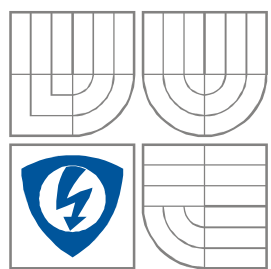


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

RYCHLOST UČENÍ VÍCEVRSTVÉ SÍTĚ

SPEED OF LEARNING MULTILAYER NETWORK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

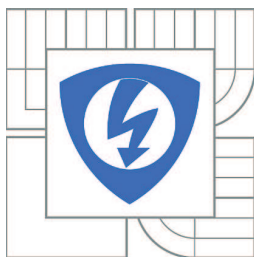
AUTOR PRÁCE
AUTHOR

ALEŠ MACEČEK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. VÁCLAV JIRSÍK, CSc.

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Aleš Maceček

ID: 115219

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Rychlost učení vícevrstvé sítě

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s problematikou popisu a rozpoznávání písmen pomocí neuronové sítě.
2. Seznamte se s vícevrstvou neuronovou sítí s algoritmem učení backpropagation.
3. Proved'te rozbor vlivu topologie a ostatních parametrů vícevrstvé neuronové sítě na její rychlosti učení a správnosti rozpoznávání písmen.
4. Dosažené výsledky zhodno'te.

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

Termín zadání: 7.2.2011

Termín odevzdání: 30.5.2011

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

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Teoretický rozbor umělých neuronových sítí, zvláště jejich typů topologií a učení sítí. Zvláštní zaměření je na vícevrstvou neuronovou síť s učením backpropagation. Uvedený algoritmus učení backpropagation jednoduché sítě společně s popisem parametrů ovlivňujících učení sítě a také metody zhodnocení kvality naučení sítě. Definice momentů invariantních na otočení, posun a změnu měřítko. Optimalizace parametrů neuronové sítě k nalezení nejrychleji učící se neuronové sítě, a také sítě s nejlepší hodnotou rozpoznání vzorů písmen z testovací množiny.

Klíčová slova

Neuron, neuronová síť, backpropagation, momentové invarianty, rozpoznávání, klasifikace vzorů

Abstract

Theoretical study about neural networks, especially their types of topologies and networks learning. Special attention is attended to multilayer neural network with learning backpropagation. Introduced learning algorithm backpropagation of simple networks in conjunction with descriptions of parameters affecting network learning also methods to evaluation quality of network learning. Definition moment invariants to rotation, translation and scaling. Optimization parameters of neural networks to find the network which has the fastest learning and also the networks with the best value of recognition patterns of letters from testing set.

Keywords

Neuron, neural network, backpropagation, moment invariants, recognition, pattern classifications

Bibliografická citace:

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..

Prohlášení

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

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **25. května 2011**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské 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 bakalářské práce.

V Brně dne: **25. května 2011**

.....
podpis autora

Obsah

1	Úvod	8
2	Umělé neuronové sítě	9
2.1	Biologický neuron	9
2.2	Formální neuron	10
2.3	Topologie neuronové sítě	11
2.3.1	Vícevrstvá neuronová síť	12
2.4	Učení neuronové sítě	13
2.4.1	Tréninková a testovací množina	14
2.4.2	Přeučení neuronové sítě	14
2.5	Typy neuronových sítí	15
2.6	Neuronová síť určená pro rozpoznávání vzorů	16
3	Vícevrstvá Neuronová síť s typem učení backpropagation	18
3.1	Algoritmus učení backpropagation	18
3.1.1	Chybová funkce	21
3.1.2	Koeficient učení	22
4	Popis objektů	24
4.1	Příznakový vektor	24
4.2	Momentový popis	25
4.2.1	Normalizace momentů	27
5	Popis funkce vytvořených programů	28
6	Optimalizace parametrů neuronové sítě z hlediska rychlosti učení	30
6.1	Použité vzory písmen	30
6.2	Testování vlivu počátečních vah na rychlost učení	31
6.3	Testování vlivu topologie sítě na rychlost učení	33
6.4	Testování vlivu koeficientu učení na rychlost učení	34
6.5	Testování vlivu koeficientu momentu na rychlost učení	36
6.6	Testování vlivu strmosti přenosové funkce na rychlost učení	38
6.7	Testování vlivu topologie sítě na rychlost učení s nejvhodnějšími parametry	40
6.8	Zhodnocení testů	41
7	Rozpoznávání písmen neuronovou sítí	42
7.1	Výsledky testů rozpoznávání písmen	42
7.1.1	Závislost rozpoznání písmen na velikosti parametrů sítě	46
7.2	Závislost rozpoznání písmen na hodnotě celkové chyby sítě	48

8	Rešerše prací zabývající se rozpoznáváním znaků neuronovou sítí.....	50
9	Závěr.....	52
10	Přílohy	58

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 mnoha dalších oborech, proto je jim věnována stále větší pozornost. Umělá neuronová síť je složena z mnoha vzájemně propojených procesorů, které nazýváme umělé neurony, protože velmi zjednodušeně modelují biologické neurony. Při vytváření umělých neuronových sítí se nesnažíme o vytvoření identické kopie mozku, jde nám jen o inspirování se některými jeho základními vlastnostmi.

První část práce spočívá v rozboru problematiky umělých neuronových sítí, zvláštní pozornost je věnována vícevrstvé neuronové síti s typem učení backpropagation. V práci byla snaha k odbornému výkladu termínů vkládat souvislosti a příklady názorného vztahu teorie umělých neuronových sítí k situacím z běžného lidského života. Praktická část práce se zabývá rozбором vlivu parametrů neuronové sítě na rychlost učení a kvalitu rozpoznání písmen testovací množiny.

Druhá kapitola se zabývá popisem umělých neuronových sítí a jejich základním stavebním prvkem, umělým neuronem. Je zde uveden i popis modelu biologického neuronu, který slouží především k uvědomění si analogických souvislostí mezi biologickým a umělým neuronem (nejde o úplný popis biologického neuronu). Dále je v kapitole uveden matematický popis formálního neuronu, teoretická definice i s příklady topologií a učení umělých neuronových sítí.

Umělá neuronová síť s učením backpropagation je uvedena ve třetí kapitole. Je zde uveden algoritmus učení backpropagation. Také je v kapitole popsána chybová funkce a parametry ovlivňující rychlost učení.

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

Pátá kapitola stručně popisuje programy, které byly vytvořeny pro řešení bakalářské práce.

V šesté kapitole je uveden postup optimalizace parametrů neuronové sítě s cílem nalezení nejrychleji učící se sítě.

Sedmá kapitola obsahuje popis testování sítí na rozpoznávání písmen tvořící testovací množinu. Také je zde uveden test, při kterém se zjišťuje vliv celkové chyby sítě na kvalitu rozpoznávání písmen.

V Osmé kapitole je uvedena řešerše problematiky rozpoznávání znaků neuronovou sítí. V rámci kapitoly je uveden stručný popis jiných přístupů k řešení tohoto problému.

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 rozdělena na tři dílčí dynamiky: 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.), analýza (EKG, řeči aj.) a v mnoha dalších oborech.

Hlubší výklad problematiky umělých neuronových sítí je uveden v literatuře [1] až [7] a také v anglicky psané literatuře [12], [13] a [14]. Stručnější přehled je uveden ve třech částech seriálu věnovaném umělým neuronovým sítím, viz literatura [15] až [17].

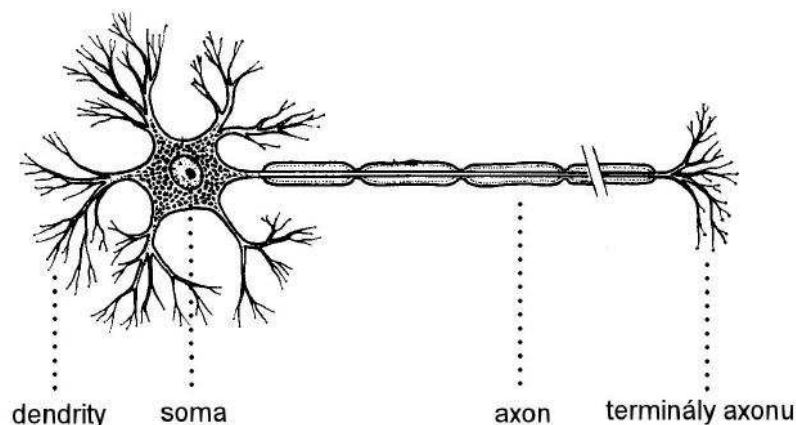
2.1 Biologický neuron

Neuron je samostatná specializovaná jednotka sloužící ke sbírání, zpracování, uchování a přenosu informací. Neuron se rozděluje na tři základních částí (soma, dendrity, axon), jak je vidět na obrázku 2.1. Na tělo neuronu (soma) jsou připojeny tisíce vstupních přenosových kanálů (dendrity) a jeden výstupní přenosový kanál (axon). Dendrity jsou krátké výběžky, které se rozdělují na větévky, tzv. trny, začínající vnímavým zakončením. Axon, tzv. neurit, je dlouhý výběžek, z jehož konce vychází několik větví, tzv. terminálů, zakončených blanou. Trny slouží k přenášení signálů od ostatních neuronů (pomocí terminálů) a senzorických buněk do těla neuronu.

Přenos informací mezi neurony se děje pomocí unikátního mezineuronového rozhraní, tzv. synapse. Dle [1] je míra synaptické propustnosti nositelem všech významných informací během celého života organismu. Dle šíření vzruchu v nervové soustavě se synapse dělí na excitační (šíří vzruch) a inhibiční (tlumí vzruch).

Informace se šíří díky tomu, že axon i tělo neuronu jsou obaleny membránou, jejíž schopností je za určitých okolností generovat elektrický impuls. Impulsy se přenášejí z axonu na dendrity jiného neuronu pomocí synaptických bran, jejich propustnost ovlivňuje intenzitu podráždění dalších neuronů. Podrážděné neurony samy generují elektrický impuls (zajišťují šíření informace) teprve, až jsou aktivovány. Neuron je aktivován, když hodnota budících vstupních signálů je větší než hodnota

tlumících vstupních signálů o určitou hraniční hodnotu, tzv. práh. Předpokladem paměťové schopnosti neuronů je, že synaptická propustnost vah se mění při každém průchodu signálu.

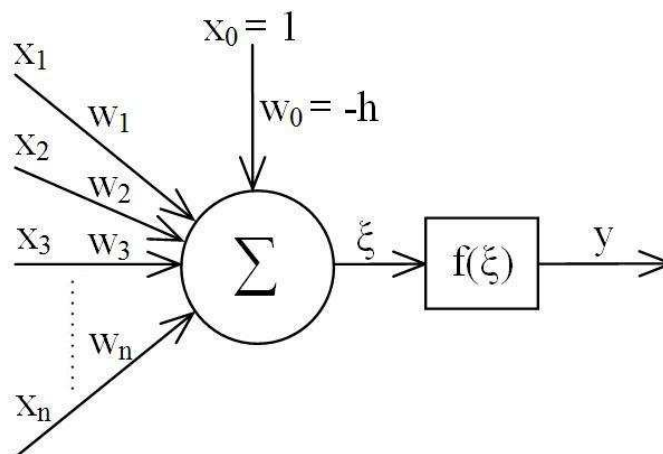


Obr. 2.1 : Biologický model neuronu [1]

Vzájemné propojení neuronů během života organismu není stálé. Mohou se vytvářet nová spojení během učení, anebo se spoje přerušují při zapomínání.

2.2 Formální neuron

Formální neuron, dále jen neuron, je matematický model, který je principiálně stejný jako biologický neuron. Má konečný počet vstupů $x_1, \dots, x_n$, které zastupují dendrity, a jeden výstup y reprezentující axon. Váhy jednotlivých vstupů $w_1, \dots, w_n$ určují citlivost, s jakou vstupy působí na výstup. Bereme-li v úvahu, že vstupy jsou výstupy jiných neuronů, tak váhy napodobují propustnost synaptických bran biologického neuronu. Váhy mohou být i záporné, tím se vyjádří jejich inhibiční charakter. Struktura formálního neuronu je schematicky znázorněna na obrázku 2.2.



Obr. 2.2 : 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 (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.

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

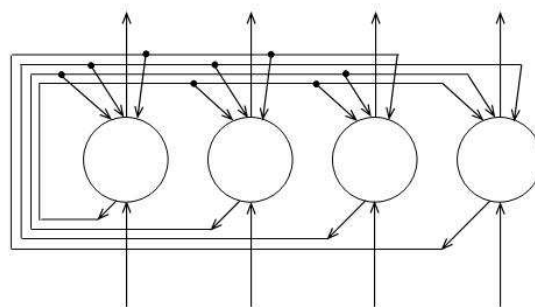
Přenosová (aktivační) funkce slouží k převádění vnitřního potenciálu neuronu do definovaného oboru výstupních hodnot. Přenosová funkce může být lineární nebo nelineární. Nelineární funkce jsou nejvíce používány: ostrá nelinearita, standardní (logická) sigmoida a hyperbolický tangens. Neuron s prahovou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

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

2.3 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 obrázku 2.3, jedná se o Hopfieldovu síť.



Obr. 2.3 : Příklad cyklické topologie (Hopfieldova síť)

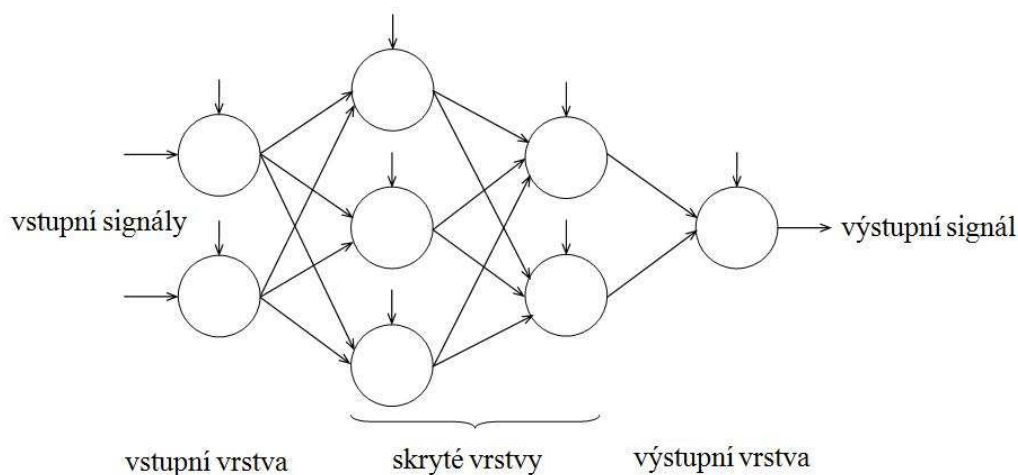
Acyklická topologie se vyznačuje tím, že se informace v síti šíří pouze jedním směrem. Neurony jsme schopni rozdělit do vrstev (např. nad sebou) a vazby mezi neurony vedou pouze z nižších vrstev do vyšších tak, že mohou přeskakovat libovolný počet vrstev. Do acyklické topologie patří neuronová vícevrstvá síť, kterou se budeme zabývat v podkapitole 2.3.1.

2.3.1 Vícevrstvá neuronová síť

U této neuronové sítě rozlišujeme tři typy vrstev: vstupní, skrytá a výstupní. Vrstvy značíme číslicemi, nula odpovídá vstupní vrstvě, pak následuje libovolný počet skrytých vrstev a poslední vrstva je výstupní. Jednotlivé neurony v sousedících vrstvách jsou mezi sebou propojeny tak, že tvoří úplný bipartitní graf [3], tj. výstup neuronu dané vrstvy je vstupem všem neuronům následující vrstvy (nelze přeskakovat vrstvy). Pokud vazba mezi neurony chybí, tak ji chápeme, jako vazbu s nulovou vahou. Díky úplnému propojení mezi neurony můžeme topologii sítě popsat pouze počty neuronů v jednotlivých vrstvách oddělených pomlčkou směrem od vstupní k výstupní vrstvě. Vstupní vrstva není zahrnuta do počtu vrstev sítě. V celé práci bude tedy pod pojmem dvouvrstvá neuronová síť myšlena síť, která má tyto vrstvy: vstupní, jednu skrytou a výstupní vrstvu. Na obrázku 2.4 je ukázaná topologie třívrstvé sítě 2-3-2-1.

Vstupní vrstva slouží ke vstupu určitého signálu z okolí. Tento signál se pouze distribuuje na všechny neurony první skryté vrstvy (použita lineární přenosová funkce), anebo před distribucí na skryté neurony ještě dochází k normalizaci vstupního signálu (nejčastěji se používá přenosová funkce ostrá nelinearita), tento případ je zobrazen v obrázku 2.4.

Neurony ve skrytých vrstvách transformují vstupy z předchozí vrstvy do následující, jejich hlavním úkolem je zvýšení aproximačních vlastností neuronové sítě jako celku. Výstupní vrstva předává do okolí výstupní signály, které jsou odezvou neuronové sítě na přivedené vstupní signály.



Obr. 2.4 : Vícevrstvá neuronová síť s topologií 2-3-2-1

Množství neuronů ve vstupní a výstupní vrstvě vychází z typu řešeného problému (např. logická funkce AND má dva vstupy a jeden výstup). Počet skrytých vrstev a neuronů v jednotlivých vrstvách nelze jednoznačně určit, jejich počet je závislý na složitosti řešeného problému. Nejčastěji se počet určuje experimentálně. V některých případech je možné použít doporučení týkající se počtu skrytých neuronů. Dle [12] by neměl celkový počet vstupních signálů N být větší než tzv. stupeň volnosti, který zastupuje počet všech vah a prahových hodnot (prahové hodnoty vstupní vrstvy se nepočítají) v celé neuronové síti. Z hlediska učení neuronové sítě je nejvhodnější, pokud stupeň volnosti je menší než $N/10$.

2.4 Učení neuronové sítě

Před začátkem učení jsou váhy nastaveny na počáteční hodnoty, které bývají náhodně zvolené 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:

- Při učení s učitelem je známý požadovaný výstup 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. Celý proces se opakuje pro další tréninkový vstup. Jako příklad učení s učitelem je učení backpropagation, RBF.
- Při učení bez učitele není žádná možnost zjištění správnosti výstupu sítě. Algoritmus učení je navržen tak, že hledá ve vstupních datech určité vzorky se společnými vlastnostmi [6]. Učení bez učitele je samoorganizací. Pro příklad jsou to síť Kohonenova, ART a Hammingova.

2.4.1 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).

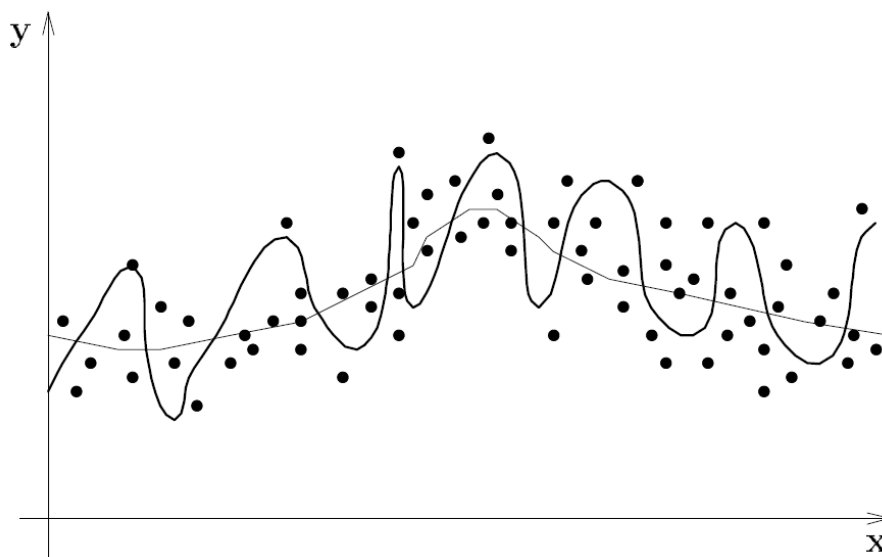
Rychlost a úspěšnost učení lze také ovlivnit strategií výběru vzorů z tréninkové množiny (sekvenční, náhodné, některé vzory se mohou vybírat častěji). Cyklus je časový interval, ve kterém jsou alespoň jednou vybrány všechny vzory z tréninkové množiny. K naučení sítě potřebujeme stovky až tisíce cyklů.

2.4.2 Přeučení neuronové sítě

Problém přeučení (overfitting) neuronové sítě je silně spojen s určením vhodné topologií sítě, konkrétně s počtem neuronů ve skrytých vrstvách. Pro rozebrání tohoto problému je vhodné zavést pojem generalizace neuronové sítě. Generalizace (zobecnění) popisuje, jak naučená síť rozeznává jevy, které nebyly součástí učení, ale které je možno nějakým způsobem odvodit z naučeného.

Pokud použijeme příliš malou síť, tak nedokáže správně generalizovat. Nebyla by schopna zkonstruovat plochu procházející všemi body tréninkové množiny (označováno jako interpolace vícerozměrné plochy), respektive plochu procházející „blízko“ těchto bodů (aproximace vícerozměrné plochy). Naopak velká síť by se naučila tréninkové vzory včetně jejich nepřesností a chyb (nedošlo by k zobecnění zákonitostí mezi vstupy a výstupy), pro nenaučené vzory by dávala chybné výsledky. Tento jev je označován za přeučení neuronové sítě. Pro snadnější pochopení problému budeme uvažovat příliš velkou síť, která by naprosto správně odpovídala pouze na tréninkové vzory. Odpovídá to situaci, kdy se student nazpaměť naučí pár příkladů a na jiné příklady nedokáže určit správnou odpověď.

Na obrázku 2.5 jsou graficky znázorněny funkce dvou sítí společně s tréninkovými a testovacími vzory (body). Silná čára reprezentuje přeučenou síť, která se přizpůsobila tréninkovým vzorům. Síť, která „správně“ generalizovala zákonitost v tréninkové množině, je zobrazena tenkou čarou.



Obr. 2.5 : Graf funkce přeučené sítě (tučně) a sítě se „správnou“ generalizací [1]

2.5 Typy neuronových sítí

Existuje velké množství typů neuronových sítí. Nyní si uvedeme několik základních typů neuronových sítí společně s krátkou definicí. Širší popis těchto neuronových sítí je uveden v literatuře [1], [16] a [21].

Vícevrstvá neuronová síť s učícím algoritmem backpropagation bude podrobněji rozebrána v třetí kapitole. Topologie této neuronové sítě byla již popsána v kapitole 2.3.1.

Kohonenova síť využívá učení bez učitele, tedy shlukové analýzy (algoritmus ve vstupních vzorech hledá určité vlastnosti a závislosti bez znalostí nějaké vnější informace). Počet vstupů je roven dimenzi vstupního prostoru (nejčastěji dvoudimenziální), a každý vstup je spojen s každým neuronem mřížky. Počet výstupu je daný počtem neuronů, protože každý neuron v mřížce je přímo výstupem. Učení probíhá tak, že algoritmus se snaží uspořádat neurony v mřížce do určitých oblastí, aby bylo možné rozpoznat třídy vzorů předkládaných sítí.

RBF (Radial Basic Function) síť je chováním i strukturou podobná vícevrstvé neuronové sítí. Je možné si ji představit jako dvouvrstvou neuronovou síť, kde vstupní vrstva slouží pouze k distribuci. Skrytá vrstva se skládá z RBF jednotek (středově symetrická převodní charakteristika) realizující jednotlivé radiální funkce. Výstupní vrstva je lineární. Váhy ze vstupní do skryté vrstvy jsou pevně nastaveny (rovnoměrné rozložení, náhodné vzorky, samoorganizace). Váhy ze skryté do výstupní vrstvy se během učení mění, tím se určuje dělicí nadrovina. Velkou výhodou sítě je schopnost rychlého učení.

Hopfieldova síť má tolik neuronů, kolik má vstupů. Každý neuron je spojen se vstupem sítě pouze jedním svým vstupem. Výstup každého neuronu je připojen na vstupy všech ostatních neuronů. Probíhá jednorázový proces nastavení vah, síť je tedy bez učení. Veškeré signály v síti mají přípustné hodnoty binární nebo bipolární. Vybavování je iterační proces, který probíhá tak dlouho, dokud se výstupy všech neuronů během dvou cyklů nezmění.

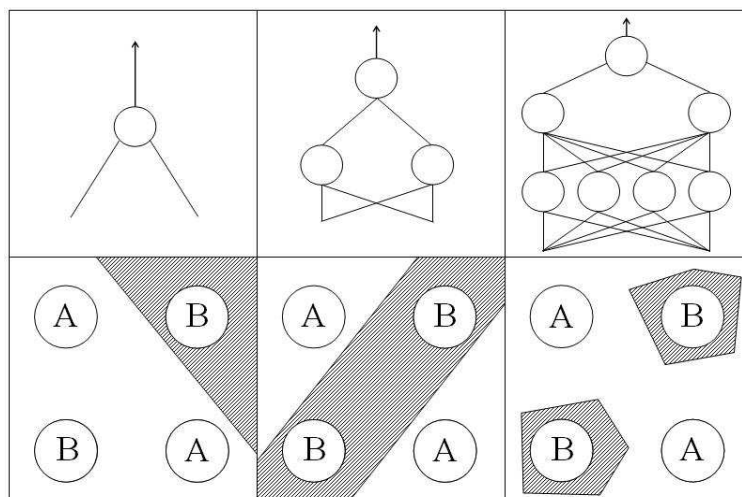
ART síť je schopna řešit problém proměnné stability. Proměnná stabilita reprezentuje neschopnost sítě naučit se novou informaci bez poškození již dříve naučené informace. Dle [16] je síť schopna přepínat mezi tzv. tvárným módem (učící stav, parametry sítě se mohou měnit) a tzv. stabilním módem (parametry pevně dány, síť vykonává zadanou funkci), aniž by došlo k poškození již naučené informace. Síť je složena ze dvou vrstev: vstupní (porovnávací) a výstupní (rozpoznávací). Síť je plně propojená, každý neuron s každým v obou směrech. Z toho vyplývá, že používá dva typy vah: dopředné a zpětné. Učení spočívá v adaptaci váhových spojení při každé výměně vstupního vzoru mezi vstupní (adaptace dopředných vah) a výstupní (adaptace zpětných vah) vrstvou, dokud není nalezený ustálený stav.

Hammingova síť se skládá ze dvou částí. Dolní část má vstupní vrstvu, počet neuronů závisí na počtu vstupních signálů, a výstupní vrstvu, jejíž velikost je dána počtem rozpoznávaných tříd. Výstupy dolní části slouží jako vstupy horní části, která se nazývá MAXNET a její zapojení je obdobné jako v Hopfieldově síti. Učení spočívá v jednorázovém nastavení váhových spojení, kdežto vybavování je iterační proces. Dle [21] lze tuto síť popsat jako třídič podle nejmenší chyby: k danému vstupu najde kategorii, jejíž reprezentant má od vstupů nejmenší tzv. Hammingovu vzdálenost, tj. počet odlišných vstupů.

2.6 Neuronová síť určená pro rozpoznávání vzorů

Neuronovou síť můžeme použít k rozeznávání (klasifikaci) jednotlivých skupin vzorů. Jednou z možností je použití vícevrstvé neuronové sítě a učení s učitelem. Neuronová síť použitá jako klasifikátor má za vstupní hodnoty reálná čísla (např. souřadnice v prostoru) a výstupní hodnoty ideálně nabývají dvou stavů (0, 1), ve skutečnosti nabývají hodnot v intervalu od 0 do 1. Ideální případ rozpoznání: všechny výstupní hodnoty jsou nulové a jeden z výstupů se rovná jedné, ten určuje třídu, do které patří vstupní vzor. V praxi se výstupní hodnoty blíží hodnotám nula nebo jedna, největší výstupní hodnota určuje třídu vstupního vzoru.

Každý neuron počítá vážený součet vstupů, který odpovídá rovnici nadroviny. Dimenze prostoru je určena počtem vstupů neuronu (dva vstupy reprezentují rovnici přímky, tři vstupy pro rovnici roviny atd.). Příklady topologií společně s řešenými klasifikacemi jsou na obrázku 2.6.



Obr. 2.6: Příklady topologií vícevrstvé neuronové sítě a jimi řešené úlohy [3]

Neurony první vrstvy určují počet nadrovin, které můžeme použít pro oddělení skupin v tréninkových datech. Následující vrstva, jež počítá kombinaci výstupu předchozí vrstvy, je schopna kombinovat nadroviny z předchozí vrstvy do konvexních tvarů (otevřených nebo uzavřených). Další vrstva kombinuje konvexní útvary do složitých nekonvexních útvarů, ty se mohou překrývat (částečně, úplně), protínat nebo jsou úplně oddělené. Tímto způsobem se dojde až k výstupní vrstvě.

V případě, kdy máme již naučenou síť (např. pro rozpoznávání číslic) a potřebovali bychom ještě přidat dodatečnou výstupní informaci (např. rozhodnutí o sudosti vstupního čísla). Přidáme jeden výstupní neuron, který se propojí se všemi neurony poslední skryté vrstvy, to vychází z topologie vícevrstvé neuronové sítě. Tréninková množina a váhy zůstávají neměnné, mění se pouze váhy příslušející přidanému výstupnímu neuronu. Tento princip ovlivní potřebný čas k naučení sítě, protože naučenou síť pouze doučíme přidané rozšíření, během doučování používáme vstupní vzory z tréninkové množiny použité k předešlému naučení neuronové sítě. Pro složitější rozšiřující úkoly bychom museli pozměnit tréninkovou množinu a provést opětovné učení neuronové sítě se změněnou topologií.

Aplikace Kolmogorova teorému: pro vyřešení libovolného problému spojeného s klasifikací vzorů, nebudeme nikdy potřebovat více jak dvouvrstvou neuronovou síť s vhodným počtem neuronů v jednotlivých vrstvách. Ve skutečnosti se však setkáváme s více jak dvouvrstvěmi sítěmi. Je to dáno tím, že síť navrhujeme na řešený problém a nesnažíme se o navržení dokonale optimální sítě. Avšak stále platí Kolmogorův teorém: navržená vícevrstvá síť může být převedena do dvouvrstvé sítě.

3 VÍCEVRSTVÁ NEURONOVÁ SÍŤ S TYPEM UČENÍ BACKPROPAGATION

Pojmenování typu učení je odvezeno od učícího algoritmu error backpropagation (zpětné šíření chyby), častěji se uvádí zkrácený název backpropagation. Během vykonávání algoritmu se chyba výstupní vrstvy zpětně přepočítává do předchozích vrstev (zpětně se šíří), kde se podle její velikosti upravují váhy spojů. Algoritmus bude podrobněji popsán v podkapitole 3.1.

Práce uvedená v literatuře [25] s názvem „Learning internal representations by error propagation“ od autorů Rumelhart D. E., Hinton G. E., Williams R. J. obsahuje první popis algoritmu učení backpropagation. Tato práce byla uvedena v roce 1986 a od té doby je algoritmus učení backpropagation nejpoužívanější učící metodou neuronových sítí.

3.1 Algoritmus učení backpropagation

Samotný algoritmus můžeme rozdělit do tří částí:

- dopředné šíření vstupního signálu z tréninkové množiny,
- zpětné šíření chyby,
- adaptace vah a prahů.

Pro učení vícevrstvé neuronové sítě s algoritmem učení backpropagation je možno použít pouze přenosové funkce splňující tato kritéria: musí být diferencovatelná, monotónně klesající a spojitá. Nejvíce používanou funkcí je standardní sigmoida (3.1) nebo hyperbolický tangens (3.2), ve vzorcích se vyskytuje strmost funkce značená písmenem S .

$$f(\xi) = \frac{1}{1 + e^{-S\xi}} \quad (3.1)$$

$$f(\xi) = \frac{1 - e^{-S\xi}}{1 + e^{-S\xi}} \quad (3.2)$$

Algoritmus učení backpropagation dvouvrstvé sítě [5]:

- Krok 1: Nastavit váhové hodnoty a prahy na libovolně malé hodnoty z rozsahu $<-1,1>$. Nastavit koeficient učení, problematika nastavení bude popsána v podkapitole 3.1.2. Nastavit celkovou chybu sítě E_C na nulu.
- Krok 2: Opakovat kroky 3 až 11 dokud není splněna podmínka ukončení algoritmu v kroku 11.
- Krok 3: Pro každou dvojici $\underline{s}; \underline{t}$ (požadované výstupy neuronové sítě $\underline{t}$ při vstupním tréninkovém vektoru $\underline{s}$) provádět kroky 4 až 10. Pokud proběhl cyklus (všechny tréninkové vzory byly alespoň jednou vybrány), tak nastavit $E_C = 0$.
- Krok 4: Aktivovat neurony vstupní vrstvy ($X_i, i = 1, \dots, n$) tak, že $\underline{x} = \underline{s}$. Vstupní vrstva slouží pouze pro distribuci vstupních signálů do první skryté vrstvy.
- Krok 5: Vypočítat vnitřní potenciál (3.3) neuronů ve skryté vrstvě ($Z_j, j = 1, \dots, p$) a následně určit jejich výstupní hodnotu (3.4) pomocí přenosové funkce.

$$z_j(in) = w_{0j} + \sum_{i=1}^n x_i w_{ij} \quad (3.3)$$

$$z_j = f[z_j(in)] \quad (3.4)$$

- Krok 6: Vypočítat vnitřní potenciál (3.5) neuronů ve výstupní vrstvě ($Y_k, k = 1, \dots, m$) a následně určit jejich výstupní hodnotu (3.6) pomocí přenosové funkce, tj. výstup celé neuronové sítě.

$$y_k(in) = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (3.5)$$

$$y_k = f[y_k(in)] \quad (3.6)$$

- Krok 7: Každý neuron ve výstupní vrstvě ($Y_k, k = 1, \dots, m$) má přiřazenou požadovanou hodnotu výstupu v závislosti na vstupním tréninkovém vzoru. Vypočítat částečné váhové korekce příslušející chybě spojení mezi neurony skryté a výstupní vrstvy (3.7), poté určit korekci vah (3.8) a prahů (3.9).

$$\delta_k = (t_k - y_k) f'[y_k(in)] \quad (3.7)$$

$$\Delta w_{jk} = \eta \delta_k z_j \quad (3.8)$$

$$\Delta w_{0k} = \eta \delta_k \quad (3.9)$$

Krok 8: Každý neuron ve skryté vrstvě (Z_j , $j = 1, \dots, p$) má přiřazenou hodnotu sumace jeho delta vstupů (3.10), tzn. z neuronů v následující vrstvě (v našem demonstračním případě je to vrstva výstupní). Vypočítat částečné váhové korekce příslušející chybě spojení mezi neurony vstupní a skryté vrstvy (3.11), poté určit korekci vah (3.12) a prahů (3.13).

$$\delta_j(in) = \sum_{k=1}^m \delta_k w_{jk} \quad (3.10)$$

$$\delta_j = \delta_j(in) f'[z_j(in)] \quad (3.11)$$

$$\Delta w_{ij} = \eta \delta_j x_i \quad (3.12)$$

$$\Delta w_{0j} = \eta \delta_j \quad (3.13)$$

Krok 9: Vypočítat chybu sítě vztaženou k r -tému tréninkovému vzoru dle (3.14) a poté vypočítat celkovou chybu sítě (3.15), kde v značí celkový počet tréninkových vzorů.

$$E_r = \frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \quad (3.14)$$

$$E_C = \sum_{r=1}^v E_r \quad (3.15)$$

Krok 10: Aktualizovat váhové hodnoty i prahy ($j = 0, \dots, p$; $i = 0, \dots, n$) u všech neuronů výstupní vrstvy (Y_k , $k = 1, \dots, m$) dle (3.16) a skryté vrstvy (Z_j , $j = 1, \dots, p$) dle (3.17).

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (3.16)$$

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij} \quad (3.17)$$

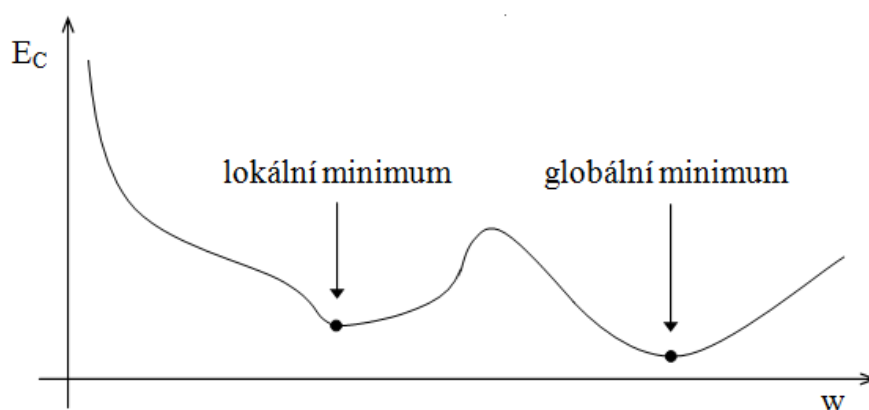
Krok 11: Ukončení algoritmu. Porovnáváme celkovou chybu E_C s požadovanou celkovou chybou, tímto způsobem lze kvalitu naučení posoudit. Avšak pokud požadujeme velmi malé hodnoty celkové chyby sítě, tak počet cyklů potřebný k jejímu naučení bude neúnosný. Za ukončovací podmínku je možné brát provedený počet cyklů, v tomto případě však nelze posoudit kvalitu naučení. Vhodným kompromisem je sledování chyby sítě během jednotlivých cyklů.

Ukázanou konstrukci algoritmu nelze považovat za striktně danou. Algoritmy se od sebe mohou lišit pro jednotlivé případy. Například: nebude se měnit hodnota váhového spojení prahů a neuronů, bude proměnný počet neuronů, koeficient učení nebude konstantní během celého algoritmu.

Pro lepší představu dopředného šíření vstupního signálu a zpětného šíření chyby může pomoci článek [22], kde je algoritmus v jednotlivých krocích popsán matematickými rovnicemi společně s grafickým zobrazením šíření vstupního signálu a chyby.

3.1.1 Chybová funkce

Chybová funkce znázorňuje závislost celkové chyby sítě na počtu cyklů. Chybová funkce se používá ke zhodnocení kvality naučení, a s tím souvisí hledání globálního minima v chybové funkci (chceme dosáhnout nejmenší hodnoty). Dle [5] je cílem učení minimalizace chyby funkce ve váhovém prostoru, pro řešení tohoto problému se používá nejjednodušší varianta gradientní metody, která vyžaduje diferencovatelnost chybové funkce. Pro snadnější vysvětlení problému hledání globálního minima s pomocí gradientní metody si ukážeme příklad chybové funkce na obrázku 3.1.



Obr. 3.1 : Průběh chybové funkce

Největším problémem gradientní metody je nalezení lokálního minima, které není minimem globálním. Gradientní metoda se v tomto lokálním minimu zastaví (nulový gradient) a celková chyba se již nebude zmenšovat. Výskyt této situace není ojedinělý, a proto byla nalezena řada způsobů, jak tento problém vyřešit:

- Změna koeficientu učení, vliv koeficientu učení na hledání globálního minima je uveden v podkapitole 3.1.2.
- Přidání šumu do rovnic pro výpočet změny vah. Tím dosáhneme změny pohybu po chybové funkci ve směru gradientu, změna spočívá v tom, že přechodně můžeme dosáhnout na místa s vyšší hodnotou chyby sítě. Díky tomu

se v některých případech podaří dostat z lokálního minima. Tento způsob řešení je jednoduchý a časově málo zatížitelný.

Elektrické šoky používané v psychiatrických léčebnách lze považovat za napodobení této popsané metody adaptace neuronových sítí.

- Nejčastější modifikací je přidáním momentu α (setrvačnost) do rovnic pro korekci vah a prahů. Hodnota momentu se volí z intervalu $\langle 0,1 \rangle$, nejčastěji blízko jedné. Moment nám říká, jak dlouho se držíme daného směru, než zabočíme podle nové změny gradientu. Jinými slovy, lokální minimum můžeme překročit o jeden krok, který je daný velikostí předchozího kroku upravený velikostí momentu. Přidáním momentu do rovnic pro korekci vah (3.12) a prahu (3.13) dostaneme modifikované rovnice (3.18) a (3.19).

$$\Delta w_{ij}(new) = \eta \delta_j x_i + \alpha \Delta w_{ij}(old) \quad (3.18)$$

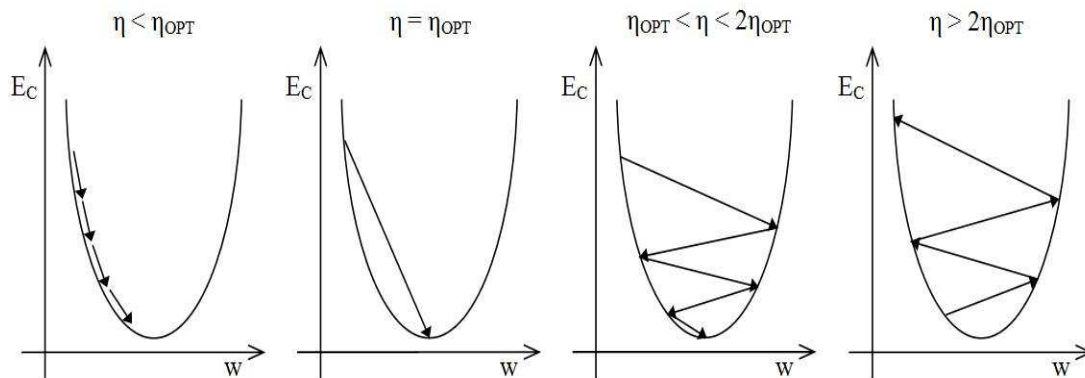
$$\Delta w_{0j}(new) = \eta \delta_j + \alpha \Delta w_{0j}(old) \quad (3.19)$$

Přidáním momentu se zvyšuje výpočetní náročnost, proto jej přidáváme pouze tam, kde pomáhá vyřešit problém s uváznutím v lokálním minimu.

- Pokud síť učíme podobné vzory, tak jejich odpovídající minima se budou nacházet blízko sebe na chybové funkci. Při malém počtu skrytých neuronů nebudou tato minima výrazná a může dojít k jejich splynutí (zhoršení vlastnosti sítě). Pokud přidáme jeden nebo více skrytých neuronů, tak dojde ke zjemnění chybové funkce a minima od sebe snadněji odlišíme. S přidáváním neuronů se doporučuje snížit hodnotu koeficientu učení.
- Uváznutí v lokálním minimu lze také ovlivnit strategií výběru vzorů z tréninkové množiny. Nejméně odolný je sekvenční výběr, naopak náhodný výběr se jeví jako nejvhodnější.

3.1.2 Koeficient učení

V popisu učicího algoritmu backpropagation se vyskytl koeficient učení η , který určuje rychlost učení sítě a tím se také podílí na kvalitě dosaženého výsledku učení. Koeficient učení ve velké míře ovlivňuje nalezení globálního minima chybové funkce. Vliv koeficientu učení je znázorněn na obrázku 3.2 a jednotlivé případy budou dále vysvětleny.



Obr. 3.2: Grafické znázornění vlivu koeficientu učení [12]

Dle [12] by bylo nejlepší použít optimální hodnotu koeficientu učení η_{OPT} , při které by docházelo ke konvergenci k minimálnímu řešení během jednoho kroku učení. Pro hodnoty $\eta < \eta_{OPT}$ je zaručena konvergence k minimálnímu řešení, ale učení může být zbytečně pomalé. Pro hodnoty učení z rozsahu $\eta_{OPT} < \eta < 2\eta_{OPT}$ bude docházet k oscilacím, avšak nakonec dosáhneme globálního minima. Pro velké hodnoty $\eta > 2\eta_{OPT}$ nastane případ, kdy dochází k divergenci u globálního minima.

Pro stanovení vhodné hodnoty koeficientu učení neexistují žádná doporučení, stanovuje se experimentálně. Hodnota je volena z intervalu $(0,1>$. Jak již bylo řečeno, koeficient učení nemusí být konstantní v průběhu celého algoritmu. V případě proměnného koeficientu učení je na začátku učení sítě nastaven na nízkou hodnotu, kterou se během učení sítě zvětšuje.

4 POPIS OBJEKTŮ

Pro pochopení a teoretický rozbor problematiky popisu objektu byla použita literatura [8] až [11]. Popis objektů je jedním ze základních kroků při zpracování a rozpoznávání obrazu. Dle [9] 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 Selektce 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 [11] 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 [11] 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 hodili, 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 [9] 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 [18] 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.

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

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

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

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

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

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

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

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 POPIS FUNKCE VYTVOŘENÝCH PROGRAMŮ

Pro řešení bakalářské práce byly vytvořeny programy, které jsou uvedeny v příloze 3, jsou typu M-file a byly vytvořeny v programu Matlab verze R2009b. Pro práci s funkcí standardní sigmoida, jenž reprezentuje přenosovou funkci, je potřeba, aby byl nainstalován Fuzzy logic Toolbox 2.2.10. Dále bude uveden stručný popis jednotlivých programů napsaných pro dvouvrstvou síť. Obdobné programy, jak pro práci s dvouvrstvou sítí jsou vytvořeny i pro třívrstvou síť. Dá se říci, že hlavní rozdíl je pouze v implementaci učení pro třívrstvou síť na místo dvouvrstvé sítě. Ostatní kód programu je v podstatě totožný.

Vypocet_momentu.m

Výstupem programu jsou vypočítané momenty vzorů písmen tvořící tréninkovou a testovací množinu. Momenty se počítají podle vztahu uvedených v kapitole 4.2 a poté dochází k jejich normalizaci, viz kapitole 4.2.1.

Dvouvrstva_sit.m

Program slouží pro učení dvouvrstvé sítě s libovolně nastavenými parametry: počet neuronů skryté vrstvy `Neuron.H`, koeficient učení `N` a momentu `A`, velikost strmosti `Strmost`, interval pro náhodné nastavení vah `WPoc.HorMez`. Také je možno zvolit celkovou chybu, na kterou se má síť naučit, a omezení počtu cyklů při uvážnutí učení v lokálním minimu. Po skončení učení se vykreslí chybová funkce. V programu je možnost uložení konfigurace dané sítě. Uloží se všechny výše uvedené libovolně nastavené parametry, počáteční náhodně nastavené váhy a vypočítané hodnoty vah po skončení učení.

Dvouvrstva_sit_TestPocetNeuron.m

Program byl vytvořen k testování topologie dvouvrstvé sítě na rychlost učení. Jsou zde opět parametry pro nastavení konfigurace sítě. Počet neuronů skryté vrstvy je definován jako posloupnost hodnot, hodnoty z této posloupnosti jsou v opakujícím se cyklu nastavovány jako aktuální a pro jednotlivé hodnoty se provádí učení sítě. Po naučení sítě jsou zjištěné výsledky učení uloženy, uloží se také daná síť (tak jak to bylo definovanou v předchozím odstavci) a dojde k načtení další hodnoty z posloupnosti a tedy učení další sítě. Výstupem programu je soubor uložených sítí a uložený souhrn výsledků učení pro všechny testované sítě.

Stejná logika testování je použita i v dalších programech pro testování koeficientu učení, momentu a strmosti na rychlost učení neuronových sítí.

Dvouvrstva_sit_Rozpoznani.m

Program slouží pro testování rozpoznání písmen z tréninkové a testovací množiny již naučené dvouvrstvé sítě. Procentuální hodnota vyjadřující počet správně rozpoznaných písmen tréninkové množiny je uložena v proměnné ProcRozPismTrenMno. Řetězec chybně rozpoznaných písmen je uložen v ChybaRozTrenMno. Obdobné proměnné jsou definovány i pro testovací množinu.

Dvouvrstva_sit_TestRozpoznani.m

Jádro programu je stejné jako v programu z předchozího odstavce. Rozdíl je pouze v tom, že testování probíhá pro více sítí postupně za sebou. Pomocí posloupnosti MenKof si definujeme hodnoty měněného parametru skupiny sítí (je nutné používat skupiny sítí vytvořené testováním jednotlivých koeficientů). V cyklu jsou načítány jednotlivé sítě, na kterých je prováděn test rozpoznání. Do proměnné statistika se k jednotlivým vzorům ukládá počet chybných rozpoznání daného vzoru, tato proměnná vystihuje počet chybných rozpoznání pro celou skupinu testovaných sítí. Program ukládá všechny potřebné parametry (Neuron.H, N, A, Strmost) pro určení testovaných sítí společně s řetězcí chybně rozpoznaných písmen pro jednotlivé sítě, procentuální hodnoty rozpoznání a také již výše zmíněnou statistiku.

6 OPTIMALIZACE PARAMETRŮ NEURONOVÉ SÍTĚ Z HLEDISKA RYCHLOSTI UČENÍ

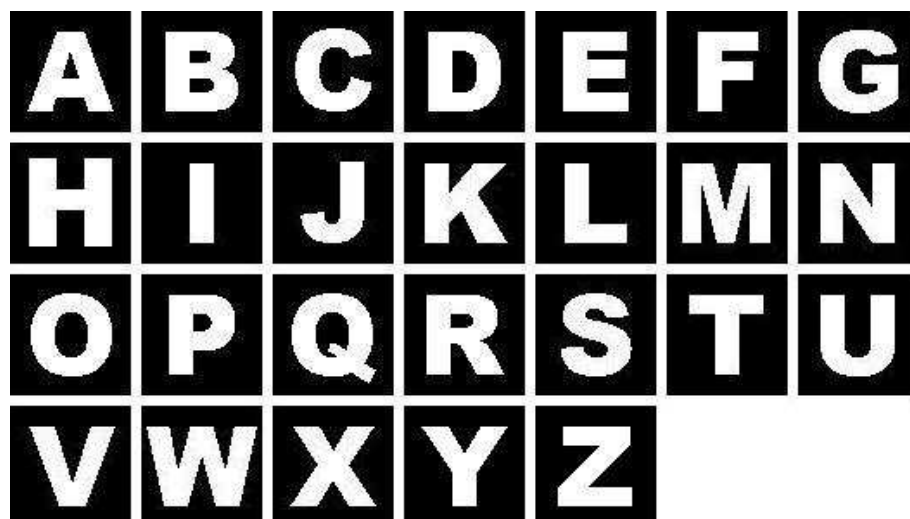
V této kapitole si ukážeme jaký vliv má topologie a hodnoty dalších parametrů na rychlost učení neuronové sítě. Budeme používat vícevrstvou neuronovou síť s učícím algoritmem backpropagation. Pro popis písmen využijeme momentový popis, jenž byl definován ve čtvrté kapitole.

Rychlost učení (tím rozumíme počet cyklů nutných k naučení sítě) budeme sledovat při učení tréninkové množiny na celkovou chybu 0.1 (ukončovací podmínka učení sítě v případě velmi pomalého učení nebo uváznutí v lokálním minimu je 20000 cyklů). Veškeré sítě společně s parametry použité pro testování se nacházejí v příloze 3, kde jsou také uvedeny zpracované výsledky všech testování.

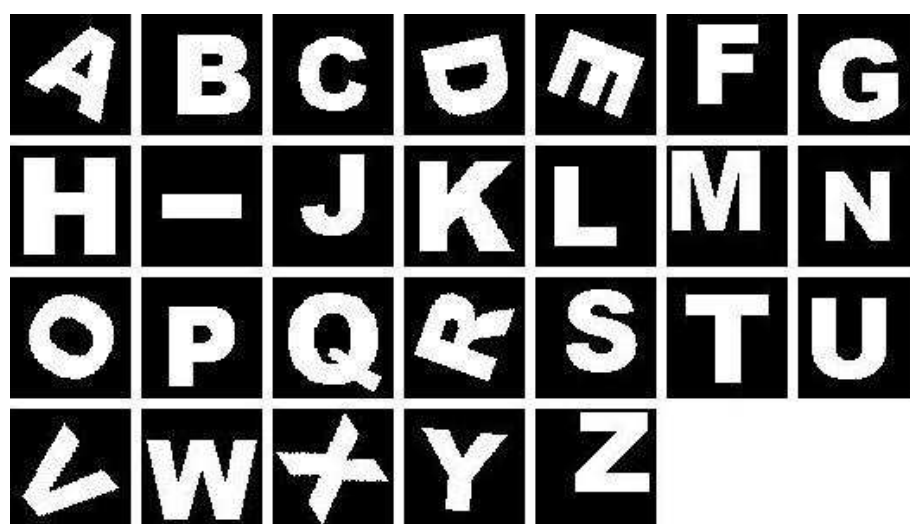
6.1 Použité vzory písmen

Pro učící vzory bylo použito 26 písmen od A do Z s použitým motivem 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).

Tréninková množina je tedy tvořena 26 vzory, jeden vzor pro každé středově orientované písmeno stejné velikosti. Testovací množina se skládá ze 130 vzorů (5 vzorů pro každé písmeno), písmena v testovací množině byla změněna. Změny byly voleny tak, aby byly otestovány výhody momentového popisu. A proto testovací množinu tvoří písmena, jež jsou otočena o libovolný úhel, posunuta v libovolném směru, zvětšena nebo zmenšena. Na obrázku 6.1 a 6.2 jsou uvedena písmena tvořící tréninkovou množinu a také část testovací množiny. Celá testovací množina se nachází v příloze 3.



Obr. 6.1 : Vzory písmen tvořící tréninkovou množinu



Obr. 6.2 : Vzory písmen tvořící část testovací množiny

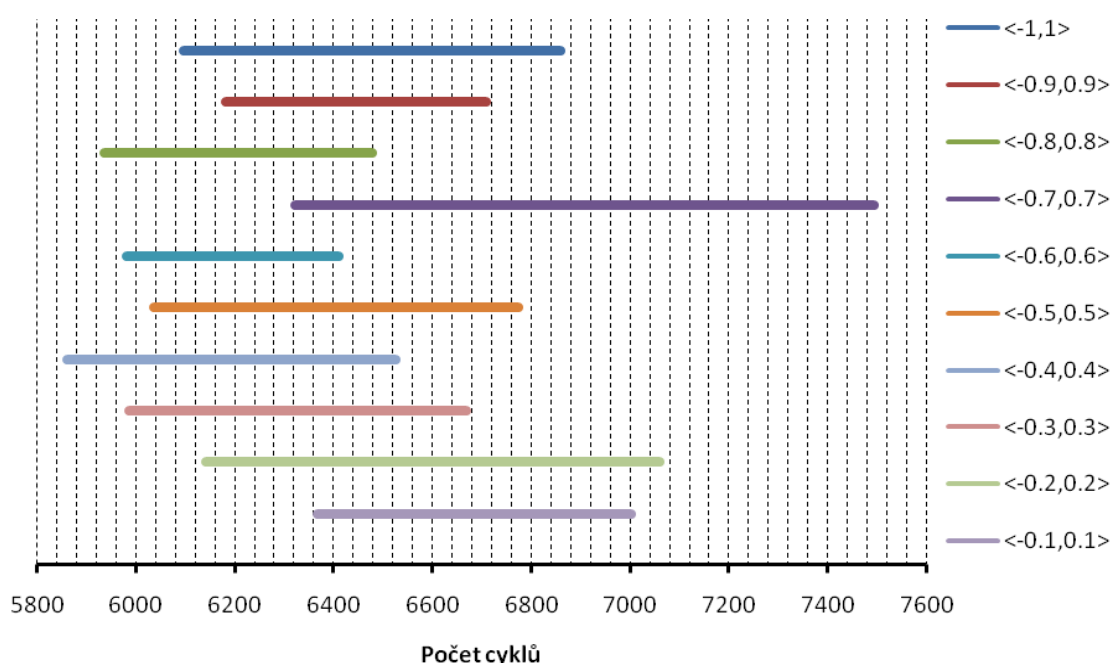
6.2 Testování vlivu počátečních vah na rychlost učení

V této kapitole dojde ke zhodnocení vlivu počátečních vah na rychlost učení. Hodnoty počátečních vah se náhodně vybírají z intervalů, které jsou symetrické kolem nuly, např. $\langle -0.5, 0.5 \rangle$. Pro každý testovaný interval bylo učení neuronové sítě provedeno pětkrát. Zpracované výsledky celého testování jsou uvedeny v příloze 3.

Při testování jsou všechny ostatní parametry neuronové sítě konstantní: koeficient učení 0.5, koeficient momentu 0.6, práh neuronů 1, použitá přenosová funkce

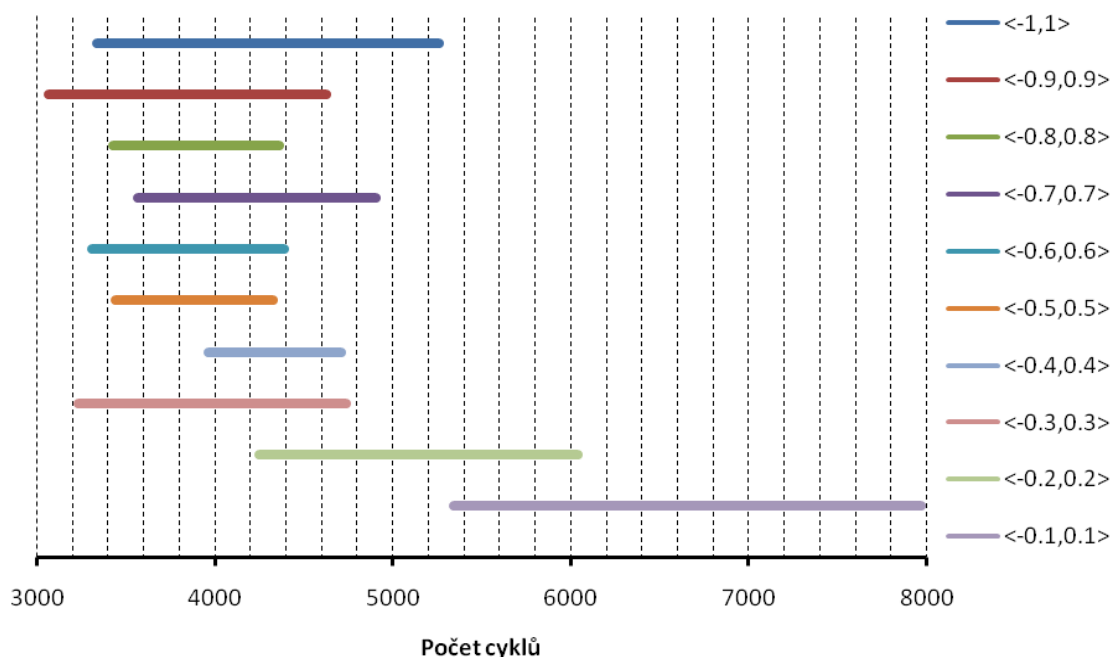
standardní sigmoida se strmostí jedna. Tyto parametry byly zvoleny na základě zkušeností a předběžného testování.

Dvouvrstvá síť: Pro topologii 7-120-26 byly testováním nalezeny tři nejvhodnější intervaly ($<-0.8,0.8>$, $<-0.6,0.6>$, $<-0.3,0.3>$), z kterých se náhodně vybírají hodnoty pro nastavení počátečních vah sítě, z hlediska vlivu na rychlost naučení sítě. Při výběru intervalů se posuzovala rychlost učení sítě pro daný interval a rozptyl učení sítě v daném intervalu (tento požadavek měl větší váhu). Na obrázku 6.3 je znázorněn rozptyl všech testovaných intervalů pro topologii 7-120-26. U dalších dvou sítí 7-100-26 a 7-160-26 bylo zjištěno, že u všech tří sítí interval $<-0.6,0.6>$ zajišťoval výbornou rychlost učení sítě a nejmenší rozsah, při kterém došlo k naučení sítě.



Obr. 6.3 : Rozptyl všech testovaných intervalů pro topologii 7-120-26

Třívrstvá síť: Pro topologii 7-80-20-26 byly testováním nalezeny tři nejvhodnější intervaly ($<-0.8,0.8>$, $<-0.5,0.5>$, $<-0.4,0.4>$), z kterých se náhodně vybírají hodnoty pro nastavení počátečních vah sítě, z hlediska vlivu na rychlost naučení sítě. Intervaly se vybírali obdobně jako u dvouvrstvé sítě, na obrázku 6.4 je znázorněn rozptyl všech testovaných intervalů pro topologii 7-80-20-26. U dalších dvou sítí 7-120-50-26 a 7-90-80-26 bylo zjištěno, že u všech tří sítí interval $<-0.5,0.5>$ zajišťoval výbornou rychlost učení sítě a nejmenší rozsah, při kterém došlo k naučení sítě.



Obr. 6.4 : Rozptyl všech testovaných intervalů pro topologii 7-80-20-26

6.3 Testování vlivu topologie sítě na rychlost učení

Pro testování vlivu topologie sítě byla vybrána dvouvrstvá a třívrstvá síť. Jednovrstvá síť se nebyla schopna naučit tréninkovou množinu za adekvátní počet cyklů a více jak třívrstvá síť je zbytečně velká pro řešení našeho problému (rozpoznávání písmen). Budeme se zabývat pouze vhodným počtem neuronů ve skrytých vrstvách, protože počet neuronů vstupní vrstvy je určen příznakovým vektorem, v našem případě je to sedm vstupních neuronů pro momenty ϕ_1 až ϕ_7 . Vstupní vrstva bude sloužit pouze pro distribuci vstupních signálů (momentů popisujících vzory písmen). Počet výstupních neuronů je určen počtem klasifikovaných tříd, v našem případě potřebujeme pro každé písmeno jeden výstupní neuron, tedy 26 výstupních neuronů.

Při testování topologie sítě jsou všechny ostatní parametry konstantní: koeficient učení 0.5, koeficient momentu 0.6, práh neuronů 1, použitá přenosová funkce standardní sigmoida se strmostí jedna, pevný interval pro počáteční náhodné nastavení vah zjištěný v kapitole 6.2.

Dvouvrstvá síť: Při testování vlivu topologie na rychlost učení byly učené sítě 7-H-26, kde počet neuronů skryté vrstvy H byl postupně vybírán z řady (10, 20, ..., 300), vzniklo tedy 30 sítí. Nejlepšího výsledku dosáhla topologie 7-130-26 s 5690 cykly nutnými k naučení. Při 10 skrytých neuronech trvalo učení nejdéle (9204 cyklů), to může být zapříčineno tím, že síť má k dispozici malé množství váhových

spojení, které při učení může měnit. Rychlost učení většiny ostatních sítí se pohybovala v rozmezí 6000 až 7000 cyklů.

Třívrstvá síť: Při testování vlivu topologie na rychlost učení byly učené sítě 7-H1-H2-26, kde počet neuronů první skryté vrstvy H1 byl postupně vybírán z řady (10, 20, ..., 100, 120, 140, ..., 200, 250, 300) a pro každý počet H1 byl z téže řady postupně vybírán počet neuronů druhé skryté vrstvy H2, vzniklo tedy 289 sítí. Nejlepšího výsledku dosáhla topologie 7-50-40-26 s 3221 cykly nutnými k naučení. Učení sítě 7-100-10-26 se jako jediné z testovaných zastavilo v lokálním minimu, učení bylo po 20000 cyklech zastaveno na chybě 0.5. Rychlost učení většiny ostatních sítí se pohybovala v rozmezí 3500 až 5000 cyklů.

6.4 Testování vlivu koeficientu učení na rychlost učení

Pro testování vlivu koeficientu učení byly použity nejvhodnější intervaly k počátečnímu nastavení vah zjištěné v kapitole 6.2 a nejvhodnější topologie zjištěné v kapitole 6.3, tedy dvouvrstvá síť (7-130-26, $<-0.6, 0.6>$) a třívrstvá síť (7-50-40-26, $<-0.5, 0.5>$). Ostatní parametry jsou konstantní: koeficient momentu 0.6, práh neuronů 1, použitá přenosová funkce standardní sigmoidy se strmostí jedna.

Testované hodnoty koeficientu učení byly vybírány z intervalu $(0, 1>$. Hodnoty koeficientu učení byly vybírány z řady (0.001, 0.01, 0.04, 0.07, 0.1, 0.2, ..., 1). Zkušebně byly otestovány i větší hodnoty, ale učení při těchto hodnotách bylo nevyhovující. Nejprve docházelo k občasnému uváznutí v lokálním minimu a při dalším zvětšování koeficientu učení (u dvouvrstvé sítě pro hodnotu 3, u třívrstvé sítě pro hodnotu 2) docházelo k pravidelnému uváznutí v lokálním minimu.

Učení sítě pro každou hodnotu koeficientu učení bylo provedeno čtyřikrát, aby došlo k zohlednění vlivu počátečního náhodného nastavení vah na učení. Za směrodatnou hodnotu vypovídající o kvalitě učení bereme průměrnou hodnotu z těchto čtyř měření (ve všech tabulkách šesté kapitoly je uvedena tato průměrná hodnota počtu cyklů). Tento princip testování koeficientu učení bude použit i pro další testované parametry v kapitolách 6.5 a 6.6.

U obou topologií byly vybrány tři nejvhodnější hodnoty učení, a to kvůli eliminaci vlivu konstantní hodnoty momentu, tzn. pro některou hodnotu učení (kdy učení sítě dosahovalo kvalitních výsledků, ale ne nejlepších) nemusí být vhodná právě zvolená konstantní hodnota momentu. A pro jiné hodnoty momentu (ty budou testovány v kapitole 6.5), může dojít ke zlepšení rychlosti učení.

Dvouvrstvá síť: Koeficienty učení, při kterých docházelo k nejlepší rychlosti učení, jsou uvedeny v tabulce 6.1. U velikosti koeficientu učení 0.1 a menší bylo učení velmi pomalé a po 20000 cyklech bylo zastaveno, aniž by dosáhlo požadované celkové chyby (se zvyšující hodnotou koeficientu učení se zmenšovala dosažená celková chyba sítě).

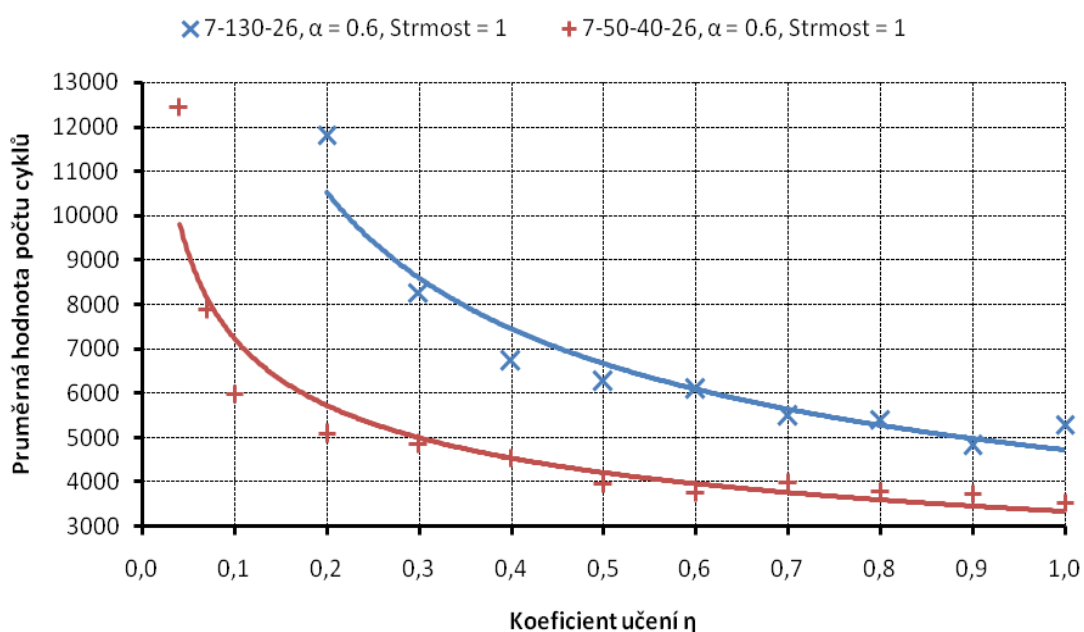
Až na pár výjimek platil trend, kdy se rychlost učení snižovala se zvětšující hodnotou koeficientu učení.

Tab. 6.1 : Výsledek testování vlivu koeficientu učení na rychlost učení

DVOUVRSTVÁ SÍŤ 7-130-26		TŘÍVRSTVÁ SÍŤ 7-50-40-26	
Koeficient učení	Počet cyklů	Koeficient učení	Počet cyklů
0.8	5409	0.6	3745
0.9	4810	0.9	3735
1	5290	1	3530

Třívrstvá síť: Koeficienty učení, při kterých docházelo k nejlepší rychlosti učení, jsou uvedeny v tabulce 6.1. U velikosti koeficientu učení 0.01 a menší bylo učení velmi pomalé a po 20000 cyklech bylo zastaveno, aniž by dosáhlo požadované celkové chyby (se zvyšující hodnotou koeficientu učení se zmenšovala dosažená celková chyba sítě). Až na pár výjimek platil trend, kdy se rychlost učení snižovala se zvětšující hodnotou koeficientu učení.

Pro přehlednost byly do grafu na obrázku 6.5, jenž znázorňuje vliv koeficientu učení na rychlost učení, vynášeny pouze hodnoty učení, u kterých nedocházelo k uvážnutí v lokálním minimu. Vynesené hodnoty byly prokládány spojnici trendu (typ trendu byl vybrán v závislosti na změřených hodnotách), aby lépe vynikla závislost rychlosti učení na hodnotě koeficientu učení, u závislosti na obrázku 6.5 byl použitý mocninný trend. Popsaný postup tvorby grafů bude použit i v kapitolách 6.5 a 6.6.



Obr. 6.5 : Závislost rychlosti učení na koeficientu učení dvouvrstvé a třívrstvé sítě

U dvouvrstvých i třívrstvých sítí, jejichž učení bylo po 20000 cyklech ukončeno, bylo provedeno testování koeficientu učení i pro jiné hodnoty momentu. Protože jak již bylo řečeno, nevhodná kombinace těchto dvou parametrů může způsobit nežádoucí průběh učení. Některým sítím (dvouvrstvá síť s hodnotami učení 0.1 a 0.07, třívrstvá síť s hodnotou učení 0.01) změna hodnoty momentu pomohla, ale výsledky rychlosti učení se pohybovaly okolo 10000 cyklů, a z tohoto hlediska nelze brát tyto hodnoty koeficientu učení za vhodné při požadavku velké rychlosti učení sítě.

6.5 Testování vlivu koeficientu momentu na rychlost učení

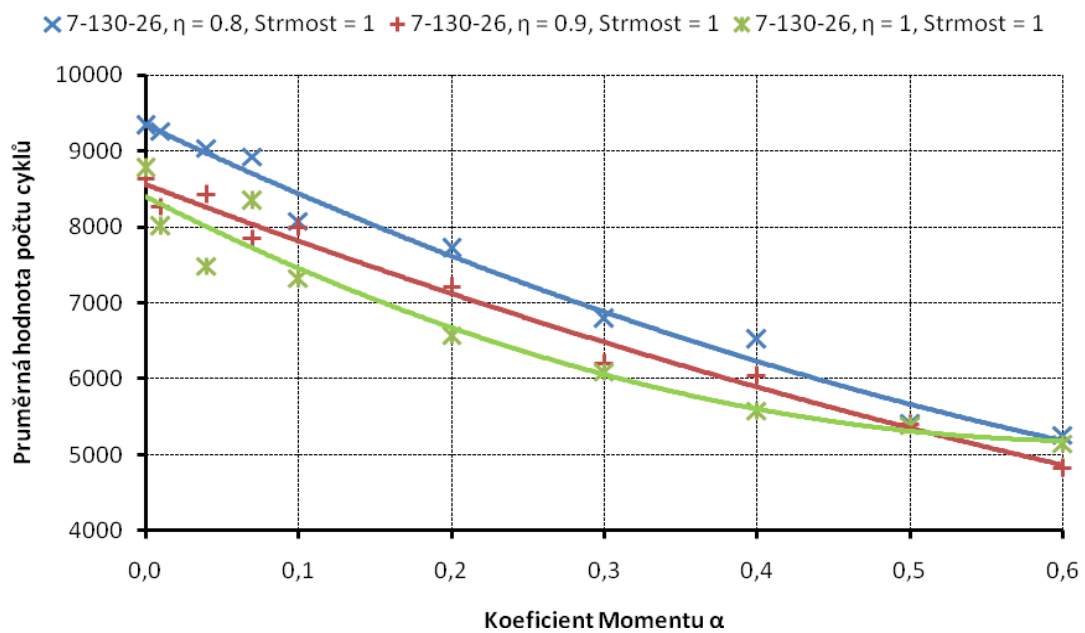
Pro testování vlivu koeficientu momentu byly použity nejvhodnější intervaly k počátečnímu nastavení vah zjištěné v kapitole 6.2 a nejvhodnější topologie zjištěné v kapitole 6.3 s hodnotami koeficientu učení z kapitoly 6.4. Vznikne tedy šest sítí (viz tabulka 6.1) v následujících konfiguracích: dvouvrstvá síť (7-130-26, $<-0.6, 0.6>$, hodnoty učení 0.8, 0.9 a 1) a třívrstvá síť (7-50-40-26, $<-0.5, 0.5>$, hodnoty učení 0.6, 0.9 a 1). Ostatní parametry jsou konstantní: práh neuronů 1, použitá přenosová funkce standardní sigmoida se strmostí jedna.

Testované hodnoty koeficient momentu byly vybírány z intervalu $<0, 1>$. Hodnoty koeficientu momentu byly vybírány z řady (0, 0.01, 0.04, 0.07, 0.1, 0.2, ..., 1). Hodnotu momentu nula chápeme jako učení sítě bez použití momentu. U obou topologií byly vybrány dvě nejvhodnější hodnoty momentů.

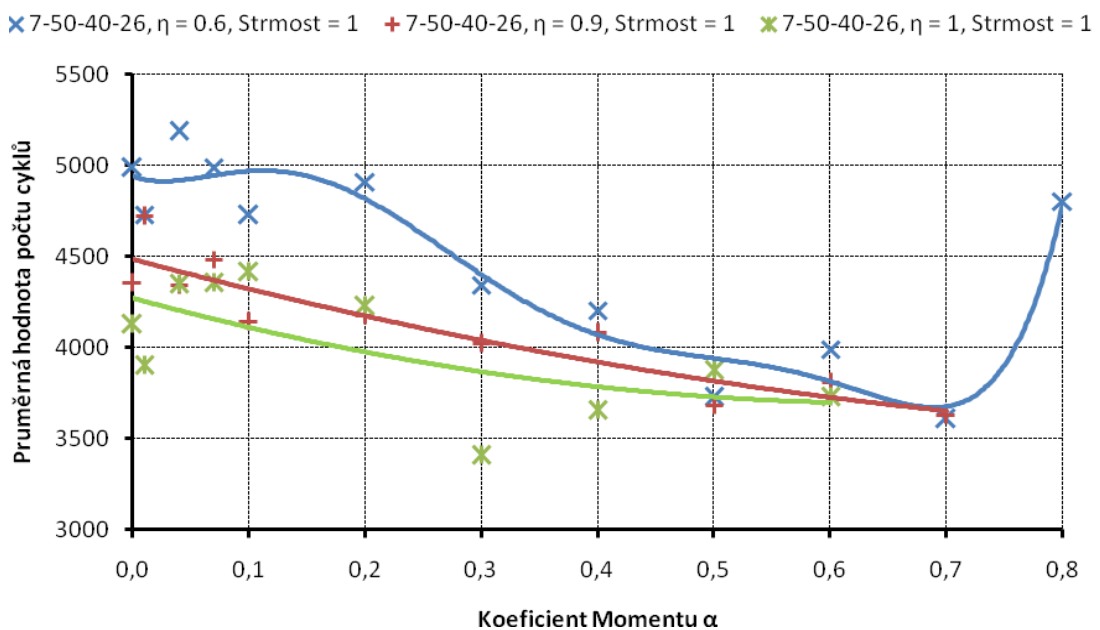
Dvouvrstvá síť: Koeficienty momentu, při kterých docházelo k nejlepší rychlosti učení, jsou uvedeny v tabulce 6.2. U všech tří sítí 7-130-26 s hodnotami učení (0.8, 0.9, 1) a pro hodnoty momentu (0.7, 0.8) docházelo během učení k občasnému uváznutí v lokálním minimu. A proto nebyly tyto hodnoty vybrány jako nejvhodnější, i když rychlost učení v případě, že nedošlo k uváznutí v lokálním minimu, byla lepší než pro vybrané hodnoty 0.6. Pro hodnoty momentu větší jak 0.9 u všech testovaných sítí docházelo k pravidelnému uváznutí v lokálním minimu. U všech tří použitých sítí lze říci, že pro ostatní hodnoty momentu (až na pár výjimek) platí: se zvětšující hodnotou momentu dochází k rychlejšímu naučení sítě. Popsané závislosti jsou na obrázku 6.6 (použitý polynomický trend druhého řádu).

Tab. 6.2 : Výsledek testování vlivu koeficientu momentu na rychlost učení

DVOUVRSTVÁ SÍŤ 7-130-26			TŘÍVRSTVÁ SÍŤ 7-50-40-26		
Koeficient učení	Koeficient momentu	Počet cyklů	Koeficient učení	Koeficient momentu	Počet cyklů
0.9	0.6	4823	0.6	0.7	3609
1	0.6	5151	1	0.3	3408



Obr. 6.6 : Závislost rychlosti učení na koeficientu momentu dvouvrstvé sítě



Obr. 6.7 : Závislost rychlosti učení na koeficientu momentu třívrstvé sítě

Třívrstvá síť: Koeficienty momentu, při kterých docházelo k nejlepší rychlosti učení, jsou uvedeny v tabulce 6.2. Během učení sítí docházelo k uváznutí v lokálním minimu pro následující kombinace hodnot učení a momentu: učení 0.6 a moment (0.9, 1), učení 0.9 a moment (0.8, 0.9, 1), učení 1 a moment (0.7, 0.8, 0.9, 1). Pro ostatní

hodnoty momentu (až na pár výjimek) společně s hodnotami učení 0.9 a 1 se rychlost učení zmenšovala se zvětšující hodnotou momentu. Při koeficientu učení 0.6 není závislost tak jednoznačná: nejprve dochází k pomalému růstu počtu učících cyklů, kolem hodnoty momentu 0.1 dochází ke klesání a od momentu 0.7 k prudkému růstu počtu cyklů (pro hodnoty větší jak 0.8 dochází k uváznutí v lokálním minimu). Avšak zajímavým faktem je, že rychlost učení dosažená s nulovým momentem a při učení 0.6, 0.9 a 1 nemá nejhorší výsledek (jak tomu bylo u dvouvrstvé sítě), ale stále se řadí mezi ty horší rychlosti učení. Popsané závislosti jsou na obrázku 6.7 (použitý polynomický trend šestého řádu pro $\eta = 0.6$ a druhého řádu pro $\eta = 0.9; 1$).

6.6 Testování vlivu strmosti přenosové funkce na rychlost učení

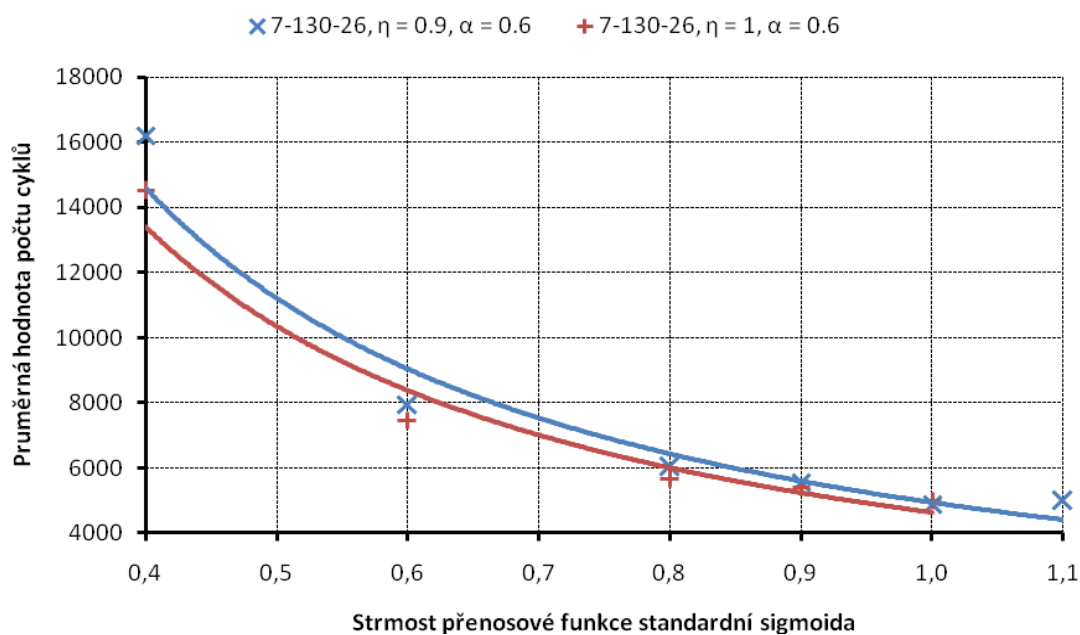
Pro testování vlivu strmosti přenosové funkce byly použity nejvhodnější intervaly k počátečnímu nastavení vah zjištěné v kapitole 6.2 a nejvhodnější topologie zjištěné v kapitole 6.3 s hodnotami koeficientu učení z kapitoly 6.4 a s hodnotami koeficientu momentu z kapitoly 6.5. Dostaneme tedy čtyři sítě, jejichž konfigurace jsou přehledně zobrazeny v tabulce 6.2. Ostatní parametry jsou konstantní: práh neuronů 1.

Testované hodnoty strmosti byly vybírány z řady (0.4, 0.6, 0.8, 0.9, ..., 1.2, 1.4). Při testování hodnot strmosti větší jak 1.4 docházelo k uváznutí v lokálním minimu pro dvouvrstvou i třívrstvou síť. U obou topologií byla vybrána nejvhodnější hodnota strmosti.

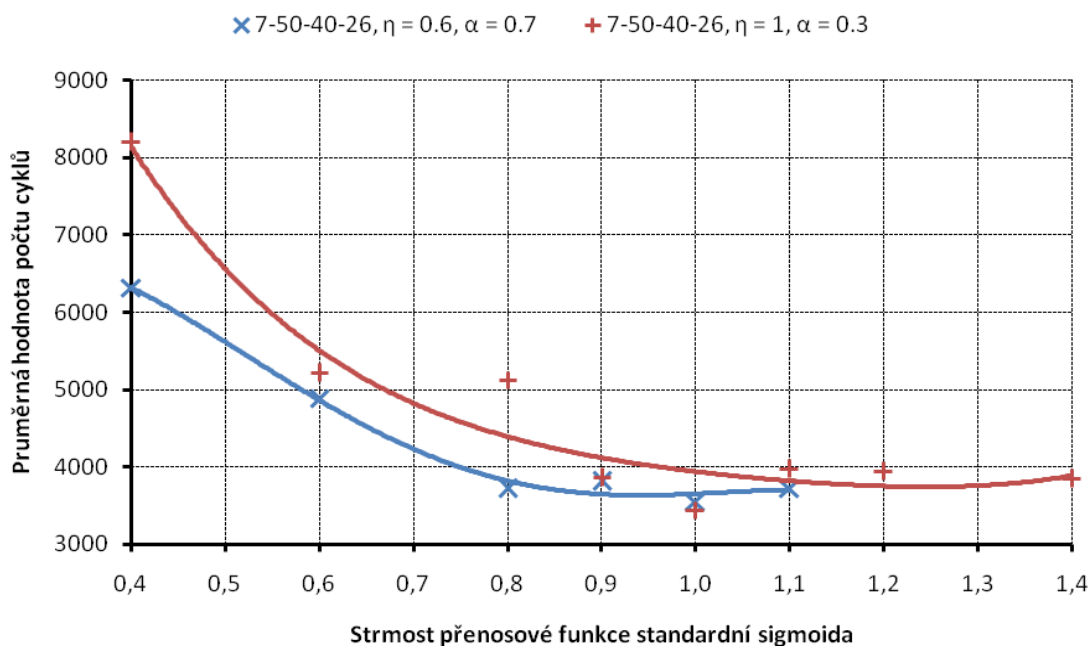
Dvouvrstvá síť: V tabulce 6.3 je zaznamenána nejvhodnější hodnota strmosti. Pro učení 0.9 a moment 0.6 se vyskytovalo občasné uváznutí v lokálním minimu při strmostech 1.2 a 1.4. Uváznutí v lokálním minimu bylo také nalezeno pro kombinaci učení 1, moment 0.6 a strmosti 1.1, 1.2, 1.4. U ostatních hodnot strmosti byl zjištěn vliv: se zvětšující hodnotou strmosti je lepší rychlost učení. Popsané závislosti jsou na obrázku 6.8 (použitý mocninný trend).

Tab. 6.3 : Výsledek testování vlivu strmosti přenosové funkce na rychlost učení

DVOUVRSTVÁ SÍŤ 7-130-26				TŘÍVRSTVÁ SÍŤ 7-50-40-26			
Koeficient učení	Koeficient momentu	Strmost	Počet cyklů	Koeficient učení	Koeficient momentu	Strmost	Počet cyklů
0.9	0.6	1	4889	1	0.3	1	3441



Obr. 6.8 : Závislost rychlosti učení na strmosti přenosové funkce dvouvrstvé sítě



Obr. 6.9 : Závislost rychlosti učení na strmosti přenosové funkce třívrstvé sítě

Třívrstvá síť: V tabulce 6.3 je zaznamenána nejvhodnější hodnota strmosti. Občasné uváznutí v lokálním minimu bylo nalezeno u sítí s kombinací parametrů: učení 0.6, moment 0.7, strmost 1.2 a 1.4. U ostatních hodnot strmosti byla zjištěna závislost: od první testovací hodnoty 0.4 docházelo ke klesání počtu cyklů nutných k naučení sítí,

kolem hodnoty 1, kde byla nejmenší rychlost učení, se závislost změnila v pomalu rostoucí. Popsané závislosti jsou na obrázku 6.9 (použitý polynomický trend čtvrtého řádu).

V tabulce 6.3, kde je uvedena nejlepší velikost strmosti dvouvrstvé a třívrstvé sítě, vidíme, že nejvhodnější hodnota strmosti je jedna. Tato hodnota je stejná, jak byla zvolena na počátku celého testování.

6.7 Testování vlivu topologie sítě na rychlost učení s nejvhodnějšími parametry

Na závěr zkoumání vlivu parametrů sítě na rychlost učení bylo provedeno opětovné testování počtu skrytých neuronů pro dvouvrstvou a třívrstvou síť s výše uvedenými parametry (tabulka 6.3), které byly zjištěny jako nejvhodnější.

Toto testování bylo provedeno, protože veškeré zkoušky hodnot jednotlivých parametrů se odvíjely od zvoleného počtu skrytých neuronů u daných sítí. Nejvhodnější počet skrytých neuronů byl u obou sítí zvolen při nastavených parametrech, ty byly zvoleny na základě zkušeností a předběžného testování. Jak vyplynulo z testování, tak tyto parametry nebyly z hlediska rychlosti učení nejvhodnější. A proto bylo provedeno opětovné hledání nejlepšího počtu skrytých neuronů s ambicí, že nalezneme topologii, jejíž rychlost učení je lepší než pro síť doposud prohlášené za nejlepší.

Při prvním testování vlivu topologie na rychlost učení bylo učení jednotlivých sítí prováděno pouze jednou. Avšak při tomto testování, bylo učení provedeno třikrát pro každou kombinaci skrytých neuronů, aby došlo k zohlednění vlivu počátečního náhodného nastavení vah na začátku učení.

Počty skrytých neuronů dvouvrstvé i třívrstvé sítě byly vybíraný ze stejných řad, které již byly definovány při prvním testování vlivu topologie v kapitole 6.3.

Dvouvrstvá síť: Tři nejlepší průměrné hodnoty rychlosti učení během tří měření dosáhly topologie: 7-20-26 s 4369 cykly, 7-110-26 s 4634 cykly, 7-160-26 s 4785 cykly. U těchto tří topologií byl testován vliv počátečního náhodného nastavení vah, během 10 učení pro každou topologii bylo zjištěno, že topologie 7-20-26 a 7-110-26 mají značnou závislost na počátečních vahách (rozdíl maximální a minimální dosažené rychlosti učení se pohyboval okolo 1300 cyklů). Topologie 7-160-26 byla nejméně závislá, síť se při 10 měřeních naučila v intervalu <4739,5219>. Takže z těchto měření vyplývá, že jednou z nejrychlejších a nejméně závislou na počátečních vahách je topologie 7-160-26. Srovnáním rychlosti učení s hodnotou v tabulce 6.3 vidíme, že došlo pouze k malému zlepšení rychlosti učení. Rychlost učení ostatních naučených sítí, jež se dokázaly naučit s celkovou chybou 0.1, se pohybovala mezi 4500 až 7000 cykly.

Třívrstvá síť: Testování probíhalo obdobně jako u dvouvrstvé sítě. Tři nejlepší průměrné hodnoty rychlosti učení během tří měření dosáhly topologie: 7-30-80-26

s 2977 cykly, 7-60-20-26 s 3249 cykly, 7-90-60-26 s 3338 cykly. U těchto tří topologií byl testován vliv počátečního náhodného nastavení vah, během 10 učení pro každou topologii bylo zjištěno, že topologie 7-30-80-26 a 7-60-20-26 mají značnou závislost na počátečních vahách (rozdíl maximální a minimální dosažené rychlosti učení se pohyboval okolo 2000 cyklů). Testováním topologie 7-90-60-26 bylo zjištěno, že je nejméně závislá. Síť se při 10 měřeních naučila v intervalu $\langle 3138, 3852 \rangle$. Takže z těchto měření vyplývá, že jednou z nejrychlejších a nejméně závislou na počátečních vahách je topologie 7-90-60-26. Srovnáním rychlosti učení s hodnotou v tabulce 6.3 vidíme, že došlo pouze k malému zlepšení rychlosti učení. Rychlost učení ostatních naučených sítí, jež se dokázaly naučit s celkovou chybou 0.1, se pohybovala mezi 3500 až 9000 cykly.

U testování dvouvrstvých i třívrstvých sítí nebyla vybrána ta síť, která byla ze tří měření nejrychlejší, protože u ní byla zjištěna značná závislost na nastavení počátečních vah. Dále budeme považovat za nejrychlejší síť ty, jejichž rychlost patřila mezi nejlepší během tří měření a zároveň měly nejmenší závislost na počátečním nastavení vah. Nejlepší přístup k tomuto závěrečnému testování topologie (případně i všech předchozích testování) by bylo provedení mnohem více měření pro dané konfigurace sítí. Tím by se velikost závislosti na počátečních vahách objevila ihned v průběhu testování a nemusel by být prováděn dodatečný test pro zjištění závislosti na počáteční hodnotě vah. Větší počet testování jednotlivých konfigurací sítí nebyl v našem případě možný kvůli časové náročnosti testů. Pro počet testování byl zvolen kompromis mezi velkým počtem měření a jediným měřením, a to hodnota 3. U předchozích testování koeficientů to byla hodnota 4.

6.8 Zhodnocení testů

Z testování vyplynulo nejlepší nastavení dvouvrstvé a třívrstvé sítě, při kterém docházelo k nejrychlejšímu naučení tréninkové množiny s celkovou chybou 0.1 a které měly nejmenší závislost na nastavení počátečních vah:

- topologie 7-160-26, interval $\langle -0.6, 0.6 \rangle$ pro náhodné nastavení vah, koeficient učení 0.9, koeficient momentu 0.6, hodnota strmosti přenosové funkce standardní sigmoida 1, práh neuronů 1
- topologie 7-90-60-26, interval $\langle -0.5, 0.5 \rangle$ pro náhodné nastavení vah, koeficient učení 1, koeficient momentu 0.3, hodnota strmosti přenosové funkce standardní sigmoida 1, práh neuronů 1

Za nejrychlejší síť však považujeme třívrstvou síť se získanou kombinací parametrů, protože její učení bylo o 1000 cyklů rychlejší než pro dvouvrstvou síť s nejvhodnější kombinací parametrů.

7 ROZPOZNÁVÁNÍ PÍSMEN NEURONOVOU SÍTÍ

V šesté kapitole jsme se zabývali výběrem neuronové sítě z hlediska nejlepší hodnoty rychlosti učení. V této kapitole však budeme hledat neuronovou síť, která nejlépe rozpoznává písmena tréninkové i testovací množiny. Vzory písmen tréninkové a testovací množiny již byly popsány v kapitole 6.1. Neuronové síť použité pro testování společně se zpracovanými výsledky testování se nalézají v příloze 3.

Pro testování byly použity neuronové síť získané testováním z šesté kapitoly. Ze 782 neuronových sítí, kdy každá síť měla nastavenou jinou kombinaci parametrů, byly vyřazeny ty síť, u které docházelo k uvážnutí v lokálním minimu (např. pokud se učení sítě zastavilo v lokálním minimu v jednom měření ze čtyř měření, tak již byla vyřazena z testu rozpoznávání písmen). Dostáváme tedy 737 sítí. V šesté kapitole bylo učení sítí opakované vícekrát, pro test rozpoznávání písmen byly vždy vybírány síť z prvního měření.

7.1 Výsledky testů rozpoznávání písmen

Jak již bylo řečeno, pro testování rozpoznávání písmen bylo použito 737 sítí: 104 dvouvrstevných a 633 třívrstevných sítí. Třívrstevných sítí je mnohem více, protože při testování topologií vzniklo více kombinací pro třívrstevnou síť.

Výstupem testování jsou procentuální hodnoty vyjadřující počet správně rozpoznávaných písmen tréninkové a testovací množiny. Pro každou testovanou síť je také ukládán řetězec chybně detekovaných tréninkových i testovacích vzorů ve formátu: Rozpoznávané písmeno (číslo vzoru v tréninkové/testovací množině) => 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 změnou měřítka, viz kapitola 6.1) bylo chybně rozpoznáno jako písmeno Q.

Pro dvouvrstevnou a třívrstevnou síť byla také během testování vytvořena statistika popisující, který vzor písmena a kolikrát byl chybně detekován. Tato statistika byla poté vyjádřena v procentuálních hodnotách, tyto procentuální hodnoty byly použity v obrázku 7.1. Přepočet na procenta se prováděl zvlášť pro dvouvrstevnou a třívrstevnou síť. Počet chybných detekcí daného vzoru byl podělen maximálním počtem chybných detekcí, které mohly u daného vzoru nastat. Maximální počet chybných detekcí je dán počtem testovaných sítí, tedy 104 pro dvouvrstevnou síť a 633 pro třívrstevnou síť. Pokud bychom přepočet na procenta provedli pro dvouvrstevnou a třívrstevnou síť dohromady, tak bychom dostali zavádějící výsledky (ty by byly způsobeny rozdílným počtem dvouvrstevných a třívrstevných sítí).

U všech testovaných sítí se potvrdil předpoklad, že rozpoznávání tréninkových vzorů bude stoprocentní. Rozpoznání vzorů testovací množiny se pohybovalo mezi 99.23% až 93.08%. Neuronové sítě, které dosáhly hodnoty 99.23%, chybně rozpoznaly pouze jediný vzor, a to vzor 85 reprezentující zmenšené písmeno Q. U tohoto vzoru došlo k chybné detekci pro všechny testované sítě. A proto síť s hodnotou rozpoznání 99.23% budeme považovat za nejlepší. Z celkového počtu sítí, bylo nalezeno 14 nejlepších sítí. Jejich nastavené parametry jsou uvedeny v tabulce 7.1, pro zajímavost k nim byly uvedeny i jejich rychlosti učení zjištěné v kapitole 6.

Tab. 7.1 : Přehled nejlepších sítí z hlediska rozpoznávání testovacích vzorů

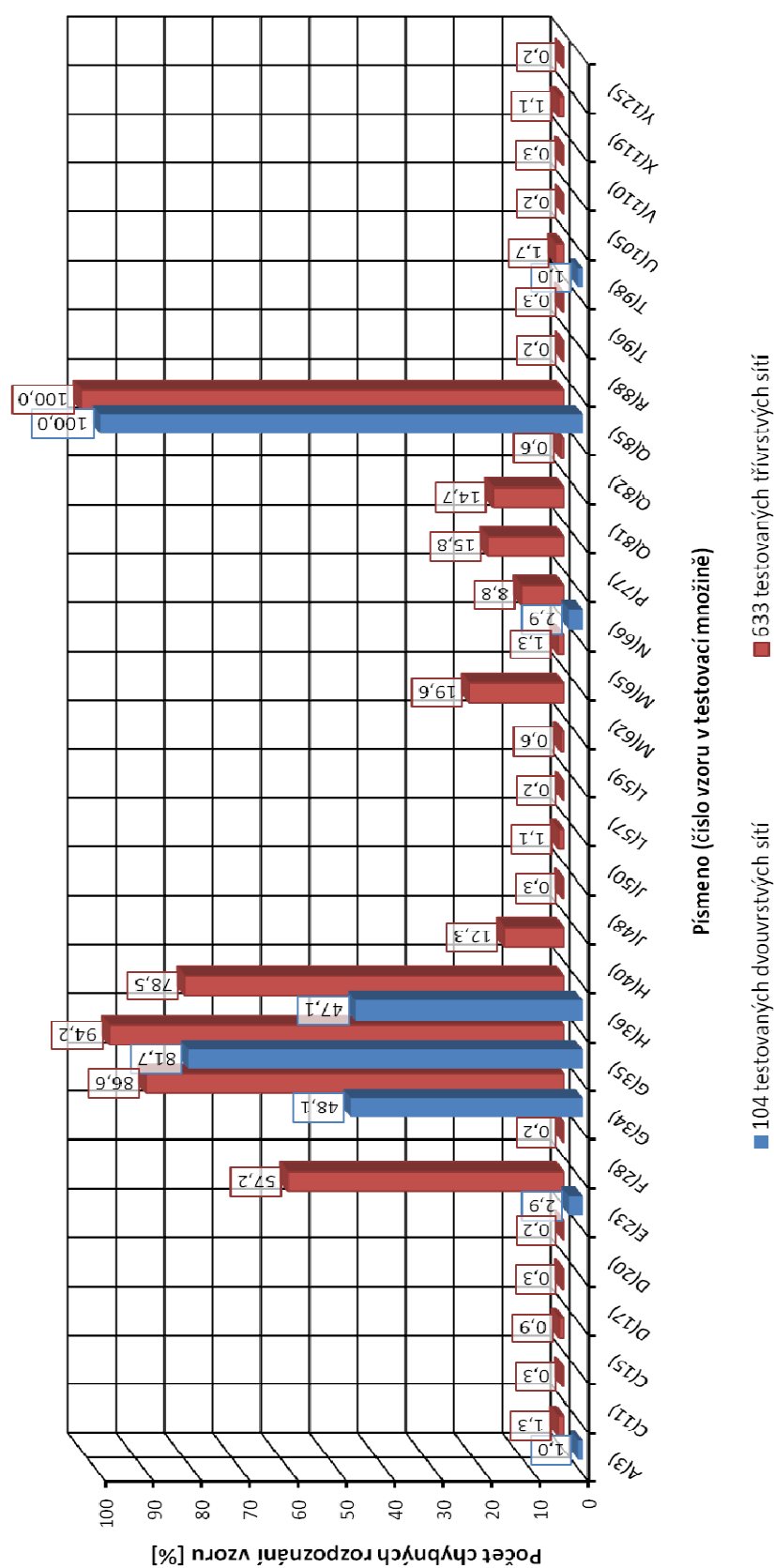
Topologie	Koeficient učení	Koeficient momentu	Strmost	Rozpoznání testovací množiny [%]	Rychlost učení [Počet cyklů]
7-130-26	0.80	0.04	1.0	99.23	9008
7-130-26	0.90	0.04	1.0	99.23	8281
7-130-26	0.20	0.60	1.0	99.23	12118
7-130-26	0.90	0.60	0.4	99.23	16762
7-130-26	1.00	0.60	0.4	99.23	14369
7-40-26	0.50	0.60	1.0	99.23	6339
7-60-26	0.50	0.60	1.0	99.23	6011
7-120-26	0.50	0.60	1.0	99.23	6334
7-140-26	0.50	0.60	1.0	99.23	6492
7-260-26	0.50	0.60	1.0	99.23	7807
7-10-26	0.90	0.60	1.0	99.23	4149
7-50-40-26	0.04	0.60	1.0	99.23	12085
7-60-80-26	0.50	0.60	1.0	99.23	3889
7-60-160-26	1.00	0.30	1.0	99.23	4264

V páté kapitole byla zmíněna závislost rychlosti učení na nastavených počátečních vahách. Z hlediska rozpoznání se tato závislost také vyskytuje. Pro síť 7-130-26, učení 0.8, moment 0.04, strmost 1 se rozpoznání vzorů testovací množiny pro 10 těchto naučených sítí pohybovalo od 99.23% do 97.69%. Hodnota rozpoznání 99.23% (jeden chybně rozpoznáný vzor: Q(85)) měla největší počet výskytů, hodnota 97.69% (tři chybně rozpoznané vzory: Q(85), G(34), G(35)) se vyskytla pouze jednou.

Na obrázku 7.1 je uveden sloupcový graf znázorňující počet chybně detekovaných vzorů testovací množiny v procentech, tzn. počet chybných detekcí je podělen celkovým počtem možných detekcí (pro dvouvrstvou síť 104 a třívrstvou síť 633) a převeden na procenta. V grafu se vyskytují pouze ty vzory, které byly alespoň jednou chybně rozpoznány dvouvrstvou nebo třívrstvou sítí. Dvouvrstvá síť chybně rozpoznávala menší počet vzorů než třívrstvá síť. Takže pokud u určitého vzoru pro dvouvrstvou síť není zobrazena hodnota, tak to znamená, že dvouvrstvá síť daný vzor

rozpoznala vždy správně a do grafu je tento vzor zařazen, protože ho chybně rozpoznávala pouze třívrstvá síť.

Z tabulky 7.1 a grafu uvedeného na obrázku 7.1 je patrné, že třívrstvá síť měla mnohem větší problémy s rozpoznáváním testovací množiny. Dvouvrstevných sítí s hodnotou rozpoznání 99.23% je 11, což je 10.57% ze 104 testovaných sítí. Naopak u 633 testovaných třívrstevných sítí pouze tři sítě dosáhly 99.23% rozpoznání testovací množiny, to je 0.47% ze všech třívrstevných sítí. Rozdíl je také vidět v počtu testovacích vzorů, které dané sítě měly problém rozpoznat. Dvouvrstevná síť měla problémy s 8 vzory, kdežto třívrstevná síť měla potíže s rozpoznáním 29 vzorů. Z testování je zřejmé, že správné rozpoznání písmen se změnou měřítka je problematičtější než u písmen, které byly otočeny nebo posunuty.



Obr. 7.1 : Sloupcový graf vyjadřující počet chybných rozpoznání testovacích vzorů v procentech pro dvouvrstvé a třívrstvé sítě

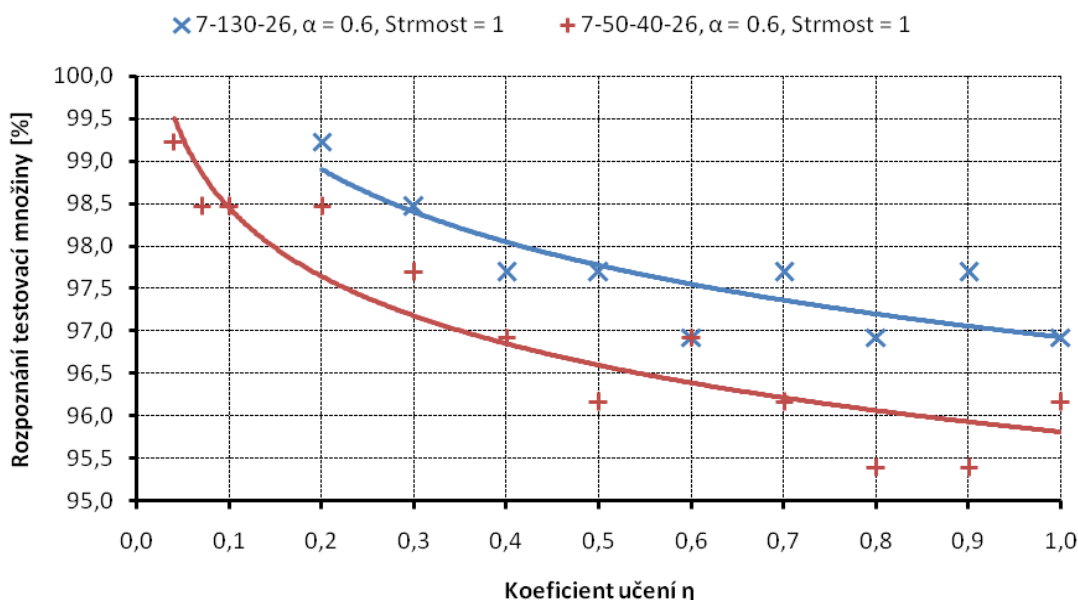
7.1.1 Závislost rozpoznání písmen na velikosti parametrů sítě

V šesté kapitole jsme se zabývali vlivem parametrů neuronové sítě na rychlost učení a zjišťovali jsme průběhy těchto závislostí. V této kapitole popíšeme vliv těchto parametrů na rozpoznávání testovací množiny, tyto závislosti nejsou tak zřejmé jako u rychlosti učení.

Počet skrytých neuronů: Při změně množství skrytých neuronů u dvouvrstvé a třívrstvé sítě nebyla zjištěna žádná konkrétní závislost mezi počtem skrytých neuronů a kvalitou rozpoznání testovacích vzorů.

Koeficient učení: Nejvíce zřejmý byl vliv právě u koeficientu učení: se zvětšující hodnotou učení docházelo ke zhoršujícímu se rozpoznání testovací množiny. U třívrstvé sítě byla strmost klesání této závislosti větší než u dvouvrstvé sítě. Nejlepší hodnota rozpoznání 99.23% byla vždy dosažena pro nejmenší hodnotu koeficientu učení, která byla testována. U dvouvrstvé sítě to byla hodnota 0.2, 0.04 pro třívrstvou síť. Hodnoty koeficientu učení se liší, protože uvážnutí v lokálním minimu u dvouvrstvé a třívrstvé sítě se vyskytovalo pro odlišné intervaly hodnot, viz kapitola 6.

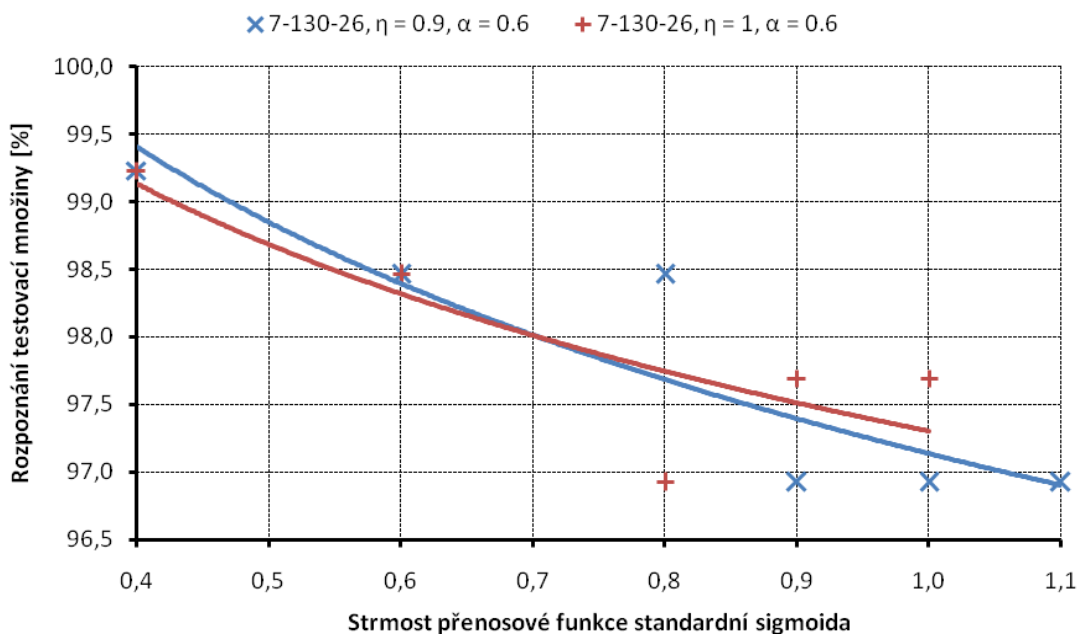
Popsané závislosti rozpoznání testovací množiny na koeficientu učení pro dvouvrstvou i třívrstvou síť jsou vyneseny do grafu na obrázku 7.2. Vynesené hodnoty byly prokládány spojnici trendu (typ trendu byl vybrán v závislosti na zjištěných hodnotách), aby lépe vynikla závislost. U závislosti na obrázku 7.2 je použitý mocninný trend.



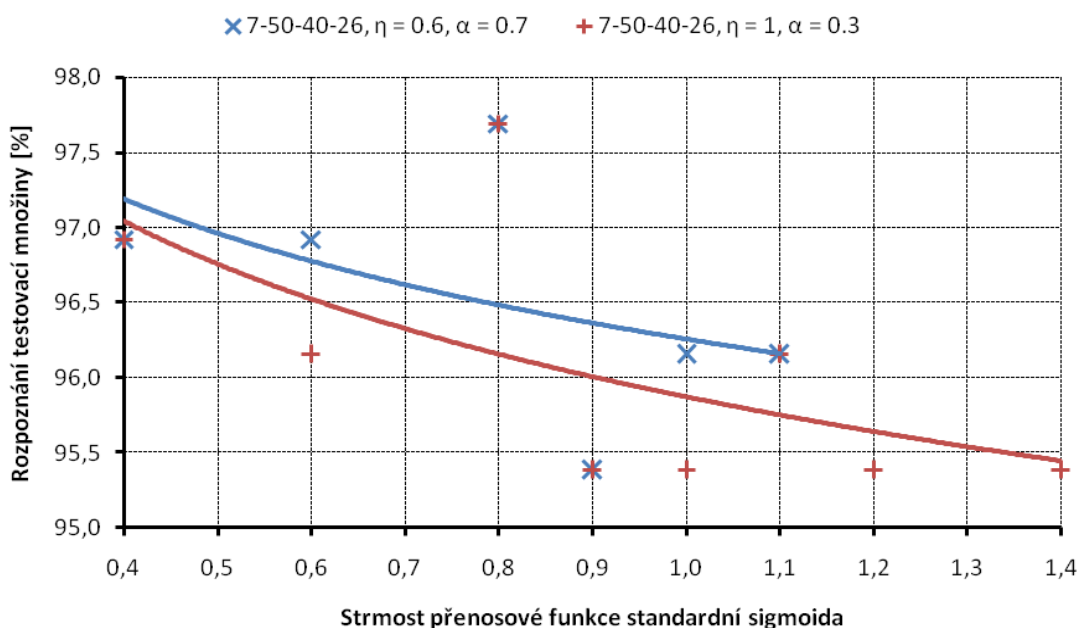
Obr. 7.2 : Závislost rozpoznání testovací množiny na koeficientu učení dvouvrstvé a třívrstvé sítě

Koeficient momentu: Naopak závislost rozpoznání testovací množiny na momentu byla nejméně zřetelná. Ze zjištěných hodnot, které jsou vyneseny do grafu uvedených v příloze 1 a 2, se závislost jeví, jako klesající se zvětšující hodnotou momentu. Tento závěr je však pouhým odhadem.

Strmost přenosové funkce: Závislost rozpoznání testovací množiny na strmosti přenosové funkce standardní sigmoida měla po proložení spojnicí trendu podobný průběh, jak u koeficientu učení. Docházelo ke zhoršování rozpoznání se zvětšující hodnotou strmosti. U dvouvrstvé i třívrstvé sítě docházelo k tomu že, hodnota rozpoznání zůstávala konstantní pro poslední dvě až tři testované hodnoty strmosti. Závislosti byly vytvořeny obdobně jako u koeficientu učení, jsou uvedeny na obrázku 7.3 pro dvouvrstvou síť a obrázku 7.4 pro třívrstvou síť, u všech závislostí na obrázku 7.3 a 7.4 použitý mocninný trend.



Obr. 7.3 : Závislost rozpoznání testovací množiny na strmosti přenosové funkce dvouvrstvé sítě



Obr. 7.4 : Závislost rozpoznání testovací množiny na strmosti přenosové funkce třívrstvé sítě

7.2 Závislost rozpoznání písmen na hodnotě celkové chyby sítě

Nejprve otestujeme závislost rozpoznání písmen na hodnotě celkové chyby sítě u neuronových sítí z tabulky 7.1, které mají nejlepší hodnotu rozpoznání testovací množiny. Pro testování byly vybrány jedna dvouvrstvá a třívrstvá síť, viz tabulka 7.2.

Tab. 7.2 : Test dvouvrstvé a třívrstvé sítě s nejlepší hodnotou rozpoznání testovací množiny

Celková chyba	Rozpoznání testovací množiny [%]	
	7-10-26,N(0.9),A(0.6),Strmost(1)	7-50-40-26,N(0.04),A(0.6),Strmost(1)
1.00E-01	99.23	99.23
5.00E-02	99.23	99.23
2.50E-02	99.23	99.23
1.25E-02	99.23	99.23
6.25E-03	98.46	99.23

Z tabulky 7.2 vidíme, že pouze u dvouvrstvé došlo ke změně hodnoty rozpoznání testovací množiny, a to při celkové chybě sítě 0.00625. Dvouvrstvá síť při celkové

chybě 0.00625 začala chybně rozpoznávat další vzor: písmeno J na vzoru 50. Obě sítě od začátku testování chybně rozpoznávaly písmeno Q na vzoru 85.

Dále otestujeme závislost rozpoznání písmen na hodnotě celkové chyby sítě u neuronových sítí s nižšími hodnotami rozpoznání testovací množiny. Ze všech testovaných dvouvrstvých a třívrstvých sítí byly vybrány ty sítě, které měly nejhorší hodnotu rozpoznání, viz tabulka 7.3.

Tab. 7.3 : Test dvouvrstvé a třívrstvé sítě s nižšími hodnotami rozpoznání testovací množiny

Celková chyba	Rozpoznání testovací množiny [%]	
	7-130-26,N(0.9),A(0.1),Strmost(1)	7-250-20-26,N(0.5),A(0.6),Strmost(1)
1.00E-01	96.92	93.08
5.00E-02	98.46	94.62
2.50E-02	99.23	95.38
1.25E-02	99.23	95.38
6.25E-03	99.23	95.38

Během všech učení v šesté kapitole byla požadována celková chyba 0.1. Při zmenšení celkové chyby na polovinu se rozpoznání u obou sítí zlepšilo, sítě správně rozpoznaly o dva vzory více než při celkové chybě 0.1. Dalším zmenšením celkové chyby na polovinu začaly sítě správně rozpoznávat jeden další vzor. Opětovné zmenšování celkové chyby už nemělo vliv na kvalitu rozpoznávání. Dvouvrstvé sítě, která měla na začátku testování obstojnou hodnotu rozpoznání, učení na menší celkovou chybu sítě pomohlo k hodnotě rozpoznání 99.23% (chybně rozpoznané písmeno Q na vzoru 85). Učení na menší celkovou chybu třívrstvé sítě zlepšilo rozpoznávání, ale ne tak, aby se síť zařadila mezi ty nejlepší.

Z testu tedy vyplývá, že celková chyba sítě 0.025 je nejvhodnější pro ukončení učení. Protože při této hodnotě dojde u neuronových sítí s nižšími hodnotami rozpoznání testovací množiny k nejvýraznějšímu zlepšení této hodnoty. A zároveň u sítí s kvalitní hodnotou rozpoznání ještě nedochází ke změně této hodnoty. Rozpoznání tréninkové množiny se během testování nijak nezměnilo. V příloze 3 je uvedena tabulka 7.2 a 7.3 společně s chybně rozpoznávanými vzory.

8 REŠERŠE PRÁCI ZABÝVAJÍCÍ SE ROZPOZNÁVÁNÍM ZNAKŮ NEURONOVOU SÍTÍ

V rámci této kapitoly bude uveden stručný popis několika přístupů, které byly zvoleny jinými autory, k řešení problematiky rozpoznávání znaků vícevrstvou neuronovou sítí s učícím algoritmem backpropagation. Tento rozbor má zejména demonstrovat obrovskou rozmanitost řešení problematiky rozpoznávání písmen neuronovou sítí.

V práci uvedené v literatuře [26] se autoři zabývají rozpoznáváním ručně psaných číslic. Jako příznakový vektor použili 11 normovaných centrálních momentů ($\vartheta_{02}, \vartheta_{11}, \vartheta_{03}, \vartheta_{30}, \vartheta_{12}, \vartheta_{21}, \vartheta_{04}, \vartheta_{40}, \vartheta_{13}, \vartheta_{31}, \vartheta_{22}$), které vypočetli dle vzorce (8.1). Tento jimi navržený přístup řeší problém rozpoznávání znaků s různým poměrem stran. Parametry vícevrstvé neuronové sítě jsou: topologie 11-25-50-20-10, koeficient učení 0.7, koeficient momentu 0.9, přenosová funkce standardní sigmoida. Dosažené výsledky touto metodou: 99.97% rozpoznání tréninkové množiny a 98.73% u testovací množiny. Autoři také testovali rozšíření příznakového vektoru na 17 normovaných centrálních momentů, zlepšení v rozpoznání testovací množiny bylo velmi malé, a to 98.9%.

$$\vartheta_{pq} = \frac{u_{00}^2 \frac{p+q}{2} + 1}{u_{20}^2 \frac{p+q}{2} + u_{02}^2 \frac{p+q}{2} + 1} u_{pq} \quad (8.1)$$

Rozpoznáváním tištěných číslic a písmen (písmena malé i velké abecedy) fontu Times new roman se zabývá práce uvedená v literatuře [27]. Příznakový vektor obsahuje 62 příznaků, které reprezentují zjištěnou hodnotu histogramu pro 32 vertikálních a horizontálních vektorů z obrázku o velikosti 32x32 pixel. Nejlepšího výsledku rozpoznání testovací množiny 98.38% bylo dosaženo s topologií 64-32-62.

V práci, viz literatura [28], autoři používají standardní učící algoritmus backpropagation a vylepšený algoritmus backpropagation. Vylepšení spočívá v zamezení stavu „ustálené učení“, při kterém nemusí dojít ke správné adaptaci vah. Tento stav nastává, pokud všechny výstupní hodnoty dané vrstvy jsou blízké 0 nebo 1. Dle [28] je stav registrován pomocí dvou zavedených podmínek, pokud jsou splněny, dojde ke změně strmosti přenosové funkce standardní sigmoidy. Jako vzory byly použity ručně psaná čísla. Autoři použili několik přístupů k rozpoznávání číslic:

- Globálně propojená neuronová síť (každý neuron skryté vrstvy je spojen se všemi neurony vstupní vrstvy). Maximální hodnota rozpoznání je 98.28% pro 100 skrytých neuronů.

- Lokálně propojená neuronová síť (každý neuron skryté vrstvy je spojen se vstupními neurony vymezující určitou lokální oblast). Maximální hodnota rozpoznání je 98.88% pro 240 skrytých neuronů. Počet lokálních oblastí byl 12 (na každou oblast připojeno 20 skrytých neuronů).
- Příznaky založené na obrysu a směru zakřivení hranic. Maximální hodnota rozpoznání je 99.14% pro 100 skrytých neuronů.

Uvedené hodnoty rozpoznání testovací množiny platí pro vylepšený učicí algoritmus backpropagation. Hodnoty rozpoznání pomocí standardního algoritmu backpropagation byly vzhledem k uvedeným hodnotám rozpoznání menší, změna však byl v řádech desetin procenta.

9 ZÁVĚR

Jedním z cílů práce bylo seznámení se s teoretickými předpoklady potřebnými pro řešení problémů s využitím umělých neuronových sítí. Za základní znalost lze považovat teoretický, ale i matematický popis umělých neuronů, ze kterých se skládají veškeré neuronové sítě. Tyto teoretické znalosti byly využity při řešení praktické části práce.

V šesté kapitole byl navrhnut postup pro optimalizování parametrů dvouvrstvé a třívrstvé neuronové sítě s požadavkem na nejrychlejší naučení sítě, tedy nejmenší počet cyklů nutných k naučení. Z testování vyplývá, že navržený postup optimalizace parametrů neuronové sítě je funkční: během optimalizace parametrů docházelo ke zlepšování rychlosti učení. Jedinou výjimkou byla strmost přenosové funkce, kdy došlo k ověření, že hodnota jedna zvolená na počátku optimalizace je z hlediska rychlosti učení sítě nejvhodnější. Nejrychlejší neuronová síť získaná optimalizací má konfiguraci: topologie 7-90-60-26, interval $[-0.5, 0.5]$ pro náhodné nastavení vah, koeficient učení 1, koeficient momentu 0.3, hodnota strmosti přenosové funkce standardní sigmoidu 1, práh neuronů 1.

Během řešení optimalizace parametrů neuronové sítě, byla zjištěna závislost rychlosti učení sítí na počátečním nastavení vah sítí. Kvůli zohlednění této závislosti byly testy jednotlivých parametrů prováděny vícekrát. Velikost závislosti nebylo možné přesně určit, mimo jiné se odvíjela od nastavené kombinace ostatních parametrů sítě.

Při optimalizaci parametrů neuronové sítě bylo použito 782 neuronových sítí, kdy každá síť měla nastavenou jinou kombinací parametrů. Pokud bereme v úvahu fakt, že učení jednotlivých sítí bylo opakováno vícekrát, tak dostáváme 1852 naučených sítí, které jsou v příloze 3.

Ke zjištění závislosti rozpoznávání písmen na koeficientu učení, momentu a hodnotě strmosti přenosové funkce byly použity naučené neuronové sítě vzniklé při optimalizaci parametrů v šesté kapitole. Z výsledků je patrné, že dvouvrstvá síť se jeví jako vhodnější pro rozpoznávání písmen než třívrstvá síť. Dvouvrstvých neuronových sítí s nejlepší hodnotou rozpoznání testovací množiny 99.23% bylo téměř čtyři krát více než třívrstvých neuronových sítí, viz tabulka 7.1. Z čehož také vyplývá, že dvouvrstvá síť měla problém s rozpoznáním menšího počtu vzorů písmen testovací množiny než třívrstvá síť. Kapitola 7.2 obsahuje popis vlivu celkové chyby sítě na kvalitu rozpoznání písmen testovací množiny. V našem případě se hodnota celkové chyby sítě 0.025 jeví jako nejvhodnější hodnota pro ukončení učení.

Proces rozpoznání je také ovlivněn použitým popisem vzorů písmen tzn., jaké příznaky zjištěné z předloženého vzoru písmene jsou předkládány neuronové síti jako vstupní hodnoty. Jako popis písmen bylo zvoleno sedm momentových invariantů nezávislých na rotaci, změně měřítka a posunutí. Tyto momenty se ukázaly, jako vhodné pro popis písmen. Kvalitu rozpoznání by bylo možné vylepšit přidáním dalších

příznaků ke stávajícím sedmi momentovým invariantům, např. detektory rohů a hran, nebo některé z radiometrických příznaků (velikost, obvod atd.). Je však důležité si uvědomit, že rozšířením příznakového vektoru dojde ke zvýšení výpočetní náročnosti učení. Takže by bylo vhodné použít pouze ty příznaky, které by pomohly ke kvalitnějšímu rozpoznání písmen.

Literatura

- [1] ŠÍMA J., NERUDA R. *Teoretické otázky neuronových sítí*. 1. vyd. Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9. [cit. 2010-10-11]
- [2] MAŘÍK V., ŠTĚPÁNKOVÁ O., LAŽANSKÝ J. a kol. *Umělá inteligence 1*. 1. vyd. Praha: Academia, 1993. 264 s. ISBN 80-200-0496-3.
- [3] MAŘÍK V., ŠTĚPÁNKOVÁ O., LAŽANSKÝ J. a kol. *Umělá inteligence 4*. 1. vyd. Praha: Academia, 2003. 476 s. ISBN 80-200-1044-0. [cit. 2010-10-29]
- [4] BÍLÁ J. *Umělá inteligence a neuronové sítě v aplikacích*. 1. vyd. Praha: ČVUT, 1995. 115 s. ISBN 80-01-01275-1.
- [5] VOLNÁ E. *Neuronové sítě* 1. 2. vyd. Ostrava: Ostravská univerzita, 2008. 86 s.
- [6] ŠNOREK M. *Neuronové sítě a neuropočítače*. 1. vyd. Praha: ČVUT, 2002. 156 s. ISBN 80-01-02549-7. [cit. 2010-10-18]
- [7] JIŘINA M. *Neuronové sítě*. 1. vyd. Praha: Univerzita Karlova, 1995. 103 s.
- [8] SOBOTKA M., SOBOTKA Z. *Počítačová analýza a rozpoznávání obrazu*. 1. vyd. Praha: Dům techniky ČSVTS, 1990. 277 s. ISBN 80-02-00739-5.
- [9] HLAVÁČ V., ŠONKA M. *Počítačové vidění*. 1. vyd. Praha: Grada, 1992. 272 s. ISBN 80-85424-67-3. [cit. 2011-2-26]
- [10] KOTEK Z. *Metody rozpoznávání a jejich aplikace*. 1. vyd. Praha: Academia, 1993. 195 s. ISBN 80-200-0297-9.
- [11] HORÁK K., KÁLOVÁ I., PETYOVSKÝ P., RICHTER M.: *Počítačové vidění*. Brno: Vysoké učení technické v Brně, 2008. 132 s. [cit. 2011-3-1]
- [12] 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. [cit. 2010-11-7]
- [13] SCHALKOFF R. J. *Artificial neural network*. 1. vyd. McGraw-Hill, 1997. 421 s. ISBN 0-07-115554-6.
- [14] FAUSETT L. *Fundamentals of neural networks: architectures, algorithms and applications*. 1. vyd. Prentice-Hall, 1994. 461 s. ISBN 0-13-334186-0.
- [15] DRÁBEK O., TAUFER I., SEIDL P. *Umělé neuronové sítě – základy teorie a aplikace* (3). CHEMagazín, 2006, roč. 16, č. 1, s. 12-14. ISSN 1210-7409. [cit. 2011-1-24]
- [16] DRÁBEK O., TAUFER I., SEIDL P. *Umělé neuronové sítě – základy teorie a aplikace* (4). CHEMagazín, 2006, roč. 16, č. 2, s. 33-36. ISSN 1210-7409.
- [17] DRÁBEK O., TAUFER I., SEIDL P. *Umělé neuronové sítě – základy teorie a aplikace* (6). CHEMagazín, 2006, roč. 16, č. 6, s. 31-33. ISSN 1210-7409.
- [18] HU M. K. *Visual pattern recognition by moment invariants*. In *IRE transactions on information theory*, 1962, roč. 8, č. 2, s. 179-187. [cit. 2011-2-18]
- [19] KUKAL J. *Úvod do neuronových sítí*. Automa [online]. 2005, roč. 11, č. 1 [cit. 2010-10-17]. ISSN 1210-9592. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30255.
- [20] KUPČA V. *Neuron živého organismu* [online]. Poslední revize 13.3.2003 [cit. 2010-10-10]. Dostupné z: <http://neuron.felk.cvut.cz/courseware/data/chapter/36nan026/s01index.html>.

- [21] Neuronové sítě – referát k předmětu Praktikum z programování MFF CUNI [online].
Poslední revize 26.7.2005 [cit. 2011-1-26]. Dostupné z:
<<http://cgg.mff.cuni.cz/~pepca/prg022/mucha/>>.
- [22] Principles of training multi-layer neural networks using Backpropagation [online].
Poslední revize 1.6. 2008 [cit. 2010-10-15]. Dostupné z:
<http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html>.
- [23] The Back Propagation Algorithm [online]. Poslední revize 18.10.2005 [cit. 2010-1-11].
Dostupné z:
<<http://www4.rgu.ac.uk/files/chapter3%20-%20bp.pdf>>.
- [24] SHUTLER J. Statistical moments [online]. Poslední revize 15.1.2004 [cit. 2011-2-25].
Dostupné z:
<http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/CVonline_moments.html>.
- [25] RUMELHART D. E., HINTON G. E., WILLIAMS R. J. Learning internal representations by error propagation. In *Rumelhart D. E., McClelland J. L. Parallel Distributed Processing*. Cambridge, MIT Press: MA, 1986, roč. 1, s. 318-362. [cit. 2011-4-10]
- [26] PAN F., KEANE M. A new set of moment invariants for handwritten numeral recognition. In *Image processing, 1994*. 1994, roč. 1, s. 154-158. [cit. 2011-4-29]
- [27] MANI N., VOUMARD P. An optical character recognition using artificial neural network. In *Systems, man, and cybernetics, 1996*. 1996, roč. 3, s. 2244-2247. [cit. 2011-4-29]
- [28] YAMADA K., KAMI H., TSUKUMO J., TEMMA T. Handwritten numeral recognition by multilayered neural network with improved learning algorithm. In *Neural networks, 1989*. 1989, roč. 1, s. 259-266. [cit. 2011-4-29]

Seznam symbolů

ξ	vnitřní potenciál neuronu
h	práh neuronu
η	koeficient učení
η_{OPT}	optimální hodnota koeficientu učení
α	koeficient momentu
$\underline{x}$	vstupní vektor $\underline{x} = (x_1, \dots, x_i, \dots, x_n)$
$\underline{s}$	vstupní tréninkový vektor $\underline{s} = (s_1, \dots, s_i, \dots, s_n)$
$\underline{t}$	požadované výstupy neuronové sítě $\underline{t} = (t_1, \dots, t_k, \dots, t_m)$ při vstupním tréninkovém vektoru $\underline{s}$
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, $k = 1, \dots, m$
δ_k	částečné váhové korekce příslušející chybě spojení mezi neurony skryté a výstupní vrstvy
δ_j	částečné váhové korekce příslušející chybě spojení mezi neurony vstupní a skryté vrstvy
$w_{ij} (w_{jk})$	váhová hodnota spoje mezi neurony X_i a Z_j (Z_j a Y_k)
$w_{0j} (w_{0k})$	práh neuronu Z_j (Y_k)
E_r	chyba sítě aktuálního tréninkového vzoru, $r = 1, \dots, v$
E_C	celková chyba sítě
m_{pq}	obecný moment
u_{pq}	centrální moment
ϑ_{pq}	normovaný centrální moment
$\varphi_1 - \varphi_7$	sedm momentových invariantů

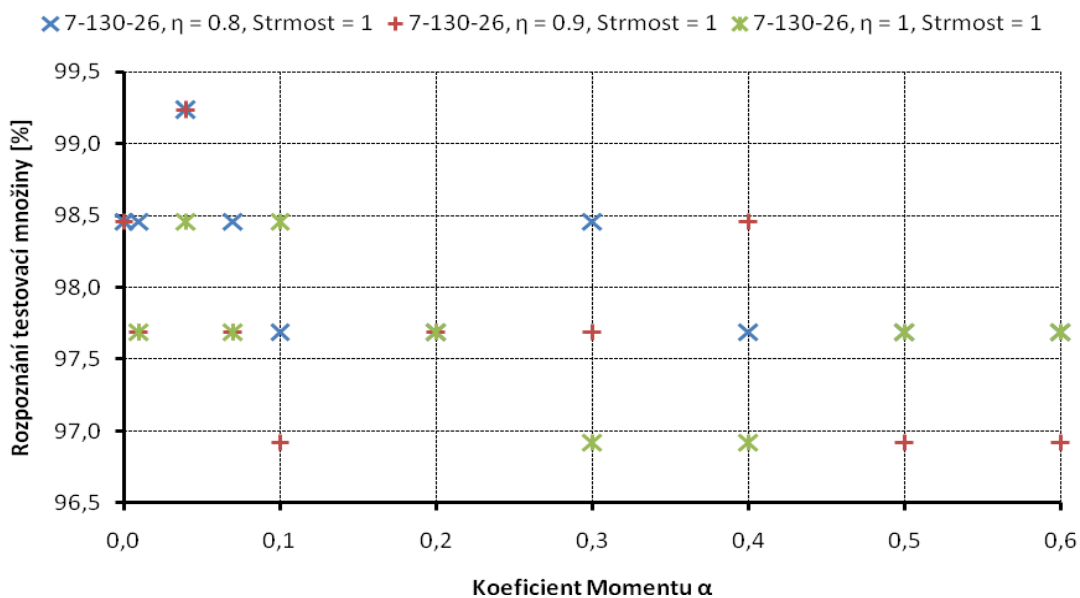
Seznam příloh

- Příloha 1. Graf znázorňující závislost rozpoznání testovací množiny dvouvrstvou sítí na velikosti koeficientu momentu
- Příloha 2. Graf znázorňující závislost rozpoznání testovací množiny třívrstvou sítí na velikosti koeficientu momentu
- Příloha 3. CD obsahující:
- elektronickou verzi bakalářské práce
 - vzory tvořící tréninkovou i testovací množinu
 - tréninková a testovací množina (Matlab R2009b)
 - vytvořené programy (Matlab R2009b)
 - uložené všechny testované sítě (Matlab R2009b)
 - zpracované výsledky testování (Excel 2003)

10 PŘÍLOHY

Příloha 1.

Graf znázorňující závislost rozpoznání testovací množiny dvouvrstvou sítí na velikosti koeficientu momentu.



Příloha 2.

Graf znázorňující závislost rozpoznání testovací množiny třívrstvou sítí na velikosti koeficientu momentu

