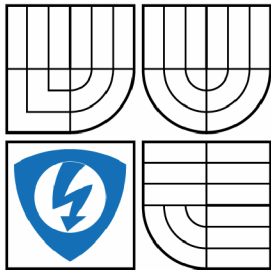


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

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

UMĚLÁ NEURONOVÁ SÍŤ RCE

ARTIFICIAL NEURAL NETWORK RCE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

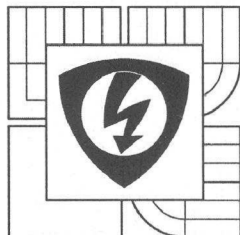
Bc. ALEŠ MACEČEK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. VÁCLAV JIRSÍK, CSc.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Aleš Maceček

Ročník: 2

ID: 115219

Akademický rok: 2012/13

NÁZEV TÉMATU:

Umělá neuronová síť RCE

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s umělou neuronovou sítí RCE a naprogramujte její rozhraní ve zvoleném prostředí.
2. Seznamte se s problematikou popisu a rozpoznávání písmen pomocí neuronových sítí.
3. Vytvořte tréninkovou a testovací množinu písmen.
4. Proveďte rozbor parametrů RCE sítě na správnost rozpoznávání písmen.
5. Porovnejte správnost rozpoznávání písmen pomocí vícevrstvé neuronové sítě s algoritmem učení backpropagation a pomocí RCE sítě.
6. Dosažené výsledky zhodnoťte

DOPORUČENÁ LITERATURA:

Šíma J., Neruda R.: Teoretické otázky neuronových sítí. Matfyzpress, Praha 1996
Patterson D.: Artificial Neural networks - Theory and Applications. Singapore: Prentice Hall 1996
Maceček A.: Rychlost učení vícevrstvé sítě. VUT v Brně, FEKT 2011

Termín zadání: 11.2.2013

Termín odevzdání: 20.5.2013

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

Konzultanti diplomové práce: Ing. Tomáš Hynčica

doc. Ing. Václav Jirsík, CSc.

předseda oborové rady



UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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 částí druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Práce se zabývá umělou neuronovou sítí RCE, zvláště popisem topologie, vlastností a algoritmem učení sítě. Práce obsahuje popis vytvořeného programu uTeachRCE pro učení RCE sítě a programu RCEin3D, vytvořeného pro vizualizaci procesu učení RCE sítě ve 3D prostoru. RCE síť je srovnávána s vícevrstvou umělou neuronovou sítí s algoritmem učení backpropagation při praktické aplikaci rozpoznávání písmen. Pro popis písmen byly zvoleny momenty invariantní na otočení, posun a změnu měřítko obrazu.

Klíčová slova

Neuron, neuronová síť, RCE, Restricted Coulomb Energy, backpropagation, momentové invarianty, rozpoznávání, klasifikace vzorů, vizualizace

Abstract

This paper is focused on an artificial neural network RCE, especially describing the topology, properties and learning algorithm of the network. This paper describes program uTeachRCE developed for learning the RCE network and program RCEin3D, which is created to visualize the RCE network in 3D space. The RCE network is compared with a multilayer neural network with a learning algorithm backpropagation in the practical application of recognition letters. For a descriptions of the letters were chosen moments invariant to rotation, translation and scaling image.

Keywords

Neuron, neural network, RCE, Restricted Coulomb Energy, backpropagation, moment invariants, recognition, pattern classification, visualization

Bibliografická citace:

MACEČEK, A. *Umělá neuronová síť RCE*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 69 s. Vedoucí diplomové práce doc. Ing. Václav Jirsík, CSc.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Umělá neuronová síť RCE jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2013**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Václavovi Jirsíkovi, CSc. a konzultantovi Ing. Tomáši Hynčicovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **20. května 2013**

.....
podpis autora

Obsah

1	Úvod.....	9
2	Umělé neuronové sítě.....	10
2.1	Formální neuron.....	10
2.2	Topologie neuronové sítě.....	11
2.3	Učení neuronové sítě.....	12
2.4	Vybavování neuronové sítě.....	13
2.5	Tréninková a testovací množina.....	13
3	Neuronová síť RCE.....	14
3.1	RBF neurony.....	14
3.2	Topologie RCE sítě.....	15
3.3	Vlastnosti RCE sítě.....	16
3.4	Algoritmus učení.....	17
3.5	Rešerše prací zabývajících se aplikací RCE sítě.....	22
4	Popis objektů.....	24
4.1	Příznakový vektor.....	24
4.2	Momentový popis.....	25
4.2.1	Normalizace momentů.....	27
5	Program uTeachRCE pro učení RCE sítě.....	28
5.1	Popis programu.....	28
5.2	Vstupní data programu.....	29
5.3	Základní struktura programu.....	30
5.4	Funkce programu.....	31
6	Rozpoznávání písmen pomocí RCE sítě.....	32
6.1	Použité vzory písmen.....	32
6.2	Rozpoznávání písmen množiny B.....	34
6.3	Rozpoznávání písmen množiny C.....	37
6.4	Srovnání RCE sítě a vícevrstvé sítě s učícím algoritmem backpropagation.....	42
6.5	Rozpoznávání písmen pomocí fuzzy RCE sítě.....	47
7	Program RCEin3D pro vizualizaci učení RCE sítě.....	48
7.1	Návrh prostředí pro realizaci programu.....	48
7.2	Zobrazení RCE sítě programem při klasifikaci do tří tříd.....	49
7.3	Popis programu.....	50
7.4	Ovládání programu.....	54

7.5	Vstupní data programu.....	55
7.6	Funkce programu	55
7.7	Zobrazení RCE sítě programem při klasifikaci do více tříd.....	57
7.8	Návrh úloh do počítačových laboratoří předmětu Umělá inteligence.....	59
7.8.1	Úloha č.1: Základní funkce RCE sítě.....	59
7.8.2	Úloha č.2: Prstény	59
7.8.3	Úloha č.3: Učení tří shluků	60
7.8.4	Úloha č.4: Učení jednoho shluku	61
7.8.5	Úloha č.5: Závislost počtu skrytých neuronů na počtu tréninkových bodů	62
8	Závěr.....	64

1 ÚVOD

Umělé neuronové sítě jsou jedním z mnoha výpočetních modelů používaných v umělé inteligenci. Avšak vlastnosti umělé neuronové sítě je vhodné využít i v dalších oborech, proto je jim věnována stále větší pozornost. Umělá neuronová síť je složena z několika vzájemně propojených procesorů, které nazýváme umělé neurony, protože velmi zjednodušeně modelují biologické neurony.

Druhá kapitola se zabývá popisem umělých neuronových sítí a jejich základním stavebním prvkem, formálním neuronem. Dále je v kapitole uveden matematický popis formálního neuronu, jeho definice i s příklady topologií a učení umělých neuronových sítí.

Třetí kapitola obsahuje informace nutné k pochopení umělé neuronové sítě typu RCE. V této kapitole je uveden popis RBF neuronů, které se v této síti používají, dále je zde uvedena topologie a vlastnosti RCE sítě. V kapitole je také podrobně popsán způsob učení RCE sítě s uvedením algoritmu učení a grafickými ukázkami učení sítě. Na závěr kapitoly je uvedena rešerše prací zabývajících se aplikací RCE sítě.

Ve čtvrté kapitole je zhodnocení možností popisu vzorů písmen. Kapitola obsahuje definici společně se vzorci invariantních momentů, které byly zvoleny, jako nejvhodnější příznaky pro popis písmen.

Pátá kapitola obsahuje popis programu uTeachRCE pro učení RCE sítě: základní popis programu, načítání vstupních dat a popis funkcí programu.

V šesté kapitole je uvedeno testování kvality rozpoznání písmen pomocí RCE sítě a zkoumání nastavení RCE sítě na kvalitu rozpoznání písmen. V průběhu celé kapitoly jsou zjištěné výsledky porovnávány s vícevrstvou neuronovou sítí s algoritmem učení backpropagation, kterou se zabývala bakalářská práce [Mac11]. Na závěr kapitoly je uvedeno srovnání těchto dvou neuronových sítí.

Sedmá kapitola obsahuje popis programu RCEin3D umožňující vizualizaci učení RCE sítě: základní popis programu, způsob ovládání programu a načítání vstupních dat, popis funkcí programu. Dále je v kapitole uveden popis pěti úloh navržených do počítačových laboratoří předmětu Umělá inteligence.

2 UMĚLÉ NEURONOVÉ SÍTĚ

Umělé neuronové sítě, dále jen neuronové sítě, je matematický model, který má za vzor biologické neuronové sítě. Základním stavebním prvkem neuronových sítí je formální neuron. Jejich základní vlastností je učit se, znalosti jsou ukládány především prostřednictvím vah jednotlivých vazeb.

Neuronové sítě se používají v případě, kdy nejsme schopni popsat veškeré matematické vztahy odehrávající se v daném procesu. Jejich použití je také vhodné v případě, kdy matematický model sestavíme, ale jeho řešení je velice složité.

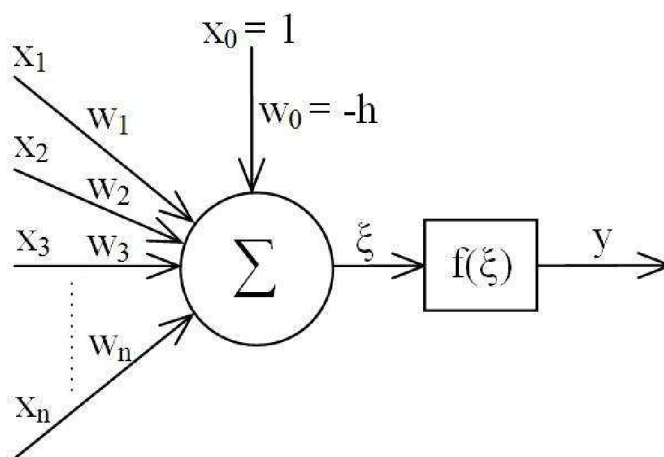
Dynamika celé neuronové sítě je dělena do tří podskupin: organizační (topologie sítě), adaptivní (učení sítě) a aktivní (vybavování sítě). Popisem jednotlivých dynamik dostáváme různé modely neuronových sítí pro řešení určitých tříd úloh.

V praxi jsou umělé neuronové sítě používané pro predikci (počasí, energetiky, ekonomie aj.), rozpoznávání a rekonstrukce obrazů (otisky, podpisy aj.), analýzu (EKG, řeči aj.) a v mnoha dalších oborech.

Problematika umělých neuronových sítí byla čerpána z literatury [ŠN96], [Bíl95], [Šmo02] a také v anglicky psané literatuře [DHS00], [Scha97], [Fau94].

2.1 Formální neuron

Formální neuron, dále jen neuron, je matematický model. Má konečný počet vstupů x_1, \dots, x_n a jeden výstup y . Váhy jednotlivých vstupů w_1, \dots, w_n určují citlivost, s jakou vstupy působí na výstup. Váhy mohou nabývat kladných i záporných hodnot. Struktura formálního neuronu je schematicky znázorněna na Obr. 1.



Obr. 1: Formální neuron

Pokud vnitřní potenciál neuronu ξ , který je vyjádřen sumou vstupů s příslušnými vahami, překročí prahovou hodnotu h , tak dochází k indikaci výstupu y . Častěji se používá formální úprava dle rovnice (2.1), kdy vlastní práh je brán, jako váha ($w_0 = -h$) dalšího vstupu s konstantní hodnotou ($x_0 = 1$). Takto získáme neuron s prahovou hodnotou v nule.

$$\xi = -h + \sum_{i=1}^n x_i w_i = x_0 w_0 + \sum_{i=1}^n x_i w_i = \sum_{i=0}^n x_i w_i \quad (2.1)$$

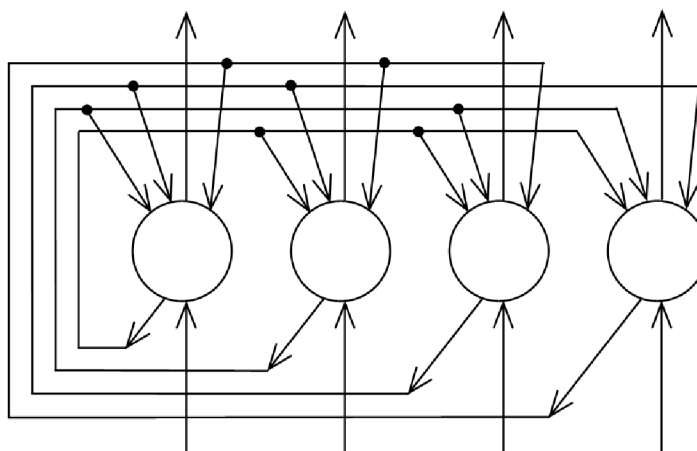
Přenosová (aktivační) funkce $f(\xi)$ slouží k převádění vnitřního potenciálu neuronu do definovaného oboru výstupních hodnot. Přenosová funkce může být lineární nebo nelineární. Nelineární funkce jsou nejvíce používány: ostrá nelinearita, standardní (logická) sigmoida a hyperbolický tangens. Neuron s prahovou hodnotou v nule a přenosovou funkcí ostrá nelinearita má definované hodnoty výstupu dle rovnice (2.2).

$$y = f(\xi) = \begin{cases} 1 & \text{pro } \xi \geq 0 \\ 0 & \text{pro } \xi < 0 \end{cases} \quad (2.2)$$

2.2 Topologie neuronové sítě

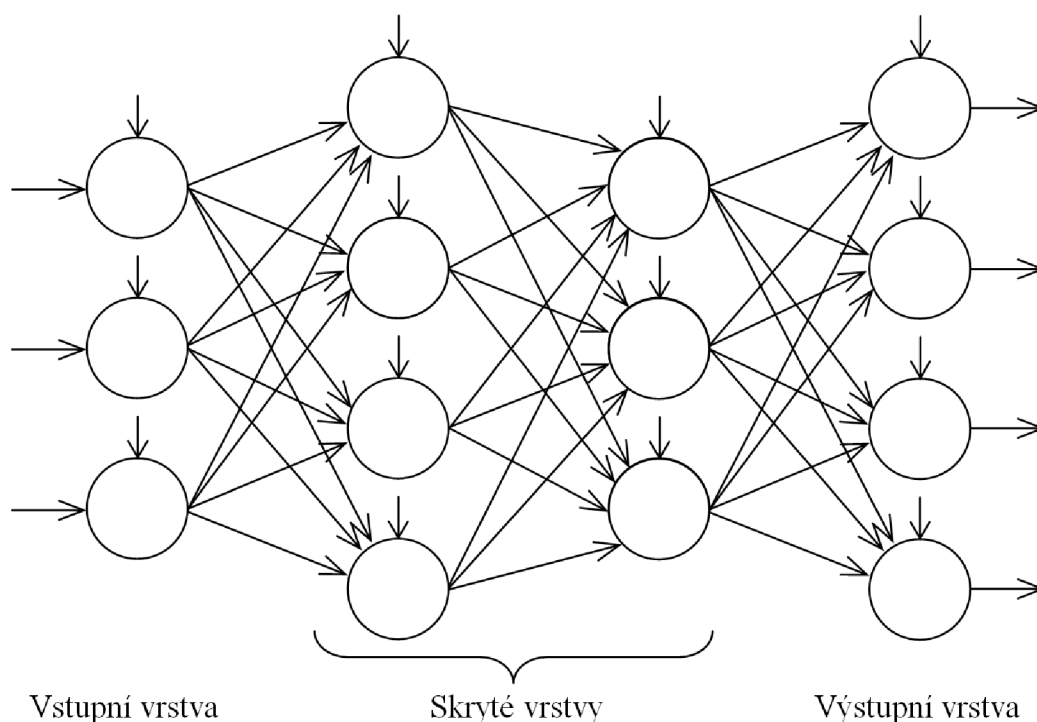
Počet neuronů a jejich vzájemné propojení určuje topologii neuronové sítě. Rozlišujeme dva základní typy topologií: cyklická (rekurentní) a acyklická (dopředná).

V cyklické topologii skupiny neuronů vytvářejí kruh (cyklus). Nejjednodušším typem je zpětná vazba neuronu. Naopak nejvíce cyklů je v úplné topologii cyklické neuronové sítě, kde je výstup každého neuronu vstupem do všech ostatních neuronů. Příklad cyklické topologie je na Obr. 2, jedná se o Hopfieldovu síť.



Obr. 2: Příklad cyklické topologie (Hopfieldova síť)

Acyklická topologie se vyznačuje tím, že se informace v síti šíří pouze jedním směrem. Neurony jsme schopni rozdělit do vrstev (např. nad sebou) a vazby mezi neurony vedou pouze z nižších vrstev do vyšších tak, že mohou přeskaakovat libovolný počet vrstev. Topologie sítě se zapisuje, jako počty neuronů v jednotlivých vrstvách oddělených pomlčkou směrem od vstupní k výstupní vrstvě (např. 3-4-3-4). Příklad acyklické topologie je na Obr. 3, jedná se o vícevrstvou neuronovou síť.



Obr. 3: Příklad acyklické topologie (vícevrstvá neuronová síť)

2.3 Učení neuronové sítě

Před začátkem učení jsou váhy nastaveny na počáteční hodnoty, které bývají náhodně zvoleny z určitého intervalu. Během samotného učení dochází ke změnám hodnot vah a prahů, tím se síť adaptuje na řešení daného problému. Topologie sítě se u většiny neuronových sítí během učení nemění. V procesu učení jsou síti předkládány tréninkové vstupy, na které síť poskytuje odezvy. Používáme dva typy učení:

- Učení s učitelem: máme požadované výstupy sítě pro jednotlivé tréninkové vstupy. Tudíž je možné vypočítat rozdíl mezi požadovaným a skutečným výstupem. S touto chybou pracuje algoritmus, který mění váhy všech spojů pro jednotlivé tréninkové vstupy.

- Učení bez učitele: není žádná možnost zjištění správnosti výstupu sítě. Algoritmus učení je navržen tak, že ve vstupních datech hledá vzorky, které mají společné vlastnosti [Šno02]. Učení bez učitele je samoorganizací.

Učení neuronové sítě se z hlediska průběhu nastavení vah dělí na proces iterativní a jednorázový. Toto rozdělení není nijak spojeno s rozdělením učení na učení s učitelem a bez učitele.

2.4 Vybavování neuronové sítě

Vybavování neuronové sítě zjišťuje odezvu již naučené neuronové sítě na testovací vzory. Proces vybavování neuronové sítě se nejprve používá k ověření kvality rozpoznání u naučené sítě, poté se využívá při práci s neuronovou sítí v praxi.

Pro vybavování se tak, jako u učení, používá rozdělení na proces iterativní a jednorázový. Platí následující pravidlo: pokud je učení sítě iterativní/jednorázový proces, tak vybavování sítě je proces jednorázový/iterativní.

2.5 Tréninková a testovací množina

Vstupní množina obsahuje všechny vstupní vzory (vektory) předkládané síti. Vstupní množinu rozdělujeme na tréninkovou a testovací množinu. Tréninková množina by měla co nejlépe vystihovat vstupní množinu. Pomocí tréninkové množiny učíme neuronovou síť, tréninková množina tedy ovlivňuje rychlost a kvalitu naučení.

Kvalitu naučení neuronové sítě kontrolujeme pomocí testovací množiny. Testovací množina musí být také reprezentativní, ale neměla by obsahovat stejné vzory jako tréninková množina (nedokázali bychom posoudit kvalitu naučení).

Rychlost a úspěšnost učení lze také ovlivnit strategií výběru vzorů z tréninkové množiny (sekvenční, náhodné, některé vzory se mohou vybírat častěji). Cyklus je časový interval, ve kterém jsou alespoň jednou vybrány všechny vzory z tréninkové množiny. K naučení sítě může být zapotřebí stovky až tisíce cyklů dle [Mac11].

3 NEURONOVÁ SÍŤ RCE

Neuronová síť typu RCE (Restricted Coulomb Energy) je třívrstvá neuronová síť, která se řadí do kategorie učení s učitelem. Topologie sítě je popsána v kapitole 3.2. Vlastnosti RCE sítě jsou popsány v kapitole 3.3. V rámci kapitoly 3.5 je uvedena rešerše prací zabývajících se aplikacemi RCE sítě.

RCE neuronová síť byla poprvé prezentována v roce 1982 v práci s názvem „A neural model for category learning“ od autorů Reily D. L., Cooper L. N. a Elbaum C., viz [RCE82]. RCE síť byla studovaná z literatury [Has95], [Lee89], [Xie03].

3.1 RBF neurony

RCE síť je nejvíce podobná neuronové síti RBF (Radial Basic Functions), a to v použití neuronů s radiální bázovou funkcí (dále jen RBF neurony) ve skryté vrstvě, jejichž význam pro fungování neuronových sítí bude vysvětlen v dalších odstavcích.

Vnitřní potenciál RBF neuronů se vypočítá dle rovnice (3.1), ve kterém je použita euklidovská metrika. Naopak neuron popsaný v kapitole 2.1 používá skalární metriku. Vnitřní potenciál RBF neuronů tedy vyjadřuje vzdálenost mezi vstupním vektorem $\underline{x} = (x_1, \dots, x_i, \dots, x_n)$ a vektorem $\underline{c} = (c_1, \dots, c_i, \dots, c_n)$, který reprezentuje střed RBF neuronu. Použitím euklidovské metriky je zaručeno, že všechny vstupní vektory \underline{x} při konstantním vnitřním potenciálu leží kolem jediného bodu (středu) určeného vektorem \underline{c} , který zastupuje váhové spojení mezi RBF neuronem a vstupní vrstvou: vstupní vektory tedy leží na hyperkouli se středem v bodě \underline{c} [Šno02].

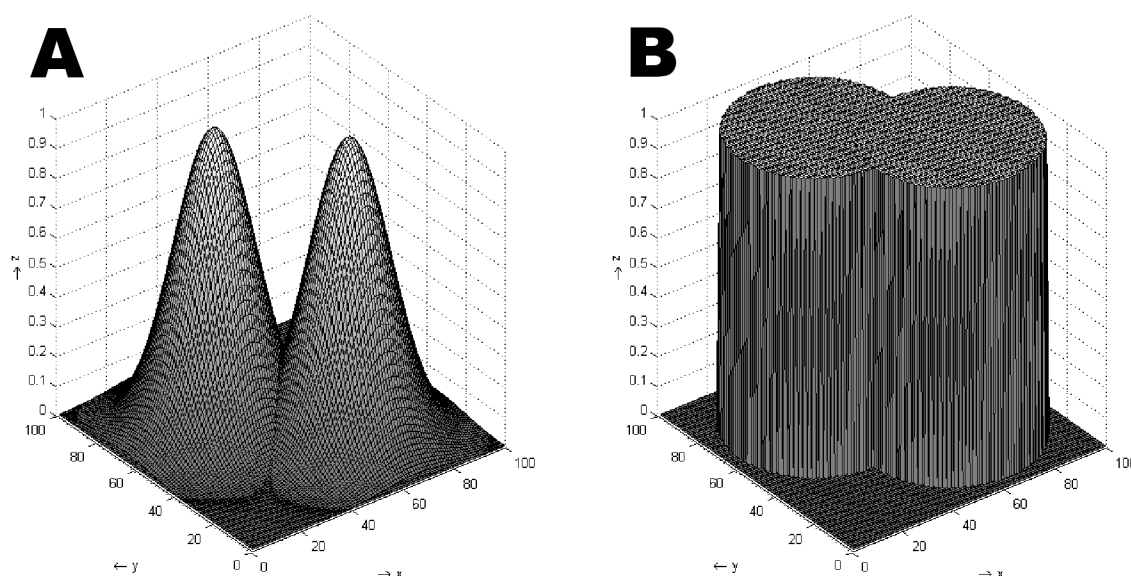
$$\xi = \sqrt{\sum_{i=1}^n (c_i - x_i)^2} \quad (3.1)$$

Hlavní rozdíl mezi RBF neurony a neurony používající skalární metriku je v rozdělení vstupního prostoru. RBF neurony rozdělují vstupní prostor na menší oblasti (hyperkoule), jejichž skládáním mohou být reprezentovány komplexní oblasti ve vstupním prostoru. Kdežto neurony se skalární metrikou rozdělují vstupní prostor na dva podprostory. Skládání jednotlivých podprostorů do složitějších oblastí může být někdy problematické.

Jednou z používaných přenosových funkcí u RBF neuronů je Gaussova funkce, která vyjadřuje míru příslušnosti vstupního vektoru ke středu hyperkoule (pokud je výstup RBF neuronu roven jedné, tak je vstupní vektor totožný jako vektor reprezentující střed). V této práci však budeme využívat nespojitou přenosovou funkci (3.2), kde parametr R určuje poloměr hyperkoule. Nespojitá přenosová funkce v podstatě modeluje požadované hyperkoule.

$$y = f(\xi) = \begin{cases} 1 & \text{pro } \xi \leq R \\ 0 & \text{pro } \xi > R \end{cases} \quad (3.2)$$

Při použití nespojité přenosové funkce je výstup neuronu roven jedné, pokud vstupní vektor leží uvnitř hyperkoule, jinak je nulový. Oblast, kdy je neuron aktivní, se také nazývá sférou vlivu [Šno02]. Na Obr. 4 jsou znázorněny popsané přenosové funkce pro dva RBF neurony. Při promítnutí Obr. 4B (nespojité přenosové funkce) do roviny xy dostaneme překrývající se kružnice (hyperkoule v dvourozměrném prostoru) o poloměru R.



Obr. 4: Přenosové funkce pro dva RBF neurony, (A) Gaussova funkce, (B) nespojitá funkce

3.2 Topologie RCE sítě

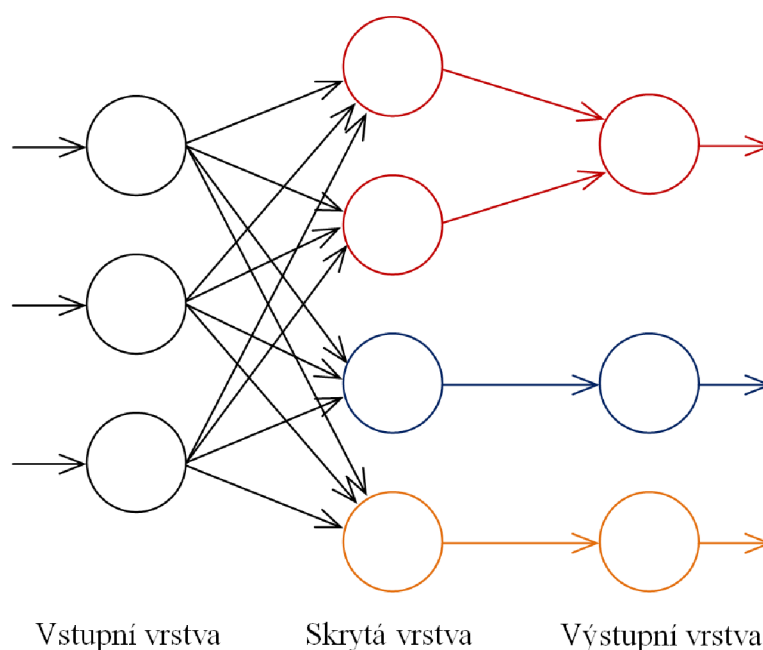
Neuronová síť typu RCE se skládá ze tří vrstev (vstupní, skrytá a výstupní), topologie RCE sítě patří do skupiny acyklických topologií. V každé vrstvě mají neurony odlišný typ přenosové funkce.

Vstupní vrstva slouží k distribuci vstupních signálů z okolí na všechny neurony ve skryté vrstvě (použita lineární přenosová funkce), propojení mezi vstupní a skrytou vrstvou je tedy úplné. Počet vstupních neuronů je závislý na počtu vstupních signálů přiváděných do neuronové sítě.

Skrytá vrstva obsahuje RBF neurony, které byly popsány v kapitole 3.1. Počet skrytých neuronů se dynamicky navyšuje v průběhu učení tréninkových vzorů, viz kapitola 3.4.

Počet neuronů ve výstupní vrstvě odpovídá počtu klasifikovaných tříd. Neuron výstupní vrstvy je spojen pouze s neurony skryté vrstvy, které reprezentují klasifikovanou třídu určenou tímto výstupním neuronem. Neuron ve skryté vrstvě může být připojen pouze k jednomu výstupnímu neuronu. Přenosová funkce výstupních neuronů je tedy logický součet (OR). Propojení mezi skrytou a výstupní vrstvou je částečné a hodnoty váhových spojení nabývají pouze hodnot 0 a 1.

Na Obr. 5 je ukázána RCE síť provádějící klasifikaci do tří tříd (propojení skrytých a výstupních neuronů v rámci tříd je odlišeno barvami)



Obr. 5: Neuronová síť typu RCE

Při dynamickém navyšování počtu skrytých neuronů není potřeba experimentálně určovat jejich počet jako u jiných neuronových sítí. Z hlediska učení neuronové sítě neexistuje žádné omezení pro počet neuronů ve skryté vrstvě, omezení počtu skrytých neuronů vzniká pouze v závislosti na použitém hardwaru, jak je uvedeno v [Xie03].

3.3 Vlastnosti RCE sítě

Neuronová RCE síť použitá jako klasifikátor je vhodná k rozpoznání lineárně i nelineárně oddělitelných oblastí, ale zároveň je schopna také rozpoznat neoddělitelné oblasti [YX02], [YGX01]. Těchto vlastností je zejména dosaženo pomocí RBF neuronů ve skryté vrstvě.

RCE síť může také sloužit jako aktivní paměť [YX02], toho je docíleno díky dynamickému přidávání skrytých i výstupních neuronů během učení sítě.

Jak uvádí literatura [TX10] tak se RCE síť vyznačuje mnohými výhodami oproti ostatním neuronovým sítím (např. RBF síť, vícevrstvá síť s učením backpropagation): kratší čas nutný k naučení tréninkových vzorů, jednodušší učení tréninkových vzorů, nikdy učení sítě neuváže v lokálním minimu.

3.4 Algoritmus učení

Na počátku učení RCE sítě není v síti žádný skrytý neuron, cílem učení je tedy vytvářet nové skryté neurony případně modifikovat ty, co jsou již vytvořeny v závislosti na předkládaných tréninkových vzorech.

Učení RCE sítě řadíme do kategorie učení s učitelem. Avšak v této síti nedochází k aktualizaci vah v síti pomocí algoritmu počítající s chybou zjištěnou na výstupu sítě, jak tomu bylo například u učení vícevrstvé sítě s algoritmem učení backpropagation [Mac11]. Znalost o třídě, do které patří tréninkový vzor, je využívána k tvorbě chybových signálů, které nabývají následujících hodnot:

- Chyba = 1; Tréninkový vzor nevyvolal odezvu na žádném výstupním neuronu. Do sítě je zařazen nový skrytý neuron (střed hyperkoule je umístěn do bodu určeným tréninkovým vzorem), který je připojen na adekvátní výstupní neuron.
- Chyba = -1; Tréninkový vzor vyvolal odezvu na výstupním neuronu určujícím odlišnou výstupní třídu. Dochází ke zmenšení poloměru hyperkoule u skrytých neuronů, které jsou aktivní pro daný tréninkový vzor.
- Chyba = 0; Tréninkový vzor vyvolal odezvu na výstupním neuronu určujícím správnou výstupní třídu. Nedochází k žádným změnám v síti.

Tyto chybové signály, které jsou také uvedeny v [DSCUM93], jsou zakomponovány do níže uvedeného algoritmu. Chybový signál Chyba = -1 není zakomponován do algoritmu v přesném znění, ale je mírně modifikován.

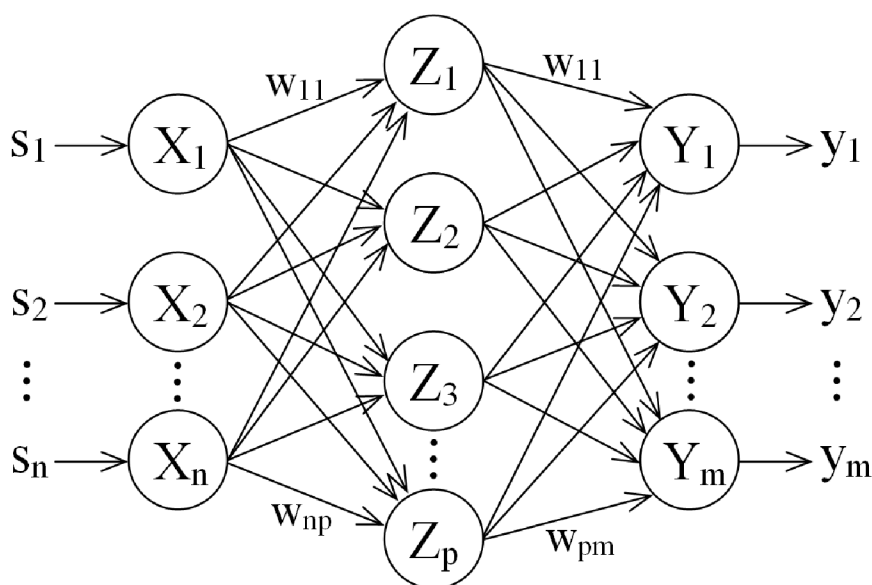
Hodnoty výstupních signálů RCE sítě jsou určeny použitým režimem. U RCE sítě rozeznáváme dva režimy: rychle reagující režim (fast response mode) a pravděpodobnostní režim (probability response mode). V našem algoritmu je použit rychle reagující režim, tzn. výstupní hodnoty neuronů ve výstupní vrstvě nabývají pouze hodnot 0 nebo 1 (neuron je aktivní).

Zjednodušený výpočet výstupních signálů při pravděpodobnostním režimu je následující: Nejprve je nutno vypočítat pravděpodobnosti skrytých neuronů, které udávají, jak moc se tréninkový vzor podobá středu hyperkoule. Dále je potřeba počítat kolikrát byly jednotlivé skryté neurony aktivní při předkládání tréninkových vzorů. Z těchto dvou parametrů je určena pravděpodobnost výstupních neuronů. Vzorce

potřebné pro pravděpodobnostní režim jsou uvedeny v knize [Xie03] anebo v článku [YX02].

V některých literaturách zahrnuje algoritmus učení RCE sítě i dynamické přidávání neuronů ve výstupní vrstvě v závislosti na předkládaných tréninkových vzorech. V dále popsaném algoritmu se spokojíme se statickým počtem neuronů ve výstupní vrstvě, protože v naší úloze budeme dopředu znát počet rozpoznávaných tříd.

Jako hlavní předloha při sestavování algoritmu byl použit poloformální popis algoritmu učení RCE sítě uvedeném v literatuře [Zbo12], který byl rozepsán do jednotlivých kroků a doplněn o podstatné matematické vztahy.



Obr. 6: Obecné značení RCE sítě

Algoritmus učení:

- Krok 1: Nastavit počet neuronů vstupní vrstvy n na počet vstupních signálů. Počet neuronů výstupní vrstvy m nastavit na počet klasifikovaných tříd.
- Krok 2: Parametr C nastavit do nuly (parametr C určuje zda byl tréninkový vzor sítí správně klasifikován). Pokud je parametr p nulový (nebyl doposud vytvořený žádný skrytý neuron), pokračovat na krok 8 (vytvoření nového skrytého neuronu). Pro každou dvojici t a v (číselné označení klasifikované třídy v , do které patří vstupní tréninkový vektor t) provádět kroky 3 až 8.
- Krok 3: Aktivovat neurony vstupní vrstvy (X_i , $i = 1, \dots, n$) tak, že $x = t$. Vstupní vrstva slouží pouze pro distribuci vstupních signálů do skryté vrstvy.
- Krok 4: Pro každý neuron skryté vrstvy (Z_j , $j = 1, \dots, p$) provádět kroky 5 až 6.

Krok 5: Vypočítat vnitřní potenciál (euklidovská vzdálenost) skrytého neuronu Z_j podle rovnice (3.3) a následně určit jeho výstupní hodnotu pomocí přenosové funkce (3.4). Parametr R_j udává poloměr hyperkoule neuronu Z_j .

$$z_j(in) = \sqrt{\sum_{i=1}^n (w_{ij} - x_i)^2} \quad (3.3)$$

$$z_j = \begin{cases} 1 & \text{pro } z_j(in) \leq R_j \\ 0 & \text{pro } z_j(in) > R_j \end{cases} \quad (3.4)$$

Krok 6: Jestliže platí $z_j = 1$ (tzn. aktuální tréninkový vzor \underline{t} se nachází uvnitř hyperkoule určené neuronem Z_j) a zároveň platí rovnost $G_j = v$ (neuron Z_j je spojen s výstupním neuronem Y_k , jenž zastupuje stejnou třídu jako tréninkový vzor), tak nastavit parametr C na hodnotu 1. Pak pokračovat dalším tréninkovým vzorem v kroku 4.

Krok 7: Jestliže platí rovnost $C = 0$, tak pokračovat krokem 8, jinak pokračovat dalším tréninkovým vzorem v kroku 2.

Krok 8: Navýšit počet skrytých neuronů (3.5). Vytvořit nový skrytý neuron Z_p s maximálním poloměrem hyperkoule (3.6) a informací o tom, jakou třídu reprezentuje (3.7). Podle rovnice (3.8) nastavit váhové spojení mezi vstupní vrstvou (X_i , $i = 1, \dots, n$) a novým neuronem Z_p (tzn. nově vytvořená hyperkoule bude mít souřadnice středu určené vstupním vektorem \underline{x}). Podle rovnice (3.9) nastavit váhové spojení mezi novým neuronem Z_p a výstupní vrstvou (Y_k , $k = 1, \dots, m$).

$$p = p + 1 \quad (3.5)$$

$$R_p = R_{MAX} \quad (3.6)$$

$$G_p = v \quad (3.7)$$

$$w_{ip} = x_i \quad (3.8)$$

$$w_{pk} = \begin{cases} 1 & \text{pro } k = G_p \\ 0 & \text{pro } k \neq G_p \end{cases} \quad (3.9)$$

Zkontrolovat zda nedochází k překrývání hyperkoule neuronu Z_p s hyperkoulami ostatních neuronů skryté vrstvy (Z_j , $j = 1, \dots, p-1$). Jestliže k překrývání dochází a zároveň neurony zastupují odlišné třídy ($G_p \neq G_j$), tak poloměry R_p a R_j nastavit na polovinu vzdálenosti jejich středů dle rovnice (3.10). Pokud by některý z poloměrů byl menší než

nově vypočítaná hodnota poloměru podle rovnice (3.10), tak se jeho hodnota měnit nebude.

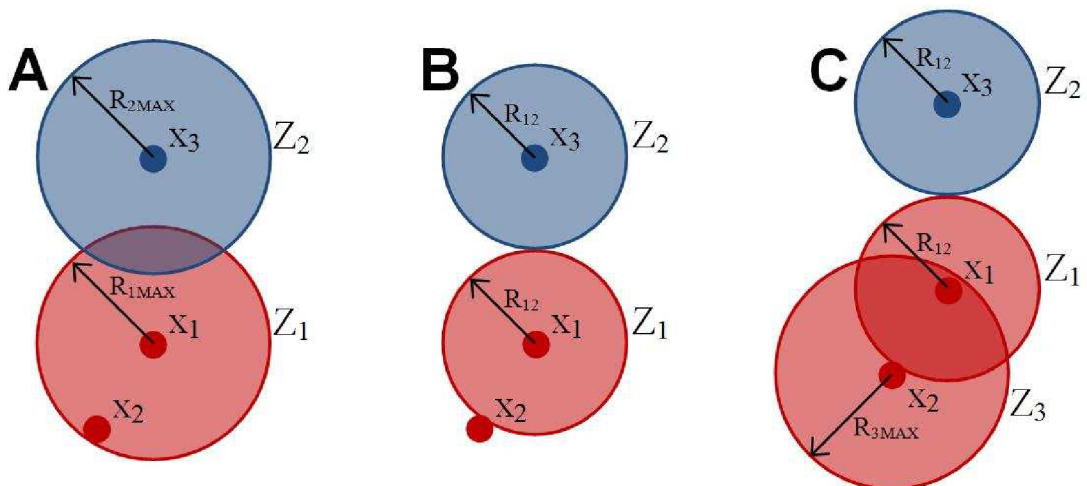
$$R_p = R_j = \frac{\sqrt{\sum_{i=1}^n (w_{ij} - w_{ip})^2}}{2} \quad (3.10)$$

Krok 9: Opakovat kroky 2 až 8 tak dlouho, dokud v síti bude docházet ke změnám.

V následujících odstavcích budou popsány kroky osm a devět učícího algoritmu společně s obrázky popisující učení RCE sítě. V kroku 9 je uvedena podmínka, při které je RCE síť naučena na tréninkové vzory (nesmí docházet ke změnám v síti). Na první pohled by se dalo říci, že síť bude naučena během jednoho cyklu, protože dochází k přidávání skrytých neuronů pro tréninkové vzory, které nezpůsobily aktivitu žádného výstupního neuronu.

Tento předpoklad však platí jen v ojedinělých případech (např. pokud by se odlišné třídy vzorů nacházely daleko od sebe v příznakovém prostoru). Na Obr. 7 je ukázán příklad, proč je důležité opakovaně předkládat tréninkové vzory, dokud nebude docházet ke změnám v síti.

Na Obr. 7A je situace, kdy byl předložen první červený vzor x_1 , pro který byl vytvořen nový skrytý neuron Z_1 s poloměrem R_{1MAX} , poté byl předložen druhý červený vzor x_2 , který spadá do sféry vlivu neuronu Z_1 , a nakonec byl předložený modrý vzor x_3 , pro který byl vytvořen nový skrytý neuron Z_2 s poloměrem R_{2MAX} .



Obr. 7: První příklad učení sítě RCE

Na Obr. 7B je znázorněna situace po zmenšení poloměru obou skrytých neuronů na hodnotu $R_{12} < R_{MAX}$, aby nedocházelo k překrývání kružnic u odlišných tříd vzorů. Hodnota R_{12} odpovídá polovině vzdáleností jejich středů dle rovnice (3.10).

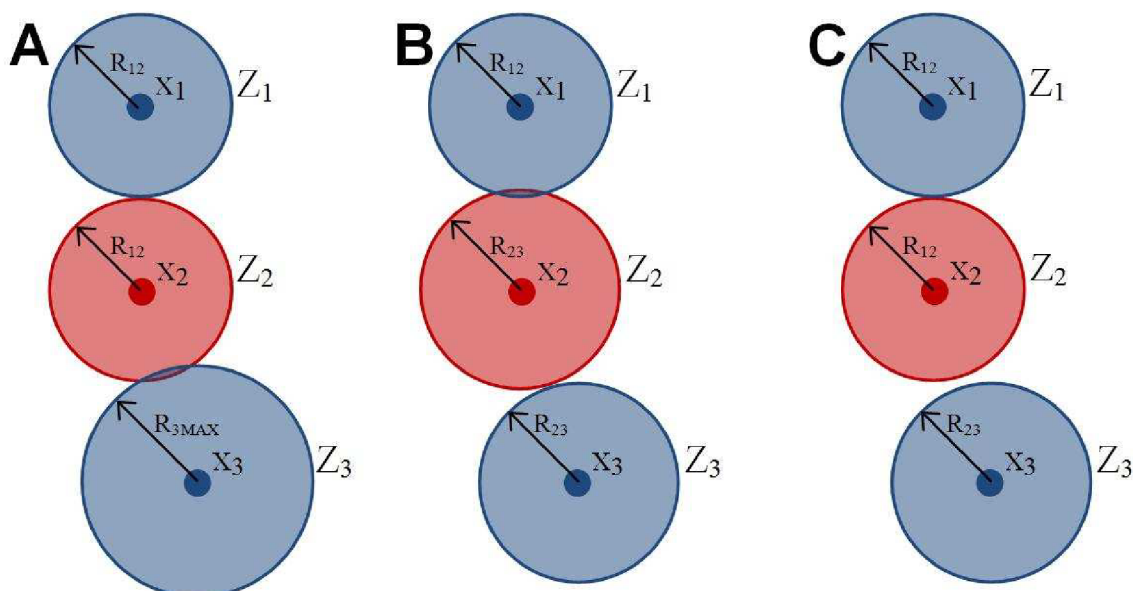
Pro vzor x_2 musel být v druhém cyklu vytvořen nový skrytý neuron Z_3 viz Obr. 7C, protože vlivem zmenšení poloměrů už nespadal do sféry vlivu neuronu Z_1 . V tomto příkladě byla síť naučena za dva cykly.

Dále přiblížíme možné problémy při zmenšování poloměru hyperkoule u nově přidaného skrytého neuronu v osmém kroku algoritmu.

Na Obr. 8A je znázorněna situace, kdy pro modrý vzor x_1 a červený vzor x_2 byly během učení přidány neurony Z_1 a Z_2 . Jejich poloměr byl zmenšen na hodnotu R_{12} , aby nedocházelo k překrývání kružnic. Následně byl předložen modrý vzor x_3 , pro který byl vytvořen neuron Z_3 s poloměrem R_{3MAX} .

Pokud bychom poloměry neuronů Z_2 a Z_3 nastavili na polovinu jejich vzdáleností R_{23} dle rovnice (3.10), tak bychom vyřešili překrytí kružnic u neuronů Z_2 a Z_3 , avšak překrytí kružnic by vzniklo u neuronů Z_1 a Z_2 (toto překrytí by muselo být vyřešeno během dalšího učícího cyklu). Tato situace je na Obr. 8B.

Abychom předešli této situaci, tak je v osmém kroku zavedena podmínka, že poloměry se budou měnit podle rovnice (3.10) pouze za předpokladu, že jejich hodnota bude větší než nově vypočítaná hodnota. Zavedení této podmínky je znázorněno na Obr. 8C, kdy poloměr neuronu Z_3 byl podle rovnice (3.10) změněn na hodnotu R_{23} a poloměr Z_2 zůstal na hodnotě R_{12} ($R_{12} < R_{23}$). Nyní je síť správně naučena na tyto tři vzory.



Obr. 8: Druhý příklad učení sítě RCE

3.5 Rešerše prací zabývajících se aplikací RCE sítě

Neuronová síť RCE se nejčastěji aplikuje v úlohách, které řeší klasifikaci nebo segmentaci. Při klasifikaci se využívá jedné z vlastností RCE sítě, a to použití sítě jako aktivní paměti. Klasifikaci alfanumerických znaků pomocí neuronových sítí bylo vždy věnováno hodně pozornosti a úsilí, řešení této problematiky pomocí RCE sítě je v pracích [YX02] a [HSIII96]. RCE síť používaná pro segmentaci dosahuje velmi dobrých výsledků [YGX01] a [SKR11], a to především díky použití RBF neuronů. Síť, která je naučena na pixel obrazu reprezentovaným některým barevným modelem, správně segmentuje i pixely, jejichž barva se trochu odlišuje od barvy naučeného pixelu (míra odlišnosti je určena poloměrem hyperkoule).

V práci [YX02], která nese název „An incremental representation of conceptual symbols using RCE neural network“, autoři využívají RCE neuronovou síť jako aktivní paměť pro robota, který má být schopen rozpoznávat číslice a malá nebo velká písmena. Jednotlivé alfanumerické znaky jsou segmentovány a normalizovány na stejnou velikost. Sedmdesát šest neuronů vstupní vrstvy odpovídá počtu vstupních signálů, které reprezentují počet pixelů tvořících alfanumerický znak ve všech řádcích (48) a sloupcích (28). Po dokončení učení se účinnost rozpoznání jednotlivých číslic pohybovala od 89,6% do 96,2%. Pro rozpoznání číslic nebyla vytvořena jedna velká síť, ale pro každou číslici byla vytvořena jedna RCE síť. Počet skrytých neuronů v těchto sítích se pohyboval od 22 do 123.

Klasifikací alfanumerických znaků pomocí RCE sítě se zabývali také autoři v článku „An adaptive neural network: Application to character recognition on X-ray films“ [HSIII96]. Počet vstupních neuronů je 180, na každý neuron je připojen jeden pixel vstupního obrazu 12x15 pixel. Síť je schopna rozpoznat číslice a písmena (S, T, M), počet výstupních neuronů je 13. Během učení RCE sítě bylo potřeba vytvořit 55 skrytých neuronů a úspěšnost rozpoznání je 97,8%.

Rozpoznávání seismických signálů je popsáno v práci „Classifying seismic signals via RCE neural network“ [SLM90]. Pro klasifikaci byla použita RCE síť s 64 vstupními a třemi výstupními neurony, u které byla dosažena úspěšnost rozpoznání 87,9%. Autoři se v práci také zabývali kombinací dvou RCE sítí, které měly 64 neuronů ve vstupní a tři neurony ve výstupní vrstvě. Klasifikovaný signál byl nejprve zpracován první sítí, pokud byl výsledek jeho klasifikace neurčitý, tak byl zpracován ještě druhou RCE sítí. Před klasifikací v druhé síti byl neurčitý klasifikovaný signál upraven tak, aby se zvýraznil jeho průběh. Použitím druhé RCE sítě vzrostla úspěšnost rozpoznání na hodnotu 91,7%.

Další aplikace RCE sítě je popsána v práci [YGX01] s názvem „Hand image segmentation using colour and RCE neural network“, kde se autoři zabývají segmentací lidské ruky v barevných snímcích. Autoři pro segmentaci využívali barevné modely RGB, HSI a L^*a^*b (RCE síť má tedy tři vstupní neurony), které jsou často využívané

v oboru počítačové vidění. Reprezentované výsledky ukazují, že barevný model L*a*b je pro segmentaci nejvhodnější (při jeho použití bylo dosaženo nejlepších hodnot a nejkratšího učení sítě). Autoři také zkoumali, jak se změní barevné rozložení kůže u různých etnik. Segmentace byla účinná i přesto, že tvar ruky nebyl přesně daný a pozadí scény nebylo konstantní, a dosahovala 95% správného rozpoznání.

Problematikou segmentace ruky na snímcích se také zabývá práce „A restricted coulomb energy (RCE) neural network system for hand image segmentation“ [SKR11]. Síť má opět tři vstupní neurony, na které jsou přivedeny složky barevného modelu YCbCr.

4 POPIS OBJEKTŮ

Pro pochopení a teoretický rozbor problematiky popisu objektu byla použita literatura [HKPR08], [HŠ92], [Kot93] a [SS90]. Popis objektů je jedním ze základních kroků při zpracování a rozpoznávání obrazu. Dle [HŠ92] je posloupnost základních kroků:

- *snímání, digitalizace a uložení obrazu v počítači,*
- *předzpracování,*
- *segmentace obrazu na objekty,*
- *popis objektů,*
- *rozpoznávání objektů.*

Z výše uvedených kroků si podrobněji rozebereme popis objektů. Předchozími kroky se dále nebudeme zabývat, protože námi popisované objekty jsou uměle vytvořeny, tudíž není nutno provádět uvedené kroky: snímání, digitalizace, předzpracování, segmentace.

Při popisu objektů lze využít dvou odlišných popisů: strukturální popis a popis pomocí příznakového vektoru. Strukturální popis umožňuje klasifikaci do tříd, ale také umožňuje nahlédnout do struktury objektů. A to díky souboru primitiv (základních popisných elementů), jejich vlastnostmi a relacemi mezi nimi. Zaměříme se pouze na popis pomocí příznakového vektoru.

4.1 Příznakový vektor

Příznakový vektor se skládá z vybraných příznaků, které byly získány z daných objektů. Volba vhodné množiny příznaků je velmi komplikovaná. Musíme volit příznaky, které jsou nejvhodnější pro danou úlohu rozpoznávání.

U některých objektů lze selektovat velké množství příznaků. Potom je vhodné provést redukci dimenze příznakového vektoru pomocí:

- U selekce jde o výběr nejlepší podmnožiny z původních příznaků bez transformace. Vybrané příznaky mají původní význam. Dochází k redukci počtu příznaků.
- Extrakce používá transformaci, kdy se příznaky vypočítávají z původních příznaků. Nové příznaky nemají původní význam. Také dochází k redukci počtu transformovaných příznaků, ale je stále nutné zjišťovat všechny původní příznaky.

Dle [HKPR08] jsou na příznaky kladeny často protichůdné požadavky, mezi základní požadavky patří:

- spolehlivost – příznaky mají podobné hodnoty pro objekty ze stejné třídy,
- diskriminabilita – příznaky mají odlišné hodnoty pro různé objekty,
- invariantnost – příznaky jsou stále při změně určitého parametru (rotace, jas, měřítko atd.),
- nezávislost – příznaky musí být vzájemně nekorelované.

Příznaky se rozdělují do mnoha skupin podle různých hledisek. Dále budeme uvádět pouze příznaky používané pro segmentované objekty. Dle [HKPR08] jsou základní tři typy rozdělení: doména popisované vlastnosti objektu, oblast výpočtu, oblast popisu.

Podle domény popisované vlastnosti objektu rozeznáváme příznaky fotometrické (popisují optické vlastnosti objektu) a radiometrické (popisují geometrické vlastnosti objektu). Podle oblasti výpočtu rozeznáváme příznaky založené na regionech (potřeba znalosti jasových hodnot pixelů objektu) a hranicích (potřeba znalosti hranice objektu). Podle oblasti popisu rozeznáváme globální příznaky obrazu, globální a lokální příznaky objektu.

Pro námi rozpoznávané objekty, 26 písmen od A do Z, zvolíme momentový popis, protože využijeme výhod momentového popisu: invariantnost vůči rotaci, změně měřítka a posunutí. Vlastnosti tohoto popisu by se zejména hodily, pokud by naše objekty nebyly uměle vytvořené, ale objekty bychom získávali pomocí snímání, digitalizace, předzpracování a segmentace.

4.2 Momentový popis

Momentový popis oblastí interpretuje normalizovanou jasovou funkci obrazu, jako hustotu pravděpodobnosti dvojrozměrné náhodné veličiny. Dle [HŠ92] lze vlastnosti této veličiny vyjádřit prostřednictvím statických charakteristik (momentů), které je možno použít k popisu binárních i šedotónových oblastí.

Obecných momentů stupně (řádu) $p+q$ je možné sestavit teoreticky nekonečně mnoho pro popisovaný objekt, čím více momentů pro popis použijeme, tím docílíme věrnějšího popisu objektu. Obecný moment je vyjádřen vztahem (4.1), kde x a y jsou souřadnice bodu oblasti. Uváděné vzorce platí pro digitální obraz. Moment nultého řádu je pouze jeden a pro binární obraz udává velikost.

$$m_{pq} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} x^p y^q f(x, y) \quad (4.1)$$

V binárním obrazu jsou pouze dvě barvy reprezentované binárně: 0 pro černou barvu (jas černé je 0) a 1 pro bílou barvu (jas bílé je 255). Klasický obraz má bílé

pozadí a černou barvou jsou zastoupeny rozpoznávané objekty. Ze vzorce (4.1) je zřejmé, že počítáme momenty bílého pozadí ($f(x,y) = 1$). To je však nevhodné, a proto se používají invertované obrazy. Tím docílíme i zmenšení výpočetní náročnosti, budeme počítat momenty pro menší plochu.

Souřadnice těžiště $[x_T, y_T]$ obrazu (4.2) získáme z obecných momentů nultého a prvního řádu. Pro docílení invariantnosti vůči posunutí je nutné použít centrální momenty (4.3). Invariantnosti vůči změně měřítka je docílena u normovaných centrálních momentů dle (4.4), kde $TRUNC(x)$ značí celou část x .

$$x_T = \frac{m_{10}}{m_{00}} \quad y_T = \frac{m_{01}}{m_{00}} \quad (4.2)$$

$$u_{pq} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} (x - x_T)^p (y - y_T)^q f(x, y) \quad (4.3)$$

$$\vartheta_{pq} = \frac{u_{pq}}{u_{00}^\gamma} \quad \gamma = TRUNC\left(\frac{p+q}{2}\right) + 1 \quad (4.4)$$

Dle [Hu62] vypočteme sedm momentových invariantů pomocí normovaných centrálních momentů, jak je uvedeno ve vzorcích (4.5) až (4.11). Ty jsou nezávislé na rotaci, změně měřítka, posunutí neboli translaci (ta je dána přesnou segmentací obrazu). Uvedené momenty jsou však stále závislé na lineární šedotónové transformaci oblastí, avšak nám tato skutečnost nevádí, protože se budeme dále zabývat pouze binárními obrazy.

$$\varphi_1 = \vartheta_{20} + \vartheta_{02} \quad (4.5)$$

$$\varphi_2 = (\vartheta_{20} - \vartheta_{02})^2 + 4\vartheta_{11}^2 \quad (4.6)$$

$$\varphi_3 = (\vartheta_{30} - 3\vartheta_{12})^2 + (3\vartheta_{21} - \vartheta_{03})^2 \quad (4.7)$$

$$\varphi_4 = (\vartheta_{30} + \vartheta_{12})^2 + (\vartheta_{21} + \vartheta_{03})^2 \quad (4.8)$$

$$\varphi_5 = (\vartheta_{30} - 3\vartheta_{12})(\vartheta_{30} + \vartheta_{12})[(\vartheta_{30} + \vartheta_{12})^2 - 3(\vartheta_{21} + \vartheta_{03})^2] + \\ + (3\vartheta_{21} - \vartheta_{03})(\vartheta_{21} + \vartheta_{03})[3(\vartheta_{30} + \vartheta_{12})^2 - (\vartheta_{21} + \vartheta_{03})^2] \quad (4.9)$$

$$\varphi_6 = (\vartheta_{20} - \vartheta_{02})[(\vartheta_{30} + \vartheta_{12})^2 - (\vartheta_{21} + \vartheta_{03})^2] + \\ + 4\vartheta_{11}(\vartheta_{30} + \vartheta_{12})(\vartheta_{21} + \vartheta_{03}) \quad (4.10)$$

$$\varphi_7 = (3\vartheta_{21} - \vartheta_{03})(\vartheta_{30} + \vartheta_{12})[(\vartheta_{30} + \vartheta_{12})^2 - 3(\vartheta_{21} + \vartheta_{03})^2] - \\ - (\vartheta_{30} - 3\vartheta_{12})(\vartheta_{21} + \vartheta_{03})[3(\vartheta_{30} + \vartheta_{12})^2 - (\vartheta_{21} + \vartheta_{03})^2] \quad (4.11)$$

4.2.1 Normalizace momentů

Při učení neuronové sítě pomocí momentů $\varphi_1 - \varphi_7$ je výhodné provést normalizaci jejich hodnot do určitého intervalu, např. $\langle 0;1 \rangle$. Hodnoty momentů zjištěných pro různé objekty se pohybují v úzkých intervalech hodnot. Pokud není provedena normalizace, tak neuronová síť má problémy s rozpoznáváním jednotlivých objektů, jejichž hodnoty momentů leží velmi blízko sebe.

Například při normalizaci momentů φ_1 do intervalu $\langle 0;1 \rangle$, musíme najít minimum a maximum ze všech momentů φ_1 a pomocí vzorce (4.12) přepočítat všechny ostatní momenty φ_1 . Obdobně se postupuje pro další momenty.

$$\varphi_{1NORMALIZOVANÉ} = \frac{\varphi_1 - \varphi_{1MIN}}{\varphi_{1MAX} - \varphi_{1MIN}} \quad (4.12)$$

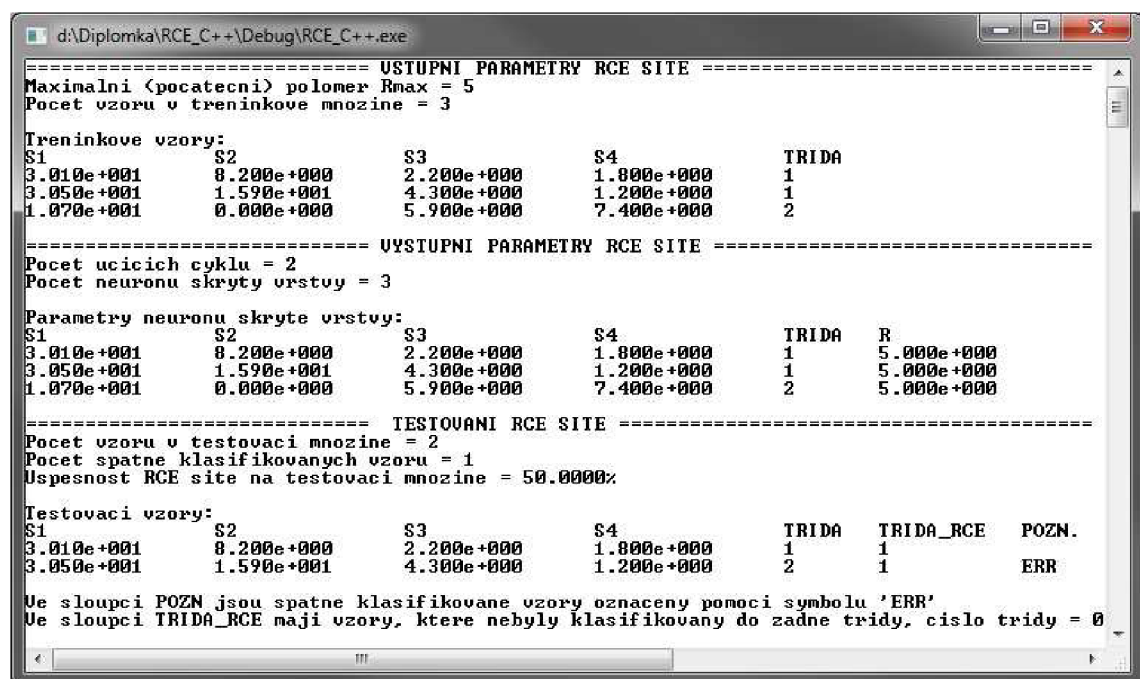
5 PROGRAM uTeachRCE PRO UČENÍ RCE SÍTĚ

Program uTeachRCE pro učení RCE sítě bezprostředně vychází z učícího algoritmu popsaného v kapitole 3.4. Program bude využíván pro zkoumání rozpoznávání písmen pomocí RCE sítě, viz kapitola 6.

Program je vytvořen v programovacím prostředí Microsoft Visual Studio 2008, kde je využíván programovací jazyk C++. V příloze 1 je uveden zdrojový kód programu.

5.1 Popis programu

Program umožňuje naučit RCE síť vstupní data pro libovolné n z n -rozměrného prostoru, které jsou rozděleny do libovolného počtu klasifikovaných tříd. Vstupní data jsou programu předávána pomocí XML souboru, jeho popis je v kapitole 5.2. Program uživateli předává informace pomocí výstupního textového okna, viz Obr. 9.



```
==== USTUPNI PARAMETRY RCE SITE =====
Maximalni (pocatecni) polomer Rmax = 5
Pocet vzoru v treninkove množine = 3

Treninkove vzory:
$1          $2          $3          $4          TRIDA
3.010e+001  8.200e+000  2.200e+000  1.800e+000  1
3.050e+001  1.590e+001  4.300e+000  1.200e+000  1
1.070e+001  0.000e+000  5.900e+000  7.400e+000  2

==== VYSTUPNI PARAMETRY RCE SITE =====
Pocet ucicich cyklu = 2
Pocet neuronu skryte vrstvy = 3

Parametry neuronu skryte vrstvy:
$1          $2          $3          $4          TRIDA  R
3.010e+001  8.200e+000  2.200e+000  1.800e+000  1      5.000e+000
3.050e+001  1.590e+001  4.300e+000  1.200e+000  1      5.000e+000
1.070e+001  0.000e+000  5.900e+000  7.400e+000  2      5.000e+000

==== TESTOVANI RCE SITE =====
Pocet vzoru v testovaci množine = 2
Pocet spatne klasifikovanych vzoru = 1
Uspesnost RCE site na testovaci množine = 50.0000%

Testovaci vzory:
$1          $2          $3          $4          TRIDA  TRIDA_RCE  POZN.
3.010e+001  8.200e+000  2.200e+000  1.800e+000  1      1
3.050e+001  1.590e+001  4.300e+000  1.200e+000  2      1      ERR

Ue sloupci POZN jsou spatne klasifikovane vzory oznaceny pomoci symbolu 'ERR'
Ue sloupci TRIDA_RCE maji vzory, ktere nebyly klasifikovany do zadne tridy, cislo tridy = 0
```

Obr. 9: Ukázka výstupního textového okna programu

Textové okno je děleno na následující bloky:

- Vstupní parametry sítě vypisují maximální poloměr a tréninkové vzory získané ze vstupního XML souboru.

- Výstupní parametry sítě shrnují informace o naučené RCE síti. Informují o jednotlivých neuronech skryté vrstvy: souřadnicích středu, poloměru a třídě, jenž zastupují.
- Testování sítě ukazuje odezvy sítě na testovací vzory. Dále také ukazuje statistické informace týkající se testovacích vzorů. Tento blok se zobrazuje pouze tehdy, když síti předložíme testovací vzory.

Při výpisu čísel souřadnic vzorů a neuronů byl zvolen vědecký zápis čísel (číslo je rozděleno na mantisu a exponent), protože tento zápis je vhodný k zápisu malých čísel. Malá čísla budeme používat v kapitole šest k rozpoznávání písmen pomocí RCE sítě, protože bude použita normalizace příznaků do intervalu $(0;1)$ z důvodů uvedených v kapitole 4.2.1.

5.2 Vstupní data programu

Vstupní data pro učení RCE sítě jsou programu předávána pomocí XML souboru (Data.xml). Struktura XML souboru je znázorněna na Obr. 10. Zadávaná vstupní data musí splňovat předepsané podmínky, aby program fungoval správně.

Pro parametr m udávající počet tříd platí podmínka, že se musí jednat o celé číslo (minimální hodnota je jedna). Parametr třída u jednotlivých bodů je v průběhu zpracování XML souboru kontrolován, jeho hodnota musí být celé číslo z intervalu 1 až m , kde m udává počet tříd.

Poloměr (R_{MAX}) musí být reálné číslo větší jak nula. Souřadnice (S_i , $i = 1, \dots, n$) jednotlivých bodů jsou libovolná reálná čísla. Při zpracování XML souboru je prováděna kontrola, že všechny zadané body mají stejný počet souřadnic.

Body testovací množiny nemusí být uvedeny, ale aby načtení XML souboru proběhlo v pořádku, tak v XML souboru musí zůstat element TEST_MNOZINA, ve kterém nebudou uvedeny žádné testovací body.

Způsob zadávání vstupních dat formou XML souboru je výhodný zejména, pokud máme větší množství dat získané např. měřením, tak soubor XML můžeme velmi jednoduše vytvořit pomocí specializovaných programů a nemusíme vstupní data pracně přepisovat do uživatelského rozhraní programu.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RCE>
  <POLOMER R="5"/>
  <POCET_TRID P="2"/>
  <TREN_MNOZINA>
    <BOD S1="32.1" S2="8.2" S3="2.2" S4="1.8" TRIDA="1"/>
    <BOD S1="30.5" S2="15.9" S3="4.3" S4="1.2" TRIDA="1"/>
    <BOD S1="10.7" S2="0" S3="5.9" S4="7.4" TRIDA="2"/>
  </TREN_MNOZINA>
  <TEST_MNOZINA>
    <BOD S1="32.1" S2="8.2" S3="2.2" S4="1.8" TRIDA="1"/>
    <BOD S1="30.5" S2="15.9" S3="4.3" S4="1.2" TRIDA="2"/>
  </TEST_MNOZINA>
</RCE>

```

Obr. 10: Struktura vstupních dat v XML souboru

5.3 Základní struktura programu

Z popisu této struktury vyplývá celková filozofie programu. Při vytváření této struktury byl kladen důraz na její komplexnost. Struktura slouží k načítání bodů tréninkové množiny, učení RCE sítě a později i k vizualizaci učení sítě, viz kapitola 7.

Struktura `LearnStructure` obsahuje proměnné:

- vector <double> `Center`
- unsigned int `Class`
- bool `ShowSphere`
- double `R`

Nejprve je struktura naplněna body tréninkové množiny, proměnná `Center` uchovává souřadnice bodu a `Class` číslo třídy bodu. Proměnná `LearnSet` je jednorozměrné pole (velikost je určena počtem bodů tréninkové množiny), jejíž prvky jsou struktury `LearnStructure`.

Proměnná `LearnSet` je poté použita pro učení RCE sítě. V momentě, kdy je potřeba vytvořit nový skrytý neuron, tak dojde pouze k nastavení proměnné `ShowSphere` na hodnotu `True` u daného tréninkové bodu. Hodnotu poloměru nového neuronu nastavíme v proměnné `R`. Střed neuronu je stejný jako střed bodů, kolem kterého byl vytvořen (proměnná `Center`).

Pro testovací množinu se používá struktura `TestStructure` s mnohem jednodušší koncepcí.

5.4 Funkce programu

V následujících odstavcích jsou popsány základní funkce programu sloužící k učení/testování RCE sítě a načítání vstupního souboru.

ReadXML: Funkce zajišťuje načítání vstupního XML souboru a zápis bodů tréninkové a testovací množiny do proměnných, jenž jsou později použity pro učení a testování RCE sítě. Struktura vstupního XML souboru je popsána v kapitole 5.2. Pro práci s XML souborem je využívána open-source knihovna TinyXML¹. V rámci funkce dochází ke kontrole datových typů případně i rozsahů u všech načítaných proměnných. Při načítání bodů je kontrolováno, že mají všechny body stejný počet souřadnic. U bodů tréninkové množiny se také kontroluje, zda dva stejné body nemají rozdílnou třídu (RCE síť by se nebyla schopna naučit).

LearnRCE: Funkce zajišťuje učení RCE sítě podle algoritmu z kapitoly 3.4.

AddHideNeuron: Tato funkce je volána z funkce LearnRCE v případě, kdy je potřeba vytvořit nový skrytý neuron. Funkce zajišťuje i další kroky, které souvisí s vytvořením nového neuronu: nastavení váhových spojení, kontrola překrytí koulí u nového neuronu a stávajících neuronů.

TestRCE: Funkce určená k testování již naučené RCE sítě. Pro jednotlivé body testovací množiny zjišťuje odezvu sítě.

EuclideanDist: Funkce vypočítává euklidovskou vzdálenost dvou středů zapsaných pomocí vektorů.

Z výše popsaných funkcí nebyla vytvořena knihovna, i když se funkce znovu používají v dalším programu RCEin3D, kapitola 7. Tento způsob řešení nám umožňuje mále úpravy funkcí, které ve výsledku zrychlí a zjednoduší chod programu RCEin3D. Tyto úpravy jsou popsány v kapitole 7.6.

¹ Knihovna TinyXML je dostupná z: <<http://www.grinninglizard.com/tinyxml/>>.

6 ROZPOZNÁVÁNÍ PÍSMEN POMOCÍ RCE SÍTĚ

Tato kapitole se zabývá zkoumáním kvality rozpoznání písmen pomocí RCE sítě. Dosažené výsledky budou srovnávány s výsledky pro vícevrstvou neuronovou síť s učícím algoritmem backpropagation, v práci používáme zjednodušený název backpropagation. V průběhu celé kapitoly srovnáváme obě sítě z různých hledisek, kapitola 6.4 obsahuje závěrečné srovnání sítí. Síť backpropagation byla testována v rámci bakalářské práce [Mac11]. Zpracované výsledky všech testování se nacházejí v příloze 1.

Aby srovnání dvou neuronových sítí mělo vypovídající hodnotu, tak budou při testování RCE sítě použity stejné podmínky, jako při testování sítě backpropagation v bakalářské práci: pro popis písmen využijeme momentový popis (kapitola 4.2), použijeme stejné množiny písmen (kapitola 6.1).

6.1 Použité vzory písmen

Pro učící vzory bylo použito 26 písmen od A do Z s použitým typem písma Arial Black o velikosti 48. Všechna písmena uložená v bitmapě jsou středově orientována na ploše 70x70 pixel. Z důvodu uvedeného v kapitole 4.2 jsou vzory invertovány (pozadí černé a písmena bílé).

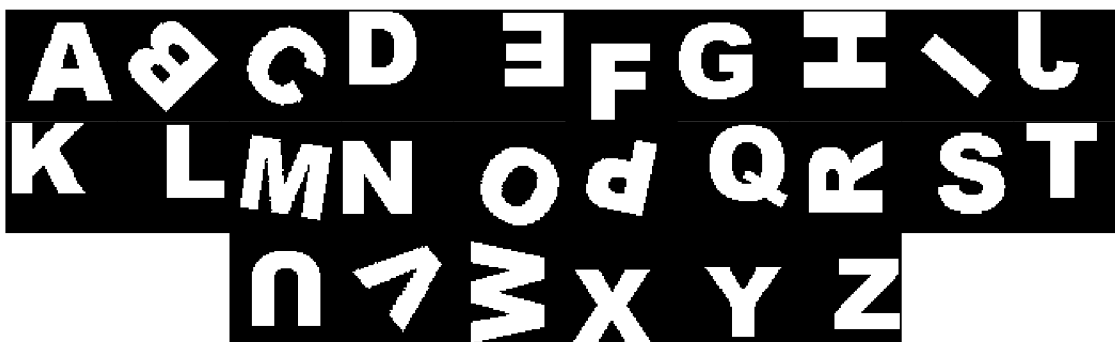
Množina A je tedy tvořena 26 vzory, jeden vzor pro každé středově orientované písmeno stejné velikosti. Množina B se skládá ze 130 vzorů (5 vzorů pro každé písmeno), množina C se skládá z 26 vzorů (1 vzor pro každé písmeno). Písmena v množinách B a C byla změněna. Změny byly voleny tak, aby byly otestovány výhody momentového popisu. A proto množiny B a C tvoří písmena, jenž jsou otočena o libovolný úhel, posunuta v libovolném směru, zvětšena nebo zmenšena. Na Obr. 11, 12 a 13 jsou uvedena písmena tvořící všechny tři vytvořené množiny (množiny jsou taktéž uvedeny v příloze 1).



Obr. 11: Vzory písmen tvořící množinu A



Obr. 12: Vzory písmen tvořící množinu B



Obr. 13: Vzory písmen tvořící množinu C

Množiny A a B byly převzaty z bakalářské práce [Mac11], množina C byla vytvořena pro RCE síť, důvod jejího vytvoření bude popsán v kapitole 6.3.

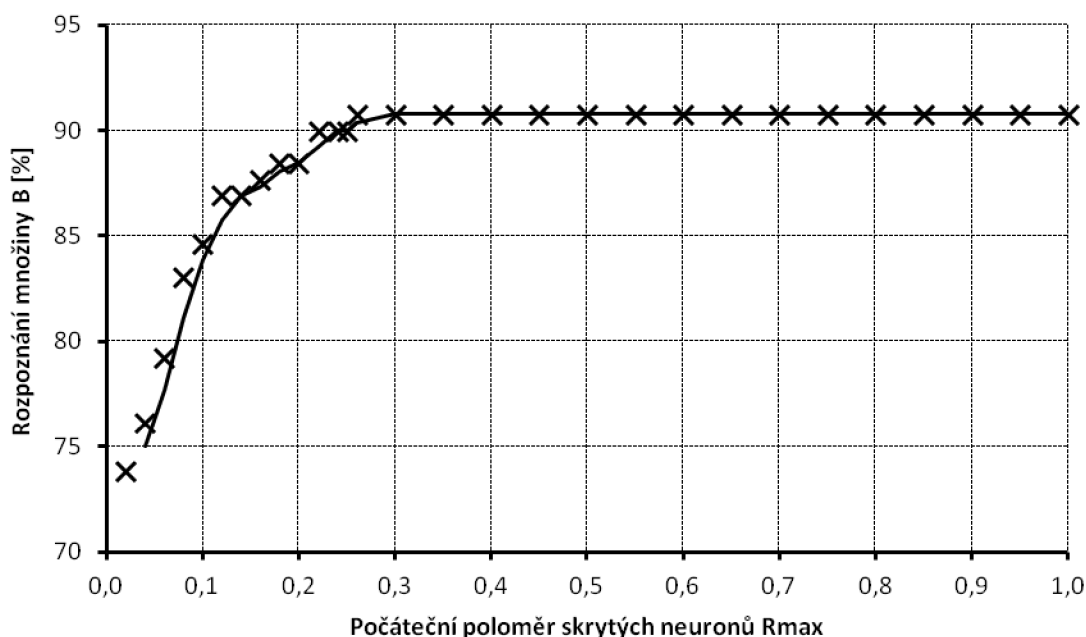
6.2 Rozpoznávání písmen množiny B

Při prvním testování kvality rozpoznání písmen pomocí RCE sítě byly zavedeny stejné podmínky, jako při učení sítě backpropagation. RCE síť byla naučena na množinu A (26 vzorů) a kvalita rozpoznání písmen byla ověřována pomocí množiny B (130 vzorů).

Během tohoto testování byl proveden rozbor vlivu počáteční hodnoty poloměru skrytých neuronů R_{MAX} na kvalitu rozpoznání písmen. U RCE sítě se jedná o jediný parametr, který může být testován. Poloměr R_{MAX} byl nastavován z intervalu od 0,02 do 1. Zvyšování poloměru R_{MAX} nad hodnotu jedna je bezvýznamné, protože příznaky pro popis písmen byly normalizovány do intervalu $\langle 0;1 \rangle$.

Závislost rozpoznání množiny B na počáteční hodnotě poloměru skrytých neuronů R_{MAX} je na Obr. 14 (body jsou proloženy klouzavým průměrem). Ze zobrazené závislosti je zřejmé že, pro $R_{MAX} \geq 0,26$ je rozpoznání množiny B konstantní: 90,77%.

Během testování bylo zjištěno, že hodnota poloměru R_{MAX} nemá vliv na rychlost učení RCE sítě: všechny sítě se naučily během dvou učících cyklů. Dále poloměr nemá vliv na počet vytvořených skrytých neuronů: všechny sítě potřebovaly 26 skrytých neuronů (jeden skrytý neuron pro jednu klasifikovanou třídu, respektive jedno písmeno), z této informace je jasné že rozpoznání tréninkové množiny (množiny A) je stoprocentní.



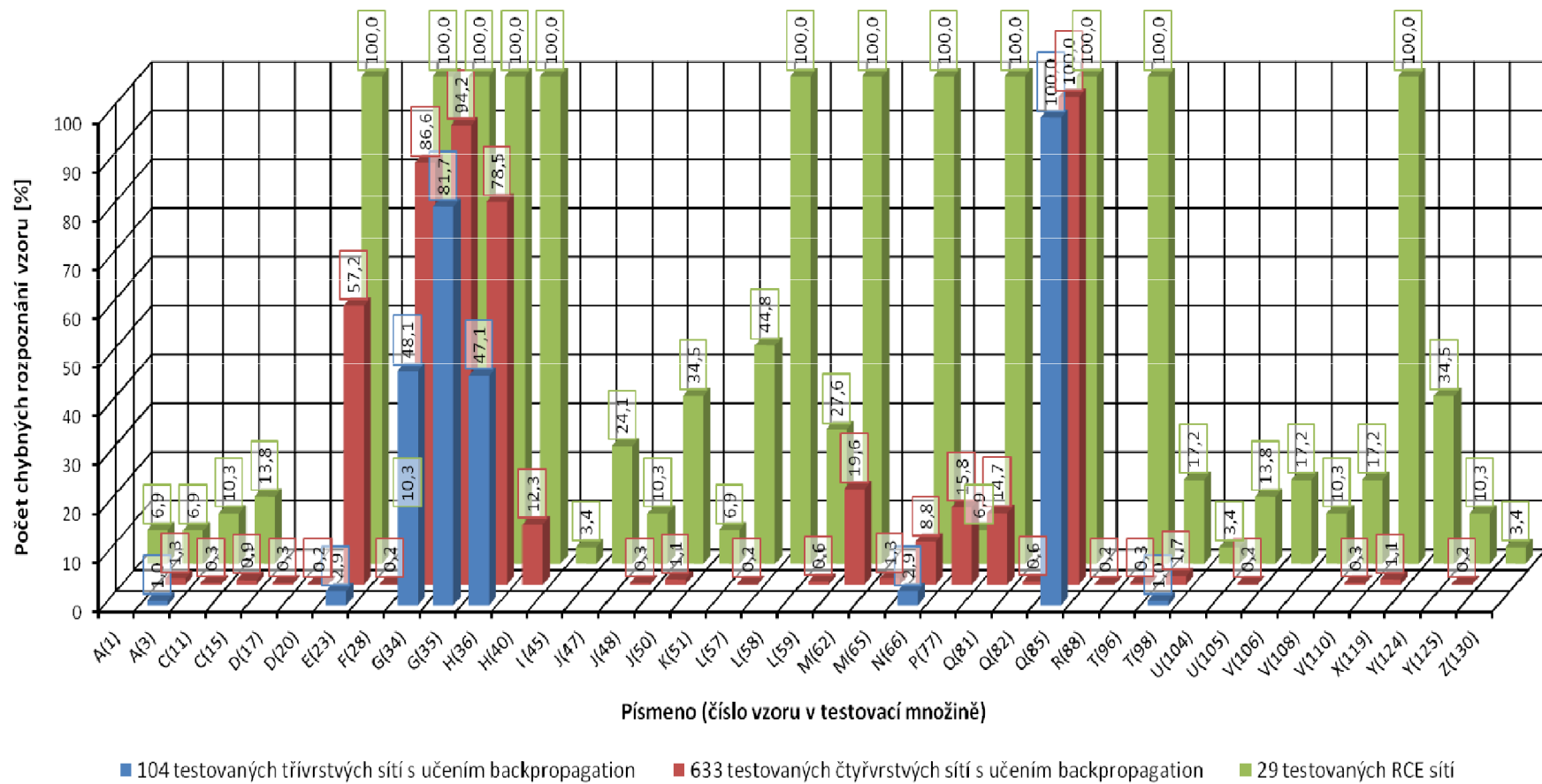
Obr. 14: Závislost rozpoznání množiny B na počátečním poloměru skrytých neuronů R_{MAX}

Pro každou testovanou RCE síť je ukládán řetězec chybně detekovaných testovacích vzorů ve formátu: Rozpoznávané písmeno (číslo vzoru v množině B) => Detekované písmeno; další rozpoznávané písmeno. Například řetězec 'G(35)=>Q;' nám říká, že testovací vzor 35, na němž je písmeno G (upravené pomocí otočení nebo posunutí nebo změny měřítka, viz kapitola 6.1) bylo chybně rozpoznáno jako písmeno Q.

Dále byla vytvořena statistika popisující, který vzor písmena a kolikrát byl chybně detekován. Tato statistika byla poté přepočtena na procenta dle vztahu (6.1): počet chybných detekcí daného vzoru Err_x byl podělen maximálním počtem chybných detekcí All_x , které mohly u daného vzoru nastat. Maximální počet chybných detekcí je dán počtem testovaných sítí.

$$STAT_x = \frac{Err_x}{All_x} \cdot 100 \quad (6.1)$$

Obdobné statistiky byly vytvořeny i pro třívrstvou a čtyřvrstvou síť s algoritmem učení backpropagation v bakalářské práci [Mac11]. Na Obr. 15 je uveden sloupcový graf znázorňující počet chybně detekovaných vzorů množiny B v procentech. V grafu se vyskytují pouze ty vzory, které byly alespoň jednou chybně rozpoznány jednou ze sítí.



Obr. 15: Graf vyjadřující počet chybných rozpoznání vzorů množiny B u jednotlivých sítí

Z grafu na Obr. 15 je patrné:

- třívrstvá síť backpropagation má problémy s rozpoznáním 8 vzorů,
- čtyřvrstvá síť backpropagation má problémy s rozpoznáním 29 vzorů,
- RCE síť má problémy s rozpoznáním 34 vzorů,
- RCE síť má nejvíce 100% nerozpoznatelných vzorů (12), hodnoty rozpoznání těchto vzorů u sítí backpropagation jsou většinou nadprůměrné.

Při testování byl objeven fakt, že všech 445 špatně rozpoznávaných vzorů RCE síť nezařadila do žádné z klasifikovaných tříd. Tento jev je nespornou výhodou RCE sítě, protože jsme ihned při testování kvality naučení sítě schopni posoudit, zda je síť potřeba doučit na další vzory.

U sítě backpropagation tuto možnost nemáme, protože její způsob učení/vybavování vždy zajistí, že i naprosto neznámý vzor je zařazen do některé z klasifikovaných tříd. Jedinou možností by bylo zkoumání hodnoty výstupního signálu u vítězného výstupního neuronu, ale toto zjištění by nebylo natolik průkazné, jako u RCE sítě.

Jelikož bylo zjištěno nedostatečné naučení RCE sítě vzhledem k množině B, tak budeme pokračovat v dalším testování sítě s využitím množiny C, viz kapitola 6.3.

6.3 Rozpoznávání písmen množiny C

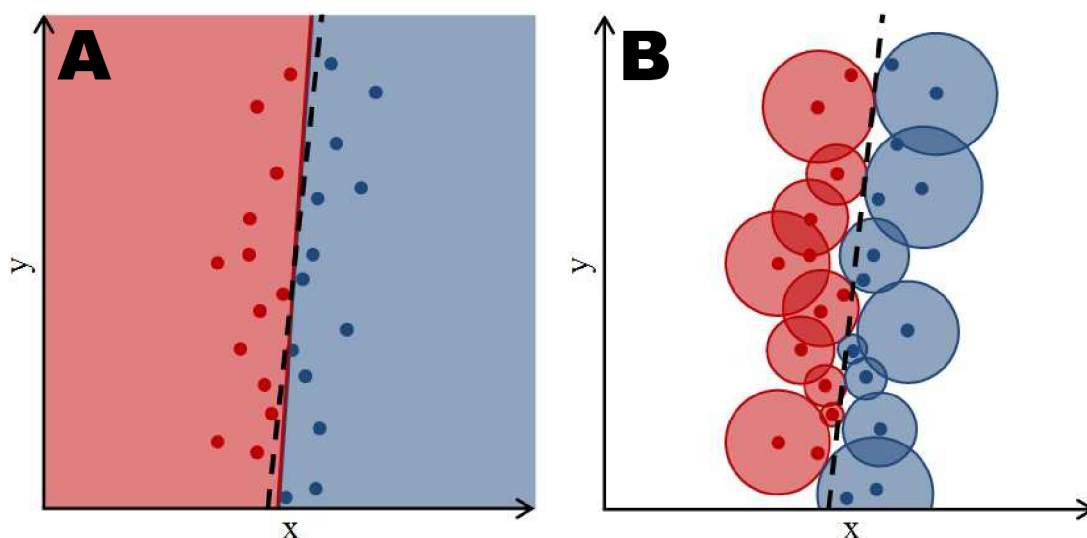
V kapitole 6.2 jsme zjistili, že RCE síť je při naučení 26 tréninkových vzorů (množina A) nedostatečně naučena vzhledem k množině B (na rozdíl od sítě backpropagation [Mac11]). Pomocí Obr. 16 vysvětlíme, proč RCE síť nedosáhla obdobných výsledků jako backpropagation. Situace znázorněná na Obr. 16 jsou pouze orientační, nejedná se o skutečné rozdělení vstupního prostoru dle učících algoritmů jednotlivých sítí.

Na Obr. 16A je znázorněn příklad rozdělení vstupního prostoru pomocí vícevrstvé neuronové sítě s učením backpropagation. Díky učicímu algoritmu backpropagation a vhodně vybraným bodům tréninkové množiny (tmavě modré a červené body) dojde k téměř přesnému rozdělení roviny XY pomocí přímky (čárkovaná čára naznačuje skutečné rozdělení roviny XY do dvou tříd). Výhodou backpropagation je, že pomocí malé (avšak reprezentativní) tréninkové množiny dostaneme dělicí přímku, která je platná v celé rovině XY bez ohledu na velikost jednotlivých souřadnic. Je zaručeno, že každý bod z roviny XY je klasifikován do modré nebo do červené oblasti.

Příklad rozdělení vstupního prostoru RCE sítí je na Obr. 16B. Z ukázky rozdělení vstupního prostoru pomocí sítě backpropagation a RCE sítě je zřejmý problém, který nastal i při učení RCE sítě v kapitole 6.2. Učící algoritmus RCE sítě vytvořil pro dané tréninkové body skryté neurony (světle modré a červené kruhy), ale zbytek roviny XY zůstal neurčený. Tudíž několik tréninkových bodů, které stačily u sítě

backpropagation k správnému rozdělení roviny XY, zajistí u RCE sítě správnou klasifikaci pouze v relativně úzkém pásu kolem skutečné dělicí přímky (čárkovaná čára). Tento problém nevyřeší ani velká hodnota počátečního poloměru skrytých neuronů R_{MAX} , protože dojde k jejich zmenšení vlivem blízkosti obou klasifikovaných oblastí.

Při vyskytnutí tohoto problému u RCE sítě je nutné posoudit, zda problém bude vyřešen dostatečným rozšířením tréninkové množiny, čímž dojde k vytvoření únosného počtu skrytých neuronů. Jinak by bylo vhodné přemýšlet o použití sítě backpropagation.



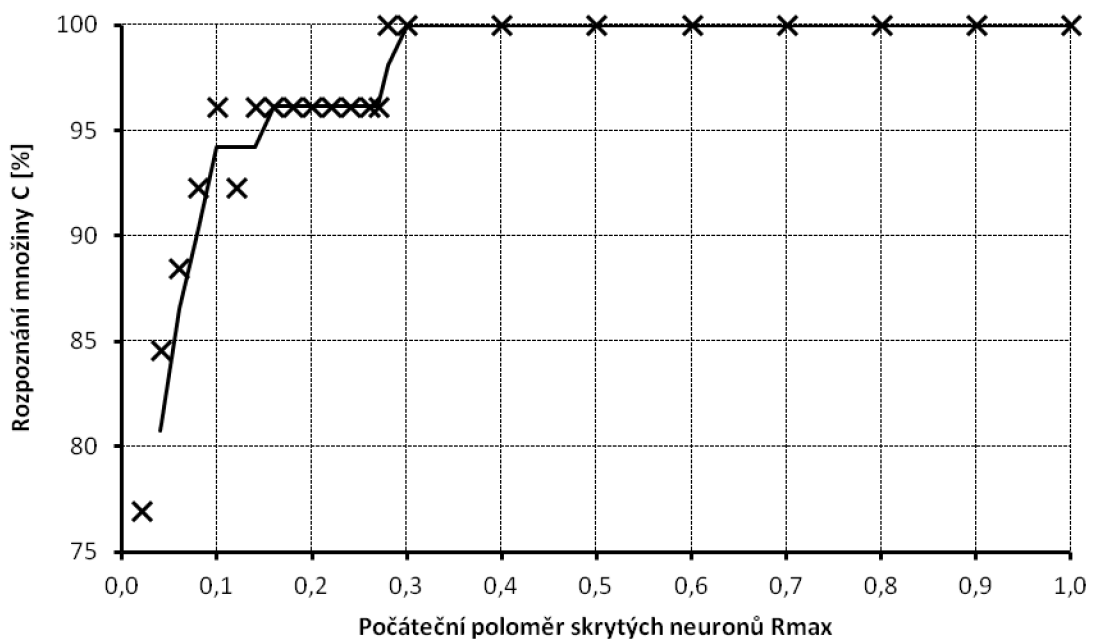
Obr. 16: Rozdělení vstupního prostoru do dvou tříd, (A) Vícevrstvá neuronová síť s algoritmem učení backpropagation, (B) RCE neuronová síť

Vzhledem k výše popsaným důvodům bylo provedeno druhé testování RCE sítě: síť byla naučena na novou tréninkovou množinu, která vznikla spojením množiny A a B (dohromady 156 vzorů), a kvalita rozpoznání byla ověřena na množině C (26 vzorů).

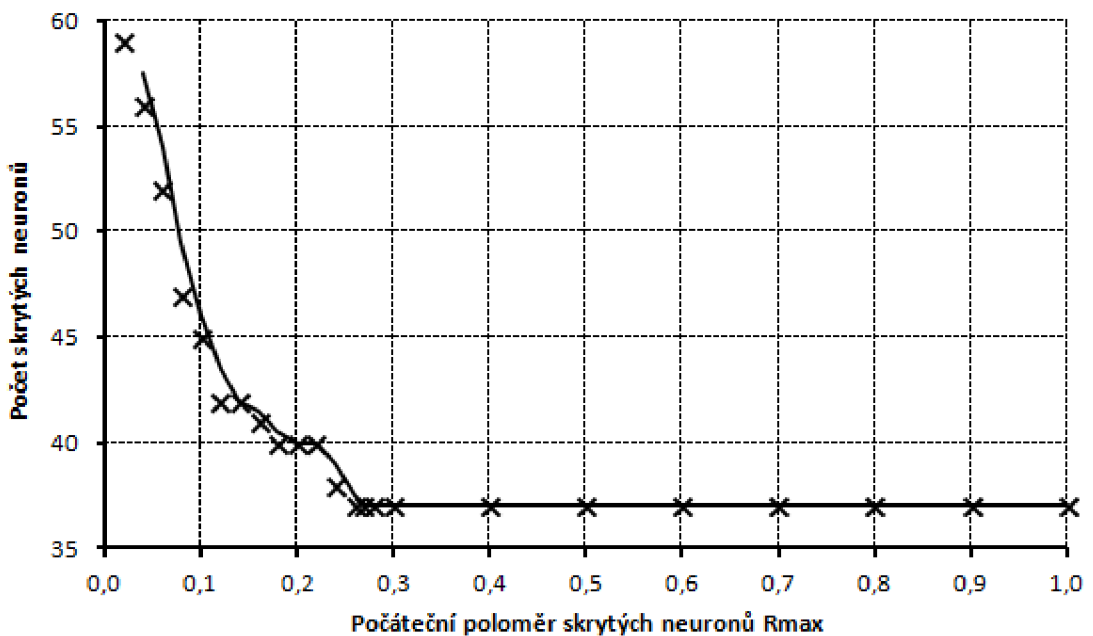
Stejně jako při prvním testování v kapitole 6.2 byl i při tomto testování zkoumán vliv poloměru R_{MAX} na kvalitu rozpoznání písmen množiny C. Poloměr R_{MAX} byl nastavován z intervalu od hodnoty 0,02 do 1. Na Obr. 17 je zobrazena závislost rozpoznání množiny C na počátečním poloměru skrytých neuronů R_{MAX} (body jsou proloženy klouzavým průměrem). Rozšířením tréninkové množiny bylo docíleno sto procentní úspěšnosti rozpoznání písmen množiny C pro hodnoty $R_{MAX} \geq 0,28$.

Při tomto testování se opět všechny RCE sítě naučily množinu A a B za dva učící cykly a všechny špatně rozpoznané vzory RCE síť nezařadila do žádné z klasifikovaných tříd (tyto zjištění jsou stejné jako v kapitole 6.2).

Při učení RCE sítě na množinu A a B (156 vzorů) už nebyl počet skrytých neuronů konstantní v průběhu celého testování, ale jejich počet se měnil v závislosti na hodnotě poloměru R_{MAX} viz Obr. 18.



Obr. 17: Závislost rozpoznání množiny C na počátečním poloměru skrytých neuronů R_{MAX}



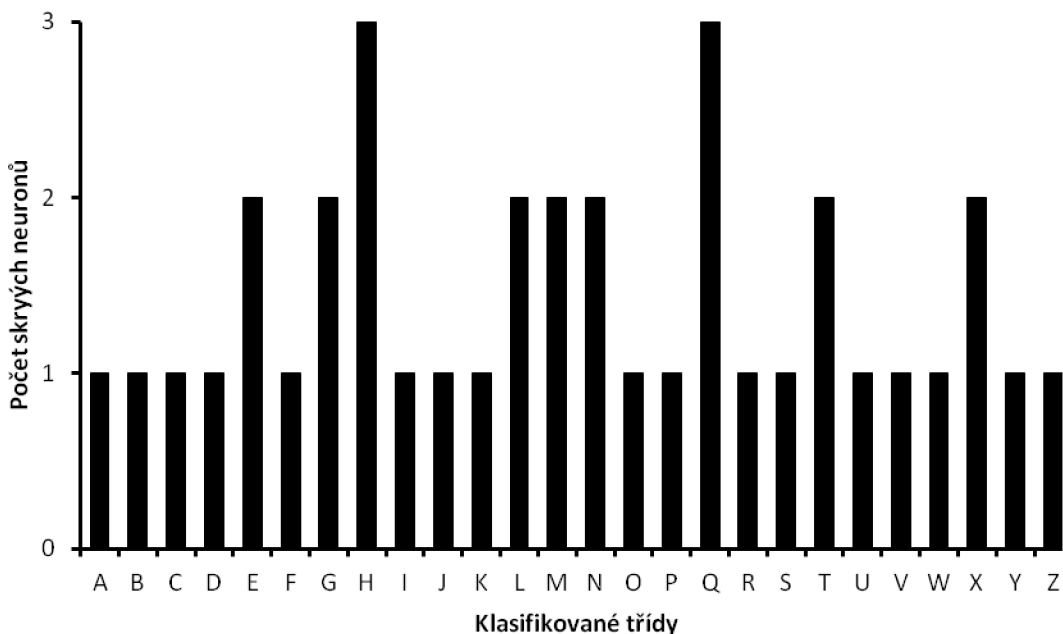
Obr. 18: Závislost počtu skrytých neuronů na počátečním poloměru skrytých neuronů R_{MAX}

Závislost počtu skrytých neuronů na hodnotě počátečního poloměru R_{MAX} zobrazená na Obr. 18 má předpokládaný průběh: se vzrůstající hodnotou počátečního poloměru R_{MAX} se počet skrytých neuronů zmenšuje, pro hodnoty $R_{MAX} \geq 0,26$ je počet skrytých neuronů na konstantní hodnotě 37. Při zobrazení závislosti byly body proloženy klouzavým průměrem.

Srovnáním charakteristik na Obr. 17 a 18 je zřejmé, že pro hodnoty 0,26 a 0,27 poloměru R_{MAX} je počet skrytých neuronů 37 (tj. optimální hodnota, viz Obr. 18), ale úspěšnost rozpoznání množiny C je 96,15%. K dosažení stoprocentního rozpoznání množiny C dojde až při zvýšení R_{MAX} na hodnotu 0,28.

Při prvním testování v kapitole 6.2 byl počet skrytých neuronů na konstantní hodnotě 26, kdy každá klasifikovaná třída byla zastoupena jedním neuronem. Při tomto testování se počet skrytých neuronů ustálil na hodnotě 37, jejich rozdělení pro jednotlivé klasifikované třídy je zobrazeno na Obr. 19 (rozdělení je platné pro poloměry R_{MAX} od 0,28 do 1).

Srovnáním grafu z Obr. 15, jenž ukazuje chybně rozpoznané vzory množiny B, a grafu na Obr. 19, který ukazuje počet skrytých neuronů při doučení RCE sítě na množinu B, vidíme: všechny stoprocentně nerozpoznané vzory (to jsou vzory, které nebyly ani jednou správně detekovány testovanými RCE sítěmi) patří do klasifikovaných tříd (E, G, H, L, M, N, Q, T, X), které na rozdíl od ostatních potřebují k naučení 2 až 3 skryté neurony.



Obr. 19: Graf vyjadřující počet skrytých neuronů pro jednotlivé klasifikované třídy, pro R_{MAX} od 0,28 do 1

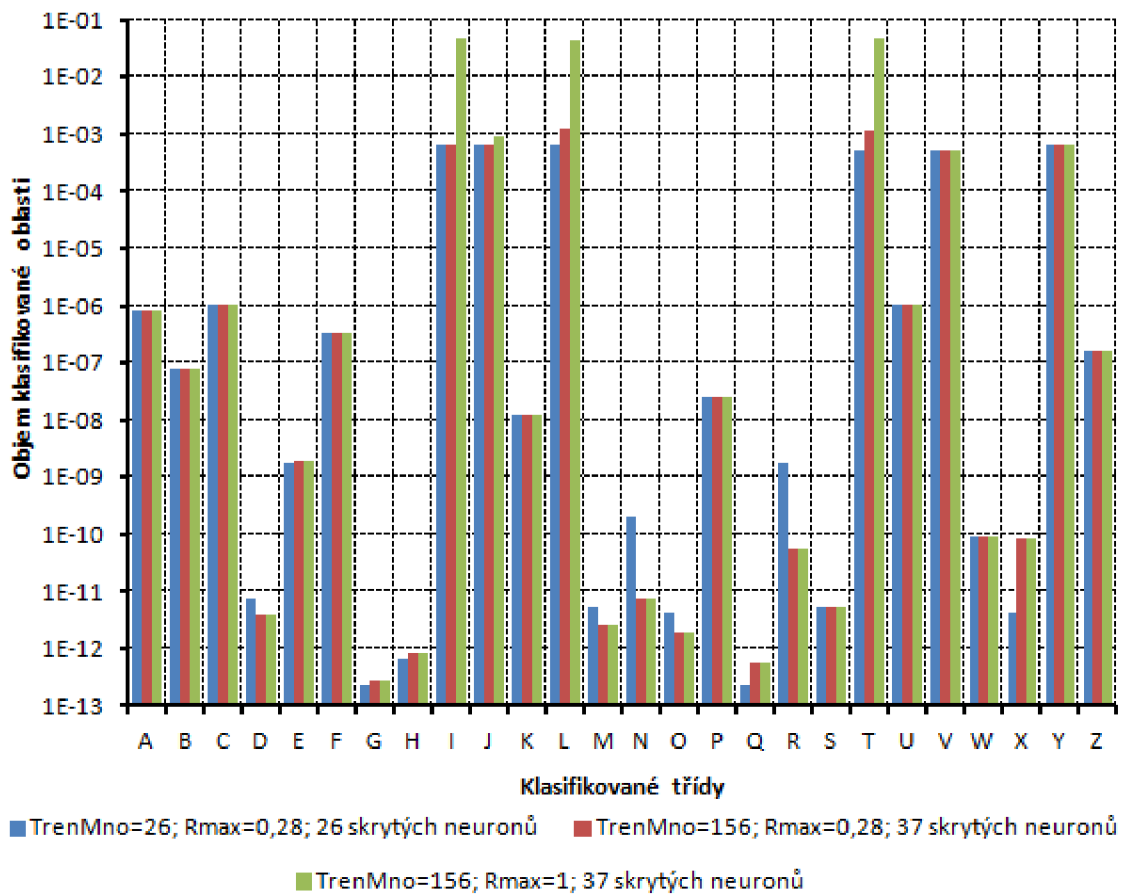
Doposud jsme se věnovali kvalitě rozpoznání, rychlosti učení a počtu skrytých neuronů, ale díky programu uTeachRCE je možno porovnat i objem jednotlivých klasifikovaných oblastí při různém nastavení RCE sítě. Program uTeachRCE pro každý skrytý neuron uchovává informaci o velikosti jeho poloměru. Abychom získali objem jednotlivých tříd, tak budeme počítat objem hyperkoule [MM04] v 7-rozměrném prostoru dle vzorce (6.2), kde parametr r určuje poloměr hyperkoule a znak Γ vyjadřuje gama funkci. Při výpočtu objemů hyperkoulí u stejných klasifikovaných tříd budeme zanedbávat případné překrytí hyperkoulí.

$$V = \frac{r^7 \pi^{\frac{7}{2}}}{\Gamma\left(\frac{7}{2} + 1\right)} = \frac{r^7 \pi^3 2^4}{105} \quad (6.2)$$

Na Obr. 20 je graf vyjadřující objem klasifikované oblasti pro jednotlivé klasifikované třídy u tří RCE sítí. Pro svislou osu grafu bylo zvoleno logaritmické měřítko, aby bylo možné přehledně zobrazit objemy všech klasifikovaných tříd. Modře je znázorněna RCE síť naučená na množinu A (viz kapitola 6.2), pro kterou bylo vytvořeno 26 skrytých neuronů. Červeně je zobrazena RCE síť naučená na spojení množiny A a B (156 vzorů), která má 37 skrytých neuronů. Pro obě sítě byl zvolen stejný poloměr R_{MAX} , aby porovnání objemů u klasifikovaných tříd bylo vypovídající. Byla zvolena hodnota 0,28, protože při této hodnotě obě sítě dosahují maximální úspěšnosti rozpoznání písmen. Srovnáním objemů bylo zjištěno, že změny nastaly u 12 klasifikovaných tříd. Větší změny v objemech jsou viditelné z grafu, jinak kompletní tabulka výpočtu objemů je uvedena v příloze 1.

Zeleně zobrazená RCE síť je stejná jako modře zobrazená síť s tím rozdílem, že poloměr R_{MAX} byl zvětšen na hodnotu 1, abychom zjistili, jak moc se objemy klasifikovaných tříd změny. I když už u hodnoty $R_{MAX} = 0,28$ měla RCE síť stoprocentní rozpoznání množiny C. Změny v objemech nastaly pouze u pěti klasifikovaných tříd, ostatní klasifikované oblasti zřejmě nemohly být zvětšeny kvůli blízkosti klasifikovaných oblastí. Sledováním poloměru skrytých neuronů pro $R_{MAX} = 1$ bylo zjištěno, že maximální hodnota poloměru skrytých neuronů je 0,52.

Porovnáním grafů na Obr. 15 a 20 bylo zjištěno, že většina klasifikovaných tříd (G, H, M, N, Q), u kterých se vyskytovaly stoprocentně nerozpoznané vzory, zaujímá velmi malé objemy ve vstupním prostoru.



Obr. 20: Graf vyjadřující objem klasifikované oblasti u klasifikovaných tříd

Na závěr bylo provedeno dodatečné testování rozpoznání množiny C u sítí backpropagation. Pro testování bylo vybráno 14 sítí (11 třívrstevých a 3 čtyřvrstevé), které byly v rámci bakalářské práce vybrány jako nejúspěšnější v rozpoznání množiny B (všechny sítě dosáhly rozpoznání 99,23%). Tyto sítě jsou naučeny pouze na množinu A, nebyly doučeny na množinu B. Všechny sítě měly stoprocentní úspěšnost rozpoznání množiny C jako RCE síť, ale je nutno poznamenat, že doba učení se pohybuje od 3800 do 16700 učících cyklů. Seznam těchto sítí s uvedeným nastavením všech parametrů je v příloze 1.

6.4 Srovnání RCE sítě a vícevrstvé sítě s učícím algoritmem backpropagation

V kapitole 6.3 (Obr. 16) už byl naznačen zásadní rozdíl při rozdělování vstupního prostoru pomocí RCE a backpropagation sítě. Na Obr. 21 jsou tři typy rozdělení vstupního prostoru pro RCE a backpropagation síť. Ukázky rozdělení vstupního

prostoru pro síť backpropagation byly získány pomocí bezplatného software Sharky Neural Network¹. Pro zjištění rozdělení vstupního prostoru u RCE sítě byl použit program RCEin3D, viz. kapitola 7. Použité tréninkové množiny pro obě sítě byly získány pomocí software Sharky Neural Network, který má funkce na generování bodů pro předdefinované typy rozdělení vstupního prostoru.

Pro demonstraci rozdělení vstupního prostoru byly vybrány tři typy rozdělení: "vlna", kruh a prstenec. Při učení sítě backpropagation byla vybrána možnost, aby se přeúčila. Toto nastavení bylo zvoleno, aby srovnání rozdělení vstupního prostoru bylo nejvíce vypovídající. Nastavení parametrů všech použitých sítí je uvedeno v Tab. 1.

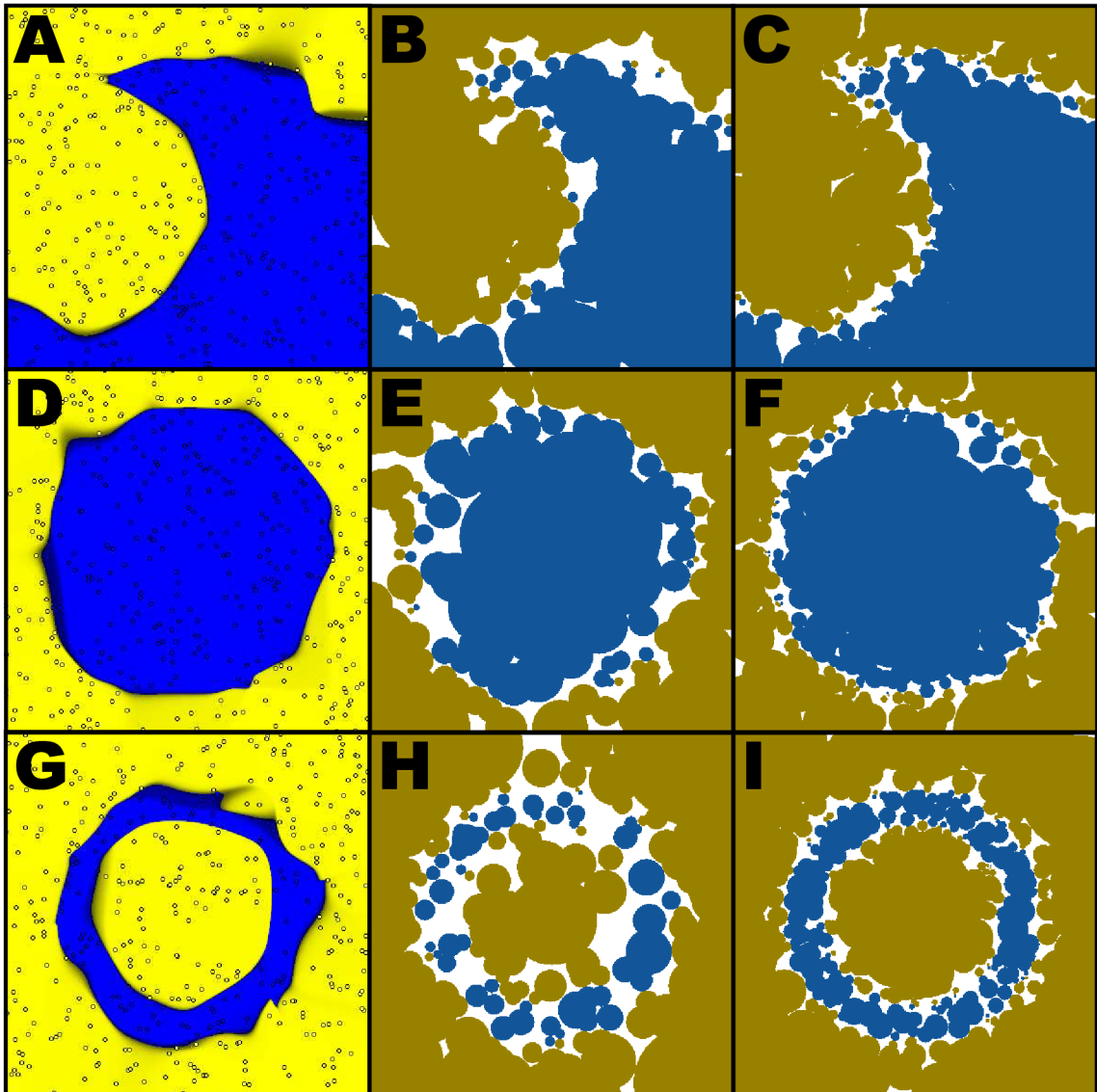
Tab. 1: Parametry sítí použitých při ukázkách rozdělení vstupního prostoru

Tréninková množina	Označení na Obr. 21	Typ sítě	Topologie sítě	Ostatní parametry sítě	Počet cyklů
400 vzorů	A	BP	2-10-10-2	$\eta = 0,1; \alpha = 0,5$	901
	B	RCE	2-106-2	$R_{MAX} = 0,5$	4
	D	BP	2-10-10-2	$\eta = 0,1; \alpha = 0,5$	1736
	E	RCE	2-120-2	$R_{MAX} = 0,4$	3
	G	BP	2-10-10-2	$\eta = 0,1; \alpha = 0,5$	2026
	H	RCE	2-144-2	$R_{MAX} = 0,5$	4
1000 vzorů	C	RCE	2-179-2	$R_{MAX} = 0,25$	4
	F	RCE	2-229-2	$R_{MAX} = 0,5$	4
	I	RCE	2-304-2	$R_{MAX} = 0,5$	4

Pozn.: BP = vícevrstvá síť s učícím algoritmem backpropagation

Na Obr. 21A, 21D a 21G je zobrazeno rozdělení vstupního prostoru sítí backpropagation pro tréninkovou množinu o 400 bodech (body tréninkové množiny jsou v obrázcích zobrazeny jako malé kruhy). Na stejnou tréninkovou množinu byla naučena i RCE síť, výsledky jsou zobrazeny v polích B, E a H na Obr. 21. Obě sítě byly schopny klasifikovat daný problém se stoprocentním rozpoznáním tréninkové množiny, ale RCE síť (4 až 5 cyklů) se učily mnohonásobně rychleji než síť backpropagation (901 až 2026 cyklů). Avšak u RCE sítě nastaly výše zmíněné problémy (viz kapitola 6.3, Obr. 16), kdy se ve vstupním prostoru objevily neklasifikované oblasti reprezentované bílou barvou (v největší míře se objevily na Obr. 21H při rozdělení do prstence).

¹ Software Sharky Neural Network je dostupný z: < <http://www.sharktime.com> >.



Obr. 21: Ukázka tří rozdělení vstupního prostoru pomocí vícevrstvé sítě s učением backpropagation (pole A, D, G), RCE sítě (pole B, E, H) a RCE sítě s větší tréninkovou množinou (pole C, F, I)

U klasifikace rozdělení vstupního prostoru do kruhu RCE sítí (Obr. 21E) lze dosáhnout i lepších výsledků. A to tak, že RCE síť učíme pouze na jednu třídu (modrá oblast) a všechny ostatní body budou náležet do druhé třídy (žlutá oblast). Pokud ještě provedeme předzpracování dat, abychom našli střed daného shluku, tak při poloměru $R_{MAX} = 0,86$ bude RCE síť potřebovat pouze jeden skrytý neuron. Tento postup klasifikace dvou tříd se v praxi používá při segmentaci lidské kůže [YGX01], [SKR11].

Pro RCE síť bylo navíc vyzkoušeno, jak se rozdělení vstupního prostoru změní při rozsáhlejší tréninkové množině. Na Obr. 21C, 21F a 21I je zobrazeno rozdělení RCE

sítě pro 1000 bodů tréninkové množiny. U všech tří typů rozdělení bylo dosaženo lepší rekonstrukce skutečného tvaru rozdělení při rozšíření tréninkové množiny.

Zvětšením tréninkové množiny 2,5-krát mělo za následek následující změny u jednotlivých rozdělení:

- U rozdělení "vlna" se počet skrytých neuronů RCE sítě 1,7-krát zvětšil, plocha neklasifikované oblasti se zmenšila 1,48-krát.
- Pro rozdělení do kruhu došlo ke zvýšení počtu skrytých neuronů 1,9-krát a plocha neklasifikované oblasti se také zmenšila 1,48-krát.
- U rozdělení do prstence byly změny největší, počet skrytých neuronů se 2,1-krát zvětšil. Plocha neklasifikované oblasti se zmenšila dvojnásobně.

Dalším důležitým aspektem, který jsme při srovnání nezhodnotili je čas učení a vybavování sítí. Časy učení a vybavování byly změřeny na počítači se čtyřjádrovým procesorem Intel Core i7-2670QM (2,2GHz) a vnitřní pamětí 8GB.

Pro měření časů u sítě backpropagation byly jako zástupci vybrány jedna třívrstvá a čtyřvrstvá síť, které měly nejmenší počet učicích cyklů (sítě byly vybrány z 14 sítí backpropagation, které dosáhly 99,23% rozpoznání množiny B). V Tab. 2 je uvedeno nastavení všech parametrů sítí společně se změřenými časy. Čas t_U je doba učení sítě celé množiny A (26 vzorů), v tabulce je uvedena průměrná hodnota z pěti měření. Čas učení u obou sítí se pohybuje v jednotkách sekund, ale učení třívrstvé sítě je dvakrát rychlejší než u čtyřvrstvé. Čas t_V je doba vybavování jednoho testovacího vzoru sítí, v tabulce je uvedena průměrná hodnota z 26000 měření.

Tab. 2: Rychlost učení a vybavování vícevrstvé sítě s učením backpropagation

Topologie	Počáteční váhy	Koeficient učení	Koeficient momentu	Strmost	Počet cyklů	t_U [s]	t_V [μ s]
7-10-26	<-0,6; 0,6>	0,9	0,6	1	4990	17,27	48,41
7-60-80-26	<-0,5; 0,5>	0,5	0,6	1	3835	35,31	77,46

V Tab. 3 jsou uvedeny změřené časy pro RCE síť při učení množiny A (26 vzorů) jako u sítě backpropagation a rozšířené tréninkové množiny (spojení množiny A a B, 156 vzorů). Uvedený čas t_U je průměrná hodnota ze 100 měření, t_V je průměrná hodnota z 26000 měření.

Srovnáním Tab. 2 a 3 je zřejmé, že časová náročnost učení 26 vzorů u RCE sítě je mnohonásobně rychlejší než u sítě backpropagation, ale kvalita rozpoznání písmen srovnatelná není (RCE 90,77%, backpropagation 99,23%). Proto byl změřen i čas učení RCE sítě na spojenou množinu A a B, kdy je kvalita rozpoznání písmen téměř srovnatelná (RCE 100%, backpropagation 99,23%). Čas učení RCE sítě na spojenou množinu A a B (156 vzorů) je téměř 16-krát pomalejší než při učení na množinu A

(26 vzorů), ale i přesto učení této RCE sítě je přibližně 80-krát rychlejší než učení třívrstvé sítě s algoritmem učení backpropagation.

Tab. 3: Rychlost učení a vybavování RCE sítě

Tréninková množina	Topologie	Počáteční poloměr	Počet cyklů	t_U [ms]	t_V [μ s]
Množina A	7-26-26	0,26	2	13,32	24,27
Množina A+B	7-37-26	0,28	2	212,07	27,30

Čas vybavování jednoho testovacího vzoru t_V se u obou typů sítí pohybuje ve stejných řádech, ale vybavování RCE sítě je přibližně 2-krát rychlejší než vybavování třívrstvé sítě s algoritmem učení backpropagation. Rozdíly v časech vybavování t_V nejsou tak markantní, jako u časů učení sítě t_U .

Rozhodnout, která ze sítí je lepší nelze jednoduše říci: výběr neuronové sítě by se měl odvíjet od typu její aplikace. Pro snadnější rozhodnutí mezi RCE sítí a vícevrstvou sítí s algoritmem učení backpropagation jsou v Tab. 4 přehledně shrnuty jejich výhody a nevýhody.

Tab. 4: Výhody a nevýhody vícevrstvé sítě s učením backpropagation a RCE sítě

VÝHODY	
BACKPROPAGATION	RCE
<ul style="list-style-type: none"> vhodně vybraná tréninková množina správně rozdělí celý vstupní prostor (Obr. 16 a 21) 	<ul style="list-style-type: none"> rychlejší učení nemůže dojít k zaseknutí učícího algoritmu naučení tréninkové množiny je vždy bezproblémové pouze jeden parametr k nastavení neznámý vzor nemusí být klasifikován do některé ze tříd možnost použít síť jako aktivní paměť
NEVÝHODY	
BACKPROPAGATION	RCE
<ul style="list-style-type: none"> pomalejší učení může dojít k uváznutí učícího algoritmu mnoho parametrů k nastavení neznámý vzor je vždy klasifikován do některé ze tříd 	<ul style="list-style-type: none"> pro rozdělení vstupního prostoru potřebuje bohatou vhodně zvolenou tréninkovou množinu (Obr. 16 a 21)

Pro aplikaci rozpoznávání písmen je vhodnější RCE síť z těchto důvodů:

- Nastavení parametrů RCE sítě je jednodušší a učení sítě je rychlejší.
- U aplikace rozpoznávání písmen je velmi vhodná vlastnost RCE sítě: aktivní paměť.
- Proces učení RCE sítě je průhlednější: možnost zjistit, kolem kterých vzorů se vytvořily skryté neurony. Dále je možno zjistit orientační uspořádání klasifikovaných tříd ve vstupním prostoru (podle poloměru a středu skrytých neuronů).

6.5 Rozpoznávání písmen pomocí fuzzy RCE sítě

Tato kapitola obsahuje shrnutí výsledků semestrálního projektu vypracovaného v rámci předmětu Fuzzy Systémy (MFSY) zabývající se rozpoznáním písmen pomocí umělé neuronové fuzzy RCE sítě [RCHF93].

Algoritmus fuzzy RCE sítě je téměř stejný, jako algoritmus RCE sítě popsany v kapitole 3.4. Hlavní rozdíl je ve skrytých neuronech. U fuzzy RCE sítě by se každý skrytý neuron dal nazvat, jako "fuzzy buňka", protože každý neuron má svoji funkci příslušnosti trojúhelníkového tvaru s definovaným středem a šířkou. Střed funkce příslušnosti se nemění, během učení se zmenšuje pouze šířka funkce příslušnosti, která byla na začátku učení nastavena na počáteční maximální hodnotu. Kompletní popis fuzzy RCE sítě společně se vzorci je uveden v práci [RCHF93]. Bližší popis fuzzy logiky je uveden v knize [Jur03].

V příloze 1 je program uTeachFuzzyRCE pro učení fuzzy RCE sítě, který je téměř totožný jako program uTeachRCE z kapitoly 5. Fuzzy RCE síť byla naučena na množinu A, kvalita rozpoznání písmen byla testována pomocí množiny B. Kvalita rozpoznání písmen byla zkoumána v závislosti na dvou parametrech: počáteční hodnota šířky funkcí příslušnosti b_{MAX} a prah TH. Práh TH se používá při zmenšení šířky funkce příslušnosti u skrytých neuronů, které byly aktivní pro tréninkový vzor jiné klasifikované třídy.

Nejllepší hodnoty (99,23%) rozpoznání písmen množiny B bylo dosaženo pro parametr $b_{MAX} = 1$ a $TH = 0,9$, kdy se fuzzy RCE síť naučila množinu A za 12 cyklů.

Zavedením funkcí příslušnosti do umělé neuronové sítě RCE byla získána lepší hodnota rozpoznání písmen množiny B (zlepšení o 9%). Ale došlo ke snížení rychlosti učení: fuzzy RCE síť potřebovala 6-krát více učících cyklů než RCE síť. U fuzzy RCE sítě se přestala (až na pár výjimek) projevovat vlastnost RCE sítě popisovaná v kapitole 6.2, že všechny špatně rozpoznané vzory nejsou zařazeny do některé z klasifikovaných tříd.

7 PROGRAM RCEin3D PRO VIZUALIZACI UČENÍ RCE SÍTĚ

Program RCEin3D navazuje na program uTeachRCE popsaný v kapitole 5. Jeho hlavním účelem je vizualizace učení RCE sítě a jeho cílem je usnadnit pochopení průběhu učení RCE sítě při výuce.

Při testování vytvořeného programu RCEin3D bylo zjištěno, že je užitečný především pro zkoumání vlastností RCE sítě a pro výukové účely. A proto po domluvě s vedoucím diplomové práce začal být program využíván v počítačových laboratořích předmětu Umělá inteligence (MUIIN), kde se umělá neuronová síť RCE vyučuje. Pro počítačové laboratoře předmětu MUIIN byly i navrženy laboratorní úlohy, které budou studenti řešit při výuce. Jejich popis je uveden v kapitole 7.8.

Příloha 1 obsahuje zdrojový kód programu a zkompilovaný program s potřebnými knihovnamí nutnými k jeho funkčnosti. Dále je v příloze 1 uveden i zkompilovaný program pro cvičení MUIIN, který byl doplněn o kontrolní soubor Check.xml (zda je uživatel oprávněn k používání programu).

Program RCEin3D samostatně provádí volbu mezi dvěma základními typy zobrazení RCE sítě, volba se provádí podle počtu klasifikovaných tříd (ostatní funkce a nastavení programu popisovaných v následujících kapitolách jsou stejné):

- Zobrazení do tří tříd: zobrazení RCE sítě je uzpůsobeno pro větší přehlednost naučené sítě. Toto zobrazení popsané v kapitole 7.2 je vhodné pro výuku RCE sítě, proto i všechny navržené úlohy v kapitole 7.8 mají pouze tři klasifikované třídy.
- Zobrazení do více tříd: při tomto zobrazení je kladen zejména důraz na přehlednost RCE sítě při velkém počtu klasifikovaných tříd, odlišnosti při tomto zobrazení jsou popsány v kapitole 7.7

7.1 Návrh prostředí pro realizaci programu

Program je vytvořen v programovacím prostředí Microsoft Visual Studio 2008, kde se využívá programovací jazyk C++. Kvůli požadavku na vizualizaci učení RCE sítě ve 3D prostoru je vhodné použít externí knihovny v jazyce C++.

Pro vizualizaci učení je využíván open-source software Delta3D¹. Delta3D se zaměřuje na modelování, simulaci, virtuální realitu atd. Součástí Delta3D je i několik dalších open-source produktů. Jeden z těchto produktů je open-source knihovna

¹ Software Delta3D je dostupný z: <<http://www.delta3d.org/>>. V této práci se využívá poslední verze 2.7.4 z roku 2008.

pro modelování a simulace OpenSceneGraph², dále jen OSG. Právě kvůli přítomnosti knihovny OSG byl vybrán software Delta3D.

Toto řešení umožní programování pomocí nástrojů OSG knihovny (např. vytváření modelů geometrických primitiv), tak i využívání nástrojů Delta3D, jenž usnadňují práci s knihovnou OSG (např. práce s modely zobrazovaných ve scéně).

Pro snadnější pochopení nástrojů OSG a Delta3D bylo vytvořeno mnoho tutoriálů různých obtížností. Používáním nástrojů knihovny OSG se zabývá i kniha OpenSceneGraph3.0: Beginner's Guide [WQ10].

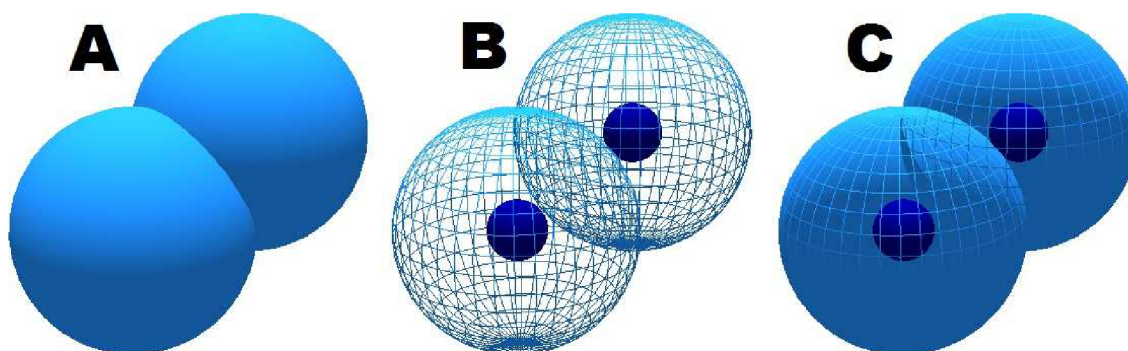
Pro práci se softwarem Delta3D je nutná jeho instalace do počítače. Avšak při používání programu využívajícího nástroje Delta3D na jiném počítači není nutné software Delta3D opět instalovat. K vytvořenému programu je pouze nutné nahrát knihovny z Delta3D, které v programu využíváme.

7.2 Zobrazení RCE sítě programem při klasifikaci do tří tříd

V této kapitole je popsáno zobrazení RCE sítě pro tři klasifikované třídy. Jelikož vizualizace učení RCE sítě probíhá ve 3D prostoru, tak je zřejmé omezení ve vzorech předkládaných síti: vzory mohou mít pouze tři příznaky (v programu označené jako souřadnice 3D prostoru: X, Y, Z).

Filozofie zobrazení naučené sítě je taková, že pro odlišení jednotlivých tříd se používá modrá, červená a zelená barva. Protože jsme ve 3D prostoru, tak tréninkové body i hyperkoule skrytých neuronů jsou zastoupeny koulemi. Odstín koulí tréninkových bodů je mnohem tmavší než odstín koulí skrytých neuronů, tato skutečnost je názorná z Obr. 22.

Programovací prostředí má nástroje pro změnu renderování grafických objektů ve scéně. To je využíváno pro změnu zobrazení RCE sítě, jak je vidět na Obr. 22.



Obr. 22: Ukázka tří zobrazení RCE sítě

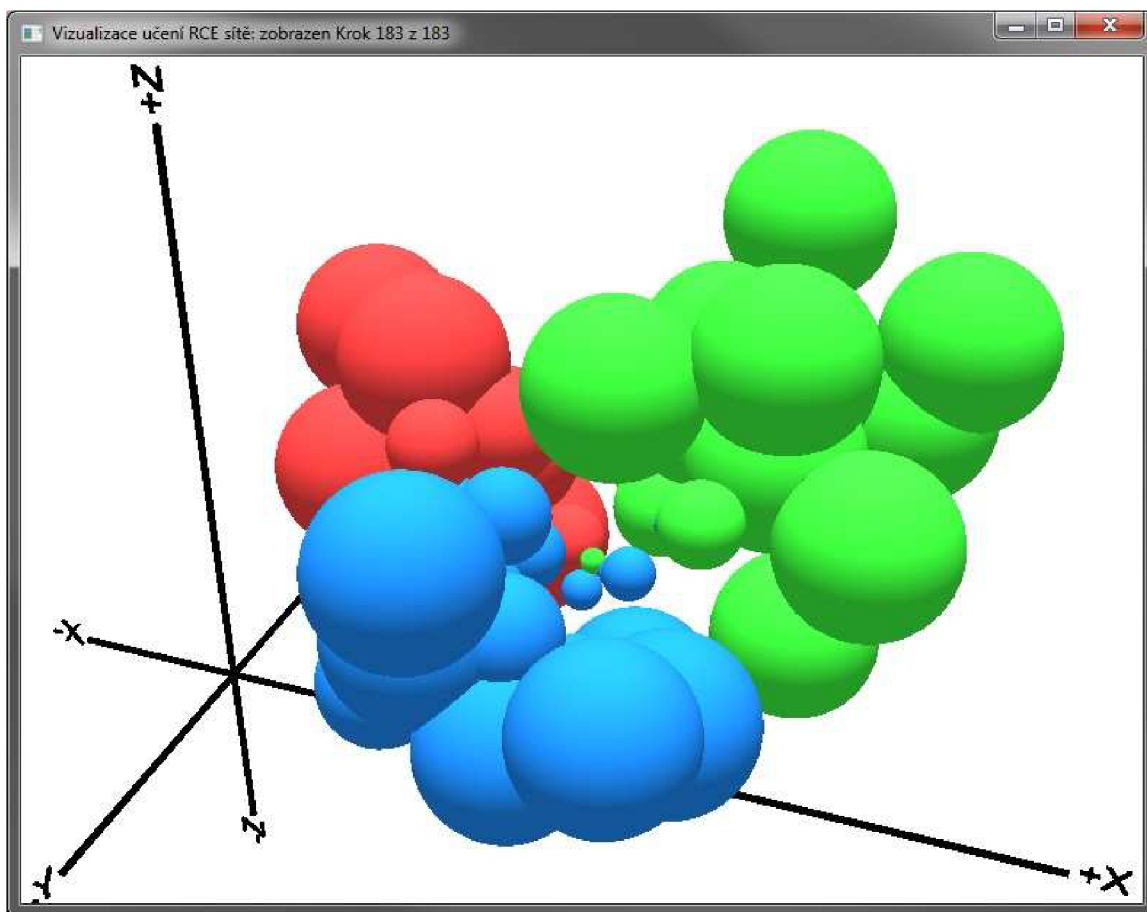
² Knihovna OSG je dostupná z: <<http://www.openscenegraph.org>>.

Program umožňuje střídat tři zobrazení:

- Zobrazení modelu se stínovanými plochami (Obr. 22A): koule tréninkových bodů nejsou vidět, koule neuronů jsou vykresleny plnými plochami.
- Zobrazení drátového modelu (Obr. 22B): koule neuronů jsou vykresleny, jako drátový model. Tréninkové body jsou vykresleny plnými plochami.
- Kombinace dvou předchozích zobrazení (Obr. 22C): jedná se o zvláštní případ zobrazení, kdy přední část (z pohledu kamery) koule neuronu je vykreslena, jako drátový model, a zbytek koule je vykreslena plnými plochami. Tréninkové body jsou vykresleny plnými plochami.

7.3 Popis programu

Program RCEin3D komunikuje s uživatelem prostřednictvím dvou oken. První okno, dále grafické okno, slouží k vizualizaci učení RCE sítě pomocí 3D grafiky. Ukázka grafického okna naučené RCE sítě je na Obr. 23. Celý program se vypíná zavřením grafického okna.



Obr. 23: Ukázka grafického okna programu

Druhé okno, dále textové okno, má spíše informativní charakter a slouží k výpisu důležitých nastavení a parametrů RCE sítě. Je rozděleno na logické bloky: ovládání programu, vstupní parametry sítě, výstupní parametry sítě, testování sítě. Ukázka textového okna bez bloku ovládání programu je na Obr. 24.

```

d:\Diplomka\RCE\Release\RCE.exe
===== USTUPNI PARAMETRY RCE SITE =====
Maximalni (pocatecni) polomer Rmax = 5
Pocet vzoru v treninkove mnozine = 3

Treninkove vzory:
X      Y      Z      TRIDA
8.751  7.023  8.774  1-Cervena
13.321 10.225 13.567 1-Cervena
24.24  17.56  17.287 2-Zelena

===== VYSTUPNI PARAMETRY RCE SITE =====
Pocet ucicich cyklu = 2
Pocet neuronu skryty vrstvy = 3

Parametry neuronu skryte vrstvy:
X      Y      Z      TRIDA      R
8.751  7.023  8.774  1-Cervena  5
13.321 10.225 13.567 1-Cervena  5
24.24  17.56  17.287 2-Zelena   5

===== TESTOVANI RCE SITE =====
Pocet vzoru v testovaci mnozine = 3
Pocet spatne klasifikovanych vzoru = 2
Uspesnost RCE site na testovaci mnozine = 33.3333%

Testovaci vzory:
X      Y      Z      TRIDA      TRIDA_RCE      POZN.
8.751  7.023  8.774  1-Cervena  1-Cervena
11.308 10.319  7.483  3-Modra   1-Cervena      ERR
9.912  -1.23  3.626  2-Zelena  0
cislo tridy = 0

Ne sloupci POZN jsou spatne klasifikovane vzory oznaceny pomoci symbolu 'ERR'
Ne sloupci TRIDA_RCE maji vzory, které nebyly klasifikovany do zadne tridy,
cislo tridy = 0

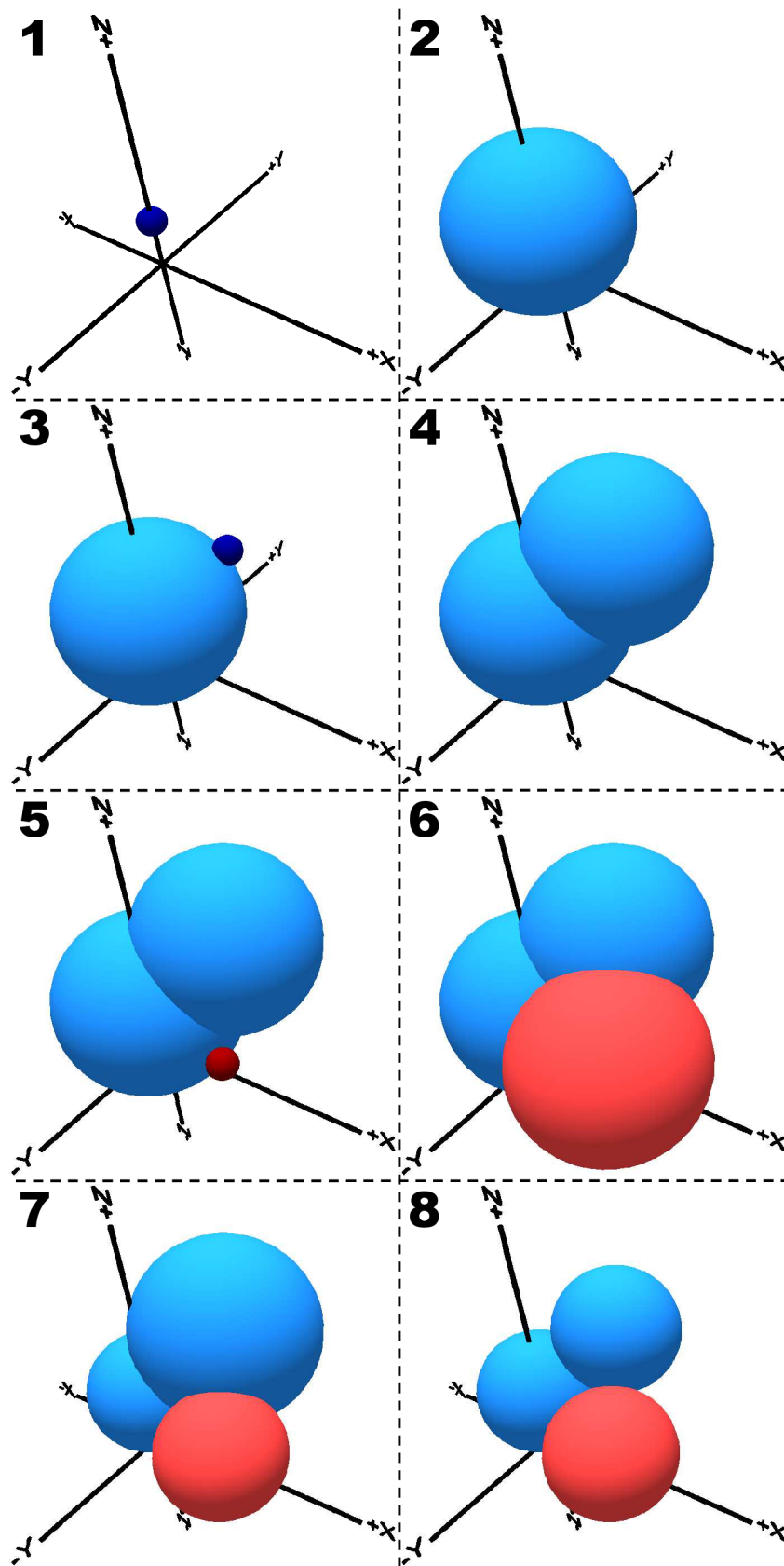
```

Obr. 24: Ukázka textového okna programu

Program nezobrazuje pouze konečnou podobu naučené RCE sítě, ale umožňuje i zobrazovat jednotlivé kroky v průběhu učení RCE sítě. Pomocí klávesových zkratk je možno prohlížet průběh učení RCE sítě. Číslo aktuálně zobrazovaného kroku je vypisováno v záhlaví grafického okna, viz Obr. 23. Důvod k vytvoření nového učicího kroku jsou tyto tři případy:

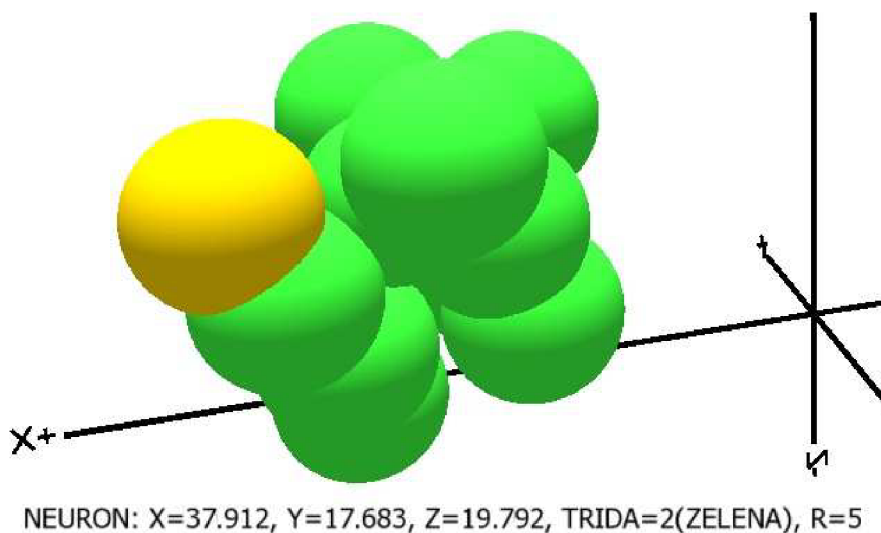
- Přidání nového tréninkového bodu (při prvním učícím cyklu se tréninkové body přidávají do scény).
- Vytvoření nového skrytého neuronu.
- Změna poloměru koule u již vytvořeného neuronu.

Ukázka krokování RCE sítě je na Obr. 25, kde je zobrazeno osm učících kroků nutných k naučení tří tréninkových vzorů.



Obr. 25: Příklad krokování RCE sítě

Program má také funkci pro zvýraznění koule reprezentující neuron nebo tréninkový bod. Takto zvýrazněný objekt je označen zlatou barvou a ve spodní části grafického okna se zobrazí doprovodný text pro kouli (souřadnice ve 3D prostoru, označení třídy, v případě neuronu i poloměr koule). Ukázka zvýraznění neuronu je na Obr. 26.



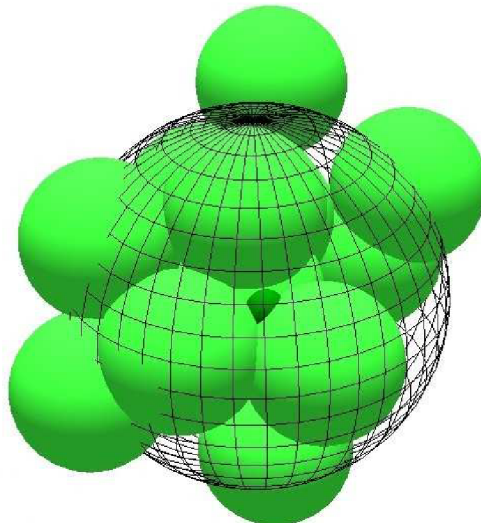
Obr. 26: Zvýraznění neuronu RCE sítě (barva zvýraznění je zlatá)

Program sice zobrazuje osy souřadného systému, ale nezobrazuje měřítko jednotlivých os. A proto je toto zvýraznění výhodné, abychom zjistili souřadnice tréninkového bodu nebo neuronu. Objekt zůstává zvýrazněný stále, i když dochází ke krokování učení sítě. To může být výhodné, pokud např. chceme sledovat jeden neuron během průběhu učení sítě. Zvýraznění je možné zrušit klávesou DELETE.

V programu je zakomponována i funkce, která posuzuje kvalitu naučení jednoho shluku dat. Tato funkce vznikla zejména pro potřeby počítačových laboratoří předmětu MUIN. Funkci je rozumné volat pouze na shluky dat (ideálně s normálním náhodným rozložením), např. podobný shluk jako je uveden na Obr. 27A.

Funkci je nutno zadat vstupní parametry: číslo klasifikované třídy a poloměr koule, ve které zkoumáme kvalitu naučení shluku. Na hodnotě poloměru shluku závisí časová náročnost funkce. Funkce nejprve zjistí střed shluku tréninkových bodů dané třídy a poté vypočítá procentuální zastoupení jednotlivých tříd (i prostoru, který nenáleží žádné třídě) ve zkoumané oblasti. Způsob zadávání vstupních dat a výpis funkce je zřejmý z Obr. 27B.

Po dokončení výpočtu funkce se v grafickém okně zobrazí černý drátový model koule, ve které zkoumáme kvalitu naučení shluku. Toto zobrazení nám dopomůže k lepší vizuální představě naučené RCE sítě v oblasti výskytu shluku.

A**B**

```

===== INFORMACE O USPESNOSTI NAUCENI SHLUKU =====
Pojem hlavni koule zastupuje kouli se stredem stejnym, jako je stred shluku,
a polomer urcuje oblast, ve které zkoumame uspesnost nauceni RCE site.
Zadejte polomer hlavni koule (kladne realne cislo), R = 12
Zadejte tridu shluku (cele cislo) z intervalu 1 az 3 , TRIDA = 2
Uppocitany stred shluku (hlavni koule) je (28.4743, 10.6649, 15.2843)

Procentualni zastoupeni objemu jednotlivych trid v hlavni kouli je:
TRIDA      OBJEM [%]
Neurceno   28.60
1-Červena  0.00
2-Zelena   71.32
3-Modra    0.00
  
```

Obr. 27: Funkce pro zhodnocení úspěšnosti naučení shluku

7.4 Ovládání programu

Program se ovládá pomocí myši a klávesových zkratk použitých pro volání jednotlivých funkcí programu. Zadávání klávesových zkratk funguje pouze pokud je aktivní grafické okno. Ovládání programu je následující:

- ESC - ukončení programu,
- LEVÉ TL. MYŠI - otáčení scény,
- PRAVÉ TL. MYŠI - posouvání scény,
- STŘEDNÍ TL. MYŠI - přibližování/oddalování scény,
- F(f) - zobrazení modelu se stínovanými plochami,
- L(l) - zobrazení drátového modelu,
- K(k) - kombinace dvou předchozích zobrazení,
- C(c) - nastavení kamery do počáteční polohy,
- O(o) - výpočet úspěšnosti naučení jednoho shluku,
- ŠIPKA VPRAVO - krok dopředu v učení RCE sítě,
- ŠIPKA VLEVO - krok dozadu v učení RCE sítě,
- HOME - první krok v učení RCE sítě,
- END - poslední krok v učení RCE sítě,
- ŠIPKA NAHORU - posun ve výběru modelu neuronu/bodu pro zvýraznění,

- ŠIPKA DOLU - posun ve výběru modelu neuronu/bodu pro zvýraznění,
- DELETE - zrušení zvýraznění modelu neuronu/bodu.

Myš se využívá pro pohyb kamery ve scéně. Pod pojmem kamera rozumíme virtuální kameru (objekt programovacího prostředí Delta3D), pomocí které jsme schopni prohlížet scénu. Například pokud chceme scénu otočit, tak držíme levé tlačítko myši a pohybem myši určíme směr natočení. Scéna se otáčí kolem bodu, ve kterém je umístěna kamera. Polohu umístění kamery ovlivníme použitím pravého tlačítka myši, které nám slouží k posouvání scény.

7.5 Vstupní data programu

Vstupní data programu jsou obdobná jako vstupní data programu uTeachRCE z kapitoly 5.2. Jedna z odlišností je, že parametr m udávající počet tříd je celé číslo z intervalu $\langle 1;120 \rangle$. Počet souřadnic bodu je omezen na tři (X, Y, Z), souřadnice jsou libovolná reálná čísla.

V kapitole 7.2 bylo uvedeno, že každá třída je reprezentována barvou, ale do XML souboru zadáváme třídu, jako celé číslo. A proto při zobrazení do tří tříd platí, že třída označená číslem jedna je reprezentována červenou barvou, dvojka modrou a trojka zelenou.

Při zadávání vstupních dat je vhodné, aby se hodnoty souřadnic bodů pohybovaly v jednotkách až desítkách. Toto opatření zajišťuje přehlednost naučené RCE sítě, protože program neprovádí normalizaci vstupních dat.

Parametry zobrazení programu RCEin3D (rozlišení obrazovky, režim celé obrazovky atd.) se nastavují v XML souboru (config.xml). Tento soubor byl automaticky vytvořen prostředím Delta3D.

7.6 Funkce programu

Program RCEin3D převzal základní strukturu LearnStructure programu uTeachRCE, její popis je v kapitole 5.3. Byla provedena jediná změna: do struktury přidána proměnná Draw typu bool.

Pro krokování učení sítě je použito jednorozměrné pole AllStepsRCE (velikost je určena počtem učících kroků), jejíž prvky jsou jednorozměrná pole LearnSet (kapitola 5.3), které uchovávají konfiguraci RCE sítě v daném kroku učení. Při vizualizaci učení sítě je využívána proměnná Draw. Pokud je v daném kroku Draw nastaveno na hodnotu True, tak to značí, že bod/neuron (výběr se řídí proměnnou ShowSphere) má být v tomto kroku učení sítě vykreslen ve scéně. Hodnota Draw je nastavována už během učení sítě.

Program RCEin3D také převzal funkce pro obsluhu RCE sítě (ReadXML, LearnRCE, AddHideNeuron, TestRCE, EuclideanDist) z programu uTeachRCE, jejich popis je v kapitole 5.4. Do funkcí byly přidány nové kroky, které zjednodušují práci při vizualizaci učení sítě. Ve funkci ReadXML se plní globální proměnná `BoundsForAxis`, která uchovává mezní hodnoty bodů tréninkové množiny ve všech osách. Při vykonávání funkcí LearnRCE a AddHideNeuron dochází k ukládání jednotlivých kroků učení sítě do proměnné `AllStepsRCE`.

V následujících odstavcích jsou popsány funkce, které zajišťují vizualizaci učení RCE sítě.

FillVectorObjects (private): Každý neuron a tréninkový bod je graficky zastoupen 3D objektem používaných pro vizualizaci. Všechny tyto objekty jsou sjednoceny do dvou vektorů: vektor neuronů a bodů. Hlavní úkol funkce je počáteční nastavení těchto objektů: přiřazení barevného modelu, nastavení souřadnic. U objektů tréninkových bodů nastavíme i měřítko poloměru. U objektů neuronů se měřítko poloměru mění při běhu programu.

NetworkRCE (public): Funkce slouží pro obsluhu RCE sítě. V této funkci dochází k volání privátních funkcí v následujícím pořadí: ReadXML, LearnRCE, TestRCE, FillVectorObjects.

Config (public): Funkce zděděná ze základní třídy Application prostředí Delta3D. Jedná se o konfigurační funkci programu. V rámci této funkce se provádí základní nastavení scény: počáteční nastavení kamery, nastavení modelu pro pohyb myši ve scéně, vytvoření textového pole pro výpis informací na obrazovce, vykreslení os souřadného systému a prvního kroku učení sítě.

PreFrame (public): Funkce zděděná ze základní třídy Application prostředí Delta3D. Jak již vyplývá z názvu, tak je tato funkce volána před vykreslením každého snímku.

KeyPressed (public): Funkce zděděná ze základní třídy Application prostředí Delta3D. Funkce zajišťuje komunikaci programu s klávesnicí. Pro jednotlivé klávesy je možno nadefinovat akce, které mají být provedeny.

SetCameraPosition (private): Funkce slouží k nastavení kamery do počáteční polohy: kamera je zaměřena na první skrytý neuron sítě a je umístěna v jeho blízkosti.

ChangeRenderState (private): Funkce umožňuje měnit zobrazení koulí neuronů, možnosti zobrazení viz kapitola 7.2.

CreateModel (public): Funkce vytváří modely 3D objektů používaných ve scéně. Vytvořené modely se ukládají na disk, funkce modely vytváří jen pokud nejsou již vytvořeny na disku. Funkce může vytvářet koule, válec, 3D text o libovolné barvě. Rozměry těchto modelů jsou ve funkci pevně nastaveny, jejich velikosti mohou být změněny při běhu programu pomocí jiných funkcí.

CreateAxis (public): Funkce vykresluje osy souřadného systému ve 3D prostoru společně s jejich popisky pomocí 3D textu. Osy jsou vykreslovány pouze v rozsahu,

který odpovídá umístění bodů tréninkové množiny (zajištěno proměnnou `BoundsForAxis`). Avšak vždy je zaručeno, že se osy protnou v souřadnicích (0, 0, 0).

CreateAndShow2DText (private): Funkce dle vstupních parametrů vytvoří vrstvu v grafickém okně, do které je možno zapisovat 2D text. V rámci funkce se nastavují i vlastnosti textu v této vrstvě.

ShowStep (private): Funkce slouží k vykreslení aktuálního kroku učení. Je to hlavní funkce zajišťující krokování učení RCE sítě. Číslo aktuálního kroku je uchováváno v globální proměnné. Funkce zobrazuje/nezobrazuje koule neuronů/bodů dle proměnné `Draw`, jenž má každý objekt přidělen díky použití struktury `LearnStructure`, popis struktury viz kapitola 5.3. Všechny kroky učení jsou uloženy v globální proměnné `AllStepsRCE`.

ShowGoldSphere (private): Funkce pro zvýrazňování jedné koule náležící neuronu nebo bodu. Zvýrazněním je myšleno změna barvy koule na zlatou. Měnit zvýraznění neuronů/bodů je možno v obou směrech. Při směru nahoru (ovládání viz kapitola 7.4) se nejprve prochází neurony a pak až body. Při směru dolů je to naopak. Index zvýrazněné koule se uchovává v globální proměnné. Ukázka zvýrazněného neuronu je na Obr. 26.

HideGoldSphere (private): Funkce pro zrušení zvýraznění jedné koule náležící neuronu nebo bodu. Zejména se jedná o navrácení původní barvy (dle třídy) dané kouli.

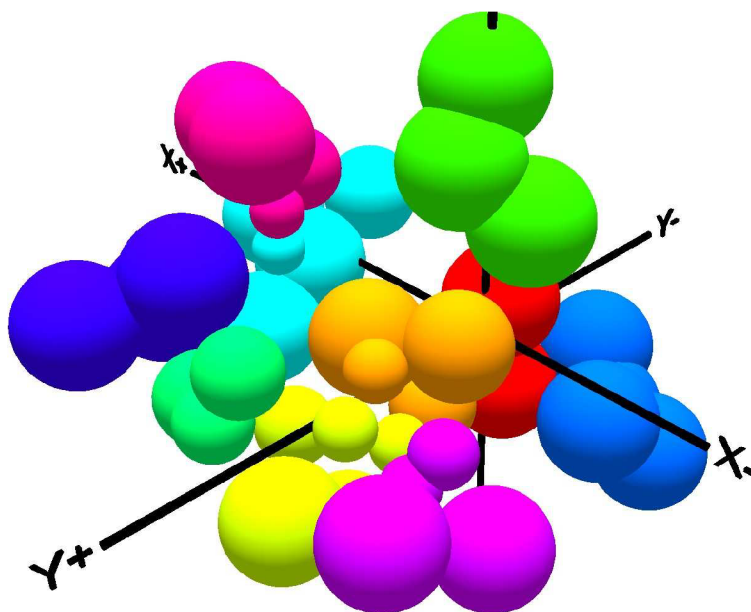
PrintGoldSphereInfo (private): Funkce pro výpis informací o zvýrazněném neuronu nebo bodu. Ukázka výpisu informací pro neuron je na Obr. 26.

CalculateVolume (private): Funkce slouží ke zhodnocení kvality naučení jednoho shluku RCE sítě. Bližší popis funkce je uveden v kapitole 7.3. Na Obr. 27 jsou znázorněny výstupy funkce. Při výpočtu objemů byl použit krok 0,25 což odpovídá přesnosti výpočtu 93,79%.

ReadDoubleNumber, **ReadIntNumber** (private): Funkce pro načtení čísla od uživatele s kontrolou datového typu.

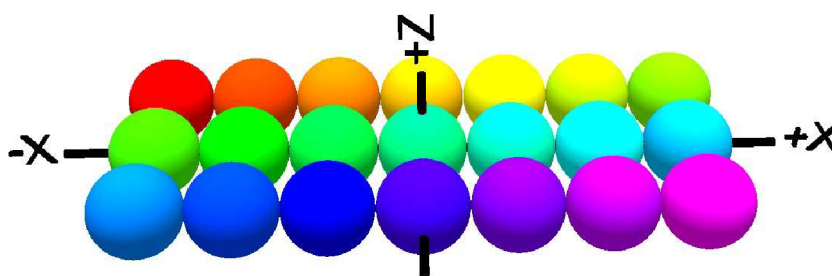
7.7 Zobrazení RCE sítě programem při klasifikaci do více tříd

Ukázka vizualizace RCE sítě při klasifikaci do deseti tříd je na Obr. 28. Aby bylo docíleno většího počtu klasifikovaných tříd, tak se muselo slevit z nároků na přehlednost naučené sítě: barva koulí tréninkových bodů a neuronů je stejná, už nejsou odlišeny různým odstínem barvy, jak tomu bylo u zobrazení do tří tříd (viz Obr. 22). Koule neuronů a tréninkových bodů se nejlépe rozpoznají při zobrazení drátového modelu pomocí klávesy `L(1)`, kdy koule bodů zůstávají vykresleny plnými plochami.



Obr. 28: Ukázka vizualizace RCE sítě při klasifikaci do deseti tříd

Algoritmus vybírá nejvíce odlišné barvy pro daný počet tříd. Algoritmus pro míchání barev vytvoří 120 různých barev, program tedy může být použit pro klasifikaci do 120 tříd. Pro míchání barev v barevném prostoru RGB byl zvolen krok 0,05 (jednotlivé složky jsou z rozsahu $\langle 0;1 \rangle$). Avšak je třeba upozornit, že přehlednost bude rapidně klesat s rostoucím počtem klasifikovaných tříd. Na Obr. 29 vidíme, že odstíny některých barev, vybraných pro klasifikaci do 21 tříd, už mohou lidskému oku splývat. Algoritmus míchání i volba zobrazení pro tři nebo více tříd se řídí počtem klasifikovaných tříd zadaných v elementu POCET_TRID vstupního XML souboru, viz Obr. 10.



Obr. 29: Vybrané barvy pro klasifikaci do 21 tříd

Také byla změněna barva pro zvýraznění koule reprezentující neuron nebo tréninkový bod na hnědou barvu. A to z důvodu, aby nedocházelo ke splývání barvy pro zvýraznění s namíchanými barvami.

Při zobrazení do tří tříd program RCEin3D vždy vypisuje i barevné pojmenování klasifikované třídy ve všech částech programu, viz Obr. 24 a 26. To umožňuje začínajícímu uživateli rychlejší srovnání souvislostí mezi barevným rozlišením tříd v programu a jejich číselným označením ve vstupním XML souboru. U zobrazení do více tříd program vždy vypisuje pouze číselné označení tříd.

7.8 Návrh úloh do počítačových laboratoří předmětu Umělá inteligence

Obsahem této kapitoly je popis pěti úloh navržených pro potřeby počítačových laboratoří předmětu MUIN. Úlohy byly vytvořeny tak, aby při jejich vypracování byly zřejmé principy a výhody RCE sítě. V příloze 1 jsou uvedeny XML soubory pro navržené úlohy. Veškerá data uvedená v XML souborech u jednotlivých úloh byly uměle vytvořeny.

7.8.1 Úloha č.1: Základní funkce RCE sítě

Úkoly:

1. Zkopírujte soubor "Data.xml" ze složky "1. Základní funkce RCE sítě".
2. Prohlédněte si průběh učení RCE sítě pro počáteční poloměry neuronů $R = 6$ a $R = 3$.
3. Rozhodněte, která hodnota poloměru je z hlediska učení sítě vhodnější.

Účel úlohy:

Cílem úlohy je seznámení se s průběhem učení RCE sítě. Body tréninkové množiny jsou vytvořeny tak, aby se při krokování učení sítě demonstrovaly veškeré zákonitosti uvedené v algoritmu učení, viz kapitola 3.4.

7.8.2 Úloha č.2: Prsteny

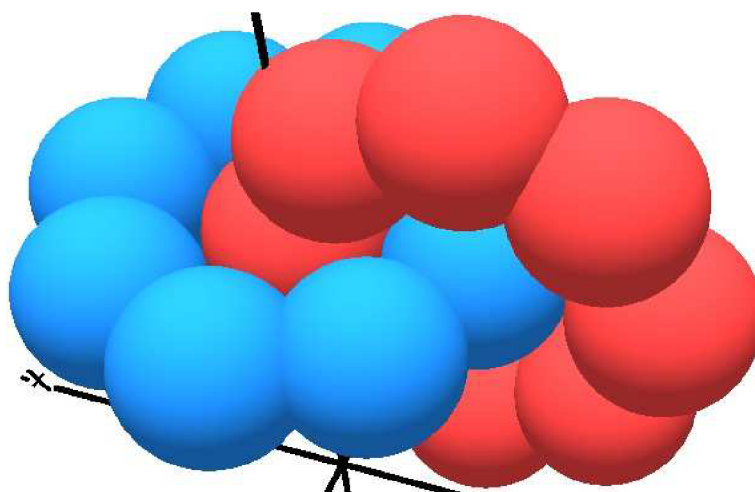
Úkoly:

1. Zkopírujte soubor "Data.xml" ze složky "2. Prsteny".
2. Nejprve si prohlédněte průběh učení RCE sítě v počáteční konfiguraci.
3. Rozhodněte, zda výsledný obrazec bude stále stejný pro různé hodnoty počátečních poloměrů. Svě domněnky si ověřte na vhodně zvolených hodnotách poloměrů. Během učení sledujte, jak nastavení poloměru ovlivňuje (pozitivně/negativně) průběh učení RCE sítě.
4. Rozhodněte, zda výsledný obrazec bude stále stejný pro různé pořadí bodů tréninkové množiny. Svě domněnky si ověřte na různých pořadí bodů, například při poloměru $R = 7$. Během učení sledujte, jak změna pořadí bodů ovlivňuje (pozitivně/negativně) průběh učení RCE sítě.

Účel úlohy:

Body tréninkové množiny tvoří dva prstény, viz Obr. 30. Tato úloha je specifická tím, že výsledná podoba naučené sítě je vždy stejná: nezáleží na pořadí bodů, ani na počáteční hodnotě poloměru. Studenti by si tyto skutečnosti měli ověřit a zároveň zjistit, že při nevhodném pořadí bodů či velikosti poloměru může dojít k nežádoucímu prodloužení učení sítě.

Úloha také demonstruje vlastnost popsanou v kapitole 3.3, že síť je schopna rozpoznat neoddělitelné oblasti.



Obr. 30: Naučená RCE síť v úloze č.2

7.8.3 Úloha č.3: Učení tří shluků

Zadání:

Body tréninkové množiny se skládají ze 3 shluků. Každý shluk obsahuje 30 vzorků. Vzorky ve shluku mají normální náhodné rozložení se směrodatnou odchylkou $\delta = 4$. Obdobné shluky jsou umístěny i v testovací množině pro kontrolu generalizace sítě. Kvůli velkému počtu kroků učení bude vhodnější prohlížet pouze poslední krok učení (pomocí klávesy End).

Tab. 5: Umístění středu shluků v úloze č.3

X	Y	Z	TŘÍDA
10	10	10	1 - ČERVENÁ
30	10	15	2 - ZELENÁ
18	0	8	3 - MODRÁ

Úkoly:

1. Zkopírujte soubor "Data.xml" ze složky "3. Učení tří shluků".
2. Sledujte vliv nastavení počátečního poloměru na počet neuronů skryté vrstvy a počet učících cyklů (tyto hodnoty program vypisuje v textovém okně).
3. Popište, jak počáteční poloměr ovlivňuje generalizaci sítě (jak RCE síť správně rozpoznává vzory testovací množiny). Procentuální úspěšnost RCE sítě na testovací množině program vypisuje v textovém okně.
4. Zjistěte, od které počáteční hodnoty poloměru se koncové nastavení poloměru neuronů nemění. Uvažujte pouze celá čísla.

Účel úlohy:

V této úloze se RCE síť učí tři shluky, ukázka naučené sítě je na Obr. 23. Nejprve je cílem sledovat průběh učení sítě (zejména počet neuronů ve skryté vrstvě) pro různé hodnoty počátečního poloměru.

V této jediné úloze je uvedena i testovací množina, takže dalším úkolem je sledování generalizace sítě pro různé hodnoty počátečního poloměru. Výstupem tohoto pozorování by měl být závěr: čím větší hodnota počátečního poloměru, tím lepší úspěšnost. Pro hodnoty poloměru větší jak deset nedochází ke značnému zlepšování úspěšnosti rozpoznání.

7.8.4 Úloha č.4: Učení jednoho shluku

Zadání:

V tomto úkolu budeme zkoumat úspěšnost naučení jednoho shluku. K tomu slouží funkce, která je volána pomocí kláves O(o). Pojem hlavní koule zastupuje kouli se středem stejným, jako je střed shluku, a poloměr určuje oblast, ve které zkoumáme úspěšnost naučení RCE sítě. Funkci je nutno zadat vstupní parametry: číslo klasifikované třídy a poloměr koule, ve které zkoumáme kvalitu naučení shluku.

Pozor: Na hodnotě poloměru shluku závisí časová náročnost funkce. Nejprve volte menší hodnoty poloměru.

Pozn.: Při výpočtu objemů je použit krok 0,25, to odpovídá přesnosti výpočtu 93,8%.

Úkoly:

1. Vytvořte množinu tréninkových bodů, tak aby tvořily shluk. Hodnotu poloměru a vzdálenost bodů volte tak, aby se při učení vytvořilo 4 až 5 skrytých neuronů.
2. Vypočítejte střed shluku u vytvořených tréninkových bodů.
3. Do souboru "Data.xml" zadejte vytvořené tréninkové body.
4. V programu spusťte funkci pro zkoumání úspěšnosti naučení shluku a zkontrolujte si správnost výpočtu středu shluku.
5. Sledujte úspěšnost naučení shluku při různém poloměru hlavní koule, ale i pro různé počáteční poloměry skrytých neuronů. Sledujte nejen

procentuální vyjádření úspěšnosti učení (tyto hodnoty program vypisuje v textovém okně), ale i grafické vyjádření zaplnění hlavní koule naučenou RCE sítí (barva hlavní koule je černá).

Účel úlohy:

Úloha slouží ke sledování generalizace sítě. Pomocí speciální funkce sledujeme, jakou část naučená síť zaujímá v oblasti výskytu shluku. Ukázka výstupu této úlohy je na Obr. 27.

V úloze se řeší pouze generalizace sítě při rozpoznávání jedné třídy, nemůže tedy docházet ke zmenšování poloměru u neuronů. Laicky bychom tedy řekli, že je nejlepší nastavit velkou hodnotu poloměru u skrytých neuronů, abychom stoprocentně rozpoznávali danou třídu. Problémem však je, že síť klasifikuje velké množství dat, které do třídy nepatří.

Správný přístup k problému je takový, že nás musí zajímat správné rozpoznání oblasti náležící třídě a zároveň musíme sledovat i velikost oblasti, která se nesprávně klasifikuje jako daná třída.

Tento problém se v praxi obtížně řeší, vyskytuje se například při klasifikaci kůže v prostoru určujícím barvu (např. RGB), viz [YGX01] a [SKR11].

7.8.5 Úloha č.5: Závislost počtu skrytých neuronů na počtu tréninkových bodů

Zadání:

Tréninkovou množinu tvoří jeden shluk. Vzorky ve shluku mají normální náhodné rozložení se směrodatnou odchylkou $\delta = 4$.

Tab. 6: Umístění středu shluku v úloze č.5

X	Y	Z	TŘÍDA
0	0	0	3 - MODRÁ

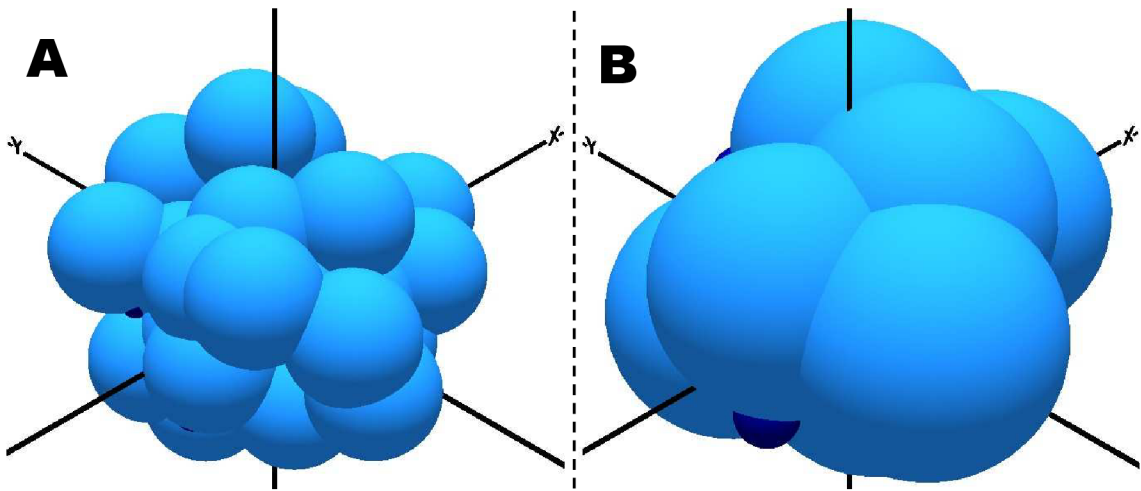
Úkoly:

1. Otevřete složku "5. Závislost počtu skrytých neuronů na počtu tréninkových bodů". V jednotlivých složkách se nachází soubor "Data.xml" pro tréninkové množiny s 30, 150, 300 body.
2. Pro konstantní počáteční poloměr $R = 5$ a $R = 10$ sledujte, jak se mění počet skrytých neuronů v závislosti na počtu bodů tréninkové množiny (30, 150, 300 bodů).

Účel úlohy:

Hlavním účelem úlohy je uvědomit si vztah mezi počtem neuronů a počtem bodů shluku v tréninkové množině: počet neuronů roste a konverguje k nějaké konečné hodnotě.

Počty neuronů jsou i samozřejmě závislé na počáteční hodnotě poloměru. Výstup úlohy je vidět na Obr. 31. Na Obr. 31A je RCE síť s poloměrem pět a pro 300 tréninkových bodů potřebovala 31 skrytých neuronů. Síť s poloměrem deset už potřebovala pouze 8 skrytých neuronů, Obr. 31B.



Obr. 31: Dvě naučené RCE sítě z úlohy č.5

8 ZÁVĚR

Vytvořil jsem program uTeachRCE (kapitola 5), jenž umožňuje naučit RCE síť vstupní data pro libovolné n z n -rozměrného prostoru, které jsou rozděleny do libovolného počtu klasifikovaných tříd.

V kapitole 6.1 jsou uvedeny tři množiny písmen (celkem 182 písmen), které jsem vytvořil pro aplikaci rozpoznávání písmen RCE sítí.

Pro testování rozpoznávání písmen jsem využíval program uTeachRCE. Nejprve jsem RCE síť naučil na množinu A (26 vzorů) a úspěšnost rozpoznání písmen množiny B RCE sítí byla 90,77% (kapitola 6.2). Při porovnání s vícevrstvou sítí s algoritmem učení backpropagation (stejné podmínky testování), která má nejlepší úspěšnost rozpoznání množiny B 99,23% [Mac11], je tato hodnota zřetelně horší. Z grafu na Obr. 15 v kapitole 6.2 je zřejmé, že RCE síť měla mnohem více problémů s rozpoznáním jednotlivých vzorů než síť backpropagation.

U všech špatně rozpoznávaných vzorů RCE sítí z množiny B jsem zjistil, že tyto vzory nebyly zařazeny do žádné z tříd (kapitola 6.2). Proto jsem provedl učení RCE sítě na rozšířenou tréninkovou množinu, která vznikla spojením množiny A a B (celkem 156 vzorů): úspěšnost rozpoznání písmen množiny C je 100% (kapitola 6.3).

Důvod, proč RCE síť nestačila a síť backpropagation stačila tréninková množina s 26 vzory, jsem vysvětlil pomocí Obr. 16 v kapitole 6.3. Spočívá to ve způsobu dělení vstupního prostoru: backpropagation rozděluje celý vstupní prostor na klasifikované oblasti (neexistuje neklasifikovaná oblast), RCE síť pokrývá vstupní prostor hyperkoulemi určujícími klasifikované oblasti (existuje neklasifikovaná oblast).

U všech stoprocentně nerozpoznaných vzorů množiny B (Obr. 15, kapitola 6.2) jsem zjistil, že právě klasifikované třídy, do kterých patří tyto vzory, potřebují více skrytých neuronů při učení RCE sítě na množinu A a B (Obr. 19, kapitola 6.3). U většiny klasifikovaných tříd, do kterých patří stoprocentně nerozpoznané vzory množiny B, jsem zjistil, že zaujímají velmi malé objemy ve vstupním prostoru (Obr. 20, kapitola 6.3).

Při učení RCE sítě na spojenou množinu A a B se už úspěšnost rozpoznání písmen množiny C vyrovnala úspěšnosti sítě backpropagation. Při měření časů jsem zjistil, že rychlost učení a vybavování RCE sítě je rychlejší než u sítě backpropagation (Tab. 2 a 3, kapitola 6.4). Učení RCE sítě na množinu A a B (156 vzorů) je téměř 80-krát rychlejší než učení třívrstvé sítě s algoritmem učení backpropagation na množinu A o 26 vzorech.

Při používání obou neuronových sítí na aplikaci rozpoznávání písmen jsem učinil závěr, že RCE síť je pro tuto aplikaci vhodnější (kapitola 6.4). A to z důvodů, že nastavení parametrů RCE sítě je jednodušší, učení je rychlejší, pro danou aplikaci se kladně projevila vlastnost RCE sítě aktivní paměť.

Vytvořil jsem program RCEin3D určený k vizualizaci učení RCE sítě ve 3D prostoru (kapitola 7). Program dokáže pracovat s tréninkovými/testovacími vzory se třemi příznaky, které jsou rozděleny do 120 tříd. Program RCEin3D umožňuje dopředné/zpětné krokování učení sítě, testování naučené sítě, výpis důležitých informací o naučené RCE síti. Programu RCEin3D jsem prezentoval na konferenci Student EEICT 2013 [Mac13].

Vizualizační program RCEin3D se ukázal, jako velmi vhodný nástroj k výukovým účelům: program RCEin3D se využívá při výuce v počítačových laboratořích předmětu Umělá inteligence, kde se RCE síť vyučuje. A proto jsem vytvořil pět výukových úloh pro program RCEin3D (kapitola 7.8), které by měly usnadnit pochopení učícího algoritmu sítě a co nejvíce studentům přiblížit základní vlastnosti sítě.

Literatura

- [Bíl95] BÍLÁ J. *Umělá inteligence a neuronové sítě v aplikacích*. 1. vyd. Praha: ČVUT, 1995. 115 s. ISBN 80-01-01275-1.
- [DHS00] DUDA R. O., HART P. E., STORK D. G. *Pattern classification*. 2. vyd. John Wiley & Sons, 2000. 680 s. ISBN 978-0-471-05669-0.
- [DSCUM93] DURG A., STOECKER W. V., COOKSON J. P., UMBROUGH S. E., MOSS R. H. Identifications of variegated coloring in skin tumors: neural networks vs. rule-based induction methods. In *Engineering in Medicine and Biology*. 1993, roč. 12, s. 71-74.
- [Fau94] FAUSETT L. *Fundamentals of neural networks: architectures, algorithms and applications*. 1. vyd. Prentice-Hall, 1994. 461 s. ISBN 0-13-334186-0.
- [Has95] HASSOUN M. H. *Fundamentals of artificial neural networks*. 1. vyd. Cambridge: MIT Press, 1995. 544 s. ISBN 0-262-08239-X.
- [HKPR08] HORÁK K., KÁLOVÁ I., PETYOVSKÝ P., RICHTER M. *Počítačové vidění*. Brno: Vysoké učení technické v Brně, 2008. 132 s. [cit. 2013-1-21]
- [HŠ92] HLAVÁČ V., ŠONKA M. *Počítačové vidění*. 1. vyd. Praha: Grada, 1992. 272 s. ISBN 80-85424-67-3. [cit. 2013-1-21]
- [HSIII96] HASEGAWA A., SHIBATA K., ITOH K., ICHIOKA Y., INAMURA K. An adaptive neural network: Application to character recognition on X-ray films. In *Neural Network*. 1996, roč. 9, s. 121-127.
- [Hu62] HU M. K. Visual pattern recognition by moment invariants. In *IRE transactions on information theory*. 1962, roč. 8, č.2, s. 179-187. [cit. 2013-1-21]
- [Jur03] JURA P. *Základy fuzzy logiky pro řízení a modelování*. 1. vyd. Brno: Vutium, 2003. 132 s. ISBN 80-214-2261-0.
- [Kot93] KOTEK Z. *Metody rozpoznávání a jejich aplikace*. 1. vyd. Praha: Academia, 1993. 195 s. ISBN 80-200-0297-9.
- [Lee89] LEE Y. *Classifiers; Adaptive modules in pattern recognition systems*. Cambridge: Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1989. 163 s.
- [Mac11] MACEČEK A. *Rychlost učení vícevrstvé sítě*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 58 s. Vedoucí bakalářské práce doc. Ing. Václav Jirsík. CSc.
- [Mac13] MACEČEK A. Visualization of learning process of artificial neural network RCE. In *Student EEICT*. 2013, roč. 19, s. 90-92.
- [MM04] MILITKÝ J., MELOUN M. Analýza vícerozměrných dat. V *Zajištění kvality analytických výsledků*. 2004, roč. 6, s. 19-28.

- [RCE82] REILY D. L., COOPER L. N., ELBAUM C. A neural model for category learning. In *Biological Cybernetics*. 1982, roč. 45, s. 35-41.
- [RCHF93] ROAN S., CHIANG CH., FU H. Fuzzy RCE neural network. In *Fuzzy Systems*. 1993, roč. 1, s. 629-634.
- [Scha97] SCHALKOFF R. J. *Artificial neural network*. 1. vyd. McGraw-Hill, 1997. 421 s. ISBN 0-07-115554-6.
- [SLM90] SHYU H. J., LIBERT J. M., MANN S. D. Classifying seismic signals via RCE neural network. In *Neural Network*. 1990, roč. 1, s. 101-105.
- [SKR11] SUI CH., KWOK N. M., REN T. A restricted coulomb energy (RCE) neural network system for hand image segmentation. In *Computer and Robot Vision*. 2011, s. 270-277.
- [SS90] SOBOTKA M., SOBOTKA Z. *Počítačová analýza a rozpoznávání obrazu*. 1. vyd. Praha: Dům techniky ČSVTS, 1990. 277 s. ISBN 80-02-00739-5.
- [ŠN96] ŠÍMA J., NERUDA R. *Teoretické otázky neuronových sítí*. 1. vyd. Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9. [cit. 2012-02-11]
- [Šno02] ŠNOREK M. *Neuronové sítě a neuropočítače*. 1. vyd. Praha: ČVUT, 2002. 156 s. ISBN 80-01-02549-7. [cit. 2012-02-18]
- [TX10] TAN CH., XIAO N. Improved RCE neural network ant its application in human-robot interaction based on hand gesture recognition. In *Information Science and Engineering*. 2010, roč. 2, s. 1260-1263.
- [WQ10] WANG R., QIAN X. *OpenScenGraph3.0: Beginner's guide*. 1. vyd. Birmingham: Packt Publishing Ltd, 2010. 412 s. ISBN 978-1-849512-82-4.
- [Xie03] XIE M. *Fundamentals of robotics: linking perception to action*. 1. vyd. World Scientific Publishing, 2003. 718 s. ISBN 981-238-313-1.
- [YGX01] YIN X., GUO D., XIE M. Hand image segmentation using color and RCE neural network. In *Robotics and Autonomous System*. 2001, roč. 34, s. 235-250.
- [YX02] YUAN M. L., XIE M. An incremental representation of conceptual symbols using RCE neural network. In *Development and Learning*. 2002, roč. 2, s. 102-107.
- [Zbo12] ZBOŘIL F. *Neuronové sítě RBF a RCE. Topologicky organizované neuronové sítě, soutěživé učení, Kohonenovy mapy*. Přednáška předmětu Soft computing vedená na Vysokém učení technickém v Brně na Fakultě informatiky. 2012.

Seznam symbolů

ξ	vnitřní potenciál neuronu
h	práh neuronu
\underline{c}	vektor určující střed hyperkoule $\underline{c} = (c_1, \dots, c_i, \dots, c_n)$
\underline{x}	vstupní vektor $\underline{x} = (x_1, \dots, x_i, \dots, x_n)$
\underline{y}	výstupní vektor $\underline{y} = (y_1, \dots, y_k, \dots, y_m)$
\underline{t}	vstupní tréninkový vektor $\underline{t} = (t_1, \dots, t_i, \dots, t_n)$
v	číselné označení klasifikované třídy, do které patří vstupní tréninkový vektor \underline{t}
X_i	i -tý neuron vstupní vrstvy, $i = 1, \dots, n$
Z_j	j -tý neuron skryté vrstvy, $j = 1, \dots, p$
Y_k	k -tý neuron výstupní vrstvy, $k = 1, \dots, m$
R_{MAX}	maximální hodnota poloměru hyperkoule
R_j	poloměr hyperkoule j -tého neuronu skryté vrstvy, $j = 1, \dots, p$
G_j	číselné označení klasifikované třídy, do které patří j -tý neuron skryté vrstvy, $j = 1, \dots, p$
w_{ij} (w_{jk})	váhová hodnota spoje mezi neurony X_i a Z_j (Z_j a Y_k)
C	parametr určující zda byl tréninkový vzor správně klasifikován
δ	směrodatná odchylka normálního náhodného rozložení
Γ	označení gama funkce
r	poloměr hyperkoule v 7-rozměrném prostoru
V	objem hyperkoule v 7-rozměrném prostoru
t_U	čas učení sítě na celou tréninkovou množinu
t_V	čas vybavování sítě jednoho testovacího vzoru
η	koeficient učení vícevrstvé sítě s algoritmem učení backpropagation
α	koeficient momentu vícevrstvé sítě s algoritmem učení backpropagation
m_{pq}	obecný moment
u_{pq}	centrální moment
ϑ_{pq}	normovaný centrální moment
$\varphi_1 - \varphi_7$	sedm momentových invariantů
S_i	i -tá souřadnice vektoru tréninkového/testovacího vzoru v programu uTeachRCE, $i = 1, \dots, n$
$STAT_X$	statistika X -tého testovacího vzoru vyjadřující počet chybných detekcí
Err_X	počet chybných detekcí X -tého testovacího vzoru
All_X	maximální počet chybných detekcí X -tého testovacího vzoru
b_{MAX}	počáteční hodnota šířky funkce příslušnosti u fuzzy RCE sítě
TH	práh u fuzzy RCE sítě

Seznam příloh

Příloha 1. CD obsahující:

- elektronickou verzi diplomové práce
- programy uTeachRCE a RCEin3D (Microsoft Visual Studio 2008)
- úlohy do počítačových laboratoří předmětu MUIN pro program RCEin3D
- vzory tvořící tréninkovou a dvě testovací množiny
- zpracované výsledky testování (Excel 2003)