prvky.

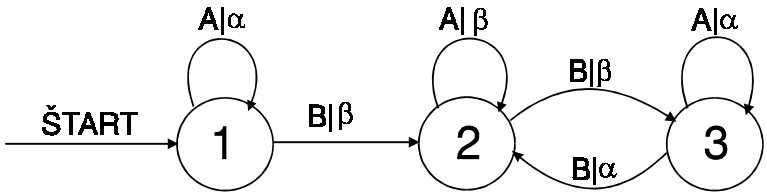
Ak máme šťastie, aj takéto jednoduché rozšírenie viacvrstvovej architektúry môže priniesť úspech a sieť typu TDNN je schopná popri priestorovej štruktúre postihnúť aj časovú štruktúru skrytú v trénovacích dátach. Výhodou architektúry TDNN je pomerná jednoduchosť a možnosť trénovania klasickou procedúrou BP vhodnou pre zvyčajné viacvrstvové siete. Nevýhodou tejto architektúry je, že spôsob reprezentácie časového kontextu nemusí byť dostatočne silný na zvládnutie časovej štruktúry trénovacích dát. Treba podotknúť, že aj v prípade, keď TDNN je schopná reprezentovať časovo-priestorovú štruktúru dát, nie je jednoduché len na základe trénovacej množiny správne odhadnúť dĺžku D "okna do minulosti" (viac podrobností nájde prípadný záujemca v [5-6]). Napriek tomu architektúra TDNN našla uplatnenie v mnohých oblastiach pracujúcich s časovo-priestorovými štruktúrami, napríklad v robotike, rozpoznávaní reči, atď. [7-10].

Pokúsme sa teraz odpovedať na otázku, kedy je architektúra TDNN apriori nevhodná na reprezentáciu časovo-priestorovej štruktúry trénovacích dát. Pre jednoduchosť si predstavme, že máme len konečnú množinu možných vstupných vektorov (napríklad $A,B \in R^N$) a konečnú množinu výstupných vektorov (napríklad $\alpha,\beta \in R^M$). Potom môžeme vstupy aj výstupy reprezentovať symbolmi z nejakej konečnej abecedy. Predpoklad architektúry TDNN, že na úvahu o možnom výstupe v čase t nám postačí informácia o terajšom vstupe a D predošlých vstupoch je analogický predpokladu stacionárneho markovovského procesu rádu D+1, kde pravdepodobnosť symbolu v reťazci závisí len od D+1 jeho bezprostredných predchodcov.

Predstavme si však, že proces reprezentovaný trénovacou množinou

$$A\rightarrow\alpha,\ A\rightarrow\alpha,\ B\rightarrow\beta,\ B\rightarrow\beta,\ B\rightarrow\alpha,\ A\rightarrow\beta,\ A\rightarrow\beta,\ A\rightarrow\beta$$

je popísaný Mealyho automatom [11,12} na obr. 6.2.



Obrázok 6.2. Mealyho automat popisujúci asociačnú úlohu s časovým kontextom. V závislosti od momentálneho stavu sú k vstupom *A,B* jednoznačne určené ich asociačné výstupy.

Spracovanie vstupného slova automatom sa začína v stave 1 označenom šípkou ŠTART. Po príchode vstupného symbolu $V \in \{A,B\}$ sa presunieme do nového stavu z množiny stavov {1,2,3} pozdĺž šípky prislúchajúcej vstupnému symbolu *V*, pričom so symbolom *V* asociujeme výstup $W \in \{\alpha,\beta\}$ podľa pravidla $V|W$. Teda po príchode symbolu *A* sa z počiatočného stavu 1 dostávame slučkou späť do stavu 1 a asociovaným výstupom je α. To sa zopakuje aj po opätovnom príchode vstupu *A*. Avšak vstup *B* nás prenesie do stavu 2 a príslušný asociovaný výstup je β, atď...

Stavy automatu kódujú históriu vstupných vektorov, aby sme mohli vždy bez váhania odpovedať na otázku, čo bude asociovaný výstup k danému vstupu pri danej histórii predkladaných vstupov. Vidíme, že takáto *stavová reprezentácia časového kontextu* predkladaných vzoriek môže byť omnoho úspornejšia ako reprezentácia časového kontextu pomocou "okna do minulosti" a niekedy aj nevyhnutná. Môže sa totiž stať, že by sme potrebovali potenciálne neobmedzene dlhé okno do minulosti. Ak by sme boli v stave 1 automatu na obr. 6.2, môže prísť ľubovoľný počet vstupov *A* a asociovaný výstup je α. To isté patrí aj o stave 3. Podstatný rozdiel je však vo výstupe asociovanom so vstupom *B*. Ten je β, v prípade stavu 1 a α v prípade stavu 3. Nie je možné zvoliť žiadne konečné *D,* aby za každých okolností bolo možné na základe minulých vstupov rozhodnúť o výstupe asociovanom k vstupu *B*. Zrejme pre dobré zovšeobecnenie trénovacej množiny reprezentujúcej časovo-priestorovú štruktúru popísanú automatom na obr. 6.2 bude architektúra TDNN nevyhovujúca.

6.1.2 Predbežný príklad rekurentnej neurónovej siete

Inšpirovaní predchádzajúcimi úvahami uveďme architektúru neurónovej siete (obr. 6.3) zloženej z dvoch viacvrstvových sietí, a to

- *asociačnej siete* - realizujúcej asociáciu výstupu s daným vstupom na základe "vnútornej pamäti" siete a

- *stavovej siete* - realizujúcej kódovanie doterajšej histórie vstupov predložených sieti.

Architektúra bola navrhnutá v [13]. Obe viacvrstvové siete zdieľajú spoločnú vstupnú vrstvu, ktorá sa skladá zo vstupných neurónov zabezpečujúcich prekopírovanie vstupného vektora $\boldsymbol{I}^{(t)} = \left(I_1^{(t)}, I_2^{(t)}, \ldots, I_n^{(t)}, \ldots, I_N^{(t)}\right)$ v čase t do siete a zo "stavových" neurónov, ktorých aktivácie v čase t tvoria *stav siete* $\boldsymbol{S}^{(t)} = \left(S_1^{(t)}, S_2^{(t)}, \ldots, S_l^{(t)}, \ldots, S_L^{(t)}\right)$ kódujúci históriu predkladaných vstupov $I^{(\tau)}, \tau < t$. Asociačná sieť má tri vrstvy, okrem vstupnej vrstvy, ešte skrytú vrstvu neurónov druhého rádu, ktorých aktivácie v čase t sú počítané nasledovne (j je index neurónov v skrytej vrstve)

$$H_j^{(t)} = g\left(\sum_{l,n} Q_{j\,ln} S_l^{(t)} I_n^{(t)}\right) \tag{6.1}$$

kde g je obvyklá sigmoidálna aktivačná funkcia

$$g(u) = \frac{1}{1+e^{-u}} \tag{6.2}$$

Výstupná vrstva neurónov prvého poriadku reprezentuje výstup siete, ktorým sú v čase t aktivácie (m je index výstupného neurónu)

$$O_m^{(t)} = g\left(\sum_k V_{mk} H_k^{(t)}\right) \tag{6.3}$$

Stavová sieť sa skladá len z dvoch vrstiev. Úlohou druhej vrstvy je vypočítať reprezentácie nového časového kontextu (ktorý sa bude považovať za stav siete v čase t+1), ktorý vznikol príchodom vstupu $\boldsymbol{I}^{(t)}$

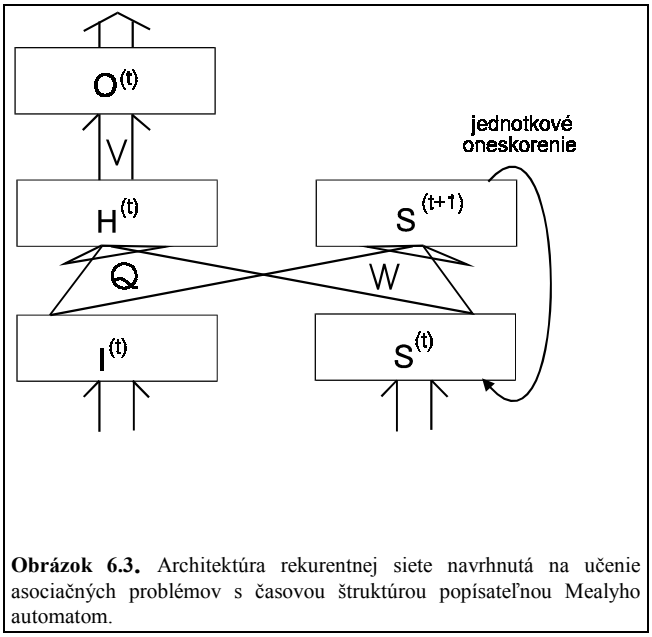
$$S_i^{(t+1)} = g\left(\sum_{l,n} W_{i\,ln} S_l^{(t)} I_n^{(t)}\right) \tag{6.4}$$

Ak sú vstupné vektory $\boldsymbol{I}^{(t)}$ z R^N, výstupné vektory $\boldsymbol{o}^{(t)}$ z R^M, stav siete je kódovaný L rozmerným vektorom z R^L (máme L stavových neurónov) a sieť má K neurónov v skrytej vrstve, potom v architektúre siete je L^2N váh $W_{i\,ln}$, KLN váh $Q_{j\,ln}$ a MK váh V_{mk}. V čase t sa na základe vstupu $\boldsymbol{I}^{(t)}$ a stavu $\boldsymbol{S}^{(t)}$ vypočíta stav siete v čase t+1, ktorý sa prekopíruje do časti vstupnej vrstvy v nasledujúcom kroku diskrétneho času.

Zrejme takáto architektúra siete je schopná reprezentovať časovo-priestorové štruktúry podobné štruktúre zobrazenej ako automat na obr. 6.2. Asociačnú *viacvrstvovú sieť sme* totiž *rozšírili o vnútornú pamäť*. Navyše, vo vrstve stavových neurónov si sieť môže utvoriť vlastnú stavovú reprezentáciu časového kontextu predkladaných vstupov.

Namieste je však otázka, ako učiť takýto typ siete, teda ako na základe *trénovacej množiny časovo usporiadaných asociačných dvojíc* (vstup, výstup) vyprodukovať váhy *W*, *Q*, *V*, ktoré zabezpečia "správnu" funkciu siete (zodpovedajúcu trénovacej množine). Treba si uvedomiť, že k dispozícii máme len dvojice (vstup, výstup) a preto aj stavové neuróny možno považovať za skryté. Z tohoto pohľadu máme v architektúre dva typy skrytých neurónov:

- *rekurentné* - vo vrstvách $S^{(t)}$, $S^{(t+1)}$,
- *nerekurentné* - vo vrstve $H^{(t)}$.



Obrázok 6.3. Architektúra rekurentnej siete navrhnutá na učenie asociačných problémov s časovou štruktúrou popísateľnou Mealyho automatom.

V čase t je na vstupe vektor $\boldsymbol{I}^{(t)} = \left(I_1^{(t)}, I_2^{(t)}, \ldots, I_N^{(t)}\right)$ a výstup siete je vektor $\boldsymbol{O}^{(t)}$ aktivácií výstupných neurónov $\boldsymbol{O}^{(t)} = \left(O_1^{(t)}, O_2^{(t)}, \ldots, O_M^{(t)}\right)$. Pokúsme sa zareagovať na vzniknutú disproporciu medzi skutočným výstupom siete $\boldsymbol{O}^{(t)}$ a želaným výstupom $\boldsymbol{D}^{(t)}$ zmenou váh, ktorá by ju zmiernila. Inšpirovaní myšlienkou procedúry BP z viacvrstvových sietí definujeme chybový funkcionál (účelovú funkciu, pozri formulu (5.2a))

$$E = \frac{1}{2}\sum_m \left(D_m^{(t)} - O_m^{(t)}\right)^2$$

a upravme váhy V, Q, W proporcionálne k inverzným gradientom

$$\Delta V_{mk} = -\alpha \frac{\partial E}{\partial V_{mk}} \tag{6.5}$$

$$\Delta Q_{j\,In} = -\alpha \frac{\partial E}{\partial Q_{j\,In}} \tag{6.6}$$

$$\Delta W_{i\,In} = -\alpha \frac{\partial E}{\partial W_{i\,In}} \tag{6.7}$$

kde α je "malá" kladná konštanta nazývaná *rýchlosť učenia* (angl. *learning rate*).

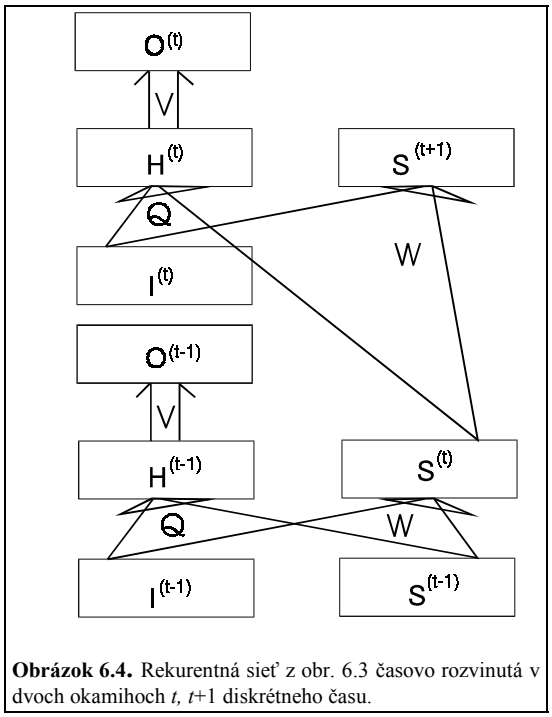
Pre podrobnejší výpočet parciálnych derivácií uvedených v (6.5), (6.6) a (6.7) je vhodné prekresliť obr. 6.3 na *sieť rozvinutú v čase* (konkrétne v časoch t, t+1) (obr. 6.4). Postupom analogickým postupu pri odvodení klasického algoritmu BP (pozri kapitolu 5) dostávame parciálne derivácie chybového funkcionálu E podľa váh V a Q

$$\frac{\partial E}{\partial V_{mk}} = \left(O_m^{(t)} - D_m^{(t)}\right) g'\left(\phi\left(O_m^{(t)}\right)\right) H_k^{(t)} \tag{6.8}$$

$$\frac{\partial E}{\partial \phi\left(O_m^{(t)}\right)} = \left(O_m^{(t)} - D_m^{(t)}\right) g'\left(\phi\left(O_m^{(t)}\right)\right) \tag{6.9}$$

$$\frac{\partial E}{\partial Q_{j\,In}} = S_l^{(t)} I_n^{(t)} g'\left(\phi\left(H_j^{(t)}\right)\right) \sum_m V_{mj} \frac{\partial E}{\partial \phi\left(O_m^{(t)}\right)} \tag{6.10}$$

kde g' je derivácia funkcie g, $g'(u) = g(u)(1 - g(u))$ a ϕ je inverzná funkcia k funkcii g. Konkrétne $\phi\left(O_m^{(t)}\right) = \sum_k V_{mk} H_k^{(t)}$ a $\phi\left(H_j^{(t)}\right) = \sum_{l,n} Q_{j\,In} S_l^{(t)} I_n^{(t)}$. Inverzná funkcia ϕ odpovedá tzv. postsynaptickému potenciálu neurónu (pozri kapitolu 1).



Obrázok 6.4. Rekurentná sieť z obr. 6.3 časovo rozvinutá v dvoch okamihoch *t, t*+1 diskrétneho času.

Výpočet parciálnych derivácií $\partial E/\partial W_{i\,ln}$ je troška zložitejší. V čase *t*-1 váha $W_{i\,ln}$ priamo ovplyvňuje iba aktiváciu $S_i^{(t)}$ *i*-teho stavového neurónu v budúcom kroku (v čase t) a teda

$$\frac{\partial E}{\partial W_{i\,ln}} = \frac{\partial E}{\partial S_i^{(t)}} \frac{\partial S_i^{(t)}}{\partial W_{i\,ln}} \tag{6.11}$$

Chyba *E* závisí od stavu $S_i^{(t)}$ prostredníctvom váh *V*,*Q* asociačnej siete, a preto

$$\frac{\partial E}{\partial S_i^{(t)}} = \sum_m \frac{\partial E}{\partial \phi\left(O_m^{(t)}\right)} \sum_k V_{mk}\, g'\left(\phi\left(H_k^{(t)}\right)\right) \sum_n Q_{kin} I_n^{(t)} \tag{6.12}$$

Stav $S_r^{(t)}$ *r*-tého stavového neurónu v čase *t* závisí priamo od váhy $W_{i\,ln}$ len ak *i*=*r*, no je dôležité si uvedomiť, že nepriamo závisí aj od všetkých ostatných váh $W_{i\,ln}$, keďže

$$S_r^{(t)} = g\left(\sum_{a,b} W_{rab} S_a^{(t-1)} I_b^{(t-1)}\right)$$

a stavy $S_b^{(t-1)}$ závisia len od váh *W* (a, pravda, vstupu $I^{(t-2)}$). Dostávame teda

$$\frac{\partial S_r^{(t)}}{\partial W_{i\,ln}} = g'\left(\phi\left(S_r^{(t)}\right)\right)\left[\delta_{ri} S_l^{(t-1)} I_n^{(t-1)} + \sum_{a,b} W_{rab} I_b^{(t-1)} \frac{\partial S_a^{(t-1)}}{\partial W_{i\,ln}}\right] \tag{6.13}$$

kde δ_{ri} je Kroneckerovo delta: $\delta_{ri} = 1$, pre $r = 1$, $\delta_{ri} = 0$, pre $r \neq i$. Pomocou rekurentného vzťahu (6.13) je možné v každom kroku trénovania nanovo prepočítať potrebné parciálne derivácie $\partial S_r^{(t)} / \partial W_{i\,ln}$. Tieto sa použijú v ďalšom kroku pre nový výpočet parciálnych derivácií podľa vzťahu (6.13). Na začiatku trénovania je vhodné zvoliť $\partial S_r^{(0)} / \partial W_{i\,ln}$ nulové.

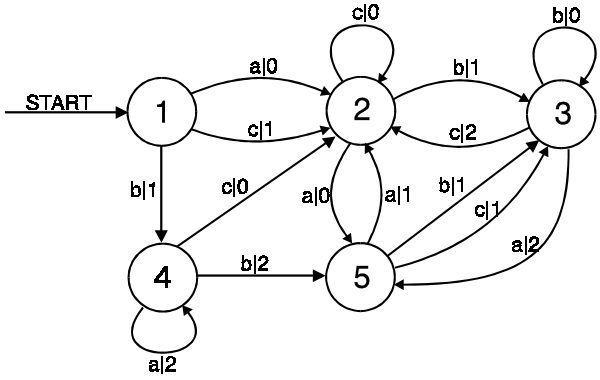
6.1.3 Príklad trénovania rekurentnej neurónovej siete

V tejto stati si krátko popíšeme proces trénovania rekurentnej siete na vzorkách generovaných Mealyho automatom. Pre ilustráciu procesu učenia rekurentnej siete uvažujme automat na obr. 6.5 [13].

Sieti budeme predkladať dvojice (vstupné slovo, odozva na vstupné slovo), ktoré reprezentujú náš automat. Začneme s kratšími slovami a postupne pokročíme k dlhším slovám, napríklad

acccb→00001, *caaab*→10101, *bbbbbb*→121000, atď.

Dôležité je reprezentovať v trénovacej množine všetky aspekty automatu, teda aj to, že spracovanie každého vstupného slova sa začína v iniciálnom stave 1. Jedna z možností je zavedenie zvláštneho vstupného aj výstupného symbolu, ktoré by signalizovali "reset". Teda z každého stavu automatu by príchod symbolu "!" znamenal prechod do stavu 1 s príslušným asociovaným výstupom "*x*". Trénovacia množina by potom vyzerala nasledovne



Obrázok 6.5. Mealyho automat použitý na generovanie trénovacej množiny pre rekurentnú sieť na obr. 6.3.

$$acccb!\rightarrow 00001x,\ caaab!\rightarrow 10101x,\ bbbbbb!\rightarrow 121000x,\ \text{atď.}$$

alebo

$$acccb!caaab!bbbbbb!... \rightarrow 00001x10101x121000x...$$

čo je v prepise do časovo usporiadanej trénovacej množiny (vstup, výstup)

$$a\rightarrow 0,\ c\rightarrow 0,\ c\rightarrow 0,\ c\rightarrow 0,\ b\rightarrow 1,\ !\rightarrow x,\ c\rightarrow 1,\ a\rightarrow 0,\ a\rightarrow 1,\ a\rightarrow 0,\ b\rightarrow 1,\ 1\rightarrow x,\ \text{atď.}$$

Máme teda štyri vstupné symboly ***a,b,c,"!"*** a štyri výstupné symboly **0,1,2,"*x*".** Reprezentujme ich binárne v tzv. "one-hot" kódovaní - 4-dimenzionálne kódy s práve jednou 1 a troma 0, pričom pozícia 1 kóduje príslušný symbol. Možné kódovanie je uvedené v nasledujúcej tabuľke 6.1:

Tabuľka 6.1

vstupný symbol	výstupný symbol	kód
a	0	1000
b	1	0100
c	2	0010
!	x	0001

Zrejme potrebujeme 4 vstupné (N=4) a 4 výstupné (M=4) neuróny. Pre náš experiment použijeme 4 rekurentné stavové neuróny (L=4) a 4 skryté nerekurentné neuróny (K=4). Trénovaciu množinu sme vygenerovali tak, že k 600 náhodne vybratým vstupným slovám nad abecedou {a,b,c} sme pomocou automatu na obr. 6.5 určili prislúchajúce výstupné slová nad abecedou {0,1,2}. Na koniec každého trénovacieho vstupného slova sme vložili "resetovací" symbol "!" a na koniec odpovedajúceho výstupného slova bol vložený symbol "x". Dĺžka slov sa pohybovala od 3 do 12 a rástla od najkratších k najdlhším. Na začiatku trénovania boli náhodne vygenerované váhy V,Q,W z intervalu [-0.5, 0.5] podľa rovnomerného rozdelenia pravdepodobnosti. Počas učenia sa z trénovacej množiny postupne berie vstup za vstupom a ich asociované výstupy, pričom po prezentácii každého vstupu sa príslušne upravia váhy. Trénovací proces sa ukončil po 18 epochách (jedna epocha spočíva v *postupnom* prejdení všetkých asociačných dvojíc v trénovacej množine) s trénovacou chybou 0,075. Naučená sieť bola testovaná na náhodne vygenerovaných vstupných slovách omnoho väčšej dĺžky ako 12 (čo bola maximálna dĺžka trénovacích slov). Odpovede na všetky testovacie (vstupné) slová, t.j. k nim prislúchajúce výstupné slová generované sieťou, zodpovedali automatu na obr. 6.5. Pred predložením každého testovacieho vstupného slova bola sieť "resetovaná" pomocou vstupu "!".

Ako zaujímavosť spomenieme, že v tomto prípade bolo možné "porozumieť" vnútornej reprezentácii problému (implicitne určeného trénovacou množinou) v naučenej rekurentnej sieti. Experimentálne sa totiž ukázalo, že stavy siete (4-rozmerné vektory aktivácií stavových neurónov) nepokrývajú stavový priestor $(0,1)^4$ rovnomerne, ale sú koncentrované v dobre detekovateľných zhlukoch. Navyše, tieto zhluky zodpovedajú stavom automatu, na základe ktorého bola vygenerovaná trénovacia množina.

Takýmto spôsobom možno z naučenej siete "vytiahnuť" automat, ktorý vyhovuje trénovacej množine a navyše ju aj zovšeobecňuje.

6.2 Rekurentné siete a ich trénovanie

6.2.1 Modely rekurentných sietí

V predošlej podkapitole sme si na príklade intuitívne vymedzili pojem rekurentnej siete a ukázali sme si prístup k jej učeniu. Vo všeobecnosti možno za rekurentnú sieť považovať akúkoľvek neurónovú sieť, v ktorej istá podmnožina neurónov (*rekurentné neuróny*) je schopná uchovať informáciu o svojich aktiváciách v predošlých časoch pre výpočet aktivácií neurónov v čase *t*+1. "Odpamätané" hodnoty sa objavia v čase *t*+1 ako aktivácie tzv. *kontextových neurónov*. Napríklad v architektúre rekurentnej siete z predošlej podkapitoly sú rekurentné neuróny vo výstupnej vrstve stavovej siete a kontextové neuróny tvoria časť vstupnej vrstvy asociačnej aj stavovej siete, kde sa v čase *t*+1 objavia aktivácie rekurentných neurónov z kroku *t*. Povedali sme si, že takýmto spôsobom rozširujeme neurónovú sieť o *vnútornú pamäť*.

Historicky vzniklo niekoľko modelov rekurentných sietí, ktoré možno považovať za viacvrstvové siete obohatené o rekurentné neuróny. Na obr. 6.6 a-c uvádzame tri modely tohoto typu, ktoré možno nájsť v literatúre. Vrstvy neurónov sú zobrazované obdĺžnikmi. Podobne ako v tradičných viacvrstvových sieťach, v rámci jednej vrstvy nie sú neuróny navzájom prepojené a prepojenia existujú len medzi neurónmi susedných vrstiev. Hrubé šípky reprezentujú prepojenia z každého neurónu spodnej vrstvy do každého neurónu hornej vrstvy. Tieto prepojenia majú váhy, ktoré sú modifikovateľné (počas trénovacieho procesu). Jednoduché šípky predstavujú *rekurentné prepojenia* medzi zodpovedajúcimi neurónmi východzej a cieľovej vrstvy. Prepojenia majú váhu 1, ktorá je nemenná a existujú len medzi *i*-tym neurónom východzej a *i*-tym neurónom cieľovej vrstvy. Na týchto prepojeniach sú oneskorovacie členy, ktorých doba oneskorenia zodpovedá jednotke diskrétneho času. Funkcia rekurentných prepojení spočíva v odpamätaní aktivácií rekurentných neurónov a ich zavedení do kontextových neurónov.

Architektúru na obr. 6.6a navrhol Elman [14]. Kontextová vrstva obsahuje kópie aktivácií skrytých neurónov z predošlého kroku. Autorom siete uvedenej na obr. 6.6b je Jordan [15]. Obr. 6.6c predstavuje kombináciu modelov na obr. 6.6a-b. Ako navrhuje Bengio [16], je možné mať zvláštnu kontextovú vrstvu pre rôzne vrstvy pôvodnej viacvrstvovej siete, ako je tomu pri architektúre na obr. 6.6c.

Ďalším variantom odpamätávania aktivácií rekurentných neurónov v kontextovej vrstve je postupné "nabaľovanie" hodnôt aktivácií v minulých krokoch diskrétneho času, s prvkom "zabúdania" dávnejších aktivácií. Nech $S_i^{(t)}$ je aktivácia *i*-teho rekurentného neurónu v čase *t* a $K_i^{(t)}$ aktivácia *i*-teho kontextového neurónu v čase t. Potom

$$K_i^{(t+1)} = \alpha K_i^{(t)} + S_i^{(t)} \tag{6.14}$$

$0 < \alpha < 1$ je konštanta reprezentujúca "*rýchlosť zabúdania*". Iterovaním (6.14) totiž dostávame

$$K_i^{(t+1)} = S_i^{(t)} + \alpha S_i^{(t-1)} + \alpha^2 S_i^{(t-2)} + \ldots = \sum_{\tau=0}^{t} \alpha^{t-\tau} S_i^{\tau} \qquad (6.15)$$

Pre koeficient α blízky 1 kódujú kontextové aktivácie časový kontext aktivácií odpovedajúcich rekurentných neurónov v širokom časovom rozpätí, avšak zároveň strácame detailnú informáciu o posledných aktiváciach rekurentných neurónov. Naproti tomu, pri hodnotách koeficienta α blízkych 0 kódujeme vývoj aktivácií rekurentných neurónov len v bezprostrednej minulosti, avšak s väčším dôrazom na detailnú informáciu o ich hodnotách.

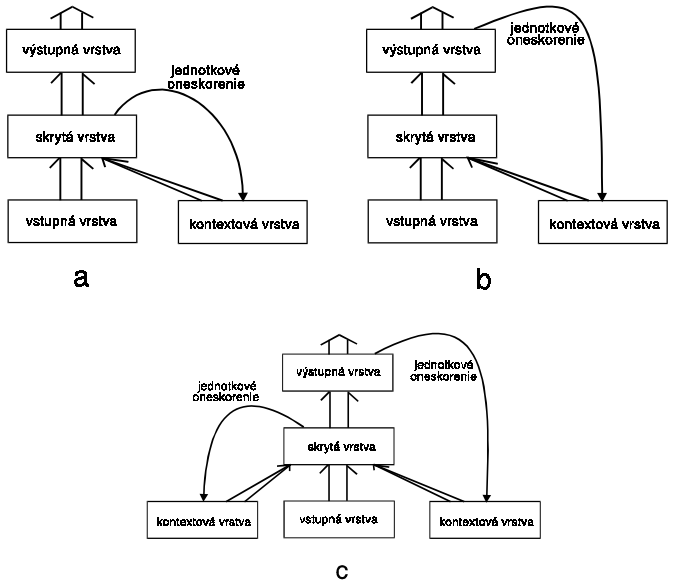
Takýto typ "odpamätávania" aktivácií rekurentných neurónov budeme označovať čiarkovaným prepojením medzi rekurentnou a kontextovou vrstvou. Architektúru na obr. 6.7a navrhol Jordan [15]. Ide o rozšírenie siete z obr. 6.6b. *i*-ty kontextový neurón uchováva informáciu o svojich aktiváciach v minulých krokoch a o aktivácii *i*-teho výstupného neurónu v predošlom kroku.

Model na obr. 6.7b pochádza od Stornetta a spol. [17]. Ide vlastne o klasickú viacvrstvovú sieť, ktorej vstupné neuróny kódujú históriu predložených vstupov v minulosti. Mozer [18] je autorom architektúry na obr. 6.7c. Sieť má k dispozícii informáciu o minulom vývoji aktivácií neurónov v skrytej vrstve.

Je len pochopiteľné, ak si čitateľ na tomto mieste položí dve otázky:

1. Prečo existuje toľko rôznych variantov rekurentných sietí?

2. Ako sformulovať pravidlá pre učenie takýchto sietí ?



Obrázok 6.6. Architektúry rekurentných sietí, ktoré možno považovať za klasické viacvrstvové siete obohatené o rekurentné neuróny. Aktivácie rekurentných neurónov v kroku t sa objavia ako aktivácie kontextových neurónov v kroku t+1.

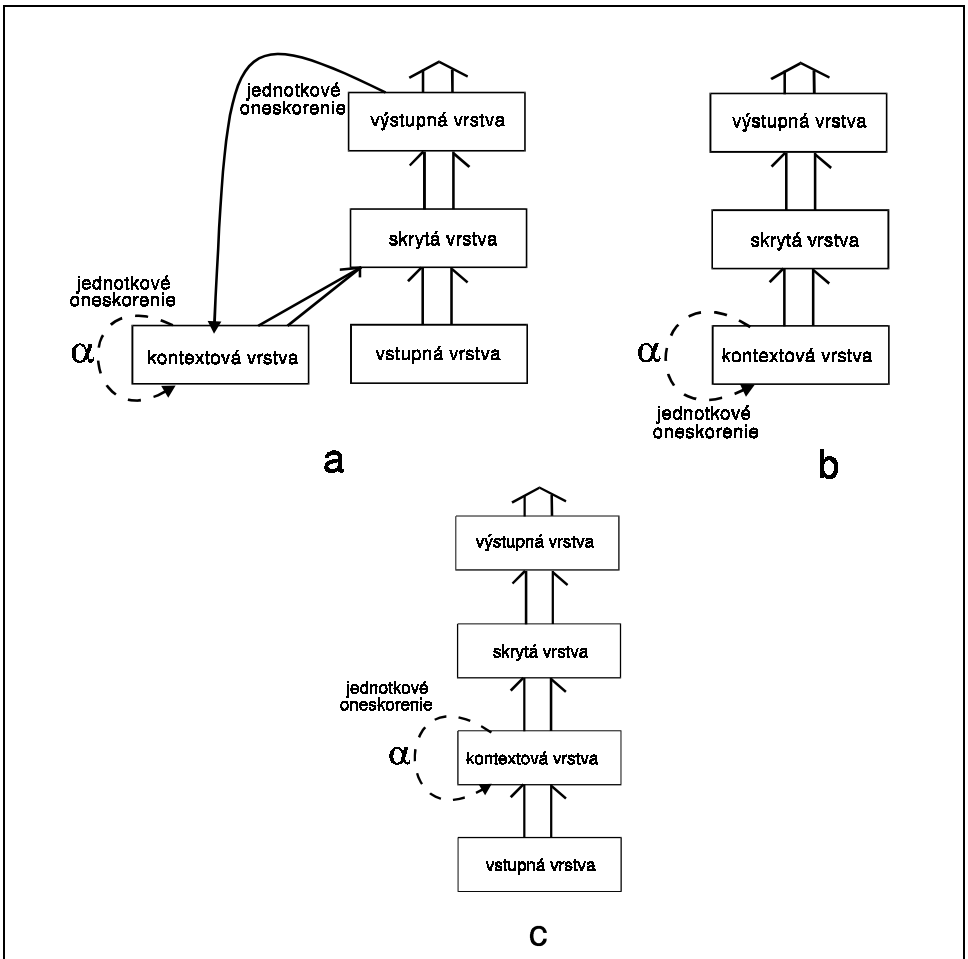
Odpoveď na prvú otázku je viac-menej priamočiara. Rôzne typy úloh sa vyznačujú rôznym typom časovo-priestorovej štruktúry a rôzne spôsoby kódovania časového kontextu vyhovujú rôznym časovým štruktúram v dátach. Nemalú úlohu hrá samozrejme aj otázka použiteľnosti trénovacieho procesu pri danom spôsobe kódovania časového kontextu. V zásade sa neurónové siete uplatňujú pri troch typoch úloh:

I. Klasifikačné alebo *asociačné úlohy*, aké poznáme už z kapitoly o viacvrstvových sieťach, avšak *s časovým kontextom.*

II. Predikčné úlohy.

III. Generatívne úlohy.

Pri prvom type úloh ide, v prípade klasifikácie, o rozhodnutie či práve ukončená postupnosť vstupov patrí, alebo nepatrí do nejakej triedy, prípadne do ktorej z možných tried ju možno zaradiť. Sem možno zaradiť napríklad klasifikáciu postupností symbolov z nejakej konečnej abecedy A (teda slov nad abecedou A) na základe príslušnosti k danému jazyku [19]. Príklad asociačnej úlohy s časovým kontextom bol uvedený v minulej podkapitole.



Obrázok 6.7. Architektúry rekurentných sietí s odpamätávaním aktivácií rekurentných neurónov v širšom časovom rozmedzí s prvkom zabúdania.

V druhom type úloh sa pokúšame nájsť časovú štruktúru v postupnosti dát, ktorá by umožnila na základe určitého úseku histórie dát v časoch menších ako *t* predpovedať dáta v čase väčšom ako *t*.

Tretím typom úloh je komplikovanejšia verzia predikčných úloh. Tentoraz nejde len o predikovanie hodnoty dát v niektorom budúcom čase. Na základe pozorovania určitého úseku vývoja dát je úlohou *pokračovať* v časovom rade dát zohľadňujúc základnú tendenciu dát skrytú v dostupnom úseku. Napríklad, ak by sme pozorovali úsek dát

23123123123123123...

zrejmé pokračovanie by bolo

... 123123123...

V reálnych úlohách však časová štruktúra dát môže byť omnoho zložitejšia než prísna periodicita časového radu. V prípade zložitejších postupností je cieľom modelovania časových radov vystihnutie základných štatistických či dynamických charakteristík modelovanej časovej postupnosti, napríklad, invariantnej miery, metrickej entropie, fraktálnej dimenzie atraktora, atď. Samotné generovanie pokračovania úseku časovej rady sa môže realizovať napríklad nasledovným spôsobom: Po predložení dostupného úseku dát (do času t) sieť vygeneruje predikciu možnej hodnoty dát v nasledujúcom čase t+1. Táto predikcia sa priradí k pôvodnému úseku a na základe takto vytvoreného nového úseku dát vygenerujeme predikciu pre čas t+2, atď... Napríklad model siete zobrazený na obr. 6.6a bol úspešne použitý pre klasifikáciu kratších slov nad konečnou abecedou symbolov, ako aj na generovanie krátkodobých pokračovaní symbolických postupností [14].

Model z obr. 6.7a bol použitý Andersonom [20] na kategorizovanie hovorených slabík anglického jazyka. Sieť trénovaná na jednej skupine hlasov bola schopná správne fungovať pri nových, vopred nepočutých hlasoch. Architektúra na obr. 6.7c sa lepšie hodí na problém klasifikácie postupností ako pre generatívne úlohy [4].

6.2.2 Trénovanie rekurentných sietí

V predošlej podkapitole sme si na príklade ukázali ako zovšeobecniť trénovaciu procedúru BP, pôvodne navrhnutú pre viacvrstvové siete, aby bola použiteľná aj pre viacvrstvové siete obsahujúce rekurentné neuróny. V tejto podkapitole sa budeme systematickejšie zaoberať problémom učenia rekurentných sietí. Spomenieme si dva najčastejšie používané prístupy, ktoré je možné vystopovať v literatúre. Oba sú založené na myšlienke minimalizačnej metódy najprudšieho spádu (angl. *steepest descent*) do minima chybového funkcionálu E pohybom proti smeru gradientu ∇E.

Ako pri algoritme BP, aj tu bude základnou úlohou nájdenie analytických vzťahov vyjadrujúcich parciálne derivácie chybového funkcionálu E podľa jednotlivých modifikovateľných váh siete. Kvôli jednoduchosti prezentácie budeme uvažovať rekurentnú sieť zloženú z dvoch rekurentných neurónov prvého rádu. Pre ich aktivácie platí

$$S_i^{(t+1)} = g\left(\sum_{j=1}^{2} w_{ij} S_j^{(t)} + I_i^{(t)}\right) \qquad (i = 1,2) \tag{6.16}$$

kde $S_1^{(t)}$ (resp. $S_2^{(t)}$) je aktivácia prvého (resp. druhého) rekurentného neurónu v čase t a $I_1^{(t)}$ (resp. $I_2^{(t)}$) je vonkajší vstup (cez kanál váhy 1) do prvého (resp. druhého) rekurentného neurónu v čase t, g môže byť napríklad známa sigmoidálna funkcia (6.2). Prípad, že aj vonkajšie vstupy vchádzajú do rekurentných neurónov cez kanály s modifikovateľnými váhami by bol riešený analogicky, ale prezentácia by bola zbytočne zaťažená.

6.2.3 Spätné šírenie v čase

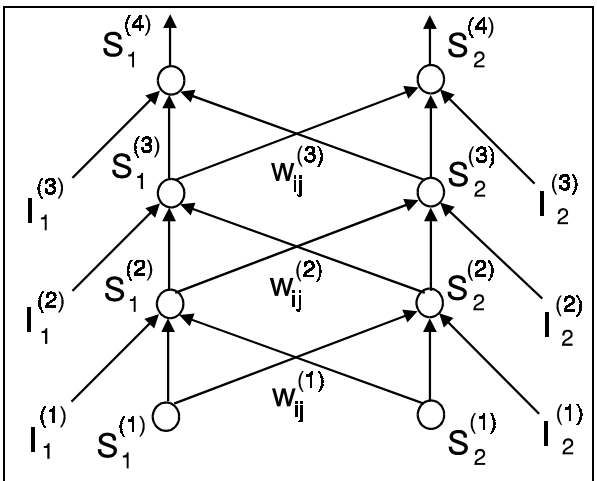
Predstavme si, že na vstup $(I_1^{(t)}, I_2^{(t)})$ privádzame konečné postupnosti (napríklad kódovaných symbolov) a až na konci postupností (teda napríklad na konci slov nad nejakou abecedou) máme k dispozícii učiaci signál, ktorý vypovedá o charaktere práve predvedenej postupnosti (napríklad, či patrí do nejakého jazyka, alebo nie). Treba si uvedomiť, že učiaci signál nie je vo všeobecnosti k dispozícii v každom kroku diskrétneho času (na rozdiel od príkladu uvedenom v predošlej podkapitole). Musíme teda počkať T krokov, zodpovedajúcich dĺžke vstupného reťazca, aby sme získali informáciu o smere v korekcii váh. Uvažujme, že prezentácia prvého vstupu zo vstupného reťazca sa udiala v čase t=1 a prezentácia posledného vstupu vstupného reťazca sa udeje v čase T.

Je možné si predstaviť rekurentnú sieť pracujúcu v T krokoch ako jednoduchú viacvrstvovú sieť, ktorá má 2(T+1) neurónov. V sieti sú kópie rekurentných neurónov s príslušnými prepojeniami pre každý krok diskrétneho času, pričom váhy prepojení sa v časoch $1 \le t \le T$ nemenia. Obr. 6.8 predstavuje rekurentnú sieť rozvinutú v čase $1 \le t \le 4$ pri prezentácii vstupnej postupnosti dĺžky T=3.

Idea rozvinutia rekurentnej siete v čase $1 \le t \le T$ bola pôvodne navrhnutá už Minskym a Papertom [21] a kombinovaná s procedúrou BP je uvedená v [22]. Opäť zdôrazňujeme, že váhy w_{ij} sú nezávislé od času $1 \le t \le T$ a v tomto intervale sa ich hodnoty nemenia. Chybový signál, ktorý sa objaví v čase t=4 (po skončení vstupnej postupnosti dĺžky T=3), necháme spätne sa šíriť cez časovo rozvinutú sieť (obr. 6.8) metódou BP. Algoritmus učenia môžeme vyjadriť pomocou týchto dvoch krokov:

1. Pri počítaní parciálnych derivácií $\partial E / \partial w_{ij}$ považujeme váhy $w_{ij}^{(t)}$ v rôznych časoch t za nezávislé a štandardným postupom spätného chodu získame parciálne derivácie $\partial E / \partial w_{ij}^{(t)}$, pre i,j=1,2 a t=1,2,3.
2. Modifikácia váhy w_{ij} bude priamo úmerná súčtu navrhnutých modifikácií v rôznych krokoch diskrétneho času $\Delta w_{ij} = -\varepsilon \sum_{t=1}^{T} \partial E / \partial w_{ij}^{(t)}$, kde $\varepsilon > 0$ je konštanta nazývaná rýchlosť učenia (angl. *learning rate*), podobne ako v klasickej procedúre BP.

Takáto procedúra trénovania rekurentných sietí založená na modifikácii tradičného prístupu BP v časovo rozvinutej sieti sa nazýva spätné šírenie v čase (angl. *Back Propagation Through Time*, BPTT) [25]. Nevýhodou procedúry BPTT sú veľké pamäťové nároky v prípade rozsiahlejších sietí a dlhších trénovacích postupností (veľké T). Aj keď BPTT sa neujala ako široko používaná trénovacia metóda, Rumelhart, Hinton a Williams [22] ju úspešne použili pre učenie rekurentných sietí trénovaných imitovať správanie sa posuvného registra.



Obrázok 6.8. Časovo rozvinutá sieť s dvoma rekurentnými neurónmi v krokoch t=1,2,3 diskrétneho času.

6.2.4 Rekurentné učenie v reálnom čase

Hlavná myšlienka tohoto prístupu k trénovaniu rekurentných sietí spočíva v úprave váh v *každom* kroku diskrétneho času bez potreby čakania na ukončenie vstupnej trénovacej postupnosti. Týmto sa redukuje problém premenlivej dĺžky postupnosti. Nie je totiž potrebné dopredu určiť maximálne prípustnú dĺžku trénovacej postupnosti a mizne aj potreba alokovania pamäti proporcionálne k dĺžke postupnosti. Autormi tejto metódy známej ako *rekurentné učenie v reálnom čase* (angl. *Real Time Recurrent Learning*, RTRL) sú Williams a Zipser [23].

Uvažujme opäť jednoduchú neurónovú sieť s dvoma neurónmi s dynamikou určenou vzťahom (6.16). Na vstupe siete prezentujeme postupnosť $\left\{\left(I_1^{(t)}, I_2^{(t)}\right)\right\}_{t=1}^{T}$, a nech očakávaný výstup siete po skončení postupnosti (v čase T+1) je (O_1, O_2). Definujme chybové funkcionály

$$E_k^{(t)} = \begin{cases} O_k - S_k^{(T+1)} & \left(\text{pre } t = T+1\right) \\ 0 & \left(\text{pre } \quad 1 \le t \le T\right) \end{cases} \tag{6.17}$$

pre k=1,2. Potom celkový chybový funkcionál je

$$E^{(t)} = \frac{1}{2}\sum_{k=1}^{2}\left(E_k^{(t)}\right)^2 \tag{6.18}$$

Zmena $\Delta w_{ij}(t)$ váhy w_{ij} v čase t bude

$$\Delta w_{ij}(t) = -\varepsilon \frac{\partial E(t)}{\partial w_{ij}} \tag{6.19}$$

kde $\varepsilon > 0$ je rýchlosť učenia. Z (6.19) máme

$$\Delta w_{ij}(t) = \varepsilon \sum_{k=1}^{2} E_k^{(t)} \frac{\partial S_k^{(t)}}{\partial w_{ij}}$$

a podobnou úvahou ako pri zavedení vzťahu (6.13) (tentoraz máme neuróny prvého rádu) dostávame

$$\frac{\partial S_r^{(t)}}{\partial w_{ij}} = g'\left(\phi\left(S_r^{(t)}\right)\right)\left[\delta_{ri} S_j^{(t-1)} + \sum_{a=1}^{2} w_{ra} \frac{\partial S_a^{(t-1)}}{\partial w_{ij}}\right] \tag{6.20}$$

Pripomíname, že podobne ako vo vzťahu (6.13), ϕ je inverzná funkcia k funkcii g a δ_{ri} je Kroneckerovo delta. Celý postup je vhodné inicializovať postulovaním

$$\frac{\partial S_r^{(0)}}{\partial W_{i\,ln}} = 0.$$

Analogicky príkladu z úvodnej podkapitoly aj tu využívame vzťah (6.20) pre postupné prepočítavanie parciálnych derivácií pre budúce kroky. Williams a Zipser [23,24] odporúčajú menšie hodnoty rýchlosti učenia ε. Pochopiteľne, triky známe zo štandardnej procedúry BP pre urýchlenie učenia, či zabránenie uviaznutiu vo veľmi nežiadúcom lokálnom minime možno použiť aj v procedúre RTRL. Napríklad v [13] bol použitý momentový člen pre zvýšenie robustnosti antigradientového poklesu na chybovom povrchu voči slabším lokálnym minimám.

Metóda RTRL našla živnú pôdu medzi užívateľmi rekurentných neurónových sietí. Úspešne bola použitá pri problémoch inferencie konečného akceptora regulárneho jazyka na základe pozitívnych príkladov slov patriacich do jazyka a negatívnych príkladov slov nepatriacich do daného regulárneho jazyka [19]. Použili sme ju aj pre inferenciu Mealyho automatu na základe príkladov jeho činnosti [13,26] a bola použitá aj v prípade inferencie iných automatov rekurentnými sieťami [28,29].

6.3 Na záver

Po úvodnej podkapitole, intuitívne navodivšej potrebu neurónových sietí s vnútornou pamäťou a naznačujúcej spôsob učenia takýchto sietí, sme si v podkapitole 6.2 metodickejším spôsobom predstavili niektoré základné modely rekurentných neurónových sietí a dva najbežnejšie prístupy k ich učeniu.

V poslednom čase je záujem o rekurentné neurónové siete obrovský a s tým súvisí aj explózia literatúry venovanej takýmto sieťam [30]. Určitý podiel na záujme o rekurentné siete má aj existencia výsostne praktických problémov reálneho sveta vykazujúcich časovo-

priestorové štruktúry. Či už je to v oblasti riadenia technologických procesov, robotiky, predikcie odberu elektrickej energie v rozvode generátora, predikcie vývoja na aukčnej burze, atď.

Iste, existujú mnohé iné (napríklad štatistické) metódy pre hľadanie štruktúry v časovej postupnosti a následnom využití vystopovanej štruktúry, napríklad pre predikciu možného budúceho vývoja postupnosti. V mnohých praktických aplikáciách je úspešnosť rekurentných neurónových sietí porovnateľná s úspešnosťou tradične používaných metód. Ako vo všetkých oblastiach modelovania dát, aj tu treba zvoliť rozumný kompromis. Neurónové siete, napríklad, pracujú v testovacom (t.j. pracovnom) móde pomerne rýchlo, pretože väčšina "modelovacej práce" bola presunutá do trénovacej fázy. Na druhej strane, rigorozita dosiahnutých výsledkov je pomerne malá. Pre dosiaľ nevidenú vzorku sieť síce ponúkne odpoveď, avšak bez informácie o miere dôveryhodnosti v ponúknutú odpoveď. V tomto ohľade sú výstupy tradičných štatistických metód rigoróznejšie. Čas potrebný pre získanie odpovede pre každú vzorku však môže byť podstatne väčší ako pri neurónových sieťach. V literatúre sa dajú nájsť pokusy spojiť výhody neurónového a tradične štatistického prístupu k modelovaniu dát [3,31], avšak spravidla musia byť zaplatené väčšou pamäťovou a časovou náročnosťou.

Značné úsilie je venované aj skúmaniu rekurentných neurónových sietí pomocou aparátu teórie dynamických systémov [32]. Rekurentné siete totiž možno považovať za dynamické systémy. Parametrami zobrazenia stavového priestoru sú váhy synaptických prepojení. Keďže premenlivé vonkajšie vstupy sa v tej, či onej miere podieľajú na determinovaní stavového zobrazenia, rekurentná sieť predstavuje neautonómny dynamický systém, vyšetrovanie ktorého je veľmi obtiažne. Väčšina prác je preto venovaná rekurentným sieťam v autonómnom režime (vonkajšie vstupy považujeme za konštantné).

Dvoma najhlavnejšími prúdmi v skúmaní rekurentných sietí ako dynamických systémov sú tieto oblasti:

1) *Popis invariantných atraktívnych množín v stavovom priestore siete.* Invariantné atraktívne množiny sú dôležitým faktorom pri determinovaní asymptotického správania sa rekurentnej siete [33-36]. Zaujímavým výsledkom bol aj experimentálny a analytický dôkaz existencie chaotického režimu v rekurentných sieťach [37].

2) *Interpretácia trénovacieho procesu ako postupnosti bifurkácií vedúcej k indukcii želanej dynamiky siete.* V priebehu trénovania meníme váhy synaptických prepojení (parametre dynamického systému), až pokiaľ dynamické správanie sa siete nezodpovedá želanému stavu. Objasnenie bifurkačného mechanizmu vzniku napríklad nových atraktívnych pevných bodov [26,36], či atraktívnych periodických orbít [39] napomáha pochopeniu trénovacieho procesu.

Viac-menej úspešné modelovanie chaotických časových postupností rekurentnými neurónovými sieťami [41,42] je obmedzené na krátkodobé predpovede budúceho možného vývoja pokračovania danej postupnosti. Hlavným problémom je obrovská citlivosť chaotických trajektórií na malé zmeny v počiatočných podmienkach. Okrem toho, aj keď postupnosť nie je chaotická, ale len dostatočne zložitá, t.j. zahŕňa korelácie medzi časovo značne vzdialenými prvkami postupnosti, učenie rekurentných sietí gradientovými metódami zlyháva. Časovo vzdialenejšie prvky postupnosti totiž majú omnoho menší vplyv

na výsledný gradient ako súčasné prvky postupnosti (ak je stav siete "blízko" nejakej atraktívnej množiny v stavovom priestore siete) [40]. Sieť nie je schopná premietnuť potenciálne dôležitú informáciu o vzťahu minulých vstupov k súčasnému vstupu do primeranej úpravy váh prepojení, umožňujúcej modelovanie takýchto vzťahov.

Isté východisko ponúka napríklad Schmidhuber [44]. Trénuje rekurentnú sieť na pôvodnej postupnosti vstupov. Po dosiahnutí lokálneho minima chybového funkcionálu testuje sieť na pôvodnej postupnosti vstupov. Zaznamenáva vstupy, pri ktorých sa sieť v odpovedi pomýlila a v ďalšom kroku trénuje novú rekurentnú sieť už len na vstupoch, na ktorých predošlá sieť zlyhala (spolu s týmito vstupmi predkladá aj kódovanú informáciu o čase a kontexte v ktorom sa objavili). Tento postup možno rekurentne opakovať a vybudovať hierarchickú štruktúru rekurentných sietí modelujúcu danú "zložitú" vstupnú postupnosť.

Na záver už len spomenieme, že analogicky k teorémam o ľubovoľne dobrej aproximácii spojitých funkcií nad kompaktnou oblasťou (napríklad v L_2 norme) viacvrstvovými neurónovými sieťami [38] možno vysloviť tvrdenia o ľubovoľne presnej aproximácii diskrétneho dynamického systému daného spojitým zobrazením kompaktného stavového priestoru rekurentnými neurónovými sieťami. Ako sme už videli aj v kapitole o viacvrstvových sieťach, teoretická schopnosť systému modelovať určité dáta nemusí mať veľa spoločného s našou schopnosťou nastaviť parametre systému tak, aby dané dáta skutočne dobre modeloval. Navyše, pokrytie stavového priestoru prvkami postupnosti pri veľmi zložitých, "chaotických" postupnostiach nemusí byť "rovnomerné" ako by sa žiadalo pre dobrú aproximáciu stavového zobrazenia, ale je dané invariantnou mierou príslušného generujúceho zobrazenia. Dôsledkom môže byť, že aproximácia stavového zobrazenia nad oblasťami malej miery rekurentnou sieťou bude veľmi nepresná.

Literatúra

[1] B.K. Natarajan. *Machine Learning - A Theoretical Approach*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1991.

[2] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4: 415-447, 1992.

[3] D.J.C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4: 720-736, 1992.

[4] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction To The Theory Of Neural Computation*. Addison-Wesley Publishing Company, Redwood City, California, 1991.

[5] D-T. Lin, J.E. Dayhoff, and P.A. Ligomenides. Trajectory production with the adaptive time-delay neural network. *Neural Networks*, 3: 447-461, 1995.

[6] U. Bodenhausen and A. Waibel. The Tempo2 algorithm: Adjusting time-delays by supervised learning. In: J.E. Moody, R.P. Lippmann, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems*, pp. 155-161, Vol. 3, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1991.

[7] J.L. McClelland and J.L.Elman. Interactive processes in speech perception: The TRACE Model. In: J.L. McClelland, D.E. Rumelhart, and PDP Research Group,

Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 2, chapter 15, MIT Press, Cambridge, 1986.

[8] J.L.Elman and D. Zipser. Learning the hidden structure of speech. *Journal of the Acoustical Society of America*, 83: 1615-1626, 1988.

[9] A. Weibel. Modular construction of time-delay neural networks for speech recognition. *Neural Computation*, 1: 39-46, 1989.

[10] R.P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1: 1-38, 1989.

[11] J.E. Hopfcroft and J.D. Ullman. *Introduction To Automata Theory, Languages and Computation*. Addison-Wesley Publishing Company, Redwood City, California, 1991.

[12] M.W. Shields. *An Introduction To Automata Theory*. Blackwell Scientific Publications, London, UK, 1987.

[13] P. Tiňo and J. Šajda. Learning and extracting initial mealy machines with a modular neural network model. *Neural Computation*, 4: 822-844, 1995.

[14] J.L.Elman. Finding structure in time. *Cognitive Science*, 14: 179-211, 1990.

[15] M.I. Jordan. Serial Order: A parallel distributed processing approach. In: J.L. Elman and D.E. Rumelhart, editors, *Advances in Connectionist Theory: Speech*, Erlbaum, Hillsdale, 1989.

[16] Y. Bengio, R. Cardin and R. De Mori. Speaker independent speech recognition with neural networks and speech knowledge. In: D.S. Touretzky, editor, *Advances in Neural Information Processing Systems* II, pp. 218-225, Vol. 3, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1990.

[17] W.S. Stornetta, T. Hogg, and B.A. Huberman. A dynamical approach to temporal pattern processing. In: D.Z. Anderson, editor, *Neural Information Processing Systems*, pp. 750-759, Vol. 3, American Institute of Physics, New York, 1988.

[18] M.C. Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex Systems*, 3: 349-381, 1989.

[19] C.L. Giles, C.B. Miller, D. Chen, G.Z. Sun, H.H. Chen, and Y.C. Lee. Learning and extracting finite state automata with second order recurrent neural networks. *Neural Computation*, 4: 393-405, 1992.

[20] S.J. Anderson, J.W.L Merrill, and R. Port. Dynamic speech categorization with recurrent networks. In: D.S. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, Pittsburgh, pp. 398-406, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1989.

[21] M.L. Minsky and S.A. Papert. *Perceptrons*. MIT Press, Cambridge, 1969.

[22] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning Internal Representations by Error Propagation. In: J.L. McClelland, D.E. Rumelhart, and PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1: Foundations, chapter 8, MIT Press, Cambridge, 1986.

[23] R.J. Williams and D. Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1: 270-280, 1989.

[24] R.J. Williams and D. Zipser. Experimental Analysis of the Real-Time Recurrent Learning Algorithm. *Connection Science*, 1: 87-111, 1989.

[25] P.J. Werbos. Backpropagation Through Time: What It Is and How to Do It. *Proceedings of the IEEE*, 10: 1550-1560, 1990.

[26] P. Tiňo, B.G. Horne, C.L. Giles, and P.C. Collingwood. Finite State Machines and Recurrent Neural Networks - Automata and Dynamical Systems Approaches. In: J.E. Dayhoff and O. Omnivar, editors, *Progress in Neural Networks*, special volume on "*Temporal Dynamics and Time-Varying Pattern Recognition*", Albex, 1996.

[27] M.P. Casey. *Computation in Discrete-Time Dynamical Systems*. Ph.D. Thesis, University of California, San Diego, Department of Mathematics, 1995.

[28] A. Cleeremans, D. Servan-Schreiber, and J.L. McClelland. Finite State Automata and Simple Recurrent Neural Networks. *Neural Computation*, 3: 372-381, 1989.

[29] Z. Zeng, R.M. Goodman, and P. Smyth. Learning Finite State Machines With Self-Clustering Recurrent Networks. *Neural Computation*, 6: 976-990, 1993.

[30] M.C. Mozer. Neural Net Architectures For Temporal Sequence Processing. In: A. Weigend and N. Gershenfeld, editors, *Predicting the Future and Understanding the Past*, Addison-Wesley Publishing Company, Redwood City, California, 1993.

[31] D.J.C. MacKay. A Practical Bayesian Framework for BackProp Networks. *Neural Computation*, 3: 448-472, 1992.

[32] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*. Springer-Verlag, 1982.

[33] M. Vidyasagar. Location and Stability of the High-Gain Equilibria of Nonlinear Neural Networks. *IEEE Transactions on Neural Networks*, 4: 660-672, 1993.

[34] L. Jin, P.N. Nikiforuk, and M.M. Gupta. Absolute Stability Conditions for Discrete-Time Recurrent Neural Networks. *IEEE Transactions on Neural Networks*, 6: 954-963, 1994.

[35] M.W. Hirsh. Saturation at High Gain in Discrete Time Recurrent Networks. *Neural Networks*, 3: 449-453, 1994.

[36] E.K. Blum and X. Wang. Stability of Fixed Points and Periodic Orbits and Bifurcation in Analog Neural Networks. *Neural Networks*, 5: 577-587, 1992.

[37] X. Wang. Period-Doubling to Chaos in a Simple Neural Network: An Analytical Proof. *Complex Systems*, 5: 425-441, 1991.

[38] K. Hornik, M. Stinchcombe and H. White. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*, 2: 359-366, 1989.

[39] K. Doya. Bifurcation in the Learning of Recurrent Neural Networks. In: *Proceedings of 1992 IEEE International Symposium on Circuits and Systems*, pp. 2777-2780, 1992.

[40] Y. Bengio, P. Simard, and P. Frasconi. Learning Long-Term Dependencies with Gradient Is Difficult. *IEEE Transactions on Neural Networks*, 2: 157-166, 1994.

[41] J.M. Kuo and J.C. Principe. A Systematic Approach to Chaotic Time Series Modeling With Neural Networks. In: *IEEE Workshop on Neural Nets for Signal Processing*, Ermioni, Greece, 1994.

[42] J.C. Principe, A. Rathie, and J.M. Kuo. Prediction of Chaotic Time Series with Neural Networks and the Issue of Dynamic Modeling. *International Journal of Bifurcation and Chaos*, 4: 989-996, 1992.

[43] M. Casdagli. Nonlinear Prediction of Chaotic Time Series. *Physica D*, 35: 335-356, 1989.

[44] P. Schmidhuber. Learning Complex, Extended Sequences Using the Principle of History Compression. Neural *Computation*, 4: 234-242, 1992.

7. Samoorganizujúce sa mapy

7.1 Úvod

V tejto kapitole sa budeme venovať ďalšiemu zo základných modelov umelých neurónových sietí, známemu pod názvom *SamoOrganizujúca sa Mapa* (SOM), ktorej autorom je Teuvo Kohonen [30]. Ako vyplýva už z názvu, SOM patrí do kategórie modelov, ktoré sa učia *bez učiteľa* (samoorganizovane, angl. *unsupervised learning*), t.j. algoritmus učenia nemá informáciu o požadovaných aktivitách výstupných neurónov v priebehu trénovania (ako napr. v algoritme back-propagation, kapitola 5), o ktoré by sa mohol "opierať". Ako v iných algoritmoch samoorganizácie, i tu adaptácia váh odzrkadľuje štatistické vlastnosti trénovacej množiny, ktorá je sieti prezentovaná vo forme vstupných vzorov (vektorov).

Špecifickou črtou SOM je to, že (pri splnení istých podmienok) umožňuje realizovať *zobrazenie zachovávajúce topológiu* a *zobraziť* tak *charakteristické príznaky (črty)* trénovacej množiny dát. Za týmto účelom sú neuróny zoradené v pravidelnej, zväčša dvojrozmernej alebo jednorozmernej štruktúre (mriežka alebo reťaz). Takto uvažované usporiadanie neurónov predstavuje výstupný priestor, v ktorom vzdialenosť dvoch neurónov je obyčajne daná euklidovskou vzdialenosťou vektorov ich súradníc v uvažovanej štruktúre. Zobrazenie zachovávajúce topológiu, ktoré vznikne po natrénovaní SOM, má dôležitú vlastnosť: ľubovoľné dva vzory blízke vo vstupnom priestore evokujú v sieti odozvy na neurónoch, ktoré sú tiež fyzicky blízke (vo výstupnom priestore).

Fenomén topologického zobrazenia príznakov (angl. *feature mapping)* má výrazné zastúpenie v biologických neurónových sieťach, konkrétne v mozgoch vyšších cicavcov i človeka [23]. *Topografické mapy*, ktorých existencia bola zistená v jednotlivých častiach mozgu, hlavne mozgovej kôry, predstavujú efektívny spôsob reprezentácie dôležitých parametrov vstupných dát. Jedná sa o mapy, ktoré buď priamo reprodukujú periférnu reprezentáciu, tzv. projekčné oblasti (napr. mapa povrchu tela), alebo reprezentované parametre sú nejakým spôsobom vypočítavané. Ako príklady možno uviesť vizuálne mapy (napr. mapa orientácie čiarových stimulov), sluchové mapy (napr. mapa frekvencií a amplitúd akustických stimulov), a tiež mapy v motorickej oblasti (napr. riadenie pohybu očí). Komplexnejším príkladom je mapa pozícií zdroja zvuku, ktorá sa "počíta" z jednoduchších sluchových máp. Z toho vyplýva, že mozgová kôra je do značnej miery priestorovo organizovaná a že je pre ňu charakteristická *lokálnosť odoziev* na vstupné podnety.

Ďalšou skutočnosťou potvrdenou experimentálne je fakt, že topografické mapy nie sú úplne vyvinuté už pri narodení, ale formujú sa v počiatočných štádiách vývoja v dôsledku zmyslovej skúsenosti. Inými slovami, hoci usporiadanie jednotlivých častí mozgu a ich funkcie sú dané geneticky, je tu priestor i pre *modifikovateľnosť* týchto štruktúr. Navyše je

evidentné, že proces modifikácie prebieha na základe podnetov prichádzajúcich z okolitého prostredia, a teda z pohľadu spôsobu učenia prebieha samoorganizovane.

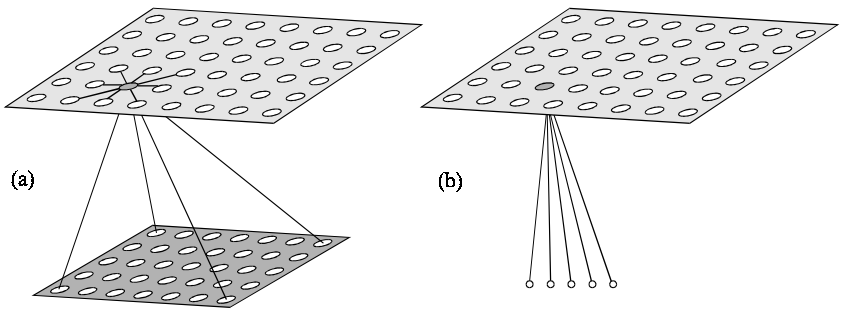
Uvedené fakty boli inšpiračným zdrojom pre snahu simulovať proces samoorganizácie pomocou výpočtového modelu. Navyše, ako sa neskôr ukázalo, model SOM vďaka svojej relatívnej jednoduchosti a ilustratívnosti našiel uplatnenie pri riešení rôznych praktických problémov, kde sa využíva topologické zobrazenie, ako napr. pri rozpoznávaní vzorov (najmä hlások reči), v robotike (transformácia súradníc, generovanie modelu prostredia), kompresii obrazov, riadení procesov v priemysle, optimalizačných úlohách, či spracovaní prirodzeného jazyka. Pomerne rozsiahly zoznam aplikácií možno nájsť v [30].

7.1.1 Prvé biologicky inšpirované modely

Jednými z prvých, ktorí sa zaoberali problémom topologického mapovania pomocou neurónových sietí, boli Willshaw a von der Malsburg [46]. V snahe porozumieť biologicky zaujímavému problému — mechanizmu projekcie zo sietnice na mozgovú kôru (retinotopický problém) — navrhli model neurónovej siete s architektúrou ako vidieť na obr. 7.1a.

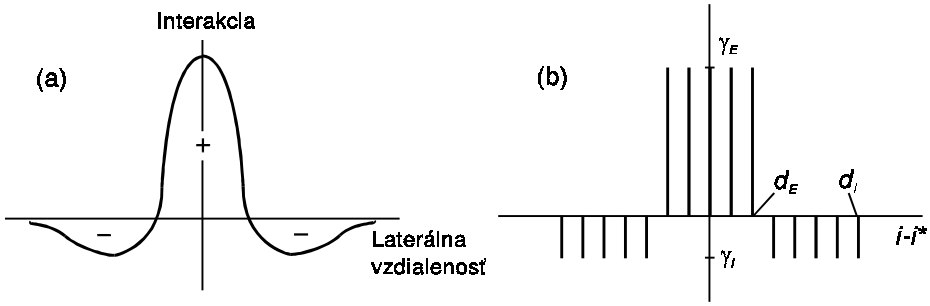
Receptívne bunky spodnej, vstupnej vrstvy (reprezentujúcej sietnicu oka, tzv. retinu) sú spojené s neurónmi mozgovej kôry vo výstupnej vrstve, a to spôsobom každý s každým. Okrem toho, neuróny vo výstupnej vrstve obsahujú nemenné *laterálne (bočné) prepojenia,* ktoré sa vzájomne privádzajú na vstup neurónov v rámci laterálneho dosahu. Sila vplyvu týchto prepojení, reprezentovaná synaptickými váhami, sa so vzdialenosťou od neurónu mení podľa profilu tvaru mexického klobúka. Ako vidieť na obr. 7.2, laterálna interakcia má *excitačný* účinok pre blízke neuróny, a v menšej miere *inhibičný* účinok pre navzájom vzdialenejšie neuróny. Vstup každého neurónu vo výstupnej vrstve je teda súčtom dvoch váhovaných zložiek: doprednej (od neurónov vstupnej vrstvy) a spätnoväzbovej (z výstupov okolitých neurónov v mape, váhovaných podľa spomínaného profilu). Vstupnými vzormi boli rôzne tzv. dipólové stimuli, t.j. dva susedné receptory na retine aktívne a ostatné neaktívne. Učenie bolo založené na štandardnom Hebbovom pravidle[1] s následným normovaním váh. Výsledkom trénovania na dipólových stimuloch bolo získanie projekcie typu pozícia verzus pozícia, s dodržanou vlastnosťou topologického usporiadania odoziev vo výstupnej vrstve.

[1] Hebbovo pravidlo je založené na princípe zvyšovania hodnoty synaptických váh medzi neurónmi, ktoré sú synchrónne aktívne, t.j. $\Delta w_{ij} \propto y_i y_j$ (v spomínanom modeli namiesto vstupného neurónu figuruje receptor). Adaptujú sa všetky spojenia, pričom miera adaptácie teda závisí od korelácie medzi výstupmi oboch neurónov.



Obrázok 7.1. Dva typy architektúry neurónovej siete na topologické zobrazenie príznakov. (a) Biologicky inšpirovaný model, napr. na simulovanie retinotopického problému. Vrstvy sú úplne prepojené, pre názornosť sú zakreslené len niektoré spojenia. (b) Konvenčný model s modifikovanou reprezentáciou vstupov, ktorá vo vzťahu k predchádzajúcemu prípadu môže predstavovať napr. súradnice aktuálneho stimulu (ak použijeme dvojrozmerné vstupy).

Treba spomenúť, že očakávaný efekt topologického usporiadania by sa nebol dostavil, keby



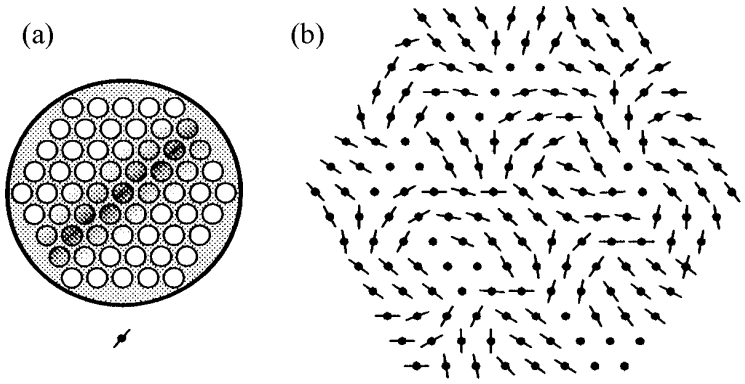
Obrázok 7.2. Profily vplyvu laterálnej interakcie: (a) Typ mexického klobúka. (b) Zjednodušený profil používaný v simuláciách. V oboch prípadoch je daný profil spätnoväzbových prepojení rovnaký pre všetky neuróny, hodnota konkrétnej váhy je funkciou vzájomnej vzdialenosti neurónov i a i^*. d_E, (d_I) označuje dosah laterálnej excitácie (inhibície), γ_E, (γ_I) označuje váhu excitačnej (inhibičnej) synapsy.

bol len jeden receptor naraz aktívny. Simultánna aktivita minimálne dvoch susedných receptorov predstavuje redundanciu a umožňuje sieti zachytiť korelačné, priestorové vzťahy na vstupe, čo je nutnou podmienkou pre dosiahnutie správnej samoorganizácie.

K zaujímavému výsledku sa následne dopracovali Takeuchi a Amari [43], ktorí analyzovali modifikovanú verziu spomínaného modelu v jednorozmernom prípade. Zistili, že k topologicky správnej organizácii na výstupe dochádza za predpokladu, že excitačná časť mexického klobúka je dostatočne široká v porovnaní s veľkosťou (šírkou) lokálnych vstupov. V opačnom prípade je konfigurácia nestabilná a prejavuje sa existenciou tzv. stĺpcových mikroštruktúr.

Projekcia typu pozícia verzus pozícia nie je jediným typom, ktorý bol študovaný. Ešte skôr sa von der Malsburg [32] venoval príbuznému biologicky zameranému problému — simulácii formovania orientačnej selektivity neurónov vo vizuálnej kôre. Na rozdiel od

predchádzajúceho modelu, tu išlo o projekciu typu uhol orientácie versus pozícia, pri ktorej boli stimulom čiary rôznych orientácií vo vstupnej vrstve (t.j. v poli receptorov), prechádzajúce jej stredom (obr. 7.3a). Obr. 7.3b ilustruje situáciu vo výstupnej vrstve po natrénovaní. Je vidieť, že preferovaná orientačná selektivita neurónov v mape sa mení spojite (s občasnými skokmi), čo je pozorované i v biologických sieťach.



Obrázok 7.3. Ilustrácia problému orientačnej selektivity neurónov. (a) Jeden z čiarových stimulov použitých v modeli [32] počas trénovania. (b) Pole orientačne selektívnych neurónov po natrénovaní. Ku každému neurónu je prikreslená čiarka takej orientácie, pri ktorej je daný neurón najviac aktívny. Prevzaté z [10].

7.1.2 Formovanie lokálnych odoziev vplyvom laterálnej spätnej väzby

Je užitočné ozrejmiť si, akú úlohu zohráva laterálna spätná väzba v spomínaných samoorganizujúcich sa modeloch pri formovaní výstupnej aktivity neurónov. Jej vplyv možno najlepšie ilustrovať na príklade. Pre jednoduchosť budeme uvažovať jednorozmerný prípad, t.j. receptorové bunky i neuróny vo výstupnej vrstve zoradené v reťazi. Nech vektor $\boldsymbol{y}$ označuje distribúciu aktivít jednotlivých neurónov vo výstupnej vrstve (reťazi) a zložky vektora $\boldsymbol{net}$ označujú analógie postsynaptických potenciálov. Dynamiku takejto siete možno popísať rovnicou

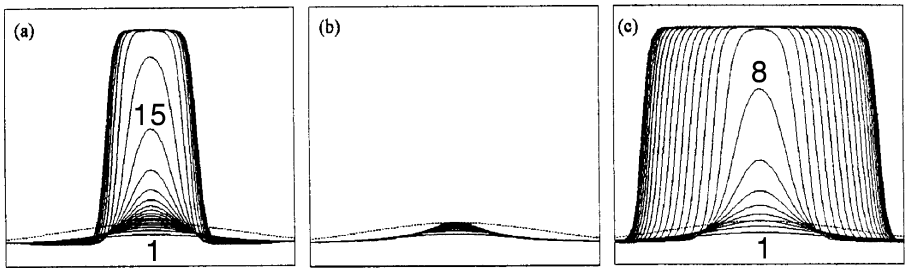
$$\boldsymbol{y}(t+1) = \boldsymbol{S}[\boldsymbol{net}] = \boldsymbol{S}\left[\boldsymbol{z} + \boldsymbol{L} \cdot \boldsymbol{y}(t)\right] \tag{7.1}$$

Na vstup sa teda privedie lokálny stimul $\boldsymbol{x}$ (aktivita receptorového poľa), vypočíta sa počiatočná vnútorná aktivita výstupných neurónov $\boldsymbol{z} = \boldsymbol{W}.\boldsymbol{x}$, kde $\boldsymbol{W}$ je matica váh medzi receptormi a neurónmi, a laterálna spätná väzba sa nechá niekoľko iterácií pôsobiť. Je daná symetrickou maticou $\boldsymbol{L}$, koeficienty ktorej možno jednoducho získať z profilu na obr. 7.2b.[2] Vektorová funkcia $\boldsymbol{S}$, pozostávajúca zo sigmoid, zabraňuje nekonečnému nárastu výstupnej aktivity.

[2] V simuláciách sa pou□íva zjednodušený profil laterálnej spätnej väzby, ktorý má však kvalitatívne rovnaký efekt ako pôvodný "mexický klobúk". Excitačné synaptické váhy sú pri zjednodušení reprezentované kladnou konštantou, inhibičné zápornou.

Vplyv laterálnej väzby vyjadrený vzťahom (7.1) je v konkrétnom prípade ilustrovaný na obr. 7.4a-c. Vo všetkých troch prípadoch bola vstupným stimulom reťaze zloženej zo 100 neurónov aktivita $\boldsymbol{x}$ gaussovského tvaru (znázornená čiarkovane) so zložkami generovanými podľa vzťahu $x(i) = 1 \cdot \exp\left(-(i-50)^2 / 2.30^2\right)$, pre jednoduchosť i=1,2,...,100, t.j. dimenzia vstupu rovná počtu výstupných neurónov. Výstup y každého neurónu bol ohraničený sigmoidálnou funkciou tvaru $y = s(net) = 10/\left(1+\exp\left(-2\cdot(net-2.5)\right)\right)$. V snahe poukázať len na vplyv spätnej väzby možno vzťah (7.1) zjednodušiť uvažovaním $\boldsymbol{W} = \boldsymbol{I}$, t.j. že aktivita receptorov sa priamo prenesie na vstup neurónov. Obrázky 7.4a-c odpovedajú trom kvalitatívne rôznym prípadom vplyvu spätnej väzby (v závislosti od veľkosti laterálnej inhibície) pre tie isté hodnoty parametrov: $d_E = 10,\ d_I = 30, \gamma_E = 0.032$ (podľa obr. 7.2b). V prípade (a) γ_I = 0,009; v prípade (b) 0,013; v prípade (c) 0,004.

Pri správnom pomere excitácie a inhibície (prípad a) má sieť tendenciu vytvárať "zhluk" aktivít na výstupe, ktorý sa vplyvom spätnej väzby zosilňuje. Podobne, v dvojrozmernom prípade možno pozorovať nárast "aktivačnej bubliny", ktorá vzniká na mieste, v ktorom mala počiatočná výstupná aktivita neurónov maximum. Platí, že veľkosť aktivačnej bubliny ("zhluku" aktivít) závisí od "pomeru síl" excitačnej a inhibičnej spätnej väzby: čím väčší vplyv excitácie, tým väčšia bublina, a naopak. Ak je laterálna inhibícia príliš silná, k vytvoreniu bubliny nedôjde a výstupná aktivita sa utlmí (prípad b). Príliš slabá inhibícia zase znamená "predimenzovanie" výstupnej aktivity (prípad c).



Obrázok 7.4. Vplyv laterálnej spätnej väzby na výstupnú aktivitu siete. Číslo nad krivkou označuje iteráciu. (a) Optimálny pomer laterálnej excitácie a inhibície. (b) Laterálna inhibícia je príliš silná. (c) Laterálna inhibícia je príliš slabá.

7.2 Kohonenov algoritmus

Kohonenov model samoorganizujúcej sa mapy, na ktorý sa zameriame, predstavuje výpočtové zjednodušenie modelu Willshawa a von der Malsburga [46], a to v dvoch krokoch. Prvý sa týka náhrady laterálnej interakcie funkciou okolia neurónov, ktorá je zahrnutá v učiacom algoritme: spätnoväzbové spojenia, ktoré sú navyše časovo náročné na simuláciu, v Kohonenovom modeli SOM nejestvujú, zato však každý neurón má

definovaných svojich fyzických susedov. Druhá úprava spočíva v uvažovaní reprezentácie vstupov vo forme *N*-rozmerných vektorov s reálnymi zložkami (obr. 7.1b).

7.2.1 ED verzia algoritmu

Spôsob akým sa zisťuje pozícia primárnej odozvy v SOM na aktuálny podnet sa nazýva *súťaženie.* Výsledkom súťaženia v každom kroku (po predložení konkrétneho vstupu) je *víťazný neurón,* ktorý najviac reaguje na daný vstup $\boldsymbol{x}$. Jednou možnosťou je hľadať maximum výstupu lineárneho neurónu, t.j. $i^* = \arg\max\left(\boldsymbol{w}_i^T \boldsymbol{x}\right)$, kde i^* je index víťazného neurónu. O tejto verzii sa bližšie zmienime v podkapitole 7.5. V základnej, ED (angl. *Euclidean Distance*) verzii algoritmu SOM figuruje iná miera podobnosti: hľadá sa neurón, ktorého váhový vektor je najbližšie k aktuálnemu vstupu v zmysle euklidovskej vzdialenosti:

$$i^* = \arg\min_i \left\| \boldsymbol{x} - \boldsymbol{w}_i \right\| \tag{7.2}$$

Obe miery podobnosti však navzájom súvisia: hľadanie $\max\left(\boldsymbol{w}_i^T \boldsymbol{x}\right)$ odpovedá hľadaniu $\min\left\| \boldsymbol{x} - \boldsymbol{w}_i \right\|$ za predpokladu, že v prvom prípade sú váhové vektory $\boldsymbol{w}_i$ normované (ležia na povrchu hypergule). Vidieť to dobre z rovnosti

$$\left\| \boldsymbol{x} - \boldsymbol{w}_i \right\|^2 = \left\| \boldsymbol{x} \right\|^2 - 2\boldsymbol{w}_i^T x + \left\| \boldsymbol{w}_i \right\|^2 \tag{7.3}$$

Keďže aktuálny vstup je nezávislý od i, a $\left\| \boldsymbol{w}_i \right\|^2$ je konštantné vďaka normovaniu, tak neurón, ktorého váhový vektor je najbližšie k vstupu $\boldsymbol{x}$, je súčasne neurónom, ktorého skalárny súčin $\boldsymbol{w}_i^T \boldsymbol{x}$ je najväčší.

Po nájdení víťaza nasleduje adaptácia váh — *učenie*. To zabezpečí, že váhové vektory víťazného neurónu a jeho topologických susedov sa posunú smerom k aktuálnemu vstupu podľa vzťahu

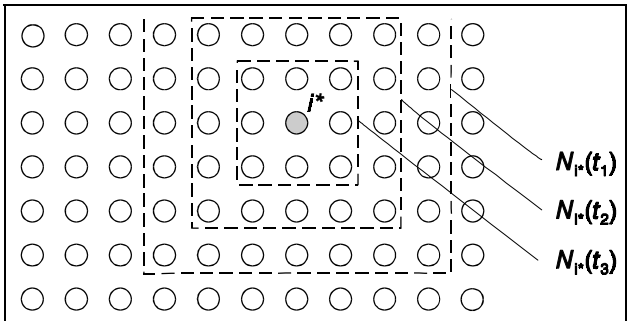
$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t) \cdot h(i^*, i) \cdot \left[\boldsymbol{x}(t) - \boldsymbol{w}_i(t) \right] \tag{7.4}$$

Funkcia $\alpha(t) \in (0,1)$ predstavuje učiaci parameter (rýchlosť učenia), ktorý, podobne ako v iných algoritmoch pre neurónové siete, s časom klesá k nule (napr. podľa vzťahu $1/t$, resp. $\exp(-kt)$), čím sa zabezpečí ukončenie procesu učenia. Funkcia okolia $h(i^*,i)$ (obr. 7.5) definuje rozsah *kooperácie* medzi neurónmi, t.j. koľko váhových vektorov prislúchajúcich neurónom v okolí víťaza bude adaptovaných, a do akej miery.

Najjednoduchšou používanou funkciou je pravouhlé okolie

$$h(i^*, i) = \begin{cases} 1, & \text{ak } d_M(i^*, i) \le \lambda(t) \\ 0, & \text{v ostatných prípadoch} \end{cases} \tag{7.5}$$

pričom $d_M(i^*, i)$ predstavuje vzdialenosť typu "Manhattan" medzi neurónmi i^* a i v mriežke mapy (t.j. sumu absolútnych hodnôt rozdielov ich súradníc). Na základe numerických simulácií dospel Kohonen k záveru, že najlepšie výsledky sa dosiahnu, ak sa veľkosť okolia s časom diskrétne zmenšuje (priemer okolia odpovedá hodnote $2\lambda(t)$). Z obr. 7.5 taktiež vidieť, že v blízkosti okrajov mapy okolie nie je symetrické (týka sa to najmä počiatočných fáz algoritmu, keď polomer okolia je väčší), čo má za následok, ako spomenieme v ďalšom texte, kontrakciu váhových vektorov na okrajoch mapy.



Obrázok 7.5. Príklad okolia víťazného neurónu i^*, definovaného pre rôzne časové okamihy, $t_1 < t_2 < t_3$.

Druhou často používanou voľbou je gaussovské okolie, ktoré možno popísať rovnicou

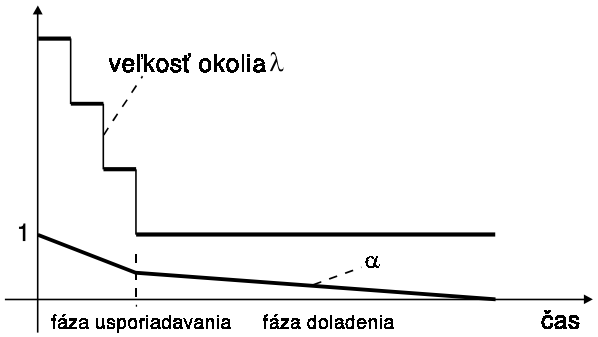
$$h(i^*, i) = \exp\left(-\frac{d_E^2(i^*, i)}{\lambda^2(t)}\right) \tag{7.6}$$

kde $d_E(i^*, i)$ predstavuje euklidovskú vzdialenosť neurónov i^* a i v mriežke, t.j. $d_E(i^*, i) = \|\boldsymbol{r}_{i^*} - \boldsymbol{r}_i\|$, kde $\boldsymbol{r}_i$ označuje vektor súradníc i-teho neurónu v SOM. Parameter λ (t) klesá s časom k nule, čím sa zabezpečuje zmenšovanie okolia počas učenia.

Algoritmus učenia teda spočíva v troch základných krokoch, ktoré sa opakujú po predložení každého vstupného vzoru, vybraného náhodne z trénovacej množiny vzorov (počiatočné hodnoty váh sú náhodné malé čísla):

1) nájdenie víťaza medzi neurónmi (vzťah 7.2),
2) adaptácia váh víťazného neurónu a jeho topologických susedov (vzťah 7.4),

1) aktualizácia parametrov učenia (h, α).



Obrázok 7.6. Príklad vhodnej voľby poklesu polomeru okolia a parametra α počas učenia SOM.

7.2.2 Voľba parametrov učenia

Spôsob, akým je vhodné upravovať veľkosť okolia λ i parametra učenia α je zhruba znázornený na obr. 7.6. Ako vidieť, v procese učenia možno rozlíšiť dve fázy. Počas prvej, nazývanej *fáza usporiadavania*, klesá veľkosť okolia diskrétne s časom. Počas druhej fázy — *fázy doladenia* — možno ponechať najbližších susedov súčasťou okolia, až kým učenie neskončí. Na funkcii poklesu parametra učenia α v praxi až tak veľmi nezáleží (pozri podmienky (7.15) a (7.16)), dôležité je, aby to bola monotónne klesajúca funkcia z nejakej hodnoty blízkej 1, s malými hodnotami (rádovo 0,1-0,01) počas fázy doladenia. Možnou voľbou je napr. lineárna lomená funkcia (obr. 7.6), exponenciálna funkcia atď.

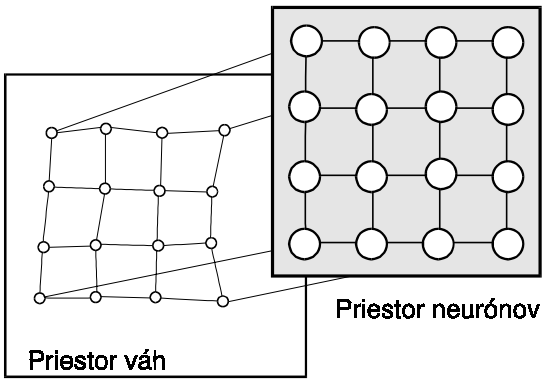
Na presnom počte iterácií takisto nezáleží. Kohonen uvádza empiricky získanú pomôcku, podľa ktorej počet iterácií má byť minimálne 500-násobok počtu neurónov v sieti. Bežne sa počet iterácií pohybuje v rozmedzí rádovo 10000-100000. Dôležité je, aby počas fázy usporiadavania, pokiaľ je ešte parameter α relatívne veľký, sieť "stihla" správne zoradiť svoje váhové vektory, ktoré sa v zvyšnom čase lokálne doladia. Na základe simulácií sa takisto ukázalo, že je vhodné rozdeliť celkovú dobu trénovania tak, že na fázu doladenia sa ponechá viac času ako na prvú fázu.

7.3 Príklady jednoduchých zobrazení

Z matematického hľadiska predstavuje SOM zobrazenie z množiny vstupných vzorov na diskrétnu množinu neurónov, t.j. $\xi : \boldsymbol{X} \subset \Re^N \rightarrow \Im = \{i; 1,2,\ldots,n\}$, kde N je dimenzia vstupov, a n počet neurónov v sieti[3]. Takto máme do činenia s dvoma priestormi —

[3] Ako vidie□, v modeli SOM je výstupná informácia reprezentovaná inak ako v ostatných modeloch ako napr. viacvrstvový perceptrón, èi nejaká lineárna dopredná sie□. Namiesto výstupnej aktivity siete tu nás zaujíma len pozícia ví□azného neurónu. V snahe nazrie□ na SOM ako na systém realizujúci transformáciu φ:X→Y i tu je

priestorom vstupov $\boldsymbol{X}$, ktorý obyčajne predstavuje časť euklidovského priestoru, a priestorom neurónov, ktorý je definovaný topológiou ich usporiadania (mriežka, reťaz). Keďže dimenzia váh je zhodná s dimenziou vstupov (ako vyplýva i z rovnice 7.4), možno vstupný priestor chápať súčasne ako priestor váh. Stav SOM sa vyjadruje váhovými vektormi a tie možno zobraziť ako body v priestore váh. Situáciu znázorňuje obr. 7.7.

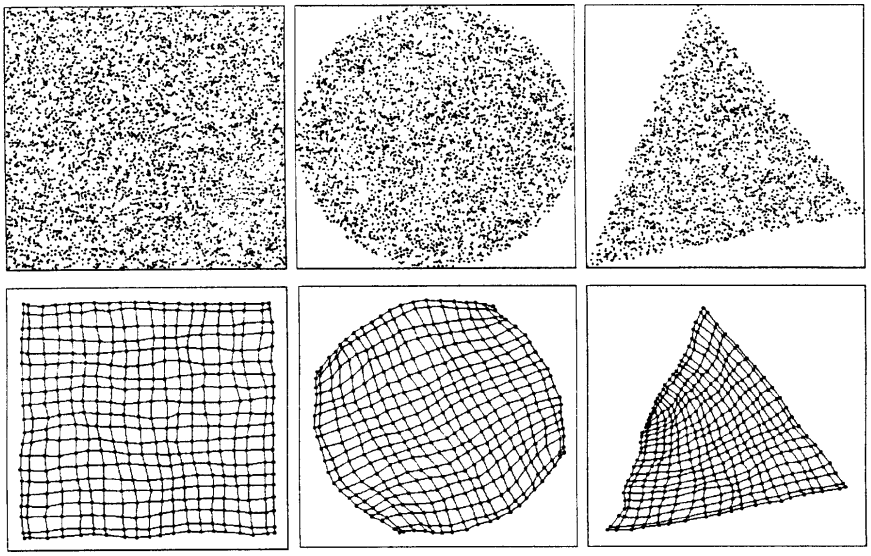


Obrázok 7.7. Princíp reprezentácie stavu siete použitý v modeli SOM. Váhové vektory — body v priestore váh — sú vzájomne pospájané za účelom znázornenia stavu ich usporiadanosti.

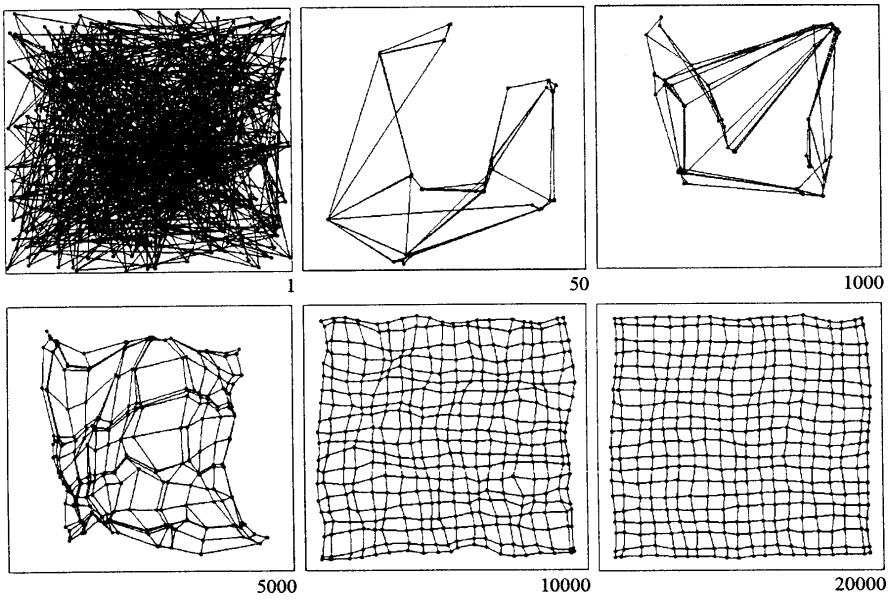
Spájajúce čiary medzi váhovými vektormi znázorňujú skutočnosť, že každé dva spojené váhové vektory prináležia dvojici neurónov, ktoré sú bezprostrednými susedmi v mriežke. Pri pohľade na zobrazené váhy takto dostávame vizuálnu informáciu o ich rozmiestnení i vzájomnom usporiadaní. Podobne vo výstupnom priestore, neuróny nie sú fyzicky spojené, spojnice len znázorňujú ich vzájomné topologické vzťahy, s ktorými súvisí definované okolie.

---

možné vypoèítava výstupnú aktivitu *i*-tého neurónu v tvare $y_i = f(\|\boldsymbol{x} - \boldsymbol{w}_i\|)$, kde f je monotónne klesajúca funkcia. Tento alternatívny prístup, v ktorom je výstupná informácia SOM reprezentovaná celkovou aktivitou všetkých neurónov, t.j. vektorom $\boldsymbol{y}$ (a teda nie pozíciou víaza i^*) je výpoètovo nároènejší, ale biologicky prijatelnejší [17].



Obrázok 7.8. Príklady jednoduchých množín dát s rovnomerným rozdelením v rovine (horná trojica obrázkov) a stavy SOM (dolná trojica obrázkov) zobrazené v priestore váh po natrénovaní (20000 iterácií). Ako vidieť, SOM aproximuje funkciu hustoty vstupných dát.

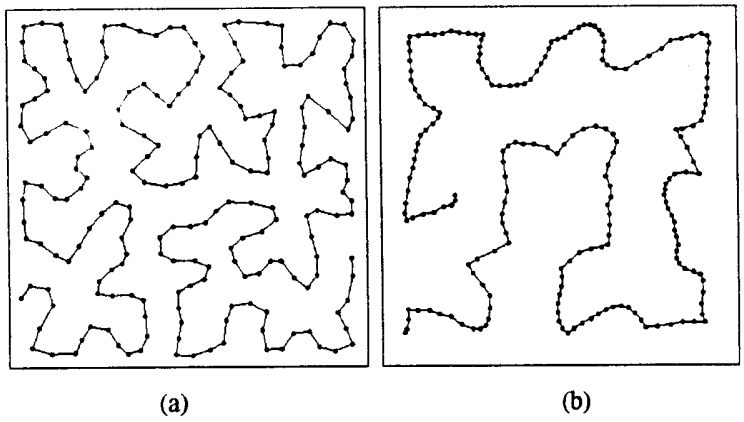


Obrázok 7.9. Usporiadavanie váhových vektorov počas trénovania na dátach s rovnomerným rozdelením. Na prvých troch záberoch vidieť, že SOM sa nachádza vo fáze usporiadavania, zatiaľ čo ostatné tri zábery už odpovedajú fáze dolaďovania.

Ak budeme pre začiatok uvažovať, že máme dvojrozmerné vstupy $\boldsymbol{x} = [x_1, x_2]^T$, môžeme priestor váh priamo zobraziť v rovine. Na obr. 7.8 sú príklady portrétov váh SOM (dolná trojica obrázkov) získaných po natrénovaní na dátach s rovnomerným rozdelením (horná trojica obrázkov). V každom z príkladov boli počas trénovania jednotlivé vstupné vzory vyberané náhodne z trénovacej množiny. Je vidieť, že váhy majú tendenciu aproximovať funkciu hustoty trénovacej množiny.

Na sekvencii obrázkov 7.9 je ilustrovaný proces usporiadavania a dolaďovania váhových vektorov SOM trénovanej na "štvorcových" dátach. Je vidieť, že už približne po štvrtine celkového počtu iterácií (20000) sieť prechádza do fázy dolaďovania.

Ďalším zaujímavým fenoménom, ktorý je možné u SOM ilustrovať, je rozdielny vplyv pravouhlého (rovnica 7.5) a gaussovského (7.6) okolia, parametrizovaných hodnotou λ.



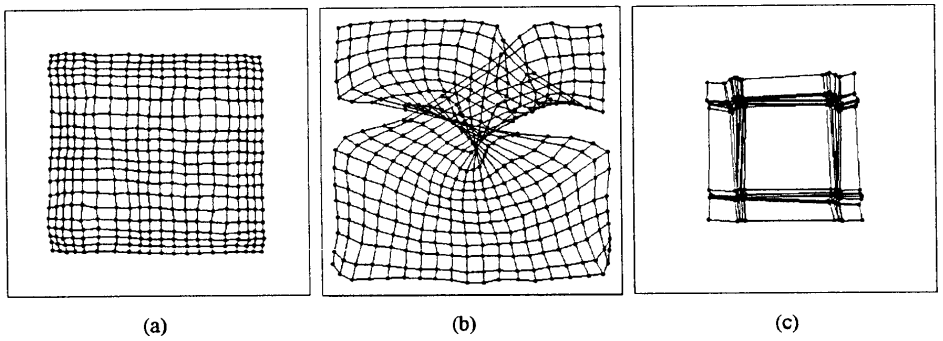
Obrázok 7.10. Rozdiel vplyvu (a) pravouhlého a (b) gaussovského okolia na výsledné usporiadanie váh.

Väčší rozsah gaussovského okolia má za následok vznik "stiahnutejšej" reprezentácie pri rovnakom počte neurónov (obr. 7.10).

7.3.1 Niektoré špeciálne efekty

I keď proces samoorganizácie v SOM prebehne pri pomerne širokom rozpätí jednotlivých voliteľných parametrov, existujú podmienky, pri ktorých proces zlyhá. Na obr. 7.11a-c sú príklady nežiaducich efektov, ktoré môžu vzniknúť. Všetky 3 simulácie boli nastavené na 20000 iterácií na dátach s rovnomerným rozdelením, pri použití siete 20×20 neurónov.
Efekt na obr. 7.11a vznikol v dôsledku príliš rýchleho poklesu parametra α (pomer dĺžok prvej a druhej fázy učenia bol 1:500), pri pravouhlom okolí so štandardným poklesom polomeru okolia λ. Na obr. 7.11b je tzv. "motýlí" efekt (dokonca kvázi dvojitý)[4], ktorý

[4] Použitý termín nemá nijaký súvis s rovnomenným termínom používaným v teórii chaosu.



Obrázok 7.11. Niektoré špeciálne efekty, ktoré môžu vzniknúť pri trénovaní SOM: (a) Neúplné rozvinutie siete v dôsledku rýchleho poklesu α. (b) Tzv. "motýlí" efekt spôsobený rýchlym poklesom α v porovnaní s poklesom λ. (c) Tzv. "pinch" efekt vznikajúci pri príliš pomalom poklese λ.

môže vzniknúť, ak necháme príliš rýchlo klesať λ v porovnaní s α (v simulácii bol pokles λ na hodnotu 1 pri gaussovskom okolí už po prvej stotine iterácií, zatiaľ čo α klesalo štandardne). Obr. 7.11c ilustruje tzv. "pinch" efekt, ktorý bol docielený vďaka pomalému poklesu λ (v simulácii mal parameter λ hodnotu 10 počas prvej polovice iterácií, potom jeho hodnota poklesla na 9).

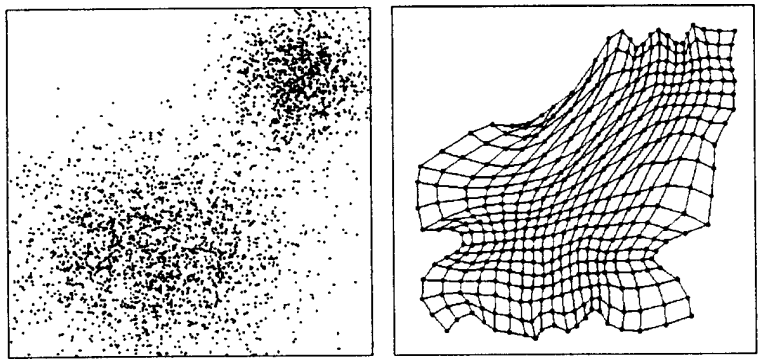
7.3.2 Hraničný efekt

Pri pohľade na výsledné stavy sietí na obr. 7.8 vidieť, že ich okrajové časti sú mierne kontrahované smerom dovnútra. Tento okrajový defekt je dôsledkom asymetrie okolia (obr. 7.5), ktorá spôsobuje, že váhové vektory okrajových neurónov sú štatisticky v priemere viac adaptované smerom dovnútra siete. Ak je tento efekt nežiaduci, možno ho odstrániť tak, že sa okolie algoritmicky uzavrie do slučky. Potom nastane situácia, že všetky neuróny budú pozične ekvivalentné: v prípade reťaze bude mať *každý* neurón dvoch bezprostredných susedov, v prípade mriežky ôsmich. Podobne sa hraničný efekt prejavuje i v modeli s laterálnou inhibíciou, kde to možno obísť analogicky — uzavretím spätnej väzby do slučky.

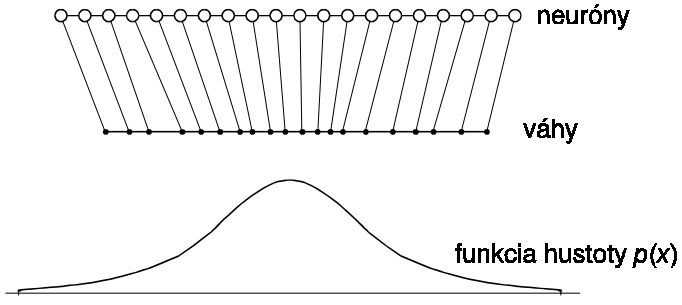
7.3.3 Magnifikačný faktor

Na základe simulácií na dátach s rovnomerným rozdelením bolo vidieť, že váhové vektory mali tendenciu rovnomerne pokryť "trénovaciu oblasť" (obr. 7.8). Inými slovami, SOM sa snažila aproximovať funkciu hustoty vstupných dát. Tento efekt sa prejavuje i v prípade nerovnomernej distribúcie vstupných dát, čoho výsledkom je odpovedajúce rozloženie váhových vektorov. Na obr. 7.12 je znázornená situácia po natrénovaní siete 20×20 neurónov na 2D dátach zložených z dvoch gaussovských rozdelení. Ako vidieť, v oblastiach centier zhlukov je zhustené rozloženie váhových vektorov siete. Takúto tendenciu algoritmu SOM možno interpretovať ako snahu o optimálne rozloženie svojich zdrojov.

Túto vlastnosť algoritmu popisuje *magnifikačný faktor* (počet váhových vektorov pripadajúcich na jednotkovú plochu vstupného priestoru), ktorý nie je v tomto prípade konštantný, ale je funkciou pozície v mape. V jednorozmernom prípade sa dá povedať, že je



Obrázok 7.12. Ilustrácia aproximácie funkcie hustoty dát s gaussovským rozdelením v rovine pomocou SOM (20000 iterácií).



Obrázok 7.13. Ilustrácia aproximácie funkcie hustoty jednoroz-merných dát s gaussovským rozdelením pomocou SOM.

zhruba úmerný funkcii hustoty dát *p(x)*, ako vidieť na obr. 7.13. Tým, že si sieť "vyhradí" proporcionálne väčší počet neurónov na reprezentáciu viac zahustených oblastí, tým je jej citlivosť na zmenu *dx* (zmena víťazného neurónu pri malej zmene *x*) v týchto miestach väčšia, a naopak.[5]

7.4 Teoretická analýza algoritmu SOM

Kohonenov algoritmus v sebe skrýva mnoho otázok, na ktoré je vo všeobecnosti ťažké nájsť odpoveď. Problémy ako: Konverguje algoritmus vždy? Nevystupuje tu problém lokálnych miním? Koľko iterácií treba na konvergenciu? Existujú optimálne hodnoty pre voľbu parametra α a funkcie okolia $h(i, i^*)$? atď., sú analyzovateľné len v špeciálnych, jednoduchších prípadoch, a preto sa väčšina doterajších teoretických prác o SOM zameriavala na jednorozmerný prípad, t.j. jednorozmernú SOM s jednorozmernými vstupmi. V zložitejších prípadoch je situácia sťažená skutočnosťou, že [8] (a) sa nepodarilo definovať, čo je to korektne usporiadaná konfigurácia váh v prípade dimenzie vyššej ako jedna, a (b) Erwin a spol. dokázali, že v prípade spojitých dát nemožno vo všeobecnosti nájsť globálnu kriteriálnu (účelovú) funkciu, z ktorej by sa dal odvodiť Kohonenov algoritmus [15].

I v jednoduchších prípadoch vystupuje niekoľko matematických problémov, ktoré súvisia s procesom samoorganizácie. Sú to: (a) vlastnosť *vektorovej kvantizácie*, (b) hľadanie *kriteriálnej funkcie* minimalizovanej algoritmom SOM, (c) *usporiadavanie váh* a (d) *konvergencia váh* SOM. O každom probléme sa stručne zmienime v ďalšom texte.

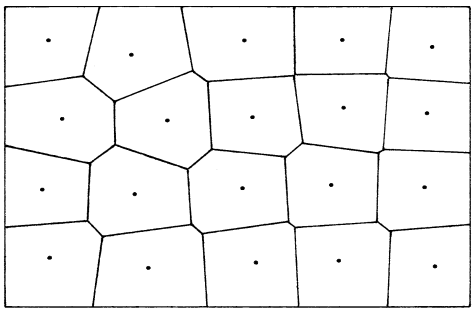
7.4.1 Vektorová kvantizácia

Ak si na moment odmyslíme špecifickú črtu SOM — usporiadanie neurónov v pravidelnej štruktúre a s tým súvisiacu schopnosť zachovania topológie — môžeme sa na SOM pozrieť ako na vektorový kvantifikátor.

Pri probléme vektorovej kvantifikácie [21] je úlohou nahradiť danú množinu alebo distribúciu vstupných dát $\boldsymbol{X}$ početne menšou množinou referenčných vektorov, nazývaných *prototypy*. Účel takejto náhrady sa uplatňuje napr. pri prenose údajov v telekomunikáciách či v kompresii dát (obrazových, rečových), keď sa dosiahne výrazné zníženie objemu dát pri minimálnom poklese kvality. Vďaka vektorovej kvantizácii stačí namiesto celej množiny $\boldsymbol{X}$ preniesť (či uchovať) len množinu prototypov spolu s informáciou (index prototypu) o príslušnosti každého vstupného vektora ku tomu-ktorému prototypu (na základe podobnosti v zmysle euklidovskej vzdialenosti). Vektorovou kvantizáciou sa priestor $\boldsymbol{X}$ rozdelí na disjunktné oblasti, ktoré tvoria tzv. *Voronoiho mozaiky* (obr. 7.14).

---

[5] V biologických sieťach predstavuje tento efekt dôle□itú vlastnosť. Prejavom takejto reprezentácie napr. v somatosenzorickej kôre je skutočnosť, □e organizmus má na svojom povrchu fyzicky viac i menej citlivé miesta.



Obrázok 7.14. Voronoiho mozaiky. Priestor je rozdelený na polyedrálne oblasti určujúce "sféry vplyvu" jednotlivých prototypov (ozn. ako body) podľa minima euklidovskej vzdialenosti. Hranice medzi oblasťami sú tvorené hyperrovinami (na obr. úsečkami), ktoré prechádzajú stredmi spojníc susedných prototypov a sú na ne kolmé.

Kritériom vektorovej kvantizácie je nájdenie optimálnych pozícií týchto prototypov $\{\boldsymbol{w}_i : i = 1,2,\ldots,n\}$ (počet ktorých sa obyčajne vopred stanoví), a to tak, aby sa minimalizovala stredná kvadratická odchýlka medzi vstupom $\boldsymbol{x}$ a jeho prototypom $\boldsymbol{w}_c$, pričom $c = \arg\min\|\boldsymbol{x} - \boldsymbol{w}_i\|$. V prípade spojitého rozdelenia dát daného funkciou hustoty *p(x)* udáva strednú kvadratickú odchýlku (nazývanú tiež chybou rekonštrukcie) vzťah

$$E = \int_{\boldsymbol{x}\in\boldsymbol{X}} \|\boldsymbol{x} - \boldsymbol{w}_c\|^2 . p(\boldsymbol{x})\, d\boldsymbol{x} \tag{7.7}$$

V diskrétnom prípade, ak vstupná množina pozostáva z *M* prvkov, má chyba rekonštrukcie tvar

$$E = \frac{1}{M}\sum_j \| \boldsymbol{x}_j - \boldsymbol{w}_c \|^2 \tag{7.8}$$

Pri uvažovaní, že prototypy odpovedajú váhovým vektorom, predstavuje algoritmus SOM štandardný vektorový kvantifikátor [36], ktorý minimalizuje funkcionál (7.7), resp. (7.8). Súčasne je však podmienkou $\lambda = 0$, t.j. v každej iterácii sa adaptuje len váhový vektor víťazného neurónu (učenie typu "víťaz berie všetko", angl. *winner-takes-all*).

Za účelom vektorovej kvantizácie boli použité i zložitejšie algoritmy so súťažením, pracujúce na princípe "víťaz berie väčšinu" (angl. *winner-takes-most*) [1], jedným z ktorých je aj Kohonenov algoritmus. Tieto zložitejšie algoritmy sú vďaka kooperácii medzi neurónmi schopné eliminovať nežiaduce efekty vznikajúce pri "winner-takes-all" prístupe, ako napr. existencia "mŕtvych" neurónov (t.j. takých, ktoré sú zo súťaže vylúčené a prestanú sa učiť). Súčasne však väčšia zložitosť týchto algoritmov má za následok, že sú vo všeobecnosti pritvrdým orieškom pre teoretickú analýzu.

7.4.2 Kriteriálne funkcie

Ako v iných algoritmoch, i pri SOM vyplýva snaha nájsť kriteriálnu funkciu minimalizovanú algoritmom adaptácie synaptických váh z toho, že taká funkcia by umožnila zjednodušiť teoretickú analýzu. V prípade Kohonenovho algoritmu bolo dokázané, že vo všeobecnosti pri spojitej množine $\boldsymbol{X}$ taká globálna kriteriálna funkcia neexistuje [15]. Väčšina prác sa preto zameriava na jednorozmerný prípad.

V špeciálnom prípade s nulovým okolím bolo ukázané, že SOM funguje ako vektorový kvantifikátor, ktorý v spojitom prípade minimalizuje funkcionál (7.7), resp. v diskrétnom prípade (7.8). Pri nenulovom okolí je situácia zložitejšia. Kohonen [27] analyzoval spojitý prípad, pre ktorý navrhol funkcionál

$$E(w) = \sum_i \int_{x \in \Omega_i} \sum_k h(i,k) \| x - w_k \|^2 p(x)\, dx \tag{7.9}$$

Pri odvodzovaní učiaceho pravidla (vzťah 7.4) ukázal, že vypočítaný gradient ∇E pozostáva z dvoch členov: prvý člen predstavuje zmenu váh vnútri Voronoiho buniek Ω_i , druhý vyjadruje posun hraníc medzi nimi.

Erwin a spol. [15] navrhli namiesto jednej kriteriálnej funkcie systém takýchto funkcií — pre každý neurón jednu. Odvodzovanie adaptačného pravidla potom spočívalo v minimalizácii každej z týchto funkcií zvlášť. Podobne ako u Kohonena [27], i tu pozostávala každá kriteriálna funkcia z dvoch členov: prvý predstavoval chybu danú pozíciou váh, druhý vyjadroval zmenu hodnoty sumárnej kriteriálnej funkcie pri posune hraníc medzi Voronoiho bunkami. Súčasne bolo pozorované, že v prvej fáze učenia, keď váhy ešte nie sú usporiadané, má väčšiu hodnotu druhý člen, zatiaľ čo vo fáze doladenia zase dominuje prvý člen.

Kohonen [27] študoval i diskrétny prípad (množina vstupov $x_1, x_2, ..., x_N$), v ktorom navrhol kriteriálnu funkciu tvaru

$$E(w) = \frac{1}{N} \sum_{j=1}^{N} \sum_i h(i^*, i) \left\| x_j - w_i \right\|^2 \tag{7.10}$$

kde N udáva počet vstupov. Minimalizáciou funkcie (7.10) dospel k štandardnému tvaru učiaceho pravidla pre SOM (vzťah 7.4).

O niečo skôr sa podobným prístupom zaoberali Ritter a spol. [38], ktorí navrhli funkcionál

$$E(w) = \frac{1}{2}\sum_{i^* i} h(i^*, i) \sum_{x_j \in \Omega_{i^*}} p_j \left(x_j - w_i\right)^2 \tag{7.11}$$

kde p_j je pravdepodobnosť, že zvolený vstup bude mať hodnotu x_j. Ukázali, že učiace pravidlo pre SOM minimalizuje tento funkcionál.

Snáď najúplnejšie výsledky možno nájsť v práci Růžičku [41], kde bolo dokázané, že pre sieť pozostávajúcu z n neurónov, s N vstupmi s rovnomerným rozdelením možno odvodiť Kohonenov algoritmus minimalizáciou funkcionálu

$$E(w) = \frac{1}{N}\sum_{i=1}^{n} \sum_{x_j \in \Omega_i} \sum_{m=1}^{n} h(i,m).\left\| x_j - w_m \right\|^2 \tag{7.12}$$

7.4.3 Usporiadavanie váh

Dôkazy usporiadavania váh (samoorganizácie) existujú pre jednorozmerný prípad, pre ktorý je ľahké definovať stav usporiadania. Správne usporiadanie predpokladá, že v reťazi neexistujú inverzie (nesprávne zoradené trojice susedných váh), t.j. váhy spĺňajú jednu z podmienok $w_1 < w_2 < ... < w_n$ (vzostupné usporiadanie) alebo $w_n < w_{n-1} < ... < w_1$ (zostupné usporiadanie).

Jednou z možností je definovať nejaký parameter, ktorý kvantifikuje mieru usporiadania. Kohonen [25] definoval pre reťaz n neurónov a prípad jednorozmerných vstupov tzv. index usporiadania

$$\mathrm{D} = \sum_{i=2}^{n} \left|w_i - w_{i-1}\right| - \left|w_n - w_1\right| \tag{7.13}$$

a dokázal, že pri náhodnom výbere vstupov x z uvažovanej množiny tento index s pravdepodobnosťou jedna konverguje k nule (takmer istá konvergencia). Obdobný prístup je použitý v práci [18].

Stanovenie podmienky pre pokles indexu D v prípade jednorozmernej SOM je značne zjednodušené v práci [7], pričom odvodená podmienka je aplikovateľná aj pre viacrozmerné vstupy. Navyše, ako zdôrazňujú autori, ich prístup vnáša geometrický vhľad do problému usporiadavania váh a intuitívne vysvetľuje charakter problému, ktorý vzniká pri snahe zovšeobecniť závery pre viacrozmernú SOM.

Pri alternatívnom prístupe sa usporiadavanie váh chápe ako Markovov náhodný proces, pričom konfigurácia váh predstavuje stav procesu. Cieľom je nájsť tzv. absorbčnú množinu (ktorá predstavuje usporiadanú konfiguráciu váh) a dokázať, že existuje sekvencia vstupov (vyberaných náhodne, s nenulovou pravdepodobnosťou), ktorá s pravdepodobnosťou jedna zabezpečí vniknutie do tejto množiny v konečnom čase. Dôkazy sú uvedené napr. v [4, 15].

7.4.4 Konvergencia váh

Problém konvergencie súvisí s usporiadavaním váh, avšak líši sa v tom, že stav po skonvergovaní nemusí byť automaticky usporiadaným stavom (napr. v prípade lokálnych miním kriteriálnej funkcie, ktoré existujú pri nevhodnej funkcii okolia [16]).

Jednorozmerný prípad bol analyzovaný v prácach [24, 25]. Vychádzajúc z predpokladov, že vstupné dáta majú konštantnú funkciu hustoty, že konštantné okolie zahŕňa len najbližších susedov a že počiatočný stav je usporiadaným stavom, Kohonen ukázal, že zo všetkých takýchto počiatočných stavov konvergujú váhy s pravdepodobnosťou jedna do toho istého konečného stavu. Pri tých istých podmienkach dospeli k podobnému záveru i Bouton a spol. [4]. Všeobecnejšia funkcia okolia bola aplikovaná v práci [31], pričom výsledky boli rozšírené i pre dvojrozmerný prípad (mapa s dvojrozmernými vstupmi).

V prípade nerovnomernej distribúcie dát zohráva úlohu lokálny magnifikačný faktor *M(x),* ktorý kvantifikuje, ako váhy aproximujú funkciu hustoty vstupných dát *P(x).* Výsledkom analýzy takéhoto všeobecného prípadu [39] je vzťah

$$M(x) \propto P(x)^{2/3} \tag{7.14}$$

ktorý možno interpretovať tak, že SOM má tendenciu podhusťovať váhami oblasti s vysokou pravdepodobnosťou výskytu a prehusťovať "riedke" oblasti. Treba pripomenúť, že odvodený vzťah sa vzťahuje len na dva špeciálne prípady: jednorozmerný prípad (jednorozmerná SOM, skalárne vstupy) a dvojrozmerný prípad (dvojrozmerná SOM, dvojrozmerné vstupy), v ktorom však funkcia hustoty vstupov sa dá vyjadriť ako súčin dvoch jednorozmerných funkcií hustôt.

V práci [38] bola študovaná konvergencia viacrozmernej SOM a bola zovšeobecnená na prípady, keď sa dimenzia vstupu nezhoduje s dimenziou siete. Ritter a Schulten odvodili nutné a postačujúce podmienky pre parameter učenia α, ktoré zaručujú konvergenciu do ustáleného stavu (t je diskrétny čas):

$$\sum_{t=0}^{\infty} \alpha(t) = \infty \tag{7.15}$$

$$\lim_{t \to \infty} \alpha(t) = 0 \tag{7.16}$$

V práci bola študovaná i stabilita takéhoto stavu v procese zmeny distribúcie vstupných dát. Po natrénovaní mapy na dátach s dominantnou disperziou v dvoch smeroch (pozdĺž ktorých bola mapa rozvinutá), autori sledovali vplyv fluktuácií pri zväčšovaní disperzie v treťom smere, kolmom na dominantné dva. Na tomto príklade ilustrovali tendenciu SOM reorganizovať sa v snahe aproximovať novú funkciu hustoty, ak bola prekročená nejaká kritická hodnota disperzie v treťom smere. Tento fenomén označil Kohonen ako *automatický výber dimenzií príznakov* [25] (pozri podkapitolu 7.6.1).

Doteraz najuniverzálnejší výsledok zrejme poskytuje práca Yina a Allinsona [47], ktorí pri analýze uvažovali všeobecný prípad viacrozmernej SOM s viacrozmernými vstupmi. Funkcia okolia bola taktiež všeobecná — ľubovoľná monotónne klesajúca funkcia centrovaná okolo víťazného neurónu. Autori pre takýto všeobecný prípad formálne dokázali, že hodnoty váh SOM konvergujú do pozícií, ktoré určujú finálny efekt aproximácie funkcie hustoty vstupných dát.

V niektorých prácach sa autori venovali problému, ako optimálne modifikovať veľkosť okolia λ a parametra učenia α tak, aby sa urýchlila konvergencia procesu [9, 14]. Navrhnuté

riešenia porovnávali so štandardným prístupom (t.j. s vopred zvolenými funkciami okolia *h* i α a ich *ad hoc* aktualizáciou počas učenia), pričom výsledky boli lepšie v nimi navrhnutých riešeniach.

7.5 DP verzia Kohonenovho algoritmu

Ako bolo spomenuté v podkapitole 7.2.1, víťazný neurón možno hľadať i na základe inej miery podobnosti, a to skalárneho súčinu

$$d_i = \boldsymbol{w}_i^T . \boldsymbol{x} = \|\boldsymbol{w}_i\| . \|\boldsymbol{x}\| . \cos\varphi_i \tag{7.17}$$

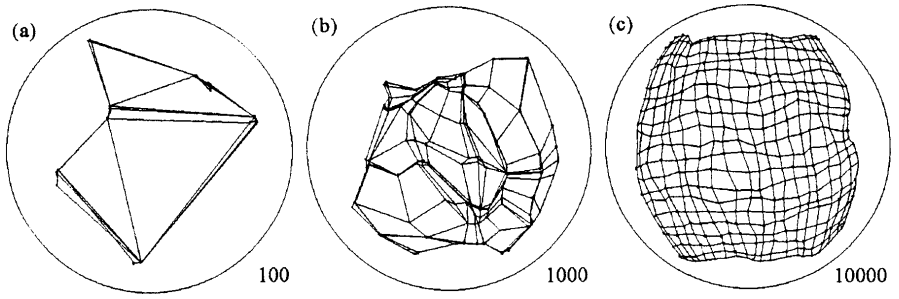
Takto dostávame DP (angl. *Dot Product*) verziu algoritmu SOM (čo je v skutočnosti prvotná verzia algoritmu [26], z ktorej potom prešiel Kohonen k ED verzii). Učiace pravidlo tu spočíva v čiastočnej rotácii váhového vektora víťazného neurónu a jeho topologických susedov k vstupnému vektoru $\boldsymbol{x}$ (jedná sa v podstate o učenie Hebbovho typu s následným normovaním váh), a to podľa vzťahu

$$\boldsymbol{w}_i(t+1) = \frac{\boldsymbol{w}_i(t) + \alpha(t)\, h(i^*, i)\, \boldsymbol{x}}{\left\| \boldsymbol{w}_i(t) + \alpha(t)\, h(i^*, i)\, \boldsymbol{x} \right\|} \tag{7.18}$$

pričom opäť možno použiť okolie pravouhlé (rovnica 7.5) alebo gaussovské (rovnica 7.6). Parameter $\alpha(t)$ predstavuje rýchlosť učenia, podobne ako v štandardnej verzii algoritmu, počiatočné hodnoty by v tomto prípade však mali byť >>1, aby sa urýchlil proces konvergencie. Normovanie váh je potrebné na to, aby sa zamedzil ich nekonečný rast počas učenia. Súčasne sa tým dosiahne efekt výberu víťaza na základe jediného parametra, a tým je uhol medzi váhovým vektorom a aktuálnym vstupným vektorom (ako je zrejmé z rovnice 7.17). Tým, že váhy sú normované, ležia na povrchu hypergule s dimenziou *N-1*. V princípe sa ponúkajú dve možnosti:

(a) Priamym aplikovaním DP verzie algoritmu SOM dostaneme aproximáciu vstupného priestoru *invariantnú od normy vstupných vektorov,* t.j. všetky vektory s tou istou orientáciou budú zobrazované na ten istý váhový vektor, ktorý s nimi zviera najmenší uhol. Takto sa napr. trojrozmerné vstupy premietnu na povrch gule, čo je dvojrozmerná plocha.

(b) Ak chceme aproximovať vstupný priestor podobne ako so štandardnou, ED verziou Kohonenovho algoritmu, je potrebné najprv urobiť transformáciu vstupných dát do priestoru s dimenziou o jednu vyššou. Takýmto spôsobom napr. dvojrozmernú plochu tvorenú vektormi $\boldsymbol{x} = [x_1, x_2]^T$ premietneme na povrch gule pridaním tretej, konštantnej zložky. Pôvodné dve zložky vektora $\boldsymbol{x}$ takto predstavujú uhly, tretia odpovedá norme vektora. Prevod zložiek rozšíreného vektora $\boldsymbol{x} = [x_1, x_2, 1]^T$ zo sférickej súradnicovej sústavy do karteziánskej súradnicovej sústavy $[s_1, s_2, s_3]^T$ je daný trojicou vzťahov



Obrázok 7.15. Usporiadavanie váhových vektorov počas trénovania pomocou DP verzie algoritmu SOM. Číslo vpravo dole označuje iteráciu.

$$\begin{aligned} s_1 &= 1 \cdot \cos(x_1) \cdot \cos(x_2) \\ s_2 &= 1 \cdot \sin(x_1) \cdot \cos(x_2) \\ s_3 &= 1 \cdot \sin(x_2) \end{aligned} \tag{7.19}$$

Ak vektory $[s_1,s_2,s_3]^T$ budeme predkladať DP-SOM ako vstupy počas trénovania, sieť bude aproximovať dvojrozmernú plochu v trojrozmernom priestore. Keďže tretia dimenzia je redundantná, mapované súradnice budú odpovedať pôvodným súradniciam vektorov $\boldsymbol{x}$. Súčasne treba poznamenať, že transformácia (7.19) je jednoznačná, ak platí $-\pi/2 \le x_1 \le \pi/2$ a taktiež $-\pi/2 \le x_2 \le \pi/2$.

Situácia je zobrazená na nasledovnej sérii obrázkov. Vstupnou množinou boli dáta s rovnomerným rozdelením na intervale $\langle -1,1 \rangle^2$, transformované pomocou vzťahov (7.19) do trojrozmerného priestoru, čím sa premietli na časť pologule s kladnou prvou súradnicou. Obr. 7.15a-c zobrazujú váhové vektory SOM, pre názornosť premietnuté do roviny s nulovou x-ovou súradnicou. Ako vidieť, SOM v konečnej fáze aproximuje trénovaciu množinu.

7.6 Zachovanie topológie

Zo simulácií SOM na dvojrozmerných dátach s rovnomerným i nerovnomerným rozdelením bolo vidieť, že množina váhových vektorov mala tendenciu adekvátne pokryť dátovú oblasť (vlastnosť vektorovej kvantizácie, podkapitola 7.4.1). Čo je však dôležité, výsledné rozloženie váh nadobudlo v oboch prípadoch formu, ktorú možno intuitívne nazvať ako *usporiadanú.* Inými slovami, sieť sa javila ako plne rozvinutá, bez násilného kríženia uzlov (váhových vektorov). Takýto finálny stav siete súvisí s matematickým aspektom realizovaného zobrazenia — *vlastnosťou zachovania topológie.* Túto vlastnosť možno formulovať nasledovne:

> *Ak sa vstupný vektor* $\boldsymbol{x}$ *zobrazí na neurón i, potom všetky vstupné vektory blízke* $\boldsymbol{x}$ *(v zmysle euklidovskej metriky) sa zobrazia na ten istý neurón alebo na jeho topologických susedov.*

Vlastnosť zachovania topológie medzi vstupným a výstupným priestorom má svoj význam. Ako bolo spomenuté v úvode kapitoly, v biologických neurónových sieťach je vďaka topografickým mapám dosiahnutá konzistentná a veľmi úsporná reprezentácia geometrie vstupných stimulov. Z matematického pohľadu nám takéto zobrazenie umožňuje zbaviť sa redundantných dimenzií a pracovať v kvalitatívne rovnakom (topologicky ekvivalentnom) výstupnom priestore. Takto sa nielen zníži výpočtová náročnosť riešenej úlohy, ale navyše, pri práci s dátami vo výstupnom priestore môžu byť výsledky konkrétnej úlohy lepšie (napr. pri klasifikácii dát [5]), než by sa dosiahli s dátami uvažovanými v ich pôvodnej reprezentácii, teda vstupnom priestore.

Podľa matematickej definície spojité zobrazenie medzi dvoma topologickými priestormi zachováva topológiu vtedy, ak je prosté (jednojednoznačné) a súčasne existuje k nemu inverzné zobrazenie, ktoré je tiež spojité. Také zobrazenie sa nazýva *homeomorfizmus*. Z tejto definície možno intuitívne dedukovať, že homeomorfné zobrazenie zachováva vzťahy podobnosti (blízkosti). Voľnejšou požiadavkou pre zachovanie topológie je, že stačí, ak také zobrazenie zachováva len usporiadanie vzťahov podobnosti (pozri obr. 7.16). Vyplýva to z vety o topologickej ekvivalencii, ktorej dôkaz možno nájsť v práci [20]:

Veta (o topologickej ekvivalencii)*: Nech f a g sú metriky a nech (X,f), (Y,g) sú metrické priestory so spočitateľnými hustými podmnožinami*[6] *. Nech H :* $X \to Y$ *je bijektívne zobrazenie také, že platí:*

$$\forall x_1, x_2, x_3, x_4 \in X\text{:}\ \ f(x_1, x_2) < f(x_3, x_4) \Rightarrow g(H(x_1), H(x_2)) \le g(H(x_3), H(x_4))$$ a

$$\forall y_1, y_2, y_3, y_4 \in Y\text{:}\ \ g(y_1, y_2) < g(y_3, y_4) \Rightarrow f(H^{-1}(y_1), H^{-1}(y_2)) \le g(H^{-1}(y_3), H^{-1}(y_4))$$

potom H je homeomorfizmus a priestory X a Y sú topologicky ekvivalentné.

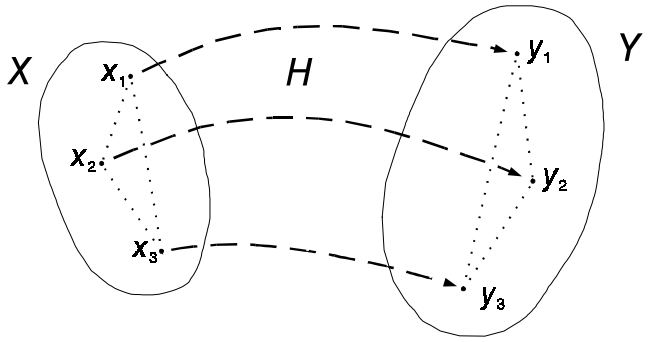
---

[6] Takýmto metrickým priestorom je napr. priestor $X = \Re^N$, N<∞, s euklidovskou metrikou $f = d_E(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$.

Táto veta hovorí, že pre zachovanie topológie je nutnou podmienkou to, aby usporiadanie podobností (blízkosti) medzi dvojicami bodov vo vstupnom i výstupnom spojitom priestore bolo monotónne. Keďže toto je prísnejšia podmienka, než aká je potrebná na to, aby zobrazenie bolo homeomorfizmom, možno *H* nazvať *topologickým (topografickým) homeomorfizmom* [20].

V prípade diskrétnych priestorov (s ktorými máme do činenia pri riešení praktických problémov pomocou SOM) nie je spojitosť (podmienka homeomorfizmu) definovaná tradičným spôsobom pomocou metriky. Môžeme však povedať, že zobrazenie medzi diskrétnymi priestormi, ktoré spĺňa vlastnosť usporiadania podobností, je diskrétnou aproximáciou topologického homeomorfizmu.

Pre potreby definovania vlastnosti zachovania topológie medzi diskrétnymi priestormi, teda pri použití neurónovej siete, bola zavedená aj iná, praktickejšia definícia tejto vlastnosti. Jej autorom je Martinetz [35], o práci ktorého sa ešte zmienime v podkapitole 7.9. Vychádza sa pri nej z predstavy, že máme množinu váhových vektorov, ktoré ležia na myslenej nelineárnej ploche — *variete M* (angl. *manifold*) danej štruktúrou vstupných dát, pričom uvažovanú SOM môžeme chápať ako *graf* pozostávajúci z vrcholov (neuróny) a spájajúcich hrán (definujúcich vzťahy susednosti neurónov). Najprv je potrebné



Obrázok 7.16. Ilustrácia zobrazenia *H* zachovávajúceho topológiu medzi vstupným priestorom *X* (s metrikou *f*) a výstupným priestorom *Y* (s metrikou *g*). Usporiadanie vzájomných vzťahov podobnosti medzi prvkami x_1, x_2 a x_3 ako aj medzi ich obrazmi y_1, y_2 a y_3 (získanými pomocou zobrazenia *H*) je rovnaké, hoci vzájomné vzdialenosti obrazov sú proporcionálne väčšie.

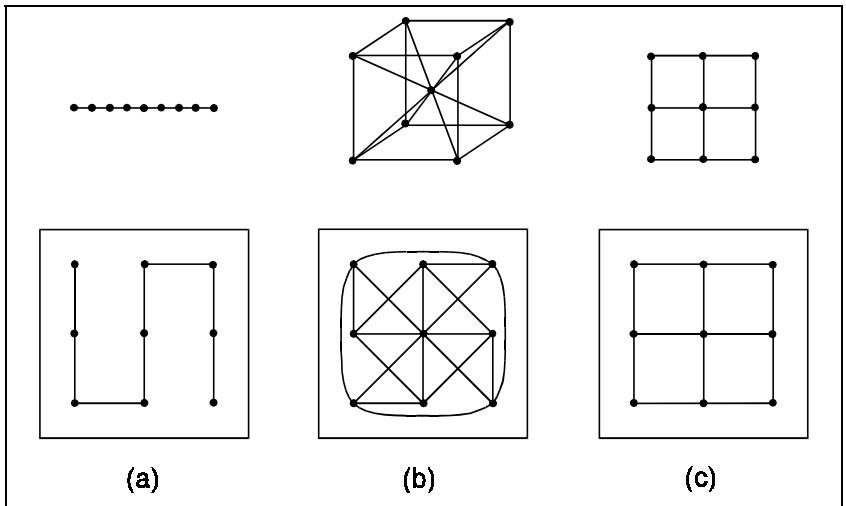
zadefinovať susednosť dvoch vektorov $\boldsymbol{w}_i$, $\boldsymbol{w}_j$ (bodov v euklidovskom priestore). Situácia je triviálna v jednorozmernom prípade (1D varieta M): dva body $\boldsymbol{w}_i$, $\boldsymbol{w}_j$ sú susedné, ak medzi nimi neleží žiaden bod $\boldsymbol{w}_k$. Použijúc termín Voronoiho polyédrov, ekvivalentným tvrdením pre N-rozmerný priestor je: dva body $\boldsymbol{w}_i$, $\boldsymbol{w}_j \in \Re$ sú susedné, ak ich Voronoiho polyédre sú susedné, t.j. ak $V_i \cap V_j \neq \varnothing$. Voronoiho polyéder je pre N-rozmerný prípad definovaný ako

$$V_i = \left\{ \boldsymbol{x} \in \Re^N \,/\, \|\boldsymbol{x} - \boldsymbol{w}_i\| \leq \|\boldsymbol{x} - \boldsymbol{w}_j\|, j = 1, 2, \ldots, n \right\} \quad i = 1, 2, \ldots, n \qquad (7.20)$$

Za účelom zovšeobecnenia definície susednosti dvoch bodov vo viacrozmernom priestore, t.j. pre body $\boldsymbol{w}_i$, $\boldsymbol{w}_j \in M \subseteq \Re^N$ bol zadefinovaný pojem *maskovaný* Voronoiho polyéder, ktorý predstavuje časť Voronoiho polyédra, ktorá leží na variete M (lebo nás zaujíma susednosť bodov $\boldsymbol{w}_i$, $\boldsymbol{w}_j$ na podpriestore tvorenom touto varietou). Množina všetkých maskovaných Voronoiho polyédrov tvorí celkovú hyperplochu M, t.j. $M = V_1^{(M)} \cup \ldots \cup V_n^{(M)}$. Rozšírenie definície susednosti je potom priame: Dva body $\boldsymbol{w}_i$, $\boldsymbol{w}_j \in M \subseteq \Re^N$ sú susedné na M, ak ich maskované Voronoiho polyédre $V_i^{(M)} = V_i \cap M$, $V_j^{(M)} = V_j \cap M$ sú susedné, t.j. ak $V_i^{(M)} \cap V_j^{(M)} \neq \varnothing$. Teraz môžeme pristúpiť k zamýšľanej definícii zobrazenia zachovávajúceho topológiu.

Definícia (zachovania topológie): *Nech G je graf (neurónová sieť) s vrcholmi* i=1,2,...,n *a hranami (určujúcimi topológiu siete) definovanými pomocou (symetrickej) matice "susedností"* $\boldsymbol{A}$ *s prvkami* A_{ij} = {0,1}, i,j=1,...,n. *Nech* $M \subseteq \Re^N$ *je daná varieta a nech* S = $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$ *je množina bodov* $\boldsymbol{w}_i \in M$, *z ktorých každý je priradený k vrcholu i grafu G. Potom graf G so svojimi vrcholmi i odpovedajúcimi bodom* $\boldsymbol{w}_i \in M$ *predstavuje* ***mapu zachovávajúcu topológiu*** *práve vtedy, keď platí, že zobrazenie* $\Phi_S : M \to G$ *ako aj inverzné zobrazenie* $\Phi_S^{-1} : G \to M$ *zachovávajú susednosť.*

Ostáva ešte objasniť vlastnosť zachovania susednosti, ktorá, ako vidieť, musí platiť oboma smermi: Zobrazenie $\Phi_S : M \to G$ zachováva susednosť, ak každej dvojici bodov $\boldsymbol{w}_i$, $\boldsymbol{w}_j$, ktoré sú susedné na M, odpovedajú vrcholy i, j, ktoré sú susedné v grafe G (obr. 7.17a). Analogicky, zobrazenie $\Phi_S^{-1} : G \to M$ zachováva susednosť, ak každej dvojici susedných vrcholov i, j grafu G odpovedajú body $\boldsymbol{w}_i$, $\boldsymbol{w}_j$, ktoré sú susedné na M (obr. 7.17b).



Obrázok 7.17. Zobrazenie štvorcovej variety M na tri grafy rôznych topológií (ktoré sú umiestnené v hornej časti obrázku). Každý z grafov má 9 vrcholov, korešpondujúce váhové vektory (zobrazené ako body) sú rozmiestnené rovnomerne na ploche M. V prvých dvoch prípadoch graf nezachováva topológiu, pretože zachovanie susednosti platí v prípade (a) len v "smere" zobrazenia Φ_S^{-1}; v prípade (b) len v "smere" Φ_S. Jedine v prípade (c) vlastnosť zachovania topológie platí.

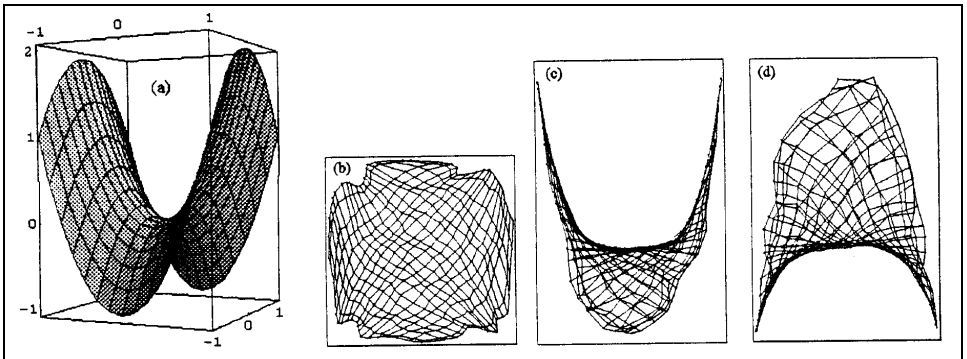
Pri spätnom pohľade na predošlé podkapitoly je vidieť, že s prípadmi (a) a (c) sme sa už v simuláciách stretli (obr. 7.8 a 7.10). Prípad (b) nastáva v praktických problémoch zriedka; už aj kvôli tomu, že dimenzia dát býva málokedy rovná jednej a topológiu siete obvykle volíme kvôli znázorniteľnosti nanajvýš rovnú dvom. Ak sa zamyslíme nad tým, ako zachovanie topológie medzi varietou M a grafom G súviselo s voľbou topológie SOM (grafu), uvedomíme si, že vlastnosť bola dodržaná, len ak sme použili SOM, ktorej topológia má dimenziu *zhodnú* s dimenziou dát (obr. 7.8), teda dva. Reťaz neurónov SOM síce tiež aproximuje funkciu hustoty dát, avšak vlastnosť zachovania topológie nemohla byť dodržaná, a to práve kvôli nerovnosti dimenzií. Pri pohľade na obr. 7.10 vidieť, že pre niektoré oblasti vstupov je vlastnosť splnená, t.j. že v rámci nich vybraté dva blízke vstupy majú topologicky blízkych víťazov, ale súčasne možno vybrať iné dva vstupy, pre ktoré to neplatí. Z pohľadu definície platí susednosť len v opačnom smere, Φ_S^{-1}.

Podmienka zhody dimenzií musí platiť vo všeobecnosti. Ak pracujeme s množinou dát, ktorej prvky sú popísané viacerými súradnicami ($\boldsymbol{x}_i \subseteq \Re^N$), často nastáva situácia, že prvky $\boldsymbol{x}_i$ nevypĺňajú celý N-rozmerný priestor, ale len jeho časť. Inými slovami, skutočná, *inherentná dimenzia* geometrickej štruktúry tvorenej prvkami $\boldsymbol{x}_i$ býva často menšia ako N. Inherentná dimenzia je daná minimálnym počtom premenných, potrebných na parametrický popis uvažovanej množiny dát. Ak teda zvolíme topológiu SOM[7] tak, že jej dimenzia sa

[7] Be□ne sa pou□ívajú architektúry v tvare mrie□ky alebo reťaze, mo□no pou□iť i trojrozmerný "kryštál". V princípe je algoritmus SOM pou□iteľný i pre viacrozmerné architektúry, avšak má to viaceré

bude zhodovať s inherentnou dimenziou dát, potom je pravdepodobné, že SOM sa naučí aproximovať dátovú štruktúru, navyše topologicky usporiadane. "Pravdepodobné" preto, že správna aproximácia nastáva s výnimkou prípadov extrémnych nelinearít v dátach.

Situácia je ilustrovaná na obr. 7.18a. Dátová množina bola generovaná podľa vzťahu $z = 2x^2 - y^2 + rnd(-0{,}2;0{,}2)$, pričom $x \in \langle -1{,}1 \rangle$, $y \in \langle -1{,}1 \rangle$ a $rnd(.;.)$ je generátor bieleho šumu na uvedenom intervale.[8] Vzhľadom k tomu, že rozptyl v smere *z*-ovej osi je relatívne malý, inherentná dimenzia generovanej množiny nimi tvorenej je len dva (pri parametrizácii zohľadňujúcej malý aditívny šum), hoci dáta ležia v trojrozmernom priestore. Pri použití SOM dostaneme výsledok, ktorého projekcie do rovín tvorených dvojicami súradníc sú na obr. 7.18b-d. Vďaka zhode dimenzií je dvojrozmerná SOM schopná pokryť (aproximovať) nelineárnu štruktúru (plochu) tvorenú dátami.



Obrázok 7.18. (a) Dátová štruktúra tvaru sedla, a odpovedajúce priemety natrénovanej SOM (20×20 neurónov) do projekčných rovín (b) x-y, (c) x-z a (d) y-z.

V praktických problémoch z toho vyplýva nutnosť *a priori* odhadnúť inherentnú dimenziu dát a následne zvoliť sieť s odpovedajúcou architektúrou (topológiou). V prípade nesprávneho odhadu je potom potrebné vyskúšať SOM s inou topológiou. Pri úspešnom použití siete dostávame *nelineárne zobrazenie* zo vstupného priestoru do priestoru neurónov, ktoré má tú dôležitú vlastnosť, že *redukuje dimenziu popisu dát*. Algoritmus SOM teda umožňuje "objaviť" a aproximovať nízkorozmerné nelineárne štruktúry pri zachovaní topológie, a to pri splnení vyššie spomínanej podmienky zhody dimenzií. Pre porovnanie, lineárne štatistické metódy ako napr. metóda hlavných komponent (angl. *principal component analysis,* PCA) sú schopné detekovať len lineárne vzťahy medzi vstupnými súradnicami, v dôsledku čoho sa napr. štruktúra tvaru sedla (obr. 7.18a) javí ako trojrozmerná, hoci jej inherentná dimenzia je rovná dvom.

---

nevýhody: s dimenziou exponenciálne narastá výpočtový čas i počet potrebných neurónov v sieti; navyše, stráca sa ilustratívnosť výstupnej reprezentácie, ktorá je vizualizovateľná len pre architektúry s dimenziou ≤ 3.

[8] Zobrazený útvar síce pre jednoduchosť predstavuje prípad pre *rnd(0,0)*, použité dáta pre trénovanie SOM však mali *z*-ovú zložku "zašumenú" podľa stanoveného intervalu.

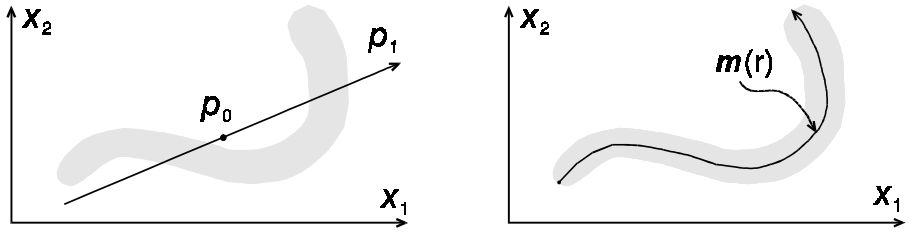
7.6.1 Extrakcia a topologické zobrazenie príznakov

Oba algoritmy — PCA i SOM — patria do kategórie algoritmov, ktoré umožňujú *extrahovať príznaky* (angl. *feature extraction*), t.j. charakteristické črty vstupných dát. Účelom extrakcie príznakov je transformovať dáta do priestoru nižšej dimenzie (f: $\Re^N \rightarrow \Re^D$), a to tak, aby sa ich vzájomné vzťahy (čo možno najviac) zachovali. Má to dve hlavné výhody: (a) jednak sa tým zníži dimenzia popisu dát, čím sa zjednoduší ich ďalšie spracovanie a (b) súčasne sa umožní vizualizácia a analýza dát (ak $D \leq 3$).

Algoritmy na extrakciu príznakov sú založené na klasických prístupoch [42] alebo sa jedná o algoritmy implementované pomocou neurónových sietí [33]. Hoci väčšina "neurónových" algoritmov na extrakciu príznakov je len priamou implementáciou klasických algoritmov (čo je i prípad PCA), neurónový prístup má niektoré výhody. Napríklad, implementácia Sammonovho algoritmu pomocou neurónovej siete už má schopnosť zovšeobecnenia, zatiaľ čo pôvodný algoritmus túto vlastnosť nemá [33].

V prípade PCA sa jedná o lineárnu transformáciu f. Tá "nájde" v dátach *hlavné komponenty* (angl. *principal components*), tvoriace ortogonálny súradnicový systém, v ktorých majú dáta maximálnu varianciu (stred novej súradnicovej sústavy sa pritom presunie do ťažiska $\boldsymbol{p}_0$ dátovej množiny, pozri obr. 7.19a). V situácii na obr. 7.19a ide o zobrazenie f: $\Re^2 \rightarrow \Re^1$ (do smeru hlavnej komponenty $\boldsymbol{p}_1$). Na jeho implementáciu stačí použiť jeden lineárny neurón s dvoma vstupmi (a jedno z pravidiel učenia Hebbovho typu [22]), pričom jeho výstup bude udávať hodnotu extrahovaného príznaku (súradnica v smere $\boldsymbol{p}_1$). Keďže nová súradnica je lineárnou kombináciou pôvodných súradníc, v prípade nelineárnej štruktúry dát sa stratí po premietnutí do lineárneho podpriestoru (kolmý priemet bodov $\boldsymbol{x}$ na priamku $\boldsymbol{p}_1$) značné množstvo informácie.

V prípade nelineárnej projekcie na krivku $\boldsymbol{m}(r)$ (obr. 7.19b) sa táto strata informácie podstatne redukuje. Analogicky k lineárnemu prípadu, $\boldsymbol{m}$ predstavuje *hlavnú krivku* (angl. *principal curve*) uvažovanej množiny dát [37]. Krivka $\boldsymbol{m}$ je parametrizovaná len jednou premennou r, ktorá udáva pozíciu bodu na krivke (vyjadrenú napr. oblúkovou vzdialenosťou bodu od koncového bodu krivky). Jednorozmerná SOM je schopná takúto krivku diskrétne (t.j. konečným počtom) aproximovať pomocou svojich váhových vektorov, a zobrazenie vstupu $\boldsymbol{x}$ na výstupnú 1D reprezentáciu sa realizuje nájdením bodu (váhového vektora) na krivke, ktorý je najbližšie k $\boldsymbol{x}$, t.j. $\boldsymbol{w}_{i^*}$. Jeho pozícia na krivke je daná indexom i^*, čo je diskrétna aproximácia parametra r. Keďže $\boldsymbol{m}$ výstižne popisuje geometriu dát, jej parameter r predstavuje ich charakteristickú črtu — *príznak*, ktorého hodnota je pre aktuálny vstup i^*. V prípade aproximácie dvojrozmernej plochy tvorenej dátami hovoríme o *hlavnej variete* (angl. *principal manifold*), ktorej parametrom je vektor $\boldsymbol{r}$ s dvoma nelineárnymi súradnicami. Tie reprezentujú hodnoty dvoch extrahovaných príznakov.



Obrázok 7.19. Premietnutie dát v 2D do jednorozmerného podpriestoru. (a) Lineárna projekcia: PCA nájde v dátach hlavné komponenty (zobrazená je len komponenta $\boldsymbol{p}_1$), ktoré sú lineárnou kombináciou pôvodných súradníc x_1, x_2 a v ktorých majú dáta (vyznačené sivou oblasťou) maximálnu varianciu. (b) Nelineárna projekcia: Dáta sú premietnuté na krivku $\boldsymbol{m}(r)$, ktorej parameter r (popisujúci pozíciu bodu pozdĺž krivky) je v nelineárnom vzťahu s pôvodnými súradnicami. Je zrejmé, že z oboch 1D reprezentácií predstavuje nelineárna projekcia lepší popis uvažovanej množiny dát.

Okrem iného, rozdiel medzi PCA a SOM spočíva v spôsobe reprezentácie príznakov. Pri PCA je na reprezentáciu jedného príznaku použitý len jeden neurón (hodnota príznakov je kvantifikovaná výstupmi neurónov), v prípade SOM sa na reprezentácii podieľajú všetky neuróny (hodnota príznakov je daná pozíciou víťaza). S tým súvisí špecifická vlastnosť SOM — extrahované príznaky sú *topologicky zobrazované* (angl. *feature mapping*), a to pozdĺž koordinát mriežky SOM. Podobne, na rozdiel od PCA, v prípade SOM môže byť interpretácia zobrazovaných príznakov v niektorých prípadoch obtiažna, čo samozrejme závisí od typu dát a súvisí s faktom, že nepoznáme transformačný vzťah *f*. Extrahované (a zobrazované) príznaky nemusia mať žiaden fyzikálny význam. Sieť si ich "zvolí" sama, výlučne v závislosti od geometrie dát.

7.6.2 Miery zachovania topológie

Na to, aby sme vedeli posúdiť, či nami zvolená SOM sa nachádza v usporiadanom stave, potrebujeme mať nejakú informáciu, ktorá charakterizuje

jej stav (t.j. pozície jej váhových vektorov), a to z hľadiska zachovania topológie. Najjednoduchšie je, ak taký kvantifikátor — *miera zachovania topológie* — je vyjadrený skalárnou hodnotou. Doteraz bolo navrhnutých niekoľko takýchto mier, pričom každá má iné vlastnosti. Spomenieme si niektoré z nich.

Miera usporiadania Θ. V práci [12] bola navrhnutá jednoduchá skalárna miera, ktorá kvantifikuje pravidelnosť usporiadania váhových vektorov SOM a je aplikovateľná pre ľubovoľnú dimenziu váhových vektorov. Bola definovaná ako $\Theta = \sigma_1 / \mu_1$, kde

$$\mu_1 = \underset{(i,j)\in A_1}{E}\left(\left\|\boldsymbol{w}_i - \boldsymbol{w}_j\right\|\right)$$
$$\sigma_1 = \sqrt{\underset{(i,j)\in A_1}{Var}\left(\left\|\boldsymbol{w}_i - \boldsymbol{w}_j\right\|\right)} \tag{7.21}$$

pričom E je stredná hodnota, Var je variancia a A_1 predstavuje množinu dvojíc indexov, ktoré odpovedajú susedným neurónom v štruktúre SOM, t.j. $A_1 = \left\{(i,j) \middle| \left\|\boldsymbol{r}_i - \boldsymbol{r}_j\right\| = 1\right\}$. Presnejšie povedané, Θ odzrkadľuje rovnomernosť rozostupov medzi susednými dvojicami váhových vektorov (kvantifikovanú štandardnou odchýlkou σ_1), teda pri rovnomernom rozostupe váh $\Theta \to 0$. Tým, že Θ sa vzťahuje na okamžitý stav siete, možno ju aplikovať opätovne po každej iterácii (resp. každej n-tej iterácii), čím dostávame informáciu o stave usporiadania váhových vektorov v priebehu učenia. Miera Θ má však dva nedostatky: (1) Pri váhových vektoroch vyšších dimenzií podlieha dôsledku vety o centrálnej limite.[9] Ako vyplýva zo vzťahu pre disperziu normy množiny náhodných vektorov, jej veľkosť sa pre dostatočne veľké n blíži ku konštantnej hodnote b, z čoho vyplýva, že vektory sú viac-menej normované. Tým dostávame pri aplikáciách s vysokorozmernými dátami hodnotu $\Theta \approx 0$ i v počiatočnom štádiu učenia, t.j. keď sú váhy ešte neusporiadane rozložené. Koniec koncov, i v prípade nízkorozmerných váh môže Θ nadobúdať "nesympaticky" nízku hodnotu, čo je však v tomto prípade spôsobené väčšou hodnotou menovateľa μ (je zrejmé, že ak sú váhy náhodne rozložené, je priemerná vzdialenosť susedných dvojíc väčšia). (2) Druhý nedostatok Θ spočíva v tom, že je ťažké z jej hodnoty usúdiť, či architektúra siete vyhovuje dimenzii dát vstupnej množiny (pozri tab. 7.1).

Topografický súčin P [2]. Podobne ako v predchádzajúcom prípade, na výpočet P nie je potrebná znalosť o štatistike vstupných dát, počíta sa výlučne zo vzájomných vzdialeností váhových vektorov vo vstupnom priestore a zo vzdialeností neurónov vo výstupnom priestore. Vzťah pre výpočet P je nasledovný:

---

[9] Vetu o centrálnej limite možno struène formulova nasledovne: Majme množinu náhodných vektorov $\boldsymbol{x}=[x_1,x_2,...,x_n]^T$, ktorých zložky sú nezávislé. Potom pre strednú hodnotu normy vektorov $\boldsymbol{x}$ a ich disperziu platia vz ahy $\mu_{\|\boldsymbol{x}\|} = E(\|\boldsymbol{x}\|) = \sqrt{an-b} + O(1/n)$ a $\sigma^2_{\|x\|} = Var(\|x\|) = b + O(1/\sqrt{n})$. Parametre a a b závisia výluène od centrálnych momentov rádu 1, 2, 3 a 4, a to pod¾a vz ahov $a = \mu^2 + \sigma^2$, $b = (4\mu^2\sigma^2 - \sigma^4 + 4\mu\mu_3 + \mu_4)/(4a)$, kde $\mu_k = E\left((\|\boldsymbol{x}\| - \mu)^k\right)$ je centrálny moment k-tého rádu.

$$P = \frac{1}{N.(N-1)} \sum_{j=1}^{N} \sum_{k=1}^{N-1} \log \left[\prod_{m=1}^{k} \frac{d^{in}\left(\mathbf{w}_j, \mathbf{w}_{n_m^{out}(j)}\right)}{d^{in}\left(\mathbf{w}_j, \mathbf{w}_{n_m^{in}(j)}\right)} \cdot \frac{d^{out}\left(j, n_m^{out}(j)\right)}{d^{out}\left(j, n_m^{in}(j)\right)} \right]^{\frac{1}{2k}} \tag{7.22}$$

kde N označuje počet neurónov, $n_m^{in}(j)$ (resp. $n_m^{out}(j)$) predstavuje m-tého najbližšieho suseda j-teho neurónu v priestore váh (resp. neurónov), d^{in} sa vzťahuje na výpočet vzdialenosti v priestore váh a d^{out} na výpočet fyzických vzdialeností neurónov v štruktúre SOM.[10] Ako vidieť, výpočet P je zložitejší, avšak poskytuje viac informácie v porovnaní s mierou Θ. Na základe hodnoty P možno zistiť, či zvolená architektúra siete "pasuje" na vstupné dáta (v takom prípade $P \approx 0$); navyše dáva informáciu o tom, či v prípade potreby treba znížiť alebo zvýšiť dimenziu architektúry siete, aby sme dostali lepšiu hodnotu P. Z hodnôt uvedených v tab. 7.1 vidieť, že najlepší výsledok pre danú vstupnú množinu bol získaný so sieťou 16×16, ktorá svojou štruktúrou a pomerom dĺžok strán najlepšie odpovedá vstupným dátam. V iných prípadoch, podhodnotenie optimálnej dimenzie (prípad reťaze) znamená $P < 0$, nadhodnotenie dimenzie (kryštál) dáva $P > 0$. V prípadoch 64×4 a 32×8 má sieť síce správnu dimenziu, avšak vzhľadom k pomeru jej strán sa sieť javí ako kvázi-jednorozmerná, preto $P < 0$.

Miera C. V práci [20] bola navrhnutá skalárna miera, ktorá, použijúc označenie pre neurónovú sieť, má tvar:

$$C = \sum_{i=1}^{N} \sum_{j<i} F(i,j) \cdot G\left(\mathbf{w}_i, \mathbf{w}_j\right) \tag{7.23}$$

kde N *je* počet neurónov, F je funkcia podobnosti vo vstupnom priestore, G vo výstupnom priestore. Pri uvažovaní funkcií F a G ako euklidovských vzdialeností platí, že čím "lepšie" zobrazenie, tým väčšia hodnota C. Uvedená miera bola odvodená z podmienky zachovania usporiadania podobnosti medzi dvoma priestormi. Autor dokázal, že ak medzi dvoma priestormi existuje homeomorfizmus, tak zobrazenie, pri ktorom je zachované usporiadanie podobnosti, dáva maximálnu hodnotu C. Z hodnôt uvedených v tabuľke 7.1 možno usúdiť, že v rámci architektúr tej istej dimenzie hodnota C odpovedá tej najlepšej, t.j. 16×16 neurónov. Problematické je však na základe C určiť, či zvolená architektúra SOM vyhovuje dimenzii vstupných dát.

Reprezentácia dy-dx. Okrem skalárnych mier užitočnou pomôckou pri práci so SOM môže byť i grafická reprezentácia stavu. Príkladom takejto je reprezentácia *dy-dx* [11], ktorá poskytuje grafickú informáciu o pravidelnosti usporiadania siete (je v podstate zovšeobecnením miery Θ). Spočíva v zobrazení vzájomných vzdialeností váhových

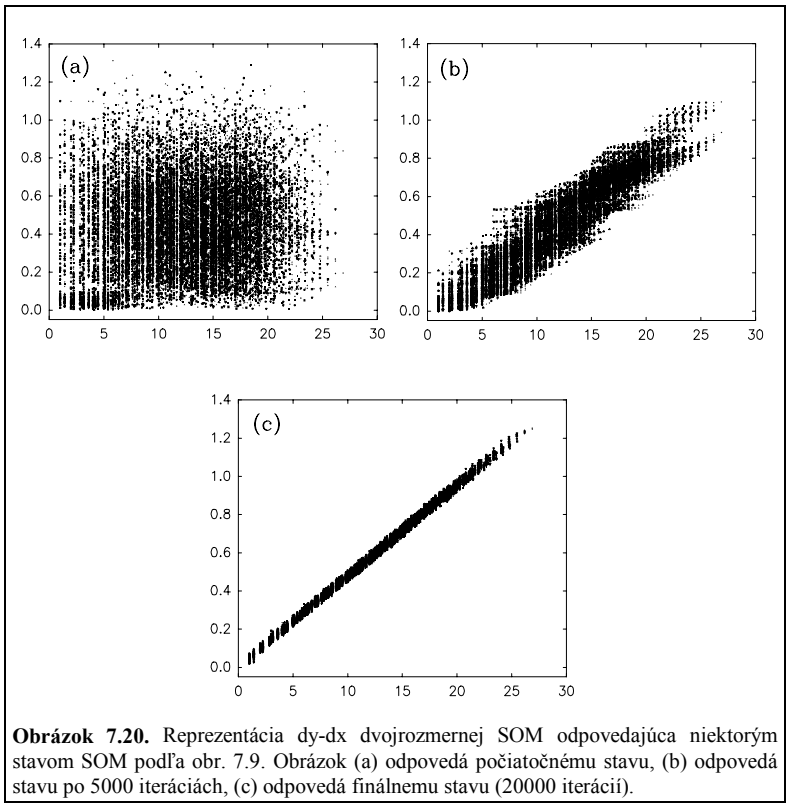
---

[10] Vzájomné vzdialenosti neurónov v SOM nadobúdajú len diskrétne hodnoty, a to $1, \sqrt{2}, 2, \sqrt{5}, \ldots$ atï. Vyplýva to z pravidelnosti usporiadania neurónov v uvažovanej štruktúre, prièom vzdialenos medzi dvoma susednými neurónmi je rovná jednej.

vektorov $dy(i,j) = \left\| \boldsymbol{w}_i - \boldsymbol{w}_j \right\|$ ako funkcie vzájomných vzdialeností im odpovedajúcich neurónov v štruktúre SOM, t.j. $dx(i,j) = \left\| \boldsymbol{r}_i - \boldsymbol{r}_j \right\|$. Výsledkom zobrazenia je množina bodov, z ktorých každý odpovedá nejakej dvojici (*i*,*j*) a je zobrazený v rovine *dy*-*dx*. Ako vidieť na obr. 7.20, reprezentácia *dy*-*dx* graficky popisuje mieru korelácie medzi *dy* a *dx,* ktorá závisí od stavu usporiadania váhových vektorov. Dá sa však očakávať, že pri aproximácii nelineárnych dátových štruktúr nebude stav usporiadania siete podľa tejto reprezentácie taký evidentný, čo vyplýva z komplikovanejšej korelácie medzi *dy* a *dx*.

Tabuľka 7.1. Priemerné hodnoty spomínaných mier zachovania topológie, pre rôzne architektúry SOM po natrénovaní. Vstupnou množinou boli vo všetkých prípadoch dvojrozmerné vektory s rovnomerným rozdelením na intervale $\langle 0,1 \rangle^2$, vyberané náhodne. Optimálnou architektúrou spomedzi uvádzaných je sieť typu 16×16 neurónov, ktorá najlepšie vyhovuje geometrii vstupných dát.

Architektúra SOM	Θ	P	C
256	0,350±4,4%	-0,1050±9,5%	1694182±1,7%
64×4	0,468±3,4%	-0,0660±3,0%	529876±1,6%
32×8	0,356±7,0%	-0,0301±3,3%	552382±1,3%
16×16	0,232±5,0%	0,0005±4,0%	649241±0,4%
10×10×2	0,604±0,7%	0,0076±3,9%	73663±0,3%
6×6×6	0,501±12,%	0,0382±0,2%	94220±0,4%



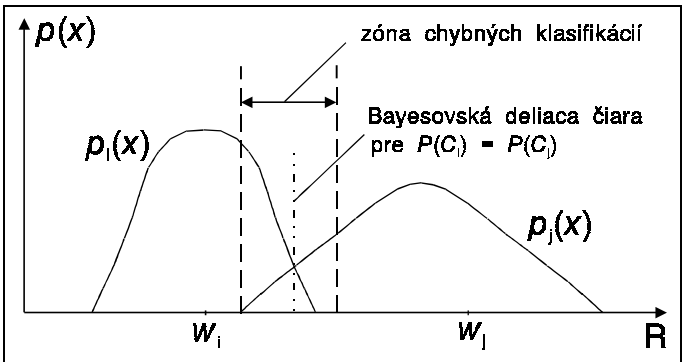
Obrázok 7.20. Reprezentácia dy-dx dvojrozmernej SOM odpovedajúca niektorým stavom SOM podľa obr. 7.9. Obrázok (a) odpovedá počiatočnému stavu, (b) odpovedá stavu po 5000 iteráciách, (c) odpovedá finálnemu stavu (20000 iterácií).

7.7 Hybridné učenie s učiteľom — algoritmy LVQ

Doteraz sme o SOM uvažovali ako o samoorganizovanom modeli, ktorý je schopný premietnuť vstupné vzory do priestoru nízkej dimenzie, a to pri zachovaní ich vzájomných topologických vzťahov. Teraz si ukážeme, ako možno takúto SOM následne použiť na štatistické rozpoznávanie vzorov. Za týmto účelom boli vyvinuté *algoritmy LVQ* (*Learning Vector Quantization*) [29,25], ktoré sa aplikujú po natrénovaní SOM. Ich úlohou je doladiť váhové vektory tak, aby sa minimalizoval počet chybných klasifikácií, ktoré vznikajú v dôsledku prekryvu medzi triedami.

Pri úlohe štatistického rozpoznávania vzorov pracujeme s množinou $\boldsymbol{X}$ vstupných vzorov $\boldsymbol{x}$, z ktorých každý patrí do jednej z uvažovaných tried. Úlohou klasifikátora je po predložení vstupu $\boldsymbol{x}$ rozhodnúť, do ktorej triedy patrí. V prípade SOM sa klasifikácia vstupu $\boldsymbol{x}$ určí na základe návestia jemu najbližšieho váhového vektora, ktoré podobne ako u všetkých váhových vektorov označuje príslušnosť k tej-ktorej triede. Na rozdiel od problému vektorovej kvantizácie, s ktorou sme sa stretli v podkapitole 7.4.1, tu nie je dôležité, ktorý konkrétny neurón zvíťazí; podstatné je, aby to bol jeden z neurónov reprezentujúcich tú "správnu" triedu. Z toho možno usúdiť, že problematické situácie môžu

vznikať práve v oblasti hraníc medzi susediacimi triedami. Preto kľúčovým krokom je stanovenie *optimálnych rozhodovacích plôch* (dimenzie *N-1* pri uvažovaní $\boldsymbol{X}\subset\Re^N$), ktoré rozdelia vstupný priestor $\boldsymbol{X}$ na zóny prislúchajúce jednotlivým triedam.



Obrázok 7.21. Ilustrácia problému klasifikácie do dvoch tried v 1D prípade. V dôsledku prekryvu medzi triedami vzniká nenulová chyba klasifikácie, pričom aktuálna zóna chybných klasifikácií je väčšia ako minimálna, ktorá by nastala v prípade, ak by deliaca čiara medzi váhami bola totožná s teoretickou bayesovskou.

Problém je v jednorozmernom prípade (vstupy i váhy skalárne) znázornený na obr. 7.21, na ktorom je každá z dvoch tried *i* a *j* (s prekrývajúcimi sa funkciami hustôt $p_i(x)$ a $p_j(x)$) reprezentovaná jednou váhou. V dôsledku neoptimálnych pozícií váh je deliaca čiara medzi nimi (pravá čiarkovaná čiara) posunutá od optimálnej bayesovskej, ktorá spĺňa podmienku (pozri napr. [44], kapitola 4)

$$p(x|C_i).P(C_i) = p(x|C_j).P(C_j) \ , \qquad (7.24)$$

kde $p(x|C_i)$ je funkcia hustoty vstupných vzorov x, ktoré patria do triedy C_i a $P(C_i)$ je apriórna pravdepodobnosť, že pri náhodnom výbere vstupu x bude ten patriť do triedy C_i. Pri znalosti pravdepodobností figurujúcich v rovnosti (7.24) možno uvažovať o *diskriminačných funkciách* tvaru $\delta_i(x)=g(p(x|C_i)P(C_i))$, kde g je ľubovoľná monotónne rastúca funkcia. Klasifikácia sa potom vykoná na základe nájdenia $\max(\delta_i(x))$; na hraniciach medzi triedami C_i a C_j platí $\delta_i(x)=\delta_j(x)$.

Snahou LVQ je aproximovať teoretické bayesovské hranice bez znalosti štatistiky vstupných dát. Podobne ako v algoritme SOM, i tu sa adaptácia uskutočňuje iteratívnym posunom váhových vektorov — prototypov, avšak využívajúc trénovacie vstupy s návestiami, teda *s učiteľom*.

Prv než možno pristúpiť k algoritmu LVQ, je potrebné váhové vektory najprv *inicializovať* a potom *priradiť návestie* každému z nich. Jednou z možností inicializácie je samotný algoritmus VQ (podkapitola 7.4.1), ktorý má tendenciu minimalizovať strednú kvadratickú chybu (vzťah 7.7, resp. 7.8). V prípade, že sa dá predpokladať vysoká štruktúrovanosť dát (ležia na nízkorozmernom podpriestore), je efektívne použiť na inicializáciu SOM, ktorá navyše umožňuje odhaliť a zobraziť vzájomné topologické vzťahy medzi vstupnými vzormi.

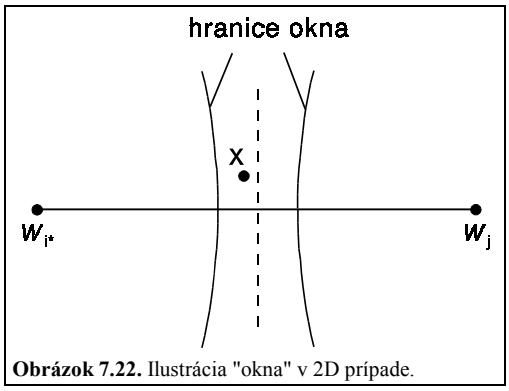
Následné priradenie návestí možno realizovať tak, že pri testovaní na reprezentatívnej množine vstupov so známou klasifikáciou sa vyhodnotí početnosť "víťazstiev" každého prototypu a návestie sa určí na základe väčšinového pravidla. Takýmto spôsobom získame pre každú triedu (obyčajne) viacero prototypov, ktoré tvoria konvexné množiny.

LVQ1. Prvá verzia algoritmu LVQ na doladenie prototypov využíva koncept "odmeny a trestu" (ktorý figuruje aj pri perceptróne), a to v aplikácii len na víťazný neurón (nájdený podľa minima euklidovskej vzdialenosti):

$$\begin{aligned} \boldsymbol{w}_{i^*}(t+1) &= \boldsymbol{w}_{i^*}(t) + \alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_{i^*}(t)\right], \quad \text{ak} \quad cls(\boldsymbol{x}) = cls(\boldsymbol{w}_{i^*}) \\ \boldsymbol{w}_{i^*}(t+1) &= \boldsymbol{w}_{i^*}(t) - \alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_{i^*}(t)\right], \quad \text{ak} \quad cls(\boldsymbol{x}) \neq cls(\boldsymbol{w}_{i^*}) \end{aligned} \tag{7.25}$$

pričom $\alpha(t) \in (0,1)$ je monotónne klesajúca funkcia, avšak s výrazne menšou počiatočnou hodnotou, rádovo 0,01 (v porovnaní s algoritmom SOM). $cls(\boldsymbol{x})$ označuje triedu, do ktorej $\boldsymbol{x}$ patrí. Efekt, ktorý možno pozorovať pri LVQ1, je tendencia prototypov vzďaľovať sa od bayesovských hraníc. LVQ1 je však stabilný pri "slušnom" tvare funkcií hustôt jednotlivých tried a vhodnej inicializácii prototypov.

LVQ2. Na rozdiel od LVQ1 sa v LVQ2 adaptujú v každej iterácii dva prototypy — víťazného neurónu i^* a druhého najbližšieho neurónu j k vstupu $\boldsymbol{x}$. Snahou LVQ2 je posunúť deliacu čiaru medzi nimi smerom k pomyslenej bayesovskej hranici. Pre tento účel sa zavádza pojem "okna" W okolo deliacej čiary, ako to vidieť na obr. 7.22.



Obrázok 7.22. Ilustrácia "okna" v 2D prípade.

Korekcie sú definované ako

$$\begin{aligned} \boldsymbol{w}_{i^*}(t+1) &= \boldsymbol{w}_{i^*}(t) - \alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_{i^*}(t)\right] \\ \boldsymbol{w}_{j}(t+1) &= \boldsymbol{w}_{j}(t) + \alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_{j}(t)\right] \end{aligned} \quad \text{pre} \quad \boldsymbol{x} \in W \quad cls(\boldsymbol{x}) = cls(\boldsymbol{w}_j) \neq cls(\boldsymbol{w}_{i^*}) \tag{7.26}$$

pričom platí, že $\boldsymbol{x}$ leží v okne W, ak $\min(d_{i^*} / d_j, d_j / d_{i^*}) > s$, kde $d_j = \left\| x(t) - w_j \right\|$.

Hranice okna tvoria Apolloniove hyperplochy (dimenzie N-1), pričom ak w je relatívna šírka okna W v jeho najužšom mieste, tak $s=(1-w)/(1+w)$. Optimálna veľkosť okna závisí od počtu trénovacích vzorov: čím viac vzorov, tým užšie okno možno zvoliť. Aby sa však

dosiahla dobrá štatistická presnosť, počet vzorov ktoré "padnú" do okna musí byť dostatočne veľký. Kohonen uvádza kompromisnú hodnotu $w = 0{,}2$.

Ako sa však ukázalo, po väčšom počte iterácií (rádovo 10000) spôsoboval LVQ2 zhoršovanie výsledkov, a to kvôli jeho tendencii monotónne zmenšovať vzdialenosti $\|\boldsymbol{w}_{i^*} - \boldsymbol{w}_j\|$. Na kompenzáciu tohto nežiaduceho efektu navrhol Kohonen modifikáciu *LVQ2.1,* využívajúc rovnaké vzťahy (7.26), avšak s podmienkou, že $\boldsymbol{w}_{i^*}$ *a* $\boldsymbol{w}_j$ *sú dva najbližšie prototypy* k $\boldsymbol{x}$ (teda i^* nemusí byť víťaz) a súčasne platí $\boldsymbol{x} \in W$ a $cls(\boldsymbol{x}) = cls(\boldsymbol{w}_j) \neq cls(\boldsymbol{w}_{i^*})$. Takáto modifikácia má za následok dvojnásobný počet korekcií prototypov (v porovnaní s LVQ2), pričom je zabezpečený i nárast $\|\boldsymbol{w}_{i^*} - \boldsymbol{w}_j\|$, čím sa zvyšuje štatistická presnosť v učení.

LVQ3. Nakoľko i pri LVQ2.1 bolo pozorované čiastočné zhoršovanie výsledkov po dlhšom trénovaní v dôsledku posunov hraničných prototypov do suboptimálnych pozícií, prišiel autor s myšlienkou zahrnúť do algoritmu efekt "kohézie" prototypov v rámci tried. Týka sa modifikácie prototypov v prípade *jednoznačne klasifikovateľných* vstupov, t.j. tých, ku ktorým sú dva najbližšie prototypy z tej istej triedy. Učenie má tvar

$$\begin{aligned} \boldsymbol{w}_{i^*}(t+1) &= \boldsymbol{w}_{i^*}(t) - \alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_{i^*}(t)\right] \quad \text{pre } \boldsymbol{x} \in W \text{ a } cls(\boldsymbol{x}) = cls(\boldsymbol{w}_j) \neq cls(\boldsymbol{w}_{i^*}) \\ \boldsymbol{w}_j(t+1) &= \boldsymbol{w}_j(t) + \alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_j(t)\right] \end{aligned} \tag{7.27a}$$

$$\boldsymbol{w}_k(t+1) = \boldsymbol{w}_k(t) + \varepsilon.\alpha(t).\left[\boldsymbol{x}(t) - \boldsymbol{w}_k(t)\right], \quad k \in \{i, j\}$$

$$\text{pre } cls(\boldsymbol{x}) = cls(\boldsymbol{w}_j) = cls(\boldsymbol{w}_{i^*}) \tag{7.27b}$$

kde parameter ε sa volí v závislosti od veľkosti okna: pri jeho malej šírke musí byť ε tiež malé. Na základe simulácií dospel autor k záveru, že vhodným rozsahom hodnôt je interval (0,1-0,5). Simulácie taktiež potvrdili stabilizačný účinok LVQ3 i pri rozsiahlejších dobách trénovania (rádovo 100000). Voľba optimálnej verzie algoritmu závisí od problému. Avšak štatisticky významné rozdiely vo výslednej chybe klasifikácie sa v simuláciách nepotvrdili. V čom sa verzie viac líšia, je ich stabilita. Z tohto hľadiska je vhodnejšie voliť LVQ1 alebo LVQ3, ktoré majú stabilizačný účinok i v dlhších trénovacích cykloch.

7.8 Niektoré aplikácie SOM

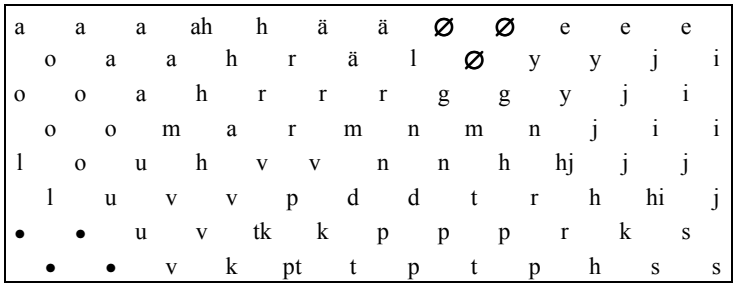
Aplikácií SOM je veľa, rozsiahly zoznam referencií na ne možno nájsť v Kohonenovej knihe o SOM [30]. Tu spomenieme aspoň niektoré z nich.

Rozpoznávanie reči [28]. Táto aplikácia SOM spočíva v transformácii akustického rečového signálu na sekvenciu hlások, tzv. fonetický prepis (ako napr. v slovenčine sekvencia ď-j-e-u-č-a). Prostriedkom tejto transformácie je natrénovaná SOM — tzv. *fonémová mapa* (obr. 7.23), ktorá svojou odpoveďou je schopná lokalizovať odozvu na

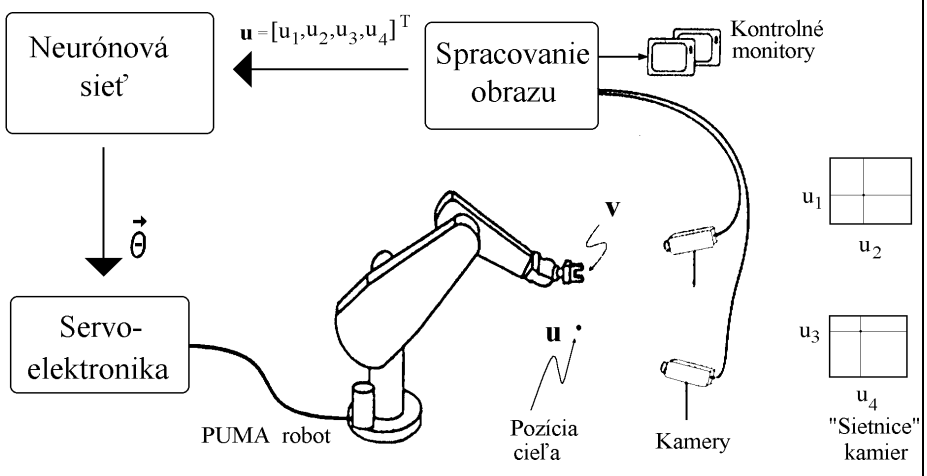
aktuálny vstup $\boldsymbol{x}$ odpovedajúci nejakej hláske (fonéme). Aplikácia bola urobená pre fínsky a japonský jazyk, a to predovšetkým z dvoch dôvodov: (a) oba jazyky sú inflexnými jazykmi (t.j. príbuzné slová sa v nich tvoria pomocou predpôn a prípon), preto je výhodnejšie voliť malé akustické jednotky (hlásky, slabiky) ako objekty rozpoznávania, lebo ich počet je obmedzený (rozpoznávanie na úrovni slov by prestávalo byť efektívne pri ich vysokom počte, pretože každý tvar slova treba považovať sa samostatnú kategóriu); (b) oba jazyky obsahujú fonémy, ktoré sú ľahko rozlíšiteľné na základe ich stacionárnych spektrálnych vlastností.

Na to, aby bolo možné trénovať SOM, je potrebné rečový signál predspracovať. Vo všeobecnosti existuje viacero spôsobov predspracovania rečového signálu, ktoré sa bežne používajú. Kohonen použil spektrálnu analýzu. Predspracovanie signálu (počnúc krokom (3) realizované na signálovom procesore) spočívalo v nasledovných krokoch: (1) dolnopriepustná filtrácia s f = 5,3 kHz, (2) 12-bitová AD konverzia s frekvenciou vzorkovania 13,02 kHz, (3) 256-bodová rýchla Fourierova transformácia (FFT) každých 9,83 ms použijúc Hammingovo okno, (4) logaritmizácia spektra a jeho vyhladenie, (5) umiestnenie 15 zložiek vhodným zlúčením komponent FFT v rozsahu 200-5000 Hz do spektrálneho vektora $\boldsymbol{x}$, (6) centrovanie $\boldsymbol{x}$ okolo strednej hodnoty (odčítaním strednej hodnoty od všetkých zložiek), (7) normovanie $\boldsymbol{x}$.

Takýmto spôsobom sa z každého okna konštantnej dĺžky 9,83 ms (ktorú možno v reči považovať za štatisticky stacionárny úsek) získala spektrálna reprezentácia vo forme vektora $\boldsymbol{x} \in \Re^{15}$. Po natrénovaní SOM na takejto množine vektorov boli jednotlivým neurónom priradené návestia podľa väčšinového pravidla, a váhy SOM boli pred rozpoznávaním doladené pomocou LVQ (podkapitola 7.7). Pri testovaní SOM predstavovali odozvy (reprezentované pozíciou víťazných neurónov) v SOM trajektóriu, ktorej uzly odpovedali jednotlivým po sebe nasledujúcim vstupným vektorom $\boldsymbol{x}$. Z hľadiska nadobudnutej reprezentácie je zaujímavé, že súradnice SOM nemajú explicitný význam — sieť si ich vybrala automaticky počas trénovania. Z výstupnej fonetickej sekvencie bolo možné dosiahnuť ortografický prepis s 90%-nou presnosťou, pričom po odstránení koartikulačných efektov pomocou gramatických pravidiel dosiahol implementovaný *fonetický písací stroj* presnosť v rozsahu 92-97% v reálnom čase.



Obrázok 7.23. Fonémová mapa (s hexagonálnou štruktúrou). Dvojité návestia označujú neuróny, ktoré reagujú na dve fonémy. Rozlíšenie niektorých foném nie je spoľahlivé, nutná je doplnková analýza. Prevzaté z [28].



Obrázok 7.24. Schéma robotického systému s "externým zrakom" (uvažovaný je neredundantný prípad — robot s troma stupňami voľnosti). Dve kamery snímajú pozíciu cieľového objektu $\boldsymbol{u}$ i koncového bodu efektora $\boldsymbol{v}$. Blok predspracovania obrazu extrahuje 4D vektor odpovedajúci konkrétnej pozícii v 3D vstupnom priestore. Rameno robota je riadené trojicami príkazov vo forme uhlov natočenia, ktoré sú výstupom neurónovej siete.

Robotika. Jednou zo základných úloh v tejto oblasti je naučiť robota umiestniť svoje koncové rameno (efektor) do žiadanej polohy. Schéma realizovaného systému na riešenie tohto problému s použitím SOM je na obr. 7.24 [45].

Cieľom je naučiť neurónovú sieť transformáciu $\vartheta: \boldsymbol{u} \in \boldsymbol{U} \subseteq \Re^4 \rightarrow \theta \in \Re^3$ bez učiteľa. Prístup spočíva v adaptívnom kvantovaní vstupného priestoru U na N disjunktných oblastí F_i, $i \in \{1,...,N\}$ a aproximácii θ v každej oblasti lineárnym zobrazením, ktoré sa postupne "dolaďuje". Počet oblastí N je daný zvoleným počtom neurónov SOM (s 3D architektúrou), z ktorých každému je priradený váhový vektor $\boldsymbol{w}_i \in \Re^4$ aproximujúci ťažisko oblasti F_i, výstupný vektor θ_i a matica A_i (typu 3×4), ktoré spolu určujú lineárny Taylorov rozvoj $\theta(\boldsymbol{u})$ v rámci F_i:

$$\theta(\boldsymbol{u}) = \theta_i + \boldsymbol{A}_i.(\boldsymbol{u} - \boldsymbol{w}_i). \tag{7.28}$$

Na adaptáciu váh neurónov ako aj ich výstupov θ_i a $\boldsymbol{A}_i$ je použitý rozšírený algoritmus SOM[11] (ako aj algoritmus "neural gas" [34])

[11] Rozšírenie algoritmu SOM spočíva práve v priradení maticového výstupu jednotlivým neurónom a využití vlastnosti zachovania topológie pri ich adaptácii.

$$\begin{aligned} \boldsymbol{w}_i &\leftarrow \boldsymbol{w}_i + \varepsilon . h(i, i^*).(\boldsymbol{u} - \boldsymbol{w}_i) \\ \theta_i &\leftarrow \theta_i + \varepsilon' . h'(i, i^*).\Delta\theta_i \\ \boldsymbol{A}_i &\leftarrow \boldsymbol{A}_i + \varepsilon' . h'(i, i^*).\Delta\boldsymbol{A}_i \end{aligned} \tag{7.29}$$

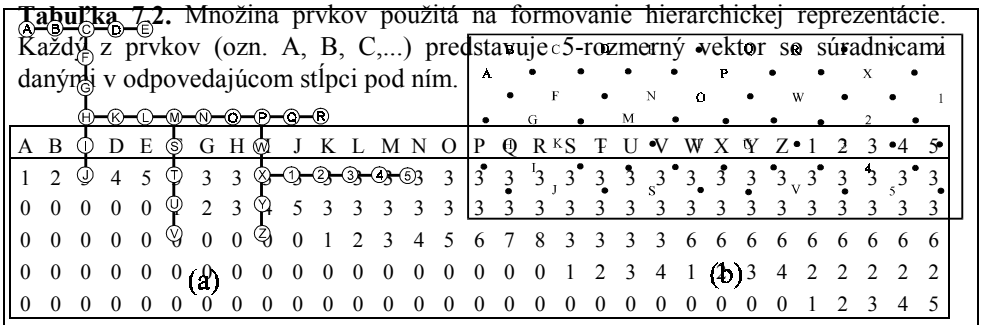
kde ε, ε', *h(i*.i)* a *h'(i*.i)* sú štandardné parametre SOM, veličiny $\Delta\theta_i$, ΔA_i sa vypočítajú spôsobom bližšie opísaným v [45]. Učenie prebieha tak, že v každom trénovacom kroku sa zadá cieľová pozícia $\boldsymbol{u}$ a rameno robota sa iteratívne približuje k cieľu. V prvom priblížení sa nájde víťaz a požadované uhly sa vypočítajú podľa (7.28). Ďalej sa generovaný výstup získa iteratívnym spôsobom, a to váhovaným spriemernením príspevkov od všetkých neurónov v okolí víťaza podľa rekurentného vzťahu

$$\theta_n^{out} = \theta_{n-1}^{out} + \boldsymbol{s}^{-1} . \sum_i h(i^*, i). \boldsymbol{A}_i . (\boldsymbol{u} - \boldsymbol{v}_n) \tag{7.30}$$

kde $\boldsymbol{s} = \sum_i h(i^*, i)$ a $\boldsymbol{v}_n$ je pozícia efektora v *n*-tej iterácii. Opakovane sa aplikujú učiace pravidlá (7.29). Po predložení asi 3000 cieľových pozícií je robot schopný dosiahnuť žiadanú pozíciu s presnosťou 1,3 mm, pričom, ako uvádzajú autori, tá by sa dala ešte zvýšiť použitím kamier s lepšou rozlišovacou schopnosťou.

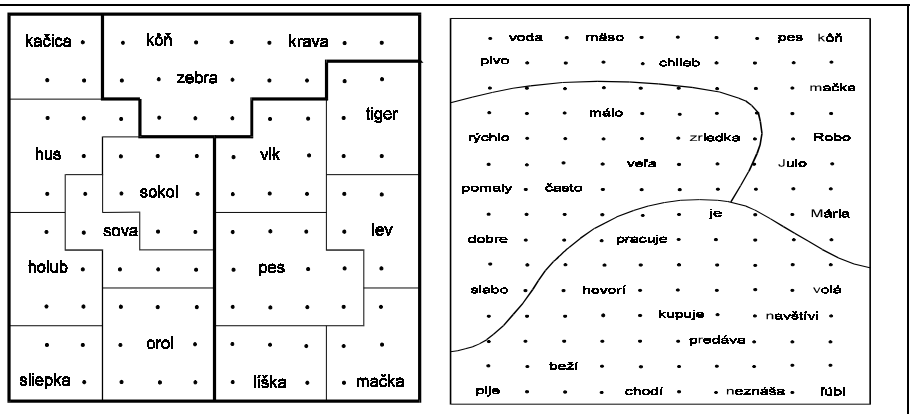
Formovanie hierarchických reprezentácií. SOM je schopná topologicky zobraziť hierarchické vzťahy medzi jednotlivými vstupnými prvkami, pokiaľ sú tieto vhodne popísané pomocou svojich súradníc [25]. Na ilustráciu tejto úlohy možno použiť reprezentáciu podľa tab. 7.2.

Tabuľka 7.2. Množina prvkov použitá na formovanie hierarchickej reprezentácie. Každý z prvkov (ozn. A, B, C,...) predstavuje 5-rozmerný vektor so súradnicami danými v odpovedajúcom stĺpci pod ním.



Obrázok 7.25. (a) Minimálny strom grafu dát uvedených v tab. 7.2. (b) Reprezentácia dát z tab. 7.2 získaná pomocou SOM.

Vzťahy medzi jednotlivými prvkami množiny možno zobraziť klasickou metódou, čím získame minimálny strom (obr. 7.25a). Pri použití SOM na tých istých dátach dostaneme mapu ako napr. na obr. 7.25b. Podobnosť medzi oboma vyobrazeniami je zrejmá: jednotlivé "vetvy" v SOM sú síce všelijako stočené (tak, aby sa "zmestili" do mapy), avšak topologické vzťahy medzi susednými vzormi sú v podstate rovnaké.



Obrázok 7.26. (a) Usporiadanie symbolov v SOM podľa ich atribútov. Ohraničené oblasti odpovedajú jednotlivým symbolom, v rámci každej z nich je neurón s maximálnou odozvou označený symbolom triedy. Usporiadanosť je evidentná medzi zvieracími druhmi (oddelenými hrubšími čiarami) i v rámci nich. (b) "Sémantická mapa" symbolov vygenerovaná na základe vzorov symbol—dvojslovný kontext. Separované oblasti označujú jednotlivé slovné druhy, v rámci ktorých je navyše evidentná usporiadanosť podľa významu zastúpených symbolov.

Topografické mapy abstraktných dát. V prípade abstraktných, symbolických dát vzniká otázka, ako možno u tých zisťovať a zobrazovať ich vzájomné, sémantické vzťahy [40]. V prípade fyzikálne relevantných dát to nie je problém, pretože ich samotná reprezentácia odzrkadľuje vzájomné vzťahy podobnosti (napr. blízkosť odpovedajúcich súradníc dvoch vektorov v zmysle euklidovskej metriky). Avšak v prípade symbolov (napr. slov prirodzeného jazyka) to neplatí, pretože medzi kódom symbolu (reprezentovaného napr. písomnou formou) a jeho významom nie je žiadna súvislosť. Keďže vzťahy medzi symbolmi nie sú zistiteľné z ich kódových reprezentácií, je potrebné ich prezentovať v spätosti s *kontextom*, v ktorom sa vyskytujú.

V prvom príklade bol kontext každého symbolu (meno zvieraťa) reprezentovaný vektorom binárnych *atribútov* (prítomnosť atribútu označená jednotkou, absencia nulou) ako veľkosť zvieraťa (malé, stredné, veľké), vonkajší popis tela (má 2 nohy, 4 nohy, srsť, kopytá, hrivu, perie) a čo rado robí (loví, behá, lieta, pláva). Takýto 13-rozmerný vektor atribútov $\boldsymbol{x}_a$ bol vygenerovaný pre každé zviera, pričom kódy zvierat $\boldsymbol{x}_s$ boli zámerne vytvorené tak, aby neniesli žiadnu informáciu o vzájomnej podobnosti medzi symbolmi: každý vektor $\boldsymbol{x}_s$ obsahoval samé nuly, až na jednu hodnotu a, ktorá figurovala na pozícii udávajúcej poradové číslo zvieraťa (1-16). Oba vektory boli zlúčené v jeden 29-rozmerný vektor $\boldsymbol{x}=[\boldsymbol{x}_s,\boldsymbol{x}_a]^T$ charakterizujúci každé zviera, pričom hodnota a bola stanovená na a = 0,2, aby vplyv atribútovej časti vektora $\boldsymbol{x}$ (nositeľ informácie o zvierati) bol väčší ako vplyv symbolovej časti. Vektory $\boldsymbol{x}$ boli napokon normované kvôli lepšej stabilizácii učenia. Počas trénovania bolo SOM prezentovaných 2000 náhodne vyberaných vzorov $\boldsymbol{x}$ z 16-prvkovej množiny. Proces určovania návestí bol však realizovaný na základe vektorov $\boldsymbol{x}=[\boldsymbol{x}_s,\boldsymbol{0}]^T$, čoho výsledkom je mapa na obr. 7.26a. Kvalitatívne rovnakú reprezentáciu by sme dostali i pri použití $\boldsymbol{x}=[\boldsymbol{x}_s,\boldsymbol{x}_a]^T$, z čoho vyplýva, že hoci reprezentácia vzájomných vzťahov podobnosti bola získaná vďaka prítomnosti atribútových častí počas trénovania, správna

odozva SOM v testovacej fáze sa generuje i pri absencii $\boldsymbol{x}_a$, t.j. len na základe symbolovej časti.

V druhej ukážke je kontext symbolu reprezentovaný pomocou iných symbolov, ako to možno pozorovať v prirodzenom jazyku. Uvažovaná množina 30 symbolov zahrňovala podstatné mená, slovesá a príslovky. Generované trénovacie vzory pozostávali zo zmysluplných trojslovných viet (napr. Robo pomaly beží, lev je mäso, atď.), pričom každý z troch symbolov bol nejako kódovaný ako 7-rozmerný vektor. Opäť, aby sa zvýraznil vplyv kontextu, bol parameter a v symbolovej časti stanovený na $a = 0{,}2$. Po natrénovaní na 2000 (21-rozmerných) vstupných vzoroch tvaru $\boldsymbol{x}=[\boldsymbol{x}_s,\boldsymbol{x}_a]^T$ boli návestia určené opäť len na základe symbolovej časti a výsledkom je mapa na obr. 7.26b.

Z uvedených príkladov vyplýva, že SOM možno použiť i na generovanie topologicky usporiadaných zobrazení symbolických dát za predpokladu, že tie sú prezentované v kontexte, ktorý nejakým spôsobom popisuje vzťahy podobnosti medzi nimi.

7.9 Príbuzné algoritmy

I napriek úspešnosti použitia SOM v rôznych aplikáciách má algoritmus SOM niektoré nedostatky. Patria k nim najmä tieto fakty:

- forma štruktúry, dimenzia mapy i počet neurónov SOM sú definované *a priori*
- možnosť vzniku "mŕtvych" neurónov
- diskrétnosť a "rovnomernosť" projekcie.

Pri použití SOM sa *a priori* predpokladá, že vstupné dáta ležia (najčastejšie) na dvoj- alebo jednorozmernom podpriestore, čo je nutná ale nepostačujúca podmienka úspešnej použiteľnosti SOM, a podľa toho sa volí dimenzia mapy (v prípade 2D obyčajne štvorcového tvaru). Ak majú dáta naozaj odhadovanú dimenziu a navyše im "vyhovuje" štandardná štruktúra mapy (myslia sa tým bežne používané formy reťaze a mriežky), potom SOM predstavuje efektívny a ilustratívny prostriedok zobrazenia ich topologických vzťahov, čo možno využiť napr. pri následnej klasifikácii v tomto výstupnom priestore. Použitie SOM ako prostriedku na analýzu dát je však obmedzené. Obmedzenosť jej všeobecného použitia spočíva práve v tom, že vyžaduje apriórny odhad inherentnej dimenzie dát — informácie, ktorá má byť práve jedným z výsledkov ich analýzy! Takisto, zvolený počet neurónov nemusí byť optimálny: príliš málo neurónov znižuje presnosť aproximácie, ich privysoký počet spôsobuje nárast výpočtovej zložitosti s možnosťou neadekvátneho zvýšenia presnosti aproximácie.

Fixovanie formy štruktúry (štvorcová, obdĺžniková) môže mať v závislosti od štruktúry dát za následok vznik tzv. "mŕtvych" neurónov (napr. ak dáta pozostávajú zo vzdialených zhlukov), t.j. takých, ktorých váhové vektory zakotvia v oblastiach s nulovou pravdepodobnosťou výskytu vstupov. Inými slovami, ide o neuróny, ktoré nikdy nezvíťazia, preto ostávajú nevyužité.

Diskrétnosť a "rovnomernosť" projekcie znamená, že konkrétny vstupný vektor sa premietne na jediný (víťazný) neurón, ktorého súradnice v mape môžu nadobúdať len rovnomerne vzdialené diskrétne hodnoty (1,2,...). V niektorých aplikáciách môže byť táto skutočnosť nevýhodou (nedostatočná presnosť), čo je na druhej strane možné eliminovať plošným zvýšením počtu neurónov.

Uvedené fakty boli podnetmi pre návrh príbuzných algoritmov samo-organizácie v snahe odstrániť tieto nedostatky, alebo aspoň niektoré z nich. Stručne sa o niektorých z nich zmienime.

VQP (Vector Quantization and Projection) [13]. Výstižný názov napovedá, že vstupné dáta sú najprv vektorovo kvantované (vo vstupnom priestore), a potom projektované do priestoru nižšej dimenzie, a to s cieľom zachovania topológie. Spoločnými znakmi so SOM je to, že dimenziu výstupného priestoru a počet neurónov *treba tiež a priori zvoliť*, avšak odlišnosť spočíva v tom, že VQP *nepredpokladá* žiadnu geometrickú štruktúru neurónov (teda ani funkcia okolia tu nefiguruje). Každému neurónu i je priradený vstupný vektor $\boldsymbol{x}_i \in \Re^N$ (odpovedajúci váhovému vektoru $\boldsymbol{w}_i$ v SOM) a výstupný vektor $\boldsymbol{y}_i \in \Re^P$ (odpovedajúci vektoru $\boldsymbol{r}_i$ v SOM), pričom $P<N$. Neuróny na výstupe teda nemajú zafixované pozície, navyše v pravidelnej štruktúre ako u SOM, ale určenie ich optimálnych pozícií je práve cieľom algoritmu VQP. Prvým krokom je vektorová kvantizácia vstupného priestoru (t.j. nájdenie prototypov $\boldsymbol{x}_i$), na ktorú autori aplikovali vylepšenú modifikáciu algoritmu "neural gas" [34]. Následná projekcia prototypov, ktorej kritériom je zachovanie lokálnej topológie, sa odvádza z minimalizácie chybovej funkcie tvaru

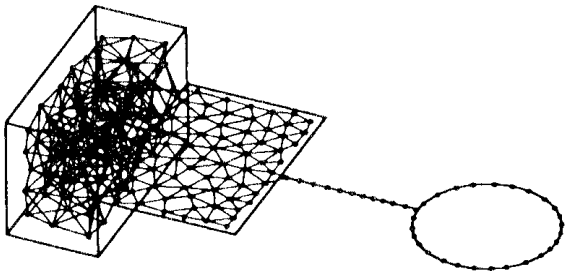
$$E = \frac{1}{2}\sum_i \sum_{j \neq i} (X_{ij} - Y_{ij})^2 . F(Y_{ij}) \tag{7.31}$$

pričom $X_{ij}=d(\boldsymbol{x}_i,\boldsymbol{x}_j)$ a $X_{ij}=d(\boldsymbol{x}_i,\boldsymbol{x}_j)$, kde $d(.,.)$ predstavuje euklidovskú vzdialenosť. $F(.)$ je pozitívna, monotónne klesajúca funkcia, aby podporovala zachovanie lokálnej topológie: čím menšia vzájomná vzdialenosť dvoch bodov na výstupe, tým väčší je ich príspevok v chybovej funkcii. Výsledkom štandardnej gradientovej metódy je vzťah pre $\Delta \boldsymbol{y}_i = \alpha \sum_{j \neq i} G(X_{ij}, Y_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)$, z čoho vyplýva výpočtová zložitosť rádu $O(n^2)$, lebo ako vidieť, adaptujú sa všetky neuróny, pričom posun každého z nich sa vypočítava ako sumácia cez všetky ostatné neuróny. V snahe znížiť výpočtovú zložitosť pri zachovaní presnosti dospeli autori k myšlienke vybrať v každom adaptačnom kroku len jeden neurón (víťaza pre aktuálny vstup) a adaptovať všetky ostatné neuróny podľa pravidla (toho istého, ale bez sumácie)

$$\Delta \boldsymbol{y}_j = \alpha \frac{X_{ij} - Y_{ij}}{Y_{ij}} \left[2F(Y_{ij}) - (X_{ij} - Y_{ij}).F'(Y_{ij}) \right] (\boldsymbol{y}_j - \boldsymbol{y}_i) \qquad \forall j \neq i \tag{7.32}$$

kde F' označuje deriváciu. Tým sa výpočtová zložitosť znížila na $O(n)$. V porovnaní so SOM je algoritmus VQP rýchlejší, a vďaka "voľnej" geometrii výstupných vektorov je schopný aproximovať dátové štruktúry bez vzniku "mŕtvych" neurónov.

TRN (Topology Representing Network) [35]. Na tento algoritmus sa možno pozerať ako na aproximáciu dátovej štruktúry pomocou (neorientovaného) grafu. Jeho výsledkom sú pozície uzlov (neurónov), ktorých počet n sa *vopred stanoví* (ako u SOM), a štvorcová matica spojení $C[n \times n]$ definujúca existenciu hrán medzi nimi (t.j. ak hrana medzi uzlami i, j existuje, potom $C_{ij}>0$, inak $C_{ij}=0$), pričom podmienkou je zachovanie topológie (v zmysle



Obrázok 7.27. Komplexná dátová štruktúra pozostávajúca z jedno-, dvoj- a trojrozmernej časti, aproximovaná grafom TRN (prevzaté z [35]).

definície uvedenej v podkapitole 7.6). Avšak na rozdiel od SOM i VQP, výstupná reprezentácia (pozície uzlov) má dimenziu vstupného priestoru, teda *N*. Inými slovami, generovaná projekcia (na uvažovaný graf) nie je sprevádzaná explicitnou redukciou dimenzie. TRN je kombináciou algoritmu "neural gas" [34], ktorým sa adaptujú uzly, a Hebbovho učenia so súťažením,[12] ktoré slúži na vytváranie nových resp. elimináciu existujúcich hrán. Adaptácia uzlov sa realizuje podľa vzťahu (podobného ako u SOM)

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t).\exp(-k_i / \lambda(t)).\left[\boldsymbol{x} - \boldsymbol{w}_i(t)\right] \tag{7.33}$$

kde parametre $\alpha(t)$ a $\lambda(t)$ sú funkcie monotónne klesajúce v čase, k_i udáva počet neurónov *j,* pre ktoré pri danom vstupe $\boldsymbol{x}$ platí: $\|\boldsymbol{x} - \boldsymbol{w}_j\| < \|\boldsymbol{x} - \boldsymbol{w}_i\|$. Takto sa v každej iterácii stanoví sekvencia neurónov $i_0,i_1,...,i_{n-1}$ (podľa blízkosti k aktuálnemu vstupu) a modifikácia prvkov matice *C,* ktoré boli pri inicializácii nastavené na 0, sa realizuje nasledovnou trojicou krokov:

(i) ak $C_{i_0 i_1} = 0$, spoj uzly i_0 a i_1, t.j. priraď $C_{i_0 i_1} > 0$

(ii) resetuj "vek" hrany $i_0 - i_1$, t.j. $t(i_0, i_1) \leftarrow 0$ a zvýš "vek" všetkých spojení s víťazom, t.j. $t(i_0, j) \leftarrow t(i_0, j) + 1 \;\; \forall j \mid C_{i_0 j} > 0$

(iii) zruš všetky spojenia s víťazom, ktorých vek presiahol limit *T,* t.j. nastav $C_{i_0 j} \leftarrow 0, \;\; \forall j \mid (C_{i_0 j} > 0) \wedge (t(i_0, j) > T)$.

Pri modifikácii spojení (hrán) $C_{i_0 i_1}$ podľa vzťahu $\Delta C_{i_0 i_1} \propto y_i . y_j$ sa výstup neurónov počíta ako $y_i = f(\|\boldsymbol{x} - \boldsymbol{w}_i\|)$, kde *f* je kladná monotónne klesajúca funkcia. Uvedený algoritmus umožňuje skonštruovať graf, ktorý je schopný aproximovať i zložité štruktúry s rôznou dimenziou v jednotlivých jeho častiach, ako napr. na obr. 7.27.

GCS (Growing Cell Structures) [19]. Zatiaľ čo doteraz spomínané modely pracovali s vopred stanoveným, konštantným počtom neurónov, v GCS sa neuróny i pridávajú resp.

[12] Hebbovo učiace pravidlo so súťa□ením v podstate predstavuje syntézu dvoch princípov: korelačného a kompetičného.

uberajú. Algoritmus takto umožňuje aproximovať širšie spektrum distribúcií dát pri zachovaní lokálnej topológie, avšak limitujúcim faktorom ostáva to, že výsledný graf je monodimenzionálny, pričom topológiu grafu treba na začiatku zvoliť.[13] Navyše, výstupná reprezentácia má tú istú dimenziu ako vstupný priestor.

Algoritmus GCS v sebe zahŕňa tri kroky: redistribúcia neurónov (uzlov), pridanie neurónu a odstránenie neurónu. Redistribúcia sa realizuje v každej iterácii, podobne ako u SOM. Rozhodnutie o následnom pridaní neurónu je založené na nasledovnej myšlienke: každý uzol i má priradenú chybovú premennú $err(i)$, ktorá sa v prípade jeho víťazstva inkrementuje o jeho vzdialenosť od aktuálneho vstupu, t.j. $err(i) \leftarrow err(i) + \|\boldsymbol{x} - \boldsymbol{w}_i\|$. Príliš vysoká hodnota *err(i)* takto signalizuje, že v danej oblasti grafu je nízka hustota neurónov (uzlov). Preto sa nájde "čierna ovca" (*bs*) medzi neurónmi, t.j. ten s najväčšou hodnotou *err(i)*, a jeho najvzdialenejší bezprostredný sused f (t.j. taký, ktorý má s ním spojenie). Do stredu medzi ne sa vloží nový neurón (*nn*): $\boldsymbol{w}_{nn} = (\boldsymbol{w}_{bs} + \boldsymbol{w}_f)/2$, a vytvoria sa jeho spojenia s okolitými neurónmi tak, aby sa zachovala topológia k-simplexov. Následne je ešte potrebné adaptovať chybové premenné.

Odstránenie neurónu je potrebné najmä pri aproximovaní nespojitej distribúcie dát. Vychádza sa z úvahy, že čím dlhšie neurón nezvíťazil, tým väčšia je pravdepodobnosť, že bude odstránený.

DCS (Dynamic Cell Structures) [6]. Algoritmus DCS je zlúčením a rozšírením myšlienok obsiahnutých vo modeloch TRN a GCS. Hlavný rozdiel oproti GCS spočíva v tom, že topológia konštruovaného grafu nemá vopred danú dimenziu, čím sa dáva možnosť vzniku multidimenzionálneho grafu (t.j. s rôznou dimenziou v jeho častiach, ako na obr. 7.27). Nové neuróny sa vkladajú podobným spôsobom ako u GCS: medzi neurón i s najväčšou hodnotou $err(i)$ a jeho bezprostredného suseda j s druhou najväčšou hodnotou $err(j)$; nie však do stredu medzi ne, ale s proporcionálnym posunom podľa pomeru hodnôt ich chybových funkcií. Víťazný neurón sa adaptuje ako u SOM, neuróny s ním susediace podľa vzťahu

$$\Delta \boldsymbol{w}_j = \alpha(t).A_{i^*j}.(\boldsymbol{x} - \boldsymbol{w}_j) \qquad \forall j: A_{i^*j} > 0,\ \ j = 1,2,\ldots,n, \tag{7.34}$$

pričom A_{i^*j} sú prvky matice susednosti (pozri definíciu, podkapitola 7.6), ktoré sa modifikujú trochu zložitejším spôsobom ako u TRN:

$$A_{ij}(t+1) = \begin{cases} 1 & \text{ak } y_i.y_j = \max\{y_k.y_l\},\ \ k,l = 1,2,\ldots,n \\ 0 & \text{ak } A_{ij}(t) < \theta \\ \varepsilon.A_{ij}(t) & \text{v ostatných prípadoch.} \end{cases} \tag{7.35}$$

[13] Graf s k-dimenziálnou topológiou pozostáva z k-simplexov, èo je jeho základný "stavebný blok", ktorý vznikne vzájomným pospájaním k+1 uzlov.

Tabuľka 7.3. Charakteristiky algoritmov samoorganizácie neurónových sietí na formovanie zobrazení zachovávajúcich topológiu. Pod "redukciou dimenzie" sa myslí zníženie dimenzie reprezentácie výstupnej informácie, "n=konšt." označuje konštantný, vopred stanovený počet neurónov v sieti. "Monodimenzionalita grafu" znamená, že graf má v celej svojej štruktúre rovnakú dimenziu *k*, t.j. že pozostáva len zo simplexov *k*-tého rádu. "Fixovanosť topológie" sa vzťahuje na apriórne určenie štruktúry grafu.

Algoritmus	Redukcia dim.	n = konšt.	Monodim. graf	Fix. topológia
SOM	áno	áno	áno	áno
VQP	áno	áno	áno	nie
TRN	nie	áno	nie	nie
GCS	nie	nie	áno	nie
DCS	nie	nie	nie	nie
GSOM	áno	nie	áno	nie

Vo vzťahu (7.35) $y_i = f(\|\boldsymbol{x} - \boldsymbol{w}_i\|)$ označuje výstup *i*-teho neurónu, kde f je kladná monotónne klesajúca funkcia, ε predstavuje konštantu zabúdania, ktorá spĺňa podmienku $0 < \varepsilon < 1$, a θ označuje prah eliminácie spojenia.

GSOM (Growing SOM) [3]. Tento algoritmus taktiež umožňuje pridávať neuróny (podobne ako GCS a DCS), avšak s obmedzujúcou podmienkou, že štruktúra usporiadania neurónov ostáva pravidelnou — v tvare (viacrozmernej) mriežky. Z toho vyplýva nutnosť pridávať nielen jednotlivé neuróny, ale podľa potreby i celé "pásy" či "vrstvy" neurónov, ak sa algoritmus "rozhodne" zvýšiť dimenziu grafu (napr. k existujúcej reťazi neurónov možno pridať buď jeden neurón, alebo celý pás neurónov, čím sa z reťaze stane mriežka). Motiváciou pre takúto dynamiku nárastu siete je fakt, že vďaka uchovanej pravidelnosti je práca s takouto sieťou podstatne jednoduchšia ako so všeobecným grafom, a navyše, GSOM redukuje dimenziu popisu dát.

Porovnaním spomínaných algoritmov samoorganizácie z hľadiska ich charakteristických čŕt dospejeme k prehľadovej tabuľke 7.3. Z pohľadu vyššie uvedených charakteristík sa najuniverzálnejším algoritmom spomedzi spomínaných javí algoritmus DCS. Umožňuje pretransformovať vstupné dáta do podoby neorientovaného grafu, ktorý svojou štruktúrou verne odpovedá dátam v každej časti vstupného priestoru. Takáto univerzálnosť výstupnej štruktúry je na jednej strane výhodou, avšak súčasne sa s tým podstatne sťaží práca s takouto výstupnou reprezentáciou. Je to práve kvôli jej nepravidelnosti, preto je nutné spracovávať a uchovávať komplikované vzťahy medzi jednotlivými uzlami grafu. Naopak, SOM je po tejto praktickej stránke veľmi jednoduchá. To si možno uvedomiť už pri jej zobrazovaní ako pravidelnej mriežky, vo fáze programovania algoritmu, či v následnej implementácii. GSOM predstavuje kompromis — zachováva si jednoduchosť výstupnej reprezentácie SOM (samotný algoritmus je však pochopiteľne zložitejší), a súčasne je pružnejšia pri aproximácii štruktúry dát.

Literatúra

[1] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, and D.E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3(3):277-290, 1990.

[2] H.-U. Bauer and K.R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks,* 3(4):570-579, 1992.

[3] H.-U. Bauer and T. Villmann. Growing a hypercubical output space in a self-organizing feature map. *Technical report TR-95-030*, ICSI Berkeley, California, 1995.

[4] C. Bouton, M. Cottrell, J.C. Fort, and G. Pagés. Self-organization and convergence of the Kohonen algorithm. In: *Probabilités Numériques* (Eds. N. Bouleau and D. Talay), INRIA, Paris, France, 163-180, 1991.

[5] W.D. Brandt, H. Behme, and H.W. Strube. Bildung von Merkmalen zur Spracherkennung mittels phonotopischer Karten. In: *Fortschritte der Akustik-DAGA 91*, Bad Honnef, Germany, 1057-1060, 1991.

[6] J. Bruske and G. Sommer. Dynamic cell structures learns perfectly topology preserving map. *Neural Computation*, 7:845-865, 1995.

[7] M. Budinich and J.G. Taylor. On the ordering conditions for self-organizing maps. In: *Proc. of ICANN'94* (Eds. M. Marinaro and P.G. Morasso), Springer-Verlag, London, UK, I. 347-349, 1994.

[8] M. Cottrell, J.C. Fort, and G. Pagés. Two or three things that we know about the Kohonen algorithm. *Technical report No.31*, Université Paris 1, France, 1994.

[9] D.A. Critchley. Stable states, transitions and convergence in Kohonen self-organizing maps. In: *Proc. of ICANN'92*, (Eds. I. Alexander and J. Taylor), North-Holland, Brighton, UK, 281-284, 1992.

[10] P. Demartines. *Analyse de données par réseaux de neurones auto-organisés*. Doktorská dizertačná práca, L'Institut National Polytechnique de Grenoble, France, 1994.

[11] P. Demartines. Organization measures and representations of Kohonen maps. In: *First IFIP Working Group 10.6 Workshop* (Ed. J. Hérault), 1992.

[12] P. Demartines and F. Blayo. Kohonen self-organizing maps: Is the normalization necessary? *Complex Systems*, 6:105-123, 1992.

[13] P. Demartines and J. Hérault. Representation of nonlinear data structures through fast VQP neural network. In: *Neuronîmes,* 411-424, 1993.

[14] E. Erwin, K. Obermayer, and K. Schulten. Convergence properties of self-organizing maps. In: *Artificial Neural Networks*, (Eds. T. Kohonen et al), Elsevier, Amsterdam, Netherlands, 409-414, 1991.

[15] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: Ordering, converge properties and energy functions. *Biological Cybernetics*, 67(1):47-55, 1992.

[16] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: Stationary states, metastability and convergence rate. *Biological Cybernetics*, 67(1):35-45, 1992.

[17] I. Farkaš. *On Vector-coded Feature Mapping Using Self-organizing Neural Maps*. Doktorská dizertačná práca, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave, 1995.

[18] E.A. Ferrán. An ordering theorem that allows for ordering changes. In: *Artificial Neural Networks 2,* (Eds. I. Alexander and J. Taylor), North-Holland, Amsterdam, Netherlands, I:165-168, 1992.

[19] B. Fritzke. Growing cell structures — a self organizing network for unsupervised and supervised training. *Neural Networks,* 7(3):1441-1459, 1994.

[20] G.J. Goodhill, S. Finch, and T.J. Sejnowski. Quantifying neighbourhood preservation in topographic mappings. *Technical report INC-9505*, Institute for Neural Computation, La Jolla, CA, 1995.

[21] R.M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1:4-29, April 1984.

[22] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.

[23] E.I. Knudsen, S. du Lac, and S.D. Esterly. Computational maps in the brain. *Annual Review of Neuroscience*, 10:41-65, 1987.

[24] T. Kohonen. Analysis of a simple self-organizing proces. *Biological Cybernetics*, 44(2):135-140, 1982.

[25] T. Kohonen. *Self-Organization and Associative Memory*. Springer, 1988.

[26] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):56-69, 1982.

[27] T. Kohonen. Self-organizing maps: Optimization approaches. In: *Artificial Neural Networks* (Eds. T.Kohonen et al.), North-Holland, Amsterdam, Netherlands, II:981-990, 1991.

[28] T. Kohonen. Speech recognition based on topology-preserving neural maps. In: *Neural Computing Architectures*, (Ed. I. Alexander), North Oxford Academic Publishers, 26-40, 1989.

[29] T. Kohonen. Statistical pattern recognition revisited. In: *Advanced Neural Computers* (Ed. R. Eckmiller), Elsevier Science Publ. B.V., North-Holland, 137-144, 1990.

[30] T. Kohonen. *Self-Organizing Maps*. Springer, 1995.

[31] Z.-P. Lo, Y. Tu, and B. Bavarian. Analysis of the convergence properties of topology preserving neural networks. *IEEE Transactions on Neural Networks*, 4(2):207-220, 1993.

[32] C. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*,14:85-100,1973.

[33] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296-317, 1995.

[34] T. Martinetz and K. Schulten. "Neural gas" network learns topologies. In: *Artificial Neural Networks*, vol.I, (Eds. T.Kohonen et al.), North-Holland, 397-402, 1991.

[35] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(3):505-522, 1994.

[36] G. Pagés. Voronoi tesselation, space quantization algorithms and numerical integration. *Proc. of ESANN'93*, D facto conf. services, Brussels, Belgium, 221-228, 1993.

[37] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-organizing Maps: An Introduction,* Addison-Wesley, 1992.

[38] H. Ritter and K. Schulten. Kohonen self-organizing maps: Exploring their computational capabilities. *Proc. IEEE ICNN*, San Diego, 109-116, 1988.

[39] H. Ritter and K. Schulten. On the stationary state of Kohonen's self-organizing sensory mapping. *Biological Cybernetics*, 54(2):99-106, 1986.

[40] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241-254, 1989.

[41] P. Růžička. On convergence of learning results for topological maps. *Neural Network World*, 4:413-424, 1993.

[42] W. Siedlecki, K. Siedlecka, and J. Sklansky. Mapping techniques for exploratory pattern analysis. In: *Pattern Recognition and Artificial Intelligence*, (Eds. E.S. Gelsema and L.N. Kamnal), North-Holland, Amsterdam, Netherlands, 277-299, 1988.

[43] A. Takeuchi and S. Amari. Formation of topographic maps and columnar microstructures in nerve fields. *Biological Cybernetics*, 35:63-72, 1979.

[44] J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley, 1974.

[45] J.A. Walter and K. J. Schulten. Implementation of self-organizing neural networks for visuo-motor control of an industrial robot. *IEEE Transactions on Neural Networks*, 4(1):86-95, 1993.

[46] D.J. Willshaw and C. von der Malsburg. How patterned neural connections can be set up by self-organization. *Proc. of the Royal Society of London* B, 194:431-445, 1976.

[47] H. Yin and N.M. Allinson. On the distribution and convergence of feature space in self-organizing maps. *Neural Computation*, 7:1178-1187, 1995.

8. Hopfieldov model

8.1 Úvod

Pri štúdiu štatistických nelineárnych kooperatívnych systémov, akými sú napr. spinové sklá, prišli fyzici na myšlienku využiť mnohé ich zaujímavé vlastnosti na vytvorenie idealizovaných neurónových sietí, ktorých správanie sa možno interpretovať ako analógie rôznych mozgových (či psychických) funkcií, ako sú napr. asociatívne vyvolávanie z pamäti, spracovanie časových postupností stimulov, zabúdanie, a iné. Prvými, ktorí poukázali na analógiu medzi procesmi prechodu z neusporiadaných do usporiadaných stavov v magnetických látkach a procesmi, ktoré by mohli prebiehať v reálnych neurónových sieťach, boli Cragg a Temperley [17, 18] a Little [57]. V stabilnom stave by podľa nich priestorové usporiadanie domén (oblastí) atómových spinov s orientáciou "hore" a domén spinov orientovaných "dole" korešpondovalo s usporiadaním oblastí aktivovaných a neaktivovaných neurónov v sieti pri vnútornej reprezentácii nejakého zapamätaného vzoru. Po publikovaní článku Johna Hopfielda [40], ktorého model predstavuje interpretáciu Sherringtonovho-Kirkpatrickovho a Isingovho modelu magnetika [44, 48] v zmysle neurónovej siete, nastala v nasledujúcom desaťročí "explózia" štúdia takto vytvorených modelov neurónových sietí. Vďaka vyvinutému fyzikálnemu aparátu patrí Hopfieldov model a jeho modifikácie medzi teoreticky najlepšie preštudované modely neurónových sietí.

Hopfieldove neurónové siete sa často označujú aj ako **atraktorové** alebo **autoasociatívne** neurónové siete. Patria do triedy tzv. celulárnych (bunečných) automatov, čo sú vo všeobecnosti dynamické systémy pozostávajúce z veľkého množstva dvojstavových (alebo viacstavových) prvkov, navzájom viazaných, s definovaným pravidlom na zmenu stavov prvkov, ktorých makroskopické správanie sa je popísané v ideálnom prípade analytickými rovnicami. Autoasociatívne siete sú iba jedným z mnohých prístupov k vytváraniu umelých neurónových sietí a k modelovaniu mozgových funkcií. Podľa nášho názoru netreba rôzne prístupy chápať ako vzájomne sa vylučujúce, ale skôr z toho hľadiska, že každý z nich lepšie vyjadruje či popisuje iné aspekty pestrej variety vlastností a činností, ktorými sa mozog vyznačuje. Na Hopfieldovych autoasociatívnych neurónových sieťach nás zaujali najmä dve veci. Po prvé, ich výpočtové vlastnosti možno interpretovať v neurobiologických súvislostiach ako model predpokladaných procesov, ktoré prebiehajú v mozgu pri kognitívnom spracovávaní informácií (takýmto kognitívnym spracovávaním je napr. vnímanie) [6, 12, 13]. Po druhé, všetky ich výpočtové vlastnosti sa vynárajú (angl. *emerge*) ako dôsledok paralelnej činnosti veľkého množstva navzájom interagujúcich jednoduchých procesorov (modelových neurónov). Autoasociatívne siete môžu tvoriť most medzi "mikroskopickými" modelmi neurónov a ich sietí, založenými na káblovej teórii [45, 53], a "makroskopickými" modelmi, ktoré sú zamerané na hierarchickú organizáciu mozgovej činnosti. Pritom sa javí atraktívnou možnosť, že mnohé javy

pozorované na vyšších úrovniach spracovávania informácií v jednotlivých oblastiach mozgu by sa mohli dať pripísať emergentným (spontánne sa objavujúcim) kolektívnym vlastnostiam sietí zložených z veľkého počtu neurónov.

8.2 Základný popis

Základný popis Hopfieldovho modelu autoasociatívnej neurónovej siete (viď obr. 8.1) a dynamiky jej časového vývoja je založený na analógii s Isingovou magnetickou teóriou a Sherringtonovom-Kirkpatrickovom modeli spinového skla [9, 44, 48]:

(1) **Neurón** (analógia Isingovho spinu atómu) môže byť v jednom z dvoch stavov, t.j. $S_i \in \{-1,+1\}$. Nech N je celkový počet neurónov v sieti. Vstupný a zároveň aj výstupný stav (**konfigurácia aktivity**) siete je vyjadrený N-rozmer-ným binárnym vektorom $\boldsymbol{S}=(S_1,S_2,...,S_N)$.

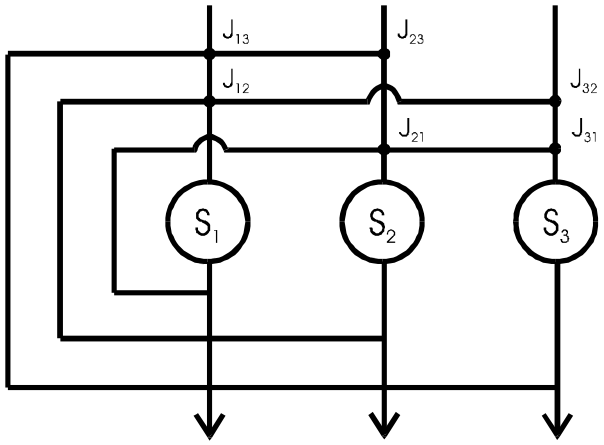
(2) J_{ij} (angl. *junction*) je **váha synapsy** tvorenej j-tym neurónom na i-tom neuróne (analógia interakčnej konštanty medzi i-tym a j-tym spinom). Pre excitačnú synapsu platí $J_{ij} > 0$ a pre inhibičnú synapsu $J_{ij} < 0$. Neurón na sebe samom synapsy netvorí, t.j. $J_{ii} = 0$. Suma príspevkov od jednotlivých neurónov S_j, pričom $j=1,...,N$, váhovaná prostredníctvom synaptických váh J_{ij}, vyjadruje postsynaptický potenciál, ktorý je mierou excitácie i-teho neurónu. **Postsynaptický potenciál** sa zvykne značiť ako h_i^{int} (analógia vnútorného magnetického poľa).

$$h_i^{\text{int}} = \sum_{j=1}^{N} J_{ij} S_j \tag{8.1}$$

(3) Neurón sa aktivuje, t.j. generuje na svojom výstupe akčný potenciál, ak postsynaptický potenciál h_i^{int} prekročí istú hodnotu prahového napätia , tzv. **prah excitácie** neurónu. Táto veličina sa tiež zvykne značiť h_i^{ext} ako jej analógia, vonkajšie pole pôsobiace na spin v magnetiku. Celkový **efektívny postsynaptický potenciál** neurónu je potom $h_i = h_i^{\text{int}} - h_i^{ext}$.

(4) **Deterministické prechodové pravidlo** pre zmenu stavu i-teho neurónu je dané týmto predpisom

$$S_i \rightarrow S_i' = \text{sign}(h_i) = \text{sign}\left(\sum_{j=1}^{N} J_{ij} S_j - h_i^{ext} \right) \tag{8.2}$$



Obrázok 8.1. Schéma Hopfieldovej neurónovej siete s dvojstavovými prvkami inšpirovanej spinovými sklami. Vstup a výstup siete je reprezentovaný konfiguráciou stavov všetkých prvkov. Výstup každého prvku pôsobí na všetky ostatné prvky cez synaptické váhy. Keď sieť naštartujeme z akejkoľvek počiatočnej konfigurácie aktivity, a necháme ju dynamicky sa vyvíjať, skonverguje do jedného z bodových atraktorov v jej stavovom priestore.

pričom funkcia sign(x) je definovaná takto:

$$\mathrm{sign}(x) = \begin{cases} +1 & \text{pre} \quad x > 0\,, \\ -1 & \text{pre} \quad x < 0\,. \end{cases} \tag{8.3}$$

Stavy neurónov môžu byť $S_i \in \{-1,+1\}$, vonkajšie pole kladieme obyčajne rovné nule, takže $h_i \neq 0$. Aktualizovanie stavu neurónov podľa vzťahu (8.2) môže prebiehať dvoma spôsobmi. Prvým je **synchrónna (paralelná) dynamika**, keď všetky neuróny menia svoj stav naraz, t.j. v čase t platí

$$S_i(t) = \mathrm{sign}\left(\sum_{\substack{j=1 \\ j \neq i}}^{N} J_{ij} S_j(t-1) - h_i^{ext}(t) \right) \quad , \quad \text{pre} \quad i = 1,\ldots,N\,. \tag{8.4}$$

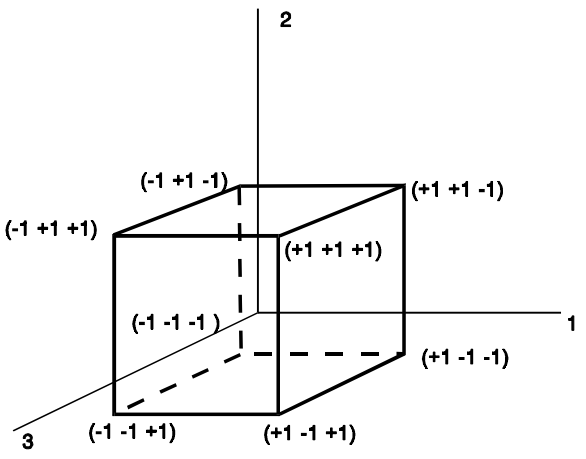
Jeden cyklus relaxácie (prechod) odpovedá aktualizácii stavu všetkých N neurónov.

Druhou možnosťou je **asynchrónna (sekvenčná) dynamika**, keď v každom časovom momente t mení svoj stav len jeden náhodne vybratý neurón i. To znamená, že v N krokoch za sebou aktualizujeme vždy znova náhodne vybratý neurón podľa vzťahu

$$S_i(t) = \mathrm{sign}\left(\sum_{\substack{j=1 \\ j \neq i}}^{N} J_{ij} S_j(t) - h_i^{ext}(t) \right) \quad . \tag{8.5}$$

Časový vývoj Hopfieldovej siete s N neurónmi, čiže sekvenciu stavov siete $\boldsymbol{S}(t)=(S_1(t),S_2(t),...,S_N(t))$ v čase, možno chápať ako trajektóriu idúcu cez vrcholy N-rozmernej hyperkocky, ktorá má 2^N možných vrcholov (obr. 8.2).
V prípade asynchrónnej dynamiky sú v jednom časovom kroku povolené iba prechody pozdĺž hrán do najbližších vrcholov hyperkocky prislúchajúcich stavom, ktoré sa od pôvodného líšia len hodnotou 1 spinu. Jeden **cyklus relaxácie (prechod)** siete odpovedá aktualizácii stavu neurónov v N krokoch.

(5) **Energia** danej konfigurácie aktivity $\boldsymbol{S}$ (hamiltonián systému) je definovaná ako



Obrázok 8.2. Ilustrácia stavového priestoru pre Hopfieldovu sieť s N=3 diskrétnymi neurónmi, ktorá sa môže nachádzať v jednom z 8 rôznych stavov, vrcholov kocky.

$$E(\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} J_{ij}S_iS_j - \sum_{i=1}^{N} S_i h_i^{ext} \quad . \tag{8.6}$$

8.3 Spontánna evolúcia Hopfieldovej siete

Výsledky pozorovaní **spontánnej evolúcie (relaxácie)** siete na základe vlastných počítačových simulácií a na základe výsledkov uvedených v [9, 40] zhrnieme do nasledujúcich bodov. Sieť štartovala z náhodného počiatočného stavu, t.j. počiatočné stavy neurónov S_i boli neurónom priradené náhodne, pričom $S_i = \pm 1$ s pravdepodobnosťou rovnou 0,5. Prahy excitácie h_i^{ext} boli buď všetky položené rovné nule alebo boli vygenerované ako malé náhodné čísla z intervalu $(-1,+1)$. Synaptické váhy J_{ij} boli tiež vygenerované ako náhodné čísla z intervalu $(-1,+1)$, a potom sa sieť nechala vyvíjať podľa vzťahu (8.4) alebo (8.5).

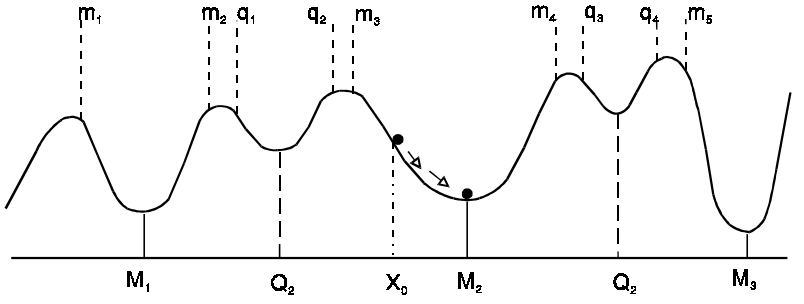
Evolúcia siete v priestore stavov silno závisí na tom, či je matica váh $\boldsymbol{J}$ symetrická ($J_{ij} = J_{ji}$) alebo asymetrická ($J_{ij} \neq J_{ji}$), na konkrétnych hodnotách J_{ij} a na tom, či je prechodová dynamika synchrónna (rovnica 8.4) alebo asynchrónna (rovnica 8.5). Tá istá sieť (s tou istou maticou váh $\boldsymbol{J}$ a tými istými prahmi h_i^{ext} má úplne inú evolúciu vtedy, keď má synchrónnu dynamiku, ako vtedy, keď má asynchrónnu dynamiku

Pre asynchrónnu a synchrónnu prechodovú dynamiku môžeme rozlíšiť tri typy asymptotického (dlhodobého) správania sa siete:

Chaotické trajektórie. Sieť "blúdi" v stavovom priestore. Energia siete v tvare (8.6) nepravidelne stúpa a klesá. Takéto správanie sa je typické pre prípad synchrónnej dynamiky siete s asymetrickou synaptickou maticou $\boldsymbol{J}$ ($\boldsymbol{J} \neq \boldsymbol{J}^{\mathrm{T}}$) a ľubovoľnými prahmi excitácie, $h_i^{\mathrm{ext}} \in (-1,+1)$. Už veľmi malá zmena v počiatočnom stave, napr. preklopenie 1 neurónu vedie k inej trajektórii v stavovom priestore. Chaotické trajektórie sú veľmi citlivé na počiatočné stavy siete.

Limitné cykly. Trajektórie, ktoré rýchlo (okolo 5 prechodov) vedú do malých cyklov, zložených najčastejšie z 2, prípadne 4 stavov. Cykly sú typické pre synchrónnu dynamiku, ale zriedka sa vyskytnú aj pri asynchrónnej dynamike. Sú teda citlivé na typ dynamiky siete.

Bodové atraktory. Trajektórie rýchlo vedú do jedného, potom už sa stále opakujúceho vzorca aktivity celej siete, $\boldsymbol{\xi}$. Z ktoréhokoľvek počiatočného stavu sa sieť po niekoľkých prechodoch dostane do jediného stavu, v ktorom už zostane aj napriek tomu, že jej stav sa aktualizuje podľa rovníc (8.4) alebo (8.5). Takýto stav sa nazýva bodový atraktor. Pre jednu a tú istú maticu synaptických váh $\boldsymbol{J}$ existuje v stavovom priestore niekoľko bodových atraktorov. Bodové atraktory zodpovedajú lokálnym minimám energie siete vyjadrenej rovnicou (8.6) — pozri obr. 8.3. Evolúcia do bodového atraktoru je pomerne nezávislá na počiatočnom stave siete v tom zmysle, že mnohé počiatočné stavy siete končia v tom istom bodovom atraktore. Okolo každého atraktora je oblasť takých stavov, ktoré všetky konvergujú do toho istého bodového atraktora a nie do iného. Táto oblasť sa nazýva **spádová oblasť** atraktora. Každá matica váh $\boldsymbol{J}$ má iné bodové atraktory.



Obrázok 8.3. Schematické znázornenie "povrchu" energie Hopfieldovej siete s lokálnymi minimami zodpovedajúcimi pravým a falošným pamäťovým stavom. (označenými písmenami M resp. Q). Malé písmená *m* a *q* označujú hranice spádových oblastí pre pravé resp. falošné atraktory. Počiatočný stav X_0 konverguje do najbližšieho atraktora.

Evolúcia siete do bodových atraktorov je typická pre asynchrónnu dynamiku. V prípade asynchrónnej dynamiky, keď je matica váh symetrická, t.j. (J_{ij}=J_{ji}), a $h_i^{ext} = 0 \text{ pre } \forall i$, ***energia*** v tvare (8.6) ***monotónne klesá***, pokým nedosiahne minimum. Odvodíme si to nasledovným postupom. Energiu takejto Hopfieldovej siete v ľubovoľnom časovom okamihu môžeme rozpísať takto:

$$\begin{aligned} E &= -\tfrac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq m}^{N} J_{ij}S_iS_j - \tfrac{1}{2}\sum_{i=1}^{N} J_{im}S_iS_m - \tfrac{1}{2}\sum_{j=1}^{N} J_{mj}S_mS_j \\ &= -\tfrac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq m}^{N} J_{ij}S_iS_j - \sum_{j=1}^{N} J_{mj}S_mS_j \end{aligned} \quad . \qquad (8.7)$$

Najskôr sme z celkovej sumy vyňali dva členy, ktoré prislúchajú *m*-tému neurónu, ktorý bol náhodne vybratý a bude aktualizovaný podľa vzťahu (8.5). V ďalšej úprave sme využili to, že synaptická matica je symetrická, a teda platí J_{mj}=J_{jm}. Preto môžeme tieto dva vyňaté členy pre *m*-tý neurón zlúčiť dohromady (je jedno, či v prvej sume pre *m*-tý neurón sumujeme cez *i* alebo cez *j*). Nový stav vybratého neurónu označme ako S'_m. Po zmene stavu jedného neurónu bude nová energia *E'* rovná

$$\begin{aligned} E' &= -\tfrac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq m}^{N} J_{ij}S_iS_j - \tfrac{1}{2}\sum_{i=1}^{N} J_{im}S_iS'_m - \tfrac{1}{2}\sum_{j=1}^{N} J_{mj}S'_mS_j \\ &= -\tfrac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq m}^{N} J_{ij}S_iS_j - \sum_{j=1}^{N} J_{mj}S'_mS_j \end{aligned} \quad . \qquad (8.8)$$

Teraz odčítame (8.7) od (8.8) a dostaneme rozdiel energií v dvoch po sebe nasledujúcich časových krokoch:

$$\Delta E = E' - E = (S_m - S'_m) \sum_{j \neq m}^{N} J_{mj} S_j = (S_m - S'_m)\ h_m = -\Delta S_m h_m \ . \qquad (8.9)$$

V týchto úpravách sme využili vzťah (8.1) a fakt, že prahy excitácie sú nulové, takže platí $h_m = h_m^{int}$. Spomeňme si na prechodové pravidlo (8.2), definíciu funkcie sign(x), a môžeme analyzovať všetky štyri možnosti zmeny stavu jedného neurónu, a čo z toho vyplýva pre zmenu energie:

$$\begin{array}{lllllll} (1) & S_m = +1 & a & S'_m = +1 & (\text{vtedy} \quad h_m > 0) & \Rightarrow & \Delta E = 0 \\ (2) & S_m = -1 & a & S'_m = -1 & (\text{vtedy} \quad h_m < 0) & \Rightarrow & \Delta E = 0 \\ (3) & S_m = +1 & a & S'_m = -1 & (\text{vtedy} \quad h_m < 0) & \Rightarrow & \Delta E < 0 \\ (4) & S_m = -1 & a & S'_m = +1 & (\text{vtedy} \quad h_m > 0) & \Rightarrow & \Delta E < 0 \end{array} \qquad (8.10)$$

Dokázali sme, že pre asynchrónnu dynamiku, keď je matica váh symetrická, t.j. (J_{ij}=J_{ji}), a h_i^{ext}= 0 pre $\forall i$, platí vždy $\Delta E \leq 0$, a teda energia Hopfieldovej siete v tvare (8.6) v priebehu relaxácie monotónne klesá, pokým nedosiahne minimum.

Existencia viacerých rôznych bodových atraktorov je dôsledkom frustrácie väzieb medzi neurónmi. Frustrácia väzieb znamená, že neexistuje taká konfigurácia siete, v ktorej by stavy jednotlivých neurónov zodpovedali polarite všetkých svojich väzieb. To bráni tomu, aby sieť dospela do jediného stavu, v ktorom by mala energia (8.6) svoje globálne minimum, ako by sa stalo v nefrustrovanom systéme. Frustrácia väzieb teda spôsobuje rôznosť atraktorov, v ktorých má energia siete (8.6) svoje lokálne minimá. V stavovom priestore existujú aj tzv. **falošné atraktory** (viď obr. 8.3). Konfigurácie aktivity siete prislúchajúce falošným atraktorom sú rôznymi lineárnymi kombináciami konfigurácií prislúchajúcich pravým atraktorom. Falošné atraktory ležia energeticky vyššie ako pravé atraktory a sú obklopené bariérami stavov s vyššou energiou. Ak je dynamika siete taká, že monotónne minimalizuje energiu (8.6), môže sa stať, že sa systém ocitne vo falošnom atraktore, a nemôže sa dostať von.

Iste je namieste otázka, ktorý typ dynamiky, synchrónna (8.4) alebo asynchrónna (8.5), vernejšie vystihuje modelovanú skutočnosť — prácu neurónov v mozgu. Stochastická povaha činnosti neurónov je lepšie vystihnutá v asynchrónnej dynamike, s náhodným výberom jednotlivých neurónov, ktoré aktualizujú svoj stav [63]. Jeden cyklus relaxácie Hopfieldovej siete možno interpretovať nasledovným spôsobom. Absolútna refraktérna doba v neurobiológii je minimálna doba kľudu medzi generovaním dvoch akčných potenciálov za sebou. Je daná charakteristikami neurónovej membrány a jej trvanie je $\Delta t = 1-2$ ms [46]. Pri modelovaní na počítači možno tento interval rozdeliť na N subintervalov, t.j. $\Delta t = N\delta t$. Po uplynutí každého δt sa s pravdepodobnosťou $1/N$ vyberie jeden neurón, ktorý bude aktualizovať svoj stav podľa vzťahu (8.5). Takto zabezpečíme, že medzi dvoma po sebe nasledujúcimi okamihmi generovania akčného potenciálu má každý neurón prestávku $\Delta t = 1-2$ ms.

8.4 Autoasociatívna pamäť

Všetky modelované kognitívne udalosti (pamäť, učenie, atď.) sa odohrávajú na úrovni celej siete, ktorá reprezentuje nejakú **populáciu neurónov** určenú pre danú úlohu. Opakujúci sa vzorec aktivity (**bodový atraktor**) je stacionárnym stavom siete. **Stacionárne stavy** siete predstavujú **pamäťové stavy** siete. To, ktorý stav siete je bodovým atraktorom, čiže jej pamäťovým stavom, je determinované maticou váh synaptických spojení ***J***. Učenie, t.j. reprezentácia nových vzorov v sieti, je spojené so zmenami hodnôt synaptických váh. Táto vlastnosť súhlasí so súčasnou predstavou o tom, že mechanizmus učenia sa a uchovávania informácií v mozgu spočíva v zmenách účinnosti synaptických spojení medzi neurónmi [14, 26, 46].

Hopfieldova sieť dospeje do tej ktorej konfigurácie (zodpovedajúcej príslušnému bodovému atraktoru) na základe podobnosti tejto výslednej konfigurácie s počiatočnou konfiguráciou siete vyvolanou externou (vstupnou) stimuláciou. Ako sme už spomenuli, každému atraktoru (pravému i falošnému) prislúcha tzv. spádová oblasť, čo je priestor stavov, v ktorom keď sa sieť ocitne, vždy sa deterministicky vyvinie do príslušného atraktora (obr. 8.3). To znamená, že neurónová sieť je schopná nájsť príslušný atraktor (pamäťový stav) aj vtedy, keď sa jej prezentuje neúplný, resp. deformovaný vstupný vzor. Táto vlastnosť sa nazýva **autoasociatívna (obsahom adresovaná) pamäť** (angl. *content addressable memory*), a je dominantnou vlastnosťou idealizovaných neurónových sietí Hopfieldovho typu.

Z hľadiska použitia Hopfieldovej siete ako modelu kognitívnych funkcií je veľmi dôležité, že synaptické váhy sa dajú navrhnúť (skonštruovať) tak, aby sa bodovými atraktormi stali vopred vybraté konfigurácie siete. Špecifický predpis pre konštrukciu váh J_{ij} je tento [9, 40]:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p} \xi_i^\mu \, \xi_j^\mu & \text{pre} \quad i \neq j\,, \\ 0 & \text{pre} \quad i = j\,. \end{cases} \qquad (8.11)$$

N-rozmerné binárne vektory $\boldsymbol{\xi}^\mu = (\xi_1^\mu, \xi_2^\mu, \ldots, \xi_N^\mu)$, $\mu = 1, \ldots, p$, predstavujú zvolené pamäťové konfigurácie a ich celkový počet je p. J_{ij} môže nadobudnúť $2p+1$ rôznych hodnôt z intervalu $\langle -p/N, +p/N \rangle$. Tento predpis je jedným z viacerých možných formálnych vyjadrení tzv. Hebbovho pravidla [36] pre zmenu synaptických váh. **Hebbovo pravidlo** hovorí, že váha synapsy rastie, ak oba neuróny spojené touto synapsou sú zároveň aktívne, a naopak váha synapsy klesá, ak je aktivita týchto dvoch neurónov nekorelovaná (nie je synchrónna). V procese učenia sa (resp. zapamätávania si viacerých vzorov) sa jednotlivé synapsy modifikujú podľa časového spriemernenia minulej aktivity neurónov. Formálne to možno vyjadriť takto: $\Delta J_{ij} \propto \left\langle S_i S_j \right\rangle_t$. Symbol $\langle \ldots \rangle_t$ značí časové spriemernenie súčinu aktivity pre- a postsynaptického neurónu počas intervalu nejakého minulého časového obdobia dĺžky t. V prípade nášho predpisu (8.10) je zmena ΔJ_{ij} lineárna, t.j. $\Delta J_{ij} = (1/N)\xi_i^\mu \xi_j^\mu$. Proces učenia v atraktorových sieťach predstavuje samostatnú bohatú problematiku teoretického štúdia, ktorá nie je ešte ani zďaleka vyčerpaná [3, 4, 5, 23, 67].

Nech sa sieť vo svojej evolúcii dostane do stavu zodpovedajúcemu jednej z pamäťových konfigurácií, napr. ξ^ν, čo znamená, že pre $\forall i$ platí $S_i = \xi_i^\nu$. Interpretujeme to ako vybavenie si (angl. *recall*) daného vzoru z pamäti (obr. 8.4). Vzniká otázka, či skutočne predpis na konštrukciu synaptických váh (8.11) garantuje, že zvolený vzor ξ^ν je naozaj stabilný vzor. Inými slovami, keď sa sieť do takéhoto stavu dostane, nemôže sa z neho dostať von? Podmienka, že určitý stav ξ^ν je dynamicky stabilný je taká, že lokálne pole *i*-teho neurónu musí mať to isté znamienko ako stav *i*-teho neurónu. **Podmienka stability** pre ľubovoľný vzor ξ^ν vyjadrená matematicky je (viď rovnicu 8.3)

$$\xi_i^\nu h_i^\nu > 0 \qquad \text{pre } \forall i \ . \tag{8.12}$$

Prepíšeme vzťah (8.12) tak, že do ľavej strany tejto nerovnosti najskôr dosadíme vzťah (8.1), a do tohto vzťahu namiesto J_{ij} dosadíme vzťah (8.11):

$$\xi_i^\nu h_i^\nu = \xi_i^\nu \sum_{j=1}^{N} J_{ij}\xi_j^\nu = \xi_i^\nu \tfrac{1}{N}\sum_{j=1}^{N}\sum_{\mu=1}^{p}\xi_i^\mu\xi_j^\mu\xi_j^\nu = \sum_{\mu=1}^{p}\xi_i^\nu\xi_i^\mu\left(\tfrac{1}{N}\sum_{j=1}^{N}\xi_j^\mu\xi_j^\nu\right) \quad > 0 \ . \tag{8.13}$$

Predposlednú sumu na ľavej strane rozdelíme na dva členy, prvý zodpovedajúci vzoru $\mu = \nu$, a druhý zodpovedajúci ostatným vzorom $\mu \neq \nu$, takto:

$$\begin{aligned}\xi_i^\nu h_i^\nu &= \sum_{\mu=1}^{p}\xi_i^\nu\xi_i^\mu\left(\tfrac{1}{N}\sum_{j=1}^{N}\xi_j^\mu\xi_j^\nu\right) = \xi_i^\nu\xi_i^\nu\left(\tfrac{1}{N}\sum_{j=1}^{N}\xi_j^\nu\xi_j^\nu\right) + \sum_{\mu\neq\nu}^{p}\xi_i^\nu\xi_i^\mu\left(\tfrac{1}{N}\sum_{j=1}^{N}\xi_j^\mu\xi_j^\nu\right) \\ &= \frac{N-1}{N} + \sum_{\mu\neq\nu}^{p}\xi_i^\nu\xi_i^\mu\left(\tfrac{1}{N}\sum_{j=1}^{N}\xi_j^\mu\xi_j^\nu\right) \quad > 0.\end{aligned} \tag{8.14}$$

V limite pre veľké *N* dostaneme

$$\xi_i^\nu h_i^\nu = 1 + \xi_i^\nu \sum_{\mu\neq\nu}^{p}\xi_i^\mu\left(\frac{1}{N}\sum_{j=1}^{N}\xi_j^\mu\xi_j^\nu\right) = 1 + C_i^\nu \quad > 0 \ . \tag{8.15}$$

Druhý člen, C_i^ν, predstavuje tzv. presluch (angl. *crosstalk*). Ak je C_i^ν=0, môžeme vysloviť záver, že vzor $\boldsymbol{\xi}^\nu$ spĺňa podmienku stability (8.12). C_i^ν sa rovná nule vtedy, keď sú pamäťové vzory navzájom ortogonálne, t.j. keď ich skalárny súčin je rovný nule, teda

$$\boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi}^\nu = \frac{1}{N}\sum_{j=1}^{N}\xi_j^\mu\xi_j^\nu = 0 \qquad \text{pre} \quad \mu \neq \nu \ . \tag{8.16}$$

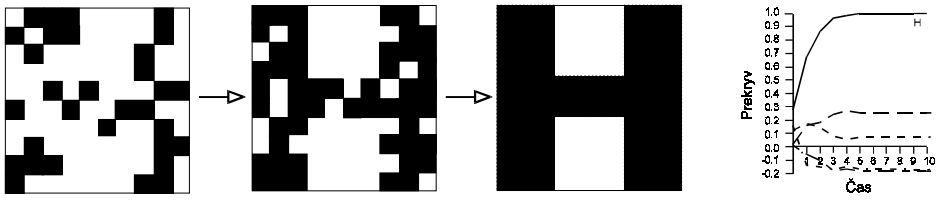
Keď sú pamäťové vzory navzájom ortogonálne, maximálny počet vzorov, ktoré možno uchovať v Hopfieldovej sieti (pamäťová kapacita) $p_{\max} \to N$. Podmienka stability (8.12) je splnená aj v prípade, keď je $\left|C_i^{\nu}\right|$<1. Táto situácia nastáva vtedy, keď sú pamäťové vzory pseudoortogonálne (t.j. keď stredná hodnota ich skalárnych súčinov, ustrednená cez všetky dvojice vzorov $\langle\langle \boldsymbol{\xi}^{\mu} \cdot \boldsymbol{\xi}^{\nu} \rangle\rangle$, je približne rovná nule) a zároveň keď p<<N. Z toho vyplýva, že takisto $p_{\max}$<<N. V týchto dvoch prípadoch sú všetky pamäťové vzory stabilné, t.j. keď je sieť naštartovaná v ktoromkoľvek z nich, aj v ňom zostane. Okrem toho je sieť schopná opraviť určité percento neurónov, ktoré sa na začiatku nachádzajú v nesprávnych stavoch, takže sieť relaxuje do správneho pamäťového vzoru (viď obr. 8.4). To znamená, že vybraté pamäťové vzory sú naozaj atraktormi systému, ktorý funguje ako autoasociatívna pamäť.

Ako **pozorovateľná premenná** sa pri Hopfieldovych neurónových sieťach najčastejšie volí prekryv $m^{\mu}(t)$ okamžitej konfigurácie siete $\boldsymbol{S}(t)$ s pamäťovými stavmi $\boldsymbol{\xi}^{\mu}$, definovaný ako

$$m^{\mu}(t) = \frac{1}{N}\sum_{j=1}^{N} \xi_j^{\mu} S_j(t) \quad , \qquad \text{pre} \quad \mu = 1,...,p\,. \tag{8.17}$$

Prekryv $m^{\mu}(t)$ je vlastne **miera podobnosti** dvoch konfigurácií. Pomocou časového vývoja okamžitých prekryvov s každým z pamäťových stavov (8.17) môžeme sledovať časovú evolúciu siete. Pamäťové stavy ξ^{μ} sú atraktormi deterministickej siete a sú v sieti uložené pomocou predpisu (8.11). Po krátkom čase sa sieť dostane do jedného z atraktorov a príslušný prekryv bude $m^{\nu} = 1$.

Pamäťová kapacita siete $p_{\max}$ sa dá odvodiť nasledovným spôsobom. C_i^{ν} z rovnice (8.15) závisí len na pamäťových vzoroch ξ^{μ}, ktoré chceme uchovať v sieti. Uvažujme, že



Obrázok 8.4. Ilustrácia činnosti Hopfieldovej autoasociatívnej pamäti. Sieť má 100 neurónov, 5 pamäťových vzorov (5 písmen) a deterministickú asynchrónnu dynamiku. Na vstupe je poškodené písmeno H, ktoré sieť vďaka svojej dynamike perfektne zrekonštruuje. Graf ilustruje vývoj prekryvov aktuálneho stavu siete s jednotlivými pamäťovými vzormi. Písmenom H je označená krivka odpovedajúca vývoju prekryvu pre písmeno H, ostatné krivky vyjadrujú vývoj prekryvu s ostatnými vzormi. Čísla na časovej osi odpovedajú cyklom relaxácie.

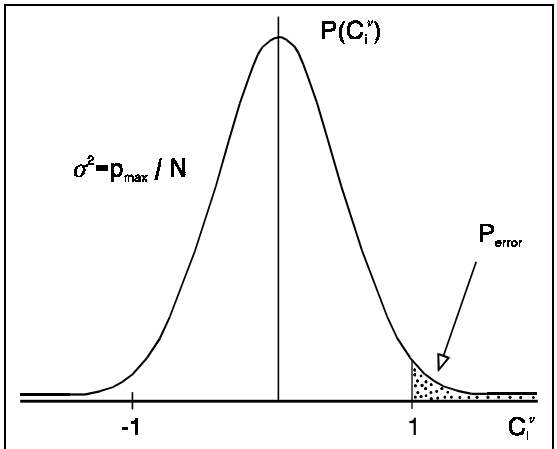
tieto vzory sú čisto náhodné konfigurácie, teda že stavy $\xi_j^{\mu} = +1$ a $\xi_j^{\mu} = -1$ sú generované s rovnakou pravdepodobnosťou. Potom môžeme určiť pravdepodobnosť P_{error}, že ľubovoľný bit (neurón) je nestabilný, ako

$$P_{error} = P(C_i^{\nu} > 1) \ . \tag{8.18}$$

P_{error} závisí na počte neurónov N a počte vzorov p. Predpokladajme, že $N >> 1$ aj $p >> 1$, čo je typický prípad a umožňuje nám to použiť vzťahy z matematickej štatistiky. Potom C_i^{ν} je $1/N$ krát suma približne Np náhodných čísiel, z ktorých každé má hodnotu +1 alebo -1 (pozri vzťah (8.15)). V matematickej štatistike [25] bolo odvodené, že takáto náhodná premenná má binomické rozloženie pravdepodobnosti so strednou hodnotou nula a s varianciou $\sigma^2=p/N$. Ale keďže Np je dostatočne veľké, môžeme binomické rozdelenie aproximovať Gaussovym rozdelením s nulovou strednou hodnotou a tou istou varianciou (obr. 8.5).

P_{error} sa rovná veľkosti vybodkovanej oblasti pod grafom Gaussovej krivky na obr. 8.5, a tak môžeme písať

$$P_{error} = \frac{1}{\sigma\sqrt{2\pi}} \int_1^{\infty} e^{-x^2/2\sigma^2} dx = \frac{1}{2}\left[1 - erf(1/\sqrt{2\sigma^2})\right] = \frac{1}{2}\left[1 - erf(\sqrt{N/2p})\right] . \tag{8.19}$$



Obrázok 8.5. Gaussovo rozdelenie pravdepodobnosti výskytu jednotlivých hodnôt presluchového člena v rovnici (8.15).

Chybová funkcia (angl. *error function*) $erf(x)$ je definovaná ako

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-u^2) du \ . \tag{8.20}$$

V tabuľke 8.1 sú uvedené hodnoty $p_{\max}/N$ prislúchajúce jednotlivým hodnotám P_{error}. Napríklad ak si zvolíme, že ľubovoľný bit sa môže preklopiť s pravdepodobnosťou $P_{error} < 0{,}01$, dostávame $p_{\max} \leq 0{,}185N$. Toto číslo tiež hovorí, že asi 1,85% neurónov bude spočiatku nestabilných, keď štartujeme sieť z niektorého pamäťového vzoru.

Tabuľka 8.1. Pravdepodobnosť P_{error}, že ľubovoľný neurón je nestabilný, vo vzťahu k pamäťovej kapacite Hopfieldovej siete $p_{\max}/N$.

P_{error}	$p_{\max}/N$
0,001	0,105
0,0036	0,138
0,01	0,185
0,05	0,37
0,1	0,61

Sofistikovanejšia analýza situácie by ukázala [38], že Hopfieldova sieť funguje skutočne spoľahlivo ako autoasociatívna pamäť pre $p_{\max} = 0{,}138N$. Ak sa budeme pokúšať zapamätať v sieti viac vzorov, výsledkom bude "výpadok pamäti" (angl. *blackout*) a môže sa nám stať, že sieť si nebude pamätať žiaden z predpísaných vzorov. Analýza kapacity autoasociatívnych neurónových sietí, čiže maximálneho počtu pamäťových konfigurácií, ktoré možno v sieti uchovať tak, aby boli vybaviteľné z pamäti, tiež tvorí bohatú časť ich štúdia [19, 20, 28-31, 47]. Kapacita siete sa vyšetruje napr. vzhľadom na to, do akej miery sú pamäťové vzory navzájom korelované, ďalej vzhľadom na predpis ich uchovania, ktorý nemusí mať vždy tvar (8.11), atď. Napríklad do tohto predpisu možno zabudovať kontinuálne učenie a zabúdanie [59].

V tomto kontexte teda vybavenie si nejakého vzoru z pamäti predstavuje proces, v ktorom sieť znovu vytvára (obnovuje) tú istú aktivitu neurónov, ktorá sa v minulosti opakovane šírila v sieti pri zapamätávaní si daného vzoru. Vzorec aktivity siete (pamäťová konfigurácia siete) je vnútornou reprezentáciou nejakého vonkajšieho podnetu. Tento vonkajší podnet môže byť "jednoduchý" vstup z nejakého senzorického orgánu alebo komplexný vstup zložený z aktivity prichádzajúcej z jednej alebo viacerých iných neurónových sietí.

8.5 Stochastický Hopfieldov model

Ukázali sme, že Hebbovo pravidlo učenia (8.11) nám dáva (pre dostatočne malé p) dynamický systém s bodovými atraktormi, ktoré sú totožné s vopred zvolenými stavmi siete ξ^μ a ktoré zodpovedajú lokálnym minimám energie systému (8.6). Avšak tieto stavy nie sú jedinými atraktormi systému. Po prvé, všetky **reverzné konfigurácie** $-\xi^\mu$ sa automaticky tiež stávajú pamäťovými stavmi, ktoré zodpovedajú tým istým minimám energie ako ξ^μ. Je to preto, lebo predpis pre konštrukciu synaptických váh (8.11) a výraz pre energiu systému (8.6) sú oba perfektne symetrické čo sa týka zámeny $S_i \leftrightarrow -S_i$. Existencia týchto atraktorov odpovedajúcich reverzným stavom nás až tak netrápi a považujeme ich za pravé atraktory.

Po druhé, sú tu však aj tzv. **zmiešané stavy** ξ^{mix}, ktoré sa nerovnajú ani jednému zapamätanému vzoru, ale lineárnej kombinácii nepárneho počtu pamäťových vzorov. Najjednoduchší prípad je takáto symetrická kombinácia troch pamäťových vzorov:

$$\xi_i^{mix} = sign(\pm\xi_i^{\mu_1} \pm \xi_i^{\mu_2} \pm \xi_i^{\mu_3}), \qquad \text{pre } \forall i \ . \tag{8.21}$$

Vďaka rôznym kombináciám znamienok dostaneme osem rôznych kombinácií troch pamäťových vzorov. Na to, aby sme si overili, či zmiešaný stav (8.21) je skutočne stabilný, treba overiť či platí (8.12). Napríklad pre takú kombináciu (8.21), kde sa vyskytujú len znamienka +, môžeme dospieť k takémuto výsledku:

$$\begin{aligned} \xi_i^{mix} h_i^{mix} &= \xi_i^{mix} \frac{1}{N} \sum_{\mu=1}^{3} \sum_{j=1}^{N} \xi_i^{\mu} \xi_j^{\mu} \xi_j^{mix} \\ &\approx (\xi_i^1 + \xi_i^2 + \xi_i^3)\ sign(\xi_i^1 + \xi_i^2 + \xi_i^3) + presluch \quad > 0\,. \end{aligned} \tag{8.22}$$

Môžeme vidieť, že podmienka stability (8.12) je pre tento zmiešaný stav splnená (ak je p dostatočne malé). Podobne môžeme kombinovať 5, 7, a viac vzorov. Systém si nevyberá ako falošné atraktory kombinácie z párneho počtu pamäťových vzorov, lebo ich príspevky sa môžu pre niektoré i vynulovať (pozri vzťah 8.21), čo nie je prípustné, keďže stavy neurónov sú ± 1.

Po tretie, v stavovom priestore existuje ešte jeden druh falošných atraktorov, ktoré nie sú kombináciou žiadneho konečného počtu pamäťových vzorov [9-11]. Tieto stavy sa nazývajú **stavy spinového skla**, kvôli ich príbuznosti so stabilnými stavmi spinových skiel.

Ako vidíme, takýto model pamäti nie je dokonalý. Okrem nami vytvorených miním energie sa tam objavujú aj minimá zodpovedajúce rozličným ďalším stavom. Hoci teória a simulácie ukázali, že spádové oblasti týchto falošných atraktorov sú oveľa užšie ako spádové oblasti pravých atraktorov, je veľmi žiadúce sa ich zbaviť. Veľmi účinným spôsobom ako sa zbaviť falošných atraktorov je zaviesť do systému šum. Pritom však pravé atraktory zostávajú naďalej dobrými atraktormi. Okrem toho, **zavedenie šumu** predstavuje aj ďalšie priblíženie sa biologickej realite, pretože reálne neuróny pracujú v "zašumenom" prostredí. Zdroje šumu sú rôzne: stochastická povaha uvoľňovania neuromediátorov, fluktuácie postsynaptických potenciálov, spontánne generovanie akčných potenciálov, atď. Je zaujímavé, že šum, ktorý v iných informačných systémoch máva negatívne dôsledky, a preto sa ho snažíme eliminovať, v Hopfieldovych neurónových sieťach plní vyslovene pozitívnu úlohu a pritom ich viac približuje k biologickej realite.

Uvažujme teda Hopfieldovu neurónovú sieť, v ktorej sa stavy neurónov nemenia podľa deterministického prechodového pravidla (8.3), ale podľa nejakého pravdepodobnostného (stochastického) pravidla. Toto stochastické pravidlo si teraz odvodíme. Majme v systéme nejaký bližšie nešpecifikovaný zdroj šumu, ktorého sila je parametrizovaná parametrom T, čo je analógia teploty termostatu, s ktorým je daný fyzikálny systém v kontakte. Na popis Hopfieldovho modelu pri nenulovej teplote T použijeme štatisticko-fyzikálny prístup [9, 38]. Predpokladajme, že systém po určitom čase dospeje do rovnovážneho stavu. V rovnovážnom stave je systém popísaný distribučnou funkciou Gibbsovho kanonického systému. Teda pravdepodobnosť výskytu konfigurácie $\boldsymbol{S}$ je rovná

$$P(\boldsymbol{S}) = \frac{1}{Z}\exp\left(-\beta(\boldsymbol{S})\right), \quad \text{kde} \quad Z = \sum_{\boldsymbol{s}} \exp\left(-\beta E(\boldsymbol{S})\right) = \exp(-\beta E) + \exp(-\beta E') \tag{8.23}$$

je stavová suma (partičná funkcia), a $\beta = 1/T$. Keďže ľubovoľný stav $\boldsymbol{S}$', ktorý vznikol so stavu $\boldsymbol{S}$, sa líši len v stave jedného neurónu m, tak pravdepodobnosť prechodu do

ľubovoľného stavu $\boldsymbol{S}'$, ktorý vznikol zo stavu $\boldsymbol{S}$ preklopením m-tého neurónu, je daná vzťahom

$$P(\boldsymbol{S} \to \boldsymbol{S}') = \frac{\exp(-\beta E')}{\exp(-\beta E') + \exp(-\beta E)} = \frac{1}{1 + \dfrac{\exp(-\beta E)}{\exp(-\beta E')}} = \frac{1}{1 + \exp(\beta \Delta E)} \quad , \tag{8.24}$$

kde zjavne $E = E(\boldsymbol{S}), E' = E'(\boldsymbol{S}'),$ a $\Delta E = E' - E$. Teraz si spomeňme na vzťahy (8.9) a (8.10) a môžeme písať

$$\Delta E = E' - E = (S_m - S'_m)\ h_m = \begin{cases} -2 h_m S'_m & \text{ak} \quad S_m = -S'_m\,, \\ 0 & \text{ak} \quad S_m = S'_m\,. \end{cases} \tag{8.25}$$

Pomocou vzťahu (8.25) teraz vyjadríme $P(\boldsymbol{S} \to \boldsymbol{S}')$ takto

$$P(\boldsymbol{S} \to \boldsymbol{S}') = \frac{1}{1 + \exp(-2\beta\, h_m S'_m)} \ . \tag{8.26}$$

Tento vzťah zovšeobecníme pre ľubovoľný stav $\boldsymbol{S}$, ktorý vznikol aktualizáciou m-tého neurónu. Zároveň si uvedomme, že táto pravdepodobnosť je vlastne totožná s pravdepodobnosťou, že m-tý neurón zmení svoj stav, a teda

$$P(S_m = \pm 1) = \frac{1}{1 + \exp(-2\beta\, h_m S_m)} \ . \tag{8.27}$$

Rovnica (8.27) je **stochastické pravidlo** na zmenu stavu jedného neurónu. Z tejto rovnice vyplýva

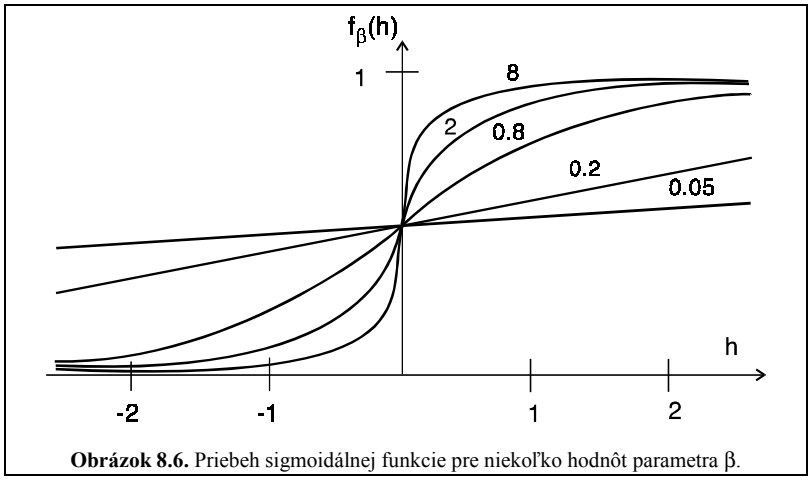
$$P(S_m = +1) = \frac{1}{1 + \exp(-2\beta\, h_m)} \ , \tag{8.28}$$

a tiež

$$P(S_m = -1) = \frac{1}{1 + \exp(+2\beta\, h_m)} = 1 - P(S_m = +1) \ . \tag{8.29}$$

Vzťah (8.28) je analógiou tzv. Glauberovej dynamiky v Isingovom modeli [32]. Vyjadruje priebeh sigmoidálnej (logistickej) funkcie (obr. 8.6).

V limite pre $T \to 0$ a $\beta \to \infty$, sa (8.28) mení na deterministické pravidlo (8.3). V limite $T \to \infty$ a $\beta \to 0$ je $P(S_m = +1) = 0{,}5$. Všetky stavy sú rovnako prav-depodobné, systém sa stáva ergodickým a nemôže ďalej pracovať ako autoasociatívna pamäť. Z toho vyplýva, že musí existovať nejaký interval teplôt (hodnôt šumu), pre ktorý je stochastická Hopfieldova sieť stále spoľahlivou autoasociatívnou pamäťou.



Obrázok 8.6. Priebeh sigmoidálnej funkcie pre niekoľko hodnôt parametra β.

V rovnovážnom stave pri nenulovej teplote systém nedospeje do jedného stabilného stavu, do jednej stabilnej konfigurácie. Očakáva sa, že pri dostatočne malých teplotách bude jeho rovnovážny stav charakterizovaný hodnotami parametrov usporiadania, ktoré sú definované ako stredné hodnoty termodynamických pozorovateľných premenných — prekryvov (8.17). Po istom počte prechodov, keď sieť dospeje do rovnovážneho stavu, bude konfigurácia siete oscilovať okolo niektorého z atraktorov, ktorý je v stavovom priestore blízko počiatočnej konfigurácie. Parametre usporiadania sú časové stredné hodnoty prekryvov (8.17) stavov siete $\boldsymbol{S}(t)$ s danými atraktormi ξ^{μ} definované ako

$$\langle m^{\mu}(t)\rangle = \frac{1}{N}\sum_{i=1}^{N}\xi_i^{\mu}\langle S_i(t)\rangle \quad , \qquad \text{pre} \quad \mu = 1,\ldots,p\,. \tag{8.30}$$

Pre atraktor, ktorý si stochastická sieť "vyberie" býva stredná hodnota prekryvu (8.30) väčšia ako 0,9, ale nedosahuje hodnotu 1. Hodnota prekryvu (8.30) s ostatnými pamäťovými stavmi osciluje blízko nuly. V Hopfieldovej neurónovej sieti so šumom nemožno hovoriť o atraktórnych stavoch, ale len o atraktórnej pravdepodobnostnej distribúcii stavov.

Podmienky, ktoré umožňujú pretrvávanie rovnovážnej distribúcie (8.23), môžeme lepšie uvidieť, keď si odvodíme **selfkonzistentné rovnice**. Majme stochastickú sieť pozostávajúcu z N neurónov, ktorá má pri nulovej teplote $p << N$ pamäťových stavov ξ^{μ}, m =1,..., p. Dynamika siete je asynchrónna s pravdepodobnosťou zmeny stavu i-teho neurónu (8.27), matica váh je symetrická a pre synaptické váhy platí predpis (8.11). V časovej strednej hodnote prekryvu (8.30) vystupuje časová stredná hodnota stavu i-teho neurónu $\langle S_i(t)\rangle$, pre ktorú platí

$$\langle S_i(t)\rangle = \sum_{S} S_i\, P(S_i) = (+1)\times P(S_i = +1) + (-1)\times P(S_i = -1) = \tanh\left[\beta\, h_i(t)\right] . \tag{8.31}$$

Za pravdepodobnosti $P(S_i)$ sme dosadili vzťahy (8.28) a (8.29). V ďalšom môžeme uvažovať všeobecne, že celkové pole $h_i(t) = h_i^{int}(t) + h_i^{ext}(t)$, kde $h_i^{ext}(t) = h_i^{ext}$. Ale na to, aby v rovnovážnom stave toto celkové pole pretrvávalo dlhšiu dobu, musia byť príspevky k jeho internej časti h_i^{int} (t.j. časti, ktorá je tvorená príspevkami od všetkých

ostatných neurónov v sieti) reprodukované strednými hodnotami všetkých neurónov spojených s miestom *i*. To znamená, že v (8.31) prejdeme k zámene skutočného fluktuujúceho poľa za jeho priemernú hodnotu $\langle h_i^{int}(t)\rangle$. Táto aproximácia je známa ako **teória stredného poľa** (angl. *mean-field approximation*). Sústreďujeme sa na jediný neurón a zanedbávame zmeny stavov individuálnych okolitých neurónov, ktoré berieme do úvahy len ako spriemernené pozadie.

Pre $\langle h_i^{int}(t)\rangle$ platí:

$$\langle h_i^{int}(t)\rangle = \tfrac{1}{N}\sum_{j=1}^{N} J_{ij}\langle S_j(t)\rangle = \tfrac{1}{N}\sum_{j=1}^{N}\sum_{\mu=1}^{p}\xi_i^{\mu}\xi_j^{\mu}\langle S_j(t)\rangle = \sum_{\mu=1}^{p}\xi_i^{\mu}\langle m^{\mu}(t)\rangle \ . \qquad (8.32)$$

Pri týchto úpravách sme využili vzťahy (8.11) a (8.30). Keďže J_{ii}=0, ale vzťah pre prekryv (8.30) obsahuje súčiny všetkých prvkov, pri prechode k strednej hodnote prekryvu chýba jeden člen prislúchajúci *i*-temu neurónu, rovný (p/*N*)$\langle S_i(t)\rangle$. Môžeme ho však zanedbať, lebo $p << N$. Teraz dosadíme (8.31) do (8.32), a tento výraz dosadíme do vzťahu pre strednú hodnotu prekryvu (8.30). Dostávame sústavu *N* viazaných nelineárnych rovníc (8.33), známych pod názvom **selfkonzistentné rovnice**, alebo rovnice stredného poľa [9, 62]:

$$\begin{aligned}\langle h_i^{int}(t)\rangle &= \tfrac{1}{N}\sum_{j=1}^{N} J_{ij}\langle S_j\rangle = \sum_{\mu=1}^{p}\xi_i^{\mu}\langle m^{\mu}(t)\rangle \\ &= \sum_{\mu=1}^{p}\xi_i^{\mu}\left\{\sum_{j=1}^{N}\xi_j^{\mu}\tanh\left[\beta\left(\sum_{\mu=1}^{p}\xi_j^{\mu}\langle m^{\mu}(t)\rangle + h_j^{ext}\right)\right]\right\}.\end{aligned} \qquad (8.33)$$

Zo selfkonzistentnosti vyplývajú pre rovnovážny stav tieto navzájom ekvivalentné podmienky:

(1) Priemerné vnútorné pole všetkých neurónov $\langle h_i^{int}(t)\rangle$ (8.33) má práve takú hodnotu ako vnútorné pole, ktoré dalo vznik danej distribúcii stredných hodnôt stavov neurónov $\langle S_i(t)\rangle$.

(2) Stredné hodnoty stavov $\langle S_i(t)\rangle$ vstupujú do vnútorného poľa práve s takými hodnotami, aké im dáva vnútorné pole, ktoré ich vyvolalo.

(3) Keďže $\langle h_i^{int}(t)\rangle$ (8.33) závisí len na hodnotách $\langle m^{\mu}\rangle$, $\mu = 1,\ldots,p$, tak tieto prekryvy sú reprodukované len tým vnútorným poľom $\langle h_i^{int}(t)\rangle$, ktoré ich vyvolalo.

V skutočnosti je simulovanie vybavovania vzorov z pamäti efektívne aj v tomto prípade, keď dynamika dovedie sieť iba do okolia uchovávaného vzoru a istá časť neurónov neustále mení svoj stav. Ak je časová stredná hodnota prekryvu meniacich sa konfigurácií s niektorým zo vzorov dostatočne vysoká, môžeme predpokladať, že výstupný vzorec aktivity siete reprezentuje vyvolanie daného vzoru z pamäti.

Riešeniami selfkonzistentných rovníc (8.33) sú parametre usporiadania (stredné hodnoty prekryvov). Parametre usporiadania charakterizujú rôzne fázy, v ktorých sa systém môže v rovnovážnom stave nachádzať v závislosti od teploty a histórie systému, t.j. stavu, z ktorého sieť vyštartovala. V ideálnom prípade, po dosiahnutí termodynamickej rovnováhy, rozpoznanie fázy podľa parametrov usporiadania zodpovedá obnoveniu informácie

adresovanej obsahom resp. degradovanej chybami v počiatočnom stave. Systém môže slúžiť ako pamäť, ak vďaka dynamike na ľubovoľný podnet (počiatočný stav) spontánne prechádza do stavov asociovaných s atraktorom, a tieto stavy majú významný prekryv s týmto naučeným vzorom. Zjednodušene možno interpretovať systém s ideálnymi pamäťovými vlastnosťami ako systém, ktorý z nerovnovážneho počiatočného stavu prejde do termodynamickej rovnováhy v jednej z možných fáz systému. Každá fáza zodpovedá jednému naučenému vzoru a spontánna voľba fázy je determinovaná (pri konštantnej teplote) iba počiatočnou konfiguráciou.

Teraz vysvetlíme, ako je to možné, že šum T destabilizuje falošné atraktory. Uvažujme stochastickú Hopfieldovu neurónovú sieť s $p << N$, $N \to \infty$ a $h_i^{ext} = 0$ pre $\forall i$. Pre strednú hodnotu stavu i-teho neurónu môžeme selfkonzistentné rovnice (8.31) prepísať takto:

$$\langle S_i(t)\rangle = \tanh\left(\beta \sum_{j=1}^{N} J_{ij}\langle S_j\rangle\right) = \tanh\left(\frac{\beta}{N}\sum_{j=1}^{N}\sum_{\mu=1}^{p}\xi_i^{\mu}\xi_j^{\mu}\langle S_j(t)\rangle\right) . \qquad (8.34)$$

Táto sústava nie je vo všeobecnosti riešiteľná, lebo pozostáva z N nelineárnych viazaných rovníc s N neznámymi. Pri jej riešení však môžeme postupovať heuristicky a navrhnúť hypotézu, že riešenie $\langle S_i(t)\rangle$ je úmerné jednému zo zapamätaných vzorov, tak že

$$\langle S_i(t)\rangle = m\xi_i^{\nu} \quad \text{pre} \quad \forall i . \qquad (8.35)$$

Parameter $m \in \mathbf{R}$ je konštantou úmernosti. Už sme si ukázali, že takéto stavy sú stabilné v deterministickej limite $T = 0$ (pre $m = 1$), takže je prirodzené hľadať podobné priemerné stavy v stochastickom prípade. Po dosadení (8.35) do (8.34) dostávame

$$m\xi_i^{\nu} = \tanh\left(\frac{\beta}{N}\sum_{j=1}^{N}\sum_{\mu=1}^{p}\xi_i^{\mu}\xi_j^{\mu}\, m\xi_j^{\nu}\right) . \qquad (8.36)$$

Takisto ako v prípade deterministickej siete môžeme argument funkcie tanh rozložiť na dva členy; jeden prislúchajúci ξ_i^{ν} a druhý člen vyjadrujúci presluch, obsahujúci prekryvy medzi ξ_i^{ν} a ostatnými zapamätanými vzormi. Keďže $p << N$ (a pamäťové vzory sú pseudoortogonálne), môžeme presluchový člen zaned-bať a dostaneme vzťah

$$m\xi_i^{\nu} = \tanh(\beta\, m\xi_i^{\nu}) . \qquad (8.37)$$

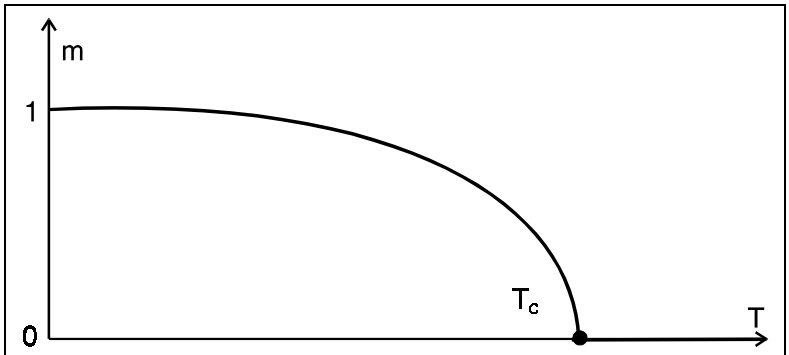
Keďže $\tanh(x) = \tanh(x)$, dostávame rovnicu na výpočet parametra m pre rovnicu (8.35):

$$m = \tanh(\beta\, m) . \qquad (8.38)$$

Grafické znázornenie riešenia rovnice (8.38) pre kladné m v závislosti od teploty T je znázornené na obr. 8.7. Z riešenia vidíme, že pre teploty väčšie ako **kritická teplota**, teda pre $T \geq T_c$ a $T_c = 1$, je $m = 0$ a žiadny pamäťový stav nie je stabilný[1]. Inými slovami, teraz vieme, že pamäťové stavy budú stabilné pre teploty $T < 1$. Chceme poukázať na to, že v správaní sa stochastickej Hopfieldovej siete je prudká skokovitá zmena pri kritickej teplote T_c. Je to ďalší príklad fázového prechodu. Mohli by sme očakávať, že správanie sa siete sa

[1] Pre $T < T_c$ je $m = 0$ tiež riešením rovnice (8.38) avšak nie je to stabilné riešenie.

spojito mení ako teplota rastie, ale v systémoch s veľkým N sa takáto závislosť často nepozoruje. Keď sa prekročí istá úroveň šumu, veľká sieť prestane fungovať ako autoasociatívna pamäť.



Obrázok 8.7. Kladné riešenie rovnice (8.38) ako funkcia teploty (šumu) T.

Podarilo sa nám ukázať, že stavy so strednou hodnotou $\langle S_i(t)\rangle$ úmernou jedinému pamäťovému vzoru (viď (8.35)) sú pri nízkych teplotách $0<T<1$ stabilné. Avšak, hoci pre $p<<N$ nie sú stavy spinového skla aktuálne, ešte stále sú tu reverzné stavy a hlavne zmiešané stavy. V prácach Amita so spolupracovníkmi [10, 11] bolo ukázané, že ***každý zmiešaný stav má svoju vlastnú kritickú teplotu*** v intervale $0<T_c \leq 0{,}46$. Keď šum prekročí túto hodnotu, zmiešané stavy už nie sú stabilné. Inými slovami, ***pre šum z intervalu 0,46 < T<1 sú falošné atraktory (zmiešané stavy) destabilizované***, a iba pamäťové stavy (a ich reverzné konfigurácie) sú stále dobrými atraktormi.

Cenou za to je nižšia vyvolávacia kvalita pamäťových vzorov a o niečo dlhšie relaxačné časy siete (doba príchodu siete do okolia atraktora). Napriek všetkému, určitá úroveň šumu zlepšuje fungovanie Hopfieldovej neurónovej siete ako autoasociatívnej pamäti. Pre vyššie hodnoty šumu, a to $T \geq 1$, sa destabilizujú aj pravé atraktory (pamäťové stavy) a systém sa stáva ergodický.

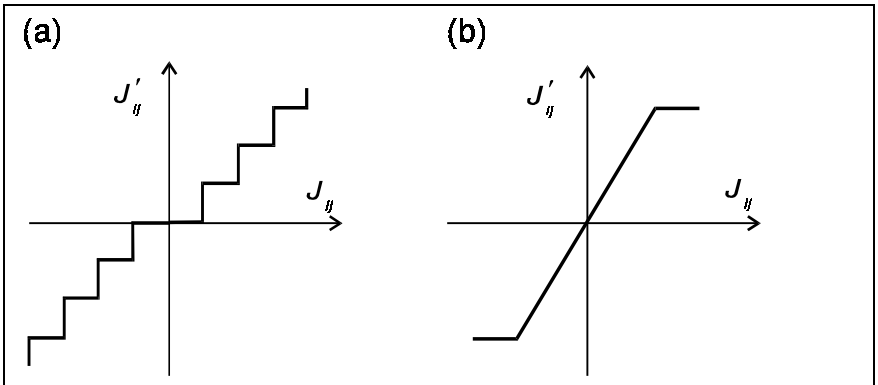
8.6 Poškodzovanie a vymazávanie synaptických spojení

Skutočné neurónové siete v mozgu dokážu vykonávať svoju funkciu aj po poškodení ich prvkov a spojení, ak tieto poškodenia nie sú príliš súvislé a priestorovo rozsiahle. V tejto časti spomenieme výsledky štúdií, ktoré sa zamerali práve na tzv. **robustnosť** resp. odolnosť **Hopfieldovych sietí voči nepresnostiam zaťažujúcim ich synaptické spojenia** vrátane ich vynulovania. Vo väčšine prípadov **kvalitatívne vlastnosti autoasociatívnej pamäti ostávajú nezmenené**, a poškodenie má iba kvantitatívne dôsledky, konkrétne ovplyvňuje najmä kapacitu siete $\alpha_c = p_{\max}/\mathrm{N}$, predlžuje dobu príchodu do atraktora, konečný prekryv s pamäťovým vzorom je o niečo menší, a pod.

Ako prvé poškodenie budeme uvažovať deviáciu hodnôt synaptických váh od hodnôt získaných pomocou Hebbovho pravidla (8.11). Táto deviácia môže byť realizovaná ako **pridanie malého náhodného čísla** k hodnote J_{ij} vypočítanej podľa vzťahu (8.11). Dôsledkom je zníženie α_c, spomalenie nájdenia atraktora, ale i odstránenie falošných atraktorov typu stavov spinového skla [20, 61, 67]. **Diskretizácia** spojení znamená, že synapsy môžu nadobúdať len diskrétne hodnoty (viď obr. 8.8a). Extrémnym prípadom diskretizácie spojení je ich **binarizácia**, t.j. synapsy môžu nadobúdať iba dve hodnoty:

$$J_{ij} = \begin{cases} \mathrm{sign}\left(\sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu} \right) & \text{pre} \quad i \neq j, \\ 0 & \text{pre} \quad i = j. \end{cases} \tag{8.39}$$

Teoretickou analýzou tohoto modelu bolo ukázané, že sieť i naďalej spoľahlivo funguje ako autoasociatívna pamäť, len kapacita α_c sa zníži z hodnoty 0,138 na hodnotu 0,1 [37]. **Limitácia** znamená obmedzenie pre synaptické váhy v tvare $|J_{ij}| \leq A$. To znamená, že hodnotu synaptickej váhy počítame podľa vzťahu (8.11), a keď vypočítaná hodnota prekročí hranicu A, nahradíme ju touto hodnotou (viď obr. 8.8b).



Obrázok 8.8. Príklady funkcií transformujúcich pôvodné hodnoty synaptických váh vypočítaných podľa Hebbovho pravidla (8.11) na nové v prípade (a) diskretizácie a (b) limitácie synaptických váh.

Limitácia synáps poskytuje zaujímavé možnosti, čo sa týka modelovania učenia a zabúdania. Predstavme si, že záznam nových pamäťových vzorov do Hopfieldovej siete inkrementujeme tak, že pridanie každého nového vzoru μ prebieha podľa vzťahu

$$J_{ij} \leftarrow J_{ij} + \eta \xi_i^\mu \xi_j^\mu \ . \tag{8.40}$$

Koeficient η označuje rýchlosť synaptickej modifikácie (rýchlosť učenia). Pritom môžu synaptické váhy nadobúdať iba hodnoty z intervalu $\langle -A, +A \rangle$. Takéto učenie sa nazýva **učenie v intervale** (angl. *learning within bounds*) [60, 61]. Posledne pridané pamäťové vzory sa budú dobre vyvolávať z pamäti, avšak staršie budú pomaly miznúť. Počet zapamätaných vzorov závisí na hodnotách η a A. Ak je hodnota η príliš veľká, iba posledne zapamätané vzory sú vyvolateľné. Ak je hodnota η príliš malá, sieť sa môže predimenzovať ($\alpha > \alpha_c$) a naplniť sa príliš veľa vzormi ešte predtým, ako váhy dosiahnu hranice intervalu A, a sieť prestane fungovať ako autoasociatívna pamäť. Pri optimálnych hodnotách η a A sa bude množina zapamätaných vzorov meniť tak, že pridávaním nových vzorov budeme vymazávať najstaršie zapamätané vzory. Takto sa dá modelovať **učenie a zabúdanie** [22, 60, 61].

Ďalší typ poškodenia je vymazanie alebo **zriedenie spojení** (angl. *dilution*). Náhodne vynulujeme konečný počet spojení medzi neurónmi tak, že

$$J_{ij} = \begin{cases} J_{ij}^{Hebb} & \text{s pravdepodobnosťou } c, \\ 0 & \text{s pravdepodobnosťou } (1-c). \end{cases} \tag{8.41}$$

J_{ij}^{Hebb} je hodnota synaptickej váhy vypočítaná podľa Hebbovho pravidla (8.11). Po zriedení bude percento zachovaných spojení rovné $100c$ a každý neurón bude mať približne cN spojení. Pri symetrickom riedení s hodnotou J_{ij}

vymazávame zároveň aj J_{ji}, pri asymetrickom riedení pristupujeme k J_{ij} a J_{ji} nezávisle. Nech c_{ij}=1, keď spojenie nie je vymazané a c_{ij}=0, keď sme spojenie zrušili. Potom pre každú váhu môžeme písať

$$J_{ij} = c_{ij} J_{ij}^{Hebb}. \tag{8.42}$$

Lokálne pole na *i*-tom neuróne je rovné

$$h_i = \sum_{j=1}^{N} c_{ij} J_{ij}^{Hebb} S_j \ . \tag{8.43}$$

Rozoznávame tzv. **slabé zriedenie**, keď $cN \geq \ln N$. **Silné zriedenie** nastáva, keď platí $cN << \ln N$. Pri slabom zriedení a veľkom N je ešte stále v sume (8.43) veľa členov, a preto môžeme aplikovať teóriu stredného poľa. Pre strednú hodnotu lokálneho potenciálu na *i*-tom neuróne dostaneme vzťah

$$\langle h_i \rangle = c \sum_{j=1}^{N} J_{ij}^{Hebb} \langle S_j \rangle \ . \tag{8.44}$$

S príslušným preškálovaním pomocou parametra c dostaneme analogické výsledky ako pre nezriedený model, v ktorom je každý neurón spojený s každým.

Silné zriedenie v deterministickej a stochastickej Hopfieldovej sieti ako prví študovali Derrida, Gardner a Zippelius [20]. Autoasociatívne vlastnosti sú zachované v jednom aj druhom prípade. Na príklade si to ukážeme pre deterministickú sieť. Nech priemerný počet spojení na jednom neuróne je $K=cN$ a $K << \ln N$. Zriedenie je asymetrické, takže c_{ij} a c_{ji} sú nezávislé náhodné premenné. Matica $\boldsymbol{J}$ sa teda stala asymetrickou. Pre výpočet váh sa používa vzťah

$$J_{ij} = \frac{1}{K} c_{ij} \sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu} \ , \tag{8.45}$$

s $1/K$ namiesto $1/N$, aby sme dostali hodnoty blízke 1. Pre ľubovoľný stav $\boldsymbol{S}(t)$ uvažujme lokálne pole príspevkov od všetkých neurónov spojených s *i*-tym neurónom rozdelené na člen pochádzajúci od konkrétneho pamäťového vzoru ν a presluchový člen:

$$h_i = \sum_{j=1}^{N} J_{ij} S_j = \frac{1}{K} \sum_{j=1}^{N} c_{ij} \sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu} S_j = \frac{1}{K} \xi_i^{\nu} \sum_{j=1}^{N} c_{ij} \xi_j^{\nu} S_j + C_i^{\nu} \ , \tag{8.46}$$

kde presluch

$$C_i^{\nu} = \frac{1}{K} \sum_{\mu \neq \nu} \xi_i^{\mu} \sum_{j=1}^{N} c_{ij} \xi_j^{\mu} S_j \ . \tag{8.47}$$

Ak v (8.46) položíme $S_i = \xi_i^{\nu}$ pre $\forall i$, vidíme, že pre dostatočne malý počet zapamätaných vzorov p nám prvý člen na pravej strane rovnice (8.46) dáva ξ_i^{ν}, a to vďaka tomu, že sme ako normalizačnú konštantu zvolili $1/K$, čo nám umožňuje vykrátiť $1/K$ a sumu s c_{ij}, lebo $\sum_j c_{ij} = K$. Keď položíme $S_i = \xi_i^{\nu}$ pre $\forall i$ v rovnici pre presluchový člen (8.47), na pravej

strane dostaneme $1/K$ krát suma Kp nezávislých náhodných ±1. V limite pre $Kp \to \infty$ má presluchový člen Gaussovo rozdelenie (viď obr. 8.5) a ďalej platí podobný záver ako pre úplne pospájanú sieť, teda že pamäťové vzory sú stabilné pre $p << N$.

Zachovanie autoasociatívnych vlastností Hopfieldovych sietí aj pri silnom rozriedení synaptickej matice umožňuje priblížiť sa k biologickej realite. Po prvé napríklad v tom, že v reálnych neurónových sieťach nie sú neuróny pospájané každý s každým monosynaptickými spojmi a po druhé v tom, že váhy spojení nie sú symetrické. Pomocou selektívneho riedenia možno skonštruovať Hopfieldovu neurónovú sieť, v ktorej je zachovaný tzv. Daleov princíp, ktorý hovorí, že jeden neurón môže vytvárať na druhých neurónoch len inhibičné alebo len excitačné synapsy, ale nie oba druhy naraz [66]. Zachovanie autoasociatívnych vlastností pri silnom rozriedení sa hodí aj pri konštrukcii viacvrstvových sietí s Hopfieldovou dynamikou, ktoré sa dajú použiť na hierarchické uchovávanie dát [34].

8.7 Neortogonálne vzory

Doteraz sme sa zaoberali Hopfieldovou neurónovou sieťou, ktorá je schopná zaznamenať a vyvolať ortogonálne resp. pseudoortogonálne pamäťové vzory, t.j. také pre ktoré platí, že stredná hodnota ich skalárneho súčinu cez všetky vzory sa rovná nule:

$$\langle\langle \xi^{\mu} \cdot \xi^{\nu} \rangle\rangle = \frac{1}{N} \sum_{j=1}^{N} \xi_j^{\mu} \xi_j^{\nu} = 0 \qquad \text{pre} \quad \mu \neq \nu . \tag{8.48}$$

Keď je v sieti veľký počet neurónov N, je možné vyhovieť tejto podmienke len ak vyberáme resp. konštruujeme pamäťové vzory tak, že v každom z nich sa počet aktívnych neurónov rovná počtu neaktívnych neurónov, t.j. že v sieti je 50%-ná priemerná aktivita, to znamená

$$\langle\langle \xi_j^{\mu} \rangle\rangle = \frac{1}{N} \sum_{j=1}^{N} \xi_j^{\mu} = 0 \quad \text{pre } \forall \mu \ . \tag{8.49}$$

O takýchto vzoroch hovoríme tiež, že sú nekorelované. V praxi však často pracujeme so vzormi, pre ktoré tieto podmienky nie sú splnené. Aj v neurónových populáciách, napríklad na kôre mozgu, je pomer počtu aktívnych neurónov k počtu neaktívnych neurónov podstatne nižší ako 50%. Pri modelovaní tejto skutočnosti sa stav každého neurónu ξ_i^{μ} v pamäťovom vzore vyberá nezávisle s pravdepodobnosťou $P(\xi) = \frac{1}{2}(1+b)\delta(\xi - 1) + \frac{1}{2}(1-b)\delta(\xi + 1)$. Symbol δ označuje Diracovu δ funkciu, kde $\delta(x) = 0, \text{ak } x \neq 0, \delta(x) = 1, \text{ak } x = 0.$ Parameter b sa nazýva **bias** a platí preň vzťah

$$b = \frac{1}{N} \sum_{i=1}^{N} \xi_i^{\mu} \quad pre \, \forall \mu \ . \tag{8.50}$$

Keďže b nadobúda hodnoty v intervale $-1 < b < +1$ môžeme pracovať s ľubovoľnou úrovňou aktivity. Stredná aktivita siete je rovná $\frac{1}{2}(1+b)$ a namiesto (8.48) platí

$\langle\langle \xi^{\mu} \cdot \xi^{\nu} \rangle\rangle = \delta^{\mu\nu} + b^2\left(1-\delta^{\mu\nu}\right)$ pre $\mu \neq \nu$. Pre takéto neortogonálne (korelované) vzory bola navrhnutá modifikácia dynamiky stochastickej Hopfieldovej neurónovej siete [7]. Predpis na konštrukciu synaptických váh (8.11) sa mení takto:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p}(\xi_i^{\mu} - b)(\xi_j^{\mu} - b) & \text{pre} \quad i \neq j\,, \\ 0 & \text{pre} \quad i = j\,. \end{cases} \tag{8.51}$$

Globálna kontrola dynamiky neurónovej siete sa dosiahne tak, že sa k energii (8.6) pridá člen, ktorý bude zabraňovať, aby sa sieť dostala do oblastí s aktivitou príliš vzdialenou pôvodnému biasu vzorov. Modifikovaná energia siete je rovná (prahy excitácie sú nulové)

$$E(\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} J_{ij}S_iS_j + \frac{r}{2N}\left(\sum_{i=1}^{N} S_i - Nb\right)^2 \quad . \tag{8.52}$$

Koeficient $r>0$ je mierou sily, ktorou je aktivita siete "tlačená" do oblasti určenej biasom vzorov. Lokálne pole pôsobiace na jeden neurón (8.1) sa modifikuje takto:

$$h_i = \sum_{j=1}^{N} J_{ij}S_j \quad - r\left(\frac{1}{N}\sum_{j=1}^{N} S_j - b\right) . \tag{8.53}$$

Vzťahy (8.50) až (8.53) zodpovedajú tzv. jemnému obmedzeniu dynamiky (angl. *softly constrained dynamics*). Táto modifikácia zachováva všetky hlavné vlastnosti pôvodného stochastického Hopfieldovho modelu. Okrem toho platí:

(1) Ak je koeficient r □ 10, dostávame spoľahlivé fungovanie siete pre akúkoľvek hodnotu biasu b.
(2) Pamäťová kapacita siete s modifikovanou dynamikou sa pohybuje v intervale: $0{,}12\,N < p_{\max} < 0{,}18N$ pre r □ 10. Konkrétna hodnota kapacity závisí od hodnoty biasu b.
(3) Kritická teplota závisí od biasu b. Pri teplotách nižších ako T_c sú všetky pamäťové vzory stabilné.

$$T_c=(1\tilde{}\ b^2)^2 . \tag{8.54}$$

Dokonca aj pri veľmi nízkych hodnotách T sú falošné atraktory (zmiešané stavy) eliminované.

(4) Symetria $S_i \leftrightarrow -S_i$ je zrušená.

8.8 Časové postupnosti vzorov

Podnety, ktoré spracováva náš mozog, prichádzajú v časových postupnostiach. Príkladom je ľudská reč, ale i vizuálne prehľadávanie okolia, sled myšlienok, predstáv, atď. Na druhej

strane, mozog sám generuje časové postupnosti, napríklad pri generovaní povelov pre svaly, čo sa vlastne deje neustále. Tzv. centrálne generátory rytmu sú neurónové obvody, ktoré generujú biologické rytmy, od riadenia cyklu spánku a bdenia až po rytmické pohyby napríklad pri dýchaní. Štúdium umelých neurónových sietí sa snaží svojimi výsledkami tiež prispieť k pochopeniu podstaty týchto procesov. Pozrime sa teraz ako sa dajú rozpoznávať a generovať časové postupnosti konfigurácií aktivít v Hopfieldovej neurónovej sieti. Vieme, že v Hopfieldovej sieti reprezentuje konfigurácia aktivity celej populácie modelových neurónov nejaký ľubovoľný podnet, resp. pamäťový stav. Pamäťové stavy siete sú atraktormi v stavovom priestore a priťahujú im podobné stavy. Formulujme si úlohu spracovania časových postupností takto:

(1) Externý podnet spôsobí vyvolanie pamäťového vzoru, ktorý je s ním asociovaný. Nech je tento pamäťový stav členom nejakej postupnosti stavov.

(2) Neurónová sieť zotrváva v tomto atraktore, ale po uplynutí nejakej doby sa presunie do iného atraktoru, ktorý je asociovaný s nasledujúcim členom v časovej postupnosti vzorov.

(3) Sieť sa postupne presúva z jedného atraktora do ďalšieho sledujúc tak preddefinovanú postupnosť stavov. V každom atraktore zotrvá istú dobu. Správnejšie by sa teda malo hovoriť o pseudo-atraktoroch alebo kvázi-atraktoroch, no pre jednoduchosť budeme i naďalej používať termín atraktor. Presun z atraktora do atraktora môže byť buď spontánny — vtedy sa jedná o **generovanie časovej postupnosti** stavov vyvolanej nejakým podnetom, alebo sa presúva z jedného atraktora do druhého na základe príchodu ďalšieho podnetu — vtedy sa jedná o **rozpoznanie časovej postupnosti**.

Podľa princípu, ktorý navrhli viacerí autori, menovite Kanter a Sompolinsky [47], Kleinfeld a Kanter [50], Gutfreund a Mézard [35], a Amit [8], dosiahneme takéto správanie sa autoasociatívnej neurónovej siete zavedením masívnej a koherentnej asymetrie do matice synaptických spojení medzi neurónmi. Majme sieť pozostávajúcu z N neurónov, ktoré môžu nadobúdať stav +1 alebo 1. Sieť uchováva p pamäťových vzorov, ktorých konfigurácia aktivity je vygenerovaná náhodne, čiže v sieti je 50%-ná priemerná aktivita. Dynamika je asynchrónna, a aktualizovanie neurónov môže byť buď deterministické alebo stochastické. V sieti sú **dve množiny synáps** J_{ij} a J_{ij}^{Δ}. J_{ij} sa vypočíta podľa vzťahu (8.11). V tejto synaptickej matici je zapamätaných p dopredu zvolených vzorov a jej úlohou je oprava chýb a stabilizácia individuálnych vzorov. Druhá množina synáps J_{ij}^{Δ} má na starosti prechod medzi atraktormi v preddefinovanej postupnosti. Vypočíta sa podľa vzťahu

$$J_{ij}^{\Delta} = \frac{\lambda}{N} \sum_{\mu=1}^{q} \xi_i^{\mu+1} \xi_j^{\mu} \quad , \tag{8.55}$$

kde $q \leq p$ je počet vzorov spojených v časovej postupnosti $\mu = 1 \to 2 \to 3 \to \ldots \to (q+1)$. J_{ij}^{Δ} definuje poradie q vzorov v postupnosti. Uzavretý cyklus sa dá zostrojiť pomocou (8.55) tak, že položíme $\xi_i^{q+1} = \xi_i^1$. Parameter λ je kladná konštanta. Nech sú prahy excitácie rovné nule. Celkové pole pôsobiace na i-ty neurón má tieto dve zložky

$$h_i(t) = \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu} S_j(t) + \frac{\lambda}{N} \sum_{j=1}^{N} \sum_{\mu=1}^{q} \xi_i^{\mu+1} \xi_j^{\mu} S_j(t-\tau) \quad . \tag{8.56}$$

Prepíšme si (8.56) pomocou prekryvov

$$m^{\mu}(t)=\frac{1}{N}\sum_{j=1}^{N}\xi_j^{\mu}S_j(t) \qquad \text{a} \qquad m^{\mu}(t-\tau)=\frac{1}{N}\sum_{j=1}^{N}\xi_j^{\mu}S_j(t-\tau), \tag{8.57}$$

tak, že dostaneme

$$h_i(t)=\sum_{\mu=1}^{p}\xi_i^{\mu}\,m^{\mu}(t)+\lambda\sum_{\mu=1}^{q}\xi_i^{\mu+1}m^{\mu}(t-\tau)\ . \tag{8.58}$$

Prvý člen závisí na okamžitých aktivitách neurónov v čase *t*, a teda na momentálnom prekryve stavu siete s jednotlivými pamäťovými vzormi. Druhý, **prechodový člen** závisí na tom, aká bola konfigurácia aktivity neurónov pred τ relaxačnými cyklami. J_{ij}^{Δ} sú tzv. **pomalé** synapsy, alebo **prechodové synapsy**, lebo sprostredkujú neurónom informáciu o stave ich susedov **s časovým oneskorením** $\tilde{\tau}$ Predstavme si, že sa sieť ocitne v stave μ, napríklad že doňho zrelaxuje po príchode nejakého podnetu. Keďže je to atraktor, sieť v ňom zotrvá. Avšak po uplynutí τ cyklov sa hodnota prekryvu $m^{\mu}(\tilde{t}\tau)$ bude rovnať 1 (alebo sa bude blížiť hodnote 1, podľa toho, či je dynamika deterministická alebo stochastická). Pozrime sa na vzťah pre lokálne pole *i*-teho neurónu (8.58) a vidíme, že ak je □ □ 1, sieť prejde do ďalšieho atraktora s poradovým číslom μ+1. V tomto atraktore zotrvá opäť po dobu τ, a potom prejde do atraktora μ+2, atď. V tomto prípade sa jedná o **spontánne generovanie časovej postupnosti** stavov siete, ktoré bolo iniciované nejakým vonkajším podnetom. **Rozpoznávanie časových postupností** bez spontánneho pokračovania v postupnosti dosiahneme tak, keď v (8.58) položíme □ □ 1 a ku každému lokálnemu poľu (8.58) pripočítame v časovom momente *t* príspevok z vonkajšieho signálu $h_i^{out}=\rho\xi_i^{\mu+1}$, kde ρ □ 1. Prechod do ďalšieho atraktora v časovej postupnosti nastane teda iba vtedy, keď sa tento stav (alebo jemu podobný stav) ocitne na vstupe do siete.

Vo vzťahu (8.56) môžeme namiesto $S_j(t-\tau)$ použiť aj iné závislosti. Položme $S_j(t-\tau)=\overline{S}_j(t)$. Tzv. **pamäťová stopa** $\overline{S}_j(t)$ môže mať vo všeobecnosti tvar $\overline{S}_j(t)=\int_{-\infty}^{t}G(t-t')S_j(t')dt'$, kde sme doteraz uvažovali funkciu $G(t)$ v tvare Diracovej delta funkcie $G(t)=\delta(t-\tau)$, ale môže to byť aj lineárna závislosť $G(t)=(\tau-t)/\tau$ alebo exponenciálna závislosť $G(t)=\tau^{-1}\exp(-t/\tau)$.

Vzťah (8.58) môže slúžiť na **rozpoznávanie viac ako jednej postupnosti** pomocou jednej a tej istej neurónovej siete len vtedy, ak každá postupnosť obsahuje iné členy μ. Gutfreund a Mézard [35] navrhli postup, ako môže Hopfieldova sieť rozpoznávať viacero rôznych časových postupností aj vtedy, keď každý stav μ má niekoľko možných pokračovaní. Tento postup si najprv ukážeme pre prípad, keď chceme, aby sieť uchovávala dve rôzne postupnosti, $\{\xi_i^{1,\mu}\}$ a $\{\xi_i^{2,\mu}\}$. Druhý, prechodový člen v (8.58) teraz nahradíme týmto výrazom

$$h_i^{\Delta}(t) = \lambda \sum_{\mu=1}^{q} (\xi_i^{1,\mu+1} + \xi_i^{2,\mu+1}) \left(m^{1,\mu}(t-\tau) + m^{2,\mu}(t-\tau) \right) . \qquad (8.59)$$

Ak je systém v stave (1, μ), môže pokračovať buď do stavu (1, μ+1) alebo (2,μ+1). Skutočný prechod bude determinovaný vonkajším signálom $h_i^{out} = \rho \xi_i^{s,\mu+1}$, kde ρ □1. Ak bude tento signál z postupnosti *s*=1, tak sa bude pokračovať v postupnosti 1, ak bude z postupnosti *s*=2, tak sa bude pokračovať v postupnosti 2. Zovšeobecnenie (8.59) pre viac ako dve postupnosti je triviálne. Pri vhodnom výbere hodnôt parametrov môže sieť rozlíšiť až 2^p rôznych časových postupností.

8.9 Invariantné rozpoznávanie vzorov

Invariantné rozpoznávanie obrazcov (vzorov) robí mozog neustále. Vo vizuálnej sfére je to napríklad rozpoznávanie objektov bez ohľadu na ich veľkosť, posunutie a otočenie. V sluchovej sfére je to napríklad rozpoznávanie melódií bez ohľadu na rozdielnu výšku tónovej oktávy, v ktorej sú zahraté, alebo invariantné rozpoznávanie hlások a slov v reči rôznych ľudí. Neurobiologické mechanizmy tohto procesu (či procesov) sú stále predmetom intenzívneho experimentálneho a teoretického výskumu.

Zaujímajú nás možnosti ako prispieť k hľadaniu riešenia pomocou autoasociatívnych neurónových sietí. Vnútorná reprezentácia nejakého vzoru je v autoasociatívnej sieti zakódovaná v aktivite celej siete. Táto aktivita zodpovedá len istej konkrétnej vstupnej stimulácii, ktorú opakovane vyvoláva daný vzor, najskôr v procese učenia a potom v procese rozpoznávania. Táto predstava zodpovedá neurobiologickej realite. Avšak ani mohutné autoasociatívne vlastnosti Hopfieldovych sietí, takých ako sme ich doteraz predstavili, nestačia na simulovanie invariantného rozpoznávania vzorov. To, ktorý atraktor si sieť vo svojej evolúcii vyberie, silno závisí na tom, ako ďaleko je v stavovom priestore počiatočná konfigurácia od jednotlivých atraktorov. Pri posunutí, zmenšení, prevrátení, atď. vzoru treba počítať s tým, že iniciálny stav siete je v priestore stavov veľmi vzdialený od príslušnej pamäťovej konfigurácie. Inými slovami, posunutému, či inak zmenenému vzoru, zodpovedá úplne iný vzorec aktivity siete, ktorý je len málo, resp. vôbec nie podobný na aktivitu siete v čase zapamätávania tohto vzoru. Keby sme sieť nechali spontánne sa vyvíjať, skončila by v atraktore, ktorý sa najviac podobá vstupnej stimulácii.

Dotsenkov algoritmus pre invariantné rozpoznávanie vzorov v autoasociatívnej sieti vzhľadom k ich posunutiu, otočeniu a zmenu rozmerov [24] je založený na využití neurónových prahov excitácie pri riadení evolúcie siete v stavovom priestore. Tento prístup si najskôr vysvetlíme pre prípad invariantného rozpoznania vzoru vzhľadom k jeho posunutiu. Neurónová sieť, pre ktorú je algoritmus navrhnutý, je pôvodným modelom Hopfieldovej stochastickej siete so symetrickou váhovou maticou, s neurónmi pospájanými každý s každým, a s asynchrónnou dynamikou. V ďalšom budeme pracovať s geometrickou predstavou neurónovej siete ako 2-rozmernej mriežky, a preto namiesto indexu *i* budeme používať vektor polohy ***r***, resp. pre *j*-ty neurón vektor ***r'***. Vzdialenosť okamžitej konfigurácie siete ***S***(*t*) od jednotlivých pamäťových vzorov ξ^{μ} vyjadruje okamžitý prekryv $m^{\mu}(t)$

$$m^{\mu}(t) = \frac{1}{N}\sum_{\boldsymbol{r}} \xi^{\mu}(\boldsymbol{r})\, S(\boldsymbol{r},t) \quad , \qquad pre \quad \mu = 1,\ldots,p\,. \tag{8.60}$$

Táto rovnica je tá istá rovnica ako (8.17), len prepísaná pomocou vektorov polohy $\boldsymbol{r}$ namiesto indexu i. Relaxačná dynamika neurónov je asynchrónna s pravdepodobnosťou zmeny stavu pre neurón so súradnicou $\boldsymbol{r}$ vyjadrenou vzťahom (viď rovnice 8.28 a 8.29):

$$S(\boldsymbol{r},t) = \begin{cases} +1 \qquad \text{s pravdepodobnosťou} \quad P_{+} = \dfrac{1}{1+\exp(-2\beta\, h(\boldsymbol{r},t))} \\ \\ -1 \qquad \text{s pravdepodobnosťou} \quad P_{-} = 1 - P_{+} \end{cases} \tag{8.61}$$

Celkové pole $h(\boldsymbol{r},t)$ pôsobiace na neurón na mieste $\boldsymbol{r}$ v čase t je rovné súčtu príspevkov od ostatných neurónov s indexami $\boldsymbol{r}'$ váhovanými cez synaptické váhy plus príspevok od lokálneho nenulového prahu excitácie $\theta(\boldsymbol{r},t) \equiv h^{ext}(\boldsymbol{r},t) \neq 0$, t.j.

$$h(\boldsymbol{r},t) = \sum_{\boldsymbol{r}\boldsymbol{r}'} J_{\boldsymbol{r}\boldsymbol{r}'} S(\boldsymbol{r}',t) + \theta(\boldsymbol{r},t)\ . \tag{8.62}$$

Prahy excitácie sú nielenže nenulové, ale majú aj časovú závislosť, ktorú špecifikujeme neskôr. Pri splnení určitých podmienok môžu prostredníctvom (8.61) veľmi účinne riadiť evolúciu siete v stavovom priestore, lebo pravdepodobnosť zmeny stavu neurónov bude závisieť hlavne od ich relatívnej hodnoty vzhľadom k príspevku od ostatných neurónov.

Váhová matica $\boldsymbol{J}$ je symetrická a neuróny na sebe netvoria synapsy. Hodnotu synaptickej váhy medzi dvoma neurónmi vypočítame podľa vzťahu (8.11), ktorý pre naše účely prepíšeme pre vektorové súradnice neurónov $\boldsymbol{r}$ a $\boldsymbol{r}'$:

$$J_{\boldsymbol{r}\boldsymbol{r}'} = \begin{cases} \dfrac{J_0}{N}\displaystyle\sum_{\mu=1}^{p} \xi^{\mu}(\boldsymbol{r})\,\xi^{\mu}(\boldsymbol{r}') & \text{pre} \quad r \neq r' \\ \\ 0 & \text{pre} \quad r = r' \end{cases} \tag{8.63}$$

Tento vzťah je všeobecnejší ako (8.11), pretože sme doňho zaviedli konštantu J_0, ktorá vo všeobecnosti nemusí byť rovná 1.

Zatiaľ budeme pracovať s ortogonálnymi (resp. pseudoortogonálnymi) vzormi, v ktorých je 50%-ná priemerná aktivita. Nech $\mathbf{S}^{(0)}$ je počiatočná konfigurácia siete, a nech je to jeden z pamäťových vzorov ξ^{μ} posunutý o vektor $(-\boldsymbol{a}^*)$. Keďže v sieti musí byť zachovaná 50%-ná priemerná aktivita, uvažujeme periodické okrajové podmienky. To znamená, že pri posunutí vzoru cez okraj mriežky sa na protiľahlom okraji objaví tá časť vzoru, ktorá sa pri posune dostala mimo mriežky (viď obr. 8.9). Pre pamäťovú konfiguráciu ξ^{μ} platí vzťah:

$$\xi^{\mu}(\boldsymbol{r}) = S^{(0)}(\boldsymbol{r} + \boldsymbol{a}^*), \quad \text{pre} \quad \forall \boldsymbol{r}. \tag{8.64}$$

Samozrejme, sieť "nevie" aká je hodnota $\boldsymbol{a}^*$.

V Dotsenkovom algoritme sa predpokladá, že počiatočný vzor $\mathbf{S}^{(0)}$ sa v čase $t = 0$ premietne na neurónové prahy excitácie tak, že tieto nadobudnú hodnoty

$$\theta(\boldsymbol{r}, t=0) = \theta_0 S^{(0)}(\boldsymbol{r}), \quad \text{pre } \forall \boldsymbol{r} \ . \tag{8.65}$$

θ_0 je kladná konštanta. V každom ďalšom časovom momente sa hodnoty prahov excitácie menia podľa tejto rovnice:

$$\theta(\boldsymbol{r}, t) = \theta_0 S^{(0)}(\boldsymbol{r} + \boldsymbol{a}(t)), \quad \text{pre } \forall \boldsymbol{r} \, . \tag{8.66}$$

Dynamická premenná $\boldsymbol{a}(t)$ vyjadruje fakt, že konfigurácia prahov excitácie je v každom časovom okamihu iná, že je daná momentálnou hodnotou vektora $\boldsymbol{a}(t)$. Je to teda konfigurácia prahov excitácie, ktorá hľadá príslušný pamäťový atraktor, a to prostredníctvom dynamiky premennej $\boldsymbol{a}(t)$. Prahy budú dominovať pri zmene stavu neurónov (8.61) a príspevok od ostatných neurónov bude slúžiť hlavne na korekciu chýb vo vzore. Dotsenko navrhol relaxačnú dynamiku pre dynamickú premennú posunu prahov $\boldsymbol{a}(t)$ nasledovne:

$$\frac{d\boldsymbol{a}}{dt} \equiv \boldsymbol{a}(t+1) - \boldsymbol{a}(t) = -\frac{\delta E}{\delta \mathrm{a}} + \boldsymbol{\eta}(t) \, . \tag{8.67}$$

V tomto vzorci je E energia siete (8.6) v tvare

$$E(\boldsymbol{S}) = -\frac{1}{2} \sum_{\boldsymbol{r}} \sum_{\boldsymbol{r}'} J_{\boldsymbol{r}\boldsymbol{r}'} S(\boldsymbol{r}) S(\boldsymbol{r}') - \sum_{\boldsymbol{r}} S(\boldsymbol{r}) \theta(\boldsymbol{r}) \ . \tag{8.68}$$

Pomocou relaxačnej prahovej dynamiky, ktorej podstata je vyjadrená rovnicou (8.67), hľadáme lokálne minimum energie systému (8.68). Rovnica (8.67) odpovedá optimalizácii metódou záporného gradientu. Pri tejto optimalizácii sa minimum danej optimalizačnej funkcie hľadá proti smeru gradientu tejto funkcie. Štartuje sa z posunutého počiatočného bodu (8.65), posunieme sa s prahmi niektorým náhodným smerom, necháme sieť zrelaxovať podľa (8.61) a vypočítame energiu (8.68). Ak je energia nového stavu menšia ako pôvodná, pokračujeme v posúvaní sa týmto smerom. Ak je energia väčšia, vygenerujeme $\boldsymbol{a}(t)$ v smere opačnom, tak ako predpisuje (8.67). Veľkosť posunu závisí od veľkosti $\partial E / \partial \boldsymbol{a}$.Takto by sa iteratívnym spôsobom mala sieť usadiť so svojou aktivitou v konfigurácii zodpovedajúcej lokálnemu energetickému minimu príslušného pamäťového vzoru. Gradientová metóda optimalizácie, ktorá teraz riadi dynamiku systému, má tú nevýhodu, že keď sa systém dostane do falošného minima, nemôže sa dostať von. Preto je nutné

zaviesť do systému náhodnosť, napríklad v podobe obyčajného teplotného šumu $\eta(t)$. V práci [13] sme dospeli k záveru, že namiesto gradientovej minimalizácie je v tomto prípade efektívnejšie použitie optimalizácie pomocou simulovaného žíhania [16, 49]. Výsledky simulácie Dotsenkovho algoritmu pre invariantné rozpoznávanie vzorov vzhľadom k posunutiu sú ilustrované na obr. 8.9.

Aké sú podmienky toho, aby prahová dynamika dominovala nad relaxáciou neurónov? Rovnicu pre celkové pole pôsobiace na jeden neurón môžeme pomocou vzťahu pre synaptické váhy (8.63) a vzťahu pre prahy excitácie (8.66) prepísať na tvar

$$h(\boldsymbol{r},t)=\frac{J_0}{N}\sum_{\mu=1}^{p}\sum_{\boldsymbol{r}'}\xi^{\mu}(\boldsymbol{r})\xi^{\mu}(\boldsymbol{r}')S(\boldsymbol{r}',t)+\theta_0 S^{(0)}(\boldsymbol{r}+\boldsymbol{a}(t)). \tag{8.69}$$

Pomocou prekryvu

$$m^{\mu}(t)=\frac{1}{N}\sum_{\boldsymbol{r}'}\xi^{\mu}(\boldsymbol{r}')S(\boldsymbol{r}',t), \tag{8.70}$$

možno celkové pole vyjadriť takto:

$$h(\boldsymbol{r},t)=J_0\sum_{\mu=1}^{p}\xi^{\mu}(\boldsymbol{r})m^{\mu}(t)+\theta_0 S^{(0)}(\boldsymbol{r}+\boldsymbol{a}(t)). \tag{8.71}$$

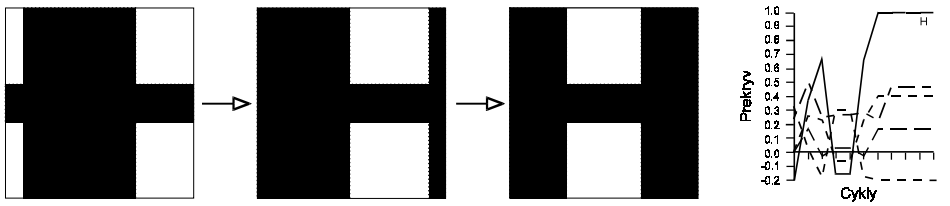
Vidíme, že druhý, prahový člen v rovnici (8.71) bude dominovať, ak $\theta_0 >> J_0$, a prvý člen môžeme zanedbať. Podľa rovnice (8.60) pre strednú hodnotu stavu neurónu $\boldsymbol{a}$ (8.71) zistíme, že stavy neurónov budú kopírovať prahové pole

$$\langle S(\boldsymbol{r},t)\rangle=\tanh\left[\beta\, h(\boldsymbol{r},t)\right]\cong\beta\theta_0 S^{(0)}\left(\boldsymbol{r}+\boldsymbol{a}(t)\right). \tag{8.72}$$

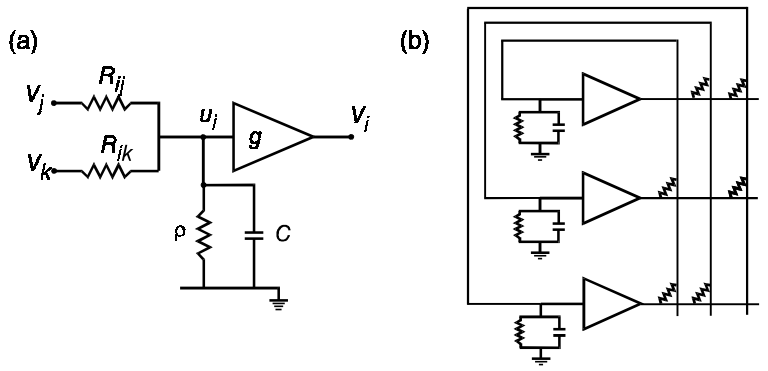
V praxi stačí, keď $\theta_0/J_0=2$.

Dotsenkov algoritmus, ktorý sme práve popísali pre invariantné rozpoznávanie vzorov vzhľadom k posunutiu, sa môže rovnakým spôsobom použiť aj na invariantné rozpoznávanie vzorov vzhľadom na ich veľkosť alebo otočenie. Rovnica (8.64) vyjadrujúca vzťah medzi pamäťovou konfiguráciou a počiatočným stavom siete bude pre zmenu veľkosti vzoru vyzerať takto:

$$\xi^{\mu}(\boldsymbol{r})=S^{(0)}(\lambda^{*}\boldsymbol{r}),\quad \text{pre}\quad \forall\boldsymbol{r}. \tag{8.73}$$



Obrázok 8.9. Dotsenkov algoritmus. Ilustrácia invariantného rozpoznávania vzorov vzhľadom k posunutiu s periodickými okrajovými podmienkami. Hopfieldova sieť má 100 neurónov, 5 pamäťových vzorov (písmen), a asynchrónnu stochastickú dynamiku. Excitačné prahy sú nenulové, a ich dynamika je riadená podľa (8.66). Graf ilustruje vývoj prekryvov s jednotlivými pamäťovými vzormi. Písmenom H je označená krivka odpovedajúca vývoju prekryvu aktuálneho stavu siete s písmenom H. Ostatné krivky zobrazujú vývoj prekryvu s ostatnými pamäťovými vzormi. (Všimnime si, že pamäťové vzory nie sú ortogonálne, t.j. ich prekryvy sú ďaleko od nulovej hodnoty.) Cykly odpovedajú jednotlivým iteráciam simulovaného žíhania.



Obrázok 8.10. Analógová implementácia Hopfieldovho modelu neurónovej siete. (a) Schéma analógového modelu neurónu. (b) Schéma zapojenia analógových prvkov do siete.

Skalár λ^* hovorí, koľkokrát je počiatočná konfigurácia zväčšená (zmenšená) oproti pamäťovej konfigurácii. Pre otočenie zavedieme operátor otočenia $\boldsymbol{\Omega}$, takže rovnica analogická rovniciam (8.64) a (8.73) bude

$$\xi^{\mu}(\boldsymbol{r}) = \boldsymbol{S}^{(0)}(\boldsymbol{\Omega}^{*}\boldsymbol{r}), \quad \text{pre} \quad \forall \boldsymbol{r}. \tag{8.74}$$

Namiesto dynamickej premennej $\boldsymbol{a}(t)$ bude v prípade transformácie veľkosti vzoru riadiť dynamiku dynamická premenná $\lambda(t)$, a v prípade otáčania dynamická premenná $\Omega(t)$.

Na záver spomenieme, že existujú aj iné prístupy k riešeniu invariantného rozpoznávania vzorov pomocou Hopfieldovych autoasociatívnych sietí, napríklad prístupy, ktoré navrhli Bienenstock a von der Malsburg [15] alebo Kree a Zippelius [56].

8.10 Analógový Hopfieldov model

Skutočné neuróny v mozgu nie sú dvojstavové prvky, ale majú spojitú, nelineárnu vstupno-výstupnú charakteristiku. Táto skutočnosť ako aj požiadavky hardwarovej implementácie Hopfieldovej neurónovej siete motivovali Hopfielda k tomu, aby svoj model rozšíril aj na siete so spojitými analógovými prvkami [41]. Skutočné neuróny aj elektronické zariadenia (ako napríklad operačný zosilňovač) majú integračné časové oneskorenia kvôli ich membránovej resp. elektrickej kapacitancii, a evolúciu ich stavu (t.j. elektrického napätia) v čase popisujú diferenciálne rovnice príslušného elektrického obvodu.
Na obr. 8.10a je nakreslený elektrický obvod, ktorý reprezentuje jeden prvok (**analógový neurón**) v analógovej implementácii Hopfieldovho modelu. Obr. 8.10b zobrazuje sieť tvorenú takýmito prvkami.

Premenná u_i označuje vstupné napätie i-teho prvku, V_i je výstupné napätie. Ako u_i tak aj V_i sú funkciou času, vyvíjajú sa v čase. Operačný zosilňovač má vstupno-výstupnú funkciu g takú, že

$$V_i = g(u_i) . \tag{8.75}$$

Najčastejšie sa používa **nelineárna funkcia** $g(u)=\tanh(\beta u)$, ktorá má hodnoty v intervale $\tilde{}1,+1$ alebo sigmoidálna funkcia $g(u)=f_\beta(u)$, s hodnotami v intervale $0,+1$ (viď obr. 8.6). Vstup každého prvku je uzemnený cez odpor ρ a kondenzátor s kapacitou C. Tieto premenné reprezentujú transmembránový odpor a kapacitu skutočného neurónu (viď obr. 1.9). Výstup j-teho prvku je spojený so vstupom i-teho prvku cez odpor R_{ij}. Z Kirchhofovho zákona o rovnosti prúdov do uzla vtekajúcich a z uzla vytekajúcich vyplýva, že

$$C\frac{du_i}{dt} + \frac{u_i}{\rho} = \sum_{j\neq i}^{N} \frac{1}{R_{ij}}(V_j - u_i) . \tag{8.76}$$

alebo tiež

$$\tau_i \frac{du_i}{dt} = -u_i + \sum_{j\neq i}^{N} w_{ij} g(u_j) , \tag{8.77}$$

kde

$$\tau_i = R_i C; \qquad \frac{1}{R_i} = \frac{1}{\rho} + \sum_{j\neq i}^{N} \frac{1}{R_{ij}}; \qquad w_{ij} = \frac{R_i}{R_{ij}} . \tag{8.78}$$

Podiel $w_{ij}=R_i / R_{ij}$ reprezentuje hodnotu synaptickej váhy synapsy tvorenej j-tym prvkom na i-tom prvku. Ak zvolíme hodnotu ρ dostatočne malú a R_{ij} veľké, môžeme členy $1/R_{ij}$ zanedbať, a potom $R_i \approx \rho$ a $w_{ij} \approx \rho / R_{ij}$ pre $\forall i$. Ak má mať niektorá váha negatívnu hodnotu, môžeme to v hardwarovej implementácii vyriešiť napríklad tak, že pred ňu zapojíme invertor, ktorý zmení hodnotu vstupného napätia na $-V_j$.

Keď systém skonverguje do atraktora, pre $\forall i$ platí $du_i/dt=0$ a $dV_i/dt=0$. Vtedy $u_i = \sum_j w_{ij} V_j$. Rovnaké riešenie ako systém dynamických rovníc (8.77) má aj systém diferenciálnych rovníc pre V_i:

$$\tau_i \frac{dV_i}{dt} = -V_i + g(u_i) = -V_i + g\left(\sum_{j \neq i}^{N} w_{ij} V_j \right) . \qquad (8.79)$$

Dynamika tohto systému je spojitá, t.j. **všetky prvky aktualizujú svoj stav spojite a súčasne** podľa rovníc (8.75) a (8.77) resp. (8.79). Sledovať evolúciu vyššie popísanej analógovej neurónovej siete nám umožní **numerická integrácia systému rovníc** (8.77) alebo (8.79) na počítači. V literatúre sa odporúča napríklad integrovanie s adaptívnou veľkosťou kroku ako je napríklad Bulirschova-Stoerova metóda [65]. Ak je strmosť funkcie $g(u)$ vysoká, výstupy prvkov sa blížia k hraniciam ±1, resp. 0 a 1. Takto môžeme získať binárne odpovede aj od prvkov spojitej analógovej siete. Pri určitých úlohách je však potrebné pracovať so spojitými hodnotami (viď aplikácie Hopfieldovho modelu).

Analógia so stochastickou sieťou: všimnime si, že rovnica (8.75) s $g(u) = \tanh(\beta u)$ je vlastne taká istá ako rovnice stredného poľa (8.34) pre stochastickú Hopfieldovu sieť. Vstupné napätie analógového prvku u_i hrá úlohu priemerného interného poľa $\langle h_i \rangle$ a výstupné napätie V_i je ekvivalentné priemernej hodnote stavu i-teho neurónu $\langle S_i \rangle$. To znamená, že analógovú sieť môžeme použiť na riešenie rovníc stredného poľa pre stochastickú Hopfieldovu sieť pri nenulovej "teplote" T.

Takisto ako pre binárnu Hopfieldovu sieť, aj pre analógovú sieť môžeme definovať energetickú funkciu tak, aby sa energia systému v priebehu evolúcie minimalizovala. Rovnovážne riešenia rovníc (8.77) a (8.79) zodpovedajú minimám tejto energetickej funkcie a im prislúchajúce konfigurácie aktivity odpovedajú atraktorom v stavovom priestore. **Energia** analógovej Hopfieldovej siete je [41]

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} V_i V_j + \sum_{i=1}^{N} \int_0^{V_i} g^{-1}(V) dV \ , \qquad (8.80)$$

kde g^{-1} je inverzná funkcia k funkcii g. Aby sme ukázali, že ***E* sa v priebehu evolúcie systému minimalizuje**, derivujme (8.80) podľa času t:

$$\frac{dE}{dt} = -\frac{1}{2} \sum_{ij} w_{ij} \frac{dV_i}{dt} V_j - \frac{1}{2} \sum_{ij} w_{ij} V_i \frac{dV_j}{dt} + \sum_i g^{-1}(V_i) \frac{dV_i}{dt}$$

$$= -\sum_i \frac{dV_i}{dt} \left(\sum_j w_{ij} V_j - u_i \right) = -\sum_i \tau_i \frac{dV_i}{dt} \frac{du_i}{dt} = -\sum \tau_i g'(u_i) \left(\frac{du_i}{dt} \right)^2 \leq 0 . \quad (8.81)$$

Pri týchto úpravách sme využili predpoklad, že synaptické váhy sú symetrické ($w_{ij} = w_{ji}$) a že $w_{ii} = 0$ a tiež vzťahy (8.75) a (8.77). Časová derivácia energie je záporná preto, lebo $\tau_i > 0$, $g(u)$ je monotónne rastúca funkcia, a teda jej prvá derivácia podľa času g' je nezáporná, a

derivácia u_i tu vystupuje v druhej mocnine. Výraz (8.81) sa rovná nule iba v rovnovážnom bode, keď $du_i/dt=0$ a $dV_i/dt=0$. Vzťah (8.81) ukazuje, že dynamická rovnica (8.77) spojito minimalizuje energiu systému, pokiaľ systém nedospeje do stavu zodpovedajúcemu lokálnemu energetickému minimu, kde $E=0$. **Rovnovážne stavy** systému určené synaptickou maticou $\boldsymbol{W}$ sú jeho atraktormi, a sú to **jediné atraktory** systému. Napríklad limitné cykly nemôžu byť atraktormi, lebo energia nemôže na uzavretej krivke spojito klesať. Symetria matice $\boldsymbol{W}$ garantuje existenciu bodových atraktorov**.** Ak by sme do synaptickej matice zaviedli asymetriu, môžu vzniknúť oscilácie alebo chaotický vývoj $V_i(t)$. V extrémnom prípade čisto náhodných váh w_{ij} s hodnotami, ktoré majú priemernú hodnotu 0 a strednú kvadratickú odchýlku σ^2, môžeme zaznamenať prechod od stabilného ku chaotickému vývoju, keď zvyšujeme hodnotu σ^2 [68].

Ďalšou možnosťou ako sledovať evolúciu tohto systému je jeho **hardwarová implementácia** pomocou elektronických prvkov. Najťažší problém pri vytváraní obvodov tejto neurónovej siete vo VLSI je výroba rezistorov R_{ij}. Pre úplne pospájanú sieť s N prvkami potrebujeme $2N^2$ rezistorov (keď používame invertory), a ich odpor musí byť dostatočne veľký, aby sa obmedzili energetické straty. Graf so spolupracovníkmi [33] vyrobil čip s N = 256 analógovými neurónmi, ktoré boli pospájané každý s každým cez približne $2N^2 >>$ 130 000 rezistorov, ktoré mali približne rovnaký odpor. Každý rezistor sa pridával individuálne na hotový CMOS čip pomocou elektrónovo-lúčovej litografie. Pri tejto metóde je hodnota odporu rezistorov raz navždy daná, ale nestráca sa po vypnutí zdroja napätia. Iní použili namiesto pasívnych odporov aktívne elektronické elementy (tranzistory) [2]. Takto získali flexibilnejšie, programovateľné čipy v zmysle modifikácie hodnôt ich synaptických spojení. Existujú aj optické a optoelektronické implementácie Hopfieldovej analógovej neurónovej siete [1].

8.11 Využitie Hopfieldovho modelu

Veľkou nevýhodou Hopfieldovho modelu neurónovej siete je, že sa nedokáže učiť z príkladov tak ako napríklad viacvrstvová sieť na základe metódy spätného šírenia sa chýb alebo Kohonenova samoorganizujúca sa mapa. Geometria pamäťových vzorov musí byť dopredu určená a slúži na naprogramovanie synaptickej matice.

Prvou oblasťou aplikácie Hopfieldovho modelu je **oblasť modelovania neurobiologických a psychických javov**. Pomocou Hopfieldovho modelu neurónovej siete a jeho rôznych variácií môžeme simulovať nasledovné kognitívne procesy:

- **Rozpoznávanie** — vnímaný objekt už bol raz kedysi vnímaný, je v pamäti.
- **Asociatívne vybavenie si z pamäti** — rekonštrukcia kompletnej položky (reprezentácie objektu) v pamäti, iniciovaná prítomnosťou fragmentu známeho vzoru na vstupe siete.
- **Klasifikácia** — konkrétny atraktor (resp. okolie atraktora) reprezentuje spracovaný a zapamätaný vzor. Pri modelovaní vnímania sú vstupom siete signály zo senzorických orgánov. Keď by sme modelovali iné mozgové funkcie ako napríklad myslenie, vstupmi budú konfigurácie aktivity z iných kortikálnych (resp. asociačných) častí mozgu. Opakujúci sa vzorec neurónovej aktivity celej siete (atraktor) je signálom toho, že nastala kognitívne signifikantná udalosť. Konkrétny vstup (zodpovedajúci počiatočnej konfigurácii aktivity siete) je kognitívne zmysluplný vtedy, keď vedie sieť rýchlo do okolia atraktora. Zmysel tomuto vstupu dáva práve daný atraktor, ktorý v minulosti vznikol na danej konkrétnej úrovni kognitívneho spracovania.

Ak chceme pomocou autoasociatívnych sietí simulovať kognitívne procesy, ako sú vnímanie, pamäť a iné, musia byť splnené o.i. tieto podmienky:

 Relaxačný čas (doba príchodu do blízkosti atraktora) musí byť dostatočne krátky, aby stav siete dosiahol atraktor ešte pred tým, ako je sieť postavená pred novú úlohu. Nová úloha môže byť reprezentovaná príchodom novej externej stimulácie, prípadne nejakým vnútorným mechanizmom, ktorý resetuje sieť. Pri väčšine kognitívnych úloh ide o časy rádovo desiatok až stoviek milisekúnd.

 Doba zotrvávania siete v blízkosti atraktora musí byť dostatočne dlhá. Ak pripisujeme atraktoru kognitívnu úlohu, výstupný mechanizmus musí byť schopný detekovať fakt, že systém skutočne dospel do blízkosti atraktora, čiže musí byť schopný odlíšiť tento stav od prechodných vzorcov aktivity siete. Odhaduje sa, že by sa mohlo uvažovať o časovom spriemernení aktivity počas obdobia 30̃40 ms [9].

Ďalej na niektorých príkladoch naznačíme, ako možno konkrétne aplikovať autoasociatívne neurónové siete na modelovanie kognitívnych procesov prebiehajúcich v mozgu resp. v nervovom systéme ako takom. Podrobnú informáciu možno nájsť v citovanej literatúre. Modely týkajúce sa nervového systému je v súčasnosti ťažko možno navrhovať a overovať na základe priamych meraní na neurónoch tak, ako to bolo možné v prípade modelu Kleinfelda a Sompolinského [51, 52] alebo Amita so spolupracovníkmi [6]. Kleinfeld a Sompolinsky navrhli analógovú autoasociatívnu neurónovú sieť, ktorá má také dynamické správanie ako majú tzv. **centrálne generátory rytmu** (neurónové okruhy generujúce a riadiace cyklické vzorce motorickej aktivity). Svoj model aplikovali na prípad únikového plávania mäkkýša *Tritonia diomedea*. Vzorce aktivity v sieti sa zhodovali s hodnotami nameranými pomocou elektród na skutočných neurónoch. S týmito hodnotami a Hopfieldovou dynamikou, umelá neurónová sieť produkovala cyklický výstup slúžiaci na riadenie svalov. Amit so spolupracovníkmi [6] skonštruovali analógovú autoasociatívnu sieť, ktorá reprodukuje **reverberácie aktivity** namerané **v mozgovej kôre** opíc učiacich sa zapamätať a rozpoznať časové sekvencie zrakových stimulov. Zrakové stimuly boli náhodné bitové obrázky a prezentovali sa vo fixnom poradí. Opica bola potom testovaná tak, že pri obrázkoch prezentovaných tentoraz v náhodnom poradí mala určiť, či sa dva po sebe idúce obrázky pri učení vyskytli za sebou alebo nie. Napriek tomu, že obrázky boli nekorelované (náhodné) vzorce aktivity, vzorce aktivity namerané v mozgovej kôre mali korelovanú distribúciu aktivity v závislosti od toho, či sa obrázky vyskytovali v postupnosti za sebou alebo nie. Veľkosť tejto korelácie bola funkciou vzájomnej "vzdialenosti" obrázkov v postupnosti (ich vzájomného poradia). Všetky empirické výsledky tejto štúdie boli reprodukované pomocou atraktorovej neurónovej siete. Treves s Rollsom [69] sa pomocou modelu atraktorovej siete pokúsili vysvetliť funkciu a **činnosť hippocampu** (mozgovej štruktúry, ktorá hrá dôležitú úlohu pri pamäťových procesoch u zvierat aj u človeka).

My sme skonštruovali stochastickú Hopfieldovu sieť s pamäťovými vzormi reprezentujúcimi hudobné tóny, pomocou ktorej sme simulovali **spektrálne invariantné rozpoznávanie hudobných tónov** [12]. Konkrétne išlo o teoretické vysvetlenie javu chýbajúcej základnej frekvencie. Treba vedieť, že zložené hudobné tóny používané v psychofyzikálnych experimentoch sa skladajú zo základnej frekvencie a z určitého konečného počtu vyšších frekvencií, ktoré sú vždy celistvým násobkom základnej frekvencie (tzv. vyššie harmonické). Tieto frekvencie definujú farbu tónu. Nota alebo výška tónu je určovaná hodnotou základnej frekvencie. Zaujímavosť fenoménu chýbajúcej

základnej frekvencie spočíva v tom, že človek počuje tón prislúchajúci určitej základnej frekvencii aj v prípade, že sluchový stimul obsahuje iba vyššie harmonické frekvencie a základná frekvencia v spektre zvuku fyzicky chýba. Pamäťové vzory sme zostrojili v súhlase so známou geometriou reprezentácie zvukových frekvencií v mozgovej kôre. Frekvencie počuteľného spektra sú reprezentované zhruba rovnobežnými pásmi neurónov, ktoré sú orientované kolmo v smere frekvenčného gradientu. Sledovali sme kvalitu vybavovania tónov (resp. ich neurónových reprezentácií) v závislosti od toho, ktoré vyššie harmonické frekvencie boli v zvuku prítomné a dosiahli sme zhodné výsledky s výsledkami psychofyzikálnych experimentov uskutočnených na ľuďoch. Neskôr sme použili Dotsenkov model invariantného rozpoznávania modifikovaný pomocou metódy simulovaného žíhania na **modelovanie transpozične invariantného rozpoznávania melódií**, vzhľadom ku konštantnému frekvenčnému posunu všetkých tónov [13]. Kvôli tomu bolo potrebné vytvoriť sieť s asymetrickou váhovou maticou, v ktorej boli zahrnuté synaptické časové oneskorenia, ktoré umožňujú rozpoznávať a generovať časové postupnosti konfigurácií.

Objavili sa aj **psychiatrické špekulácie** využívajúce pojmy atraktora, falošného atraktora a šumu na vysvetlenie obsahových a formálnych porúch reči a myslenia pri schizofrénii a mánii [39]. Normálna reč a myslenie by v tejto metafore súviseli s konvergenciou príslušnej mozgovej autoasociatívnej siete do správneho jedného bodového atraktora zodpovedajúceho jednému pojmu. Porucha reči a myslenia pri mánii by v tejto metafore zodpovedala spontánnym prechodom z jedného atraktora do druhého vplyvom príliš veľkého vnútorného šumu. Vysvetlenie špecifického charakteru obsahovej a formálnej poruchy reči a myslenia pri schizofrénii by zodpovedalo metafore, podľa ktorej mozgové autoasociatívne siete majú takú nízku úroveň šumu, že falošné atraktory, zmiešaniny pravých atraktorov (pojmov), sú veľmi dobrými atraktormi.

Druhou oblasťou problémov, na riešenie ktorých môžeme úspešne použiť algoritmy inšpirované teóriou neurónových sietí, sú **optimalizačné problémy**. Na neurónovú sieť Hopfieldovho typu sa môžeme pozerať ako na masívne paralelný algoritmus, ktorého cieľom je minimalizácia energie systému. V teórii optimalizácie sa vo všeobecnosti hovorí, že v priebehu optimalizácie sa minimalizuje **cenová resp. objektívna funkcia** (angl. ***cost*** *resp.* ***objective function***), ktorá v teórii neurónových sietí odpovedá účelovej (kriteriálnej) funkcii. Predtým ako si konkrétne načrtneme niektoré optimalizačné problémy, chceme ešte raz zdôrazniť, že keď sa hovorí, že tieto problémy sa riešia pomocou neurónových sietí, v skutočnosti sa pre riešenie každého z týchto problémov konštruuje špeciálne prispôsobený masívne paralelný algoritmus**.**

Ako prvé spomenieme **kombinatorické optimalizačné problémy** [38]. Pri týchto problémoch hľadáme riešenie v množine veľkého množstva možných kombinácií základných prvkov systému. Celkový počet riešení pre veľkosť problému s počtom prvkov N je obvykle exponenciálnou funkciou N, a tomu je úmerný aj čas potrebný na vyriešenie problému. Na ilustráciu si spomenieme ako prvý **problém váhovaného priradenia** (angl. ***weighted matching problem***). V priestore majme množinu N bodov, pričom poznáme vzdialenosti medzi jednotlivými dvojicami týchto bodov d_{ij}. Tieto body sa môžu nachádzať v euklidovskom priestore a d_{ij} môže reprezentovať Euklidovu vzdialenosť, alebo tieto body môžu byť abstraktné entity a hodnoty d_{ij} môžu reprezentovať ich vzťahy. Vo všeobecnosti sa môžu d_{ij} považovať za nezávislé náhodné premenné s pravdepodobnostnou distribúciou $P(d_{ij})$. Našou úlohou je pospájať dvojice bodov tak, aby každý bod bol spojený len s jedným iným bodom tak, aby celková dĺžka spojení bola minimálna. Praktickými príkladmi

takéhoto problému môže byť spájanie prvkov v nejakom elektronickom zariadení, optimálne mapovanie procesov na dva ekvivalentné procesory, priraďovanie žiakov do škôl, a pod. Každému páru bodov i a j, priraďme prvok n_{ij}, $i < j$. Nech $n_{ij}=1$, keď medzi i-tym a j-tym bodom existuje spojenie a $n_{ij}=0$, keď medzi nimi spojenie nie je. Sám so sebou sa bod nemôže spojiť, takže $n_{ii}=0$ a pre $j<i$ platí $n_{ij}=n_{ji}$. Funkcia, ktorej minimum hľadáme je celková dĺžka spojení

$$L=\sum_{i<j} d_{ij}n_{ij} \ . \tag{8.82}$$

Túto funkciu budeme musieť trochu modifikovať, lebo každé riešenie musí spĺňať jedno obmedzenie, a síce, že každý jeden bod môže byť spojený iba s jedným ďalším bodom, a teda $\sum_j n_{ij}=1$ pre $\forall i$. Dodržiavanie tohto obmedzenia sa rieši penalizáciou v objektívnej funkcii. Ako základ pre objektívnu funkciu zoberieme (8.82) a pridáme k nej člen, ktorý bude rásť priamo úmerne porušeniu danej podmienky. Objektívna funkcia E bude rovná:

$$E=\sum_{i<j} d_{ij}n_{ij}+\frac{\gamma}{2}\sum_i\left(1-\sum_j n_{ij}\right)^2 \ . \tag{8.83}$$

Veľkosť konštanty γ by mala byť asi taká ako priemerná hodnota d_{ij}. Pravdepodobnosť, že hodnota ľubovoľného prvku n_{ij} sa zmení, je rovná

$$P(n_{ij}\to n'_{ij})=\frac{1}{1+\exp(\beta\Delta E)} \quad kde \quad \Delta E=E(n'_{ij})-E(n_{ij}) \ . \tag{8.84}$$

Toto pravidlo hovorí, že zmeny, ktoré minimalizujú objektívnu funkciu E sú pravdepodobnejšie ako tie, ktoré ju zvyšujú. Hľadáme teda takú distribúciu n_{ij}, ktorá zodpovedá jednému z miním (8.83). Systém môžeme naštartovať z náhodnej konfigurácie n_{ij} a každú ďalšiu konfiguráciu generovať náhodne ako v **Monte Carlo simulácii** s tým, že každú novú konfiguráciu akceptujeme s pravdepodobnosťou (8.84). Je vhodné použiť i **metódu simulovaného žíhania** a v priebehu evolúcie znižovať teplotu $T=\beta^{-1}$ [16, 49]. Problém môžeme riešiť aj pomocou analógového systému buď počítačovou simuláciou alebo hardwarovou implementáciou, vtedy n_{ij} blízke 1 (0) budeme považovať za rovné 1 (0).

Ďalší kombinatorický optimalizačný problém, ktorý sa stal štandardom na testovanie efektívnosti optimalizačných metód je **problém obchodného cestujúceho (**angl. ***traveling salesman problem*****)** [42, 43]. Máme v priestore N bodov alebo miest, medzi dvojicami ktorých sú vzdialenosti d_{ij}. Našou úlohou je nájsť najkratšiu uzavretú dráhu, ktorá prechádza cez každé mesto len raz a vracia sa do pôvodného mesta, z ktorého sme vyštartovali. Praktické príklady zahŕňajú efektívne plánovanie ciest kamiónov, vlakov, lietadiel, ale aj úlohy v robotike, napr. pri plánovaní efektívneho pohybu ramena robota. Na reprezentáciu riešenia si zvolíme stochastický binárny prvok n_{ia}, pričom n_{ia}=1, keď mesto i je a-tou zastávkou na okružnej ceste. Celková dĺžka okružnej cesty je

$$L = \frac{1}{2}\sum_{ij,a} d_{ij} n_{ia}(n_{j,a+1} + n_{j,a-1}) \ . \tag{8.85}$$

Zároveň musia byť splnené dve obmedzenia, a to

$$\sum_a n_{ia} = 1 \quad \text{pre } \forall i \quad \text{a} \quad \sum_i n_{ia} = 1 \quad \text{pre } \forall a \ . \tag{8.86}$$

Prvé obmedzenie hovorí, že každé mesto *i* sa objaví počas okružnej cesty len raz, a druhé obmedzenie znamená, že každá zastávka na ceste sa týka len jedného mesta. Teraz môžeme vytvoriť objektívnu funkciu *E* s dvoma penalizáciami, ktoré sa minimalizujú vtedy, keď sú podmienky (8.86) splnené:

$$\mathrm{E} = \frac{1}{2}\sum_{ij,a} d_{ij} n_{ia}(n_{j,a+1} + n_{j,a-1}) + \frac{1}{2}\left[\sum_a \left(1 - \sum_i n_{ia}\right)^2 + \sum_i \left(1 - \sum_a n_{ia}\right)^2\right] \ . \tag{8.87}$$

Riešenie problému obchodného cestujúceho, teda konkrétne hodnoty n_{ia} zodpovedajúce optimálnej okružnej ceste, hľadáme tak ako v prípade problému váhovaného priradenia, teda pomocou počítačovej simulácie alebo hardwarovej implementácie príslušného systému. Používame buď binárne prvky so stochastickou dynamikou (8.84) a simulované žíhanie, alebo analógové prvky.

Posledným príkladom kombinatorického optimalizačného problému je **delenie grafov** (angl. ***graph bipartitioning***) [27]. Predstavme si, že navrhujeme čip s *N* prvkami, ale všetky sa nám naň nezmestia. Potom by sme chceli urobiť dva čipy, tak aby polovica prvkov bola na jednom a polovica prvkov na druhom a aby počet spojení medzi týmito dvoma čipmi bol minimálny. Uvažujme všeobecný graf, t.j. množinu *N* bodov, vrcholov grafu pospájaných hranami, ktoré spájajú dvojice vrcholov. Nech je *N* párne. Nech *p* je fixná pravdepodobnosť, že každý vrchol je spojený s iným. Priemerný počet vrcholov *pN*, ktoré sú navzájom spojené, sa nazýva valencia grafu. Našou úlohou je rozdeliť vrcholy do dvoch rovnako veľkých množín, medzi ktorými je minimálny počet hrán. Definujme si $c_{ij} = 1$, keď sú vrcholy *i* a *j* spojené hranou, a $c_{ij} = 0$, keď nie sú spojené. Ďalej si pre každý vrchol definujme premennú $S_i = +1$, keď sa vrchol nachádza v prvej množine a $S_i = -1$, keď sa vrchol nachádza v druhej množine. Chceme minimalizovať funkciu

$$L = -\sum_{\langle ij \rangle} c_{ij} S_i S_j \ , \tag{8.88}$$

kde $\langle ij \rangle$ znamená indexovanie cez každý pár vrcholov zvlášť. Premenné S_i podliehajú obmedzeniu $\sum_i S_i = 0$. V termínoch teórie magnetických látok tieto rovnice zodpovedajú feromagnetu s nulovou celkovou magnetizáciou. Každú celkovú magnetizáciu, ktorá sa vzďaľuje od 0 budeme v objektívnej funkcii penalizovať, takže objektívna funkcia systému je

$$\mathrm{E} = -\sum_{\langle ij \rangle} c_{ij} S_i S_j + \mu\left(\sum_i S_i\right)^2 \ . \tag{8.89}$$

Hodnotu μ vyberáme v intervale 0 až 1/2. Rovnica (8.89) pripomína energiu systému analogického spinovému sklu. Systém naštartujeme z náhodnej nekorelovanej konfigurácie $\boldsymbol{S}$ a necháme relaxovať, pričom S_i sa menia podľa stochastického pravidla (8.84) s S_i namiesto n_{ij}, kde energia E je vyjadrená ako (8.89). Opäť bude užitočné použiť simulované žíhanie. Feromagnet s nulovou celkovou magnetizáciou bol teoreticky extenzívne študovaný a boli nájdené mnohé zaujímavé vlastnosti (napríklad zánik symetrie replík) [59].

Ako **optimalizačný problém** môžeme formulovať aj **spracovanie obrazu** (angl. ***image processing***), čiže rekonštrukciu objektu zo zašumeného alebo rozmazaného obrazu [54]. Vstupom do paralelného systému je množina hodnôt d_i zodpovedajúcich dvojdimenzionálnej množine pixlov (*pixel* = angl. *picture cell*). Konkrétne hodnoty d_i môžu reprezentovať napríklad jas, alebo prvú či druhú priestorovú deriváciu jasu, alebo časovú deriváciu jasu, alebo binokulárnu disparitu (t.j. malú vzdialenosť medzi obrazmi toho istého bodu vnímaného pravým a ľavým okom, ktorú používa mozog na konštrukciu tretieho rozmeru objektu, resp. vzdialenosti jednotlivých objektov od pozorovateľa), alebo nejakú inú premennú, ktorá sa mení v priestore. Napríklad v prípade jasu môžu byť dáta binárne, t.j. svetlé alebo tmavé pixle, alebo spojité, reprezentujúce rozličné úrovne šedi v jednotlivých pixloch (angl. *gray-level data*). Vstupom do nášho systému budú zašumené dáta d_i a výstupom budú hodnoty V_i, ktoré reprezentujú rekonštruovaný objekt. Takto formulovaná úloha je veľmi ťažká, a preto sa vo väčšine prípadov používajú *a priorné* znalosti o objektoch, ako napríklad že majú (nemajú) hladký povrch, že majú rovné (zakrivené) okraje, že sa prekrývajú (neprekrývajú) s inými objektmi, alebo akékoľvek iné pomôcky. V prípade, že samotné dáta obsahujú príliš málo informácie, hovoríme, že tieto **problémy** sú **zle formulované** (angl. *ill-posed problems*). Existujú postupy a techniky ako transformovať zle formulované problémy na dobre formulované problémy (angl. *well-posed problems*). Týmto sa zaoberá tzv. **regularizačná teória** [64]. Postupy používané pri počítačovej rekonštrukcii objektu si naznačíme na najjednoduchšom prípade, keď máme obraz jednoliateho hladkého povrchu bez hrán. Uvažujme jednorozmerný prípad, viď obr. 8.11a. Najlepšiu rekonštrukciu hladkej nekonečnej krivky získame, keď budeme nútiť výstupy V_i, aby spĺňali dve podmienky zároveň: hladkosť a čo najväčšiu blízkosť k dátam, teda E bude

$$E = \frac{1}{2}\kappa \sum (V_i - V_{i+1})^2 + \frac{1}{2}\lambda \sum (V_i - d_i)^2 \ , \tag{8.90}$$

kde koeficienty κ a λ vyjadrujú relatívny dôraz na jednu alebo druhú podmienku. Minimum tejto objektívnej funkcie môžeme hľadať napríklad metódou záporného gradientu:

$$\kappa\tau \frac{dV_i}{dt} = -\frac{\partial E}{\partial V_i} = \kappa(V_{i+1} + V_{i-1} - 2V_i) + \lambda(d_i - V_i) \ . \tag{8.91}$$

Ak by sme položili $\lambda = 0$, a teda nevyžadovali blízkosť k dátam, pre každé V_i by sme dostali difúznu rovnicu. Jednotlivé hodnoty V_i by sa postupne rovnomerne rozptýlili. Nenulová hodnota koeficientu λ však núti V_i pridržať sa dát d_i, takže dostaneme hladkú krivku, ktorá sa snaží čo najviac približovať k dátam. Zovšeobecnenie pre dvojrozmerný prípad je v uvažovaní diferencií v (8.90) v rovine. Mohli by sme skonštruovať aj analógový obvod simulujúci rovnicu (8.91), ktorá je analogická dynamickej rovnici pre spojitú Hopfieldovu

sieť (8.77) alebo (8.79), keď k nej pridáme vonkajší vstup λd_i a $g(u) = u$. Ďalej s pomocou (8.76) môžeme identifikovať $\kappa\tau$ ako C, λ ako ρ^{-1}, a κ ako $1/R_{i,i+1}$.

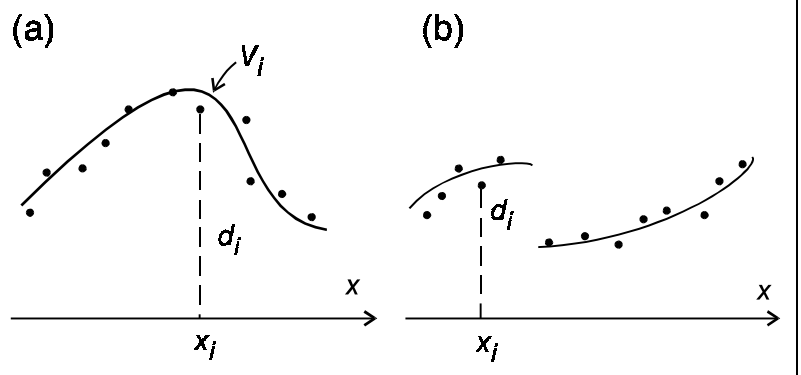
Aby sme mohli riešiť problém spracovania obrazu, ktorý zobrazuje povedzme dva objekty, musíme systém rovníc upraviť tak, aby bol schopný detekovať nespojitosti, a teda nachádzať po častiach hladké (spojité) riešenia (ilustráciu pre jednorozmerný prípad pozri na obr. 8.11b). Rieši sa to tak, že sa do systému zavedú diskrétne prvky detekujúce nespojitosti.

Ilustrujme si to na jednorozmernom prípade. Keďže nevieme dopredu, medzi ktorými bodmi sa bude nachádzať medzera (nespojitosť), medzi každý pár analógových prvkov s výstupom V_i a V_{i+1} dáme jeden binárny prvok s výstupom S_i. Hodnota $S_i = +1$ bude znamenať hypotézu, že medzi i-tym a (i+1)-vým bodom je medzera a hodnota $S_i = \tilde{1}$, bude znamenať, že je tam krivka spojitá.

Objektívnu funkciu (8.90) treba modifikovať tak, že (a) zrušíme penalizáciu za rozdiel medzi hodnotou V_i a V_{i+1}, keď je medzi nimi predpokladaná nespojitosť a (b) budeme penalizovať príliš veľa hypotéz o nespojitostiach. Nová objektívna funkcia bude teraz vyzerať takto:

$$E = \frac{1}{2}\kappa \sum_i \tfrac{1}{2}(1 - S_i)(V_i - V_{i+1})^2 + \frac{1}{2}\lambda \sum_i (V_i - d_i)^2 + \mu \sum_i S_i \quad . \qquad (8.92)$$

Pri relaxácii systému budú diskrétne prvky skúšať hypotézy o nespojitostiach stochastickým spôsobom. Výsledok môžeme vylepšiť aj tým, že v priebehu relaxácie budeme meniť hodnoty koeficientov κ, λ a μ. Objektívna funkcia (8.92) pre spracovanie dvojrozmerného obrazu zobrazujúceho viacero objektov bude



Obrázok 8.11. Ilustrácia jednorozmerného prípadu fitovania (a) hladkej a (b) po častiach hladkej krivky na zašumené dáta pri optimalizačnom probléme spracovania obrazu.

$$E = \frac{1}{2}\kappa \sum_{\langle ij \rangle} \frac{1}{2}(1 - S_{ij})(V_i - V_j)^2 + \frac{1}{2}\lambda \sum_i (V_i - d_i)^2 + \mu \sum_{\langle ij \rangle} S_{ij} - \gamma \sum_{\langle ijkl \rangle} S_{ij} S_{jk} S_{kl} S_{li} \quad , (8.93)$$

kde sa sumuje cez dvojice bezprostredných susedov $\langle ij \rangle$ na dvojrozmernej mriežke, alebo cez štvorice susedov $\langle ijkl \rangle$ tvoriace štvorce. Posledný člen nabáda systém, aby sa nespojitosti tvoriace kontúry objektu spájali do uzavretých kriviek.

Rovnice spracovania obrazu prezentované v tomto oddieli nemusia slúžiť len ako základ na interpoláciu povrchov objektov. Môžu slúžiť na detekciu hrán, na rekonštrukciu trojrozmerného tvaru objektov z ich tieňovania alebo binokulárnej disparity, rekonštrukciu tvaru objektov na základe farby alebo pohybu, a pod. Záleží to na príslušnej modifikácii týchto rovníc, na použití analógových alebo stochastických binárnych prvkov, na hodnotách koeficientov, a samozrejme na interpretácii premenných d_i a V_i.

Literatúra

[1] Y.S. Abu-Mostafa and D. Psaltis. Optical neural computers. *Scientific American*, 256: 88-95, 1987.

[2] J. Alspector and R.B. Allen. A neuromorphic VLSI learning system. In: P. Losleben, editor, *Advanced Research in VLSI: Proceedings of the 1987 Stanford Conference*, pages 313-349, MIT Press, Cambridge, 1987.

[3] D. J. Amit and S. Fusi. Constraints on learning dynamic synapses. *Network: Computation in Neural Systems*, 3: 443-449, 1992.

[4] D. J. Amit and S. Fusi. Dynamic learning in neural networks with material synapses. *Neural Computation*, 4: 957-982, 1994.

[5] D. J. Amit and N. Brunel. Adequate input for learning in attractor neural network. *Network: Computation in Neural Systems*, 4: 177-186, 1993.

[6] J.A. Amit, N. Brunel, and M.V. Tsodyks. Correlations of cortical Hebbian reverberations: Theory versus experiment. *The Journal of Neuroscience*, 14: 6435-6445, 1994.

[7] D.J. Amit, H. Gutfreund, and H. Sompolinsky. Information storage in neural networks with low levels of activity. *Physical Review A*, 35: 2293-2303, 1987.

[8] D.J. Amit. Neural networks counting chimes. *Proceedings of the National Academy of Sciences of the USA*, 85: 2141-2145, 1988.

[9] D.J. Amit. *Modeling Brain Function. The World of Attractor Neural Networks*. Cambridge University Press: Cambridge, New York, Sydney, 1989.

[10] D.J. Amit, H. Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Physical Review A*, 32: 1007-1018, 1985a.

[11] D.J. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural network. *Physical Review Letters*, 55: 1530-1533, 1985b.

[12] Ľ. Beňušková. Modelling the effect of the missing fundamental with an attractor neural network. *Network: Computation in Neural Systems*, 5: 333-349, 1994.

[13] Ľ. Beňušková. Modelling transpositional invariancy of melody recognition with an attractor neural network. *Network: Computation in Neural Systems*, 6: 313-331, 1995.

[14] Ľ. Beňušková, M.E. Diamond, and F.F. Ebner. Dynamic synaptic modification threshold: computational model of experience-dependent plasticity in adult rat barrel cortex. *Proceedings of the National Academy of Sciences of the USA*, 91: 4791-4795, 1994.

[15] E. Bienenstock and C. von der Malsburg. A neural network for invariant pattern recognition. *Europhysics Letters,* 4: 121-126, 1987.

[16] V. Černý. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. Optim. Appl.*, 45: 41-51, 1985.

[17] B.G. Cragg and H.N.V. Temperley. The organization of neurones: a cooperative analogy. *EEG Clin. Neurophysiol.*, 6: 85-92, 1954.

[18] B.G. Cragg and H.N.V. Temperley. Memory: the analogy with ferromagnetic hysteresis. *Brain*, 78: 304-316, 1955.

[19] A. Crisanti, D.J. Amit, and H. Gutfreund. Saturation level of the Hopfield model for neural network. *Europhysics Letters*, 2: 337-341, 1986.
[20] A. Crisanti and H. Sompolinsky. Dynamics of spin systems with randomly asymmetric bonds: Langevin dynamics and a spherical model. *Physical Review A*, 36: 4922-4939, 1986.
[21] B. Derrida, E. Gardner, and A. Zippelius. An exactly solvable asymmetric neural network model. *Europhysics Letters*, 4: 167-173, 1987.
[22] B. Derrida and J.P. Nadal. Learning and forgetting on asymmetric, diluted neural networks. *Journal of Statistical Physics*, 49: 993-1009, 1987.
[23] D. W. Dong and J.J. Hopfield. Dynamic properties of neural networks with adapting synapses. *Network: Computation in Neural Systems*, 3: 267-275, 1992.
[24] V.S. Dotsenko. Neural networks: translation-, rotation- and scale-invariant pattern recognition. *Journal of Physics A: Mathematics and General*, 21: L783-L787, 1988.
[25] W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. I*, John Wiley and Sons, New York, 1968.
[26] P. Fedor, Ľ. Beňušková, H. Jakeš, and V. Majerník. An electrophoretic coupling mechanism between efficiency modification of spine synapses and their stimulation. *Studia Biophysica*, 92: 141-146, 1982.
[27] Y. Fu and P.W. Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimization. *Journal of Physics A: Mathematics and General*, 19: 1605-1620, 1986.
[28] E. Gardner. Maximum storage capacity in neural networks. *Europhysics Letters*, 4: 481-485, 1987.
[29] E. Gardner. Multiconnected neural network models. *Journal of Physics A: Mathematics and General*, 20: 3453-3464.
[30] E. Gardner. The space of interactions in neural networks models. *Journal of Physics A: Mathematics and General*, 21: 257-270, 1988.
[31] E. Gardner and B. Derrida. Optimal storage properties of neural networks models. *Journal of Physics A: Mathematics and General*, 21: 270-284, 1988.
[32] R.J. Glauber. Time-dependent statistics of the Ising model. *Jornal of Mathematical Physics*, 4: 294-307, 1963.
[33] H.P. Graf, L.D. Jackel, R.E. Howard, B. Straughn, J.S. Denker, W. Hubbard, D.M. Tennant, and D. Schwartz. VLSI implementation of a neural network memory with several hundreds of neurons. In: J.S. Denker, editor, *Neural Networks for Computing (Snowbird 1986)* , pages 182-187, American Institute of Physics, New York , 1986.
[34] H. Gutfreund. Neural networks with hierarchically correlated patterns. *Physical Review A*, 37: 570-586, 1988.
[35] H. Gutfreund and M. Mézard. Processing temporal sequences in neural networks. *Physical Review Letters*, 61: 235-247, 1988.
[36] D.O. Hebb. *Organization of Behavior*. J. Wiley and Sons, New York, 1949.
[37] J.L. van Hemmen and R. Kuhn. Nonlinear neural networks. *Physical Review Letters*, 57: 913-916, 1986.
[38] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*, Addison-Wesley Publ. Comp. , Redwood City , 1991.
[39] R.E. Hoffman. Computer simulations of neural information processing and the schizophrenia-mania dichotomy. *Archives of general Psychiatry*, 44: 178-190, 1987.

[40] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79: 2554-2558, 1982.
[41] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the USA* , 81: 3088-3092, 1984.
[42] J.J. Hopfield and D.W. Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52: 141-152, 1985.
[43] J.J. Hopfield and D.W. Tank. Computing with neural circuits: a model. *Science*, 233: 625-633, 1986.
[44] E. Ising. Bertrag zur Theorie des Ferromagnetism, *Zeitschrift Für Physik*, 31: 253-287, 1925.
[45] J.J.B. Jack, D. Noble, and R.W. Tsien. *Electric Current Flow in Excitable Cells*. Clarendon Press, Oxford, 1975.
[46] E.R. Kandel, J.H. Schwartz, and T.M. Jessell. *Principles of Neural Sciences*, Elsevier Science Publishing Co., New York, 1985.
[47] I. Kanter and H. Sompolinsky. Associative recall of memory without errors. *Physical Review A*, 35: 380--392, 1987.
[48] S. Kirkpatrick and D. Sherrington. Infinite-ranged models of spin-glasses. *Physical Review B*, 17: 4384-4403, 1978.
[49] S. Kirkpatrick, C.D. Jr. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220: 671-680. 1983.
[50] D. Kleinfeld and I. Kanter. Temporal association in asymmetric neural networks. *Physical Review Letters*, 57: 2861-2864, 1986.
[51] D. Kleinfeld and H. Sompolinsky. Associative neural network model for the generation of temporal patterns. *Biophysics Journal*, 54: 1039-1051, 1988.
[52] D. Kleinfeld and H. Sompolinsky. Associative network models for central pattern generatosrs. In: C. Koch and I. Segev, editors, *Methods in Neuronal Modeling: From Synapses to Networks*, The MIT Press, Cambridge, 1989.
[53] C. Koch and I. Segev, editors. *Methods in Neuronal Modeling: From Synapses to Networks*, The MIT Press, Cambridge, 1989.
[54] C. Koch, J. Marroquin, and A. Yuille. Analog "neuronal" networks in early vision. *Proceedings of the National Academy of Sciences of the USA*, 83: 4263-4267, 1986.
[55] R. Kree and A. Zippelius. Continuous-time dynamics of asymmetrically diluted neural networks. *Physical Review A*, 36: 4421-4427, 1987.
[56] R. Kree and A. Zippelius. Recognition of topological features of graphs and images in neural networks. *Journal of Physics A: Mathematics and Geneneral*, 21: L813-L818, 1988.
[57] W.A. Little. The existence of persistent states in the brain. *Mathematical Biosciences*, 19: 101-120, 1974.
[58] W.A. Little and G.L. Shaw. Analytic study of the memory storage capacity of a neural network. *Mathematical Biosciences*, 39: 281-290, 1978.
[59] M. Mézard, J.-P. Nadal, and G. Toulouse. Solvable models of working memories. *J. Physique*, 47: 1457-1468, 1986.
[60] J.-P. Nadal, G. Toulouse, J.-P. Changeux, and S. Dehaene. Networks of formal neurons and memory palimpsests. *Europhysics Letters*, 1: 535-542, 1986.

[61] G. Parisi. Asymmetric neural networks and the process of learning. *Journal of Physics A: Mathematics and Geneneral*, 19: L675-L680, 1986.
[62] P. Peretto. Collective properties of neural networks: a statistical physics approach. *Biological Cybernetics*, 50: 51-62, 1984.
[63] P. Peretto and J.-J. Niez. Stochastic dynamics of neural networks. *IEEE Trans. Syst. Man Cybern*., 16: 73-83, 1986.
[64] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 347: 314-319, 1985.
[65] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes*. Cambridge University Press, Cambridge, 1986.
[66] S. Shinomoto. A cognitive associative memory. *Biological Cybernetics*, 57: 197-214, 1987.
[67] H. Sompolinsky. The theory of neural networks: The Hebb rules and beyond. In: J.L. van Hemmen and I. Morgenstern, editors, *Heidelberg Colloqium on Glassy Dynamics*, pages 485-527, Heidelberg , 1987.
[68] H. Sompolinsky, A. Crisanti, and H.J. Sommers. Chaos in random neural networks. *Physical Review Letters*, 61: 259-262, 1988.
[69] A. Treves and E.T. Rolls. Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network. *Hippocampus*, 2: 625-639, 1992.
[70] M.V. Tsodyks and M.V. Feigel'man. The enhanced storage capacity in neural networks with low activity level. *Europhysics Letters*, 6: 101-105, 1988.

9. Evoluční algoritmy a neuronové sítě

9.1 Úvod

Evoluční algoritmy jsou zastřešujícím termínem pro řadu přístupů využívajících modelů evolučních procesů pro účely téměř nikdy nemající nic společné s biologií. Snaží se využít představ o hnacích silách evoluce živé hmoty (případně simulace tvorby krystalů v případě simulovaného žíhání) pro účely optimalizace. Všechny tyto modely pracují s náhodnými změnami navrhovaných řešení. Pokud jsou tato nová řešení výhodnější, nahrazují předcházející řešení.

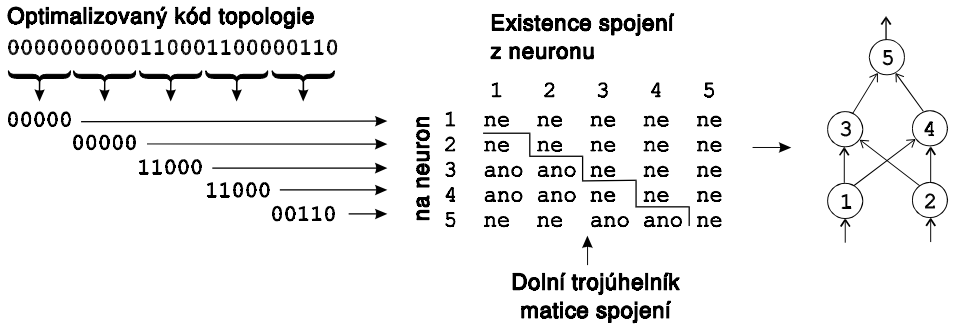
Evoluční algoritmy se používají při stochastické optimalizaci neuronových sítí. Tyto algoritmy využívají informaci z předcházejícího řešení při návrhu nové, lepší sítě. Snažíme se přitom, aby suma čtverců odchylek výstupů z neuronové sítě od předem zadaných hodnot byla co nejmenší. V této kapitole se nadále budeme zabývat typem "*feedforward*", t.j. dopředných vícevrstvých neuronových sítí. Tam, kde se budeme zabývat optimalizací topologie sítě, bude učící strategii tvořit ten nejznámější algoritmus — "*backpropagation*", t.j. metoda adaptace pomocí zpětného šíření chyb. Toto zjednodušení je oprávněné z toho důvodu, že optimalizace pomocí stochastických evolučních algoritmů se v naprosté většině používá právě pro tento základní typ neuronové sítě.

V optimalizaci neuronových sítí se vyskytují dva základní úkoly: optimalizace topologie a optimalizace vah.

■ *Optimalizace topologie neuronové sítě* spočívá v určení počtu skrytých vrstev, počtu neuronů v těchto vrstvách, existence spojení mezi nimi (obr. 9.1), popřípadě i parametrů přechodové funkce neuronu nebo parametrů pro učení sítě pomocí zpětného šíření.

Příklad uvedený na obr. 9.1 ukazuje pouze jednu z možností, co všechno se dá optimalizovat. Daný příklad se dá ještě zjednodušit. Pokud bychom nechtěli pokaždé kontrolovat, jestli je síť opravdu "feed forward", tedy jestli náhodou některá spojení netvoří cyklus, pak je nejjednodušší vyplnit pouze dolní trojúhelník matice spojení.

Tento typ zakódování topologie se nazývá "nízkoúrovňovým", protože přímo kóduje topologii. U jiných typů zakódování, tzv. "vysokoúrovňových", může být architektura sítě specifikována pravidly růstu nebo větami formálního jazyka. Tento typ je více vhodný pro optimalizaci rozsáhlejších sítí (na kterou v našich podmínkách ale stejně nemáme dostatečně rychlý hardware).

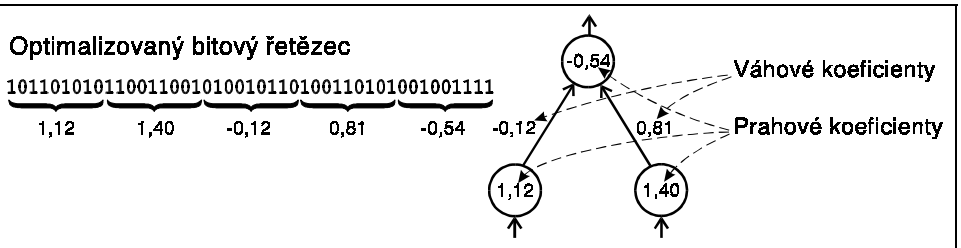


Obrázek 9.1. Příklad zakódování topologie spojení pěti neuronů. Optimalizace topologie neuronové sítě je převedena na optimalizaci úvodního bitového řetězce, přičemž první dva neurony jsou definované jako vstupní a poslední pátý neuron je výstupní.

Stochastické optimalizační algoritmy jsou v podstatě jediným systematickým přístupem k optimalizaci topologie sítě. Bohužel, tyto algoritmy potřebují vygenerovat a ohodnotit řádově tisícovky (nebo více) možných topologií neuronových sítí, aby dospěly k dobrému výsledku. Vzhledem k tomu, že ohodnocení každé topologie představuje vlastně naučení jedné neuronové sítě pomocí tisíců iterací zpětného šíření, jde o výpočetně velmi náročný úkol. Proto se místo systematického přístupu k návrhu topologie stále běžně používá spíše intuice, a příklady optimalizace topologie neuronové sítě existují většinou jen pro jednoduché problémy. Návrh sítě pro složitější problémy je přitom často doprovázen nepřípustným zjednodušováním optimalizačních algoritmů. Jejich hlavní síla, jíž je prohledávání mnoharozměrného prostoru s více minimy a schopnost dostat se z lokálních minim a skončit v globálním minimu, se pak snižuje. Zvyšuje se tím pravděpodobnost, že topologie skončí někde v lokálním minimu, t.j. že nebude ideální. Bohužel výpočetní nároky těchto metod lepší prohledávání možných topologií neumožňují.

Kromě počtu vrstev, počtu neuronů ve vrstvách a existence spojení neuronů může být optimalizováno mnoho dalších věcí. Ani optimalizovaná funkce nemusí být pouze sumou odchylek chyb předpovědí už známých faktů. Síť může být současně optimalizována např. na rychlost učení, nebo na malý počet spojení mezi neurony, nebo na nízký počet vnitřních neuronů. Existuje i speciální zakódování potenciální sítě tak, aby se lehce měnily počty neuronů a vrstev, viz [1], ale tento postup má zase jiné nedostatky a nestojí na nějaké hluboké teorii, proto jej zde nebudeme uvádět.

■ *Optimalizace váhových a prahových koeficientů* spojení v neuronových sítích (obr. 9.2), která nahrazuje metodu zpětného šíření, je další úlohou často řešenou pomocí evolučních metod. Evoluční metody však pro tuto optimalizaci nejsou nejtypičtější, protože klasická metoda zpětného šíření poskytuje dobré výsledky a je výpočetně méně náročná. Klasické algoritmy optimalizující sumu kvadrátu odchylek založené na derivaci "chybové" (účelové) funkce automaticky končí v nejbližším lokálním optimu. Naštěstí hyperplocha optimalizované funkce většinou nemá příliš mnoho lokálních minim a výsledky metody zpětného šíření jsou většinou přijatelné. Pokud se optimalizace nedaří ani po několika "nástřelech" počátečních vah, stačí často přidat několik skrytých neuronů. Chceme-li však vytvořit efektivní síť s minimem vnitřních neuronů a spojů, jsou evoluční algoritmy nenahraditelné. Dokáží relativně rychle vyhledat váhové a prahové koeficienty blízké



Obrázek 9.2. Zobrazení převodu bitového řetězce na váhové a prahové koeficienty, t.j. reálná čísla. Místo vektoru tvořeného reálnými čísly je pak možné optimalizovat bitový řetězec.

optimálním, trvá jim však dlouho přechod od téměř optimálních vah k optimálním vahám [2,3]. Tento nedostatek se často nahrazuje spojením evolučního algoritmu s metodou zpětného šíření — evoluční algoritmus najde přibližné hodnoty optima, a zpětné šíření dokončí optimalizaci do globálního optima.

Některé modifikace stochastických evolučních algoritmů pracují sice přímo s reálnými proměnnými, přesto však je nutné zde uvést **bitovou reprezentaci proměnných** [4].

Nechť $f(\boldsymbol{x})$ je funkce N proměnných (třeba váhových a prahových koeficientů neuronů) definovaná na kompaktní oblasti $D \subseteq R^N$, kde $R=(-\infty,\infty)$ je množina všech reálných čísel a $\boldsymbol{x}=(x_1,x_2,...,x_N)$ je vektor proměnných. Hledejme globální minimum $\boldsymbol{x}^*$ funkce $f(\boldsymbol{x})$ na oblasti D, $f(\boldsymbol{x}^*) = \min f(\boldsymbol{x})$, pro $\boldsymbol{x} \in D$. Zavedeme novou reprezentaci vektoru $\boldsymbol{x}$, která bude realizována pomocí binárního řetězce obsahujícího symboly 0 a 1. Předpokládejme, že každá proměnná x_i (pro $i=1,2,...,N$) je ohraničená zleva a zprava čísly a resp. b, $a \le x_i \le b$. Dále předpokládejme, že proměnné x jsou určeny s přesností na q dekadických míst za desetinnou čárkou. Potom je x přetransformováno nejprve na celé číslo a následovně na binární číslo. Na binární reprezentaci pak potřebujeme k binárních číslic, přičemž délka k binární reprezentace je určena celočíselným řešením rovnice $(b-a)\,10^q = 2^k$,

$$k = \left\lceil \frac{\ln\left[(b-a)10^q\right]}{\ln 2} \right\rceil, \tag{9.1}$$

kde $\lceil \beta \rceil$ je nejbližší větší celočíselná hodnota reálného čísla β, např. $\lceil 1{,}2 \rceil = \lceil 1{,}9 \rceil = 2$. Každému reálnému číslu $x \in \langle a,b \rangle$, vyjádřenému s přesností q, transformací (9.2) přiřadíme celé číslo $y_{(10)}$, jehož binární reprezentace $y_{(2)}$ obsahuje k binárních číslic.

$$x \to y_{(10)} = \left\lceil \frac{x-a}{b-a}\left(2^k - 1\right) \right\rceil \tag{9.2}$$

Inverzním postupem můžeme vytvořit z binárního čísla reálné číslo z intervalu $\langle a,b \rangle$.

Pro lepší pochopení uvedeme jednoduchý ilustrační příklad. Nechť x je reálné číslo z intervalu $-1 \le x \le 2$, vyjádřené s přesností na dvě dekadická místa za desetinnou čárkou, $q=2$. Pomocí vztahu (9.1) určíme konstantu k, tedy $k=9$. Nechť $x=1{,}12$, potom pomocí (9.2) určíme celé číslo $y_{(10)}=362$, a jeho binární reprezentace je $y_{(2)}=101101010$. Pro ilustraci inverzního postupu budeme studovat $y_{(2)}=110011001$, přiřazené dekadické celé číslo je $y_{(10)}= 409$, a použitím inverzního vztahu k rovnici (9.2) dostaneme $x=1{,}40$. Minimální

(maximální) binární číslo délky k=9 tvaru 000000000 (111111111) odpovídá dolní (horní) hranici intervalu, x=−1,00 (x=2,00). Vektor proměnných $\boldsymbol{x}=(x_1,x_2,...,x_N)$ lze pak vyjádřit bitovým řetězcem sestaveným ze za sebou jdoucích binárních reprezentací proměnných x_i. Zavedením binární reprezentace proměnných s ohraničenou přesností jsme redukovali původně spojitý optimalizační problém na diskrétní optimalizační problém.

■ *Extrahování pravidel z neuronové sítě* je základním problémem neuronových sítí, který se jen částečně daří řešit. Tato situace je akutní ve finančnictví, kde se investoři nervózně strachují o svoje peníze, jejichž investice jsou řízeny pomocí neuronových sítí, v jaderných elektrárnách, kdy jsou operátoři neochotní předat řízení něčemu, co není schopno ani vysvětlit svá rozhodnutí, nebo ve vědě, kdy výzkumníci tajně doufají, že se jim rozšifrováním předpovědí neuronových sítí podaří nalézt nový přírodní zákon. V podstatě nejúspěšnější je dosud přepis zadání s výsledky do tvaru pravidel jako u expertního systému. V principu by za tímto účelem ani nebylo třeba neuronových sítí, stačila by rozsáhlá databanka vstupů s požadovanými výstupy. Neuronové sítě se zde používají, je-li dat relativně málo, na jejich doplnění svými vypočítanými výstupy [5].

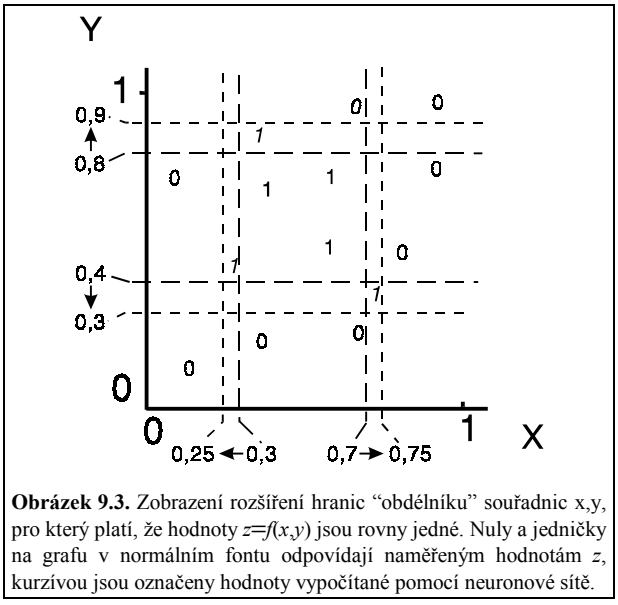
Velmi zjednodušeně si postup získávání pravidel můžeme představit z obr. 9.3, kdy dvě nezávislé proměnné x a y dávají dva možné výsledky pro z, 0 nebo 1. Pravidlo určující maximální možný rozsah výskyt hodnot 1 pak můžeme vytvořit například následovně:

IF((x>=0,3 AND x<=0,7) AND ((y>=0,4 AND y<=0,8)) THEN z=1

Lepším pravidlem je však takové, které má větší hranice, např.

IF((x>=0,25 AND x<=0,75) AND ((y>=0,3 AND y<=0,9)) THEN z=1

Vytváření a posouvání rozsahu hodnot těchto pravidel je otázkou zakódování, všechna pravidla i se svými hodnotami jsou zakódována ve formě binárního řetězce. K optimalizaci těchto pravidel se používá genetických algoritmů. Rozsah hodnot pravidel a jejich platnost pak ohodnocuje daný binární řetězec. Pokud však v daném rozsahu hodnot pravidel neexistují naměřená data se známými výstupy, simulují se výstupy pomocí neuronové sítě, a takto vytvořená data se pak používají k ohodnocení platnosti pravidel.



Obrázek 9.3. Zobrazení rozšíření hranic "obdélníku" souřadnic x,y, pro který platí, že hodnoty $z=f(x,y)$ jsou rovny jedné. Nuly a jedničky na grafu v normálním fontu odpovídají naměřeným hodnotám z, kurzívou jsou označeny hodnoty vypočítané pomocí neuronové sítě.

9.2 Přehled a základní vlastnosti stochastických optimalizačních algoritmů

Stochastické optimalizační algoritmy se používají pro optimalizaci mnohoparametrových funkcí s "divokým" průběhem, t.j. s mnoha extrémy, nebo neznámým gradientem. Standardní gradientové metody (např. nejprudšího spádu, sdružených gradientů, proměnné metriky [6,7]) nebo negradientové metody (např. simplexová [7]) nejsou vhodnými přístupy tehdy, když požadujeme nalezení globálního extrému funkce s mnoha extrémy. Tyto metody obvykle konvergují jen k extrému blízko od startovního bodu, a tento extrém už nejsou schopné opustit. Tento nedostatek se obvykle odstraňuje jejich randomizací tak, že se opakovaně náhodně zvolí počáteční řešení optimalizační úlohy a za výsledné řešení se vezme nejlepší výsledek. Stochastičnost tohoto postupu spočívá jen v náhodném výběru počátečního řešení, následovně použitý optimalizační algoritmus je striktně deterministický.

Bohužel se tento přístup u optimalizace neuronových sítí často používá ze striktně statistického pohledu nesprávně. Síť se optimalizuje gradientovou metodou pro tréninkovou množinu, ohodnotí se pomocí testovací množiny, a pokud výsledek nevyhovuje, začne se znovu s jinými náhodně zvolenými startovními vahami. Tento přístup ve svém důsledku zahrnuje vlastně testovací množinu do trénovací. Správný přístup by měl ohodnocovat řešení pouze na základě chyby dosažené pro trénovací množinu a až vybraná natrénovaná síť by měla být testována pomocí testovací množiny. Pokud se výsledek nepodařil, pak

máme smůlu a správně bychom měli nadále použít jinou trénovací a testovací množinu (k čemu nám většinou chybí data).

Stochastické optimalizační algoritmy si zachovávají svoji "stochastičnost" v celém průběhu optimalizačního procesu a ne jen ve výběru počátečního řešení. Navíc pro tyto metody byly dokázány existenční teorémy, které za jistých předpokladů zabezpečují jejich schopnost nalezení globálního extrému (ovšem v nekonečném čase). Programová implementace těchto metod je poměrně jednoduchá. Jednou z hlavních podmínek jejich úspěšného použití je vhodná reprezentace proměnných pomocí řetězce znaků (např. bitového řetězce obsahujícího symboly 0-1) a rychlost výpočtu hodnot účelové funkce v daném bodě. Zvláště tato poslední podmínka podstatně limituje úspěšné použití stochastických optimalizačních metod pro optimalizaci neuronových sítí, jednoduchost je "kompenzována" náročností na výpočetní čas.

U optimalizace neuronových sítí optimalizujeme jak diskrétní, tak i reálné proměnné. Stochastické optimalizační metody nutně fungují pomaleji než jakékoli heuristické přístupy. Pokud nemáme dopředu zadané podmínky pro globální extrém, nikdy nevíme, jestli jsme ho dosáhli a máme-li optimalizaci zastavit. Stochastické optimalizační metody však mají i zásadní výhody: jsou velmi obecně formulované a tedy aplikovatelné téměř na jakýkoli problém a dokáží se dostat z lokálního extrému. Evoluční proces prohledávání prostoru potenciálních řešení vyžaduje rovnováhu dvou cílů:

- *co nejrychleji* najít nejbližší (většinou lokální) optimum v malém okolí výchozího bodu
- *co nejlépe* prohledat prostor všech možných řešení.

Metody se liší svým zaměřením k těmto dvěma cílům, a je zhruba možné je seřadit podle posloupnosti od metod jdoucích k lokálnímu optimu až k metodám prohledávajícím prostor řešení. Tato posloupnost je následovná:

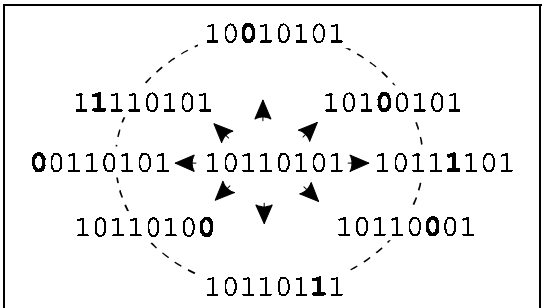
1. stochastický “horolezecký” algoritmus
2. tabu search
3. simulované žíhání
4. evoluční strategie
5. genetické algoritmy

Pro optimalizaci topologie neuronových sítí se zatím prakticky výhradně používají genetické algoritmy, pro optimalizaci váhových a prahových koeficientů se používají genetické algoritmy a simulované žíhání.

Při používání genetických algoritmů jde však spíše o zvyk, ostatní algoritmy nejsou zásadně horší, záleží vždy spíše na nastavení různých parametrů v algoritmech. Ostatně, v poslední době se stále více používá “směsi” algoritmů, t.j. metod, které přebírají a kombinují několik základních přístupů do jednoho. Zde však uvedeme základní algoritmy odděleně.

9.3 Stochastický "horolezecký" algoritmus

Název metody "horolezecký" algoritmus je volným překladem anglického termínu "hill climbing" [4]. Jde vlastně o variantu gradientové metody "bez gradientu", kdy se směr nejprudšího spádu určí kompletním prohledáním okolí. Tento algoritmus také trpí základní nectností gradientových metod, t.j. nejspíše skončí v lokálním optimu, a nedosáhne globálního optima. Vychází se zde z náhodně navrženého řešení. Pro momentálně navržené řešení se generuje pomocí konečného souboru transformací určité okolí (viz obr. 9.4) a funkce se minimalizuje jen v tomto okolí. Získané lokální řešení se použije jako "střed" nového okolí, ve kterém se lokální minimalizace opakuje.



Obrázek 9.4. Bitový řetězec vprostřed slouží jako centrum, jeho okolí je vytvořeno vždy překlopením jednoho bitu. Bitové řetězce okolí se ohodnotí, a nejvýhodnější z nich postupuje jako nový střed do další iterace.

Tento proces se iteračně opakuje předepsaný počet-krát (viz obr. 9.5). V průběhu celé historie algoritmu se zaznamenává nejlepší řešení, které slouží jako výsledné optimální řešení. Základní nevýhodou tohoto algoritmu je, že se po určitém počtu iteračních kroků vrací k lokálně optimálnímu řešení, které se již vyskytlo v předcházejícím kroku (problém zacyklení, viz obr. 9.6). Tento problém se obchází tak, že se algoritmus spustí několikrát s různými náhodně vygenerovanými počátečními řešeními, a poté se vybere nejlepší výsledek.

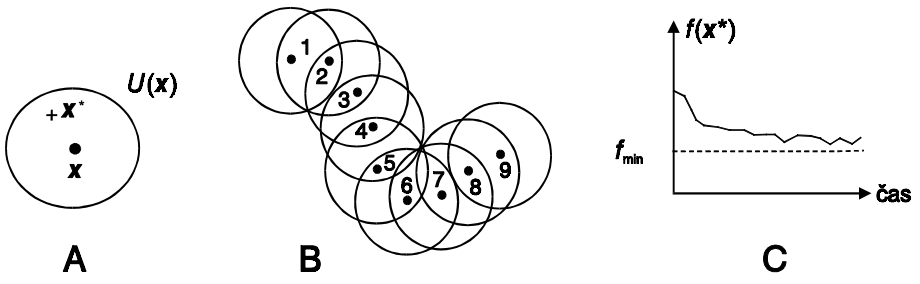
Minimalizace funkce $f(\boldsymbol{x})$ na oblasti D (například oblasti všech možných osmibitových řetězců) spočívá v hledání řešení $\overline{\boldsymbol{x}}$ (např. konkrétního osmibitového řetězce), které poskytuje minimální funkční hodnotu na oblasti D,

$$f(\overline{\boldsymbol{x}}) = \min_{\boldsymbol{x} \in D} f(\boldsymbol{x}). \tag{9.3}$$

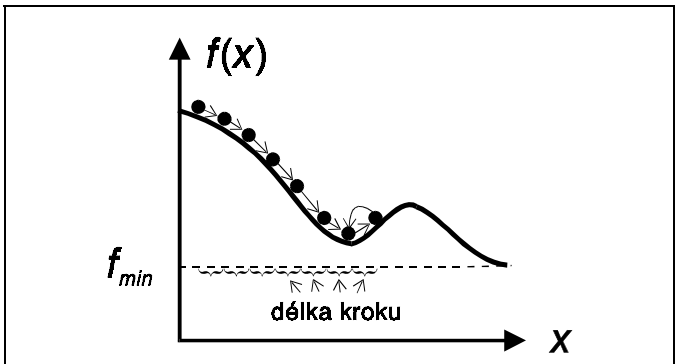
Definujme množinu přípustných transformací S, kde transformace $t \in S$ (například překlopení všech možných bitů řetězce) zobrazuje vektor $\boldsymbol{x} \in D$ na jiný vektor $\boldsymbol{x}' \in D$, kde $\boldsymbol{x}' \neq \boldsymbol{x}$, t: $D \rightarrow D$ pro $\forall\ t \in S$. Postulujme, že pro každou transformaci existuje transformace k ní inverzní. Okolí $U(\boldsymbol{x})$ obsahuje obrazy $\boldsymbol{x}$ vytvořené transformacemi $t \in S$,

$$U(\boldsymbol{x}) = \{t\boldsymbol{x}; \forall t \in S\}. \tag{9.4}$$

Nyní už máme aparát k formulaci horolezeckého algoritmu. Pro náhodně vygenerovaný vektor $\boldsymbol{x}$ hledáme minimum funkce $f(\boldsymbol{x})$ v okolí $U(\boldsymbol{x})$,



Obrázek 9.5. (A) Znázornění okolí $U(\boldsymbol{x})$ momentálního řešení $\boldsymbol{x}$ a nejlepšího řešení z tohoto okolí $\boldsymbol{x}^*$. (B) Znázornění iteračního postupu horolezeckého algoritmu. (C) Hodnoty momentálního nejlepšího řešení v závislosti na čase (iteraci). Ke konci graf začne pravidelně oscilovat.



Obrázek 9.6. Znázornění zacyklení u horolezeckého algoritmu hledajícího minimum funkce. Protože bariéra je větší než délka kroku, algoritmus ji není schopen překonat a řešení znázorněné kuličkou se vrací do již předtím nalezeného "dolíku". Tyto poslední dva kroky se pak do nekonečna opakují, poněvadž algoritmus není schopen zakázat "zpětnou transformaci" do předchozího řešení.

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}' \in U(\boldsymbol{x})} f(\boldsymbol{x}'). \tag{9.5}$$

Získané řešení $\boldsymbol{x}^*$ je použito v následující iteraci algoritmu jako "střed" nového okolí $U(\boldsymbol{x}')$, a tento proces se opakuje předepsaný počet-krát. Pseudo-pascalovský algoritmus vypadá následovně:

“Horolezecký” algoritmus:

1 $\boldsymbol{x}$:=náhodně vygenerovaný vektor;
2 time:=0; f_{min}:=∞;
3 **WHILE** time<$time_{max}$ **DO**
4 **BEGIN** time:=time+1;
5 $f(\boldsymbol{x}^*) := \min_{\boldsymbol{x}' \in U(\boldsymbol{x})} f(\boldsymbol{x}')$
6 **IF** $f(\boldsymbol{x}^*) < f_{min}$ **THEN**
7 **BEGIN** $f_{min} := f(\boldsymbol{x}^*)$;
8 $\boldsymbol{x}_{min} := \boldsymbol{x}^*$;
9 **END**;
10 $\boldsymbol{x} := \boldsymbol{x}^*$;
11 **END**;

Proměnné f_{min} a $\boldsymbol{x}_{min}$ zaznamenávají nejlepší řešení získané v průběhu celého algoritmu opakovaného $time_{max}$-krát. Hlavní omezení tohoto jednoduchého heuristického algoritmu je problém cyklických řešení. Po konečném počtu iteračních kroků se algoritmus vrátí k řešení, které se již vyskytovalo jako lokální řešení v předcházejícím iteračním kroku, přičemž nejlepší zaznamenané řešení je obvykle vzdálené od globálního minima funkce v oblasti *D*.

Kromě “okolí” bitového řetězce získaného překlopením jednoho bitu se také někdy používá jiné operace, tzv. inverze. Tato operace je užitečná především u binární reprezentace reálných čísel, kdy např. čísla 00001111 a 00010000 sousedí v reálné reprezentaci (alespoň při daném rozlišení), ale na přeměnu jednoho v druhé by bylo potřeba pěti překlopení bitu. Proto se někdy zavádí operace, která od daného bitu překlopí všechny následující bity. Tak by se z prvního řetězce dal vyrobit druhý překlopením posledních pěti bitů: Inverze_od_čtvrtého_bitu(00001111) = 00010000.

Slovo “stochastický” v názvu stochastický “horolezecký” algoritmus znamená modifikaci “horolezeckého” algoritmu, při které se začíná několikrát z rozdílného výchozího řešení, a za výsledek se bere nejlepší řešení z několika průběhů. Je možná i modifikace algoritmu, při které se neprohledává celé okolí momentálního řešení, ale pouze jeho náhodně vybraná část.

9.4 Tabu search neboli "zakázané prohledávání"

Koncem osmdesátých let navrhl Prof. Fred Glover z University of Colorado, Boulder [8] nový přístup k řešení problému globálního minima, který nazval tabu search. V současnosti patří tato metoda mezi hlavní hity v oblasti operačního výzkumu a algoritmů na řešení kombinatorických úloh a hledání globálního optima. Její základní myšlenka je velmi jednoduchá. Vychází z horolezeckého algoritmu, kde se snaží odstranit problém zacyklení. Do horolezeckého algoritmu je zavedená tzv. krátkodobá paměť, která si pro určitý krátký interval předcházející historie algoritmu pamatuje inverzní transformace k lokálně optimálním transformacím řešení použitým k získání nových "středů" pro jednotlivé iterace. Tyto inverzní transformace jsou zakázány (tabu) při tvorbě nového okolí pro dané aktuální řešení. Tímto jednoduchým způsobem je možné podstatně omezit výskyt zacyklení při pádu do lokálního minima. Takto modifikovaný horolezecký algoritmus systematicky prohledává celou oblast, ve které hledáme globální minimum funkce.

Glover [8,9] navrhl další metody intenzifikace a diverzifikace zakázaného prohledávání, konkrétně přístup tzv. dlouhodobé paměti, ve kterém se pokutují (znevýhodňují) při ohodnocení funkce f ty transformace, které sice nepatří do krátkodobé paměti, ale často se vyskytovaly v předcházející historii algoritmu. Metoda je aktivně rozvíjena, její teoretické základy nejsou však zatím dost solidní, aby daly odpověď např. na otázku, za jakých podmínek metoda zkonverguje ke globálnímu optimu. V současné době jde spíše o soubor algoritmických triků a heuristik, které jsou však vysoce numericky efektivní.

Hlavní myšlenka heuristiky je algoritmicky realizována v zakázaném seznamu T (tabu list). Ten zastupuje krátkodobou paměť, která dočasně obsahuje inverzní transformace k transformacím použitým v předcházejících iteracích. Zakázaný seznam transformací $T \subseteq S$, maximální velikosti k, je sestrojen a obnovován v průběhu chodu celého algoritmu. Jestliže transformace t patří do zakázaného seznamu, $t \in T$, pak se nemůže používat v lokální minimalizaci v rámci okolí aktuálního řešení $\boldsymbol{x}$. Při inicializaci algoritmu je zakázaný seznam prázdný, po každé iteraci se do zakázaného seznamu přidá transformace inverzní k transformaci, která poskytla lokálně optimální současné řešení přeměnou řešení z předcházející iterace (např. se do zakázaného seznamu zapíše pořadové číslo bitu, který by se nadále neměl měnit — tabu list musí být kratší než počet možných transformací). Po k iteracích zakázaný seznam už obsahuje k transformací, a každé další dodání nové transformace je doprovázené vyloučením momentálně "nejstarší" transformace (dodané právě před k iteracemi). Říkáme, že zakázaný seznam se cyklicky obnovuje,

$$T := \begin{cases} T \cup \left\{ t^{*-1} \right\} & (\text{pro } |T| < k) \\ \left(T \cup \left\{ t^{*-1} \right\} \right) \setminus \hat{t} & (\text{pro } |T| = k) \end{cases} \tag{9.6}$$

kde t^* je transformace, která vytváří lokálně optimální řešení $\boldsymbol{x}^* = t^* \boldsymbol{x}$, a $\hat{t}$ je "nejstarší" transformace zavedená do zakázaného seznamu právě před k iteracemi. Numerické zkušenosti s algoritmem zakázaného prohledávání ukazují, že velikost k zakázaného seznamu je velmi důležitým parametrem ovlivňujícím možnost vymanit se při prohledávání oblasti D z lokálních minim. Jestliže je parametr k malý, pak se může vyskytnout zacyklení algoritmu, stejně jako u klasického horolezeckého algoritmu.

Zacyklení se sice neopakuje v sousedních dvou krocích, ale řešení se může opakovat po více krocích. V případě, že je parametr k velký, potom při prohledávání oblasti D s velkou pravděpodobností "přeskočíme" hluboká údolí funkce $f(\boldsymbol{x})$, t.j. vynecháváme nadějná lokální minima, která mohou být globálními minimy. Jednou z populárních úprav algoritmu je adaptace délky zakázaného seznamu podle dosud dosažených výsledků.

Zakázaný seznam se používá ke konstrukci modifikovaného okolí aktuálního řešení $\boldsymbol{x}$,

$$U_T(\boldsymbol{x}) = \{ \boldsymbol{x}'; \forall t \in S \backslash T: \boldsymbol{x}' = t\boldsymbol{x} \}. \tag{9.7}$$

Toto okolí obsahuje vektory $\boldsymbol{x}' \in D$, které jsou vytvořeny pomocí transformací z množiny S nepatřících do zakázaného seznamu T.

Lokální minimalizace se vykonává v modifikovaném okolí $U_T(\boldsymbol{x})$ s výjimkou tzv. aspiračního kritéria. Toto kritérium porušuje restrikci zakázaného seznamu tehdy, existuje-li taková transformace $t \in S$, že vektor $\boldsymbol{x}' = t\boldsymbol{x}$ poskytuje nižší funkční hodnotu, než má dočasně nejlepší řešení.

Pascalovský pseudokód metody zakázaného prohledávání je prezentován ve formě algoritmu na následující straně.

Algoritmus zakázaného prohledávání je podobný předchozímu horolezeckému algoritmu. Hlavní rozdíl spočívá v lokálním prohledávání kombinova-ném s aspiračním kritériem uvedeným na řádcích 5-9. V řádku 12 je obnovo-ván zakázaný seznam, viz (9.6).

Algoritmus zakázaného prohledávání byl zatím diskutován jen ve své základní podobě. Možnosti jeho dalších úprav jsou široce diskutovány v literatuře [9,10]. Přístup založený na koncepci dlouhodobé paměti patří mezi základní prostředky intenzifikace a diverzifikace algoritmu směrem k získání globálního minima. Využívá možnost odmítnutí (pokutování) transformací, které se v předcházejícím průběhu algoritmu vyskytly nejčastěji (dlouhodobá paměť). Hledání lokálního minima v modifikovaném okolí $U_T(\boldsymbol{x})$ je v tomto přístupu založeno nejen na změnách funkce $f(\boldsymbol{x})$, ale i na předcházející historii algoritmu. Jednoduchá realizace této všeobecné myšlenky je použití frekvencí transformací $\omega(t)$. Při inicializaci algoritmu jsou tyto frekvence nulové, potom v

Tabu search:

```
1    x:=náhodně vygenerovaný vektor;
2    time:=0; f_min:=∞; T:=∅;
3    WHILE time<time_max DO
4    BEGIN time:=time+1; f_loc-min:=∞;
5            FOR t∈S DO
6            BEGIN x′:=tx;
7                    IF f(x′) < f_loc-min AND (t∉T OR f(x′) < f_min) THEN
8                    BEGIN x*:= x′; t*:=t; f_loc-min:= f(x′); END;
9            END;
10           IF f_loc-min< f_min THEN BEGIN f_min:= f_loc-min ; x_min := x*; END;
11           x := x*;
12           IF |T|<k THEN T:=T∪{t*^-1} ELSE T:=(T∪{t*^-1})\{ t̂ };
13   END;
```

každém iteračním kroku s výslednou transformací t^* je odpovídající frekvence zvýšena o jednotku, $\omega(t^*) \leftarrow \omega(t^*)+1$. Po předepsaném počtu kroků (obvykle řádově větším než je velikost k zakázaného seznamu T) tyto frekvence určují, jak často byly jednotlivé transformace z S použité v lokální minimalizaci. Frekvence se používají jako pokutové funkce při hledání minima v okolí $U_T(\boldsymbol{x})$. Vektor $\boldsymbol{x}' = t\boldsymbol{x} \in U_T(\boldsymbol{x})$, kde $t \in S \backslash T$, je akceptován jako dočasně nejlepší řešení, je-li splněna následující podmínka

$$f(\boldsymbol{x}')+\alpha\,\omega(t) < f(\boldsymbol{x}^*) \tag{9.8}$$

kde α je empiricky určená malá konstanta. To znamená, že minimalizace popsaná v řádcích 5-9 ve výše uvedeném algoritmu je realizována pro funkci $f(\boldsymbol{x}')+\alpha\,\omega(t)$, ovšem jako lokálně nejlepší řešení v okolí $U_T(\boldsymbol{x})$ se zaznamenává jen funkční hodnota $f(\boldsymbol{x}')$. Nejčastěji používané transformace jsou penalizovány jako důsledek vysokých hodnot frekvencí. Přístup dlouhodobé paměti dává šanci i jiným transformacím než těm, které i když poskytují lokálně nižší funkční hodnotu $f(\boldsymbol{x}')$, jsou penalizovány v důsledku jejich frekventovaného výskytu v předcházející dlouhodobé historii algoritmu.

Technika zakázaného prohledávání může být použita jak v klasickém spojení s horolezeckým algoritmem, tak i v kombinaci s jinými algoritmy, např. se simulovaným žíháním nebo genetickými algoritmy. U těchto metod však není natolik efektivní, tyto metody neprohledávají celé okolí momentálního řešení a pravděpodobnost zapůsobení zakázaného seznamu je tedy dost malá.

9.5 Simulované žíhání (simulated annealing)

Počátkem 80-tých let Kirkpatrick, Gelatt a Vecchi [11] (Watson Research Center of the IBM, USA) a nezávisle Černý [12] (MFF UK v Bratislavě) dostali geniální nápad, že problém hledání globálního minima může být realizovaný podobným způsobem jako žíhání tuhého tělesa. Přístup simulovaného žíhání [13,14] je založen na "simulování" fyzikálních procesů probíhajících při odstraňování defektů krystalové mřížky. Krystal se zahřeje na určitou (vysokou) teplotu a potom se pomalu ochlazuje (žíhá). Defekty krystalové mřížky mají vysokou pravděpodobnost zániku. Ochlazování systému zabez-pečí, že pravděpodobnost vzniku nových defektů klesá na malou hodnotu.

V simulovaném žíhání je "krystal" reprezentován binárním řetězcem $\boldsymbol{x}$, tomuto řetězci můžeme přiřadit "energii krystalu" — funkční hodnotu $f(\boldsymbol{x})$. Z výše uvedených fyzikálních úvah vyplývá, že v procesu žíhání se minimalizuje energie krystalu. Tato skutečnost naznačuje, že v metodě simulovaného žíhání minimalizujeme funkci $f(\boldsymbol{x})$. Aktuální řetězec $\boldsymbol{x}$ je náhodně přeměněný na nový řetězec $\boldsymbol{x}'$. Tento proces by měl najít nový řetězec z okolí původního řetězce, ale vzhledem k tomu, že nyní nepotřebujeme prohledávat celé okolí, okolí může být zadefinováno mnohem šířeji a volněji, než tomu bylo třeba u horolezeckého algoritmu. Nový řetězec $\boldsymbol{x}'$ nahradí původní řetězec v následném procesu simulovaného žíhání s pravděpodobností (Metropolisův vzorec [15])

$$\Pr(\boldsymbol{x}' \to \boldsymbol{x}) = \left\{1, \exp\left(-(f(\boldsymbol{x}') - f(\boldsymbol{x}))\,/\,T\right)\right\} \tag{9.9}$$

kde parametr T je formální analogií teploty. Jestliže $f(\boldsymbol{x}') \le f(\boldsymbol{x})$, pravděpodob-nost akceptace je jednotková. V tomto případě je nový řetězec $\boldsymbol{x}'$ automaticky akceptován do dalšího procesu simulovaného žíhání. V případě, že $f(\boldsymbol{x}') > f(\boldsymbol{x})$, pravděpodobnost akceptování $\boldsymbol{x}'$ je menší než jednotková, ale i v tomto případě má nový řetězec šanci pokračovat v simulovaném žíhání. V pseudopascalov-ském kódu má algoritmus simulovaného žíhání následující jednoduchou formu uvedenou na následující straně.

Teplota T je ohraničená maximální a minimální hodnotou, $T_{min} \le T \le T_{max}$, snižování teploty je realizováno ve 14. řádku, kde α je kladné číslo menší než jedna, obvykle α=0,9. Celočíselné proměnné t a k jsou počitadla pro vnější resp. vnitřní WHILE-cyklus. Proměnná t zaznamenává celkový počet "pokusů" simulovaného žíhání pro danou teplotu T, zatímco proměnná k zaznamenává počet úspěšných "pokusů", které byly akceptovány Metropolisovým vzorcem (9.9). Pro volbu konstant t_{max} a k_{max} neexistuje všeobecný předpis. Obvykle je hodnota k_{max} od několika set do několika tisíc a $t_{max} = 10\ k_{max}$. Reálná proměnná *random* (9. řádek) je náhodně generované číslo z intervalu $\langle 0,1)$. Řetězec $\boldsymbol{x}^*$ zaznamenává nejlepší řešení v průběhu celého simulovaného žíhání. Ve všeobecnosti binární řetězec $\boldsymbol{x}$ po skončení simulovaného žíhání nemusí být rovný řetězci $\boldsymbol{x}^*$.

Simulované žíhání:

```
1     x:=náhodně generovaný binární řetězec;
2     T:=T_max; x*:=x; k:=1;
3     WHILE (T>T_min ) and (k>0) DO
4     BEGIN t:=0; k:=0;
5           WHILE (t<t_max) and (k<k_max) DO
6           BEGIN t:=t+1;
7                 x′:=transformace(x);
8                 IF f(x′) ≤ f(x) THEN Pr:=1 ELSE
                  Pr:=exp(–(f(x′) – f(x))/T);
9                 IF random<Pr THEN
10                BEGIN x:=x′; k:=k+1;
11                      IF f(x) < f(x*) THEN x*:=x;
12                END;
13          END;
14    T:=α·T;
15    END;
```

V literatuře [13,14] existuje podrobná teorie simulovaného žíhání. Byly dokázány existenční teorémy, za jakých podmínek simulované žíhání poskytuje globální minimum funkce $f(\boldsymbol{x})$ v definičním oboru $\boldsymbol{x}$. V pracích [16,17] je navržené rozšíření simulovaného žíhání směrem ke genetickému algoritmu. Místo jednoho binárního řetězce se současně optimalizuje simulovaným žíháním malá populace binárních řetězců, které si s malou pravděpodobností vymění informaci operací totožnou s křížením z genetického algoritmu.

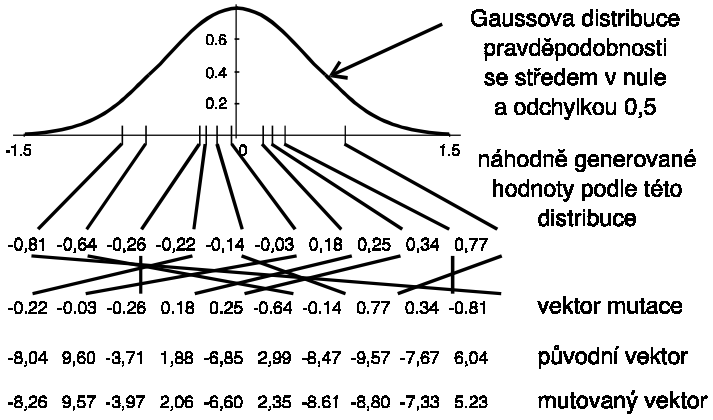
9.6 Evoluční strategie

Evoluční strategie patří historicky mezi první úspěšné stochastické algoritmy. Byla navržena už počátkem 60-tých let Rechenbergem a Schwefelem [18-20, 4]. Vychází ze všeobecných představ přirozeného výběru, ovšem o mnoho vágnějších než například u genetického algoritmu. Navíc, na rozdíl od předcházejících stochastických metod, evoluční strategie není založena na binární reprezentaci proměnných, manipuluje přímo s "reálnou" reprezentací proměnných. Pro neuronové sítě může být tedy použita pouze pro optimalizaci vah nebo jiných proměnných, nikoli pro optimalizaci topologie sítě.

Základem evoluční strategie je následující předpis, který "mutuje" aktuální řešení $\boldsymbol{x}$ na nové řešení $\boldsymbol{x}'$ (viz obr. 9.7),

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{r}(0,\sigma)\,, \tag{9.10}$$

kde $\boldsymbol{r}(0,\sigma)$ je vektor nezávislých náhodných čísel s nulovou střední hodnotou a směrodatnou odchylkou σ.



Obrázek 9.7. Příklad mutace vektoru s reálnými proměnnými za použití Gaussovy distribuce pravděpodobnosti pro určení malé náhodné změny (ne vždy je nutné mutovat všechny hodnoty vektoru, stačí si náhodně vybrat, který prvek se bude mutovat).

Problém akceptace nového řešení $\boldsymbol{x}'$ je striktně deterministický, řešení $\boldsymbol{x}'$ je akceptované (úspěšné), jestliže $f(\boldsymbol{x}') < f(\boldsymbol{x})$. Směrodatná odchylka σ se v průběhu evoluční strategie mění podle pravidla 1/5 úspěšnosti. Nechť $\varphi(k)$ je koeficient úspěšnosti definovaný jako poměr počtu úspěšných mutací v průběhu posledních k iterací k počtu k iterací, ze kterých byla úspěšnost měřena, potom

$$\sigma' = \begin{cases} c_d.\sigma & (\varphi(k) < 1/5) \\ c_i.\sigma & (\varphi(k) > 1/5) \\ \sigma & (\varphi(k) = 1/5) \end{cases} \tag{9.11}$$

kde $c_i > 1$ a $c_d < 1$ řídí zvětšování resp. zmenšování směrodatné odchylky, v literatuře [18] jsou tyto koeficienty specifikované $c_d = 0{,}82$ a $c_i = 1/c_d = 1{,}22$. Algoritmus evoluční strategie v pseudopascalu má tento tvar:

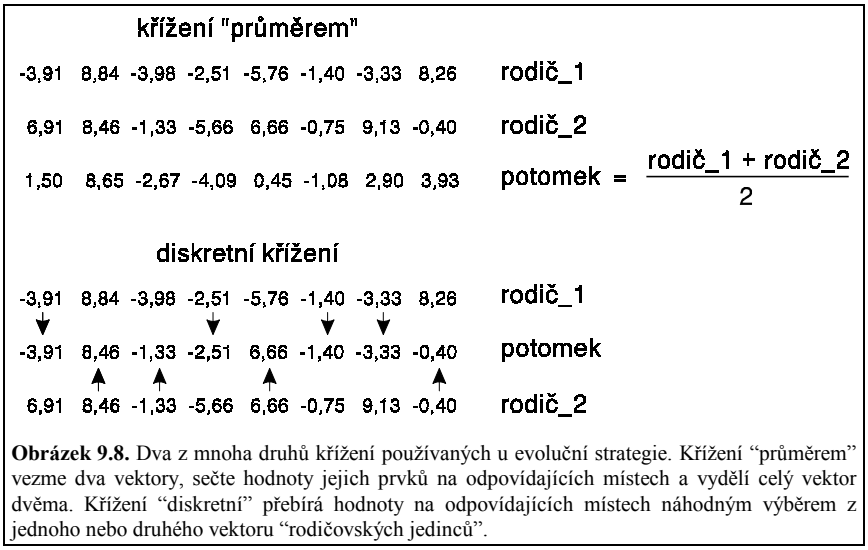
Evoluční strategie:

```
1    x:=náhodně generovaný vektor reálných proměnných;
2    t:=0; σ:=σini; x*:=x;
3    WHILE t<tmax DO
4    BEGIN i:=0; k:=0;
5          WHILE i<imax DO
6          BEGIN i:=i+1; x′:=x+r(0,σ);
7                IF f(x′) < f(x) THEN
8                BEGIN k:=k+1; x:=x′;
9                IF f(x) < f(x*) THEN x*:=x;
10               END;
11         END;
12         IF k/imax<0,2 THEN σ:=cd·σ ELSE IF k/imax>0,2 THEN σ:=ci·σ;
13   END;
```

Proměnná t je počitadlo epoch evoluční strategie. Algoritmus obsahuje dva WHILE-cykly, vnější a vnitřní. Ve vnitřním cyklu se pro dané σ opakuje elementární krok evoluční strategie i_{max}-krát, přičemž proměnná k zaznamenává úspěšnost mutací v tomto vnitřním cyklu. Vnější cyklus, s počitadlem t, aplikuje pro různé hodnoty σ evoluční strategii t_{max}-krát. Směrodatná odchylka σ je v 2. řádku inicializována hodnotou σ_{ini}. V 6. řádku se vykonává modifikace řešení $\boldsymbol{x}$ pomocí generátoru náhodných čísel s nulovou střední hodnotou a se směrodatnou odchylkou σ. Volba základních parametrů evoluční strategie (t_{max}, σ_{ini}, i_{max} a k_{max}) si vyžaduje určité experi-mentování, pomocí kterého tyto konstanty nastavíme. Obvykle je σ_{ini} blízké jedničce, a konstanta i_{max} se rovná řádově tisícům.

Podobně, jako pro simulované žíhání, bylo dokázáno i pro evoluční strategii [18], že potenciálně poskytuje globální extrém optimalizované funkce $f(\boldsymbol{x})$. Schwefelem se spolupracovníky [18-20] byly navrhnuty další sofistikovanější verze evoluční strategie, takže v současnosti je možné už mluvit o celé třídě evolučních strategií. Pracuje se zde poté s celým souborem vektorů $\boldsymbol{x}$, a kromě mutace je zde používáno křížení, t.j. částečná výměna informací mezi vektory reálných čísel (viz obr. 9.8), na rozdíl od křížení bitových řetězců používaného u dále probíraných genetických algoritmů. Nejlepší jedinci se poté vyberou z takto navrhnutých vektorů a z původních "rodičovských" vektorů.

Kromě vlastní hodnoty proměnné ve vektoru může být každá proměnná charakterizována i vektorem "strategických" proměnných. Totiž, některé proměnné jistě mají větší vliv na hodnotu funkce než jiné proměnné, a proto by i jejich změny měly mít jiné měřítko. To může být zabezpečeno tím, že každá proměnná má svůj vlastní rozptyl. Pro dvě proměnné potom vrstevnice pravděpodobnosti umístění mutovaného vektoru nepředstavují kružnici, ale elipsu. Tato elipsa však je orientována ve směru souřadnicových os. Můžeme si však představit, že optimum není umístěno ve směru delší osy elipsy, ale někde našikmo. Ideální by pak bylo, kdybychom mohli tuto elipsu vrstevnic pravděpodobností umístění mutovaného vektoru natočit tak, aby hlavní osa elipsy směřovala k optimu. Tak by mutovaný vektor měl největší šanci co nejvíce se přiblížit k optimu. Toto natočení lze zajistit kovariancemi. Vektor "strategických" proměnných tedy může pro n-rozměrný vektor $\boldsymbol{x}$ zahrnovat n rozptylů $c_{ii} = \sigma_i^2$ stejně jako $n(n-1)/2$ kovariancí c_{ij} zobecněné n-rozměrné normální distribuce s hustotou pravděpodobnosti vektoru mutací z



Obrázek 9.8. Dva z mnoha druhů křížení používaných u evoluční strategie. Křížení "průměrem" vezme dva vektory, sečte hodnoty jejich prvků na odpovídajících místech a vydělí celý vektor dvěma. Křížení "diskretní" přebírá hodnoty na odpovídajících místech náhodným výběrem z jednoho nebo druhého vektoru "rodičovských jedinců".

$$p(z) = \sqrt{\frac{\det A}{(2\pi)^n}} \exp\left(-\frac{1}{2} z^T A z\right) \tag{9.12}$$

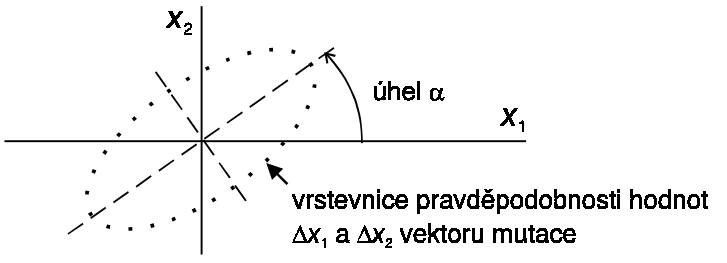
Pro zajištění kladnosti a konečnosti kovarianční matice A^{-1} algoritmus používá odpovídající rotační úhly α_j ($0 \le \alpha_j \le \pi$) místo koeficientů c_{ij}. Mutace jsou potom prováděny následovně

$$\begin{aligned} \sigma_i' &= \sigma_i \exp(\tau_0 \Delta\sigma_0) \exp(\tau \Delta\sigma_i) \\ \alpha_j' &= \alpha_j + \beta \Delta\alpha_j \\ x_i' &= x_i + z_i(\sigma', \alpha') \end{aligned} \tag{9.13}$$

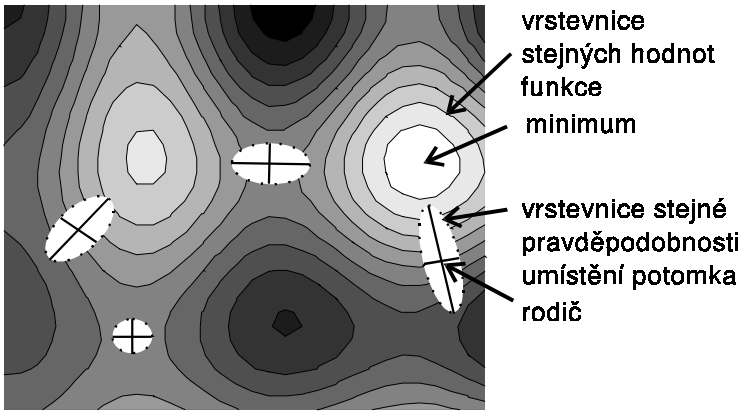
Tímto způsobem jsou mutace proměnných korelovány pomocí hodnot vektoru α, a σ poskytuje "měřítko" lineární metriky. Mutace $\Delta\sigma$ a $\Delta\alpha$ mají opět normální distribuci se středem v nule a směrodatnou odchylkou rovnou jedné, a konstanty $\tau_0 \propto 1/\sqrt{2\sqrt{n}}$ a $\tau \propto 1/\sqrt{2n}$ a $\beta \approx 0{,}0873$ ($=5^o$). Hodnota $\Delta\sigma_0$ reduko-vaná násobením τ_0 je globálním parametrem, zatímco $\Delta\sigma_i$ je individuálním parametrem generovaným zvlášť pro každou proměnnou z vektoru $\boldsymbol{x}$, dovolujíc tak individuální změny "průměrných" změn σ_i každé proměnné x_i vektoru $\boldsymbol{x}$. Bohužel, generovat náhodný vektor mutací s distribucí pravděpodobnosti $p(z)$ uvedenou v (9.12) nemusí být triviálním problémem. V praxi se tento problém nahrazuje postupnou rotací. Řekněme, že máme dvě proměnné x_1 a x_2 se směrodatnými odchylkami σ_1 a σ_2, takže vrstevnice stejných hodnot pravděpodobnosti by tvořily elipsu s osami rovnoběžnými s hlavními osami. Korelační koeficient c_{12} odpovídá úhlu α, o který se tato elipsa pravděpodobnosti pootočí (viz obr. 9.9 a obr. 9.10).

Takže jsou-li původní odchylky proměnných x_1 a x_2 se směrodatnými odchylkami σ_1 a σ_2 rovny Δx_1 a Δx_2, pak odchylky upravené pomocí korelačního koeficientu mají tvar $\Delta x_1' = \Delta x_1 \cos\alpha - \Delta x_2 \sin\alpha$, $\Delta x_2' = \Delta x_1 \sin\alpha + \Delta x_2 \cos\alpha$.

Pro tři proměnné by musely být provedeny tři následné rotace, v rovině (Δx_1, Δx_2) o úhel α_1 s výsledkem $\Delta x_1'$ a $\Delta x_2'$, v rovině ($\Delta x_1'$, Δx_3) o úhel α_2 s výsledkem $\Delta x_1''$ a $\Delta x_3'$, a v rovině ($\Delta x_2'$, $\Delta x_3'$) o úhel α_1 s výsledkem $\Delta x_2''$, $\Delta x_3''$. Výsledné změny proměnných by tedy byly $\Delta x_1''$, $\Delta x_2''$ a $\Delta x_3''$.



Obrázek 9.9. Pootočení elipsy stejných hodnot pravděpodobnosti vygenerování změn Δx_1 a Δx_2.



Obrázek 9.10. Zobrazení vrstevnic hodnot funkce dvou proměnných s optimy označenými světlou barvou. Bílé tečkované ovály označují místa stejné pravděpodobnosti umístění potomka vzniklého mutací z rodiče umístěného ve středu oválu označeného křížením os. Delší osa oválů je nasměrována směrem k optimu, tak jak se to "jedinec" naučil v průběhu optimalizace, kdy každá z proměnných si uschovává vlastní hodnotu směrodatné odchylky, a ještě se uschovává výhodný směr korelovaných odchylek.

9.7 Genetické algoritmy

Genetický algoritmus, vycházející ze všeobecných představ Darwinovy teorie přirozeného výběru, byl původně navržen Hollandem [21] jako učící se algoritmus schopný adaptivně reagovat na měnící se prostředí. Ovšem, hned po jeho vzniku se ukázalo [1,22-23], že je vhodnou metodou na hledání globálního minima úloh, které doposud nebyly řešitelné nebo byly řešitelné jen velmi obtížně. Kromě nastavování parametrů pro neuronové sítě mezi úspěšné aplikace genetických algoritmů patří rozvrhy práce pro stroje v továrnách, teorie her v managementu, všechny možné obtížně řešitelné optimalizační problémy multimodálních funkcí ve vědeckotechnických výpočtech — třeba hledání prostorového uspořádání molekul, řízení robotů, rozpoznávací systémy. V informatice jsou genetické algoritmy zvláště populární pro výhodnou možnost implementace na víceprocesorových počítačích, kde každý procesor "obhospodařuje" jeden chromozóm. V nově se rodící informatické disciplíně nazvané Umělý Život, tvořené ponejvíce simulacemi vývoje hnaného Darwinovským požadavkem přežití nejschopnějších, jsou genetické algoritmy integrální součástí většiny aplikací. Další možností využití genetických algoritmů je strojové učení s klasifikačními systémy [23] a umělá inteligence, kde ale za klasifikační postup je možné z nejobecnějšího hlediska považovat třeba jakýkoli počítačový program. V poslední době je populární také genetické programování, které ve své nejjednodušší podobě nehledá pouze ideální nastavení parametrů regresní funkce, ale i funkci samotnou (většinou složenou z několika základních matematických funkcí). Tomuto přístupu se zde však nebudeme věnovat.

Na rozdíl od neuronových sítí, které se snaží "polapit" efektivitu jednotlivého "mozku" (počet neuronů v umělých neuronových sítích je však z technických důvodů o mnoho řádů menší, než je v jakémkoli lidském nebo zvířecím mozku), genetické algoritmy svou stavbu spojují s vývojem celého společenství. Snaží se využít genetických představ o hnacích silách evoluce živé hmoty. Ačkoliv genetické algoritmy nemají již prakticky s biologií nic společného, udržely si biologickou terminologii. Omezíme-li použití genetických algoritmů na optimalizaci funkce, evolucí jsou míněny postupné změny proměnných vedoucí k nalezení extrému funkce. Soubor proměnných vstupních veličin funkce tvoří jedince. Není zde však tak důležitý jedinec, jako postupný vývoj, kooperace a fungování populace — souboru jedinců. Neúspěšní jedinci vymírají, úspěšní přežívají a množí se. Hybnou silou změn jsou mutace a křížení (výměna "genetické" informace mezi jedinci). Každý jedinec je v algoritmu reprezentován svým lineárně uspořádaným informačním obsahem (formálně nazývaným chromozóm).

Nechť P je populace obsahující M chromozómů. Pod chromozómem budeme rozumět binární řetězec délky, která je konstantní pro všechny chromozómy z populace P. Chceme-li optimalizovat funkci N proměnných, chromozóm odpovídá binární reprezentaci za sebou seřazených binárních reprezentací jednotlivých proměnných. Každý chromozóm $\boldsymbol{x} \in P$ je ohodnocený silou (fitness) $s(\boldsymbol{x})$ tak, že chromozómu $\boldsymbol{x}$ s malou (velkou) funkční hodnotou $f(\boldsymbol{x})$ je přiřazená velká (malá) síla $s(\boldsymbol{x})$. Mezi chromozómy z populace P probíhá proces reprodukce, jehož výsledkem je nová populace P' obsahující stejný počet chromozómů jako původní populace P. Reprodukce se skládá z následujících částí:

(1) Výběr chromozómů. Do procesu reprodukce vstupují dvojice chromo-zómů $\boldsymbol{x},\boldsymbol{x}' \in P$, které jsou náhodně vybrané, přičemž pravděpodobnost výběru je úměrná silám $s(\boldsymbol{x})$ a $s(\boldsymbol{x}')$.

(2) Křížení chromozómů. Vybrané chromozómy $\boldsymbol{x}$ a $\boldsymbol{x}'$ si vymění náhodně vybrané části chromozómů. Nechť $\boldsymbol{x}=(a_1 \ldots a_i \ldots a_N)$ a $\boldsymbol{x}'=(b_1 \ldots b_i \ldots b_N)$ jsou dva chromozómy a index $1 \leq i < N$ je náhodně zvolený. Potom jejich křížením dostaneme dva nové chromozómy $\hat{\boldsymbol{x}}=(a_1 \ldots a_{i-1} b_i \ldots b_N)$ a $\hat{\boldsymbol{x}}'=(b_1 \ldots b_{i-1}\, a_i \ldots a_N)$. To znamená, že chromozómy $\hat{\boldsymbol{x}}$ a $\hat{\boldsymbol{x}}'$ vznikly z původních chromozómů tak, že si vyměnily bitové podřetězce za i-tou komponentou.

(3) Mutace chromozómů. Chromozómy $\hat{\boldsymbol{x}}$ a $\hat{\boldsymbol{x}}'$ se podrobí mutaci, kde se náhodně vybrané komponenty binárních řetězců změní na jejich komplementy, t.j. $0 \rightarrow 1$ a $1 \rightarrow 0$. Pro lepší pochopení tohoto pojmu uvedeme jednoduchý příklad. Nechť (00**1**100**0**1) je bitový řetězec – chromozóm délky 8, v procesu mutací v náhodně vybraných polohách 3 a 7 se mění komponenty, dostaneme nový chromozóm (00**0**100**1**1).

Podstatným rysem genetického algoritmu je jeho úplná stochastičnost, v každém kroku jsou operace důsledně vykonávány náhodně. Genetický algoritmus v pseudopascalu je uveden na následující straně.

Proměnná t je diskrétní čas (počitadlo epoch), genetický algoritmus je ukončený, když $t=t_{\max}$, kde $t_{\max}$ je předepsaný počet epoch genetického algoritmu. P_t v algoritmu označuje populaci chromozómů v čase t, cyklus se opakuje $t_{\max}$-krát, Q je subpopulace chromozómů — potomků, které byly vytvořeny křížením rodičovských chromozómů z předcházející populace P_{t-1} a následnými mutacemi. Rodičovské chromozómy se vybírají "kvazináhodně", pravděpodobnost výběru je úměrná jejich síle. Symbol R označuje subpopulaci náhodně vybraných chromozómů s nejmenší silou. Nová populace P_t je vytvořena z populace P_{t-1} tak, že nové chromozómy vytěsní část původních chromozómů (v množinovém formalismu vyjádřené příkazem $P_t:=(P_{t-1} \setminus R) \cup Q$;). Počty chromozómů subpopulací Q a R jsou ohraničeny podmínkami $|Q|<<|P_t|$ a $|Q|=|R|$, t.j. chromozómů — potomků je ve většině aplikací podstatně méně než chromozómů v celé populaci a z populace je vytěsněno právě tolik chromozómů, jako je chromozómů — potomků.

Genetický algoritmus:

```
1    P0:={náhodně vygenerovaná populace chromozómů}; t:=0;
2    Ohodnoť každý chromozóm z populace P0 silou;
3    WHILE t<tmax DO
4    BEGIN t:=t+1;
5    Q:={kvazináhodně vybrané dvojice chromozómů z Pt-1 s největší silou pomocí
     rulety}
6    Q:=Operace_křižení(Q);
7    Q:=Mutace_jednotlivých_chromozómů(Q);
8    Ohodnoť každý chromozóm z Q silou.
9    R:={kvazináhodně vybrané chromozómy z Pt-1 s nejmenší silou};
10   Pt:=(Pt-1 \ R)∪Q;
11   END;
```

Kritickým místem algoritmu je tvorba nových chromozómů v 5-7 řádku. Jak bylo již výše uvedeno, proces reprodukce obsahuje výběr chromozómů, jejich křížení a mutaci. V průběhu celého výpočtu v každé epoše zaznamenáváme minimální hodnotu optimalizované funkce, po jeho ukončení tato hodnota reprezentuje výslednou minimální hodnotu funkce

$f(\boldsymbol{x})$ v oblasti $\boldsymbol{x}=\langle a,b\rangle^N$.

Pokud na rozdíl od dále uvedeného příkladu optimalizujeme pouze váhy neuronové sítě, pak nemusíme nic trénovat a tréninkovou množinu používáme přímo k ohodnocení sítě. Testovací množinu použijeme až k otestování výsledné sítě po skončení genetického algoritmu.

Síla chromozómů je určená podle následujícího postupu [23]: Vypočítáme funkční hodnoty funkce $f(\boldsymbol{x})$, které uspořádáme podle rostoucích funkčních hodnot, t.j. první (poslední) chromozóm má nejmenší (největší) funkční hodnotu. Chromozómu x_i z populace přiřadíme sílu podle vzorce

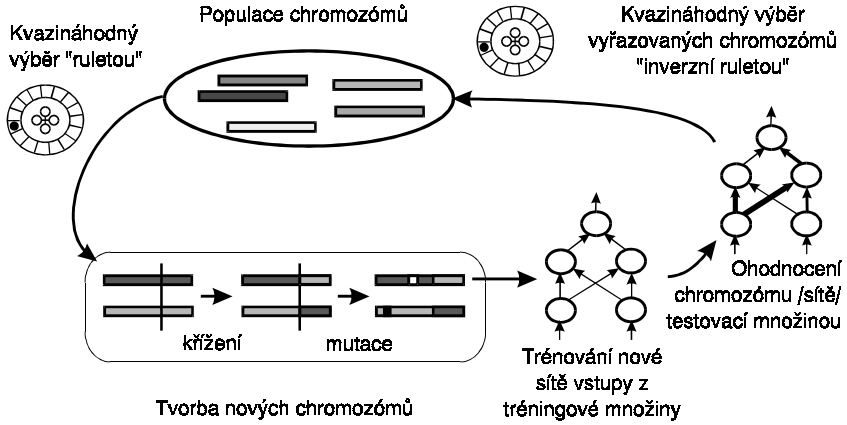
$$s(x_i) = \frac{1}{1-M}((1-\varepsilon)i + \varepsilon - M) \tag{9.14}$$

kde ε je malé kladné číslo, zvolili jsme $\varepsilon = 0{,}05$ a M je počet chromozómů.

Náhodnost výběru chromozómů do procesu reprodukce je realizována tak, aby byla úměrná jejich síle, což se uskutečňuje pomocí tzv. rulety [23]. Předpokládejme, že jsme rozdělili jednotkovou kružnici na M oblouků s délkami úměrnými velikostem sil. Náhodně zvolíme číslo $r \in \langle 0,1)$, potom toto číslo leží na některém oblouku jednotkové kružnice, jeho index odpovídá chromozómu, který je vybrán do procesu reprodukce. Tento postup připomíná ruletu, kulička se zastaví v oblouku (poloze) s pravděpodobností úměrnou délce oblouku — síle chromozómu. Ruleta nám s největší pravděpodobností vybírá ty chromozómy, které mají největší sílu. Ovšem i chromozómy s malou silou mají určitou malou šanci být vybrány do reprodukce. Tímto jednoduchým způsobem se realizuje základní atribut přirozeného výběru, a to, že do procesu reprodukce vstupují jedinci — chromozómy náhodně s pravděpodobností úměrnou jejich síle (viz obr. 9.11).

Nechť $\boldsymbol{x}$ a $\boldsymbol{x}'$ je dvojice chromozómů z populace, která byla vybrána ruletou. První krok reprodukce je křížení těchto chromozómů. Pokud se v populaci nahrazují jen některé chromozómy, a zbylé přežívají, křížení se provádí automaticky u všech vybraných chromozómů (jinak je možné přežívání některých chromozómů do další generace nahradit např. tím, že křížení se neprovádí u všech vybraných chromozómů, některé náhodně vybrané dvojice se jen kopírují). Mutace výsledných chromozómů z křížení má také stochastický charakter. Postupně se realizuje od první komponenty chromozómu (binárního řetězce obsahujícího 0 a 1) s pravděpodobností P_{mut}. Jestliže náhodné číslo $r \in \langle 0,1)$ vyhovuje podmínce $r \le P_{mut}$, potom danou komponentu zaměníme za její komplement, t.j. $0 \to 1$ nebo $1 \to 0$. Ukazuje se, že pravděpodobnost mutací musí být poměrně malá (i v přírodě jsou mutace velmi vzácné), např. $P_{mut} = 0{,}0001$ (t.j. v binárním řetězci se změní jen 0,01% komponent).

Mutace je v obecnosti malá náhodná změna jedné či několika proměnných (prvků chromozómu), která ovlivní řešení, ať už kladně nebo záporně. Může přitom jít i o reprezentaci dat reálnými čísly. Mutace je nutná k tomu, aby se zamezilo přílišné specializaci (t.j. zapadnutí celé populace řešení do jednoho suboptimálního minima), aby vždy byla možnost vytvoření zásadně nových chromozómů odpovídajících lepšímu řešení. Mutace přinášejí do chromozómů novou informaci. Kdyby však existovaly pouze mutace, genetický proces by se nelišil od metody náhodného prohledávání. Efektivita je zajištěna rekombinací neboli křížením, což je smixování dvou rodičovských chromozómů (souborů proměnných funkce) tak, aby vytvořily jiné dva chromozómy vzájemnou výměnou



Obrázek 9.11. Návrh topologie neuronové sítě pomocí genetického algoritmu. Každý z chromozómů je převeden na neuronovou síť, ta je inicializována náhodnými vahami, které jsou optimalizovány klasickým učícím algoritmem pomocí tréninkové množiny. Síť je poté testována testovací množinou a na základě výsledné chyby je nepřímo úměrně stanovena síla chromozómu. Tento chromozóm pak vytěsní kvazináhodně vybraný slabší chromozóm z populace. Výběr nových chromozómů ke křížení a mutaci se také děje pomocí "rulety".

některých hodnot proměnných. Reprodukcí dvou silných jedinců (chromozómů odpovídajících lepším řešením) dostaneme s vysokou pravděpodobností i silné potomky. Populace chromozómů je sada souborů hodnot proměnných, kde každý soubor může dávat jinou funkční hodnotu. Jednotlivé chromozómy bojují o přežití, t.j. do následující populace (nové generace) jsou vybírány s větší pravděpodobností soubory hodnot proměnných odpovídající lepšímu řešení (extrému funkce).

Otázka je, proč vlastně genetické algoritmy fungují. Příčinu je třeba hledat především ve výměně informací mezi chromozómy. Můžeme si představit, že některé pozice ovlivňují řešení více než jiné, a že kombinace těchto pozic může působit nelineárně, t.j. výsledek není součtem vlivu izolovaných změn, ale spíše jejich násobkem. Důvod, proč v tvorbě takto výhodně součinných pozic funguje tak dobře křížení, hledáme pomocí tzv. schémat. Ve schématu si v bitovém řetězci nahradíme ty hodnoty, na kterých hodnota funkce tolik nezáleží, hvězdičkami. U hvězdiček je jedno, jestli bitový řetězec nabývá hodnot 0 nebo 1. Genetický algoritmus se snaží prohledávat mnoho oblastí fázového prostoru zároveň. Nechť dva podřetězce *R* a *S*, které se vyskytují v dvou různých nepřekrývajících se částech řetězce, podstatně zvyšují sílu chromozómu. Jestli pravděpodobnost výskytu každého z nich je 10^{-6}, potom se pravděpodobnost jejich současného výskytu rovná součinu $10^{-6} \cdot 10^{-6} = 10^{-12}$. Současný výskyt *R* a *S* v chromozómu jen v důsledku mutací je tedy vysoce nepravděpodobný. Avšak vyskytují-li se chromozómy s izolovanými podřetězci *R* a *S* už v populaci, křížení sloučí tyto podřetězce s velkou pravděpodobností do jednoho chromozómu.

Důvodem výrazně vyšší efektivity genetických algoritmů oproti náhodnému výběru je fakt, že chromozóm lze považovat za umístěný ve více schématech. Máme-li chromozóm 11010, je tento jediný chromozóm členem schématu 11****, ale také třeba schématu **0*0

a mnoha dalších. Relativně malý počet chromozómů tak mapuje velké množství schémat. Tento implicitní paralelismus je patrně jedním z klíčových faktorů úspěchu genetických algoritmů. Křížení tento efekt implicitního paralelismu poněkud komplikuje, poněvadž sice schémata spojuje, ale také často rozbíjí, tím pravděpodobněji, čím více jsou schémata rozložena po délce chromozómu. Tím, že dochází k rozbití schémat, dochází ale také k vytvoření nových schémat a výsledné chromozómy mohou patřit do oblasti, do které nepatřil ani jeden z rodičů. Vzhledem k tomu, že méně často dochází k rozbití schémat nul a jedniček umístěných blízko sebe, genetické algoritmy zvlášť dobře prohledávají všechny takto definované oblasti. Dochází tak k zvýšené produkci nelineárně podporovaných oblastí. To je vyjádřené Hollandovou větou z r. 1975, která tvoří teoretický základ genetických algoritmů [21]:

Věta. Nechť r je střední hodnota síly všech chromozómů v populaci, které obsahují schéma, n je počet těchto chromozómů a konečně nechť a je střední hodnota síly všech chromozómů v populaci. Potom očekávaný počet chromozómů obsahujících schéma v následující generaci je $n \cdot r/a - z$ (kde z je počet zániků schématu v důsledku křížení a mutací).

Tato věta říká, že efektivní schéma se vyskytuje v následující generaci s rostoucí frekvencí, a naopak, neefektivní schéma se vyskytuje s klesající frekvencí. Efektivita schématu závisí na poměru r/a, jestliže platí $r/a>1$ (t.j. $a<r$, čili střední síla všech chromozómů populace je menší než střední síla chromozómů obsahujících schéma), počet chromozómů obsahujících schéma roste. V opačném případě, když $r/a<1$, potom počet těchto chromozómů klesá. Tyto jednoduché úvahy jsou založeny na předpokladu, že počet zániků schémat v chromozómech v důsledků křížení nebo mutací je podstatně menší než počet jeho opakovaných vzniků v procesu reprodukce.

Chromozómy se postupně shromažďují v oblastech odpovídajících lepším řešením. To odpovídá schopnosti kombinovat v chromozómech potomků části řešení nacházející se v rodičovských chromozómech. Samozřejmě, jako se kombinují výhodné části řešení, tak se také mohou zkombinovat nevýhodné části řešení, ale takový potomek pravděpodobně vymře. Mutace, jako náhodná změna chromozómu, tvoří jen neporovnatelně malou část změn v porovnání s křížením. Obyčejně se udává, že vhodná průměrná frekvence mutací je jedno náhodné prohození nuly a jedničky na deset tisíc bitů. Mutace neurychluje nalezení řešení, ale spíše zajišťuje možnost další evoluce v případě, že se všechny chromozómy v průběhu generací následkem křížení nahrnou do jedné oblasti.

Částečnou modifikaci křížení nabízí taková výměna informací, kdy se chromozómy "nepřeříznou" na dva díly, ale na tři, a prostřední část se pak vymění. Omezení průzkumu schémat na víceméně souvislé části chromozómů se snaží odstranit operace tzv. inverze [21] (pozor, je rozdílná od inverze bitů uvedené v podkapitole 9.3). Ta může přeskupit chromozómy tak, že vstupy umístěné v původním chromozómu daleko od sebe budou v novém chromozómu u sebe. To odpovídá předefinování schémat, které jsou víceméně souvislé a ve kterých je proto průzkum intenzívnější, poněvadž nejsou tak často rozbíjeny křížením. Avšak tento přístup se nejeví příliš úspěšný, poněvadž je třeba si pamatovat původní pozice před přeskupením chromozómu. Není předem jasné, zda nové souvislé

oblasti budou úspěšnější, a provádí-li se inverze příliš často, je pak výsledkem prohledávání zase "širší" ale méně "hluboké" a genetický algoritmus špatně konverguje.

Proč při definici genetických algoritmů stále zdůrazňujeme náhodnost výběru chromozómů? Kdybychom do reprodukčního procesu vybírali jen chromozómy s největší silou, potom bychom velmi pravděpodobně podstatně ohraničili doménu, na které hledáme optimální výsledek. Genetický algoritmus by se stal velmi "oportunistickým", za dočasný lepší výsledek bychom dostali horší konečný výsledek. Chromozóm s menší silou může stále ještě obsahovat důležitou informaci využitelnou v budoucí evoluci populace. Všechny "částečně urychlující" heuristiky jsou nejen nedostatečné, ale v konečném důsledku i zavádějící, proto se je ani nepokoušíme do genetického algoritmu zabudovat.

Nevýhodou genetických algoritmů při aplikaci na topologii neuronových sítí jsou problémy se smysluplnou výměnou informací při křížení. Konkrétně jde o váhy v neuronové síti. Řekněme, že máme čtyři vstupní a čtyři skryté neurony. Podařilo se nám vygenerovat síť, ve které jsou spojeny vstupní neurony nenulovými vahami pouze s prvními dvěma skrytými neurony, t.j. druhé dva skryté neurony můžeme zanedbat. Vzhledem k tomu, že síť je v podstatě symetrická, může existovat stejně dobrá topologie zanedbávající prvé dva skryté neurony, a používající jen druhé dva skryté neurony. Při křížení pak může nastat situace, že první potomek bude používat všechny čtyři skryté neurony, a druhý potomek bude mít všechny váhy nulové. Existují různé způsoby, jak předcházet tomuto problému křížením jen víceméně podobných jedinců – sítí, ale žádný z nich se nedá považovat za uspokojivý. Pro n skrytých neuronů totiž existuje $n!$ permutací jejich postavení a tedy $n!$ ekvivalentních sítí. Kromě tohoto druhu symetrie existuje i symetrie u vah skrytých neuronů. Je-li přechodová funkce lichá (což většinou je), pak můžeme nahradit všechna znaménka vstupních a výstupních vah skrytého neuronu opačnými znaménky, a výstupy sítě se nezmění. Pro n skrytých neuronů tak máme 2^n strukturně odlišných, ale funkčně identických sítí generovaných takovýmto přehozením znamének. Při křížení těchto sítí pak dochází k nelogičnostem, protože se může lehce stát, že polovinu vah neuronu vezmeme z jedné sítě, polovinu z druhé sítě (kde byly opačná znaménka) a výsledkem je něco, co nebylo ani v jedné síti. Dochází tak spíše k mutaci, než k výměně informací. Celkově je tedy prostor vah 2^n $n!$ větší, než by ve skutečnosti měl být. Příkladem pokusu o eliminaci této redundance je křížení vah se stejným znaménkem u dvojic neuronů se stejným počtem kladných vah a záporných vah. Počet odpovídajících dvojic lze zvýšit případným přehozením všech znamének vah u neuronu.

Genetický algoritmus je definován v zásadě velmi volně a je na uživateli, aby si zvolil formu odpovídající jeho problému. Genetický algoritmus je založen na vhodné reprezentaci dat potenciálního řešení problému a musí obsahovat definici výměny informací, která vytváří z rodičovských řešení nová řešení – potomky. K hlavním parametrům algoritmu patří velikost populace (počet chromozómů) a pravděpodobnost mutace a křížení, případně spolu s metodou (procentem) vymírání méně úspěšných chromozómů tam, kde může část starých jedinců přežívat do nové populace. Ty nejlepší nově vytvořené chromozómy se mohou popřípadě převzít do nové populace všechny, a i nejlepší rodičovské chromozómy mohou případně přežívat (ale jen malé procento z nich, abychom neohraničili prostor prohledávání).

Výhodou genetických algoritmů je jejich obecnost, dají se upravit pro řešení nejrůznějších úkolů. Nevýhody genetických algoritmů vyplývají také z jejich obecné formulace, je zde spousta parametrů pro nastavení, široký výběr reprezentace dat, definicí

mutace a křížení. Z důvodů této obecnosti neexistuje pro genetické algoritmy hlubší teorie, která by zásadně pomohla při výběru reprezentace dat a nastavování parametrů. To je třeba v praxi provádět spíše metodou pokusu a omylu. Přesto se genetické algoritmy stále více využívají — nic zásadně lepšího zatím k dispozici není.

Literatura

[1] S.A. Harp and T. Samad. Genetic Synthesis of Neural Network Architecture. In: L. Davis, editor. *Handbook of Genetic Algorithms*, pp. 202-221, Van Nostrand Reinhold, New York, 1991.

[2] J.D. Schaffer, editor. Proceedings of the *Third International Conference on Genetic Algorithms,* pp. 360-397, Morgan Kaufmann Publishers, Los Altos, CA, 1989.

[3] R.F. Albrecht, C.R. Reeves and N.C. Steele, editors. *Artificial Neural Nets and Genetic Algorithms*, pp. 628-730. Springer Verlag, Wien, 1993.

[4] Z. Michalewicz. *A Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, Berlin, 1992.

[5] R.J. Mitchell, J.M. Bishop and W. Low. Using a genetic algorithm to find the rules of a neural network. In: R.F. Albrecht, C.R. Reeves and N.C. Steele, editors. *Artificial Neural Nets and Genetic Algorithms*, pp. 664-669, Springer Verlag, Wien, 1993.

[6] L. Lukšan. *Metody s proměnnou metrikou*. Academia, Praha 1990.

[7] A. Brunovská. *Malá optimalizácia. Metódy, programy, príklady*. Alfa, Bratislava, 1990.

[8] F. Glover. Tabu Search-Part I. *ORSA J. Comp*., 1: 190-206, 1989; Tabu Search-Part II. *ORSA J. Comp.,* 2: 4-32, 1990.

[9] F. Glover, editor. Tabu Search. *Annals of Operations Research*, Vol.41, 1993.

[10] F. Glover, M. Laguna. Tabu search. In: C.R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems,* pp. 70-150. Blackwell Scientific Publications, Oxford, 1993.

[11] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi. Optimization by simulated annealing. *Science* 220: 671-680 (1983).

[12] J. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Opt. Theory Appl.* 45: 41-51, 1985.

[13] P.M.J. van Laarhoven and E.H.L Aarts. *Simulated Annealing: Theory and Applications*. Reidel, Dordrecht, The Netherlands, 1987.

[14] R.H.J.M. Otten and L.P.P.P. van Ginneken. *The Annealing Algorithm*. Kluwer, Boston, 1989.

[15] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E.J. Teller. Equation for State Calculation for Fast Computing Machines. *J. Chem. Phys.*, 21: 1087-1092, 1953.

[16] J. Pospíchal and V. Kvasnička. Fast Evaluation of Chemical Distance by Simulated-Annealing Algorithm. *J. Chem. Inf. Comput. Sci.*, 33: 879-885, 1993.
[17] V. Kvasnička, J. Pospíchal, and D. Hesek. Augmented simulated annealing algorithm for the TSP. *Central European Journal for Operations Research and Economics*, 2: 307-317,1993.
[18] H.-P. Schwefel. *Numerical Optimization for Computer Models*. Wiley, Chichester, UK, 1981.
[19] H.-P. Schwefel and R. Manner, editors. *Proceedings of the First International Conference on Parallel Problem Solving from Nature.* Dortmund, Germany, 1990.
[20] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A Survey of Evolution Strategies. In: R.K. Belew, L.B. Booker, editors. *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 2-9, Morgan Kaufmann, San Mateo, CA, 1991.
[21] J. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.
[22] L. Davis, editor. *Genetic Algorithms and Simulated Annealing*. Morgan Kaufman, Los Altos, 1987.
[23] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.