VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

KOHONENOVA SÍŤ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MICHAL KAŇA
AUTHOR

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

KOHONENOVA SÍŤ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MICHAL KAŇA
AUTHOR

VEDOUCÍ PRÁCE doc. Ing. VÁCLAV JIRSÍK, CSc.
SUPERVISOR

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student:	Michal Kaňa	***ID:***	47849
Ročník:	3	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Kohonenova síť

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s problematikou samoorganizujících neuronových sítí.
2. Proveďte rešerši aplikací pro Kohonenovou síť.
3. Na zvolené tréninkové množině zkoumejte vliv počátečného nastavení vah na konečnou "pozici" vítězných neuronů.
4. Dosažené výsledky zhodnoťte.

DOPORUČENÁ LITERATURA:

Šíma J., Neruda R.: Teoretické otázky neuronových sítí. Matfyzpress, Praha 1996

Termín zadání:	7.2.2011	***Termín odevzdání:***	30.5.2011

Vedoucí práce: doc. Ing. Václav Jirsík, CSc.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

V bakalářské práci je stručně popsána problematika umělých neuronových sítí, konkrétně poté samoorganizujících neuronových sítí a jejich využití. Práce se hlouběji věnuje problematice Kohonenovy samoorganizující sítě, popisem principu jejího učení a aplikací v praxi. V praktické části se práce zabývá problematikou nastavení počatečních vah neuronů v Kohonenově síti a jejich vlivem na konečnou pozici vítězného neuronu. U zvolené tréninkové množiny tento vliv demonstruje pomocí programového prostředí MATLAB.

KLÍČOVÁ SLOVA

Kohonenova síť, SOM, Kohonenovy mapy, ART, MATLAB, samoorganizující se sítě

ABSTRACT

The problems with artificial neuron systems, or, more precisely, with self-organising neuron systems and their usage, have been concisely described in this Bachelor's thesis. The thesis is more deeply concerned with the Kohonen self-organising system and describes the principle of its study and programmes for its simulation. The practical part of the thesis concerns the problem of regulating initial neuron weights in the Kohonen system and their effect upon the final position of the surviving neuron. This effect is demonstrated in the selected experiment set with the help of the programme MATLAB.

KEYWORDS

Kohonen Self-Orgamizing maps, SOM, ART, MATLAB, Self-Organizing Network

KAŇA, Michal *Kohonenova síť*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřící techniky, 2011. 41 s. Vedoucí práce byl doc. Ing. Václav Jirsík, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Kohonenova síť“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno ....

(podpis autora)

Na tomto místě bych rád poděkoval doc. Ing. Václavu Jirsíkovi, CSc. za vedení bakalářské práce a také za jeho podporu, trpělivost, rady, inspiraci a diskuze nejen při vypracování této bakalářské práce.

OBSAH

Úvod **9**

1 Umělé neuronové sítě **10**
1.1 Umělá neuronová síť 10
1.2 Paradigma neuronových sítí 10
1.3 Neuron 11
1.4 Organizační dynamika 12
1.4.1 Aktivní dynamika 13
1.4.2 Adaptivní dynamika 13
1.5 Učící algoritmus 13

2 Samoorganizující neuronové sítě **14**
2.1 Vektorová kvantizace 14
2.2 Kohonenovo učení 15
2.3 Neuronová síť ART 16
2.3.1 Popis funkce sítě ART 17

3 Kohonenova síť **18**
3.1 Učení Kohonenovy síťě 19
3.2 Vybavovací fáze 22

4 Rešerše aplikací Kohonenovy sítě **23**
4.1 Diagnostika neurologických onemocnění u dětí 23
4.2 Využití samoorganizující se neuronové sítě pro hodnocení chování spotřebitelů 24
4.3 Získávání znalostí z textu 24
4.4 Modelování bonity obcí pomocí Kohonenových samoorganizujících map a LVQ 25

5 Popis metodiky řešení problému **26**
5.1 Klasifikace problému 26
5.1.1 Tréninková množina 26
5.2 Výběr vhodného programu 26
5.3 MATLAB 26
5.4 Program 27
5.5 Popis programu 27
5.5.1 Funkce nnt2som 27
5.5.2 Funkce Train 28

5.5.3 Funkce Plot, Plotsom 28
5.6 Ukázka funkce programu 28

6 Ověření úkolu v zadání **31**
6.1 Změna vah jednoho neuronu 31
6.2 Síť 3x3 a 5x5 31
6.2.1 Síť 3x3 s volbou okolí=1 32
6.2.2 Síť 3x3 s volbou okolí=0 32
6.2.3 Síť 5x5 s volbou okolí=1 33
6.2.4 Síť 5x5 s volbou okolí=0 34
6.2.5 Síť 3x3, rozmezí vah mimo střed s volbou okolí=0 34
6.2.6 Síť 10x10 s volbou okolí=0 35

7 Závěr **36**

Literatura **37**

Seznam symbolů, veličin a zkratek **39**

Seznam příloh **40**

A Program **41**

ÚVOD

Umělá inteligence je otevřenou otázkou poznání již několik desítek let. Na její vývoj lze pohlížet jako na historický vývoj několika disciplín zároveň: informatiky, neurologie, kybernetiky a mnoha dalších nepomíjeje.

Neuronová síť je jedním z výpočetních modelů používaných právě v oblasti umělé inteligence. Za počátek vzniku oboru umělých neuronových sítí (UNS) je považována práce Warrena McCullocha a Waltera Pittse z roku 1943, kteří vytvořili první matematický model neuronu.

Pro UNS je příznačná inspirace architekturou lidského mozku. Na jejím základě je UNS schopna vyřešit problém, aniž by jí byl předložen způsob jeho řešení. Na řešení konkrétního problému je na rozdíl od jiných algoritmů učena, a to vystavením velkému množství dat, čímž směřuje k odhalení zákonitostí a souvislostí v těchto datech. Jedna ze samoorganizující se, respektive "samoučící se", UNS bývá nazývána podle svého tvůrce Kohonenova síť. Pro pracovníky a odborníky v různých oblastech je dnes využití UNS široké. Nezbytnou součástí jsou v mnoha automatizovaných provozech, lékařství, ekonomice, vědě a dalších odvětvích.

Bakalářská práce se zabývá problematikou samoorganizujících neuronových sítí, konkrétně tedy Kohonenovou sítí. Jejím cílem je u zvolené tréninkové množiny zkoumat vliv počátečního nastavení vah na konečnou "pozici" vítězných neuronů a z něj vyvodit zjištění.

První kapitola shrnuje poznatky o umělých neuronových sítích, jejich využitích, rozdělení a obecné charakteristice.

Následuje druhá kapitola pojednávající o samoorganizujících neuronových sítích, jejich typech a způsobech jejich učení.

Kapitola třetí předkládá blíže princip učení Kohonenovy sítě.

Poslední kapitolou dokončující vhled do problematiky neuronových sítí je kapitola čtvrtá týkající se konkrétních aplikací Kohonenovy sítě v širokém použití.

Praktická část této práce je započata kapitolou číslo pět, jež nastiňuje způsob řešení zadání práce zvolenou metodikou.

Popis výsledků práce je zanesen s komentářem do šesté kapitoly.

Závěr práce, tedy kapitola číslo sedm, získané výsledky shrnuje v konkrétní i obecné poznání.

1 UMĚLÉ NEURONOVÉ SÍTĚ

1.1 Umělá neuronová síť

Umělá neuronová síť se skládá z formálních neuronů, které jsou vzájemně propojené tak, že výstup neuronu je vstupem obecně více neuronů. Počet neuronů a jejich vzájemné propojení v síti určuje topologii neuronové sítě. Z hlediska využití jsou rozlišovány neurony v síti na vstupní, pracovní (skryté, mezilehlé) a výstupní. Zpracování a šíření informací v síti umožňují neurony změnou stavu vyskytující se mezi vstupními a výstupními neurony. Synaptické váhy všech spojů určují konfiguraci sítě a stavy všech neuronů určují stav neuronové sítě. Neuronová síť se během jednotlivých kroků mění, tedy dochází ke změnám stavu neuronů, adaptaci vah a ke změnám propojení jednotlivých neuronů. Tyto změny určují celkovou dynamiku neuronové sítě [8].

Využití neuronových sítí

Oblast uplatnění neuronových sítí je široká a nabývá stále na významu.

- Zpracování řeči
- Úprava zvuku
- Zpracování obrazu
- Odstranění rušení
- Automatické třídění dat
- Hledání podobných znaků
- Problémy aproximace funkcí
- Klasifikace do tříd
- Komprese dat

1.2 Paradigma neuronových sítí

Paradigma neuronové sítě představuje popis UNS, ať už z pohledu topologie, metody učení nebo metodiky využití [18].

- Topologie - údaje o konfiguraci UNS, počet vstupních uzlů, počet vrstev výkonných neuronů, počet prvků výstupního vektoru (počet výstupních tříd), počet a vzájemné propojení neuronů v jednotlivých vrstvách, typy přenosových funkcí a jejich parametry [18].
- Metoda učení - s učitelem, bez učitele.
- Způsob využití naučené (natrénované) UNS: způsob testování naučené sítě, kvalita a přesnost klasifikace neznámých, nebo i poškozených předloh.

V současné době existuje již velký počet paradigmat UNS. Mnohá z nich jsou jen zdokonalenými modifikacemi dosud známých principů [18].

Přehled paradigmat UNS z pohledu topologie a metod učení:

Dopředné (feedforward)

- lineární - ADALINE, lineární perceptron
- nelineární
  - s dohledem - mnohovrstvový perceptron s BP
  - bez dohledu - Kohonenovy samoorganizující mapy, Counterpropagation

Zpětnovazební (feedback)

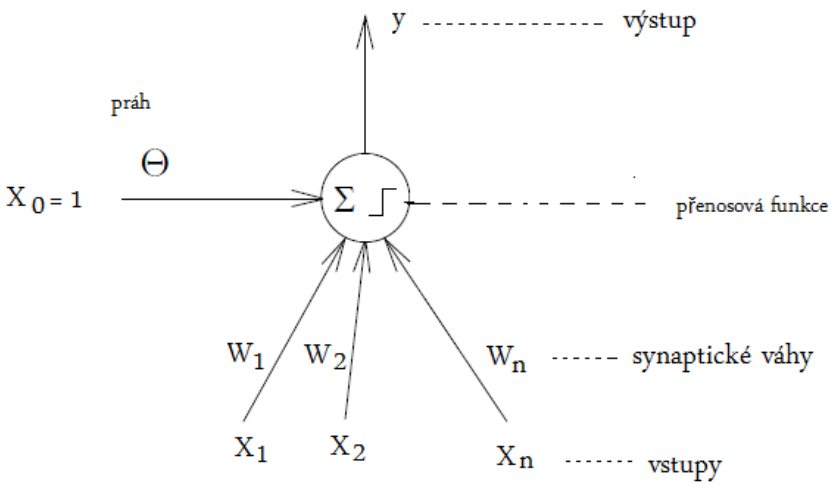
- Hopfieldovi rekurentní sítě -asociativní paměti - CAM
- obousměrné asociativní paměti - BAM
- adaptivní resonanční teorie - ART

1.3 Neuron

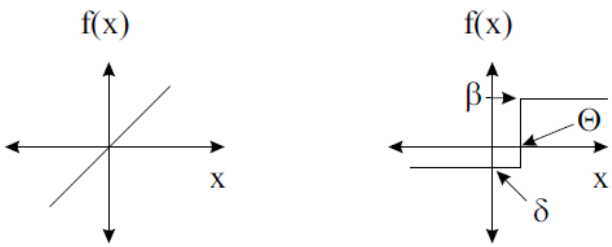
Základem matematického modelu neuronové sítě je formální neuron (Obr. 1.1), který představuje zjednodušené matematické vyjádření neurofiziologického neuronu.Podle povahy vstupních a výstupních dat jsou neurony děleny na binární a spojité. Formální neuron má $nx_1, ...x_n$ vstupů. Vstupy jsou ohodnoceny odpovídajícími obecně reálnými synaptickými vahami $w_1, ...w_n$, které určují jejich propustnost. Synaptické váhy mohou mít i zápornou hodnotu, čímž se vyjadřuje jejich inhibiční charakter. Neuron zpracovává své vstupy na základě vztahu [8].

$$y = f(\sum_{i=1}^{n} w_i x_i + \Theta) \tag{1.1}$$

- Θ - prah neuronu
- x_i - vstupy neuronu
- w_i - synaptické váhy neuronu
- $\int$- přenosová funkce neuronu



Obr. 1.1: Formální neuron



Obr. 1.2: Lineární a skoková funkce [2]

1.4 Organizační dynamika

Organizační dynamika specifikuje architekturu neuronové sítě a její případnou změnu, kdy během adaptavního režimu dochází, v případě potřeby, k rozšíření o další neurony a příslušné spoje. Existují dva typy architektury [8]:

Dopředné sítě (acyklické) - v těchto sítích se signál šíří pouze jedním směrem, a to od vstupu směrem k výstupu sítě. Propojení neuronů mezi sebou nevytváří žádný uzavřený cyklus.

Rekurentní sítě (cyklické) - tento typ sítí se vyznačuje možností šíření signálu v přímém směru, ale i zpětně a propojení neuronů vytváří orientovanou cyklickou strukturu.

1.4.1 Aktivní dynamika

Aktivní dynamika specifikuje počáteční stav neuronové sítě a její změny v čase při pevné topologii a konfiguraci. Během aktivního režimu jsou nastaveny stavy vstupních neuronů na vstup sítě a zbylé neurony jsou v počátečním stavu. Vstupní prostor sítě (stavový prostor) tedy tvoří všechny vstupy sítě. Vlastní výpočet probíhá po inicializaci stavu sítě [8].

1.4.2 Adaptivní dynamika

Adaptivní dynamika specifikuje počáteční konfiguraci sítě a určuje způsob změny vah v síti. Všechny možné konfigurace tvoří váhový prostor neuronové sítě. Během adaptivního režimu jsou tedy nejdříve nastaveny váhy všech spojů na počáteční hodnotu a poté probíhá vlastní adaptace vah. Cílem adaptace je tedy nalézt takovou konfiguraci sítě ve váhovém prostoru, která v aktivním režimu realizuje předepsanou funkci [8].

1.5 Učící algoritmus

Cíl učení spočívá v nastavení vah a prahů jednotlivých neuronů tak, aby síť provedla požadovanou činnost. Dle způsobu učení jsou sítě děleny na dva druhy [1].

Učení s učitelem

Výstupy sítě jsou postupně konfrontovány s výstupy trénovací množiny a trénovací algoritmus upravuje hodnoty vah tak, aby rozdíl mezi výstupem sítě a výstupem uloženým v trénovací množině byl minimální.

Učení bez učitele

Síť má k dospozici pouze tréninkovou množinu.

2 SAMOORGANIZUJÍCÍ NEURONOVÉ SÍTĚ

Samoorganizující neuronové sítě jsou založeny na metodě soutěžní strategie učení (učení bez učitele), tedy vyžadují pouze předkládání vstupů. Základním principem je soutěžení výstupních neuronů o to, který z nich bude aktivní. Tato metoda se také nazývá samoorganizace, což znamená, že sít je schopná se přizpůsobit a nepotřebuje znát předpokládané výstupy [1,2].

2.1 Vektorová kvantizace

Cílem vektorové kvantizace je aproximovat pomocí vektorů hustotu pravděpodobnosti $p(x)$ rozložení reálných vstupních vektorů $x \in R^n$ pomocí konečného počtu jednotek či reprezentantů $Wi \in R^n; (i = 1, ..., h)$ [1,2].

Ke každému vektoru $x \in R^n$ je přiřazena jako reprezentant ta jednotka Wc, která má nejbližší hodnotu, tedy [1]:

$$c = argmin_{l=1,..,h} \{||x - w_l||\} \tag{2.1}$$

Jedním ze způsobů řešení problémů nalezení reprezentantů je minimalizovat chybu vektorové kvantizace definovanou jako:

$$E = \int ||x - w_c|| \, p(x) dx \tag{2.2}$$

Často není hustota pravděpodobnosti $p(x)$ známa a problém je zadán konečnou tréninkovou množinou vzorů. V takovém případě se chyba vektorové kvantizace vypočítává jako[1]:

$$E = \frac{1}{k} \sum_{t=1}^{k} \left|\left| x^{(t)} - w_c \right|\right|^2 \tag{2.3}$$

Funkční závislost indexu c na vzorech x a jednotkách w způsobuje značnou složitost explicitního vyjádření gradientu chyby E. Proto byl navržen heuristický iterativní algoritmus hledající přibližné řešení, známý pod názvem Lloydův algoritmus [1].

Lloydův algoritmus

Po průchodů celé tréninkové množiny T je dle [1] ke každému vstupu X(t) určena příslušná jednotka Wc dle vztahu (2.1). Pro každou jednotku W_j je uvážena množina T_j všech vzorů x(t), pro něž byl W_j jejich reprezentantem [1].

$$T_j = \left\{x^{(t)}; j = argmin_{l=1,.,h} \left\{\left|\left|x^{(t)} - w_l\right|\right|\right\}\right\} \tag{2.4}$$

Následující krok představuje výpočet těžiště t_j množiny T_j.

$$t_j = \frac{1}{|T_j|} \sum_{x_j \in T_j} \tag{2.5}$$

a nahrazení hodnoty wj hodnotou tj. Tímto postupem je určena nová poloha všech jednotek w_j a lze pokračovat průchodem tréninkovou množinou[1].

Nevýhoda tohoto algoritmu spočívá ve změně reprezentantů až po průchodu všech vektorů tréninkové množiny.

2.2 Kohonenovo učení

Nevýhodu Lloydova algoritmu řeší jeho on-line varianta známa jako Kohonenovo učení [1,2].

Jedná se o dvouvrstvou síť s úplným propojením jednotek mezi vrstvami. Vstupní vrstvu tvoří vstupy, které slouží k distribuci vstupních jednotek $x \in R^n$. Váhy $w_j = ((w_{j1}), ..., w_{jn})$ náležející jedné výstupní jednotce j určují její polohu ve vstupním prostoru.

Vstupy $x \in R^n$ mohou být libovolná reálná čísla, výstupy y_j tvoří jen hodnoty 0 a 1, přičemž aktivní (ten s hodnotou $y_j = 1$) je právě vítězný výstupní neuron. Výpočet vzdálenosti vítězného neuronu od vstupního prostoru $x^{(t)}$ se vypočítá dle [1]:

$$y_j = \left\{j = argmin_{l=1,.,h} \left\{\left|\left|x^{(t)} - w_l\right|\right|\right\}\right\} \tag{2.6}$$

Při průchodu celou tréninkovou množinou a po předložení jednoho tréninkového vzoru dochází mezi jednotkami sítě ke kompetici(porovnání). Vítězný neuron kompetice změní své váhy dle:

$$w_{ji}^{(t)} = \left\{w_{ji}^{(t-1)} + \alpha(x_i^{(t-1)} - w_{ji}^{(t-1)})\right\} \tag{2.7}$$

Parametr α představuje učící koeficient určující míru změny vah. Vítězný neuron posune své váhy w_c o poměrnou vzdálenost směrem k aktuálnímu vstupu, čímž ještě zlepší svou relativní pozici.

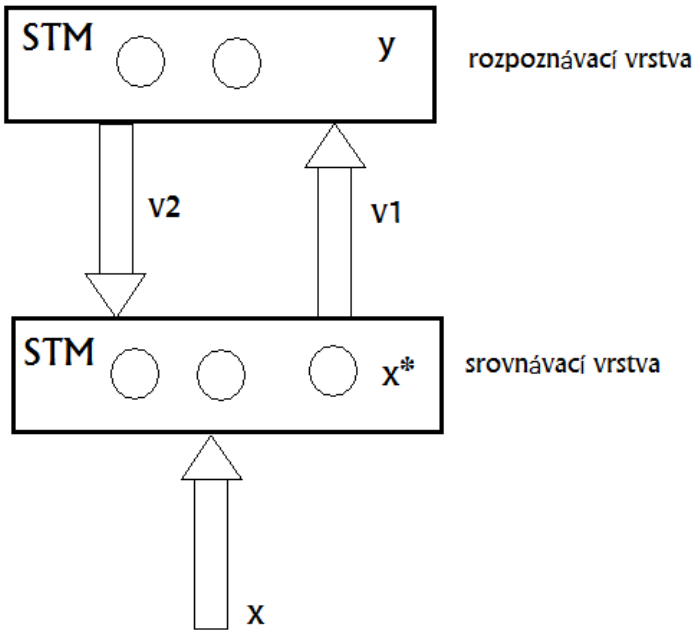
2.3 Neuronová síť ART

Síť ART (Adaptive Resonance Theory) je vícevrstvá rekurentní neuronová síť založená na teorii učení bez učitele. Princip sítě je založen na teorii adaptivní resonance, kdy po přivedení vzoru na vstup dochází k přímému i zpětnému šíření signálů, které probíhají ve vlnách, dokud se síť neustálí při shodě (zarezonuje). Ustálení signalizuje nalezení shodného, nebo podobného vzoru [4].

Existuje několik druhů ART sítí. Nejpoužívanější je síť ART 1, která zpracovává pouze binární hodnoty na vstupech, a varianta sítě ART 2, která již zpracovává reálné hodnoty na vstupech[13].

První vrstva sítě je vrstva srovnávací, tedy porovnává předložený vzor s předešlými. Druhá vrstva sítě je vrstva rozpoznávací, tedy zjišťuje podobnosti mezi předkládanými vzory a rozděluje je do kategorií. Obě vrstvy jsou vzájemně propojeny vahami, avšak zpětné šíření je využito pouze od výstupní vrstvy ke vstupní a také mezi neurony výstupní vrstvy [4].

V ART síti existují dva typy vah. Dopředné váhy vedoucí od vstupu k výstupu, dle našeho značení ($v1$), a zpětné váhy vedoucí od výstupu ke vstupu, dle našeho značení ($v2$)(Obr. 2.1). Síť také obsahuje řídicí signály řídící tok dat mezi vrstvami. Mezi vrstavmi se nachází nulovací člen, porovnávající vstup s prahem bdělosti(ρ)[4].



Obr. 2.1: Obecné schéma modelu ART

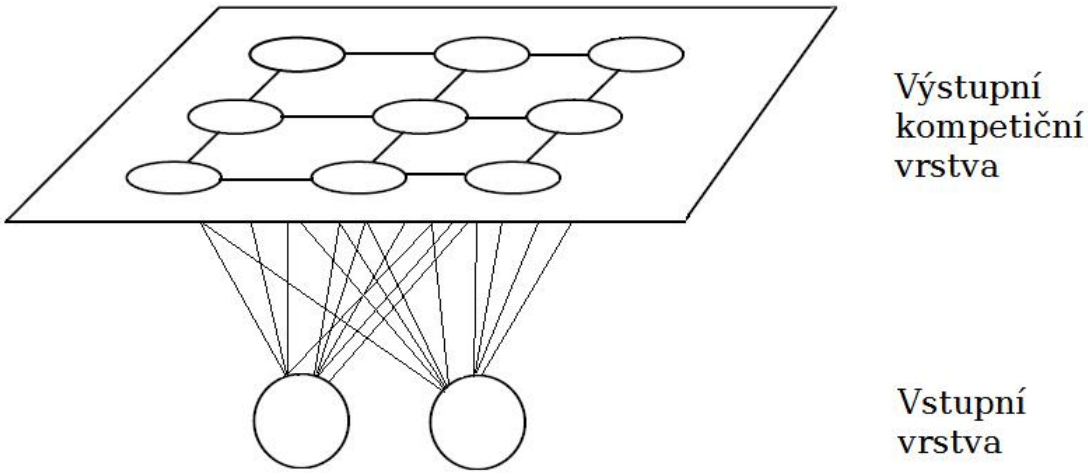
2.3.1 Popis funkce sítě ART

V této kapitole popíšeme proces fungování sítě ART, který probíhá dle [4] následovně:

- Na začátku je předložen vstupní vektor x porovnávací vrstvě. Ta aktivuje vzor v podobě kratkodobé paměti(STM).
- Prostřednictvím vazeb reprezentovaných $v1$ a laterálních vazeb je vybrán neuron s největší exitací v rozpoznávací vrstvě.
- Vítězný neuron rozpoznávací vrstvy vyšle signál $v2$ k srovnávací vrstvě. Ten se nazývá očekávání a je odvozen z předchozí zkušenosti. Tento signál aktivuje ve srovnávací vrstvě nový vzor $x*$.
- Původní vektor x je porovnán se zpětnovazebním signálem. Předdefinovaná hodnota ρ udává, zda jsou si dostatečně podobné. Pokud podobnost obou vektorů není dostatečná, pak vybraný neuron přesně nereprezentuje správnou třídu a vítězný neuron je vyřazen z množiny možných vítězů.
- Pokud existují v rozpoznávací vrstvě další neurony, které by mohly odpovídat vzoru, jsou stejným způsobem otestovány i ony. Pokud žádný takový není, je vtažen do procesu další, zatím neangažovaný neuron.
- V případě shody mezi vstupem a zpětnovazebním signálem (rozdíl je dostatečně malý), a tedy výtězný neuron reprezentuje vektor x, dochází k jevu rezonance.
- Dojde-li k rezonanci, dochází k procesu adaptace vah.

3 KOHONENOVA SÍŤ

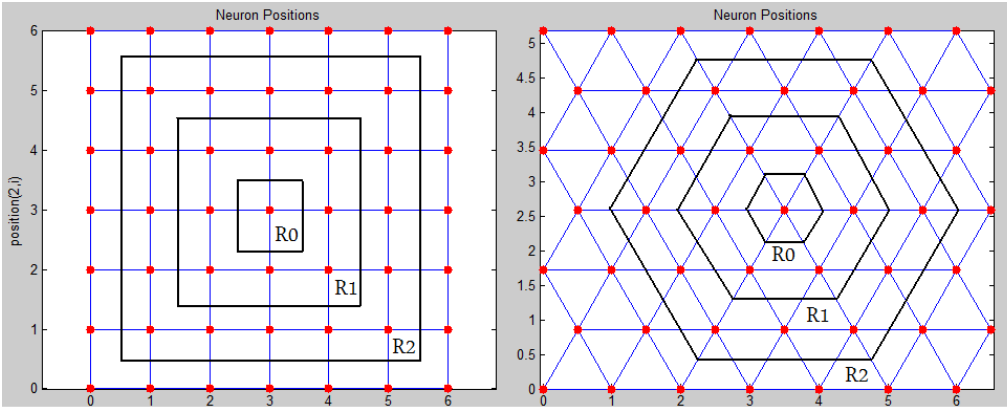
Kohonenova neuronová síť, někdy též nazývána Kohonenovy samoorganizační mapy (SOM), patří do skupiny samoorganizujících neuronových sítí, tedy s učením bez učitele. Svou schopností samoorganizace a shlukování objektů s podobnými vlastnostmi do skupin jsou Kohonenovy mapy předurčeny pro rozhodování, třídění a rozlišování dat.



Obr. 3.1: Topologie Kohonenovy samoorganizační mapy

Principem vychází Kohonenova síť z Kohonenova učení. Jedná se tedy o jednovrstvou síť s úplným propojením jednotek mezi vstupy a kompetiční vrstvou, tedy každý neuron má informaci o hodnotě každého vstupu (Obr. 3.1). Neurony v kompetiční vrstvě jsou uspořádány do topologické struktury, většinou do dvojrozměrné obdélníkové, nebo hexagonální dvojrozměrné oblasti (Obr. 3.2). Tvar topologické struktury uspořádání neuronů kompetiční vrstvy má vliv na tzv. okolí neuronu R, které vymezuje sousední, nejblíže postavené neurony. Velikost okolí je rovna počtu řad neuronů od centrálního neuronu. Váhy jednotlivého neuronu tedy definují jeho polohu v prostoru. Funkcí každého neuronu je výpočet vzdálenosti vstupního vektoru od vzoru uloženého ve vahách neuronu [2,5].

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2 \tag{3.1}$$



Obr. 3.2: Možné struktury uspořádání neuronů v okolí vítězného neuronu R

3.1 Učení Kohonenovy síťě

Princip učení neuronové sítě spočívá v uspořádání neuronů kompetiční vrstvy do oblastí, které klasifikují předložená vstupní data.

K adaptaci vah dochází během každého kroku, tedy iterativně. Během učení dochází k porovnávání vstupních vzorů a vektorů, jež jsou uloženy v každém neuronu. Nalezením neuronu kompetiční vrstvy, jenž nejlépe odpovídá vzoru, jsou váhy tohoto neuronu upraveny. Taktéž jsou upraveny i váhy neuronů v okolí vítězného neuronu.

Učení Kohonenovy sítě probíhá následovně [2,5]:

1. Inicializace:

Nastavení vah w_{ij}, $0 \leq i \leq N-1, 0 \leq j \leq M-1$ pro všechny spoje z N vstupů do M výstupních neuronů na malé náhodné počáteční hodnoty. Nastavení parametru učení α na hodnotu blížící se 1. Nastavení okolí kolem každého neuronu[1,2,5].

2. Předložení vzorů:

Předložení nového trénovacího vzoru $X(t) = \{x_o(t), x_1(t)...x_{N-1}(t)\}$ na vstupy.

3. Výpočet vzdálenosti vzorů:

Výpočet vzdálenosti d_j mezi předloženým vzorem a všemi výstupními neurony j.

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2 \tag{3.2}$$

4. Výběr nejpodobnějšího neuronu:

Výběr výstupního neuronu j*, který splňuje podmínku. Tento neuron se stává nejpodobnějším neuronem.

$$d_j* = min_j(d_j) \tag{3.3}$$

5. Přizpůsobení vah:

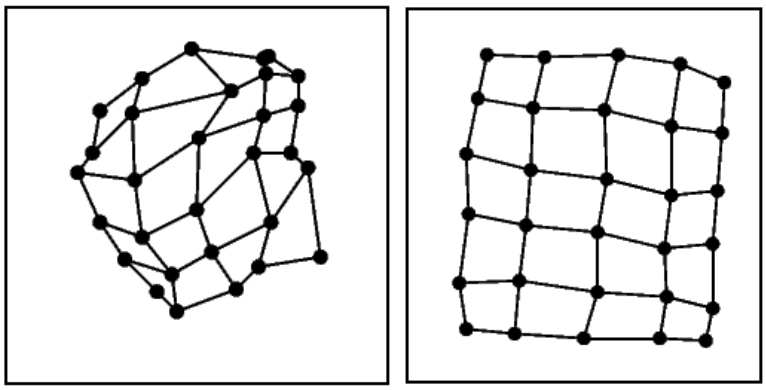
Přizpůsobení vah pro neuron j* a jeho okolí $N_(j*)(t)$ dle vztahu:

$$W_{ij}(t+1) = W_(ij)(t) + \alpha(t)\left[x_i(t) - w_{ij}(t)\right] \tag{3.4}$$

K adaptaci vah dochází po každém nalezení vítězného neuronu, a to jak pro vítězný neuron, tak pro definované okolí. Během postupného učení dochází v závislosti na učícím parametru i ke snižování okolí až na vítězný neuron. Trénování neuronové sítě lze rozdělit na dvě fáze. Během první fáze dochází k hrubému naučení sítě, kdy dochází k velkým změnám vah, a tedy k rozprostření neuronů po celé ploše vzorů. Během druhé fáze dochází k jemnému učení sítě, a tedy menším změnám vah.

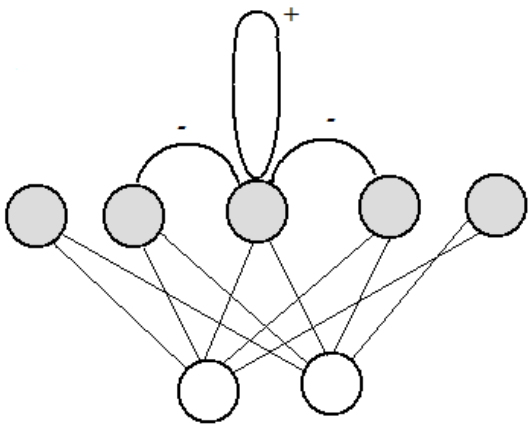
6. Pokračování učicího procesu:

Pokud nejsou vyčerpány všechny vzory, nebo počet iterací, následuje přechod ke kroku *2. Předložení vzorů*. V případě vyčerpání všech vzorů, tedy naučení sítě, následuje ukončení procesu[2,5].

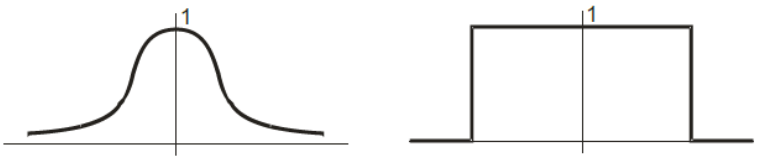


Obr. 3.3: Proces postupné adaptace mapy

Na začátku učení jsou všechny váhy mezi vstupy a výstupními neurony nastaveny na malé náhodné hodnoty. Každý z neuronů je definovám v prostoru těmito náhodnými váhovými vektory. Nastavení parametru učení α má vliv na rychlost a kvalitu naučení sítě. Na začátku je hodnota α nastavena na hodnotu blízkou k 1. Během procesu učení dochází k postupnému snižování učicího parametru. Při inicializaci je definována i volba okolí, a to jak pro jednotlivé neurony, tak pro celou síť. Mezi neurony v definovaném okolí dochází k vzájemnému ovlivňování. Vítězný neuron působí sám na sebe exitační vazbou, tedy zvyšuje svou vlastní váhu. Na okolní neurony v okolí působí inbiční vazbou, čímž snižuje jejich váhu (Obr. 3.4). Adaptační funkce určuje, jakou silnou inhibiční vazbou budou neurony v definovaném okolí kompetiční vrstvy ovlivňovány (Obr. 3.5).



Obr. 3.4: Vliv exitace a inhibice



Obr. 3.5: Biologická adaptivní funkce, zjednodušená adaptivní funkce

3.2 Vybavovací fáze

Vybavovací fáze představuje zařazení neznámého vzoru do odpovídajících tříd. Není potřeba inicializace, protože síť už je naučená a stačí pouze předložit neznámý vstupní vzor.

Vybavovací proces probíhá ve 3 fázích [1,2,3]:

1. Předložení nového vzoru:

Předložení neznámého testovacího vzoru $X(t) = \{x_o(t), x_1(t)...x_{N-1}(t)\}$ na vstupy sítě.

2. Výpočet nejbližšího neuronu:

Výpočet všech vzdáleností d_j mezi neznámým vzorem a všemi váhovými koeficienty výstupních neuronů.

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2 \tag{3.5}$$

Neuron j*, který splňuje podmínku minimální vzdálenosti d_j, představuje vítěze. Tento neuron představuje odpověď sítě na neznámý vzor a udává třídu (shluk), do které je vzor přiřazen.

$$d_j* = min_j(d_j) \tag{3.6}$$

3. Dokončení či opakování:

V této fázi existují dvě možnosti. Buď je klasifikace ukončena, nebo pokračuje krok jedna klasifikace předložených vzorů[1,2,5].

Pro zařazení neznámého vstupního vzoru do shluků rozlišujeme dle[5] dva typy:

Neadaptační režim - Dojde k zařazení neznámého vstupního vzoru do jednoho ze shluků, což je zároveň i výsledek.

Adaptační režim - Dojde k zařazení neznámého vstupního vzoru do jednoho ze shluků, následně však dochází k úpravě vah vítězného neuronu. Neuronová síť se tedy s každým neznámým vstupním vzorem doučuje.

4 REŠERŠE APLIKACÍ KOHONENOVY SÍTĚ

Tato kapitola je zaměřena na konkrétní příklady uplatnění aplikací Kohonenovy neuronové sítě. V mnoha pracech, odborných i populárně naučných, se stále jen dozvídáme, že využití Kohonenovy sítě v praxi je široké. Jako nejčastější oblasti využití jsou, stejně jako námi zmíněné v úvodu k této práci, uváděny například automatizované provozy, lékařství, ekonomika nebo věda.

Uvádíme jako příklady ty projekty, které již byly vytvořeny a jsou funkční.

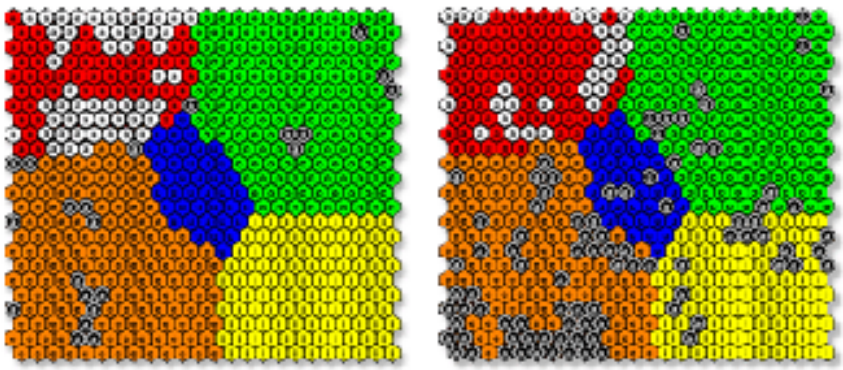
4.1 Diagnostika neurologických onemocnění u dětí

Jedním z projevů neurologických onemocnění u dětí je zhoršení řečového projevu, které může skončit až úplnou ztrátou schopnosti dítěte mluvit. Doposud lékaři neměli rychlou a spolehlivou metodu, jak zhoršující se řečový projev ověřit. Proto je Katedrou teorie obvodů ČVUT společně s Fakultní nemocnicí v Motole vyvíjen program, který bude na základě získaných dat schopen zhoršení řeči dítěte odhalit.

Postup je následující. Hlas dítěte je nahrán do souboru. Soubor je segmentován na kratší úseky – slabiky nebo hlásky. Takto upravená data jsou převedena parametrizací na číselné hodnoty koeficientů, které jsou předloženy neuronové síti pro její natrénování.

Metoda využívá Kohonenovy samoorganizující mapy pro segmentaci vstupního záznamu řeči a rozděluje jej na jednotlivé části, které jsou poté vyhodnoceny. Výstupem jsou samoorganizační mapy, ze kterých lze porovnáním zjistit, zda došlo ke zhoršení, nebo zlepšení řečového projevu dítěte (Obr. 4.1).

Tento text vznikl na základě informací získaných z [15].



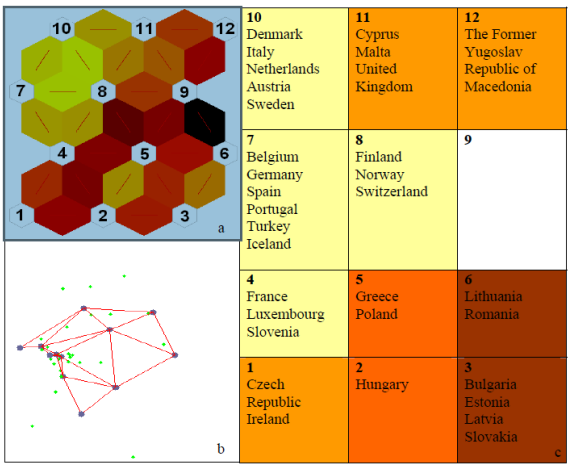
Obr. 4.1: Rozdíl mezi zdravým (vlevo) a nemocným dítětem.

4.2 Využití samoorganizující se neuronové sítě pro hodnocení chování spotřebitelů

Kohonenova samoorganizující síť byla v tomto případě využita pro hodnocení chování spotřebitelů v jednotlivých zemích Evropské unie. Jako data byly použity tabulky z databáze Eurostat o výdajích za posledních 11 let jednotlivých zemí. Do výdajů domácností byly začleněny položky: jídlo, nápoje, oděvy, obuv, bydlení, energie, doprava a podobně.

Jako vstupní data byly zformovány trojrozměrné matice (rok, země, výdaje v jednotlivých komoditách). Získané výsledky tak postihují jednotlivé charakteristiky v chování spotřebitelů a neklasifikují země jen podle celkových hodnot výdajů (Obr. 4.2).

Tyto a další údaje byly čerpány ze zdroje [14].



Obr. 4.2: Klasifikace zemí pomocí Khonenovy mapy

4.3 Získávání znalostí z textu

Získávání znalostí z relačních databází obsahujících textové dokumenty představuje problém v případě, kdy jednotlivé dokumenty představují nestrukturovaný text a není možné použít klasické metody kategorizace.

Nejpoužívanějším způsobem reprezentace textového dokumentu je použití vektoru, který má tolik složek, kolik je slov (termínů), souboru dokumentů, které je třeba zpracovávat. Jednotlivé dokumenty tedy bývají reprezentovány řídkými vektory o tisíci hodnotách.

Jednou z metod shlukování a kategorizace dokumentů představuje využití Kohonenovy samoorganizující sítě. Geometrická interpretace Kohonenovy sítě se převádí

na interpretaci pojmovou, tedy čím jsou dva shluky v Kohonenově mapě vzdálenější, tím rozdílnější obsah odpovídá příslušným dokumentům. Jednotlivé dokumenty jsou reprezentovány vektory, jejichž složky vyjadřují TFIDF (term frequency inverse document frequency), jednotlivé shluky jsou reprezentovány klíčovými slovy. Tato metoda je obzvlášť využívána v souvislosti s kategorizací dokumentů na internetu.

Problematika této aplikace byla přejata z [16].

$$TFIDF = n * log\frac{M}{m} \tag{4.1}$$

- n - počet výskytu termínu v dokumentu
- m - počet výskytu termínu v souboru dokumentů
- M - počet dokumentů v souboru

4.4 Modelování bonity obcí pomocí Kohonenových samoorganizujících map a LVQ

Časté využití Kohonenovy sítě představuje zpracování dat pro další použití. V tomto případě je Kohonenova síť využita pro zpracování bonity obcí, tedy schopnosti plnit krátkodobé a dlouhodobé závazky. Ohodnocení bonity obcí lze považovat za klasifikační problém, tudíž je vhodné použití Kohonenovy sítě.

Pro ohodnocení bonity obcí jsou používány tyto kategorie parametrů: ekonomické, dluhové, finanční a administrativní. Každá z kategorií je složena z více parametrů a z těchto parametrů je složena datová matice představující vstup Kohonenovy sítě.

Výstupem Kohonenovy sítě je mapa klasifikující obce podle bonity, v závislosti na nastavení. Pro další zpracování a vyhodnocení bonity obcí byla využita metoda LVQ.

Zde zmíněné a bližší údaje k aplikace na [17].

5 POPIS METODIKY ŘEŠENÍ PROBLÉMU

5.1 Klasifikace problému

Úkolem této práce je u zvolené množiny trénovacích vzorků zkoumat vliv počátečního natavení vah neuronů na konečnou pozici vítězných neuronů. Konečná pozice představuje přiřazení vzoru ke konkrétnímu neuronu kompetiční vrstvy po ukončení trénování.

5.1.1 Tréninková množina

Tréninková množina je tvořena stem vzorů, každý vzor obsahuje dva prvky, na kterých se Kohonenova síť učí. Data jsou náhodného rozložení v rozmezí od 0 do 1 a jsou vytvořena funkcí náhodného rozložení v programu MATLAB. Data jsou uložena v souboru data.dat na přiloženém CD.

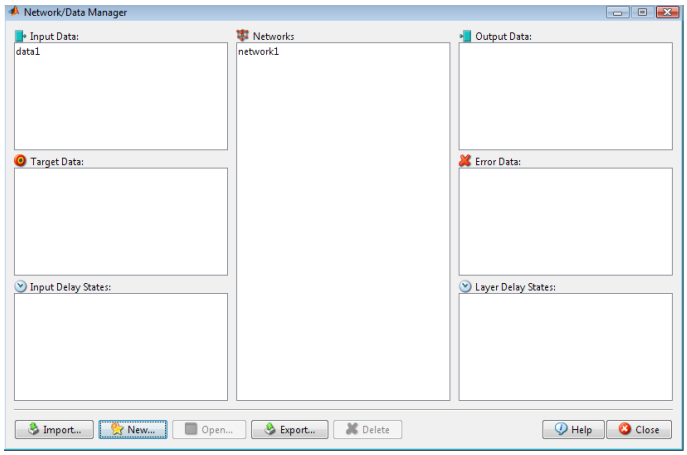
5.2 Výběr vhodného programu

Pro zjištění vlivu počátečních vah neuronů na konečnou pozici, bylo potřeba zvolit vhodný simulátor neuronových sítí. Dostupnost simulátorů je velká, avšak bylo potřeba nalézt takový, který by umožňoval libovolně nastavit počáteční váhy a zároveň byl schopen klasifikovat vstupní vzory ke konkrétním neuronům. Tato kritéria splňovalo několik simulátorů, ale dalším problémem bylo, že tyto simulátory neumožňovaly zároveň nastavení počátečních vah neuronů hromadně pro Kohonenovu síť.

Konečnou volbou proto bylo programové prostředí MATLAB.

5.3 MATLAB

MATLAB je programové prostředí a skriptovací jazyk pro výpočty, simulace a modelování. Jedním z jeho implementovaných nástrojů je i simulátor neuronových sítí Neural Network Toolbox(NNT) (Obr. 5.1). Toto grafické prostředí je určeno pro práci s neuronovými sítěmi. Simulátor se spouští příkazem **nntool**. Pomocí grafického prostředí a jeho voleb lze nastavit požadovaný typ neuronové sítě, vstupní data i parametry trénování.



Obr. 5.1: Neural Network Toolbox

5.4 Program

Po prvních výsledcích práce s programem NNT jsem zvolil jako nejlepší možné řešení vytvoření vlastního programu na základě funkcí Kohonenovy mapy implementovaných v programu MATLAB 2008a. Tento program je uložen na přiloženém CD, zdrojový kód je v příloze A. Program je uložen v souboru **program.m** a je spouštěn v prostředí MATLAB.

5.5 Popis programu

Program je složen z několika částí využívajících vnitřní funkce prostředí MATLAB.

5.5.1 Funkce nnt2som

je funkce pro vytvoření Kohonenovy sítě. Nastavením parametrů přímo v programu dochází k nastavení sítě.

net = nnt2som(PR,[D1,D2,...],W,OLR,OSTEPS,TLR,TND)

Kde parametr **net** inicializuje vybranou síť.

- PR - Rx2 matice tvořená hodnotami min a max z R vstupních elementů
- Di - dimenze vrstvy (mapy)
- W - SxR matice vah
- OLR - počáteční learning rate
- OSTEPS - počet iterací
- TLR - adaptivní fáze learning rate
- TND - adaptivní šířka okolí vítěze

5.5.2 Funkce Train

Tato funkce spouští vlastní trénování inicializované sítě.

net = train(net,P)

- P - značí načtenou matici vstupních vzorů

5.5.3 Funkce Plot, Plotsom

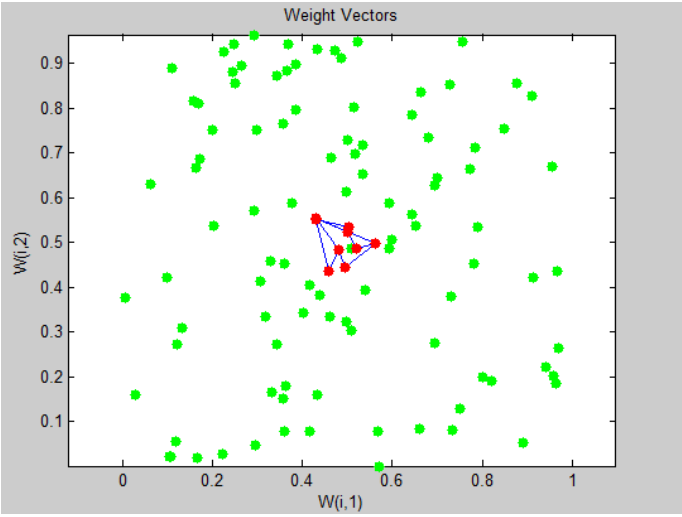
- Plot - podle nastavených parametrů zobrazuje rozložení vstupních vzorů v prostoru.
- Plotsom - podle nastavených parametrů zobrazuje rozložení neuronů v prostoru a jejich vzájemné vazby.

5.6 Ukázka funkce programu

Před spuštěním programu je možno změnit výchozí parametry trénování. Program načítá vstupní data ze souboru data.dat. Počáteční hodnoty vah neuronů jsou generovány náhodně v definovaném rozmezí.

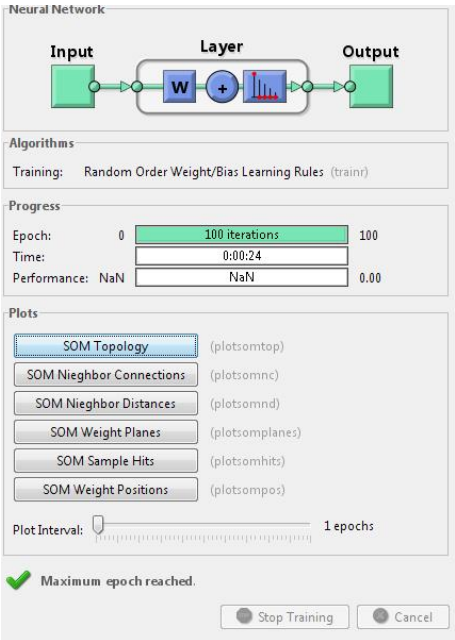
- Wmin - minimální hodnota počátečních vah
- Wmax - maximální hodnota počátečních vah
- net.trainParam.epochs - počet iterací (kolikrát bude SOM trénována na celou množinu vstupních dat)
- R*S počet neuronů kompetiční vrstvy
- další nastavení parametrů sítě již bylo popsáno v **Funkce nnt2som**

Po spuštění programu (program.m) dochází k načtení vzorů ze souboru (data.dat). Program vykreslí mapu počátečního rozložení neuronů i s jejich vazbami do mapy rozložení počátečních vzorů (Obr. 5.2).



Obr. 5.2: Počáteční rozložení neuronů s jejich vzájemnými vazbami

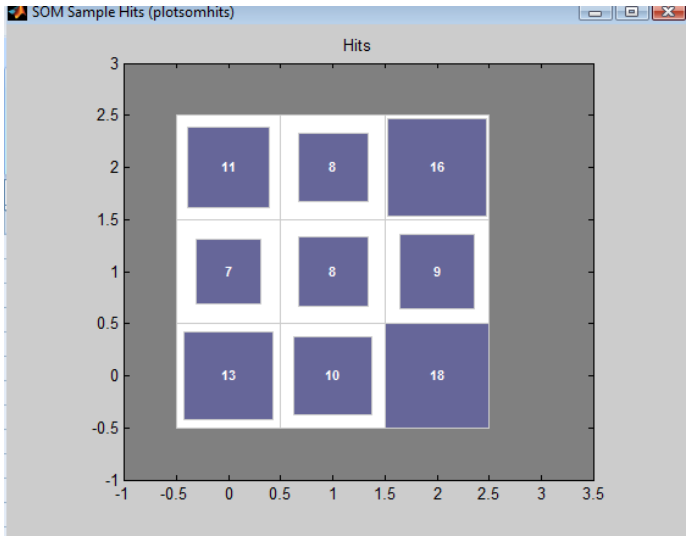
Po tomto načtení dochází k otevření okna **Neural Network Training**, které graficky zobrazuje průběh vlastního trénování (Obr. 5.3).



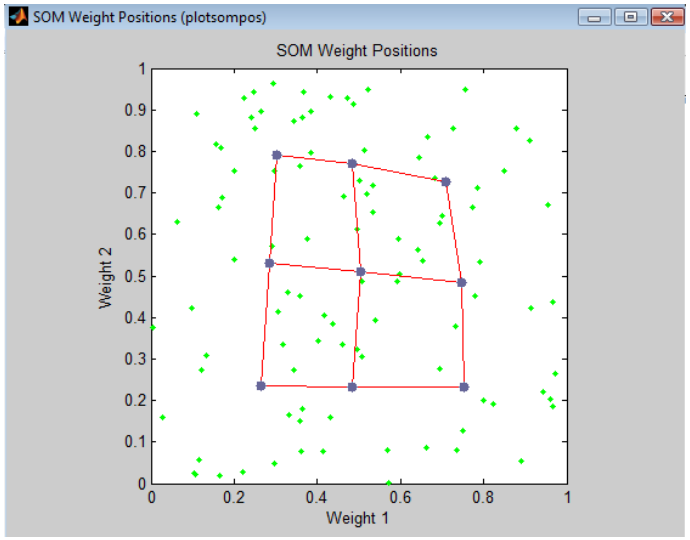
Obr. 5.3: Průběh trénování neuronové sítě

- SOM Topology - topologie sítě
- SOM Neighbor Connections - vzájemné propojení neuronů
- SOM Neighbor Weight Distances - váhové rozdíly mezi neurony
- SOM Sample Hits - počet vstupů přiřazených k neuronu (Obr. 5.4)
- SOM Weight Positions - rozložení neuronů ve vstupním protoru (Obr. 5.5)

Do proměnné **vystup** jsou uložena čísla neuronů, ke kterým byl daný vstup přiřazen.



Obr. 5.4: Grafické znázornění přiřazení vzorů k neuronům



Obr. 5.5: Konečné rozložení neuronů i s jejich vazbami

6 OVĚŘENÍ ÚKOLU V ZADÁNÍ

6.1 Změna vah jednoho neuronu

Program byl nastaven na parametry:

- učící koeficient - 0,95
- rozmezí počátečních vah - 0,4-0,5
- 100 iterací
- volba okolí = 1

Program byl spuštěn s náhodně generovanými vahami v daném rozmezí a výsledky z proměnné **vystup** uloženy, jako i hodnota vygenerovaných vah. Před druhým spuštěním byla jedna hodnota uložených vah změněna o 0,001(0,4085-0,4095). S takto upravenými vahami byl program spuštěn a výsledky uloženy z proměnné **vystup**.

Porovnáním výsledků lze zjistit, že při druhém spuštění bylo jen 5 procent vstupních vzorů přiřazeno ke stejným neuronům, jako při prvním průběhu programu. Změnou hodnoty váhy neuron posune svou pozici a tudíž během kompetice ovlivní hodnoty ostatních neuronů. Jakákoliv změna vah se tedy projeví na celé neuronové síti a dochází tedy k odlišnému naučení sítě, tudíž i změněnému přiřazení vzorů k neuronům.

6.2 Síť 3x3 a 5x5

Nastavení programu:

- učící koeficient - 0,95
- 100 iterací
- volba okolí 1
- topologie 3x3 - čtvercová

Program byl spuštěn s náhodně generovanými hodnotami vah v daném rozmezí. První výsledek přiřazení vzorů k neuronům představuje referenční výsledek. Dalších 10 spuštění programu, s náhodně generovanými vahami v daném rozmezí, vygeneruje 10 množin hodnot přiřazení vzorů k vystupním neuronům. Tyto hodnoty představují porovnávané hodnoty přiřazení vzorů k neuronům. Hodnoty byly postupně kopírovány z proměnné **výstup** a ukládány do souboru 3x3.xls. V tomto souboru byly uloženy do tabulky a porovnávacím algoritmem byl určen počet stejně přiřazených vzorů.

6.2.1 Síť 3x3 s volbou okolí=1

Nastavení programu:

- učící koeficient - 0,95
- 100 iterací
- volba okolí 1
- topologie 3x3 - čtvercová

Z výsledků v tabulce je zřejmé, že rozsah počátečních vah má na přiřazení vzorů ke stejným neuronům vliv. U nižších rozsahů počátečních vah existuje vyšší pravděpodobnost přiřazení ke stejným vzorů, u vyšších rozsahů hodnota klesá. Určitý vliv na získané výsledky má i vliv okolí, kdy dochází k exitaci vítězného neuronu, ale i k laterální inhibici neuronů ve zvoleném okolí. Avšak vše je založeno na náhodném generování počátečních vah, a tak není zaručeno, že se i u vyšších rozsahů počátečních vah budou generovat hodnoty vah v celém rozsahu. Výsledky jsou uloženy v souboru 3x3.xls.

rozmezí vah	úspěšné přiřazení vzorů										procentuálni úspěšnost
0,50-0,51	8	37	9	8	9	27	34	23	99	96	35
0,49-0,52	26	8	8	6	37	23	5	8	34	97	25,2
0,48-0,53	37	37	23	8	6	8	8	22	8	26	18,3
0,45-0,55	33	38	4	97	8	21	26	99	8	23	35,7
0,4-0,5	8	24	24	15	16	8	8	8	36	37	18,4
0,3-0,6	23	8	49	37	37	23	8	100	8	37	33
0,25-0,75	6	21	29	87	34	6	3	3	34	6	22,9
0,15-0,85	11	45	12	5	12	5	19	29	5	5	14,8

Tab. 6.1: Procentuální úspěšnost přiřazení neuronů v síti 3*3 s volbou okolí=1.

6.2.2 Síť 3x3 s volbou okolí=0

Nastavení programu:

- učící koeficient - 0,95
- 100 iterací
- volba okolí 0
- topologie 3x3 - čtvercová

Z výsledků v tabulce je zřejmé, že vliv počátečního nastavení vah po celém rozsahu nezpůspobuje výraznější rozdíly přiřazení vzorů k neuronům. Vliv okolí byl v tomto případě potlačen a ke změně vah docházelo pouze u vítězného neuronu. Výsledky jsou uloženy v souboru 3x3.xls.

rozmezi vah	úspěšné přiřazení vzorů										procentuálni úspěšnost
0,50-0,51	23	26	8	63	8	5	23	37	8	95	29,6
0,49-0,52	36	27	37	37	8	36	8	82	8	8	28,7
0,48-0,53	28	26	37	8	24	18	23	97	6	8	27,5
0,45-0,55	38	26	23	38	8	8	8	6	22	37	21,4
0,4-0,5	26	100	100	8	8	23	37	26	8	6	34,2
0,3-0,6	26	8	22	23	33	7	81	8	8	36	25,2
0,25-0,75	8	8	8	8	8	8	8	98	98	26	27,8
0,15-0,85	37	26	26	8	5	8	37	26	96	8	27,7

Tab. 6.2: Procentuální úspěšnost přiřazení neuronů v síti 3*3 s volbou okolí=O.

6.2.3 Síť 5x5 s volbou okolí=1

Nastavení programu:

- učící koeficient - 0,95
- 100 iterací
- volba okolí 1
- topologie 5x5 - čtvercová

Z tabulky je patrné, že vliv počátečního rozmezí nastavení vah není výrazný. Vyšší počet neuronů, ke kterým jsou vzory přiřazovány, snižuje pravděpodobnost přiřazení. Výsledky jsou uloženy v souboru 5x5.xls.

rozmezí vah	úspěšné přiřazení vzorů										procentuálni úspěšnost
0,50-0,51	2	2	3	12	2	16	18	20	2	8	8,5
0,49-0,52	1	8	5	19	15	12	1	13	17	66	15,7
0,48-0,53	15	15	1	14	2	14	80	17	12	18	18,8
0,45-0,55	76	16	11	11	1	12	15	1	6	16	16,5
0,4-0,5	17	19	17	1	2	15	20	89	1	16	19,7
0,3-0,6	81	94	16	2	14	9	2	2	21	2	24,3
0,25-0,75	2	1	1	14	2	2	4	10	16	8	6
0,15-0,85	73	1	18	8	1	14	1	10	12	1	13,9

Tab. 6.3: Procentuální úspěšnost přiřazení neuronů v síti 5*5 s volbou okolí=1.

6.2.4 Síť 5x5 s volbou okolí=0

Nastavení programu:

- učící koeficient - 0,95
- 100 iterací
- volba okolí 0
- topologie 5x5 - čtvercová

Z tabulky je patrné, že vliv počátečního rozmezí nastavení vah není výrazný. Pouze u rozmezí počátečního nastavení vah 0,49-0,52, došlo k 43,2 procentům přiřazení vzorů ke stejným neuronům. Tato skutečnost byla způsobena náhodným vygenerováním podobných počátečních vah. Vyšší počet neuronů, ke kterým jsou vzory přiřazeny, snižuje pravděpodobnost přiřazení ke stejným neuronům. Výsledky jsou uloženy v souboru 5x5.xls.

rozmezí vah	úspěšné přiřazení vzorů										procentuálni úspěšnost
0,50-0,51	1	1	17	34	8	11	81	7	1	11	17,2
0,49-0,52	19	72	13	2	2	2	71	93	73	85	43,2
0,48-0,53	1	15	8	1	2	47	2	18	2	1	9,7
0,45-0,55	12	2	2	7	11	2	69	14	14	14	14,7
0,4-0,5	18	16	89	18	85	2	75	14	2	13	33,2
0,3-0,6	1	15	15	1	13	15	12	20	2	1	9,5
0,25-0,75	8	85	2	2	1	2	11	18	2	2	13,3
0,15-0,85	1	6	8	1	13	1	16	5	16	1	6,8

Tab. 6.4: Procentuální úspěšnost přiřazení neuronů v síti 5*5 s volbou okolí=O.

6.2.5 Síť 3x3, rozmezí vah mimo střed s volbou okolí=0

Z tabulky je patrné, že ani změna vah mimo střed rozložení vzorů nemá vliv na umístění vítězných neuronů. Výsledky jsou uloženy v souboru 3x3-10x10.xls.

- učící koeficient - 0,95
- 100 iterací
- volba okolí 0
- topologie 3x3 - čtvercová

rozmezí vah	úspěšné přiřazení vzorů										procentuálni úspěšnost
0,1-0,2	99	26	25	8	94	47	24	8	5	5	34,1
0,1-0,3	100	8	33	37	8	8	37	8	8	34	28,1
0,7-0,9	37	23	38	26	23	8	37	23	26	34	27,5
0,8-0,9	38	23	8	8	38	26	37	37	37	23	27,5

Tab. 6.5: Procentuální úspěšnost přiřazení neuronů v síti 3x3 s volbou okolí=O.

6.2.6 Síť 10x10 s volbou okolí=0

Nastavení programu:

- učící koeficient - 0,95
- 100 iterací
- volba okolí 0
- topologie 10x10 - čtvercová

Z tabulky je patrné, že vyšší počet kompetičních neuronů snižuje pravděpodobnost přiřazení vzorů ke stejným neuronům v průměru na procenta. Na tomto vlivu se podílí svou měrou i malé množství vzorů v tréninkové množině. Výsledky jsou uloženy v souboru 3x3-10x10.xls.

rozmezí vah	úspěšné přiřazení vzorů										procentuálni úspěšnost
0,50-0,51	0	6	0	0	4	12	0	0	0	1	2,3
0,49-0,52	0	6	4	0	0	4	0	5	0	2	2,1
0,48-0,53	50	4	0	0	2	30	4	2	6	2	10
0,45-0,55	5	5	5	3	0	0	2	0	0	2	2,2
0,4-0,5	7	11	0	1	0	4	0	2	1	44	7
0,3-0,6	0	0	1	0	0	1	1	0	0	0	0,3
0,25-0,75	13	0	4	0	9	0	0	0	0	2	2,8
0,15-0,85	4	1	3	0	1	0	0	37	4	0	5

Tab. 6.6: Procentuální úspěšnost přiřazení neuronů v síti 10*10 s volbou okolí=O.

7 ZÁVĚR

Výsledkem této bakalářské práce je zjištění vlivu počátečních hodnot vah v Kohonenově síti na konečné přiřazení ke kompetičnímu neuronu. V práci je proveden popis teorie umělých neuronových sítí a samoorganizujících se sítí, zvláště pak Kohonenovy samoorganizační sítě. V rešerši je proveden popis aplikace Kohonenovy sítě v praxi.

Z důvodů zadání bylo nutno vytvořit program v prostředí MATLAB a pomocí tohoto programu demonstrovat vliv počátečního nastavení vah na přiřazení vzorů ke kompetičnímu neuronu.

Již z prvního výsledku, při změně jednoho parametru vah, je zřejmé, že došlo k pouhé pětiprocentní shodě mezi referenčním a testovacím přiřazením vzorů. Vliv změny vah jediného neuronu se projeví na celém trénování sítě, jelikož změnou počáteční váhy změní neuron svou pozici a během kompetice ovlivní i ostatní neurony.

Dále byl porovnáván vliv počátečního nastavení vah, v daném rozmezí, na úspěšnost přiřazení vzorů k neuronům kompetiční vrstvy. Z výsledků je patrné, že rozmezí počátečního nastavení vah nemá na konečné přiřazení vliv. Záleží na konkrétním nastavení vah a jejich vzájemné počáteční pozici.

Z výsledků je též patrné, že u sítě topologie 3x3 došlo k častějšímu úspěšnému přiřazení než v síti s topologií 5x5 a nebo 10x10. To je způsobeno vyšším počtem kompetičních neuronů v těchto sítích.

Program je uložen v příloze A. Veškerá získaná data, program a množina vzorů jsou uloženy na přiloženém CD.

LITERATURA

[1] ŠÍMA, J., NERUDA, R. *Teoretické otázky neuronových sítí.* Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9.

[2] JIRSÍK, V., HRÁČER, P. *Neuronové sítě, expertní systémy a rozpoznávání řeči.* Brno: Vysoké učení technické, 2002. 107 s.

[3] VONDRÁK, I. *Umělá inteligence a neuronové sítě.* Ostrava: VŠB-TU, 2009. 139 s. ISBN 978-80-248-1981-5.

[4] VONDRÁK, I. *Neuronové sítě.* Ostrava: VŠB-TU, 1994. 56 s. Elektronický text. Dostupný z:

<http://vondrak.cs.vsb.cz/download/Neuronove site.pdf>.

[5] VOJÁČEK, A. *Samoučící se neuronová síť* [online].Datum publikování 2006-05-14. Dostupné z:

<http://automatizace.hw.cz/mereni-a-regulace/ART244-samoucici-se–neuronova-sit–som-kohonenovy-mapy.html>.

[6] *The MathWorks* [online]. Dostupné z:

<http://www.mathworks.com/help/techdoc/index.html>.

[7] ŠNOREK, M. *Neuronové sítě a neuropočítače.* Praha: Vydavatelství ČVUT, 2004. 156 s. ISBN 80-01-02549-7.

[8] VOLNÁ, E. *Neuronové sítě I.* Ostrava: Ostravská univerzita, 2002. 85 s. Elektronický text. Dostupné z:

<http://files.klaska.net/cvut/ns2/volna.pdf>.

[9] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. *Umělá inteligence. (1)* Praha: Academia, 1993. 264s. ISBN 80-200-0496-3.

[10] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. *Umělá inteligence. (4)* Praha: Academia, 2003. 475s. ISBN 80-200-1044-0.

[11] *MATLAB - cvičení 5*[online]. Dostupné z: <http://amber.feld.cvut.cz/ssc/ssc-cv/MATLAB cv5 05.pdf>.

[12] *Neural Network Toolbox* [online]. Dostupné z:

<http://www.caspur.it/risorse/softappl/toolbox/nnet/nnt2som.html>.

[13] LOCHMAN, R. *Implementace neuronové sítě ART: diplomová práce.* Praha: ČVUT, 2006. Elektronický text. Dostupné z:

<https://dip.felk.cvut.cz/browse/pdfcache/lochmar 2006dipl.pdf>.

[14] WEINLICHOVÁ, J., FEJFAR, J. *Využití samoorganizující se neuronové sítě pro hodnocení chování spotřebitele.* Acta universitatis agriculturae et silviculturae Mendelianae Brunensis, 2010, roč. 63, č. 6, s. 625-632.

[15] TUČKOVÁ, J. *Umělé neuronové sítě pomáhají lékařům FN Motol.*Tripod [online]. 2008-03-10. [cit. 29. května 2011]. Dostupné z:

<http://www.3pod.cz/254-Umele-neuronove-site-pomahaji-lekarum-FN-Motol.html.>.

[16] BERKA, P. *Dobývání znalostí z databází.* Praha: Academia, 2003. 366 s. ISBN 80-200-1062-9.

[17] OLEJ, V., HÁJEK, P. *Modelování bonity obcí pomocí Kohonenových samoorganizujících se map a LVQ neuronových sítí.* Scientific papers, 2007, č. 12, s. 143-149. ISBN 978-80-7395-040-8.

[18] MIKULA, V. *Umělé neuronové sítě a fuzzy systémy.* Kunovice: EPI, s. r. o., 2003. 43 s. Elektronický text. Dostupné z:

<http://www.e-skripta.eu/epi/index.php?id=rocnik3-neuronovesiteafuzzylogika/neuronove-site-osnova-kap1lang=czakce=>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ART Neuronová síť ART - Adaptive Resonance Theory

SOM samoorganizující se mapa - Self-organizing map

UNS umělá neuronová síť

α učicí parametr, učicí koeficient

Θ práh neuronu

f přenosová funkce neuronu

$x_i, .., x_n$ vstupy neuronové sítě

$w_j, .., x_n$ hodnoty vah jednotlivých neuronů

SEZNAM PŘÍLOH

A Program **41**

Na přiloženém CD:

data.dat

program.m

3x3.xlsx

5x5.xlsx

3x3-10x10.xlsx

A PROGRAM

```
%close all
%clear all
%clc


P = load ('data.dat');                      % Načtení dat
R=3;
S=3;
Wmin=0.50;                                  % Rozmezí počátečích vah-min
Wmax=0.51;                                  % Rozmezí počátečích vah-max
w =W_min+(Wmax-Wmin) *rand(R*S,2);          % Generovaní vah

w_zaloha = w;                               % Uložení hodnot vah

%w= w_zaloha;

net = nnt2som([0 1;0 1],[R S],w,0.95,0)     % Inicializace sítě

net.trainFcn='trainr'                       % Natavení parametrů sítě
net.adaptFcn='trains'
net.inputWeights{1,1}.learnFcn='learnsom'
net.layers{1}.initFcn='initwb'


plot(P(1,:),P(2,:),'.g','markersize',20)     % Vykreslení vzorů
hold on
plotsom(net.iw{1,1},net.layers{1}.distances)  % Vykreslení neuronů

hold off


net.trainParam.epochs = 100;                  % Počet iterací
net = train(net,P);                           % Spuštění trenování sítě

a = sim(net,P);                               % Uložení čísel neuronů,

vystup = vec2ind(a);
```