

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

Rozoznávanie písmen pomocou neurónovej siete

BAKALÁRSKA PRÁCA

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FRANTIŠEK KLUKNAVSKÝ

BRNO 2008

Abstrakt

Práca sa na úlohe rozpoznávania rukou písaných písmen zaoberá implementáciou viacvrstvovej perceptrónovej siete, učením metódou spätného šírenia chyby, hľadáním ich optimálnych parametrov, šírkou skrytej vrstvy, rýchlosťou a dĺžkou učenia, zvládaním poškodených dát. Výsledky vznikli opakovaným simulovaním a testovaním neurónovej siete použitím 52 152 malých písmen anglickej abecedy. Najlepšie výsledky pri čo najmenšej sieti a najkratšom čase tréningu dosiahla sieť so 60 neurónmi v skrytej vrstve a učenie rýchlosťou 0,01. Siete so širšou skrytou vrstvou dosiahli približne rovnakú úspešnosť pri testoch na neznámych písmenách, ale vyššiu úspešnosť na silne poškodených písmenách.

Abstract

This work uses handwritten character recognition as a model problem for using multilayer perceptron, error backpropagation learning algorithm and finding their optimal parameters, hidden layer size, learning rate and length, ability to handle damaged data. Results were acquired by repeated simulation and testing the neural network using 52,152 English lowercase letters. Best results, smallest network and shortest learning time was at 60 neurons in the hidden layer and learning rate of 0.01. Bigger networks achieved the same ability to recognize unknown patterns and higher robustness at highly damaged data processing.

Kľúčové slová

neurón, neurónová sieť, písmeno, preučenie, skrytá vrstva, ocr

Keywords

neuron, neural network, letter, overfitting, hidden layer, ocr, backpropagation, learning rate

Citácia

František Kluknavský: Rozoznávanie písmen pomocou neurónovej siete, bakalárska práca, Brno, FIT VUT v Brne, 2008

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Jany Šilhovej. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

Ďakujem vedúcej Ing. Šilhovej za to, že mi umožnila zoznámiť sa a pracovať s touto zaujímavou témou aj za všetku pomoc.

Obsah

1 Úvod.....	5
2 Motivácia a použitá metodika.....	5
3 Rozpoznávanie vzoriek.....	7
3.1 Systémy rozpoznávania vzoriek	7
3.2 Klasifikácia vzoriek	8
3.3 Klasifikácia neurónovou sieťou.....	9
4 Neurónové siete.....	9
4.1 Architektúry neurónových sietí.....	9
4.2 Učenie neurónových sietí.....	10
4.3 Perceptrón.....	10
4.3.1 Algoritmus učenia perceptrónu.....	13
4.4 Viacvrstvový perceptrón.....	14
4.4.1 Architektúra.....	14
4.4.2 Algoritmus učenia.....	15
5 Rozpoznávanie znakov pomocou viacvrstvého perceptrónu.....	17
5.1 Experimentálny systém pre rozpoznávanie znakov.....	18
5.2 Úprava vstupných dát.....	18
5.3 Architektúra neurónovej siete.....	20
5.4 Implementácia neurónovej siete.....	21
5.5 Postup testovania neurónovej siete.....	24
6 Experimentálne výsledky.....	25
7 Záver.....	30
8 Zoznam použitých zdrojov.....	30
9 Zoznam použitých skratiek a symbolov.....	31

1 Úvod

Neurónové siete (ďalej len NN) patria do oblasti umelej inteligencie zvanej soft computing, ktorej cieľom je prispôbiť sa nepresnosti a neistote reálneho sveta. Ako induktívny systém sa učia z príkladov a objavujú vzťahy medzi nimi. To ich predurčuje na riešenie problémov, ktoré sa nedajú jednoducho matematicky popísať, ale vyžadujú prístup podobný ľudskej mysli - pozrieť, porovnať a odhadnúť. Podobnosť so živým mozgom je aj v štruktúre. Skladajú sa z jednoduchých výpočtových jednotiek, neurónov, ktoré spolu komunikujú po rôzne silných väzbách, synapsách. Biologické aj umelé neuróny počítajú vážený súčet svojich vstupov, aplikujú na neho jednoduchú funkciu a výsledok šíria k ďalším neurónom. Učenie spočíva v zmene váh väzieb. Rozdiel je hlavne v mechanizme týchto zmien. V živých nervových centrách ešte neboli ani zďaleka objasnené. Možno raz pokrok v neurológii pomôže zdokonaľiť umelé neurónové siete, alebo naopak, umelá inteligencia prispeje k odhaleniu analogických procesov v organizmoch.

Rozoznávanie písma je typickou úlohou vhodnou pre NN. Je jednoduchšie predložiť niekoľko ukážok písmena a spoľahnúť sa na "inteligenciu" žiaka než presne definovať kritérium príslušnosti k danému písmenu. Žiak si potom musí poradiť aj s obrazmi, ktoré nikdy predtým nevidel (a v prípade človeka - ktoré by sám ani nikdy nevymyslel).

2 Motivácia a použitá metodika

V súčasnosti najpopulárnejší model neurónovej siete je viacvrstvová perceptrónová sieť učená metódou spätného šírenia chyby (error backpropagation). Jej nevýhodou je veľká výpočtová a tým aj časová náročnosť. Problém je ešte umocnený tým, že zatiaľ nie je jednoduchý postup, ako určiť najlepšie parametre siete. Tie sa musia zistiť opakovanými pokusmi.

Cieľom tejto práce je otestovať na modelovej úlohe vhodnosť použitia NN a nájdenie jej najlepších nastavení, ktoré by sa dali použiť ako rámcový odhad pre budúce podobné úlohy.

Na základe pokynov zadania bakalárskej práce boli sformulované konkrétne ciele pre jej realizáciu:

- navrhnuť a vytvoriť experimentálny systém pre rozpoznávanie vzorov na báze viacvrstvého perceptrónu
- aplikácia systému na rozpoznávanie rukou písaných písmen

- optimalizácia architektúry NN s cieľom minimalizácie počtu neurónov
- analýza vplyvu šumu a posuvu na úspešnosť rozpoznávania vzorov
- návrh riešení pre zlepšenie úspešnosti rozoznávania

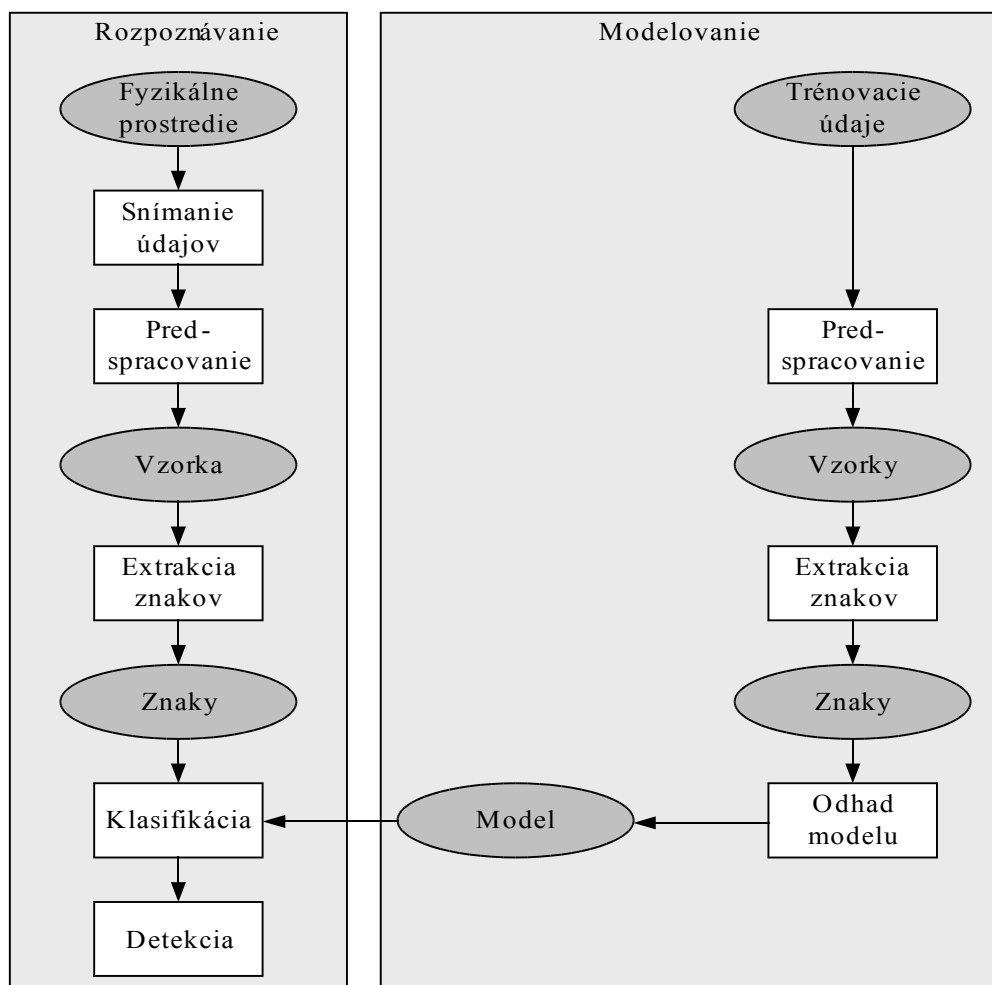
S cieľom splniť úlohu v určenom čase bol zvolený nasledovný postup:

- Naštudovanie problematiky rozpoznávania vzorov so zameraním na oblasť neurónových sietí. Výsledkom štúdia je súbor informácií o NN, ktoré umožnili implementáciu experimentálneho systému pre rozpoznávanie znakov.
- Výber vhodného súboru pre demonštráciu rozpoznávania vzorov. Pre túto úlohu bol zvolený súbor písaných znakov anglickej abecedy, pretože nevyžaduje predspracovanie interpunkčných znamienok a má veľa znakov spoločných s abecedami iných národov. Svojou variabilitou lepšie otestuje robustnosť systému než tlačené písmená.
- Návrh štruktúry systému pre rozoznávanie obrazcov umožňujúceho učenie, testovanie, verifikáciu a analýzu použitých neurónových sietí.
- Voľba implementačného prostredia a implementácia systému. Hlavný dôraz je pri tom kladený na generičnosť modelu neurónovej siete a prenositeľnosť vyvinutého softvéru.
- Experimentálna analýza vlastností implementovanej NN. Použitím vybraných súborov obrazcov zrealizovať tréning siete a rozpoznávanie vzorov v závislosti od počtu neurónov v skrytej vrstve.
- Modifikácia súboru vzorov za účelom analýzy vplyvu šumu a posunutia na úspešnosť rozoznávania.
- Analýza a diskusia dosiahnutých výsledkov, definovanie otvorených problémov, navrhnutie riešení zlepšenia systému.

3 Rozpoznávanie vzoriek

3.1 Systémy rozpoznávania vzoriek

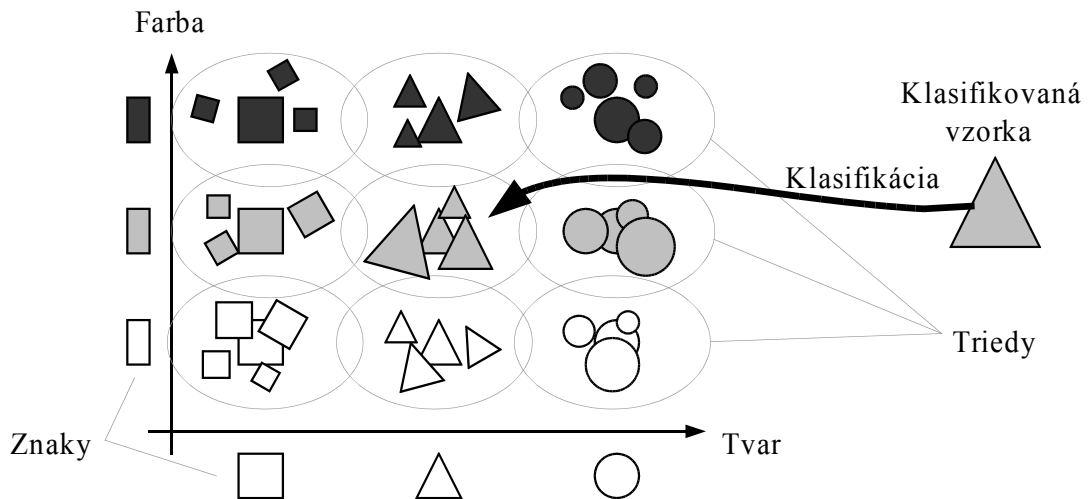
Rozpoznávanie vzoriek sa spravidla vykonáva na základe signálov pochádzajúcich z reálneho fyzikálneho prostredia. Principiálna bloková schéma systému pre rozpoznávanie vzoriek je na obrázku 1 vľavo. Zosnímané údaje sú často v dôsledku skreslenia nevhodné pre automatizované spracovanie, preto je potrebné údaje predspracovať napr. korigovať, potlačiť šum atď. a následne selektovať vzorku. Predspracované údaje reprezentujú vzorku a umožňujú spoľahlivú extrakciu jej znakov. Na základe získaných znakov sa použitím vopred definovaného modelu realizuje klasifikácia t.j. odhad miery príslušnosti vzorky k určitej triede a následnou detekciou sa začlení do najpravdepodobnejšej triedy. Úspešnosť procesu rozpoznávania je okrem kvality znakov a zvoleného klasifikačného algoritmu závislá aj od reprezentatívnosti definovaného modelu. Model určuje triedy na základe kombinácie znakov a vymedzuje presné hranice medzi nimi. Vymedzenie hraníc je dôležitý predpoklad pre úspešnú klasifikáciu vzoriek. Model sa odhaduje použitím súboru trénovacích znakov získaných z vhodne zvolených trénovacích údajov (obrázok 1 vpravo).



Obrázok 1: Schéma systému pre rozpoznávanie vzoriek

3.2 Klasifikácia vzoriek

Vzorka (tiež - vzor, obrazec , angl. pattern) reprezentuje vstupný – analyzovaný – signál, ktorý obsahuje spravidla skreslený objekt záujmu. Objekt je z pohľadu pozorovateľa charakterizovaný rôznymi vlastnosťami (znakmi) napr. tvarom, veľkosťou, farbou, umiestnením, orientáciou, štruktúrou atď. Objekty je možné rozčleniť na základe ich charakteristických vlastností do rôznych tried. Cieľom procesu rozpoznávania vzoriek je na základe charakteristických vlastností objektov umožniť ich rozpoznanie a klasifikáciu t.j. určenie ich príslušnosti k jednej z definovaných tried. Napríklad na obrázku 2 sú objekty (vzorky) klasifikované do deviatich tried na základe dvoch vlastností tvar={kruh, štvorec, trojuholník} a farba={červená, modrá, zelená}. Úlohou klasifikačného algoritmu je na základe identifikovaných vlastností (tvar, farba) určiť mieru príslušnosti vzorky k triede.



Obrázok 2: Znáznornenie procesu klasifikácie vzoriek.

3.3 Klasifikácia neurónovou sieťou

NN v procese učenia sama na základe trénovacích vzoriek odhaľuje dôležité vzťahy modelu. Napriek tomu môže byť výhodné vstupné dáta do nejakého stupňa predspracovať a extrahovať z nich vlastnosti, napríklad koeficienty Fourierovej alebo kosínusovej transformácie, počet hrán, rohov a podobne. Údaje získané v čase sa často akumulujú, v našom prípade sa z ťahov pera získa obrázok písmena.

4 Neurónové siete

Umelá neurónová sieť je abstraktný počítačový model mozgu. Skladá sa z umelých neurónov a zo spojení medzi nimi - synáps. Zobrazuje sa najčastejšie ako ohodnotený orientovaný graf, kde neuróny sú uzly a synapsy hrany. Informácia je uložená v ohodnotení (váhe) hrán. Hlavné rozdiely medzi rozličnými druhmi sietí sú vo funkcii, ktorú počítajú neuróny, v organizácii spojení medzi nimi a v algoritme učenia.

4.1 Architektúry neurónových sietí

Plne prepojená NN má výstup z každého neurónu do každého neurónu vrátane seba samého.

Ostatné architektúry sa dajú odvodiť z tejto nastavením niektorých váh na nulu. Nerekurentná NN neobsahuje spätnú väzbu. Graf neobsahuje žiaden kruh, signál z neurónu sa už do neho nevracia priamo ani prostredníctvom iných neurónov. Takáto sieť býva vnútorne organizovaná do vrstiev, kde výstupy neurónov jednej vrstvy slúžia ako vstupy ďalšej vrstvy. Rekurentná sieť má principiálne väčšie možnosti. Existujú problémy, ktoré sa dajú ľahko riešiť malými systémami so spätnou väzbou, ale ekvivalentné nerekurentné systémy musia byť veľmi veľké až teoreticky nekonečné. Dá sa dosiahnuť aj pamäťový efekt, keď výstup nezávisí len od aktuálneho vstupu, ale aj od vnútorného stavu, t.j. od predchádzajúcich vstupov.

4.2 Učenie neurónových sietí

Podľa spôsobu učenia sa umelé neurónové siete dajú rozdeliť na dve kategórie. Pevné NN nikdy nemenia váhy medzi neurónmi. Určia sa vopred na základe charakteru riešenej úlohy. Adaptívne NN majú schopnosť meniť nastavenie váh v čase. Dve základné skupiny metód učenia adaptívnych sietí sú učenie s učiteľom a učenie bez učiteľa.

Učenie s učiteľom (supervised learning) znamená, že ku každému vstupu je známy správny výstup. Sieť sa učí tak, že svoj výstup sa snaží priblížiť správnemu. Príkladom je metóda spätného šírenia chyby. Učenie a používanie NN sú dve oddelené fázy.

Pri učení bez učiteľa (unsupervised learning) alebo samoorganizácii okrem vstupu nie sú žiadne ďalšie informácie. Sieť sa sama snaží zapamätať, klasifikovať a podobne. Spravidla sa nedá rozlíšiť fáza učenia a fungovania, prebiehajú súčasne.

Existujú aj medzistupne, napríklad systém odmena-trest, kde dodatočná informácia hovorí, či bola odpoveď správna alebo nesprávna.

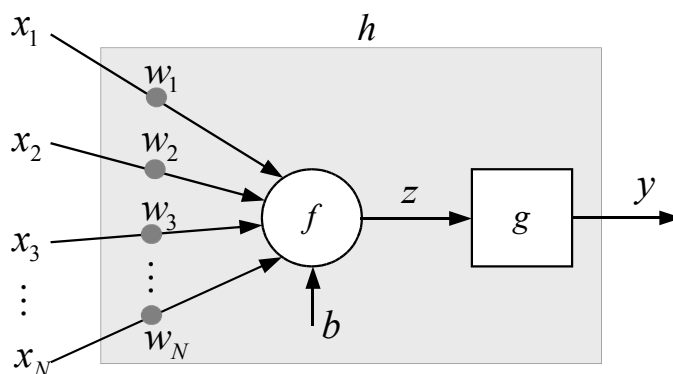
Zvláštnym javom je preučenie (pretrénovanie, overfitting). Keď je veľká neurónová sieť príliš dlho trébovaná, postupne dosiahne výbornú úspešnosť na trébovacích dátach, ale na nových neznámych vzorkách začne čoraz viac chybovať. Stráca schopnosť generalizácie.

4.3 Perceptrón

Perceptrón je model nervovej bunky tzv. neurónu (obrázok 3) a predstavuje najjednoduchšiu formu neurónovej siete. Vstupný signál perceptrónu je tvorený vektorom reálnych čísel

$x = [x_1, x_2, x_3, \dots, x_N]$, ktoré sú nosičmi signálov z výstupov iných perceptrónov. Výstupný

signál perceptrónu y má výrazne bimodálny charakter a vyjadruje aktivitu alebo pasivitu perceptrónu. Keď suma váhovaných príspevkov od ostatných perceptrónov prekročí istú hodnotu, nazývanú prah excitácie, perceptrón vygeneruje výstupný impulz. V opačnom prípade je výstup perceptrónu neaktívny.



Obrázok 3: Všeobecná schéma perceptrónu s N vstupmi, bázovou funkciou f a aktivačnou funkciou g .

Funkciu perceptrónu je možné opísať funkciou $h(\cdot)$ vzťahom

$$y = h(x) \quad (1)$$

pričom

$$h(x) = g(f(x)), \quad y = g(z), \quad z = f(x) \quad (2)$$

Perceptrón teda pozostáva z:

- *bázovej funkcie* $f(\cdot)$ realizujúcej váhovaný súčet prvkov vstupného vektora x

$$z = f(x) = x \cdot w - b = \sum_{i=1}^N x_i w_i - b \quad w = [w_1, w_2, \dots, w_N], \quad b \in \mathbb{R} \quad (3)$$

kde offset b predstavuje aktivačný prah perceptrónu, w je vektor synaptických váh, ktorého prvky sú asociované s prvkami vstupného vektora x ,

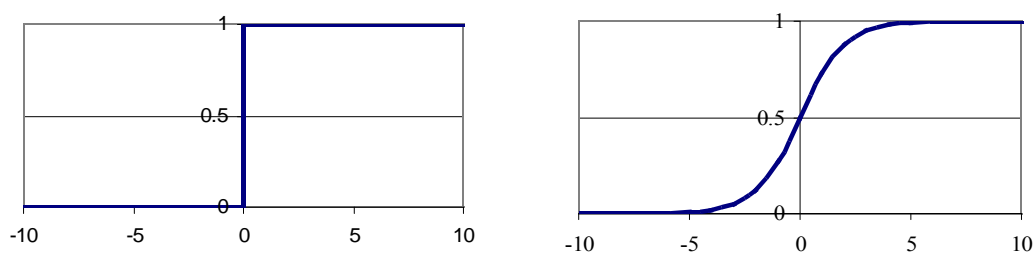
- *aktivačnej funkcie* $g(\cdot)$ produkujúcej výstupný signál y . Najčastejšie používanými aktivačnými funkciami sú lineárna funkcia, funkcia signum (znamienko) alebo

sigmoidálna funkcia. V prípade lineárnej funkcie (4) je výstupný signál y priamo úmerný signálu skalárneho produktu z . Za týchto podmienok predstavuje celá sieť lineárny systém, čo uľahčuje pochopiť jej chovanie a analýzu. Na druhej strane, nelineárne aktivačné funkcie sa viac približujú funkcii biologických neurónov a majú širšie výpočtové schopnosti. Kým výstup funkcie signum je binárny (5), výstup funkcie sigmoid (6) sa mení kontinuálne (a nelineárne) v závislosti od vstupu, čo sa najviac približuje chovaniu reálneho neurónu. Priebehy aktivačných funkcií signum a sigmoid sú znázornené na obrázku 4.

$$y = g(z) = kz \quad k \in \mathbb{R} \quad (4)$$

$$y = g(z) = \text{sgn}(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad (5)$$

$$y = g(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

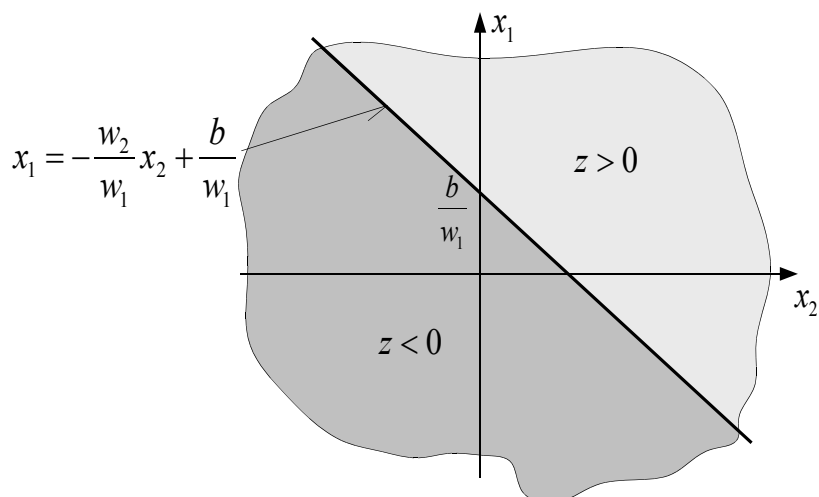


Obrázok 4: Príklady nelineárnych unipolárnych aktivačných funkcií: a) signum , b) sigmoid

Aktivačné funkcie sú anti-symetrické voči bodu $x=0$, ktorý je bodom rozhodovania aktivačnej funkcie. Znamená to teda, že ak je výstup bázeovej funkcie $f=0$, neurón sa nachádza na hranici rozhodovania t.j. výstup neurónu je neurčitý. Teda vzťah

$$z = 0 \quad \rightarrow \quad \sum_{i=1}^N x_i w_i - b = 0 \quad (7)$$

je rovnicou roviny v N -rozmernom priestore, pričom váhový vektor w predstavuje koeficienty deliacej roviny rozdeľujúcej priestor na dva podpriestory pre $z > 0$ a $z < 0$. Z toho vyplýva, že perceptrón vie klasifikovať vstupné vektory do dvoch tried oddelených deliacou rovinou. Táto vlastnosť sa označuje ako schopnosť riešiť tzv. *lineárne separovateľné problémy*. Príklad rozdelenia dvojrozmerného priestoru ($N=2$) na dva polpriestory je na obrázku 5. Deliaci priamka reprezentuje hranicu rozhodovania perceptrónu.



Obrázok 5: Príklad rozdelenia dvojrozmerného priestoru ($N=2$) na dva podpriestory (reprezentujúce dve triedy perceptrónu) prostredníctvom priamky $w_1 x_1 + w_2 x_2 - b = 0$

4.3.1 Algoritmus učenia perceptrónu

Vychádzajúc z princípu fungovania perceptrónu opísanom v predchádzajúcej časti (obrázok 5), proces učenia perceptrónu je vlastne na základe reprezentatívneho tréningového súboru vzoriek určenie optimálnych váh $w_{i,opt}, i=1,2,\dots,N$ (a prahu b) tak, aby minimalizovali celkovú chybu rozhodovania. Základné pravidlo učenia perceptrónu je:

$$w_i' = w_i + \eta \cdot (y - d) \cdot x_i \quad (8)$$

Kde w_i' je nová hodnota váhy synapsy i ,

w_i je pôvodná hodnota váhy,

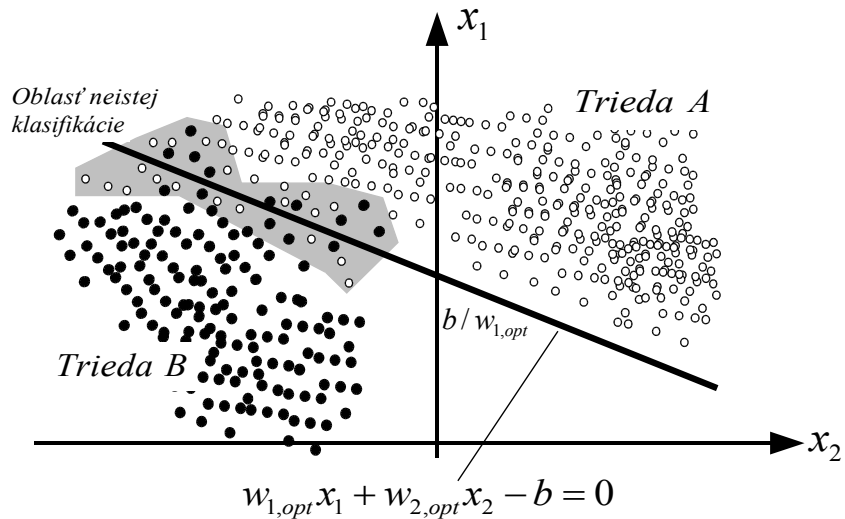
x_i je vstupný signál do synapsy,

y je výstup perceptrónu,

d je očakávaný správny výstup,

η je konštanta udávajúca rýchlosť učenia.

Príklad optimálneho rozdelenia dvojrozmerného priestoru ($N=2$) do tried A a B je na obrázku číslo 6. V dôsledku prekrývania oblastí so vzorkami z rôznych tried nie je možná ich úplná separácia. To spôsobuje chybu klasifikácie u vzoriek v oblasti hranice.



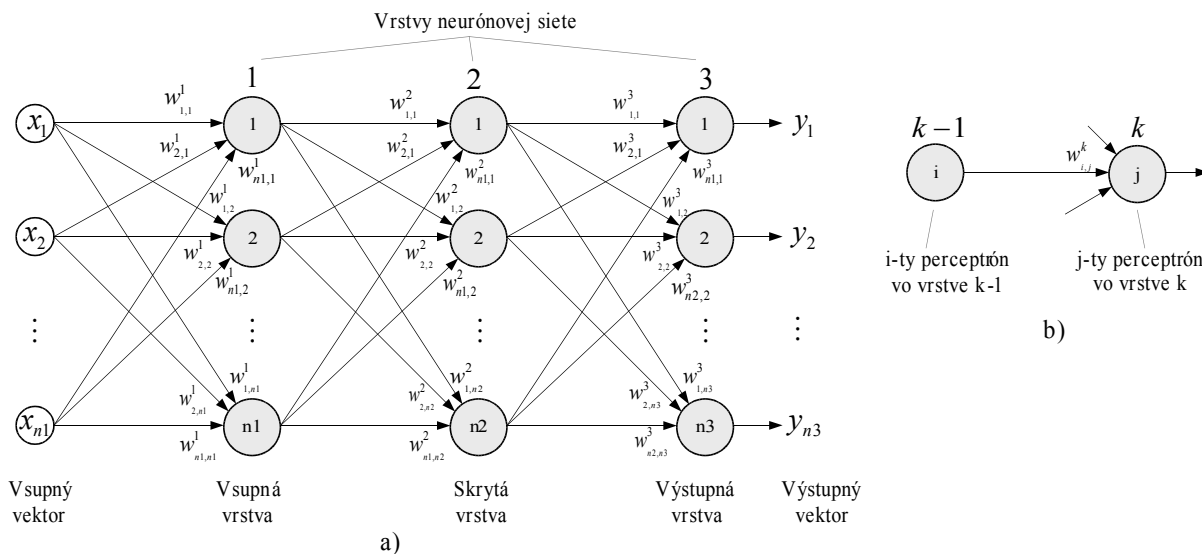
Obrázok 6: Príklad optimálneho rozdelenia dvojrozmerného priestoru do tried A a B prostredníctvom optimálnych váh $w_{1,opt}$, $w_{2,opt}$

4.4 Viacvrstvový perceptrón

4.4.1 Architektúra

Určitý počet perceptrónov vedľa seba vytvorí vrstvu. Perceptróny v jednej vrstve prijímajú rovnaké vstupy, ale každý z nich pracuje nezávisle na ostatných.

Niekoľko vrstiev za sebou je prepojených tak, že výstupy jednej slúžia ako vstupy ďalšej. Prvá vrstva, ktorá prijíma signály z okolia, sa nazýva vstupná. Posledná (výstupná) vrstva dáva žiadané výsledky.



Obrázok 7: Schéma viacvrstvového perceptrónu s jednou vstupnou, jednou výstupnou a jednou skrytou vrstvou (a). Spôsob označenia synaptických váh (b).

4.4.2 Algoritmus učenia

Pri tréovaní NN sa priloží učiaci vzor x na vstup siete a výstupný vektor y (odozva siete) sa porovnáva so žiadaným vektorom d . Chyba medzi učiacim a žiadaným vektorom je daná

$$E = \frac{1}{2} \sum_n (y_n - d_n)^2 \quad (9)$$

Učenie siete spočíva v postupnom modifikovaní váh $w_{i,j}$ tak, aby sa minimalizovala celková chyba E . Modifikáciu váh vypočítame podľa

$$w_{i,j}(m+1) = w_{i,j}(m) + \Delta w_{i,j}(m) \quad (10)$$

pričom m je index tréovacej iterácie (epochy) a $\Delta w_{i,j}$ je prírastok váhy daný vzťahom

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}} \quad (11)$$

Kde η je faktor určujúci rýchlosť učenia (rýchlosť zmeny $w_{i,j}$) a parciálna derivácia vyjadruje vplyv váhy $w_{i,j}$ na celkovú chybu. Princíp algoritmu je na obrázku 8, ktorý znázorňuje výstupnú vrstvu perceptrónov vo viacvrstvovej neurónovej sieti. Základná

myšlienka je v rekurzívnom výpočte prírastkov chýb Ey_j , Ez_j , Ew_j , Ey_i na celkovú chybu, pričom už vypočítaný prírastok chyby je použitý pre výpočty nasledujúcich prírastkov (spätne šírenie chyby).

Algoritmus spätneho šírenia chyby pozostáva zo štyroch krokov:

1. Výpočet prírastku chyby Ey_j od výstupu y_j po zohľadnení (9)

$$Ey_j = \frac{\partial E}{\partial y_j} = y_j - d_j \quad (12)$$

2. Výpočet prírastku chyby Ez_j od vstupu z_j

$$Ez_j = \frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j} \times \frac{\partial y_j}{\partial z_j} = Ey_j \cdot y_j(1 - y_j) = (y_j - d_j) y_j(1 - y_j) \quad (13)$$

pričom zo (6) vyplýva

$$\partial y_j / \partial z_j = y_j(1 - y_j) \quad (14)$$

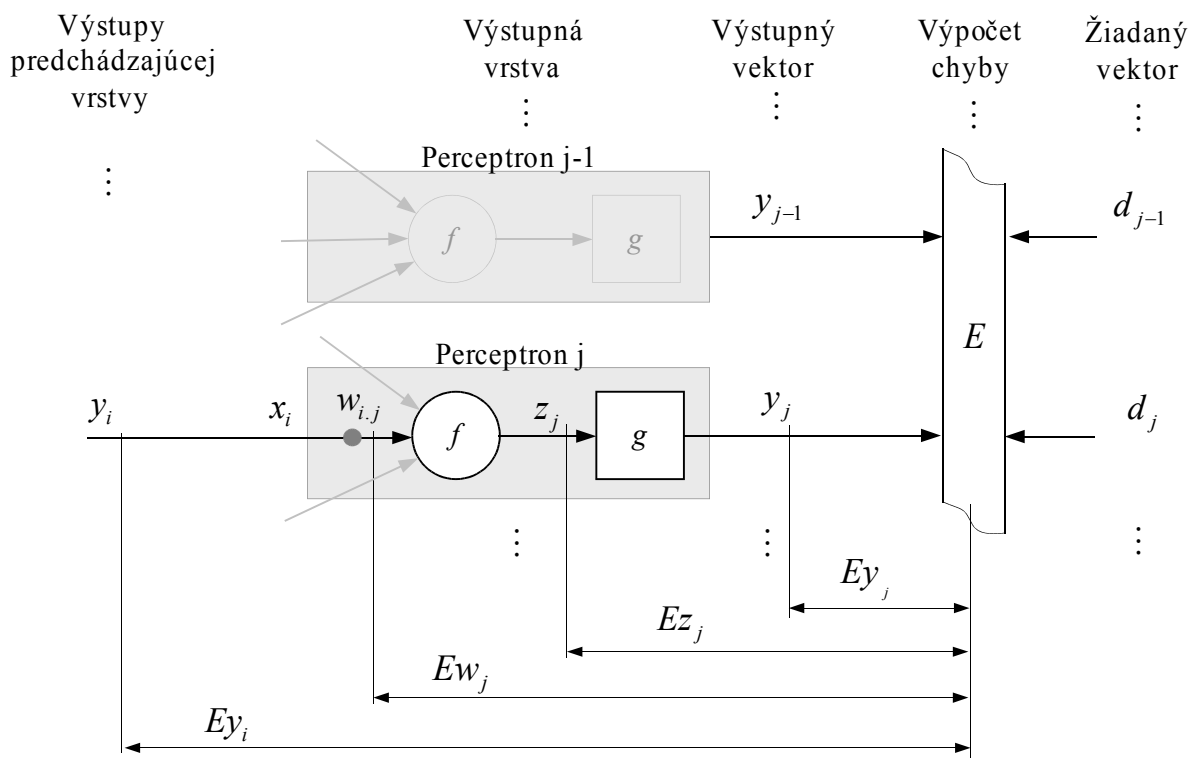
3. Výpočet prírastku chyby $Ew_{i,j}$ od váhy $w_{i,j}$

$$Ew_{i,j} = \frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial z_j} \times \frac{\partial z_j}{\partial w_{i,j}} = Ez_j \cdot y_j \quad (15)$$

4. Výpočet prírastku chyby Ey_i od výstupu predchádzajúcej vrstvy y_i

$$Ey_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial z_j} \times \frac{\partial z_j}{\partial y_i} = \sum_i Ez_j \cdot w_{i,j} \quad (16)$$

Krok 4 umožňuje aplikovať algoritmus rekurzívne na predchádzajúce vrstvy až po vstup neurónovej siete. Podľa kroku 3 sa postupne vypočítajú prírastky váh $Ew_{i,j} = \partial E / \partial w_{i,j}$ a dosadia do (11) pre aktualizovanie váh.



Obrázok 8: Znáznornenie princípu algoritmu učenia so spätným šírením chyby

5 Rozpoznávanie znakov pomocou viacvrstvého perceptrónu

Z voľne dostupných súborov znakov bol pre účely experimentovania s implementovanou neurónovou sieťou zvolený súbor znakov výskumnej skupiny MIT Spoken Language Systems Group [5]. Použité dáta sú v tvare textového súboru, kde každý riadok obsahuje údaje o jednom písmene. Okrem obrazovej podoby písmena zahŕňa správne určenie písmena ľudským čitateľom a ďalšie informácie ako napríklad príslušnosť k nejakému slovu a umiestnenie v slove. Tieto dodatočné údaje sa nepoužili.

Obrazová podoba písmena je vo forme rastrového obrázka rozmerov 8x16 bodov (obrázok 11a). Vznikla segmentáciou z ručne písaného anglického textu mnohých pisateľov. Znaky sú normalizované na jednotnú veľkosť. Každý pixel môže nadobúdať jednu z dvoch hodnôt 0

alebo 1 (biela alebo čierna). Celkový počet vzoriek je 52 152. Jednotlivé písmená sú zastúpené takmer rovnomerne. Sada obsahuje len malé písmená.



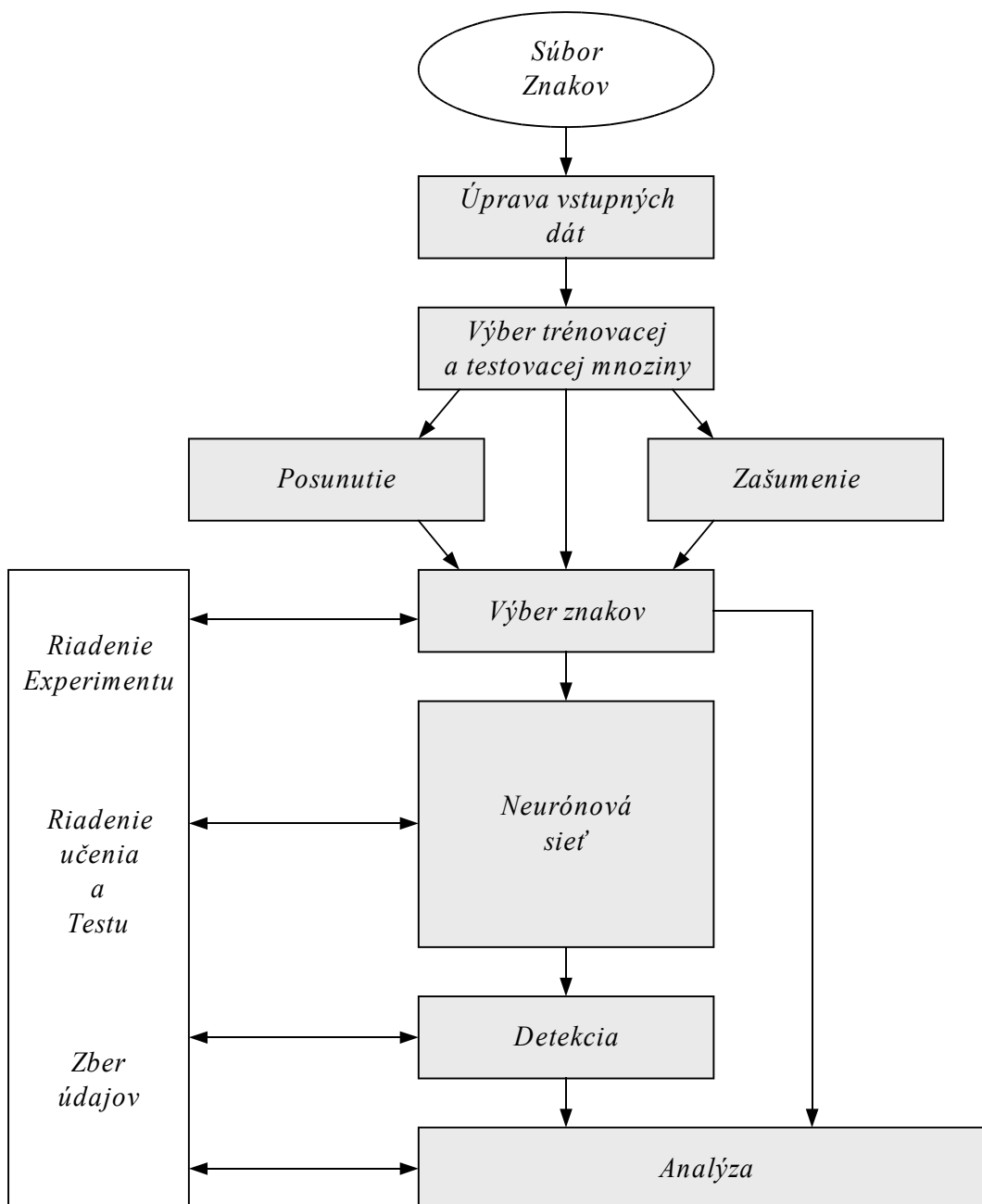
Obrázok 9: Ukážka použitej sady písmen

5.1 Experimentálny systém pre rozpoznávanie znakov

Súčasťou tejto práce bolo navrhnutie a implementácia experimentálneho systému na rozpoznávanie znakov na báze viacvrstvého perceptrónu. Skladá sa z niekoľkých základných častí. Princiálna bloková schéma systému je na obrázku 10.

5.2 Úprava vstupných dát

- Odfiltrovanie nepoužívaných údajov.
- Zmena binárneho kódovania farieb na bipolárne hodnoty -0,5 a 0,5, ktoré sa potom použijú ako vstupy neurónovej siete. Dôvodom sú osobitné vlastnosti nuly v matematických operáciách sčítania a násobenia, ktoré nepriaznivo pôsobia na výpočtové schopnosti siete.
- Zmena ASCII kódovania správnej klasifikácie písmena na kódovanie 1 z N v 26-miestnom vektore. Keďže použitá aktivačná funkcia nikdy nedosiahne hraničné hodnoty 0 a 1, nula sa zakóduje ako 0,2 a jednotka ako 0,8.
- Prevod na formát bmp, aby bolo pre človeka jednoduché prezrieť si použité vzorky v obrazovej podobe pomocou ľubovoľného prehliadača obrázkov.



Obrázok 10: Bloková schéma experimentálneho systému na rozpoznávanie znakov

Vytvorenie neurónovej siete

Váhy sa inicializujú náhodne s rovnomerným rozložením pravdepodobnosti v intervale $(-0,5; 0,5)$. Je nutný aspoň trochu kvalitný generátor, pretože pravidelnosti v nastavení váh sa počas učenia zvyknú zachovávať a tak obmedzujú možnosti siete.

Interpretovanie výstupu

Pre zvláštne tvary písmen nie je odozva siete jednoznačná. Konečné rozhodnutie o príslušnosti ale je iba jedno. Percentuálne pravdepodobnosti jednotlivých písmen môžu byť zaujímavé, ale sú ďaleko od praktického použitia. Najvyššie číslo vo výstupnom vektore sa interpretuje ako jednotka a všetky ostatné ako nuly.

Simulovanie, tréovanie a testovanie

Sú popísané v osobitných kapitolách.

5.3 Architektúra neurónovej siete

Neurón

Použil som neuróny s lineárnou bázovou funkciou a sigmoidálnou aktivačnou funkciou. Výstup každého neurónu je vážený súčet jeho vstupov, na ktorý aplikuje funkciu

$f(x) = \frac{1}{1 + e^{-x}}$. Táto funkcia je ohraničená a diferencovateľná, čo sa využije pri učení.

Jej derivácia je $f'(x) = f(x) \cdot (1 - f(x))$.

Vzťahy medzi neurónmi

Viacvrstvová dopredná sieť. Neurón je spojený so všetkými výstupmi neurónov predchádzajúcej vrstvy a jeho výstup ide ku všetkým neurónom nasledujúcej vrstvy. Túto architektúru som zvolil pre jej univerzalitu. Rôzne varianty, napr. definovanie nejakej plošnej topológie a spojenie neurónu iba s niekoľkými okolitými neurónmi predchádzajúcej vrstvy, sú väčšinou podmnožiny tejto siete a dosiahnu sa pevným nastavením ostatných váh na nulu.

Vo vstupnej vrstve je 128 neurónov, ktoré zodpovedajú 128 pixelom vstupnej bitmapy. Vo výstupnej vrstve je 26 neurónov, každý znamená jedno písmeno anglickej abecedy – kódovanie 1 z N. Určenie vhodného počtu neurónov v skrytej vrstve bola hlavná úloha tejto práce.

Algoritmus učenia

Stochastický variant metóda spätného šírenia chyby (error backpropagation). Na vstup sa priloží jedna vzorka z trénovacej množiny a vypočíta sa výstup siete. Ten sa porovná s očakávaným výstupom. Podľa vypočítanej chyby na výstupnej vrstve sa určí chyba na poslednej skrytej vrstve a postupne na každej až po vstupnú vrstvu. Váhy spojení sa upravujú tak, aby sa znížila chyba a výstup siete sa priblížil očakávanému. Potom sa vyberie ďalšia trénovacia vzorka a celý postup sa opakuje.

Presnejšie: Chyba na neuróne je daná váženým súčtom (podľa váh spojení) chýb neurónov v nasledujúcej vrstve vynásobeným deriváciou aktivačnej funkcie. Zmena váhy spojenia je daná súčinom chyby neurónu na výstupnej strane spojenia, výstupu neurónu na vstupnej strane spojenia a konštanty. Konštanta udáva rýchlosť učenia a je ďalším dôležitým optimalizačným parametrom. Pri veľkej hodnote sa sieť rýchlo učí nové vzory, ale zároveň rýchlo zabúda staršie. Pri hodnote blízkej nule sa naučí lepšie, ale učenie trvá dlhšiu dobu.

Variant, pri ktorom sa váhy neaktualizujú hneď, ale vypočítaná chyba sa kumuluje cez celú trénovaciu množinu, sa ukázal ako menej vhodný. Učenie v tomto prípade trvalo mnohonásobne dlhšie aj napriek použitej optimalizácii momentom zotrvačnosti.

5.4 Implementácia neurónovej siete

Programovací jazyk

Ako programovací jazyk som si zvolil C++, pretože úloha je náročná na rýchlosť numerických výpočtov. Na druhej strane je oveľa názornejší v porovnaní s nízkoúrovňovými jazykmi blízkymi strojovému kódu.

Použité dátové typy

Na reprezentáciu spojitých reálnych hodnôt váh väzieb a výstupov neurónov sa hodia číselné typy s plávajúcou desatinnou čiarkou float a double. V ďalšom texte sa predpokladá:

```
typedef float ndt;
```

Skratka ndt znamená natívny dátový typ (native data type) a použije sa na všetky výpočty.

Dátové štruktúry

Uvedené budú len tie položky, ktoré sú dôležité pre kľúčové algoritmy.

Synapsa

```
class synapse {
    ndt weight;
    synapse() {weight=((ndt)rand()/(RAND_MAX+1.0))-0.5;}
};
```

Jediná podstatná vlastnosť spojenia medzi neurónmi je jeho váha. Pri inicializácii je dôležité použiť vyššie bity náhodného čísla, pretože nižšie bity majú neželané periodické vlastnosti. Typickou nevhodnou operáciou, ktorá zachováva nižšie bity, je modulo (zbytok po delení).

Neurón

```
class neuron {
    int width;
    synapse *synapses;
    ndt in;
    ndt out;
    ndt err;
    void compute() {out=1/(1+exp(-in));}
};
```

Width znamená počet spojení s neurónmi predchádzajúcej vrstvy. V našom prípade je to rovné počtu neurónov v predchádzajúcej vrstve.

Hlavná zložka je synapses - odkaz na pole spojení s neurónmi predchádzajúcej vrstvy. Index synapsy v poli je zároveň indexom neurónu predchádzajúcej vrstvy, ktorému synapsa prináleží.

In je vnútorný stav neurónu vypočítaný ako vážený súčet vstupov.

Out je výstup neurónu.

Err je pomocná premenná, ktorá sa použije pri učení na odloženie vypočítanej chyby.

Metóda compute() počíta výstup neurónu aplikáciou aktivačnej funkcie na vnútorný stav.

Vrstva a sieť

```
class layer {
    int width;
    neuron *neurons;
```

```

        void compute() {for (int i=0; i<width; i++) neurons[i].compute();}
};
class network {
    int width;
    layer *layers;
};

```

Analogicky obsahujú odkaz na začiatok poľa podzložiek (neurónov a vrstiev) a údaj o ich počte.

Rozpoznávanie

Výstupy nulte vrstvy sa nastavujú na vybraný vstupný vektor. Výstupy ostatných vrstiev sa vypočítajú nasledovne:

```

for (int i=1; i<width; i++) {
    for (int j=0; j<layers[i].width; j++) {
        layers[i].neurons[j].in=0;
        for (int k=0; k<layers[i].neurons[j].width; k++) {
            layers[i].neurons[j].in +=
            layers[i-1].neurons[k].out * layers[i].neurons[j].synapses[k].weight;
        }
        layers[i].neurons[j].compute();
    }
}

```

Index *i* ide cez všetky vrstvy s výnimkou vstupnej. Index *j* ide cez neuróny vrstvy *i*. Index *k* ide cez synapsy neurónu.

Učenie

Po vypočítaní odozvy siete na aktuálny vstup sa vypočíta chyba na každom neuróne výstupnej vrstvy:

```

for (i=0; i<layers[width-1].width; i++) {
    out=layers[width-1].neurons[i].out;
    layers[width-1].neurons[i].err=(target[i]-out)*out*(1-out);
}

```

A chyba sa šíri cez zvyšok siete:

```

for (i=width-2; i>0; i--) {
    for (j=0; j<layers[i].width; j++) {
        layers[i].neurons[j].err=0;
        out=layers[i].neurons[j].out;
        for (k=0; k<layers[i+1].width; k++) {
            layers[i].neurons[j].err +=
            layers[i+1].neurons[k].synapses[j].weight*layers[i+1].neurons[k].err;
        }
    }
}

```

```

    }
    layers[i].neurons[j].err *= out*(1-out);
}
}

```

Nakoniec sa upravujú váhy spojení:

```

for (i=1; i<width; i++) {
    for (j=0; j<layers[i].width; j++) {
        for (k=0; k<layers[i].neurons[j].width; k++) {
            layers[i].neurons[j].synapses[k].weight +=
                rate*layers[i].neurons[j].err*layers[i-1].neurons[k].out;
        }
    }
}

```

Celý zdrojový kód zaberá okolo 500 riadkov textu.

5.5 Postup testovania neurónovej siete

Ešte pred začiatkom tréningu sa vyberie určitý počet vzoriek na testovanie. Tie nie sú použité pri učení, aby sa overila schopnosť poradiť si s novými objektami. Predložia sa sieti na klasifikáciu a z počtu správne zaradených vzoriek sa vypočíta percentuálna úspešnosť. To je hlavné kritérium hodnotenia.

Iné metriky, Euklidovu vzdialenosť získaného od žiadaného vektora a súčet chýb (bez znamienok) z algoritmu učenia som použil ako pomocné ukazovatele na meranie postupu tréningu. Neprejavili sa vo finálnom hodnotení.

Testovacie vzorky vybrané z čitateľných písmen nespĺnili svoj účel, ľahko sa dosiahla úspešnosť správneho určenia 99-100 percent a nebolo vidno rozdiely medzi rôznymi nastaveniami. Preto som vybral najťažšie klasifikovateľné vzorky - nečitateľne napísané, šikmé alebo silne otočené. Neukázali síce úspešnosť pri reálnom použití, ale lepšie sa prejavili rozdiely pri rôznych nastaveniach parametrov siete. Akýkoľvek iný výber testovacej sady by bol nejakou štatistickou zmesou predchádzajúcich extrémnych prípadov.

Pri meraní odolnosti proti šumu a posunutiu som použil rovnaké trénovacie aj testovacie dáta, ale pri teste boli zodpovedajúcim spôsobom upravené.

6 Experimentálne výsledky

Počet a šírka vrstiev

Neurónová sieť bez skrytej vrstvy úlohu nezvládla, úspešnosť bola okolo 40 percent.

Sieť s 1 skrytou vrstvou dosahovala najlepšie výsledky pri 60 neurónoch v skrytej vrstve. Úspešnosť na najťažších testovacích vzorkách bola 70 - 75 percent. Menšia šírka skrytej vrstvy úspešnosť drasticky znížila. Väčšia šírka až po 500 neurónov nijak nezlepšila úspešnosť rozpoznávania, iba predĺžila čas učenia.

Rozdeliť neuróny do viacerých skrytých vrstiev bolo nevýhodné, úspešnosť klesla. Dôvod je pravdepodobne výrazne väčší počet neurónov vo vstupnej vrstve ako v skrytej vrstve. Preto sa počet spojení medzi neurónmi po rozdelení do dvoch vrstiev zmenšil a výpočtová sila siete slabla. Príklad: Skrytá vrstva 60 neurónov preorganizovaná na dve vrstvy po 30. Neurón v prvej skrytej vrstve má 128 spojení na neuróny vstupnej vrstvy. Neurón v druhej skrytej vrstve má iba 30 spojení na neuróny v prvej skrytej vrstve. To je štyrikrát menej.

Rýchlosť učenia (learning rate)

Pri konštantnej rýchlosti učenia boli najlepšie výsledky okolo hodnoty 0,01. Pri väčšej rýchlosti (0,1; 0,05) sa sieť nedokázala dostatočne natrénovať, úspešnosť bola až o desiatky percent nižšia.

Pri rýchlosti pod 0,005 učenie trvalo neúnosne dlho, ani po desiatkach hodín nedosiahlo optimum.

Výhodnejšie ako konštantná rýchlosť sa ukázalo nastaviť ju na vyššiu hodnotu a postupne znižovať. Prvých niekoľko iterácií rýchlosťou rádo 0,1, potom 0,01 a na záver 0,001. Prechod na menšiu rýchlosť vždy vtedy, keď sa úspešnosť klasifikácie prestane výrazne zvyšovať. Tak sa dosiahne dobré natrénovanie siete v relatívne krátkom čase desiatok minút až niekoľkých hodín.

Preučenie (overfitting)

Takmer vôbec sa neprejavilo. Ku koncu učenia sa stávalo, že úspešnosť na testovacích vzorkách klesla o niekoľko desiatin percenta a potom ďalej stúpala. Z toho vyplýva, že prvé dosiahnuté lokálne maximum úspešnosti sa nedá použiť ako ukončovacia podmienka

trénovacieho procesu.

Ani pri sieti s 500 neurónmi v skrytej vrstve (osemkrát väčšia ako najmenšia vhodná veľkosť) sa neukázalo preučenie. Po dosiahnutí maxima ani niekoľko ďalších dní a tisícok iterácií učenia už úspešnosť neznížilo viac ako o 2 percentá.

Možná príčina je obrovský nepomer medzi počtom trénovacích vzoriek a počtom neurónov. Sieť bola príliš malá, aby si zapamätala všetky trénovacie vzorky osobitne, a bola prinútená zovšeobecňovať.

Lokálne minimum

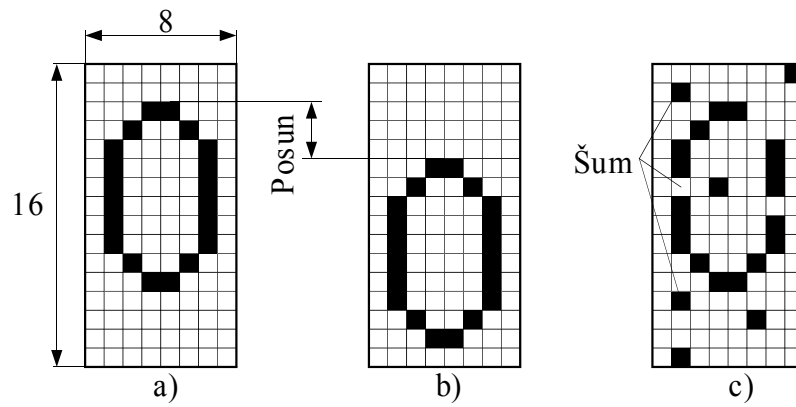
Opakované pokusy s rôznymi počiatočnými nastaveniami váh spojení medzi neurónmi skončili s veľmi podobnými výsledkami. To znamená, že proces učenia nenašiel vždy globálne najlepšie možné nastavenie, ale aj lokálne minimum je dostatočné riešenie a nepredstavuje závažný problém.

Toto tvrdenie má však len hypotetický charakter, pretože nie je známe skutočné globálne minimum. Poznanie jednoduchého analytického riešenia jeho hľadania by možno metódu spätného šírenia chyby spravilo zastaralou.

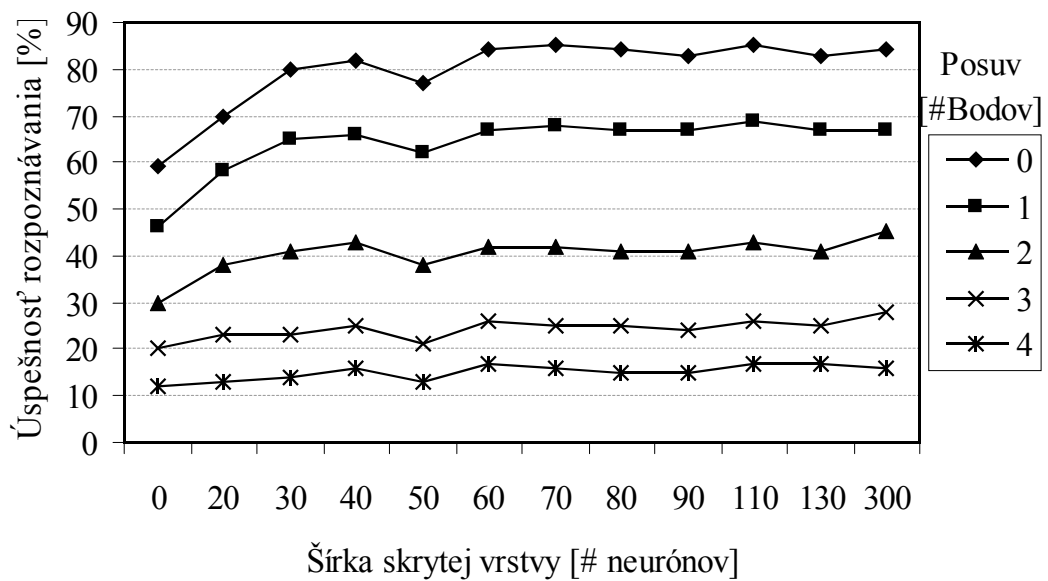
Odolnosť proti šumu a posunutiu

Prekvapivo vysokú robustnosť pri klasifikácii silne zašumených písmen mala sieť bez skrytej vrstvy. Celkovo nízka úspešnosť klesala pri zosilňovaní šumu iba minimálne, takže pri úrovni 15 percent invertovaných pixelov sa vyrovnala aj sieti s 80 neurónmi v skrytej vrstve.

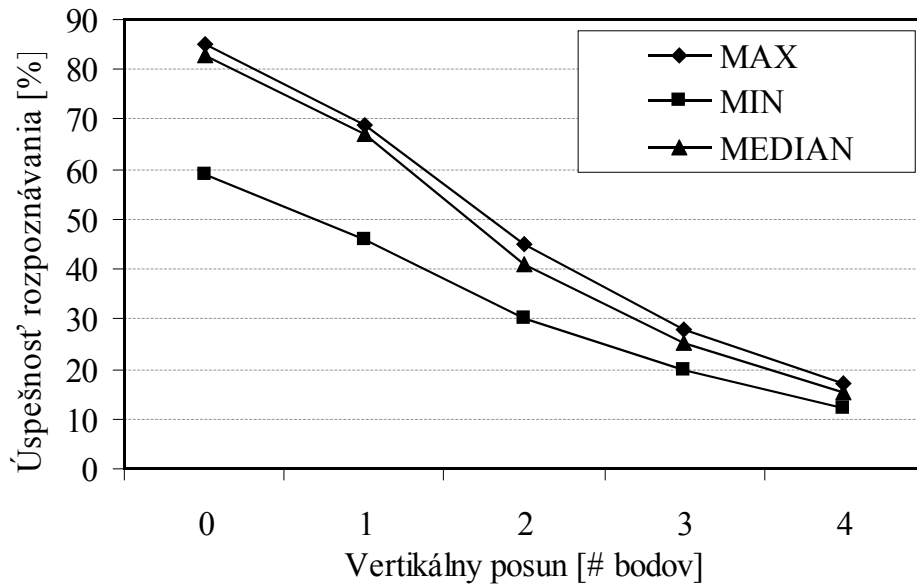
Úspešnosť určovania posunutých vzoriek prudko klesala u všetkých sietí.



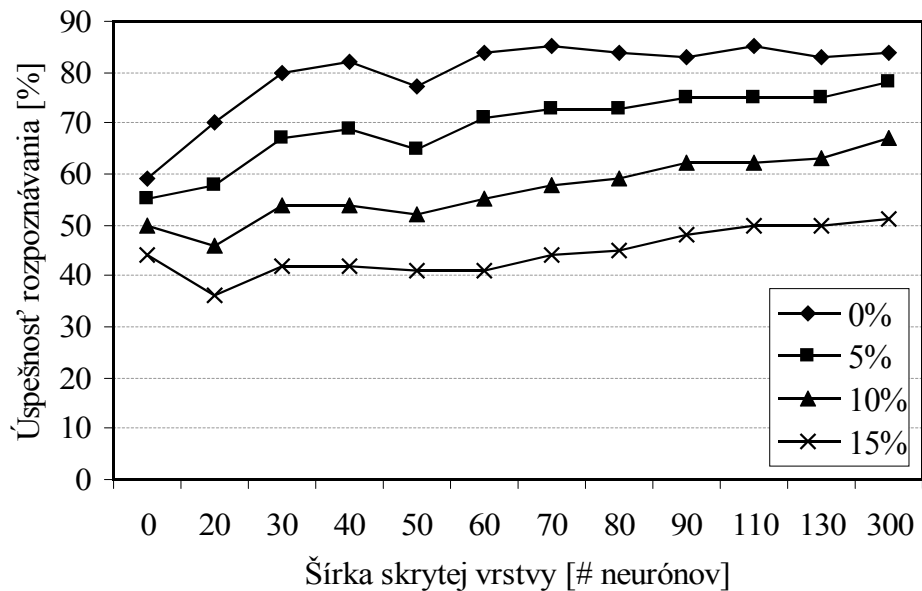
Obrázok 11: Formát obrázcov v použítom súbore znakov a) so šírkou 8 a výškou 16 obrazových bodov. Na obrázku b) je vertikálne posunutý a na c) zašumený obrazec použité v teste robustnosti siete.



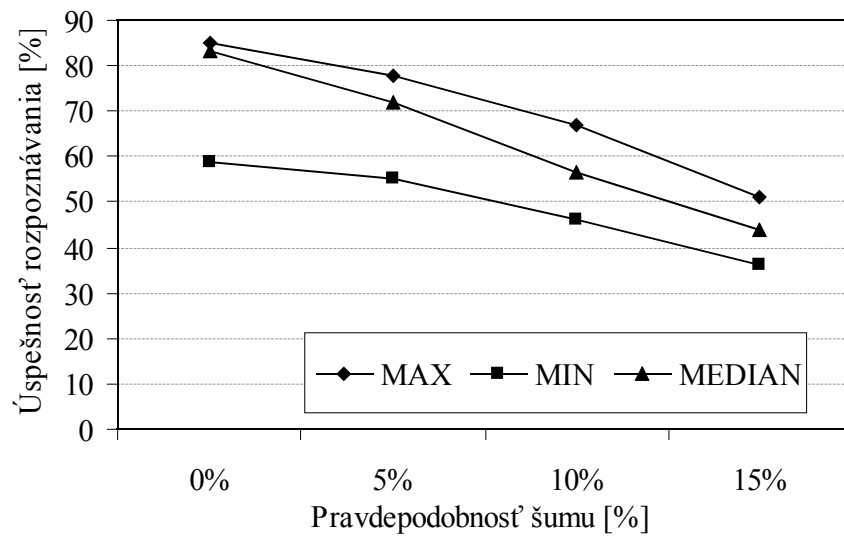
Obrázok 12: Vplyv počet neurónov v skrytej vrstve na úspešnosť rozpoznávania pre rôzne vertikálne posuvy klasifikovaných vzorov.



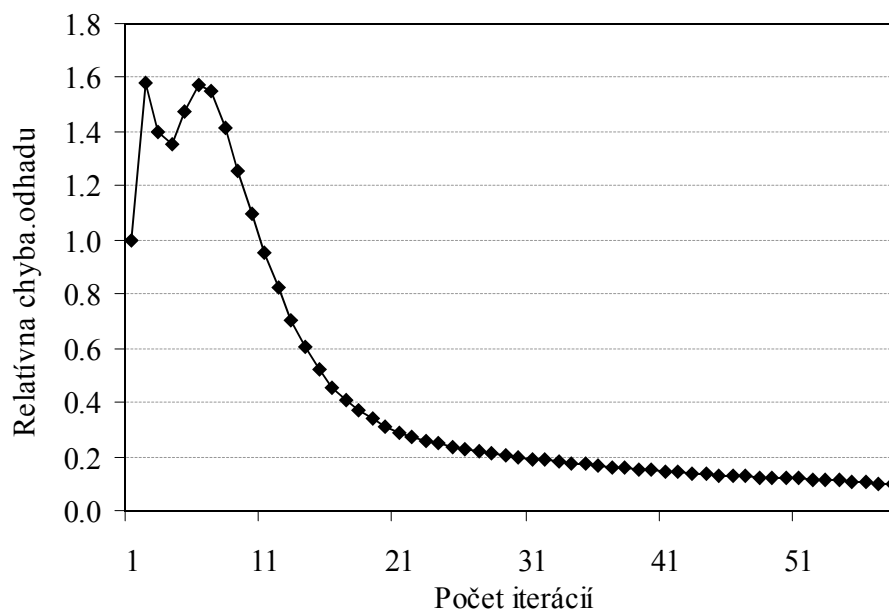
Obrázok 13: Maximum, minimum a median úspešnosti rozpoznávania obrazcov ako funkcia vertikálneho posuvu klasifikovaných obrazcov.



Obrázok 14: Vplyv počtu neurónov v skrytej vrstve na úspešnosť rozpoznávania obrazcov pre rôzne úrovne šumu v klasifikovaných vzoroch.



Obrázok 15: Maximum, minimum a median úspešnosti rozpoznávania obrazcov ako funkcia úrovne šumu v klasifikovaných obrazcoch



Obrázok 16: Priebeh relatívneho prírastku chyby výstupného vektora v závislosti od iterácie počas trénovania neuróvej siete. Veľkosť trénovacieho súboru 51892 vektorov, veľkosť výstupného vektora 26

7 Záver

V tejto práci sa podarilo naštudovať a naimplementovať viacvrstvovú perceptrónovú sieť a stochastickú variantu učenia algoritmom error backpropagation v programovacom jazyku C++ , ako aj úpravu sady písmen. Našla sa približná minimálna veľkosť tejto siete na účely OCR. Dosiahnutá úspešnosť ale nestačí na automatizovaný prepis textu bez korekcie človekom. Otestovala sa aj schopnosť spracovať poškodené vstupné dáta.

Verím, že aj táto práca prispela k otvorenej a nevyjasnenej téme vhodných parametrov neurónovej siete, a tak splnila svoj účel. Jedinými pomocníkmi sú v tomto zatiaľ odhad, pokus, omyl a skúsenosti svoje alebo druhých.

Okrem toho v nej čerstvý záujemca o neurónové siete nájde kuchársku knihu, ako naimplementovať v súčasnosti najpopulárnejšiu architektúru a algoritmus učenia. Slovný popis je často výrazne jednoduchší na pochopenie základných myšlienok a paralel k fyzickej štruktúre a funkcii neurónovej siete (biologickej alebo umelej) ako matematické vzorce. Tie sú vo väčšine dostupnej literatúry prirovnateľné k programom v strojovom kóde - sú presné, dávajú korektné výsledky ale nič v nich nie je vidieť. Ukážka konkrétneho programu dopĺňa slovný popis aspoň čiastočne o jednoznačnosť matematickej formuly.

Možnosti, ako ďalej zvyšovať úspešnosť rozoznávania písmen, vidím hlavne v častejšom opakovaní najnáročnejších vzoriek v procese učenia a v zmene architektúry siete, aby lepšie odrážala dvojrozmerné priestorové závislosti vo vstupných údajoch. Kvalitatívny rozdiel smerom k analógii ľudského vnímania by bol pokus o vektorizáciu a hľadanie smeru ťahov pera namiesto spracovania rastrového obrázka.

8 Zoznam použitých zdrojov

- [1] 1. Aliev,R.A, Aliev,R.R.: Soft Computing and its Application, World Scientific Publishing Co. Pte. Ltd., 2001, ISBN 981-02-4700-1
- [2] Mehrotra, K., Mohan, C., K., Ranaka, S.: Elements of Artificial Neural Networks, The MIT Press, 1997, ISBN 0-262-13328-8

- [3] Munakata, T.: Fundamentals of the New Artificial Intelligence, Springer-Verlag New York, Inc., 1998. ISBN 0-387-98302-3
- [4] Mařík, V., Štěpánková, O., Lažanský, J. a kol.: Umělá inteligence (4): Academia, 2003, ISBN 80-200-1044-0
- [5] Rob Kassel: Súbory testovacích znakov. MIT Spoken Language Systems Group , ai.stanford.edu/~btaskar/ocr/

9 Zoznam použitých skratiek a symbolov

NN – neurónová sieť (neural network)

OCR – rozoznávanie písmen (optical character recognition)