

TECHNICKÁ UNIVERZITA V KOŠICIACH  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Dynamická adaptácia topológie Neural Network Ensemble pri  
aplikáciách rozpoznávania vzorov

Mikuláš DARABOŠ

DIPLOMOVÁ PRÁCA

2006

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra kybernetiky a umelej inteligencie

Dynamická adaptácia topológie Neural Network Ensemble pri  
aplikáciách rozpoznávania vzorov

**DIPLOMOVÁ PRÁCA**

Mikuláš Daraboš

Vedúci diplomovej práce:

Marek Bundzel, Ph.D.

Konzultant diplomovej práce:

Marek Bundzel, Ph.D.

Košice 2006

## Analytický list

---

Autor:	Mikuláš Daraboš
Názov práce:	Dynamická adaptácia topológie Neural Network Ensemble pri aplikáciách rozpoznávania vzorov
Jazyk práce:	slovenský
Typ práce:	Diplomová práca
Počet strán:	58
Akademický titul:	-
Univerzita:	Technická univerzita v Košiciach
Fakulta:	Fakulta elektrotechniky a informatiky (KKUI)
Katedra:	Katedra kybernetiky a umelej inteligencie (KKUI)
Študijný odbor:	Umelá inteligencia
Študijný program:	---
Mesto:	Košice
Vedúci DP:	Marek Bundzel, Ph.D.
Konzultanti DP:	Marek Bundzel, Ph.D.
Dátum odovzdania:	2. 5. 2006
Dátum obhajoby:	5. 6. 2006
Kľúčové slová:	Množina lineárnych perceptrónov, dynamická adaptácia topológie, lineárna klasifikácia, perceptrón.
Kategória Konspekt:	Výpočtová technika - Umelá inteligencia
Citovanie práce:	Daraboš, Mikuláš: Dynamická adaptácia topológie Neural Network Ensemble pri aplikáciách rozpoznávania vzorov. Diplomová práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2006. 58 s.
Názov práce v AJ:	Dynamic adaptation of topology of Neural Network Ensemble in applications for pattern recognition
Podnázov práce v AJ:	-
Kľúčové slová v AJ:	Ensemble of linear perceptrons, Dynamic adaptation of topology, linear classification, perceptron



### **Abstrakt v SJ**

Táto diplomová práca sa zaoberá s využitím Neural Network Ensemble (NNE – Množina neurónových sietí) v rozpoznávaní vzorov, konkrétne množinou lineárnych perceptrónov s výstupom dôveryhodnosti (ELPCLO – Ensemble of Linear Perceptrons with Confidence Level Output) a kladie za cieľ navrhnúť rozširujúci algoritmus na zabezpečenie dynamickej adaptácie topológie na základe zložitosti riešenej úlohy. Metóda ELPCLO je založená na princípe „rozdeľuj a panuj“ a je schopná celý priestor rozdeliť na menšie podpriestory a priradiť k niektorému členu siete. V určitej miere učenie je položené na konkurenčnom princípe, v ktorom víťazná sieť bude modifikovaná iným spôsobom než ostatné. Takáto rozšírená metóda by mala smerovať počas učenia k ideálnej topológii a zároveň dosiahnuť požadovanú úspešnosť v riešení klasifikačného problému. Práca obsahuje podrobný popis algoritmu, návrhu a realizácie dynamickej adaptácie topológie a **samozejm**e experimentálne overenie funkčnosti algoritmu na umelých a skutočných dátach a ich následné vyhodnotenie.

### **Abstrakt v AJ**

This diploma work is concerning with usage of Neural Network Ensemble (NNE) in applications of pattern recognition with Ensemble of Linear Perceptrons with Confidence Level Output (ELPCLO) method and is focused on suggest an algorithm which will improve the original method by realization of dynamic adaptation of topology based on problem complexity. ELPCLO uses the “divide and conquer” strategy for training the members. It can divide the problem space into subspaces and assign every subspace to a member in ensemble. Particularly the method is based on competitive learning where winner member is adapted by another technique than the rest part of ensemble. The aim of the modified algorithm is to find the ideal topology during the learning process and approach the expected efficiency of the actual classification problem at the same time. Thesis also includes detailed description of the algorithm and its modifications. Functionality of the algorithms are verified by experiments performed on benchmark and real-world data sets.

# Zadanie práce

## TECHNICKÁ UNIVERZITA V KOŠICIACH

Fakulta elektrotechniky a informatiky

Katedra kybernetiky a umelej inteligencie  
akademický rok: 2005/2006

## ZADANIE DIPLOMOVEJ PRÁCE

Diplomant: **Mikuláš D a r a b o š**

Odbor: Umelá inteligencia

Študijný program:

*Vzhľadom k tomu, že ste splnili požiadavky učebného plánu, zadáva Vám dekan fakulty na návrh vedúceho vedecko-pedagogického pracoviska v zmysle zákona o VŠ č.131/2002 a Študijného poriadku TU §15, ods. 3, túto tému záverečnej práce:*

### Dynamická adaptácia topológie Neural Network Ensemble pri aplikáciách rozpoznávania vzorov

#### POKYNY PRE VYPRACOVANIE

##### **Osnova práce:**

1. Teoretická analýza metódy Neural Network Ensemble.
2. Realizácia prehľadu príbuzných metód adaptácie topológie neurónových sietí.
3. Návrh algoritmu pre dynamickú adaptáciu topológie Neural Network Ensemble.
4. Vytvorenie integrovaného programového prostredia.
5. Experimentálne overenie na skutočných a umelých dátach. Porovnanie s príbuznými metódami.
6. Spracovanie výsledkov, analýza a diskusia výsledkov.
7. Vypracovanie dokumentácie podľa pokynov vedúceho diplomovej práce.

**Rozsah laboratórných a grafických prác :**

Laboratórne práce na počítači, programovanie, experimenty na vybraných dátach, vedecká analýza výsledkov a zhodnotenie vlastného vedeckého prínosu. Grafické práce v potrebnom rozsahu a odovzdanie v elektronickej forme na CD nosiči pripojenom k samotnej diplomovej práci.

**Rozsah záverečnej práce:** odporúčaný počet strán 40 a viac

**Zoznam odporúčanej literatúry:**

1. Pitoyo Hartono and Shuji Hashimoto, Ensemble of Linear Perceptrons with Confidence Level Output, 4th International Conference on Hybrid Intelligent Systems, 5-8 December 2004, Kitakyushu, Japan, pages 186-191, IEEE Computer Society, 2005, isbn 0-7695-2291-2
2. Sinčák, P., Andrejková, G. (1996). Neurónové siete–inžiniersky prístup (Dopredné NS ), Vol. 1. edition Elfa-press. ISBN 80-88786-38-X , ISBN 80-88786-42-8
3. Internet

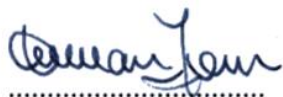
**Vedúci záverečnej práce:** Ing. Marek Bundzel, PhD.

**Konzultant:.**

**Dátum zadania záverečnej práce:** 10. 3. 2006

**Dátum odovzdania záverečnej práce:** 2. 5. 2006

V Košiciach, dňa 10. 3. 2006



prof. Ing. Dušan Kocur, CSc.  
dekan fakulty



prof. Ing. Ján Sarnovský, CSc.  
vedúci vedecko-pedagogického  
pracoviska

## **Čestné vyhlásenie**

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 2. 5. 2006

.....  
*vlastnoručný podpis*



## **Pod'akovanie**

Chcel by som poďakovať Ing. Marekovi Bundzelovi, PhD. za nápady, rady a pripomienky, ktoré mi veľmi pomohli pri vypracovaní tejto diplomovej práce. Ďalej by som chcel poďakovať ostatným spolužiakom a známym za ochotu pomôcť pri prekonávaní najrôznejších problémov, ktoré sa pri vypracovaní tejto práce vyskytli.

# Obsah

<b>Zoznam obrázkov .....</b>	<b>11</b>
<b>Zoznam tabuliek .....</b>	<b>12</b>
<b>Zoznam symbolov a skratiek .....</b>	<b>13</b>
<b>Slovník termínov .....</b>	<b>14</b>
<b>Úvod .....</b>	<b>15</b>
<b>1 Formulácia úlohy.....</b>	<b>17</b>
<b>2 Background.....</b>	<b>18</b>
2.1 Metodika rozpoznávacieho procesu .....	18
2.2 Prehľad metód klasifikácie.....	18
2.2.1 Lineárne klasifikátory .....	19
2.2.2 Klasifikátor na báze najbližšieho suseda .....	19
2.2.3 Viacvrstvé neurónové siete .....	19
2.2.4 Konvolučné siete.....	20
2.2.5 Bayesove siete.....	20
2.3 Presnosť klasifikácie .....	20
<b>3 Perceptrón.....</b>	<b>22</b>
3.1 Funkcionalita perceptrónu .....	22
3.1.1 Topológia perceptrónu.....	22
3.1.2 Učenie .....	23
<b>4 ELP - Množina lineárnych perceptrónov.....</b>	<b>24</b>
4.1 Príbuzné metódy.....	24
4.2 ELP s výstupom Confidence .....	25
4.2.1 Štruktúra .....	26
4.2.2 Učenie ELP.....	28
4.2.3 Učiaci signál $C^i(t)$ .....	29
<b>5 Implementácia základnej metódy .....</b>	<b>31</b>
5.1 Prekážky v základnom algoritme .....	31
5.1.1 Problém typu č. 1 .....	31
5.1.2 Problém typu č. 2.....	31
5.2 Úpravy základného algoritmu .....	31
5.2.1 Riešenie problém typu č. 1 .....	32

5.2.2 Riešenie problém typu č. 2 .....	32
5.3 Popis upraveného algoritmu .....	32
5.4 Ideálna štruktúra vzoriek trénovacích a testovacích množín .....	34
5.5 Grafické znázornenie fungovania ELPCLO.....	35
<b>6 Dynamická adaptácia topológie .....</b>	<b>37</b>
6.1 Návrh dynamickej adaptácie topológie .....	37
6.2 Časovanie zásahov .....	41
6.3 Dekrementácia počtu členov množiny .....	42
6.4 Inkrementácia počtu členov množiny .....	42
6.4.1 Výber najvhodnejšieho cieľového regiónu .....	43
6.4.2 Výpočet váh nového člena ELP.....	44
<b>7 Experimenty.....</b>	<b>46</b>
7.1 Experimenty na umelých dátach .....	46
7.1.1 Trénovacie a testovacie množiny.....	46
7.1.2 Nastavenie parametrov pre dáta „kruh vo štvorci“ .....	47
7.1.3 Nastavenie parametrov pre dáta „dvojité špirála“ .....	48
7.1.4 Experimenty na dátach „kruh vo štvorci“ .....	48
7.1.5 Experimenty na dátach „dvojité špirála“ .....	50
7.2 Experimenty na reálnych dátach .....	51
7.2.1 Použité dáta.....	51
7.2.2 Nastavenie parametrov metód .....	53
7.2.3 Experiment s metódou ELPCLO s DAT .....	54
7.2.4 Experiment s metódou MFARTMAP.....	54
7.2.5 Experiment s metódou Standard BP .....	54
7.3 Vyhodnotenie experimentov .....	54
<b>8 Záver.....</b>	<b>55</b>
<b>Zoznam použitej literatúry .....</b>	<b>56</b>
<b>Prílohy.....</b>	<b>57</b>
<b>Curriculum vitae.....</b>	<b>58</b>

## Zoznam obrázkov

Obr. 1 Schéma rozpoznávania.....	18
Obr. 2 Prameň: [1] Lineárna separovateľnosť v príznakovom priestore .....	22
Obr. 3 Prameň: [1] Topológia perceptrónu navrhnutého Rosenblattom .....	23
Obr. 4 Lineárny perceptrón s neurónom dôveryhodnosti .....	26
Obr. 5 Množina lineárnych perceptronov .....	27
Obr. 6 Grafické znázornenie výpočtu $C^i(t)$ .....	30
Obr. 7 Oblasti priradené k jednotlivým členom ELPCLO .....	36
Obr. 8 Výsledok klasifikácie s 10 členným ELPCLO .....	36
Obr. 9 Priamky separujúce jednotlivé podoblasti .....	36
Obr. 10 Vizualizácia trénovacích množín umelých dát .....	47
Obr. 11 Vizualizácia výsledkov na trénovacej množine - ELPCLO s DAT .....	49
Obr. 12 Vizualizácia výsledkov na trénovacej množine – MFARTMAP.....	50
Obr. 13 Vizualizácia výsledkov na trénovacej množine - MFARTMAP .....	51
Obr. 14 Oblasti ohraničené bielou čiarou boli označené expertom .....	52

---

## Zoznam tabuliek

Tab. 1 Kontingenčná tabuľka pre dve triedy .....	21
Tab. 2 Znázornenie $O_c^i(t)$ , učiaceho signálu $C^i(t)$ a chyby .....	29
Tab. 3 Kódovanie tried pomocou dvojkovej sústavy .....	34
Tab. 4 Kódovanie tried so šírkou ich počtu .....	35
Tab. 5 Experimentálne výsledky – ELPCLO s DAT– „kruh vo štvorci“ .....	49
Tab. 6 Experimentálne výsledky – MFARTMAP– „kruh vo štvorci“ .....	50
Tab. 7 [Klikni sem a napíš názov tabuľky] .....	50
Tab. 8 [Klikni sem a napíš názov tabuľky] .....	51
Tab. 9 Popis tried v reálnych dát .....	52
Tab. 10 Zastúpenie tried v trénovacej a testovacej množine .....	53

## Zoznam symbolov a skratiek

NN	<b>N</b> eural <b>N</b> etwork – Neurónová sieť
NNE	<b>N</b> eural <b>N</b> etwork <b>E</b> nsemble – Množina neurónových sietí
ELP	<b>E</b> nsemble of <b>L</b> inear <b>P</b> erceptrons – Množina lineárnych perceptrónov
CLO	<b>C</b> onfidence <b>L</b> evel <b>O</b> utput – Výstup dôveryhodnosti
ELPCLO	<b>E</b> nsemble of <b>L</b> inear <b>P</b> erceptrons with <b>C</b> onfidence <b>L</b> evel <b>O</b> utput
BP	<b>B</b> ack <b>P</b> ropagation of Error

## Slovník termínov

### Učiaci proces

**Dizertácia** je rozsiahla vedecká rozprava, v ktorej sa na základe vedeckého výskumu a s použitím (využitím) bohatého dokladového materiálu ako i vedeckých metód rieši zložitý odborný problém.

na tlačiarňi. To čo vidíme je písmo; font je súbor a nevidíme ho.

**Kritika** je odborne vyhotovený, prísny pohľad na hodnotenú vec. Medzi recenziou a kritikou je taký pomer ako medzi diskusiou a polemikou. Pri kritike treba prísnosť chápať v tom zmysle, že sa v nej okrem iného navrhuje, ako hodnotené dielo skvalitniť.

**Meter** (m) je vzdialenosť, ktorú svetlo vo vákuu prejde za časový interval  $1/299\,792\,458$  sekundy.

**Písmom** rozumieme vlastný vzhlád znakov.

**Problém** termín používaný vo všeobecnom zmysle vo vzťahu k akejkoľvek duševnej aktivite, ktorá má nejaký rozoznateľný cieľ. Samotný cieľ nemusí byť v dohľadne. Problémy možno charakterizovať/ tromi rozmermi -- oblasťou, obtiažnosťou a veľkosťou.

**Proces** je postupnosť či rad časovo usporiadaných udalostí tak, že každá predchádzajúca udalosť sa zúčastňuje na determinácii nasledujúcej udalosti.

## Úvod

Umelá inteligencia, ako už bolo mnoho krát konštatované určil si pri svojom vzniku za primárny cieľ vytvoriť model ľudského mozgu s jeho komplexnosťou a robustnosťou. Čoskoro sa však tento cieľ javil ako nerealizovateľný a postupne sa pretransformoval na hľadanie parciálnych riešení, ktoré mohli byť úspešne aplikované v reálnom svete. Rýchly technický rozvoj v blízkej minulosti prispel k rapidnému zvýšeniu záujmu o prostriedky výpočtovej inteligencie v oblastiach vytvorených novými technológiami pre riešenie problémov aké sú napr. klasifikácia, rozpoznanie prirodzeného jazyka, predikcie, reprezentácie vedomostí a riešenia matematických úloh.

V dnešnej dobe už tímy s veľkou finančnou a technickou podporou sa snažia realizovať komplexné riešenia napr. v podobe humanoidných robotov, ale dominantná časť vedeckých prác sa sústreďuje na vypracovanie a zdokonalenie čiastočných riešení, ktoré sa týkajú najmä riešenia nedeterministických problémov ako napr. rozpoznanie vzorov, obrazu, reči a nie je výnimkou ani táto práca.

Neurónové siete sú jedným z prostriedkov umelej inteligencie dnes už s veľkým všeobecným potenciálom v riešení širokého spektra problémov so základnou stavebnou jednotkou neurón. V mnohých prípadoch na riešenia spomenutých problémov využívajú zložité, prepracované neurónové siete, akými sú napr. ARTMAP, MFARTMAP, rôzne typy siete LetNet a na druhej strane sa nachádzajú riešenia založené na jednoduchých neurónových sieťach ako napr. perceptrón alebo viacvrstvový perceptrón.

Základnou myšlienkou tejto práce je práve využitie takýchto sietí, konkrétne perceptrónu, v rozpoznaní vzorov. Z vlastností perceptrónu však vidieť jeho obmedzenia z hľadiska možného počtu klasifikovaných tried a spôsobu klasifikácie. Na elimináciu týchto obmedzení bol navrhnutý všeobecný model NNE (Neural Network Ensemble), v tomto prípade ELP (Ensemble of Linear Perceptrons), ktorý zachováva jednoduchosť klasifikátora a pritom zvyšuje jeho efektivitu zaradením ich do množiny v rámci ktorej budú môcť spolupracovať a úspešne riešiť aj zložitejšie problémy.

Základnú inšpiráciu priniesla publikácia , v ktorej bol popísaný algoritmus riešenia klasifikačného problému pomocou množiny lineárnych perceptrónov



s neurónom dôveryhodnosti realizujúci teóriu „rozdiel a panuj“. V tejto metóde počet členov množiny sa určuje manuálne a je konštantné počas učenia a s tým ovplyvňuje výsledok učiaceho procesu, ich nedostatočný počet spôsobí nedostatočnú úspešnosť a naopak priveľa členov množiny prinesie očakávaný výsledok, ale môže spomaliť učiaci proces a následne znížiť rýchlosť neurónovej siete v aplikáciách. Ideálnym riešením by bola úprava algoritmu s možnosťou dynamickej adaptácie počtu členov množiny, t.j. dynamickou adaptáciou topológie a práve to je predmetom a cieľom tejto diplomovej práce. Od riešenia sa očakáva, že na konci učiaceho procesu počet členov sa nastaví na optimálnu hodnotu z hľadiska riešiteľnosti problému. V nasledovných kapitolách je podrobne popísaný algoritmus ELP s CLO, návrh, spôsob realizácie dynamickej adaptácie topológie ELPCLO, následné experimenty a ich vyhodnotenie.

# 1 Formulácia úlohy

Úlohy tejto diplomovej práce sú definované nasledovne:

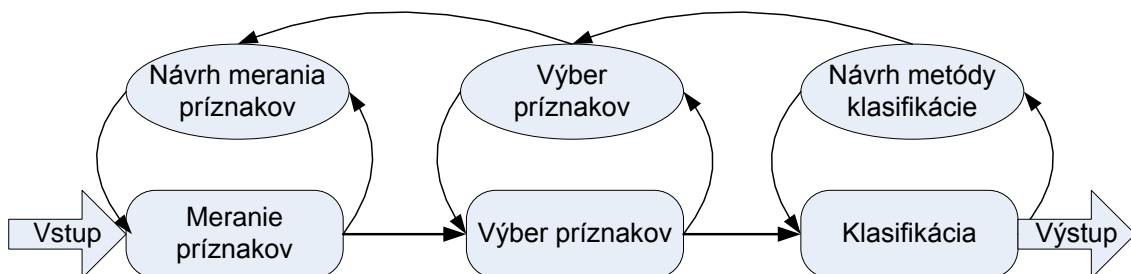
1. Realizácia rešerši metód a nástrojov umelej inteligencie v rozpoznávaní vzorov.
2. Teoretická analýza problematiky rozpoznávania vzorov pomocou množiny lineárnych perceptrónov a prehľad príbuzných metód.
3. Realizácia metódy ELPCLO a experimenty na umelých dátach.
4. Návrh a realizácia algoritmu s dynamickou adaptáciou topológie ELPCLO.
5. Realizácia experimentov na umelých a reálnych dátach
6. Vyhodnotenie experimentov a porovnanie s inými metódami

## 2 Background

### 2.1 Metodika rozpoznávacieho procesu

Doménu rozpoznávacích procesov tvorí veľká časť problémov, pre riešenie ktorých sa dnes používajú prostriedky výpočtovej inteligencie. Proces rozpoznávania spočíva v zaradovaní javov a situácií reálneho sveta do tried. Každý z posudzovaných objektov je pri dostatočne podrobnom popise jedinečný, pojem "trieda" predstavuje istú generalizáciu. Pojmy rozpoznávanie a klasifikácia spolu medzi sebou úzko súvisia, ale je medzi nimi rozdiel. Pri rozpoznávaní nepoznáme počet tried pred identifikáciou, ale triedy sa vytvárajú počas behu rozpoznávania a v klasifikačnej úlohe poznáme dopredu počet tried. Samotný proces rozpoznávania (Obr. 1) je možné rozdeliť na tri základné časti:

- **meranie príznakov** - stanovenie hľadiska, z ktorého bude proces posudzovaný
- **výber príznakov** - redukcia príznakového priestoru
- **klasifikácia** do tried



Obr. 1 Schéma rozpoznávania

### 2.2 Prehľad metód klasifikácie

Základné typy klasifikátorov predstavujú nasledovné nástroje klasifikácie: **Nie je to celkom vyčerpávajúce, chýbajú ti napr. SVM, uveď to ako „Niektoré typy klasifikačných algoritmov“.**

- Lineárny klasifikátor
- Klasifikátor na báze najbližšieho suseda
- Viacvrstvové neurónové siete

- Konvolučné siete
- Bayesove siete

### 2.2.1 Lineárne klasifikátory (patrílo by sa ku každému uviesť citáciu)

Tento skupina je najvýznamnejšia z hľadiska tejto práce, nakoľko predstavuje základný princíp klasifikačného algoritmu navrhnutého v nej.

Lineárny klasifikátor je klasifikátor, ktorý zakladá svoje rozhodnutie na výstupe lineárnej funkcie jeho vstupov. Funkcia vo prevažnej väčšine prípadov jednoduchá a mapuje hodnoty nad určitým prahom do jednej triedy a pod prahom do druhej triedy. Z hľadiska aplikácie je veľmi jednoduchý, ale jeho výsledky patria medzi najhoršie.

### 2.2.2 Klasifikátor na báze najbližšieho suseda

Jeden z ďalších jednoduchých klasifikátorov na báze algoritmu K-najbližších susedov, ktorý pracuje s Euklidovskou vzdialenosťou. Jeho výhoda je v tom, že sa nevyžaduje tréning. Nevýhodou je však pomerne veľká pamäťová náročnosť a čas rozpoznania pri väčšom rozmere vstupu môže byť zdĺhavý. V praxi sa zvykne používať hodnota  $K=3$  alebo  $K=5$ , pre ktoré experimenty vykazovali najlepšie výsledky. Pretože metóda používa Euklidovu vzdialenosť môže byť vstupná informácia reprezentovaná aj vektorovo

### 2.2.3 Viacvrstvové neurónové siete

V praxi sa používajú tieto siete s jednou, maximálne s dvoma skrytými vrstvami. Veľkosť vstupnej vrstvy je závislá od rozmeru vstupu. Počet neurónov v skrytej vrstve môže byť ľubovoľný, ale pri návrhu je potrebné zohľadniť fakt, že čím väčší počet neurónov, tým väčší počet váh a tým samozrejme rastie časová náročnosť vo fáze učenia siete. Poznatky siete sú uložené vo váhach medzi neurónmi. Vo fáze učenia, keď sa sieti poskytne tréningová množina dát, sa váhy menia. Ako algoritmus učenia môžeme použiť známy algoritmus spätného šírenia chyby (error backpropagation), prípadne jeho modifikácie. Kvalita klasifikácie pri použití doprednej neurónovej siete je ovplyvnená mnohými parametrami siete. Už pri 3-vrstvovej sieti môžeme dosiahnuť dobré výsledky. Výhodou takejto siete je aj krátky čas klasifikácie.

### 2.2.4 Konvolučné siete

Konvolučné neurónové siete sú odvodené od dopredných neurónových sietí s úplným prepojením. Môžeme ich charakterizovať ako komplexné neurónové siete, ktoré realizujú generovanie príznakov na rozpoznávanie a zároveň sú schopné klasifikovať vstupné objekty do jednotlivých tried. Tieto siete eliminujú nedostatky dopredných viacvrstvových sietí s úplným prepojením, ako nevhodnosť generovania príznakov na rozpoznávanie objektov a problémy súvisiace s invariantnosťou a šumom objektov. Boli navrhnuté rôzne architektúry konvolučných neurónových sietí. Konvolučné neurónové siete boli úspešne nasadené v rôznych oblastiach spracovania obrazov, ako napríklad on-line a off-line rozpoznávanie strojového a rukou písaného písma alebo pri rozpoznávaní tváří.

### 2.2.5 Bayesove siete

Bayesova sieť je ľubovoľný orientovaný graf, ktorý spĺňa nasledujúce vlastnosti:

- Každý vrchol v sieti predstavuje jednu pravdepodobnostnú premennú
- Každá hrana je orientovaná a určuje závislosť medzi premennými (podmienujú pravdepodobnosť).
- Pre všetky vrcholy, do ktorých nevstupuje žiadna hrana je určená distribúcia pravdepodobnosti pre jednotlivé stavy premennej.
- Pre všetky vrcholy, do ktorých vedú hrany z iných vektorov, existuje tabuľka podmienenej pravdepodobnosti

## 2.3 Presnosť klasifikácie

Odhad presnosti klasifikácie sa realizuje na množine vektorov atribútov, ktorých klasifikácia je dopredu známa, a čo najlepšie pokrýva priestor, v ktorom sa rozpoznávanie realizuje. Takúto množinu nazývame testovacou množinou. Po vyhodnotení množiny klasifikátorom sú k dispozícii skutočné výsledky klasifikácie, ako aj očakávané hodnoty **kalifikácie**. Na základe porovnania skutočných a očakávaných výsledkov klasifikácie sa definujú niektoré koeficienty odhadu presnosti klasifikácie.

Koeficient PCC vyjadruje percentuálny pomer správne klasifikovaných vektorov testovacej množiny ku všetkým vektorom testovacej množiny.

$$PCC = \frac{P_s}{P_v} \cdot 100$$

kde  $P_s$  je počet správne klasifikovaných vektorov a  $P_v$  je počet všetkých vektorov.

Kontingenčná tabuľka predstavuje podrobnejšiu analýzu klasifikačného procesu a percentuálne vyjadruje počty správne a nesprávne klasifikovaných vektorov pre každú triedu zvlášť.

**Tab. 1** Kontingenčná tabuľka pre dve triedy

		Aktuálna trieda	
		A	B
Predikovaná trieda	A'	PAA'	PBA'
	B'	PAB'	PBB'

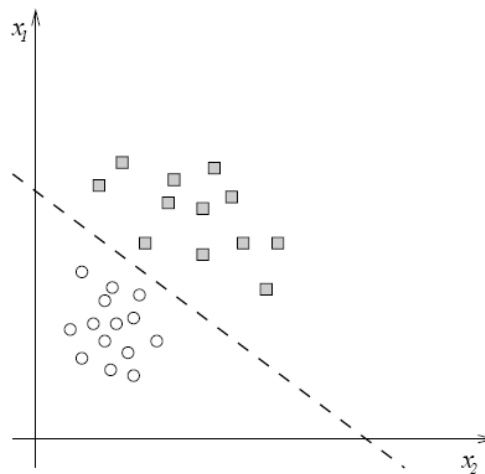
Tab. 1 znázorňuje príklad kontingenčnej tabuľky pre dve triedy, kde hodnota  $P_{AA'}$  vyjadruje percentuálne pomer počtu správne klasifikovaných vektorov triedy A ku počtu všetkých vektorov triedy A. Hodnota  $P_{AB'}$  predstavuje percentuálne pomer počtu vektorov triedy A, ktoré boli klasifikované do triedy B ku počtu všetkých vektorov triedy A. Podobne to platí aj pre hodnoty triedy B.

### 3 Perceptrón

Perceptrón bol navrhnutý Rosenblattem v roku 1958 pre dichotomickú klasifikáciu, tj. pre rozdelenie do dvoch tried, pri ktorých sa predpokladá, že triedy sú lineárne separovateľné v príznakovom priestore a následne pridal svoju konvergenčnú teóriu perceptrónu.

#### 3.1 Funkcionalita perceptrónu

Nech je daná množina vektorov  $X=\{x^j\}$ ,  $j=1,\dots,k,\dots,\infty$  v  $n$ -rozmernom priestore. O každom z týchto vektorov vieme, že určite patrí do triedy **CL1** alebo **CL2**. Pod lineárnou separovateľnosťou dvoch tried rozumieme situáciu, keď existuje možnosť oddeliť objekty v príznakovom priestore pomocou **nadroviny** (neviem či sa to používa, napíš možno radšej hyperplochy) napr. priamka v 2-rozmernom alebo rovina v 3-rozmernom priestore. Príklad lineárnej separovateľnosti v rovine je na Obr. 1.



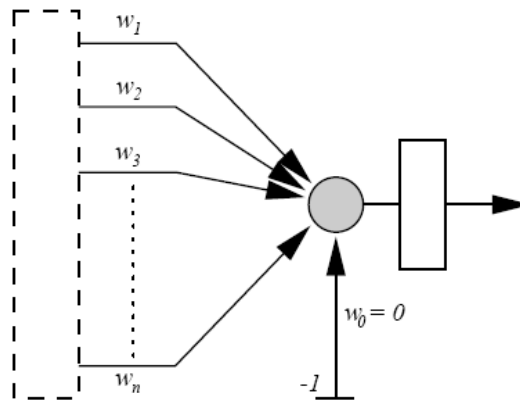
Obr. 2 **Prameň: [1]** píše sa „prebraté z ...“ Lineárna separovateľnosť v príznakovom priestore

##### 3.1.1 Topológia perceptrónu

Jednoduchý perceptrón má 3 vrstvy:

- senzorová vrstva
- asociatívna vrstva
- výstupný neurón

ktoré sú znázornené na Obr.3.



Obr. 3 Prameň: [1] Topológia perceptrónu navrhnutého Rosenblattem

Vstup do výstupného neurónu je daný rovnicou (1)

$$in(t) = \sum_{j=1}^n w_j(t)x_j(t) - \theta \quad (1)$$

Kde  $n$  je počet neurónov v asociatívnej vrstve,  $w_j(t)$  sú váhy medzi asociatívnu vrstvou a výstupným neurónom, kde neurón  $j$  je v asociatívnej vrstve a na vstupe je  $k$ -ty prvok množiny  $X$ ,  $x_j(t)$  je stav  $j$ -teho neurónu a  $\theta$  je prah. Výstup prechádza prahovaním v zmysle konečného výstupu

$$ou(t) = \begin{cases} 1 & \text{ak } in(t) \geq 0 \\ 0 & \text{ak } in(t) < 0 \end{cases} \quad (2)$$

V prípade elementárneho perceptrónu je separujúca nadrovina daná rovnicou (3)

$$\sum_{j=1}^n w_j(t)x_j(t) - \theta = 0 \quad (3)$$

### 3.1.2 Učenie

Učenie preceptróna prebieha nasledovne:

ak vstupný vektor  $x_j(t)$  je klasifikovaný správne pomocou  $w_j(t)$  potom nenastáva zmena váh pre  $w(t+1)$ :

$$w(t+1) = w(t), \quad (4)$$

ak vstupný vektor  $x_j(t)$  je klasifikovaný nesprávne pomocou  $w_j(t)$  potom nastáva zmena váh  $w(t+1)$  nasledovným spôsobom:

1. ak  $w(t)x(t) \geq 0$  a  $x(t)$  skutočne patrí do CL2 potom  $w(t+1) = w(t) - \gamma x(t)$
2. ak  $w(t)x(t) \leq 0$  a  $x(t)$  skutočne patrí do CL1 potom  $w(t+1) = w(t) + \gamma x(t)$



## 4 ELP - Množina lineárnych perceptrónov

V súčasnosti existujú rozvinuté a prepracované neurónové siete typu napr. ARTMAP, MFARTMAP a rôzne konvolučné siete, ktoré sú zložité z hľadiska topológie a funkčnosti. Na druhej strane je riešenie preferujúce využitie množiny jednoduchých, na určitej úrovni kooperujúcich sietí namiesto jednej robustnej konvenčnej NN v riešení klasifikačných úloh. V posledných rokoch bolo uvedené niekoľko riešení založené na množine neurónových sietí s cieľom zvýšiť efektivitu v situáciách, kedy klasické neurónové siete pozostávajúce z jednej siete neuspeli. Predmetom tejto diplomovej práce je rozšírenie metódy ELPCLO prof. Hashimota a prof. Hartona [2] o dynamickú adaptáciu topológie.

### 4.1 Príbuzné metódy

V minulosti boli vyvinuté neurónové siete typu používajúce metódu NNE s rôznym prístupom riešenia problému [6][7][8] s cieľom zvýšiť efektivitu, ktoré môžeme rozdeliť do nasledovných troch skupín:

1. NNE s nezávislými členmi bez rozdelenia príznakového priestoru
2. NNE s členmi priradenými ku jednotlivým triedam bez rozdelenia príznakového priestoru
3. NNE s členmi priradenými ku jednotlivým podoblastiam

V prvom prípade v NNE sú členy rovnakej alebo rôznej topológie, ktoré môžu byť navrhnuté napr. pomocou evolučných algoritmov, genetických algoritmov alebo heuristicky a podobné vlastnosti majú aj nastavenia učiacich parametrov. Počas učiaceho procesu každý jeden člen sa učí paralelne a nezávisle od ostatných členov. Na konci učiaceho procesu sa vyhodnotia jednotlivé členy a podľa analýzy sa určí člen s najväčšou úspešnosťou na celom príznakovom priestore. Ostatné členy množiny sa odstránia. Metódu patriacej do tejto skupiny predstavuje Ensemble Learning of Linear Perceptrons popísaná v odbornom článku [2].

Druhý typ predstavuje siete v ktorých členy sú inicializované najčastejšie s rovnakou topológiou a rovnakými učiacimi parametrami. Počas učenia členy sa špecializujú na jednotlivé triedy klasifikácie na celom príznakovom priestore, to

znamená, že jednotlivé členy sa aktivujú podľa jednotlivých tried. Na konci učiaceho procesu sa uloží celý sieť a v takej podobe bude aplikovaná.

Tretia skupina zahŕňa typ siete, v ktorých členy množiny sa inicializujú s rovnakou topológiou a učiacimi parametrami, učia sa paralelne a počas učenia sa stanú expertmi na jedinečných oblastiach príznakového priestoru a každý člen na svojej unikátnej oblasti klasifikuje do každej triedy klasifikácie. Príkladom tejto skupiny je metóda s názvom „Mixture of Expert“ [4] [5], ktorá priradí časť príznakového priestoru k lokálnemu expertu pomocou konkurenčného učenia a kombinuje výstupy expertov cez prahovaciú sieť. Ďalším veľmi známym algoritmom je Boosting [9][10], ktorý vykoná učenie niekoľkých slabých klasifikátorov a následne kombinuje ich s cieľom vytvoriť silný klasifikátor. **Samozrejme** do tejto skupiny patrí aj metóda tvoriaca základ tejto diplomovej práce: „Ensemble of Linear Perceptrons with Confidence Level Output“ [2] **Nie som si istý či tam patrí. Boosting počíta výslednú hypotézu ako sumu príspevkov od každej slabej hypotézy, ELP volí jednu víťaznú slabú hypotézu. Hashimoto sa mi vyjadril, že boosting je iné. Skús ten prehľad príbuzných metód rozťahnuť tak zhruba na 2 strany, pridaj obrázky z tých článkov, trochu podrobností a urob z toho samostatnú kapitolu. Možno spomeň aj Voting Neural Networks, Sinčák s tým kedysi robil.**

## 4.2 ELP s výstupom Confidence

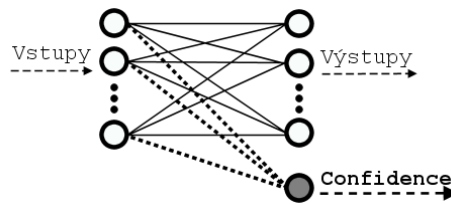
Základom prístupu je rozdeliť príznakový priestor na viac podoblastí a tieto podoblasti priradiť k jednotlivým sieťam z ELP. Vďaka jednoduchosti neurónových sietí bude očakávané rýchlejšie učenie. Jednoduché neurónové siete predstavujú v našom prípade lineárne perceptrony, ktoré budú mať na výstupe jeden neurón s výstupom s označením „Confidence Level Output“, ktorý predstavuje dôveryhodnosť danej siete pri aktuálnom vstupe a v každom prípade sa uznáva za výstup výsledok zo siete s najväčšou hodnotou dôveryhodnosti. Tým pádom každá jedna sieť systému bude mať svoju oblasť, kde bude najúspešnejšia a teda bude vyhodnotená ako najdôveryhodnejšia. Naším cieľom je zdokonaľovať tento systém, konkrétne pridaním možnosti adaptácie váh počas učenia za účelom znížiť čas učenia a zvýšiť efektívnosť. Množina sa však musí adaptovať k riešenému problému automaticky, tj. po náhodnej inicializácii má rozdeliť celú oblasť problému na viac podoblastí a tieto menšie časti

priradiť pomocou „confidence“ výstupu k jednotlivým členom množiny, ktoré budú učené na im prislúchajúcich podoblastiach.

#### 4.2.1 Štruktúra

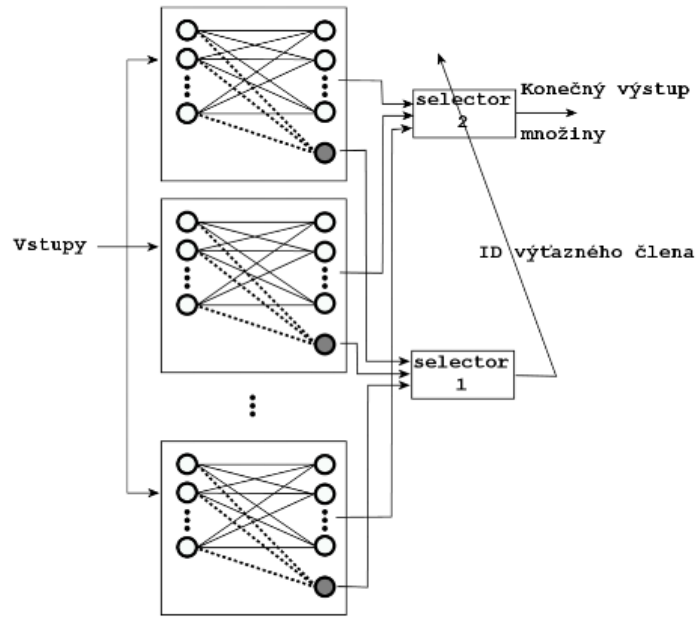
Tento systém je založený na princípe „rozdeľ a panuj“, kde neurónová sieť je schopná celý priestor rozdeliť na menšie podpriestory a priradiť k niektorému členu siete. V určitej miere učenie je položené na konkurenčnom princípe, v ktorom víťazná sieť bude modifikovaná iným spôsobom než ostatné.

V našom prípade použijeme skupinu lineárnych perceptrónov, ktoré budú mať pridaný výstupný neurón pre stupeň dôveryhodnosti („confidence“). V prípade, že výstup neurónu dôveryhodnosti bude mať veľkú hodnotu člen má vysokú úspešnosť, v opačnom prípade je málo významný v danom podpriestore, resp. pri aktuálnom vstupe.



Obr. 4 Lineárny perceptrón s neurónom dôveryhodnosti

Na Obr.4 vidíme množinu lineárnych perceptrónov, kde vstup prichádza nezávisle na každý jeden člen a je spracovaný tiež nezávisle následne výstupné hodnoty z neurónov dôveryhodnosti prichádzajú do časti *selector 1*, v ktorom sa určí člen s najväčšou hodnotou a jeho identifikačné číslo sa posielajú ďalej do časti *selector 2*, v ktorom sa rozhoduje o výstupe podľa ID, teda na konečný výstup sa dostane hodnota z člena s najväčšou hodnotou dôveryhodnosti.



Obr. 5 Množina lineárných perceptronov

Výstupy z  $j$ -tého výstupného neurónu  $i$ -tého člena  $O_j^i$  sú popísane nasledovne:

$$O_j^i = \sum_{k=1}^{N_{in}} w_{kj}^i I_k + \theta_j^i \quad (5)$$

kde,  $w_{kj}^i$  je váha spojenia medzi  $k$ -tým vstupným a  $j$ -tým výstupným neurónom v  $i$ -tého člena,  $I_k$  je  $k$ -tý element vstupného vektora,  $\theta_j^i$  je prahová hodnota  $j$ -tého výstupného neurónu  $i$ -tého člena, kým  $N_{in}$  je počet vstupných neurónov.

Výstup z neurónu dôveryhodnosti je vypočítané podľa nasledujúcej rovnice:

$$O_c^i = \sum_{k=1}^{N_{in}} v_{kc}^i I_k + \theta_c^i \quad (6)$$

kde  $v_{kc}^i$  je váha spojenia medzi  $k$ -tým vstupným neurónom a neurónom dôveryhodnosti  $i$ -tého člena a  $\theta_c^i$  je prahová hodnota neurónu dôveryhodnosti

Výstup z množiny lineárných perceptronov je nasledovný:

$$w = \arg \max_i \{O_c^i\} \quad (7)$$

$$O^{ens} = O^w \quad (8)$$

Kde  $O_c^i$  hodnota dôveryhodnosti  $i$ -tého člena a  $O^w$  je výstup člena s najväčšou hodnotou dôveryhodnosti.

#### 4.2.2 Učenie ELP

Počas učenia v tomto modeli len v prípade člena s najväčšou hodnotou dôveryhodnosti je povolená zmena váh spojení medzi klasickými výstupnými a vstupnými neurónmi podľa nasledovných vzťahoch:

$$W^w(t+1) = W^w(t) - \eta \frac{\partial E(t)}{\partial W^w(t)}, \quad (9)$$

$$E(t) = \|O^w(t) - D(t)\|^2, \quad (10)$$

$$W^i(t+1) = W^i(t) \quad \text{pre} \quad i \neq w, \quad (11)$$

kde  $W^w$  je vektor váh spojení medzi vstupnými a klasickými výstupnými neurónmi člena s najväčšou hodnotou dôveryhodnosti a  $D$  je učiaci signál.

Napriek tomu v každej jednej tréningovej iterácii je potrebné zmeniť váhy spojení medzi vstupnými neurónmi a neurónom dôveryhodnosti v každom jednom člene množiny pomocou nasledovných rovníc:

$$W_c^i(t+1) = W_c^i(t) - \eta \frac{\partial E_c^i(t)}{\partial W_c^i(t)}, \quad (12)$$

$$E_c^i(t) = \|O_c^i(t) - C^i(t)\|^2 \quad (13)$$

$$C^i(t) = 1 - \frac{\exp(\alpha \|O_c^i(t) - O_c^w(t)\|^2)}{\sum_{j=1}^N \exp(\alpha \|O_c^j(t) - O_c^w(t)\|^2)} \quad (14)$$

kde  $W_c^i$  je váha spojení medzi vstupnými neurónmi a neurónmi dôveryhodnosti  $i$ -tého člena, kým  $O_c^i$  je výstup neurónu dôveryhodnosti  $i$ -tého člena.  $\alpha$  je kladná konštanta.

$C^i$  je relatívna dôveryhodnosť  $i$ -tého člena s ohľadom na dôveryhodnosť každého člena množiny, ktorú môžeme uvažovať ako učiaci signál pre neurón dôveryhodnosti  $i$ -tého člena.  $C$  je určené pomocou rovnice (6) na zvýšenie rozmanitosti členov zosilnením chyby členov so zámerom deaktivovať členy s nízkym stupňom dôveryhodnosti.  $N$  udáva počet členov v množine. Je zrejmé, že znalosť člena s vysokým stupňom dôveryhodnosti z konkrétneho aktuálneho vstupu bude spevnená, kým člen s nízkym stupňom dôveryhodnosti nebude profitovať z aktuálneho vstupu. To znamená, že v ďalšej iterácii, keď sa znovu objaví ten istý vstup, člen s najväčším

stupňom dôveryhodnosti uspeje lepšie a samozrejme pre tento člen bude zvýšená dôveryhodnosť. Učiaci proces vedie k tomu, že sa vytvorí prednosť alebo povýšenie člena pre určitú časť vstupného regiónu, ktorý učiaci proces rozdelí na menšie regióny. Ku každému regiónu bude prislúchať jeden člen množiny, teda jeden perceptrón a práve to je podstata pri vytváraní rôznorodosti perceptrónov, aby neostali dominantné na oblastiach inicializácie, ale rozmiestnili sa po celom príznakovom priestore a každý si našiel svoju jedinečnú pozíciu.

### 4.2.3 Učiaci signál $C^i(t)$

Učiaci signál pre váhy smerujúce od asociatívnej vrstvy k neurónu dôveryhodnosti sa generuje v každej iterácii zvlášť pre každý člen pomocou aktuálnych výstupov „confidence“ neurónov.

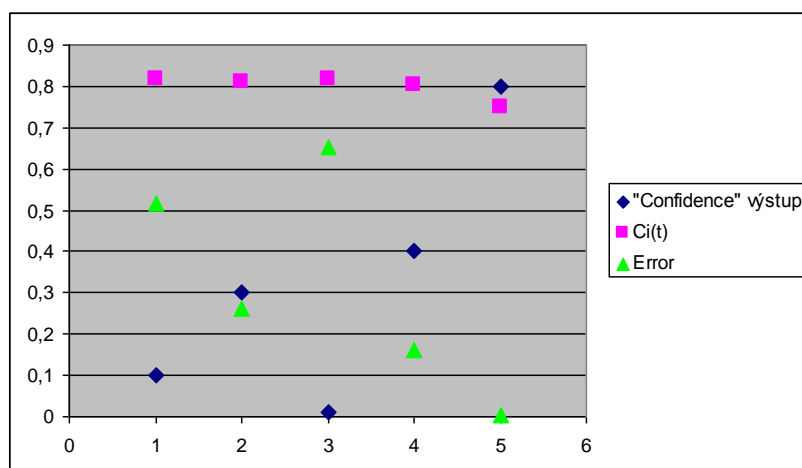
Generovanie učiaceho signálu je znázornené nasledovným príkladom:

Počet členov: 5

Tab. 2 Znáznornenie  $O_c^i(t)$ , učiaceho signálu  $C^i(t)$  a chyby

	Člen č.1	Člen č.2	Člen č.3	Člen č.4	Člen č.5
„Confidence“ výstup	0,1	0,3	0,01	0,4	0,8
$C^i(t)$	0,8177	0,8102	0,8186	0,8034	0,7501
$E_i$	0,5150	0,2603	0,6538	0,1628	0,0025

V predchádzajúcej tabuľke (Tab.2) a v nasledovnom grafe (Obr.6) je znázornený súvislosť medzi confidence výstupom členov, učiacim signálom  $C^i(t)$  a chybou, určenou podľa výrazov (12), (13) a (14).



Obr. 6 Grafické znázornenie výpočtu  $C^i(t)$

## 5 Implementácia základnej metódy

Prvotným krokom v práci predstavuje implementácia základnej metódy ELP popísanej v kapitole 4.2. Počas realizácií metódy sa objavili prekážky, ktoré vyžiadali úpravu základného algoritmu. Prekážky a vykonané úpravy sú popísané v kapitolách 5.1 a 5.2.

### 5.1 Prekážky v základnom algoritme

Problémy, ktoré sa objavili počas realizácie programového prostredia je možné rozdeliť podľa ich charakteristiky na dva typy:

1. problémy vyplývajúce z povahy reprezentácie desatinných čísel v počítačových systémoch
2. problémy vyplývajúce zo správania sa základného algoritmu

#### 5.1.1 Problém typu č. 1

Prvá prekážka sa objavila počas realizácie základnej metódy vo vzorci (14), konkrétne v nasledovnej časti výrazu:

$$\exp(\alpha \| O_c^i(t) - O_c^w(t) \|^2),$$

ktorý v prípadoch, keď hodnota  $O_c^i(t)$  (confidence výstup  $i$ -teho člena v čase  $t$ ) je rádovo menšia ako nula, hodnota  $O_c^w(t)$  (confidence výstup víťazného člena v čase  $t$ ) je rádovo väčšia ako nula presahuje maximálnu hodnotu typu *double* a tým znemožní stabilnú realizáciu programového prostredia.

#### 5.1.2 Problém typu č. 2

Problém typu č.2 sa objavil počas učiaceho procesu, keď v prevažnej väčšine experimentov prevzal dominanciu na celom príznakovom priestore jeden člen ELP. Tento jav bol spôsobený v určitej miere v dôsledku príčiny chyby popísanej v podkapitole 5.1.1, kde kvôli vznikajúcim veľkým chybám váhy medzi vstupnými a „Confidence“ neurónom začali oscilovať a zväčšovať sa smerom do nekonečna.

### 5.2 Úpravy základného algoritmu

Odstránenie chýb a úpravy sú popísané v nasledovných podkapitolách, konkrétne popis prechodu na sigmoidálnu aktivačnú funkciu, následný návrat ku klasickej



lineárnej a zavedenie normalizácie „confidence“ výstupov pomocou zmeny prahu confidence neurónu víťazného člena.

### 5.2.1 Riešenie problém typu č. 1

V odstraňovaní prekážky vo vzorci (13) boli využité vlastnosti sigmoidálnej aktivačnej funkcie namiesto lineárnej. S použitím sigmoidálnej funkcie výstupné hodnoty confidence neurónov boli redukované do intervalu (0,1) a vďaka tomu bola odstránená možnosť prekročenia šírky *double* pri výpočte učiaceho signálu  $C^i(t)$ . Eliminácia tohto problému však nevyriešila „Problém typu č. 2“ a naďalej pretrvával jav vznikajúci správaním sa jedného člena, ktorý po určitom počte iterácií (po zmene aktivačnej funkcie **síce** menej agresívne) nepretržite prevzal dominanciu na celom príznakovom priestore a s tým znemožnil úspešnosť učiaceho procesu.

### 5.2.2 Riešenie problém typu č. 2

Počas riešenia „Problému typu č.2“ bola využitá normalizácia confidence výstupu, ktorá v realizácií predstavovala znižovanie výstupnej hodnoty víťazného „confidence“ neurónu s cieľom zabrániť nadmiernemu zvyšovaniu jeho hodnoty a tým normalizovať stav „confidence“ neurónov. Táto úprava priniesla pokrok v ladení algoritmu a prispela k stabilite algoritmu.

## 5.3 Popis upraveného algoritmu

Realizácia výpočtu výstupných hodnôt a selekcia víťazného člena je identická so spôsobom v kapitole **aa**, zmeny sa týkajú učiacej časti algoritmu. Ako v pôvodnom algoritme prezentovanom v **aa**, v každej iterácií sa adaptujú váhy medzi vstupnými neurónmi a regulárnymi výstupnými neurónmi. Oproti pôvodnému algoritmu bola zmenená z gradientovej na klasický spôsob učenia neurónovej siete perceptrón popísanej v **aa** a zmena sa vykonáva, len v prípade, keď víťazný člen nesprávne klasifikoval vstupný vektor. Konečný spôsob adaptácie je prezentovaný nasledovnými vzťahmi:

$$W^w(t+1) = W^w(t) - \eta D_j X_j, \quad (14)$$

$$\theta^w(t+1) = \theta^w(t) - \eta D_j, \quad (15)$$

$$W^i(t+1) = W^i(t) \quad \text{pre} \quad i \neq w, \quad (16)$$

kde  $W^w$  je vektor váh spojení medzi vstupnými a klasickými výstupnými neurónmi člena s najväčšou hodnotou dôveryhodnosti,  $D_j$  je vektor učiaceho signálu k  $i$ -tému vzorku,  $X_j$  je vektor  $j$ -tej vstupnej vzorky a  $\theta^w$  je vektor prahov výstupných neurónov. Učiaci parameter  $\eta$  je adaptívny a počas učenia je jeho hodnota znižovaná kvôli stabilizácii regulárnej časti (ktorá zabezpečuje klasifikáciu na príslušnej oblasti) člena na oblasti ktorá nemôže byť lineárne separovateľná

Ako v pôvodnom algoritme tak aj v upravenom sa menia váhy medzi vstupnými a „confidence“ neurónom v každej jednej iterácii. Tieto zmeny sú uskutočnené podľa nasledovných vzťahov:

$$W_c^i(t+1) = W_c^i(t) - \gamma(t)C^i(t)X_j, \quad (17)$$

$$\theta_c^w(t+1) = \theta_c^w(t) - \gamma(t)\delta(t)C_i(t), \quad (18)$$

$$C^i(t) = 1 - \frac{\exp(\alpha \| O_c^i(t) - O_c^w(t) \|^2)}{\sum_{j=1}^N \exp(\alpha \| O_c^i(t) - O_c^w(t) \|^2)} \quad (19)$$

kde  $W_c^i$  je vektor váh spojení medzi vstupnými neurónmi a neurónom dôveryhodnosti  $i$ -tého člena, kým  $O_c^i$  je hodnota výstupu neurónu dôveryhodnosti  $i$ -tého člena.  $\theta_c^w$  je prah neurónu „confidence“  $i$ -tého člen množiny.  $\alpha$  je kladná konštanta,  $\delta$ ,  $\gamma$  a sú kladné adaptívne učiace parametre.  $C^i$  je relatívna dôveryhodnosť  $i$ -tého člena s ohľadom na dôveryhodnosť každého člena množiny, ktorú môžeme uvažovať ako učiaci signál pre neurón dôveryhodnosti  $i$ -tého člena.  $N$  udáva počet členov v množine.

Umelý učiaci signál  $C^i$  určené pomocou vzťahu (19) zabezpečuje postupné rozmiestnenie členov na celom príznakovom priestore. Hodnota  $C^i$  je sa pohybuje v intervale  $\langle 0,1 \rangle$ . Hodnota blízka jednej znamená, že  $i$ -ty člen je minimálne aktivovaný pri aktuálnych vstupoch a hodnota blízka nule znamená opak, teda  $i$ -tý člen je výnimočne aktivovaný pri aktuálnych vstupoch. Táto vlastnosť je využitá pri adaptácií váh medzi vstupnými neurónmi a „confidence“ neurónom takým spôsobom, že výstupná hodnota „confidence“ neurónu  $i$ -tého člena je znižovaná priamoúmerne s hodnotou  $C^i$ , čo predstavuje nadmierne zníženie aktivácie v prípade, že  $C^i$  je veľké,

teda aktivácia aktuálneho neurónu je nízka v porovnaní s ostatnými v množine. V opačnom prípade, keď regulácia je minimálna,  $C^i$  má nízku hodnotu a teda je dominantným členom množiny pri aktuálnej vstupnej vzorke.

Učiaci parameter  $\gamma$  zabezpečuje reguláciu umelého učiaceho signálu  $C^i$  a zmeny oblastí priradených ku jednotlivým členom. Jeho hodnota blížiac sa k jednej zabezpečuje, aby jednotlivé členy po náhodnej inicializácii sa rozmiestnili v príznakovom priestore a jeho malá hodnota blížiac sa k nule zabezpečuje stabilitu dominancie členov, preto je počas učiaceho procesu postupne znižovaná.

Parameter  $\delta$  slúži na normalizáciu „confidence“ výstupov a zabraňuje k tomu aby v začiatkovej fáze učiaceho procesu mohol prevziať jeden člen množiny dominanciu na celom príznakovom priestore. Počas učenia je znižovaná jeho hodnota a tým aj vplyv na učiaci proces z dôvodu.

#### 5.4 Ideálna štruktúra vzoriek tréningových a testovacích množín

Algoritmus je navrhnutý pre vstupné číselné dáta s pohyblivou desatinnou čiarkou (float) a v ideálnom prípade sú z uzavretého intervalu  $<0;1>$ , ale toto kritérium nie je vyžadované. Výstupné hodnoty, teda identifikátory tried musia nadobudnúť hodnoty z množiny  $\{-1;1\}$  resp.  $\{0;1\}$  a je výhodné triedy klasifikovať podľa dvojkovej sústavy, ako je ukázané v Tab. 3 a druhý spôsob kódovania je ukázané v Tab. 4, v ktorej šírka kódu udáva počet tried a každú triedu reprezentuje jeden prvok kódu, ktorý má hodnotu jedna a ostatné nadobudnú opačné hodnoty (0 alebo -1 podľa interpretácie).

Tab. 3 Kódovanie tried pomocou dvojkovej sústavy

Trieda	Kódovaná trieda
A	00
B	01
C	10
D	11

**Tab. 4 Kódovanie tried so šírkou ich počtu**

Trieda	Kódovaná trieda
A	0001
B	0010
C	0100
D	1000

Počas prvotných testoch algoritmu a konečných experimentoch bolo použité kódovanie druhého typu, prezentované v Tab. 4, ale v prípade keď klasifikujeme do veľkého počtu tried je výhodné použiť kódovanie prvého typu, lebo zabezpečuje menší počet váh znížením šírky kódu jednotlivých tried.

Z charakteristiky kódovania vyplýva možnosť prípadu, keď výstupné hodnoty nezodpovedajú ani jednej očakávanej triede. Preto je zavedená jedna prídavná trieda, do ktorej patria vzorky, ktorých sieť neklasifikoval ani do jednej triedy a v kontingenčných tabuľkách sa objaví ako ďalšia trieda v stĺpci predikovaných tried.

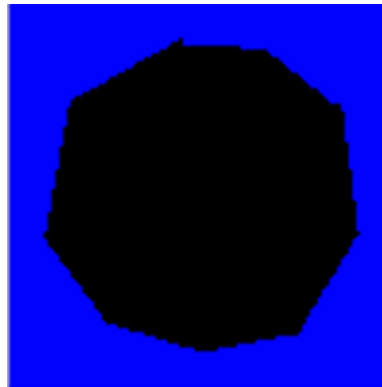
## 5.5 Grafické znázornenie fungovania ELPCL0

Keď to chceš mať tu ukáž aj tréningovú množinu kruh vo štvorci na jednom obrázku. Tiež si daj pozor na farby, ak chceš tlačiť čiernobiely aby bolo vidno rozdiely – aj vyššie. Na znázornenie fungovania algoritmu je použité riešenie problému „kruh vo štvorci“ 10 členným ELPCL0. Na Obr.7 každá farebná oblasť je priradená jednému členovi z množiny a sú vytvárané pretínaním nadrovin, ktoré generujú neuróny dôveryhodností. To znamená, že napríklad oblasť označená červenou farbou predstavuje člen, ktorého neurón dôveryhodnosti má väčšiu hodnotu pri vstupoch z červenej oblasti než ostatné členy množiny a naopak mimo červenej má menšiu.



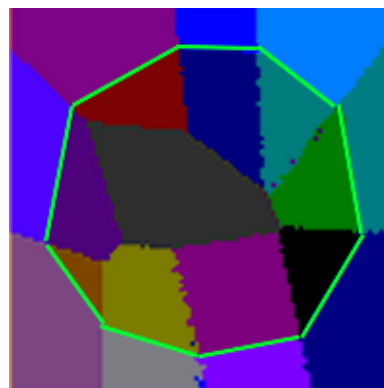
**Obr. 7** Oblasti priradené k jednotlivým členom ELPCL0

Na Obr.8 je vlastný výsledok klasifikácie, kde je možné vidieť známky lineárnej separácie tried.



**Obr. 8** Výsledok klasifikácie s 10 členným ELPCL0

Obr.9 je kombináciu predchádzajúcich dvoch obrázkov a zelenými úsečkami sú na ňom označené priamky, separujúce jednotlivé podoblasti do dvoch tried.



**Obr. 9** Priamky separujúce jednotlivé podoblasti

## 6 Dynamická adaptácia topológie

Cieľom tejto diplomovej práce bol navrhnuť algoritmus na dynamickú adaptáciu topológie ELP. V kapitole 4.2 je popísaná štruktúra člena, celej množiny LP a tiež sú znázornené na **Obr.2** a **Obr.3** z ktorých vyplýva, že topológia každého člena je zadaná štruktúrou vzoriek a preto nie je možné meniť ich topológie. V prípade vstupných vzoriek typu  $X=\{x_1, x_2, \dots, x_N\}$  a výstupných typu  $O=\{o_1, o_2, \dots, o_M\}$  je topológia člena zadaná nasledovne: v asociačnej vrstve sa nachádza N neurónov a na výstupnej M+1, kde neurón M+1 predstavuje výstup dôveryhodnosti.

Vlastná zmena topológie je obmedzená na reguláciu počtu členov množiny, ktorú je možné vykonávať dvomi základnými spôsobmi:

- dekrementovať počet členov
- inkrementovať počet členov

Tretia možnosť, ktorá by znamenala ideálnu cestu je hybridným riešením dvoch základných krokov a vyžaduje ich úspešnú realizáciu.

Dekrementácia počtu členov predstavovala z hľadiska realizácie jednoduchší prípad oproti inkrementácii, implementácia ktorej vyžadovala mnoho experimentov a bola časovo náročná.

### 6.1 Návrh dynamickej adaptácie topológie

Pri návrhu dynamickej adaptácie sa uvažovalo o nasledovných troch možnostiach zmeny topológie ELPCLD

- 1) začať učiaci proces s veľkým počtom členov a postupne dekrementovať podľa potreby
- 2) druhá možnosť predstavuje opačnú taktiku, teda začať s malým počtom členov a počas učenia postupne inkrementovať ich počet
- 3) tretia a najideálnejšia možnosť je hybridné riešenie predchádzajúcich dvoch, čo predstavuje algoritmus s tromi možnými výstupmi:
  - a) inkrementovať počet členov
  - b) dekrementovať počet členov
  - c) nezmeniť počet členov

## 1) Znižovanie počtu členov

### Inicializácia:

Ak predpokladáme, že vieme odhadnúť zložitosť problému táto metóda je výhodná, lebo môžeme odhadnúť počet potrebných členov pre úspešné riešenie problému, ale v prípade že zložitosť nepoznáme môže znamenať problém vlastné určenie inicializačného počtu členov. V tomto prípade môže vyskytnúť taká situácia, keď nami určený počet členov nie je dostatočný pre úspešné zvládnutie učiaceho procesu.

### Vyhodnocovanie počas učenia:

Počas učiaceho procesu musíme, kontrolovať určité príznaky, ako napríklad celkovú chybu, stupeň konkrétnych členov množiny, celkovú stupeň konkrétnych členov množiny.

### Teoretická štruktúra algoritmu:

1. Začiatok učiaceho procesu
2. **Ak** celková chyba klesne pod požadovanú úroveň

**Potom** Uložia sa aktuálne nastavenia a topológia ELP

Určí sa najvhodnejší člen na odstránenie

-Pomocou analýzy úspešnosti členov (daj odkaz na oddiel kde je analýza popísaná)

Po určení sa odoberie člen a pokračuje sa ďalej v učiacom procese (daj odkaz na ktorý bod sa odiaľto skáče)

3. **Ak** v určenom časovom intervale neklesne chyba pod požadovanú úroveň a nebola vykonaná ani jedna zmena v topológií vyhlásime učenie za neúspešne. V prípade, že bol odobratý aspoň jeden člen, tak sa vrátia uložené nastavenia a ukončí sa proces učenia a je považované za neúspešné. toto mi nie je celkom jasné – čo sa stane keď sa učenie vyhlási za neúspešné? Kedy je učenie považované za úspešné? Daj si na tejto časti záležať, to bude oponenta zaujímať najviac. Čo keby tu bola bloková schéma alebo vývoják?

## 2) Zvyšovanie počtu členov

### Inicializácia:

V tomto prípade neznamena problém neznáma náročnosť úlohy a vždy inicializujeme sieť s malým počtom členov. Samozrejme tým pádom sa zvyšuje časová náročnosť učiaceho procesu z dôvodu, možnému dlho trvajúcemu procesu počas ktorej sa dopracuje systém k vhodnému počtu členov. So zmenou začiatočného počtu členov množiny je možné zrýchliť tento proces.

### Vyhodnotenie počas učenia:

V prípade zmeny počtu členov ich zvyšovaním, sleduje sa celková chyba jednotlivých členov, teda to, že ktorý perceptron má na celej trénovacej množine najväčší počet zle klasifikovaných vzoriek. Z toho sa určí, že v oblasti, kde tento člen je dominantný je nedostatočne úspešný a musíme pridať ďalší člen a ak je možné tak takým spôsobom aby sa hneď dostal do tejto oblasti.

### Teoretická štruktúra algoritmu:

1. Začiatok učiaceho procesu
2. **Ak** celková chyba klesne pod požadovanú úroveň  
*Potom* ukončí sa proces učenia a považujeme ho za úspešné.
3. **Ak** v určenom časovom intervale neklesne chyba pod požadovanú úroveň alebo chyba prestane vykazovať klesajúcu tendenciu  
*Potom* Určí sa najvhodnejšia oblasť na pridanie nového člena  
-Pomocou analýzy celkovej chyby jednotlivých členov  
Po určení parametrov nového člena, pridáme ho množiny a pokračujeme v učiacom procese ďalej

## 3) Hybridné riešenie predchádzajúcich dvoch

### Inicializácia:

V prípade použitia hybridnej metódy, by malo byť prakticky jedno, že s akým počtom členom inicializujeme množinu, ak nebude obsahovať dostatočný počet členov na zvládnutie problému pridá sa člen a v opačnom prípade odoberie s čím je možné



dopracovať sa k najideálnejšiemu počtu perceptronov a práve v tom je výhoda takého algoritmu, že nehrozí skorá a nenavrátilná zmena počtu členov.

Vyhodnocovanie počas učenia:

Počas učiaceho algoritmu, v ktorom je zmena počtu členov realizovateľná oboma smermi, je potrebné sledovať viac príznakov učenia a na základe celkovej chyby rozhodovať o kroku, teda vykonať alebo nevykonať zmenu počtu členov a ak sa koná zmena topológie, tak akým smerom. Prípád nevykonať zmenu nastane vtedy, keď po dosiahnutí určitého počtu iterácií celková chyba stále klesá rýchlejšie než je určená hranica a v opačnom prípade nastane zmena. Zmena znížením počtu členov nastane vtedy, keď chyba klesne pod požadovanú úroveň. Výber člena na odstránenie bude vykonané podobným spôsobom ako v prípade 2). Zmena topológie zvýšením počtu sietí sa prejaví vtedy, keď celková chyba po určitom počte iterácií neklesne pod požadovanú úroveň a ani celková chyba nevykazuje dostatočnú zmenu smerom dole. Podľa mojich predpokladov učiaci proces bude úspešný vtedy, keď systém začne oscilovať, teda striedajú sa zmeny zvýšením a znížením počtu členov. Keď sa to nastane, môžeme považovať učiaci proces po kroku zvýšenia počtu členov a celkovej chyby pod požadovanou úrovňou za úspešný koniec učiaceho algoritmu.

Teoretická štruktúra algoritmu:

1. Začneme učiaci proces

2. **Ak** celková chyba klesne pod požadovanú úroveň

**Potom** Do zoznamu zmien uložíme typ zmeny

Skúsime nájsť najvhodnejší člen na odstránenie

-Pomocou analýzy podobnosti váh spojení [odkaz na popis](#)

-Pomocou analýzy podobnosti výstupných hodnôt [odkaz na popis](#) tzv. „confidence“ neurónov

Po určení odoberieme vybraný člen a pokračujeme ďalej v učiacom procese

3. **Ak** v určenom časovom intervale neklesne chyba pod požadovanú úroveň alebo chyba prestane vykazovať klesajúcu tendenciu

**Potom** Do zoznamu zmien uložíme typ zmeny

Uložíme aktuálne nastavenie a topológiu MLP

Skúsime nájsť najvhodnejšiu oblasť na pridanie nového člena

Pomocou analýzy celkovej chyby jednotlivých členov

4. **Ak** v zozname zmien vyskytuje „viackrát“ za sebou kombinácia zmien znížiť-zvýšiť, potom zoberieme hodnoty naposledy uložené a považujeme ich za hodnoty dosiahnuté úspešným procesom učenia

Dekrementácia počtu členov predstavovala z hľadiska realizácie jednoduchší prípad oproti inkrementácie, implementácia ktorej vyžadovala mnoho experimentov a bola časovo náročná.

## 6.2 Časovanie zásahov

Na začiatku sa inicializuje systém s konkrétnym počtom členov, ktorý ak je potrebné po určitom čase mení sa podľa pravidiel popísaných v kapitole aa. Na začiatku učiaceho procesu však nie je vhodné zasahovať do počtu členov, lebo to môže narušiť úspešnosť učiaceho procesu.

Časovanie zásahu je riadené podľa dvoch parametrov, konkrétne podľa priebehu chyby a aktuálneho počtu iterácií. Prvý možný zásah prichádza najskôr po počte iterácií 500, kedy už jednotlivé členy sa rozmiestnili na svoje jedinečné regióny a je možné analyzovať úspešnosť jednotlivých členov na im prislúchajúcich častiach príznakového priestoru. Po analýze sú prístupné údaje o jednotlivých členov s nasledovným obsahom:

$$H_i = \{T, G, B, \bar{C}\} \quad (20)$$

kde  $H_i$  je vektor hodnôt určených analýzou pre  $i$ -ty člen množiny,  $T$  je celkový počet vzoriek trénovacej množiny, ktoré boli priradené k aktuálnemu členu množiny. Hodnota  $G$  predstavuje počet dobre klasifikovaných vzoriek, naopak  $B$  obsahuje počet zle klasifikovaných vzoriek a samozrejme platí, že  $T=G+B$ . Vektor  $\bar{C}$  má rozmer počtu skutočných tried a predstavuje rozloženie priradených tried k aktuálnemu členu  $i$ . Po prekročení požadovaného začiatočného počtu iterácií analýzy a prípadné zmeny sa

vykonajú každých 50 iterácií. Takýto počet iterácií je práve dostatočný na to, aby po inkrementácií nový člen mal šancu uspieť a získať stabilnú dominanciu v oblasti do ktorej bol určený. Samozrejme nie je vhodné zasahovať do množiny keď vykazuje známky úspešného učenia a chyba sa znižuje, preto zásah je povolený len v prípade, keď chyba medzi dvomi bodmi zásahu neklesla aspoň o 10%.

### 6.3 Dekrementácia počtu členov množiny

Dekrementácia počtu členov množiny je uskutočnená po určitom počte iterácií, ktorá je stanovená heuristiky na základe experimentov, testov a predstavuje hodnotu 50 iterácií. Pred samotným krokom sa vykoná analýza úspešnosti členov a vygeneruje vektor  $H$  pre každý člen ELP. Následne sa odstránia členy, ktoré neklasifikujú ani jednu vzorku, teda majú nulovú úspešnosť z hľadiska dôveryhodnosti a hodnota  $T$  sa rovná nule.

Ďalej sa odstránia členy, ktoré klasifikujú v podoblasti, v ktorej má zastúpenie len jedna trieda a s tým oblasť odstráneného člena sa rozdelí medzi susednými členmi. Takéto členy sa určujú pomocou vektora  $\bar{C}$ , nasledovným spôsobom:

Ak  $\bar{C}$  obsahuje len jeden prvok s hodnotou väčšou ako nula, to znamená, že len jedna trieda je zastúpená v podoblasti a preto sa člen odstráni.

### 6.4 Inkrementácia počtu členov množiny

Počas učiaceho procesu je pridanie nového člena predstavuje komplexnejšiu úlohu, lebo po určitej počte iterácií by mal uspieť a vyhradiť si svoju oblasť na ktorej bude expertom.

Pri inicializácii všetky váhy členov sú určené náhodne, ale v procese učenia už nie je možné váhy nového člena inicializovať náhodou, lebo v systéme je už vytvorená určitá rovnováha medzi členmi (rovnováha sa týka váh medzi asociačnou vrstvou a neurónom dôveryhodnosti) a náhodné generovanie váh takmer 100%-nou určitosťou by spôsobil jednu z nasledovných dvoch prípadov:

- Člen by vôbec neuspel ani pri jednej vstupnej vzorke – lepší prípad – učiaci proces by mohol pokračovať a pri ďalšom zásahu by sa člen odstránil.

- Člen by prevzal dominanciu na celej trénovacej množine a tým by zmaril učiaci proces.

Kvôli zabráneniu predchádzajúcich javov bol navrhnutý systém na pridávanie členov, ktorý je popísaný v nasledovných odsekoch.

Pred inkrementáciou počtu členov je určená počet nových členov podľa nasledujúceho vzťahu:

$$P = 1 + \left\lceil 10 \frac{E}{N_p} \right\rceil \quad (21)$$

kde  $P$  je počet nových členov ktoré sa pridajú do množiny počas aktuálneho zásahu,  $E$  je aktuálna chyba, ktorá predstavuje počet aktuálne zle klasifikovaných vzoriek a  $N_p$  je celkový počet vzoriek v trénovacej množine. Hodnota  $P$ , teda počet pridaných nových neurónov sa znižuje spoločne s chybou.

Po určení kvantity budúcich nových členov sa zaháji procedúra ich zasadenia do existujúcej množiny s veľkým dôrazom na nastavení váh medzi asociačnou vrstvou a neurónom dôveryhodnosti. Procedúra pozostáva z nasledovných krokov:

1. Výber vhodného cieľového regiónu
2. Výpočet váh pre nového člena
3. Pridanie člena do množiny

#### 6.4.1 Výber najvhodnejšieho cieľového regiónu

Vhodný cieľový región sa určuje podľa úspešnosti už existujúceho a na tomto oblasti dominantného člena ELP. Konkrétne pomocou jeho váh spojujúce asociačnú vrstvu s neurónom dôveryhodnosti a jeho prahom, ktoré predstavujú jednu rovinu, keď uvažujeme o dvojrozmernom príznakovom priestore. Vstupné vzorky pri ktorých neurón dôveryhodnosti aktuálneho člena bol aktivovaný najsilnejšie predstavujú konvexnú oblasť na rovine a cieľom je rozdeliť túto oblasť na dve časti. Na jednej ostáva dominantným pôvodný člen a na druhej nový. Navrhnutá metóda, ktorá zabezpečuje takéto zasadenie nového člena je popísaná v kapitole aa. Z dôvodu, že do oblasti s najnižšou úspešnosťou je výhodné usadiť nového člena je zvolený člen s najnižšou úspešnosťou, teda s najvyšším počtom zle klasifikovaných vzoriek a tým je región dominancie.

### 6.4.2 Výpočet váh nového člena ELP

Váhy nového člena sa určujú pomocou váh pôvodného dominantného člena v cieľovej oblasti a to nasledovným spôsobom:

1. Určí sa geometrický stred vstupných vzoriek priradených k pôvodnému členu
2. Určí sa hodnota neurónu dôveryhodnosti v určenom geometrickom strede
3. Určí sa nová rovina, ktorá pretína pôvodnú rovinu v určenom geometrickom strede

Geometrický stred sa určuje **sumarizáciou** **sumáciou** hodnôt vstupných vzoriek  $X_s = \{x_1, x_1, \dots, x_{N_m}\}$ , ktoré aktivujú pôvodný člen, do vektora  $X_s = \sum X_i$ ,  $X_s = \{x_{s1}, x_{s1}, \dots, x_{sN_m}\}$  a ich následným delením počtom sčítaných vzoriek  $S$ . Konečný vektor sa určuje podľa nasledovného vzťahu:

$$X_g = \left\{ \frac{x_{s1}}{S}, \frac{x_{s1}}{S}, \dots, \frac{x_{sN_m}}{S} \right\} \quad (22)$$

Kde  $X_g$  je vektor hodnôt určiach geometrický stred vstupných vzoriek priradených k pôvodnému členu resp. oblasti,  $x_{s1}$  je suma prvého člena vektoru vzoriek,  $S$  je celkový počet vstupných vzoriek priradených k pôvodnému členu a  $N_m$  je počet prvkov vo vstupnej vzorke.

Nové váhy medzi asociatívnu vrstvou a neurónom dôveryhodnosti sa inicializujú nasledovným spôsobom:

$$W_c^{new}(t) = W_c^{orig}(t) \pm (N_A)^{-2}, \quad (23)$$

kde  $W_c^{new}$  predstavuje váhy nového člena,  $W_c^{orig}$  váhy pôvodného dominantného člena v cieľovej oblasti a  $N_A$  aktuálny počet členov v ELP.

V prípade inicializácie prahu je využitý vypočítaný geometrický stred vstupných vzoriek a to nasledovným spôsobom:

$$\theta_c^{new}(t) = O_c^{orig}(t) - O_c^{temp}(t), \quad (24)$$

$$O_c^{temp}(t) = \sum_{k=1}^{N_m} v_{kc}^{new} x_{sk} \quad (25)$$

Kde  $\theta_c^{new}$  je prah nového člena,  $O_c^{orig}$  je hodnota výstupu neurónu dôveryhodnosti pôvodného a  $O_c^{temp}(t)$  nového člena ale bez prahu,  $v_{kc}^{new}$  je váha spojenia medzi  $k$ -tým vstupným neurónom a neurónom dôveryhodnosti nového člena a  $x_{sk}$  je  $k$ -tým prvkom vektoru obsahujúceho koordináty geometrického stredu.

## 7 Experimenty

V tejto kapitole je poskytnutý prehľad experimentov vykonaných pomocou navrhutej neurónovej siete ELPCLO s dynamickou adaptáciou topológie a je rozdelený na nasledovné dve partície:

### 1. Experimenty na umelých dátach (Kapitola 7.1)

V ktorej algoritmus navrhnutý v Kapitole 5 je testovaný s umelými dátami, konkrétne na probléme kruhu vo štvorci a dvojitej špirály a následne výsledky porovnané s výkonom NN MF ARTMAP a štandardným Backpropom (BP)

### 2. Experimenty na reálnych dátach (Kapitola 7.2)

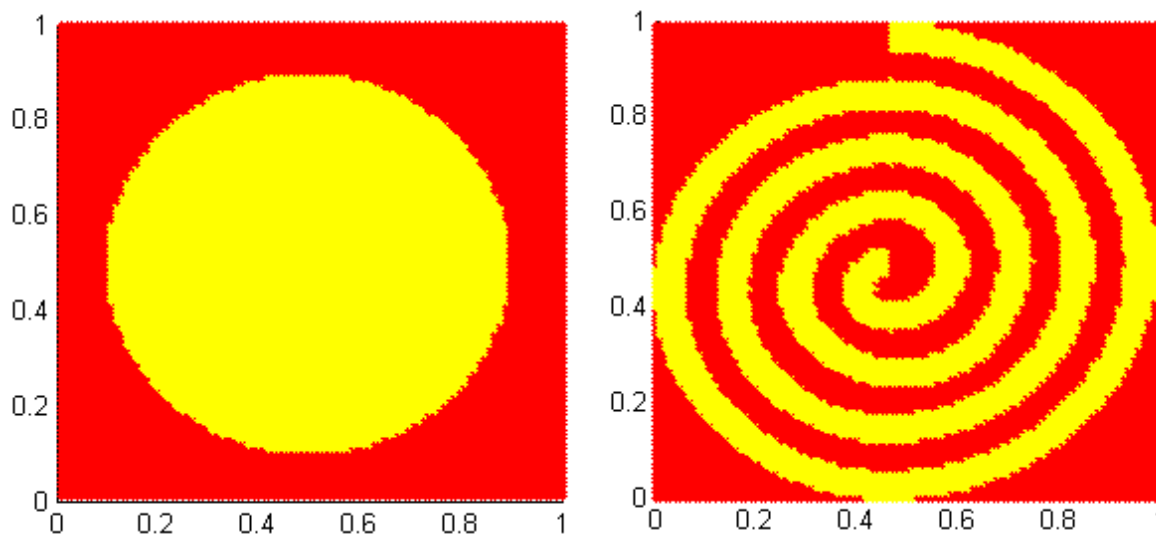
Reálne dáta predstavovali fotky zemského povrchu z veľkej výšky satelitné snímky človeče ☺ a podobne ako v predchádzajúcom prípade výsledky boli porovnané s algoritmami MF ARTMAP a štandard BP.

Výsledky experimentov sú zhrnuté v kontingenčných tabuľkách a vizualizované pomocou prílužných obrázkov.

## 7.1 Experimenty na umelých dátach

### 7.1.1 Trénovacie a testovacie množiny

Trénovacia množina predstavuje v prípade kruhu vo štvorci množina s počtom prvkov 1000 a v prípade špirály počtom 1250. Ich rozloženie je znázornené na Obr. 7. [zlý odkaz](#)



**Obr. 10** Vizualizácia trénovacích množín umelých dát **tu je zobrazená celá množina, nie len trénovačka - pozor**

Testovaciu množinu tvorí 10000 prvkov v oboch prípadoch a sú rovnomerne rozložené v príznakovom priestore.

Kruh vo štvorci a špirála slúžia na overenie funkčnosti algoritmu v riešení dvojrozmernej dichotomickej klasifikácie. Vstupné vzorky v oboch prípadoch sú dvojrozmerné a nadobudnú hodnoty z intervalu  $\langle 0;1 \rangle$  výstupy sú tiež dvojrozmerné a pre triedu A majú označenie  $\{1,-1\}$  a pre triedu B  $\{-1,1\}$ .

### 7.1.2 Nastavenie parametrov pre dáta „kruh vo štvorci“

Nastavenia učiacich parametrov počas realizácie experimentov na umelých dátach „kruh vo štvorci“ jednotlivých algoritmov sú nasledovné:

#### **ELPCLO s DAT**

Počet členov množiny pri inicializácii: 2

Inicializačná hodnota učiaceho parametru regulárnych váh : 0.1

Inicializačná hodnota učiaceho parametru váh dôveryhodnosti : 0.05

Maximálny počet iterácií: 1500

Cieľová úspešnosť: 100%

#### **MFARTMAP**



### **Standard BP**

Hodnoty boli stanovené podľa skúseností počas experimentov a zabezpečujú najefektívnejšie učenie. **Ak si to robil sám, môžeš to sformulovať trochu „vedeckejšie“ ak si to prevzal, kľudne daj odkaz na to odkiaľ.**

#### **7.1.3 Nastavenie parametrov pre dáta „dvojitá špirála“**

Nastavenia učiacich parametrov počas realizácie experimentov na umelých dátach „dvojitá špirála“ jednotlivých algoritmov sú nasledovné:

##### **ELPCLO s DAT**

Počet členov množiny pri inicializácii: 2

Inicializačná hodnota učiaceho parametru regulárnych váh : 0.1

Inicializačná hodnota učiaceho parametru váh dôveryhodnosti : 0.05

Maximálny počet iterácií: 5000

Cieľová úspešnosť: 100%

##### **MFARTMAP**

### **Standard BP**

Hodnoty boli stanovené podľa skúseností počas experimentov a zabezpečujú najefektívnejšie učenie.

#### **7.1.4 Experimenty na dátach „kruh vo štvorci“**

Experimenty priniesli očakávaný výsledok ohľadom na jednoduchosť riešenej úlohy a slúžili predovšetkým na overenie funkcionality algoritmov. Konkrétne výsledky sú prezentované jednotlivo podľa metód v nasledovnej časti tejto podkapitoly. Na Obr.aa, Obr.bb a Obr.cc farebné označenie tried je nasledovné: A - žltá, B – červená, C' – modrá farba.

##### **ELPCLO s DAT**

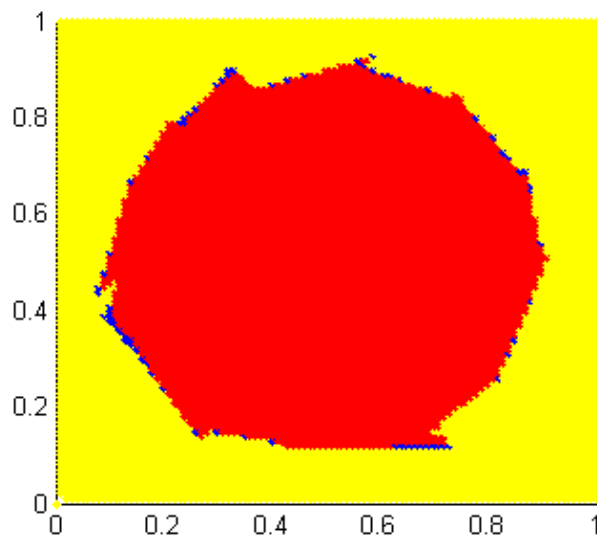
Experiment s touto metódou priniesol očakávaný výsledok, po začiatku učiaceho procesu chyba začala klesať a po dosiahnutí iterácie 1471 bola prerušená pred dosiahnutím maximálneho počtu iterácií, kvôli objaveniu „ideálnej topológie“ (**máš niekde definované čo je ideálna topológia? Odporúčam používať výraz „optimálna“**) a priblížením sa k najmenšej možnej chybe, ktorá v tomto prípade bola nula. To znamená, že každý prvok trénovacej množiny bol správne klasifikovaný,

teda úspešnosť bola stopercentná. Na testovacej množine samozrejme úspešnosť nebola stopercentná, bola dosiahnutá 97,53% a v kontingenčnej tabuľky Tab.5 sú znázornené konkrétne detaily výsledku. V časti testovacej množiny tabuľky Tab.5 niektoré vzorky do triedy C', ktorá obsahuje vzorky, ktoré algoritmus nevedel zaradiť ani do jednej zo skutočných tried. Tieto vzorky na Obr.8 sú zobrazené modrou farbou. Triedu A predstavuje žltá a triedu B červená farba.

**Tab. 5** Experimentálne výsledky – ELPCLO s DAT– „kruh vo štvorci“

Trénovacia množina					Testovacia množina				
		Skutočná trieda		UA (%)			Skutočná trieda		UA (%)
		A	B				A	B	
Predikovaná trieda	A'	492	0	100,00%	Predikovaná trieda	A'	4845	66	98,66%
	B'	0	508	100,00%		B'	109	4908	97,83%
	C'	0	0	—		C'	49	23	—
PA (%)		100,00%	100,00%		PA (%)		96,84%	98,22%	
Celková úspešnosť		<b>100,00%</b>			Celková úspešnosť		<b>97,53%</b>		

Na Obr.8, ktorá je grafickým znázornením klasifikácie testovacej množiny vzoriek, sú zreteľné znaky lineárnej klasifikácie, hranice medzi triedami pozostávajú z úsečiek, kým pri metóde MFARTMAP sú zreteľné jednotlivé zhľuky a pri Standard BP plynulosť kriviek



**Obr. 11** Vizualizácia výsledkov na trénovacej množine - ELPCLO s DAT

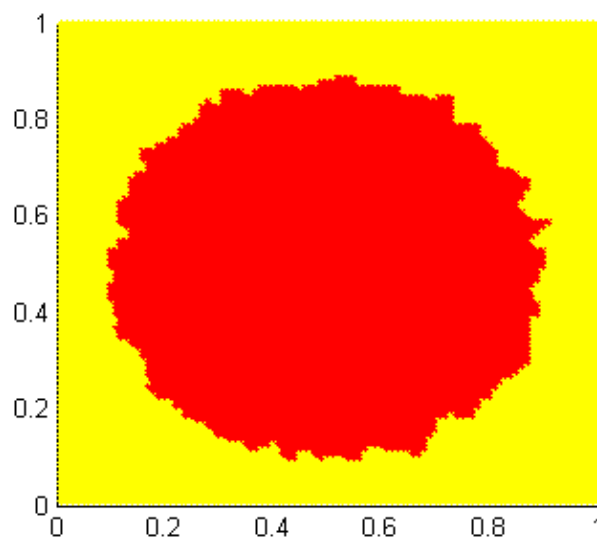
## MFARTMAP

Pri tejto metóde bol dosiahnutý podobný výsledok, ako pri ELPCPO s DAT. Na testovacej množine úspešnosť predstavuje 97,73%. Detaily sú k dispozícii v kontingenčnej tabuľke Tab.6.

**Tab. 6** Experimentálne výsledky – MFARTMAP– „kruh vo štvorci“

Testovacia množina				
		Skutočná trieda		UA (%)
		A	B	
Predikovaná trieda	A'	4895	119	97,63%
	B'	108	4878	97,83%
PA (%)		97,84%	97,62%	
Celková úspešnosť		<b>97,73%</b>		

Grafické znázornenie výsledku klasifikácie na testovacej množine pomocou MFARTMAP je na Obr.9, kde žltou farbou sú označené vzorky klasifikované do triedy A a červenou do triedy B



**Obr. 12** Vizualizácia výsledkov na trénovacej množine – MFARTMAP

### 7.1.5 Experimenty na dátach „dvojitá špirála“

#### ELPCLO s DAT

**Tab. 7** Experimentálne výsledky – ELPCLO s DAT– „dvojitá špirála“

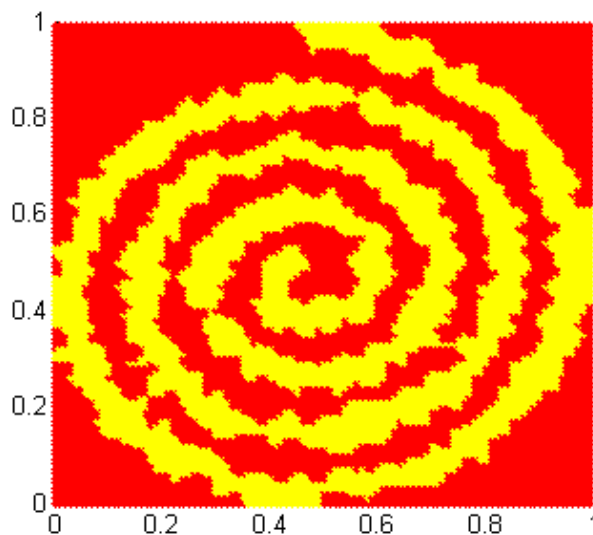
Trénovacia množina			Testovacia množina		
	Skutočná trieda	UA (%)		Skutočná trieda	UA (%)

		A	B				A	B	
Predikovaná trieda	A'	621	61	91,06%	Predikovaná trieda	A'	4446	1090	80,31%
	B'	75	491	86,75%		B'	1137	3144	73,44%
	C'	0	2	--		C'	106	77	
PA (%)		89,22%	88,63%		PA (%)		78,15%	72,93%	
Celková úspešnosť		<b>88,96%</b>			Celková úspešnosť		<b>75,90%</b>		

### MFARTMAP

Tab. 8 Experimentálne výsledky – MFARTMAP – „dvojitá špirála“

Testovacia množina				
		Aktuálna trieda		UA (%)
		A	B	
Predikovaná trieda	A'	3812	564	87,11%
	B'	499	5125	91,13%
PA (%)		88,42%	90,09%	
Celková úspešnosť		<b>89,37%</b>		



Obr. 13 Vizualizácia výsledkov na trénovacej množine - MFARTMAP

## 7.2 Experimenty na reálnych dátach

### 7.2.1 Použité dáta

Klasifikačné vlastnosti algoritmov boli overené v klasifikácii charakteristických regiónov na satelitných snímkoch z diaľkového prieskumu Zeme. Experimenty boli realizované na obrázku mesta Košíc a jeho blízkeho okolia, ktorého dátová množina pozostáva z 775x475 (368125) sedemrozmerných vzoriek ktoré reprezentujú jednotlivé nasnímané spektrá. Každá vzorka reprezentuje jeden pixel (obrazový bod) a z nich

expertom bolo klasifikovaných 6331 pixelov do siedmich tematických kategórii, ktoré sú znázornené na Obr. aa. Na obrázku sú definované triedy uvádzané v Tab.5.

**Tab. 9 Popis tried v reálnych dát**

<b>Trieda</b>	<b>Popis triedy</b>
<b>A</b>	Územie mesta
<b>B</b>	Neúrodné oblasti
<b>C</b>	Krovie
<b>D</b>	Poľnohospodárske oblasti
<b>E</b>	Lúky
<b>F</b>	Lesy
<b>G</b>	Voda

Expertom klasifikovaná množina dát bola rozdelená na tréningovú a testovaciu množinu, pričom tréningová množina pozostáva z 3164 vzoriek a testovacia množina z 3167.



**Obr. 14 Oblasti ohraničené bielou čiarou boli označené expertom**

Expertom klasifikovaná množina vzoriek bola náhodne rozdelená na dve časti, prvá predstavuje tréningovú množinu a druhá testovaciu. Rozdelenie vzoriek je znázornené v Tab. Aa.

Tab. 10 Zastúpenie tried v tréningovej a testovacej množine

Trieda	Tréningová množina	Testovacia množina
<b>A</b>	155	156
<b>B</b>	483	484
<b>C</b>	206	206
<b>D</b>	902	902
<b>E</b>	170	171
<b>F</b>	1174	1174
<b>G</b>	74	74
$\Sigma$	3164	3167

### 7.2.2 Nastavenie parametrov metód

Nastavenia učiacich parametrov počas realizácie experimentov na reálnych dátach jednotlivých algoritmov boli nasledovné:

#### **ELPCLO s DAT**

Počet členov množiny pri inicializácii: 2

Inicializačná hodnota učiaceho parametru regulárnych váh : 0.1

Inicializačná hodnota učiaceho parametru váh dôveryhodnosti : 0.05

Maximálny počet iterácií: 5000

Cieľová úspešnosť: 100%

#### **MFARTMAP**

#### **Standard BP**

Hodnoty boli stanovené podľa skúseností počas experimentov a zabezpečujú najefektívnejšie učenie.

**7.2.3 Experiment s metódou ELPCLO s DAT**

**7.2.4 Experiment s metódou MFARTMAP**

**7.2.5 Experiment s metódou Standard BP**

**7.3 Vyhodnotenie experimentov**

## 8 Záver

Záver by mal zachytiť jasnú a presnú prezentáciu dedukcií vychádzajúcich z jadra práce. Musí byť vecnou sumarizáciou vlastného prínosu alebo pohľadu na riešenú problematiku. Zahrnúť možno aj kvantitatívne údaje, ale podrobnosti by sa nemali uvádzať. Záver nemá obsahovať nič, čo nie je v texte práce a musí nadväzovať na úvahy a argumenty v texte práce.

V závere je vhodné poukázať na ďalšie otvorené (doteraz nevyriešené) problémy, ktorým je vhodné venovať pozornosť a ktoré presahujú odporúčaný rozsah diplomovej práce. Odporúčané sú popisy ďalších navrhovaných aktivít, ktoré priamo vyplývajú zo záverov alebo skúseností získaných v priebehu spracovania práce.



## Zoznam použitej literatúry

- [1] SINČÁK, P., ANDREJKOVÁ, G.: Neurónové siete (Inžiniersky prístup) 1. diel, Elfa, 1996
- [2] Pitoyo Hartono, Shuji Hasimoto, Ensemble of Linear Perceptrons with Confidence Level Output, 4th International Conference on Hybrid Intelligent Systems, 5-8 December 2004, Kitakiushu, Japan, pages 186-191, IEEE Computer Society, 2005, ISBN 0-7695-2291-2
- [3] Hara, Kazuyuki - Okada, Masato: Ensemble learning of linear perceptron; Online learning theory. In: eprint arXiv:cond-mat/0402069, 02.2004, 2004cond.mat..2069H
- [4] Jordan, M., Jacobs, R.(1994), "Hierarchical Mixture of Experts and EM Algorithm, Neural Computation, Vol.6, No. 2, pp. 181-214.
- [5] Mangasarian, O.L., Wolberg, W.H. (1990), Cancer Diagnosis via linear programming, SIAM News, Vol. 23, No. 5, pp. 1-18.
- [6] Baxt, W. (1992), "Improving accuracy of artificial neural network using multiple differently trained networks", Neural Computation, Vol. 4, pp. 772-780.
- [7] Hashem, S. (1996), "Optimal linear combination of neural networks", Neural Networks, Vol. 10, No. 4, pp. 299-313.
- [8] Sharkey, A. (1996), "On combining artificial neural nets", Connection Science, Vol. 9, Nos. 3 and 4, pp. 299-313.
- [9] Freund, Y. (1995), "Boosting a weak learning algorithm by majority", Information and Computation, Vol. 121, No.2, pp.256-285.
- [10] Freund, Y., Schapire, R.E. (1999), "A short introduction to boosting", Journal of Japanese Society for Artificial Intelligence, Vol. 14, No.5, pp. 771-780.

## **Prílohy**

Príloha A: Používateľská príručka

Príloha B: Systémová príručka

Príloha C: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe.

## Curriculum vitae

Táto časť je nepovinná. Autor tu môže uviesť svoje biografické údaje, údaje o záujmoch, účasti na projektoch, účasti na súťažiach, získané ocenenia, zahraničné pobyty na praxi, domácu prax, publikácie a pod.