

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Invariantné rozpoznávanie objektov pomocou perceptrónu s konvolúciou

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



Invariantné rozpoznávanie objektov pomocou perceptrónu s konvolúciou

Diplomová práca

Študijný program: kognitívna veda
Študijný odbor: 2503 kognitívna veda
Školiace pracovisko: Katedra aplikovanej informatiky
Vedúci bakalárskej práce: doc. Ing. Igor Farkaš, PhD.



Middle European
Interdisciplinary Master Programme
in Cognitive Science

Bratislava 2012

Gregor Zaťko



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Gregor Zaťko
Študijný program: kognitívna veda (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.11. kognitívna veda
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Invariantné rozpoznávanie objektov pomocou perceptrónu s konvolúciou

Cieľ:

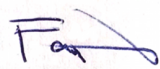
1. Naštudujte si problematiku dopredných architektúr neurónových sietí, so zameraním sa na perceptrón s konvolúciou.
2. Experimentálne preskúmajte vlastnosti tohto modelu na vybraných dátových množinách, čo sa týka úspešnosti invariantného rozpoznávania 2D objektov, v závislosti od vlastností trénovacích dát a od parametrov modelu.
3. Porovnajte tento model s iným vybraným výpočtovým modelom (napr. deep belief network) na zvolených trénovacích dátach.

Vedúci: doc. Ing. Igor Farkaš, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Dátum zadania: 31.08.2011

Dátum schválenia: 28.02.2012

prof. RNDr. Pavol Zlatoš, PhD.
garant študijného programu

.....
študent


.....
vedúci práce

Vyhlasujem na svoju česť, že predloženú diplomovú prácu som vypracoval samostatne a za použitia uvedenej literatúry.

v Bratislave
25.05.2012

Ďakujem RNDr. Kristíne Rebrovej za starostlivé kontrolovanie mojej práce a za cenné pripomienky.

Abstrakt

V mojej práci sa venujem problematike spracovania obrazových dát biologicky inšpirovanými postupmi, a to konvolučnými neurónovými sieťami, ktoré sú založené na štruktúre vizuálnej kôry v mozgu. Dosahujú výborné výsledky v úlohách na rozpoznávanie obrazových vzorov. Cieľom práce je otestovať sieť tohto typu na datasete prírodných objektov (listov) a zmerať chybovosť na rôznych konfiguráciách siete. Ďalším cieľom je porovnanie siete na konkrétnom datasete s inou technikou rozpoznávania vzorov, Deep Belief Network (DBN), a približné zhodnotenie obtiažnosti datasetu porovnaním s iným datasetom, v tomto prípade MNIST (rukou písané číslce). Ako výsledok sa ukázalo, že konvolučná sieť v tejto úlohe prevyšuje v úspešnosti klasifikácie DBN.

Kľúčové slová

rozpoznávanie vzorov, spracovanie obrazových dát, konvolučné neurónové siete

Abstract

In this work I concern with an issue of processing of image data by biologically inspired techniques, namely the convolutional neural networks, which are based on structure of visual cortex in brain. They achieve quite excellent results in tasks of image pattern recognition. The aim of the work is to test a network of this type on natural objects (leaves) dataset and measure errors of various network configurations. Another aim is the comparison of the network on this concrete dataset with other pattern recognition technique, the Deep Belief Network (DBN), and approximate evaluation of difficulty of the dataset compared to other one, in this case MNIST (handwritten digits). As the result, it is shown that convolutional network quite significantly exceeds DBN on this task by its classification accuracy.

Keywords

pattern recognition, image data processing, convolutional neural networks

Obsah

1. Úvod.....	10
1.1 Motivácia.....	10
1.2 Symbolový vs. subsymbolový prístup.....	11
1.3 Konekcionalizmus.....	11
1.4 Multidisciplinarita.....	12
2. Teória neurónových sietí.....	13
2.1 Biologické neurónové siete.....	13
2.1.1 Anatómia neurónových sietí.....	13
2.2 Umelé neurónové siete.....	15
2.2.1 Umelý neurón.....	15
2.2.2 Architektúra neurónových sietí.....	17
2.2.3 Učenie neurónových sietí.....	18
2.2.4 Učenie pomocou spätného šírenia chyby.....	19
2.2.5 Výhody a nevýhody učenia pomocou spätného šírenia chýb.....	21
2.3 Hierarchické neurónové siete.....	22
3. Konvolučné neurónové siete.....	24
3.1 Princípy konvolučných neurónových sietí.....	24
3.1.1 Receptívne polia.....	24
3.1.2 Konvolúcia.....	24
3.1.3 Konvolúcia a separovateľný kernel.....	26
3.1.4 Priestorová konvolúcia.....	27
3.1.5 Priestorová subtraktívna normalizácia.....	28
3.1.6 Zdieľanie váh.....	29
3.1.7 Subsampling.....	29
3.2 Architektúra konvolučných neurónových sietí.....	30
3.3 Trénovanie konvolučných sietí.....	30
3.3.1 Boosting.....	31
3.4 Použitie konvolučných sietí.....	31
4. Praktická časť.....	33

4.1 Experiment s datasetom listov.....	33
4.1.1 Experimentálna zostava.....	33
4.1.2 Dataset.....	34
4.1.3 Výsledky – menšie konfigurácie.....	39
4.1.4 Výsledky - komplexnejšie konfigurácie.....	46
4.2 Experiment s datasetom MNIST.....	49
4.3 Diskusia.....	50
4.4 Možné vylepšenia.....	51
5. Zhrnutie.....	52
6. Zoznam použitých zdrojov.....	53

Zoznam obrázkov

Obrázok 1: Predstava biologického neurónu.....	14
Obrázok 2: Predstava umelého neurónu.....	16
Obrázok 3: Graf sigmoidálnej funkcie.....	17
Obrázok 4: Graf prahovej funkcie.....	17
Obrázok 5: Predstava štruktúry neurónovej siete.....	17
Obrázok 6: Predstava konvolúcie na obrazových bodoch.....	26
Obrázok 7: Aplikácia konvolučného filtra - rozostrenie (blur).....	26
Obrázok 8: Aplikácia konvolučného filtra - detekcia hrán.....	26
Obrázok 9: Aplikácia konvolučného filtra – zostrenie obrazu.....	26
Obrázok 10: Obraz po štvornásobnej aplikácii operácie priestorovej konvolúcie.....	28
Obrázok 11: Naľavo pôvodný obraz, napravo po operácii normalizácie kontrastu.....	28
Obrázok 12: Predstava architektúry konvolučnej neurónovej siete.....	30
Obrázok 13: Triedy listov použité v datasete.....	34
Obrázok 14: Ukážka datasetu 1.....	35
Obrázok 15: Ukážka datasetu 2.....	36
Obrázok 16: Ukážka datasetu 3.....	37
Obrázok 17: Ukážka datasetu 4.....	38
Obrázok 18: Dataset 1 - závislosť testovacej chyby od rýchlosti učenia.....	40
Obrázok 19: Dataset 2 - závislosť testovacej chyby od rýchlosti učenia.....	42
Obrázok 20: Dataset 3 - závislosť testovacej chyby od rýchlosti učenia.....	43
Obrázok 21: Dataset 4 - závislosť testovacej chyby od rýchlosti učenia.....	45
Obrázok 22: Dataset 5 - závislosť testovacej chyby od rýchlosti učenia.....	46

Zoznam tabuliek

Tabuľka 1: Dataset 1 - konfigurácie s rýchlosťou učenia 0.01.....	39
Tabuľka 2: Dataset 1 - konfigurácie s rýchlosťou učenia 0.02.....	40
Tabuľka 3: Dataset 2 - konfigurácie s rýchlosťou učenia 0.01.....	41
Tabuľka 4: Dataset 2 - konfigurácie s rýchlosťou učenia 0.02.....	41
Tabuľka 5: Dataset 3 - konfigurácie s rýchlosťou učenia 0.01.....	42

Tabuľka 6: Dataset 3 - konfigurácie s rýchlosťou učenia 0.02.....	43
Tabuľka 7: Dataset 4 - konfigurácie s rýchlosťou učenia 0.01.....	44
Tabuľka 8: Dataset 4 - konfigurácie s rýchlosťou učenia 0.02.....	44
Tabuľka 9: Dataset 5 - konfigurácie s rýchlosťou učenia 0.01.....	45
Tabuľka 10: Dataset 5 - konfigurácie s rýchlosťou učenia 0.02.....	46
Tabuľka 11: Výsledky siete na datasete 1 (množina konfigurácií 2).....	47
Tabuľka 12: Výsledky siete na datasete 2 (množina konfigurácií 2).....	47
Tabuľka 13: Výsledky siete na datasete 3 (množina konfigurácií 2).....	48
Tabuľka 14: Výsledky siete na datasete 4 (množina konfigurácií 2).....	48
Tabuľka 15: Výsledky siete na datasete 5 (množina konfigurácií 2).....	48
Tabuľka 16: Výsledky siete na datasete MNIST.....	49
Tabuľka 17: Porovnanie najlepších výsledkov CNN a DBN.....	50

1. Úvod

1.1 Motivácia

Ľudské vnímanie je každodenne vystavené rôznym formám informácií, z ktorých je niektoré potrebné zaraďovať do kategórií. Ako príklady môžeme uviesť spracovanie textu písaného ľuďmi, triedenie medicínskych vzoriek na počítačovú diagnostiku, klasifikáciu elektronickej pošty (spam) alebo rozpoznávanie tvárí. U niektorých úloh však ide o značnú časovú náročnosť na rozpoznávanie a kategorizáciu daných objektov, preto boli vyvinuté technológie na strojové riešenie týchto úloh. Tieto môžu fungovať na princípe štatistickom alebo neštatistickom. Príkladmi takýchto riešení sú rôzne regresné metódy, rozhodovacie stromy, analýzu hlavných komponentov (PCA, Principal Component Analysis [22]) alebo Support Vector Machines [23]. Jednou z kategórií týchto metód sú aj umelé neurónové siete [1]. Kým vo väčšine prípadov ide o umelo vytvorené matematické techniky, neurónové siete sú založené na napodobnení toho, čo funguje prirodzene s výbornými výsledkami, teda princípe fungovania ľudskej nervovej sústavy a to hlavne mozgu. Vylepšenie základných modelov, tzv. konvolučná neurónová sieť [25], postupuje v tejto inšpirácii biologickými javmi a spája základnú myšlienku sietí neurónov s technikami spracovania obrazu a celými architektúrami štruktúr nachádzajúcich sa v častiach mozgu, ktoré majú za úlohu spracovať vizuálny signál prichádzajúci zo sietnice. Ukazuje sa, že takéto postupy majú značnú úspešnosť nielen v testovacích podmienkach, ale aj na úlohách, ktoré vznikajú pri ich reálnom nasadení. Konvolučná neurónová sieť je jedna z najlepších architektúr, čo sa týka výsledkov v úlohách klasifikácie obrazových dát. Táto práca si dáva za cieľ aspoň čiastočne ukázať pravdivosť tohto tvrdenia na dvoch datasetoch z reálneho sveta – prvým je množina listov, druhým je pre porovnanie množina rukou písaných číslíc. Ďalším jej cieľom je porovnať získané výsledky s inými technikami na rozpoznávanie obrazových dát a posledným je zistiť vhodné architektúry tejto siete.

Rozdelenie práce je nasledovné: v prvých častiach sa nachádza teoretický úvod do problematiky, ktorý začína prístupmi k umelej inteligencii. Pokračuje cez biologické minimum vhodné k štúdiu neurónových sietí, popis umelých neurónov, ich sietí a prístupov

k ich učeniu. Nasleduje popis techník používaných na spracovanie obrazu a v závere tejto časti sa nachádza popis konvolučných neurónových sietí a problematiky s nimi spojenej.

V praktickej časti je popísaná experimentálna zostava a následne uvedené jednotlivé výsledky pre experimenty na dvoch datasetoch. Na konci kapitoly je diskusia o zistených výsledkoch.

1.2 Symbolový vs. subsymbolový prístup

Jedno z vysvetlení fungovania mozgu je, že tento pracuje symbolovým spôsobom, tak ako napríklad funguje jazyk. To znamená, že naše myšlienky môžu byť rozdelené do symbolov, podobne ako slová v jazyku. Problém vznikne v prípade, keď začneme skúmať vnútorné reprezentácie toho, čo vnímame – toto vysvetlenie sa stane nedostatočné. Ukazuje sa, že spôsob ukladania informácií, napríklad vizuálnych, môže byť oveľa lepšie vysvetlený pomocou akejsi vrstevnatej mapy, z ktorej akýkoľvek kus informácie nemôže byť odstránený bez toho, aby ovplyvnil informácie v jeho okolí. Tu je zjavný rozdiel medzi týmito dvomi prístupmi, keďže v prvom sú jednotlivé prvky, teda slová alebo frázy, zväčša nezávislé na iných. Spomenutý prístup, prirovnaný k mape, je nazývaný konekcionizmus a ponúka značne presnejší popis kognitívnej reprezentácie, ako prvý, symbolový prístup.

1.3 Konekcionizmus

Konekcionizmus je prístup, ktorý sa snaží skúmať ľudskú kogníciu za použitia matematických modelov, ktoré sú známe pod názvom konekcionistické siete alebo používanéjšie neurónové siete. Vo všeobecnosti, konekcionistické modely sú biologicky značne viac v súlade s realitou ľudského mozgu ako symbolový prístup. Tento predpokladá, že v rámci mozgu existuje istý druh “neurálneho jazyka“, ktorý reprezentuje všetky informácie podobne ako písaná alebo hovorená forma ľudského jazyka, pričom sú uložené ako individuálne jednotky, tzv. symboly, ktorých manipulácia prebieha pomocou určených pravidiel. Súčasný stav teórie umelej inteligencie si myslí, že obidve teórie spolu nesúťažia, ale naopak, dopĺňajú sa navzájom [26].

1.4 Multidisciplinarita

V práci je pomerne silná inšpirácia postupmi, ktoré boli (a sú) pôvodne využívané v prírode. Pri neurónových sieťach už názov pripomína, že motivácia pre tieto systémy je v ľudskom nervovom a kognitívnom systéme. Skúmanie týchto javov spadá do oblastí neurovedy, prípadne výpočtovej neurovedy a samozrejme umelej inteligencie.

Pokiaľ ide o ich špecializáciu, a síce konvolučné neurónové siete, okrem tohto biologického pôvodu, ktorý je na úrovni neurónov, majú aj ďalší súvis, ktorý je skôr na architektonickej úrovni siete, a siete ako celku. Pre samotnú operáciu konvolúcie sa mi nepodarilo nájsť jej ekvivalent v biologických systémoch. Napriek tomu, to, čo je výsledkom jej použitia na obrazový vstup, má svoju podobu napríklad aj v ľudskom videní, keďže tiež funguje na princípe vytvárania máp príznakov za účelom invariance pozorovaných objektov a viacerých vrstiev takýchto máp.

2. Teória neurónových sietí

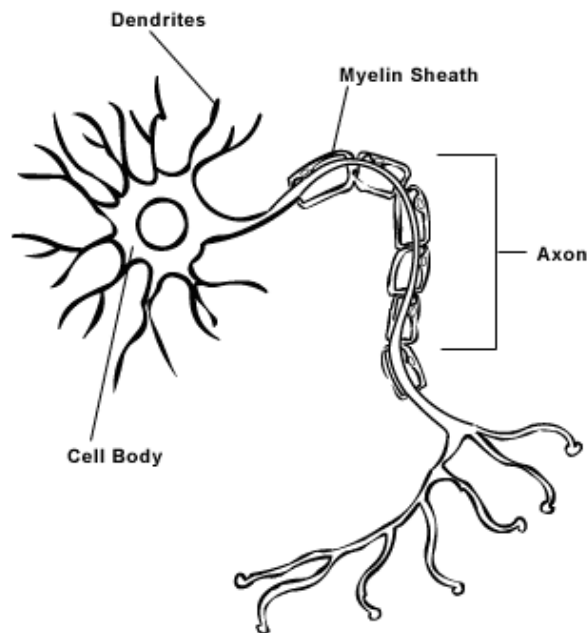
2.1 Biologické neurónové siete

Skúmanie neurónových sietí spadá do oblasti neurovedy, táto tvorí podoblasť biológie, ktorá sa zaoberá nervovým systémom. Neuroanatómia sa zaoberá štruktúrou a funkčnosťou zoskupení neurónových sietí, so zreteľom tak na časti mozgu, ako aj na štruktúry, ktoré vedú od mozgu do ďalších častí tela. Ďalšia disciplína je neuropsychológia, ktorá sa zaoberá štruktúrou a funkcionalitou mozgu vo vzťahu k abstraktnému psychologickému správaniu.

Samotný termín biologická neurónová sieť sa vzťahuje na časti spracovania informácií v ľudskom nervovom systéme, ktoré sú organizované ako zoskupenie neurónových buniek, nazývaných neuróny, ktoré sú pospájané do sietí a interagujú spolu navzájom za použitia elektrochemických signálov. Pre inšpiráciu výpočtovým modelom slúžia neurónové siete prevažne v mozgu.

2.1.1 Anatómia neurónových sietí

Štruktúra mozgu nie je homogénna – v najväčšej mierke je rozdelený na kôru, stredný mozog, mozgový kmeň a mozoček. Každá z týchto častí môže byť ďalej rozdelená na oblasti a podoblasti a to podľa anatomickej štruktúry neurónových sietí v nej alebo podľa funkcie, ktorá je v nej vykonávaná. Celková mapa zväzkov neurálnych spojení je však veľmi komplexná a známa len čiastočne. Tu sú rozlíšené dva druhy spojení: prvý tvoria dopredné spojenia, ktoré smerujú od skorších fáz spracovania, čiže od senzorických vstupov, do neskorších, teda napr. motorických výstupov. Druhý tvoria spätnoväzobné spojenia, ktoré smerujú opačným smerom. Okrem týchto dlhších spojení sú neuróny spojené aj s ich okolitými neurónmi [1].



Obrázok 1: Predstava biologického neurónu

Biologický neurón je všeobecne zložený z tela, axónu a dendritov (obr 1.), pričom axón je obvykle len jeden, dendritov môže byť viac. Axón (dlhý výbežok – odstredivý kanál) slúži na odosielanie signálov, pričom na konci obvykle býva rozdelený do ďalších vlákien a podvlákien. Dendrity (krátke výbežky – dostredivý kanál), ktoré majú stromovú štruktúru, slúžia na prijímanie signálov, ktoré sú (zhruba) sumované. Neurón páli (spike, fire) pri prekročení určitého prahu napätia vstupných signálov – tento proces je nazývaný aj depolarizácia. Po ňom nasleduje istá perióda, kedy neurón páliť nemôže.

Spojenie neurónov je zabezpečované pomocou synáps, čiže miest, kde sa koniec axónu takmer dotýka tela iného neurónu. Prenos elektrických signálov je zabezpečovaný chemicky, pomocou tzv. neurotransmiterov, ktoré sú vypustené z vysielajúceho neurónu a spojené s receptorom na prijímajúcom neuróne. Rozsah, v ktorom je signál prenesený medzi neurónmi, závisí od viacerých faktorov, napríklad množstvo dostupného neurotransmiteru, počet a usporiadanie receptorov, množstvo absorbovaného neurotransmiteru a podobne.

Jedným zo spôsobov učenia je vo vzťahu práve so synaptickými spojeniami. Tieto môžu byť v mozgu pozmeňované v ohľade ich intenzity alebo ich vznikaním, resp. zanikaním.

Toto učenie prebieha na základe skúsenosti a typicky bez učiteľa. Zmena intenzity spojení je založená na tzv. Hebbovej teórii, ktorá vraví:

“Majme dva neuróny A a B. Ak axón bunky A excituje bunku B opakovane alebo trvalo, pomocou určitých metabolických zmien alebo procesu rastu v jednej alebo oboch bunkách je zvýšená efektivita bunky A ako jednej z páliacich do B.” [3]

Tento postulát je niekedy používaný v skrátenej verzii “cells that fire together, wire together”.

2.2 Umelé neurónové siete

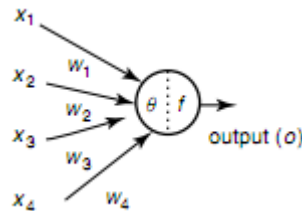
Oblasť umelých neurónových sietí (Artificial Neural Networks – ANN) sa zaoberá skúmaním výpočtových modelov inšpirovaných teóriou a pozorovaním štruktúry a funkcionality biologických sietí neurónových buniek, predovšetkým v mozgu (pozn.: v ďalšom texte bude termínom „neurónová sieť“ mienená práve umelá neurónová sieť). Sú všeobecne navrhnuté a určené pre riešenie matematických, výpočtových a inžinierskych problémov. Preto je pochopiteľné, že v dôsledku tohto je v danej oblasti množstvo interdisciplinárneho výskumu z oblastí matematiky, neurobiológie a informatiky.

Vo všeobecnosti je neurónová sieť zložená z umelých neurónov, ktoré sú navzájom pospájané s cieľom vykonávania výpočtov na vzorkách vstupných dát a vytvorenia vzoriek výstupných dát [1]. Sú to adaptívne systémy schopné prispôsobiť ich vnútornú štruktúru, obvykle sú to váhy medzi jednotlivými neurónmi v sieti. To im dovoľuje, aby boli používané na rôznorodé problémy, ktoré môžeme všeobecne zahrnúť pod aproximáciu funkcie (funkcií). Sú to napríklad regresná analýza, rozpoznávanie vzorov a klasifikácia, extrakcia vlastností, predpovedanie časových radov a numerické riadenie v robotike.

2.2.1 Umelý neurón

Model neurónu, niekedy nazývaný aj jednotka (unit) alebo uzol (node) tvorí matematická funkcia, ktorá abstrahuje neurón biologický. Tieto neuróny sú obvykle zjednodušené, hoci existujú aj modely, ktoré simulujú biologické neuróny pomerne presne v súlade s realitou

(napr. model Hodgkin–Huxley [11], Izhikevichov model [12]), avšak obvykle nebývajú použité na výpočtovo zamerané neurónové siete, ale na simulačné účely. Prvý takýto výpočtový model neurónu navrhli v roku 1943 McCulloch a Pitts [2].



Obrázok 2: Predstava umelého neurónu

Typicky má umelý neurón konštantný počet vstupov (n), pričom každý je ohodnotený tzv. váhou (w). Schéma umelého neurónu je na obr. 2. Matematicky ju môžeme zapísať v tvare

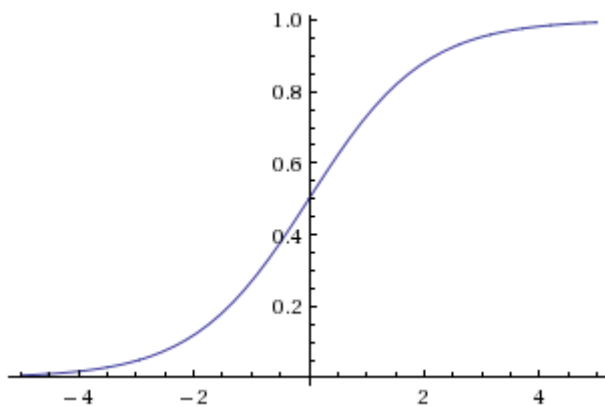
$$O = f(\text{net}) = f\left(\sum_{i=1}^n w_i x_i\right),$$

kde, w je vektor váh a $f(\text{net})$ je tzv. aktivačná funkcia. Premenná net je definovaná ako súčin váhy a vstupného vektora:

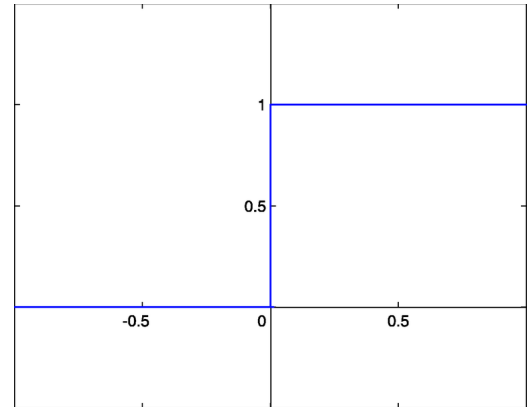
$$\text{net} = w^T x = w_1 x_1 + \dots + w_n x_n,$$

kde T je operácia transpozície matice.

Aktivačná funkcia je potrebná na pridanie nelinearity do siete – bez nej by sieť nevedela plniť jej funkciu. Obvyklá požiadavka na aktivačnú funkciu je jej derivovateľnosť a ohraničenosť, túto spĺňajú napríklad prahová funkcia, sigmoidálne funkcie alebo Gaussián. Grafy a zápisy prvých dvoch sú znázornené na obr. 3 a 4.



Obrázok 3: Graf sigmoidálnej funkcie



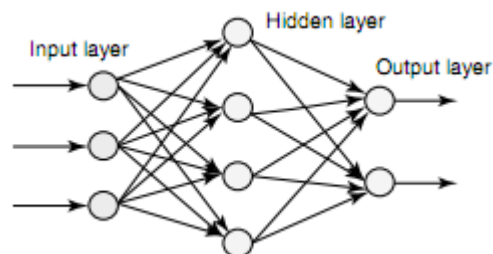
Obrázok 4: Graf prahovej funkcie

$$y = \frac{1}{1 + a e^{-x}}$$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$

2.2.2 Architektúra neurónových sietí

Neurónové siete majú najčastejšie štruktúru vrstiev neurónov, pričom môžeme rozdeliť vrstvy do troch typov: vstupné (input), výstupné (output) a skryté (hidden) (obr. 5).



Obrázok 5: Predstava štruktúry neurónovej siete

V dopredných sieťach signál postupuje od vstupných neurónov do výstupných striktne v tomto jednom smere. Môže ich tvoriť len jeden neurón, jedna vrstva až po niekoľko vrstiev. Príkladmi týchto sietí sú perceptrón [13], RBF sieť [14] a rôzne modifikácie viacvrstvových perceptrónov.

V rekurentných sieťach sú povolené slučky a štruktúra môže byť plne navzájom prepojená. Ich príkladmi sú napríklad Hopfieldova sieť [15] alebo sieť s echo stavmi [16].

Väčšina architektúr neurónových sietí je ťažko konfigurovateľná a časovo náročná na tréning, ich výhodou však je, že ak sú raz natréňované a nakonfigurované, použitie je veľmi rýchle a jednoduché. Ďalšie ich výhody sú schopnosť generalizovať riešenia problémov a dobrá odolnosť voči šumu.

2.2.3 Učenie neurónových sietí

Keďže je funkcia neurónových sietí založená na možnosti modifikácie ich vnútornej štruktúry, táto musí byť nejakým spôsobom získaná. Tento spôsob je nazvaný tréning alebo učenie a malo by zaručiť sieti schopnosť produkovať pre rôzne vstupné vzorky správne zodpovedajúce výstupy, čiže sieť by mala mať schopnosť generalizácie. Znalosť neurónovej siete o tréningových dátach by nemala byť príliš špecifická – takýto stav, v ktorom sieť síce produkuje správne výsledky prislúchajúce vstupným dátam, ale pre ostatné dáta nesprávne, je nazývaný preučenie (overfitting).

Tréning je rozdelený do troch hlavných typov: Pri tréningu s učiteľom (kontrolované, supervised) je sieť opakovane vystavovaná vstupu, ktorý má priradenú správnu očakávanú odpoveď. Vnútna reprezentácia siete je následne „posunutá“ tak, aby výstup siete lepšie zodpovedal očakávanému výsledku. Príkladom tohto spôsobu učenia je učenie pomocou spätného šírenia chyby [4].

Pri tréningu bez učiteľa (nekontrolované, unsupervised) je sieť vystavovaná vstupným vzorkám, z ktorých musí sama odvodiť význam a extrahovať ich vlastnosti. Najčastejším typom nekontrolovaného tréningu je kompetitívne učenie, kde neuróny navzájom súťažajú na základe vstupných vzoriek, aby vyprodukovali výstupné dáta. Príkladmi tohto typu učenia sú samoorganizujúce mapy [1] a neurálny plyn [17].

Pri tréningu s posilňovaním (reinforcement learning) je sieť učená, ako namapovať rôzne situácie do akcií tak, aby mala čo najväčšiu numerickú odmenu. Sieť nie je informovaná o tom, aké akcie má vykonávať, ale namiesto toho musí sama objaviť, za ktoré akcie dostane najväčšiu odmenu tým, že ich vyskúša. Miernou modifikáciou je oneskorenie odmeny, to znamená, že akcie nemajú vplyv len na okamžitú odmenu, ale aj na neskoršie situácie a tým pádom aj na neskoršie odmeny.

Proces tréovania môže byť považovaný za problém optimalizácie, kde je požadovaná minimizácia mean square error (MSE, stredná kvadratická chyba) siete na celom súbore tréovacích dát. Tento problém môže byť riešený viacerými odlišnými spôsobmi, od štandardných optimačných heuristik (napr. simulované žihanie) cez genetické algoritmy až po špecializované gradientové metódy. Z nich je pri neurónových sieťach najrozšírenejšia tzv. metóda učenia pomocou spätného šírenia chyby. Táto kvôli jej nevýhodám existuje aj v rôznych úpravách.

2.2.4 Učenie pomocou spätného šírenia chyby

Algoritmus spätného šírenia chyby [4] je jeden z najdôležitejších spôsobov tréovania váh v viacvrstvových dopredných neurónových sieťach. Používa adaptáciu váh na základe výpočtu chyby za prítomnosti vstupných vzoriek, pričom táto metóda je použitá v sieti spätne, to znamená. smerom od výstupnej vrstvy k vstupnej. Potrebuje na to sieť, ktorá má každú vrstvu plne prepojenú s nasledujúcou. Jeho cieľom je iteratívne zmenšovať celkovú chybu siete. Ako najjednoduchší príklad si môžeme určiť sieť s nasledujúcou štruktúrou vrstiev: vstupná vrstva, jedna skrytá vrstva a výstupná vrstva.

V priebehu tréovania siete môžu byť upravované dva typy parametrov: váhy a hodnota v aktivačných funkciách. Toto nie je prakticky výhodné a je lepšie, ak je upravovaný len jeden parameter. Na vyriešenie tohto problému sú používané bias neuróny. Každý z nich sa nachádza v jednej vrstve, pričom je plne prepojený s nasledujúcou vrstvou, avšak nie s predchádzajúcou a jeho hodnota je vždy 1, prípadne -1. Vzhľadom na túto skutočnosť sú váhy, pripojené na tento neurón, priamo pripočítavané k sume ostatných váh, takisto ako hodnoty v aktivačných funkciách. Pridanie biasu teda dovoľuje odstrániť hodnotu prahu z aktivačnej funkcie za ponechania úpravy váh pri tréovaní – bez jeho použitia by mohol byť výstup nulový pri všetkých vstupoch rovných nule, nezávisle od hodnôt váh.

Zjednodušený algoritmus učenia pomocou spätného šírenia chýb je zhruba takýto:

vstupy: tréovacie vzorky, počet iterácií (alebo iné zastavovacie kritérium), veľkosť problému, rýchlosť učenia

výstup: natréovaná sieť

1. inicializácia vrstiev a váh
2. výpočet výstupov podľa vybraného vstupného vzoru
3. výpočet chýb
3. výpočet delta
4. úprava váh
5. ak neboli použité všetky vzory, návrat do 2
6. ak neboli dodržané zastavovacie kritériá, návrat do 2

1. inicializácia váh

Inicializácia váh môže byť realizovaná dvomi spôsobmi: náhodne alebo pomocou určitého algoritmu, ktorého cieľom je priblížiť hodnoty váh ich pravdepodobným hodnotám po skončení tréningu, čo môže významne pomôcť ho skrátiť. Pokiaľ je rozdiel medzi počiatočnými a konečnými váhami rádový, tréningu môže trvať dlhšie, pokiaľ jeho chyba bude akceptovateľná. Pri náhodnej inicializácii sú obvykle použité malé hodnoty v intervale $(-0.1, 0.1)$. Príkladom inicializačného algoritmu je Nguyen-Widrow [18].

2. forward pass

Krok 2, nazývaný aj forward pass je vlastne výpočtom výstupu siete. Vstup je prešírený cez sieť z vstupnej vrstvy na výstupnú.

3. výpočet chýb

V tomto kroku sú najprv vypočítané chyby na jednotlivých neurónoch podľa vzťahu

$$e_k = d_k - y_k ,$$

kde e_k je chyba na konkrétnom neuróne, d_k je požadovaný výstup na neuróne a y_k je vlastný výstup neurónu. Tieto chyby sú ďalej použité na výpočet hodnoty δ_k , ktorá je znova použitá na úpravu váh. Táto hodnota je počítaná podľa vzťahu

$$\delta_k = e_k g'(y_k) ,$$

kde g' je derivácia aktivačnej funkcie neurónu. Pre predchádzajúce vrstvy je vypočítaná

hodnota δ_j , za pomoci hodnoty δ_k pomocou vzťahu

$$\delta_j = \alpha g'(y_j) \sum_{k=0}^K \delta_k w_{jk} ,$$

kde K je počet neurónov v vrstve a α je rýchlosť učenia, ktorý rozhoduje o miere, akou má byť váha w upravená. V rôznych vylepšeniach gradientových algoritmov nie je parameter rýchlosti učenia používaný, ale sú namiesto neho použité parametre, ktoré sa snažia odhadnúť vhodnú mieru posunu váhy.

4. úprava váh

Hodnoty δ sú ďalej použité na výpočet rozdielu, o ktorý sú váhy upravené, podľa vzťahu

$$\Delta w_{jk} = \delta_j y_k ,$$

ktorého výsledok je teda následne pripočítaný k váham takto:

$$w_{jk} = w_{jk} + \Delta w_{jk} ,$$

5. zastavenie algoritmu

Hlavný cyklus algoritmu, ktorý tvoria kroky 2-4, iteruje, až kým nie je dosiahnuté zastavovacie kritérium. Toto je zvyčajne určené meraním priemernej kvadratickej chyby (MSE) na tréningových dátach po každej iterácii, podľa vzťahu

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - y_n)^2 ,$$

trénovanie je zastavené, keď MSE dosiahne želanú hodnotu. Existujú taktiež pokročilejšie techniky na zastavovanie, napríklad použitie validačnej množiny dát.

2.2.5 Výhody a nevýhody učenia pomocou spätného šírenia chýb

Jednou z výhod tréningovania sietí týmto algoritmom je jeho všestrannosť – môže byť úspešne použitý na širokú škálu typov úloh, ako sú rozpoznávanie vzorov [8] až po riadenie a rozhodovanie. Ďalšími ich výhodami je jednoduchosť a možná paralelizácia pri použití dávkového učenia. To je dosiahnuté rozdelením tréningových dát do skupín a

pridelení vláknám, ktorých úpravy váh sú po každej iterácii spočítané.

Medzi hlavné nevýhody tohto algoritmu patrí jeho rýchlosť konverencie, ktorá môže byť značne nízka, aj keď táto nevýhoda môže byť čiastočne odstránená spomínanou paralelizáciou (a napr. následným použitím GPU). Pri veľkých datasetoch môže stále vznikať problém, čo býva v protiklade s požiadavkou, aby množstvo vstupných dát bolo primerane veľké na spoľahlivé naučenie siete. Naviac, konverencia pri spätnom šírení chýb nie je vždy zaručená a často môže skončiť v lokálnom minime, miesto optimálneho globálneho.

2.3 Hierarchické neurónové siete

Ako už bolo spomínané, architektúra neurónových sietí je inšpirovaná existujúcimi biologickými architektúrami, ale prezentované štruktúry (neurón, učenie apod.) skôr fungujú na takpovediac nižšej úrovni. Iným prípadom sú celé architektúry neurónových sietí (tzv. hierarchické [5]), ktoré sú skonštruované za použitia jednoduchších jednotiek alebo ich zoskupení, pričom učenie prebieha na globálnej úrovni, ale aj na lokálnych úrovniach. Tieto architektúry boli použité v funkčnom aj štruktúrnom designe neurónových sietí [6].

V funkčnom hierarchickom designe je úloha, ktorú má sieť byť naučená, rozdelená do menších častí, z ktorých každá je pridelená jednej sieti. Konečný výsledok učenia následne vznikne kombináciou čiastkových výsledkov. Vývoj v tejto oblasti prebieha najmä v tom, ako rozdeľovať jednotlivé úlohy a to napríklad extrakcia komponentov, minimizácia vzájomného pôsobenia podúloh a rozdelenie bázy znalostí a častí na učenie.

V štruktúrnom hierarchickom designe je podstata v spojení menších podsietí do výslednej siete. Členské podsiete môžu byť homogénne (ako napríklad v ensemble learning [7]) alebo heterogénne (hybridné učenie). Touto metódou môže byť takpovediac vytvorená “sieť sietí“, ktorá môže úlohu vyriešiť lepšie, ako niektorá z podsietí, prípadne riešiť úlohy, ktoré by podsiete nevedeli riešiť vôbec.

Podľa prísnej taxonómie sú aj niektoré základné modely neurónových sietí hierarchickými modelmi, ako príklad môžeme uviesť viacvrstvový perceptrón. Pri nich však vzniká

problém v robustnosti a učení na veľkých datasetoch a z dôvodu uniformnosti spojení vnútri siete nemôže byť použitá štruktúra učených znalostí na modifikáciu samotného modelu. Pri datasetoch s veľkou dimenzionalitou a veľkým počtom vstupných vzoriek je v sieťach s podobnou architektúrou potrebný veľký počet neurónov a čas tréovania je obvykle veľmi dlhý. Toto je čiastočne možné vyriešiť napríklad použitím výpočtov na grafických kartách [8].

3. Konvolučné neurónové siete

Konvolučné neurónové siete patria do kategórie hierarchických modelov. Tvoria vylepšenie viacvrstvových perceptrónov, pričom sú neurobiologicky inšpirované. Idea pre ich konštruovanie pochádza z vizuálnej kôry mačky, kde boli pomocou experimentu nájdené neuróny, ktoré sú citlivé na umiestnenie a orientáciu stimulov, pričom tvoria akúsi mapu reprezentujúcu vizuálny vstup pomocou detekcie hrán [9]. Konvulčná sieť je teda schopná extrakcie vlastností pomocou obmedzovania váh z jednej vrstvy do lokálneho receptívneho poľa v predchádzajúcej vrstve.

3.1 Princípy konvulčných neurónových sietí

Aby bolo možné prikročiť k popisu architektúr konvulčných neurónových sietí, v nasledujúcich častiach sú popísané princípy, ktoré sú v nich použité. Konkrétne ide o receptívne polia, konvolúciu, normalizáciu kontrastu, zdieľanie váh a subsampling.

3.1.1 Receptívne polia

Receptívne polia vizuálnych neurónov sú definované ako oblasti na sietnici, resp. vizuálneho priestoru, v ktorých prítomnosť alebo zánik stimulu impulzného charakteru spôsobí zmenu akčného potenciálu [27]. Zjednodušene povedané, receptívne pole neurónu A je oblasť všetkých neurónov, ktoré môžu ovplyvniť neurón A. Toto je charakteristické pre vizuálny systém človeka, pričom napríklad v primárnej vizuálnej kôre označovanej ako V1 sa nachádzajú receptívne polia citlivé na orientáciu. Táto oblasť mozgu je najlepšie preskúmaná z celého vizuálneho systému. Prebieha v nej spracovanie pohyblivých aj nepohyblivých objektov a rozpoznávanie vzorov. Obvykle čím “vyššiu” úroveň má vizuálna oblasť, tým špecializovanejšie má receptívne polia, (napr. inferior temporal cortex je citlivá na tváre).

3.1.2 Konvolúcia

Konvolúcia je operácia na dvoch funkciách f a g , pričom je výsledkom je tretia funkcia,

ktorá môže byť považovaná za filtrovanú funkciu f , funkcia g teda môže byť považovaná za filter [28]. Konvolúcia dvoch funkcií $f(x)$ a $g(x)$ pre spojitú premennú x je vyjadrená v zápise:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) g(x-\tau) d\tau ,$$

kde $*$ je operátor konvolúcie a τ je voľná premenná. Pre funkcie diskretných premenných, napríklad polia čísel, je jej zápis takýto:

$$f[x] * g[x] = \sum_{i=-\infty}^{\infty} f[i] g[x-i] .$$

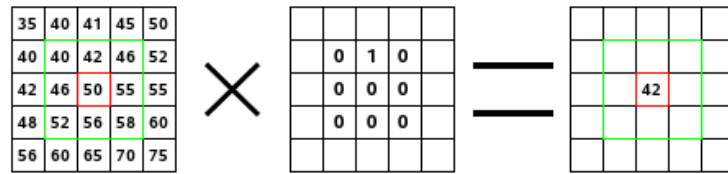
Pre funkcie dvoch premenných funkcií $f(x, y)$ a $g(x, y)$ má konvolúcia takúto formu v prípade spojitých premenných:

$$f(x, y) * g(x, y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1, \tau_2) g(x-\tau_1, y-\tau_2) d\tau_1 d\tau_2 ,$$

resp. v prípade diskretných premenných:

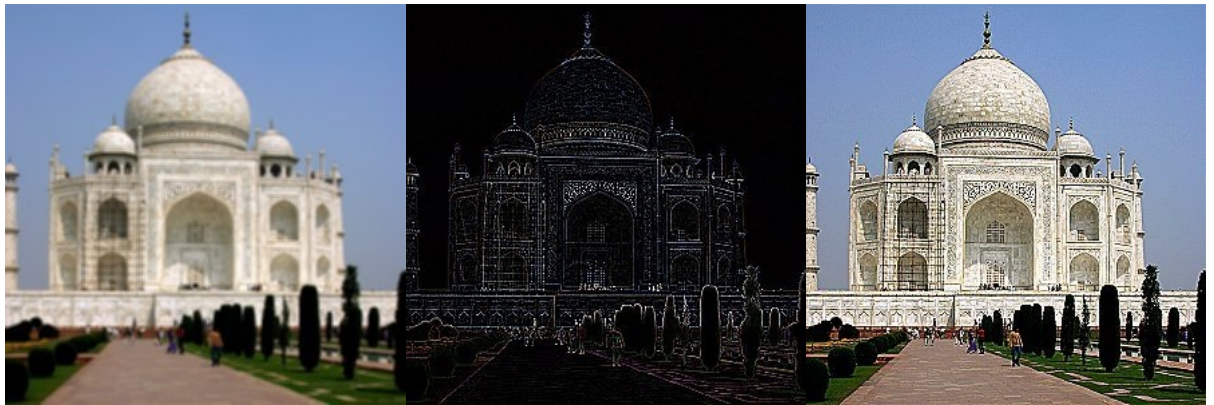
$$f[x, y] * g[x, y] = \sum_{i_1=-\infty}^{\infty} \sum_{i_2=-\infty}^{\infty} f[i_1, i_2] g[x-i_1, y-i_2] d\tau_1 d\tau_2 .$$

Pre lepšie pochopenie si môžeme konvolúciu predstaviť tak, ako je používaná v praxi na spracovanie obrazu, kde ide o operáciu dvoch matic. Prvú maticu tvorí spracovaný obraz, druhú tvorí matica filtra, ktorá je nazývaná aj kernel, pričom je štvorcového tvaru a má rozmer, ktorý môžeme nazvať aj okolie a je nepárny. Kernel je umiestnený “na” obraz a výsledná hodnota je vypočítaná nasledujúcim spôsobom: základným bodom výpočtu je bod v obraze, ktorý prislúcha stredovému bodu kernelu. V prípade kernelu rozmerov 3x3 berieme do úvahy teda ešte 8 bodov. Pre hodnoty všetkých 9 zodpovedajúcich bodov kernelu a obrazu je vypočítaný súčin a následne súčet týchto súčinov podľa vyššie uvedených vzťahov. Ten je umiestnený do základného bodu, prípadne ešte vynásobený koeficientom. Toto je následne vykonané pre každý bod obrazu. Ilustrácia tohto postupu je zobrazená na obr. 6.



Obrázok 6: Predstava konvolúcie na obrazových bodoch

Konvolúcia je používaná na vyhladzovanie alebo zostrovanie, v prípade dvojrozmerných funkcií, ako napríklad obrázkov, aj na hľadanie hrán, detekciu vlastností, detekciu pohybu a iné rôzne úlohy. Ukážky konvolučných filtrov tak, ako môžu byť použité v programe GIMP aj s príslušnými kernelmi, sú na obr. 7 až 9 [29].



Obrázok 7: Aplikácia konvolučného filtra - rozostrenie (blur)

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Obrázok 8: Aplikácia konvolučného filtra - detekcia hrán

	0	1	0	
	1	-4	1	
	0	1	0	

Obrázok 9: Aplikácia konvolučného filtra - zostrenie obrazu

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

3.1.3 Konvolúcia a separovateľný kernel

Na konštruovanie 2D kernelu môže byť použitý 1D kernel tak, aby výsledný kernel bol separovateľný. To znamená, že matica kernelu rozmeru $M \times M$ môže byť rozložená na dve matice rozmerov $M \times 1$ a $1 \times M$ nasledovným spôsobom (príklad pre maticu 3×3) [30]:

$$\begin{bmatrix} A.a & A.b & A.c \\ B.a & B.b & B.c \\ C.a & C.b & C.c \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \cdot [a \ b \ c] ,$$

kde '.' je operácia násobenia. Potom môžeme konvolúciu bodu $x[m,n]$ 2D kernelom zapísať takto:

$$x[m,n] * \begin{bmatrix} A.a & A.b & A.c \\ B.a & B.b & B.c \\ C.a & C.b & C.c \end{bmatrix} = x[m,n] * \left(\begin{bmatrix} A \\ B \\ C \end{bmatrix} \cdot [a \ b \ c] \right) = \left(x[m,n] * \begin{bmatrix} A \\ B \\ C \end{bmatrix} \right) * [a \ b \ c]$$

, kde '*' je operátor konvolúcie a '.' operátor násobenia. Z uvedeného je zrejmé, že namiesto jednej 2D konvolúcie je prevádzaná dvakrát operácia 1D konvolúcie, čím je významne ušetrený čas procesora potrebný na vykonanie konvolúcie bodu. Pri konvolúcii 2D kernelom veľkosti $N \times N$ je vykonaných $N \times N$ operácií, pri konvolúcii tým istým separovaným kernelom je počet operácií len $N+N$, čiže zrýchlenie operácie je v pomere $N^2/2N$.

3.1.4 Priestorová konvolúcia

Táto operácia konvolvuje vstupný obraz zadaným množstvom náhodných kernelov, pričom jej výsledkom sú mapy príznakov [31]. Jej parametrami sú ešte rozmery kernelov N a krok konvolúcie po obraze v oboch smeroch W a H . Rozmery výsledných obrazov sú potom

$$\begin{aligned} \text{šírka} &= \frac{(W_0 - N)}{W} + 1 \\ \text{výška} &= \frac{(H_0 - N)}{H} + 1 \end{aligned} ,$$

kde W_0 a H_0 sú šírka a výška spracovávaného obrazu. Výsledok aplikácie tejto funkcie na obraz v praxi je na obr. 10.



Obrázok 10: Obráz po štvornásobnej aplikácii operácie priestorovej konvolúcie

3.1.5 Priestorová subtraktívna normalizácia

Táto funkcia slúži na odstránenie nadbytočných častí z obrazu a tým zvýraznenie príznakov v obraze [31]. To je dosiahnuté istou formou lokálneho súťaženia medzi vlastnosťami v mape príznakov a vlastnosťami v rovnakom umiestnení, na čo je použitý kernel tvaru Gaussiánu, ktorým sú vypočítané vážené priemery bodov v okolí. Gaussián je použitý z dôvodu, že korelácia dvoch bodov klesá s ich vzrastajúcou vzájomnou vzdialenosťou. Na konštruovanie kernelu je použitý 1D gaussián tak, aby bol výsledný kernel separovateľný.

Praktický príklad takejto operácie na obraze je na obr. 11.



Obrázok 11: Naľavo pôvodný obraz, napravo po operácii normalizácie kontrastu

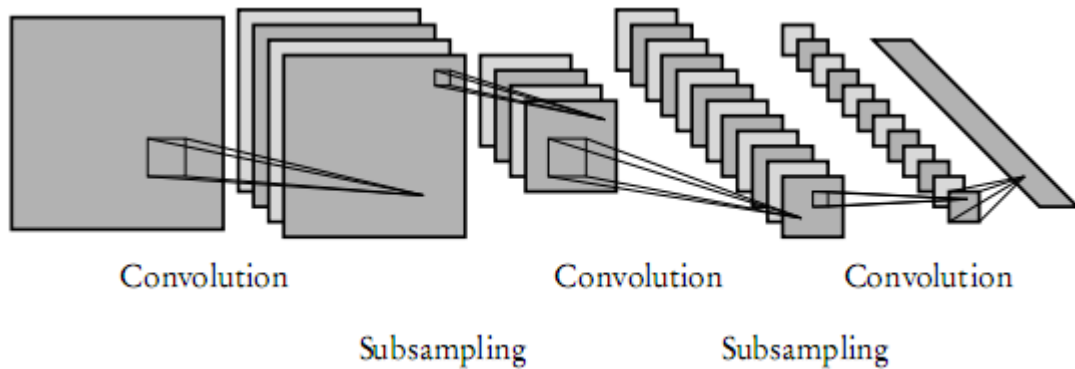
3.1.6 Zdieľanie váh

Konvolučná sieť využíva skutočnosť, že detektory príznakov sú po natrénovaní použiteľné na celej ploche vstupného obrazu, takže môže byť použitý postup zdieľania váh. Ide o množinu neurónov, ktoré tvoria tieto detektory na rôznych miestach a majú rovnaké (zdieľané) váhové vektory [25]. Ak by sme chceli sekvenčnú implementáciu tohto postupu, bola by približne nasledovná: máme jeden neurón s lokálnym receptívnym poľom, pričom týmto prechádzame vstupný obraz a uchováваме jeho stavy pre rôzne miesta. Táto operácia tvorí konvolúciu vstupného obrazu s určitou veľkosťou kernelu. Tento proces môže byť následne implementovaný tak, že nahradíme jediný neurón vrstvou neurónov, ktoré zdieľajú jeden váhový vektor. Ako výsledok je v druhej vrstve získaná takzvaná mapa príznakov (feature map), pričom konvolučná vrstva je obvykle tvorená viacerými takýmito mapami. Každá z máp má odlišný váhový vektor, aby mohli byť extrahované na každom mieste rozdielne príznaky. Posun príznaku na vstupe konvolučnej vrstvy spôsobí posun vo výstupe, ale tento inak ostane nezmenený. Po rozpoznaní príznaku sa stane jeho umiestnenie málo dôležité, keďže je rozhodujúce jeho umiestnenie vzhľadom k ostatným príznakom.

3.1.7 Subsampling

Výstup konvolučnej vrstvy je spracovávaný ďalej – je zredukované jeho priestorové rozlíšenie (t.j. subsampling), čím je dosiahnutý ďalší stupeň invariance umiestnenia a skreslenia. Toto je realizované v ďalších vrstvách, nazývaných vzorkovacie. Môže to byť pomocou jednej z techník average pooling alebo max pooling. Každá z nich pracuje znova s okolím bodu, pričom centrá týchto okolí sú navzájom od seba posunuté o viac ako 1 bod, ale menej alebo rovnako ako veľkosť okolia. Ako už napovedajú názvy, výstupom prvej techniky je vždy priemer daných bodov okolia a výsledkom druhej je maximálna hodnota z bodov v okolí. Porovnania týchto dvoch spôsobov odhaľujú, že max pooling je značne úspešnejší [10].

3.2 Architektúra konvolučných neurónových sietí



Obrázok 12: Predstava architektúry konvolučnej neurónovej siete

Jednoduchá konvolučná sieť je znázornená na obr. 12 [32]. V prvej konvolučnej vrstve sa nachádzajú mapy príznakov, v ktorej majú neuróny receptívne polia veľkosti 5x5 pixelov z vstupného obrazu. Výsledné mapy príznakov majú mierne menšie rozlíšenie ako vstupy, kvôli nespracovávaniu ich okrajov. Nasleduje vzorkovacia vrstva, obsahujúca mapy príznakov, ktoré sú škálovanými lokálnymi priermi príslušných máp príznakov v predchádzajúcej vrstve. Každý neurón v tejto vrstve má jeden trénovateľný parameter pre priemer a jeden trénovateľný bias. Môžu nasledovať ďalšie konvolučné a vzorkovacie vrstvy, fungujúce takisto ako predchádzajúce, obvykle však majú viac príznakových máp, čo vedie k zväčšeniu príznakového priestoru a vzorkovacia vrstva znova vedie k ďalšiemu zníženiu rozlíšenia. Ako výstupná vrstva je obvykle použitá plne prepojená dopredná neurónová sieť, napr. viacvrstvový perceptrón.

3.3 Trénovanie konvolučných sietí

Keďže konvolučné siete tvoria špecializáciu viacvrstvových perceptrónov, môžu byť trénované pomocou techniky gradient descent. Oproti nim majú významnú výhodu a síce, že neuróny v jednej mape príznakov zdieľajú váhy a bias, čím je značne znížený počet trénovaných parametrov, v kontraste k viacvrstvovým perceptrónom, ktoré sú plne prepojené. Vzorkovacie vrstvy majú len po jednej váhe, ktorá môže byť trénovaná (čo je

vlastne koeficient lokálneho priemeru) a jeden trénovateľný bias, čiže výsledný počet parametrov v týchto vrstvách je ešte menší, ako v konvolučných vrstvách. Malý počet voľných parametrov vedie k významne menšej potrebe výpočtového výkonu pri trénovaní ako pri viacvrstvových perceptrónoch. Vo výsledku sú konvolučné siete veľmi vhodné na implicitnú extrakciu vlastností a v istej miere invariance skreslenia a teda na klasifikačné úlohy, najmä rozpoznávanie vzorov. Veľmi úspešne boli použité napríklad na rozpoznávanie rukou písaných písmen, číslíc a iných znakov alebo rozpoznávanie tvárí.

3.3.1 Boosting

Boosting je metóda, ktorá je používaná na vylepšovanie výsledkov techník strojového učenia [33]. Využíva k tomu tzv. komisie expertov, ktoré majú za úlohu spolu rozhodnúť o konečnom riešení podľa ich spoločných znalostí, pričom úloha určená na naučenie je medzi nich rozdelená. Jeden z možných postupov je takýto: najprv je natrénovaný prvý expert (v našom prípade sieť) na celom vstupnom datasete a sú uchované vstupné vzorky, ktoré klasifikoval nesprávne. Tieto sú následne použité ako polovica vstupného datasetu pre druhého experta, pričom druhá polovica je získaná z správne klasifikovaných dát. Po natrénovaní druhej siete sú dáta testované na prvej a druhej sieti; ak niektorá z nich klasifikuje vzor nesprávne, tento je zaradený do trénovacieho datasetu pre tretiu sieť. Znova sú vybrané dáta z správne klasifikovaných ako druhá polovica pridaná k nesprávnym. Konečne je celkový výsledok vypočítaný ako spojenie troch čiastkových výsledkov dokopy.

3.4 Použitie konvolučných sietí

Keďže sú konvolučné siete používané hlavne na rozpoznávanie obrazu, uvediem ako príklady práce, ktoré boli vyrvorené práve v tejto oblasti.

MNIST [20] je databáza ručne napísaných číslíc. Obsahuje celkom 70000 vzorov, ktoré sú rozdelené v pomere 6:1 v prospech trénovacej množiny, pričom ich rozmery sú 32x32 pixelov. Jednotlivé čísla majú normalizovanú veľkosť a sú centrované v svojich obrázkoch. Táto je častým cieľom rozpoznávania najrôznejšími technikami strojového učenia, ako

napríklad Support Vector Machines (SVM), K-nearest neighbors alebo rôzne druhy neurónových sietí, medzi nimi aj konvolučné, ktoré v dobe písania tejto práce (5/2012) dosahovali najlepšie výsledky na testovacej chybe. Najlepší výsledok celkovo mala komisia sietí používajúca boosting [21] a to 0.23%, čo už je blízko limitu rozpoznávania na tejto množine (0.23% z 10000 testovacích vzorov je len 23 chybné rozpoznávaných). Naproti tomu, najlepšia samostatná konvolučná sieť dosahuje chybu 0.35% (s predspracovaním). Zaujímavé sú výsledky viacvrstvových perceptrónov – veľká 7-vrstvová sieť implementovaná na GPU [8] dosahuje výborných 0.35%, čo čiastočne potvrdzuje ich univerzálnosť.

Zaujímavým pokusom je rozpoznávanie nefunkčných batérií, ktoré treba z dôvodov ďalšieho spracovania triediť [19]. Autor porovnáva dve architektúry neurónových sietí – prvá z nich je konvolučná sieť, druhá z nich je tzv. Deep Belief Network (DBN), obe implementované na GPU. Ako vstupy im teda slúžili obrázky batérií v formáte RGB, ktoré vznikli rotovaním, v celkovom počte 10000. V testoch bola výrazne úspešnejšia konvolučná architektúra, keď jej najlepší výsledok mal hodnotu 0.08% oproti DBN s hodnotou 2.89%. Značné rozdiely boli aj v čase tréovania, ktoré boli 52 minút v prípade CNN a 147 minút v prípade DBN.

4. Praktická časť

V tejto kapitole sú uvedené výsledky testov, ktoré boli prevádzané na použitých datasetoch. Zároveň sú tu uvedené porovnania s inou biologicky inšpirovanou technikou (nielen) na rozpoznávanie vzorov a to Deep Belief Network (DBN) [24].

4.1 Experiment s datasetom listov

Na trénovanie bolo použitých 5 typov datasetov. Najprv sú uvedené výsledky pre každý z nich oddelene, potom nasleduje ich porovnanie. Datasets, ktoré boli použité na trénovanie konvolučnej siete v tejto práci, sú identické s tými, ktoré boli použité na trénovanie DBN.

4.1.1 Experimentálna zostava

Na trénovanie bola použitá viacvrstvová konvolučná sieť implementovaná pomocou knižnic Torch7 [31]. Boli v nej použité rámcovo tri vrstvy, pričom popis jednotlivých vrstiev nasleduje v ďalšom texte:

Prvá vrstva slúži na normalizáciu kontrastu, pričom použitý kernel je Gausián s okolím rozmeru 15. Nasledujúca vrstva je prvá konvolučná za použitia kernelu rozmeru 5x5. Počet konvolučných rovín je parametrom v tréovaní. Ďalšia vrstva obsahuje funkcie *tanh*, ktoré majú za úlohu spraviť výstup nelineárnym. Nasleduje vrstva poolingu (viď kap. 3.1.7), pričom rozmer okolia je 2x2, krok pohybu po mape je takisto 2 v oboch smeroch. Táto architektúra je zopakovaná, avšak s rozdielom v konvolučnej vrstve. Táto je prepojená na rozdiel od prvej len čiastočne tak, že každý výstup má určený počet prichádzajúcich spojení. Keďže sú spojenia vytvárané náhodne, pri vytváraní siete je snaha o pokiaľ možno čo najväčšie priblíženie k uniformnému rozdeleniu týchto spojení.

Nasledujúce vrstvy sú výstupné. Ako prvá z nich je vrstva na priestorovú transformáciu výstupu. Tá má za úlohu previesť výstup, ktorý bol do tohto miesta v sieti vždy dvojrozmerný, do jedného rozmeru, pričom počet výstupných parametrov je takisto parametrom. Tento je privedený na vstup lineárnej vrstvy, čo je “obyčajná” plne prepojená vrstva tak, ako je známa z viacvrstvého perceptrónu. Tu je zmenšený počet výstupov na

d'alší parameter, ktorým je počet neurónov s aktivačnou funkciou *tanh* v nasledujúcej vrstve. Výstup siete tvorí lineárna vrstva s počtom výstupných neurónov rovným počtu rozpoznávaných tried. Ako maximum epoch bolo nastavených 60.

Na trénovanie siete bola použitá optimalizačná metóda Stochastic Gradient Descent (SGD), ktorá je určená na trénovanie klasifikátorov na veľkých datasetoch optimalizáciou funkcie. Túto by malo byť možné zapísať ako sumu derivovateľných funkcií. Na úpravu parametrov používa len jednu náhodne vybranú vzorku z datasetu, pričom posun po gradiente funkcie je teda náhodnou aproximáciou skutočného posunu založeného na všetkých trénovacích dátach. Výhodou postupu je veľká rýchlosť konverencie, pre porovnanie je vhodný napr. zdroj [34].

4.1.2 Dataset

Dataset listov [24] obsahuje 11 tried stromov, ktoré sú na obr. 13.



Obrázok 13: Triedy listov použité v datasete

Každý strom má v datasete 5 rôznych listov. Použité sú vo viacerých verziách datasetu, ktoré sú od seba navzájom odlišné parametrami jednotlivých listov v datasetoch. Tie sa môžu líšiť v rotácii, veľkosti, šume a jase. Parametre majú nasledovné charakteristiky:

Rotácia

Listy môžu byť rotované okolo troch osí. Podľa osí *x* a *y* sú rotované od určených hraníc. Podľa osi *z* sú rotované po kružnici s krokom $5 + \text{náhodné číslo od } 0 \text{ po } 1$.

Veľkosť

Listy môžu byť vzdialené od kamery v určenom intervale s krokováním.

Šum

Listy môžu byť zašumené s určeným levelom, pričom šum môže byť použitý na celý obraz alebo len na pozadie. Farba šumu je náhodne vybraná z určeného intervalu.

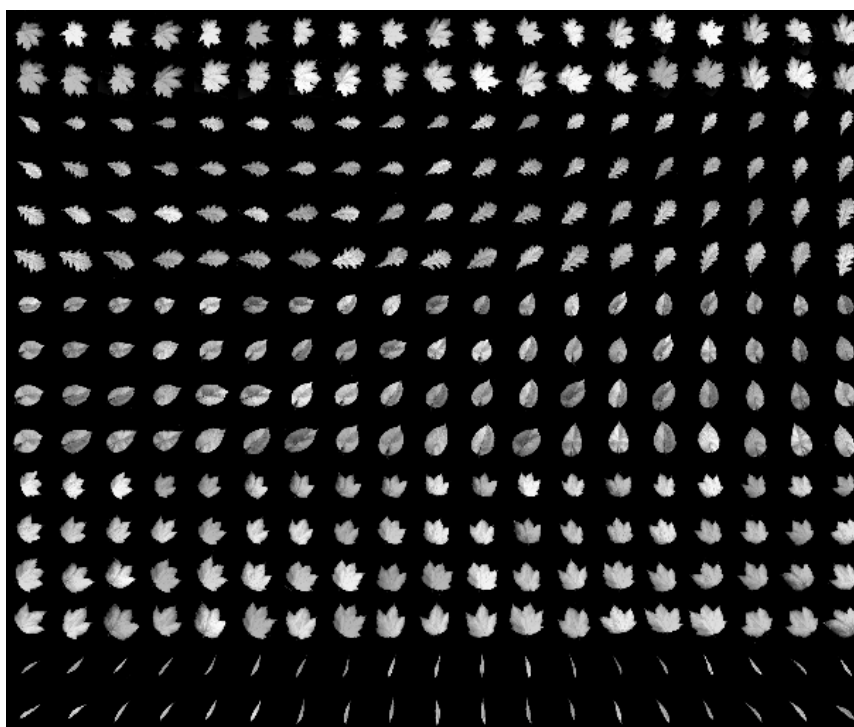
Jas

Listy môžu mať zmenený jas podľa určeného intervalu.

Na základe zmeny týchto parametrov bolo vygenerovaných 5 datasetov, ktorých popis s ukážkami nasleduje v ďalšom texte.

Dataset 1

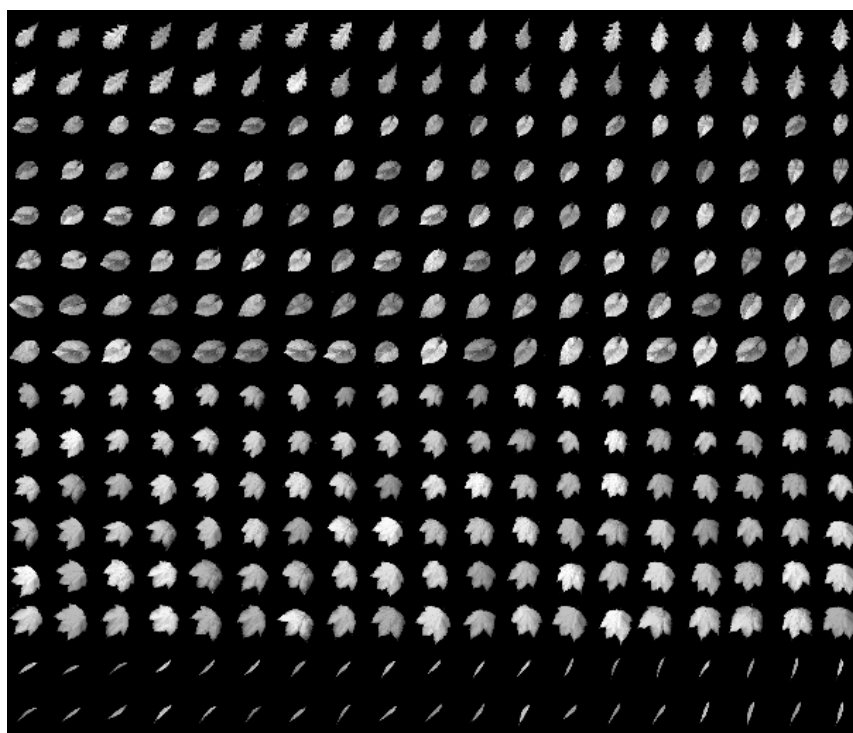
V tomto datasete sú použité ako zdroj na jeho vygenerovanie všetky listy. Ukážka tohto datasetu je na obr. 14.



Obrázok 14: Ukážka datasetu 1

Parametre datasetu:**Rotácia:** náhodne, $x(-30, 30)$, $y(-30, 30)$, z s krokom 5**Veľkosť:** $(-4, -2.8)$, krok 0.1**Šum:** bez šumu**Jas:** $(0.7, 1.0)$ **Počet vzorov:** 2000 tréningových, 904 testovacích**Dataset 2**

V tomto datasete sú použité ako zdroj na jeho vygenerovanie všetky listy. Ukážka tohto datasetu je na obr. 15.



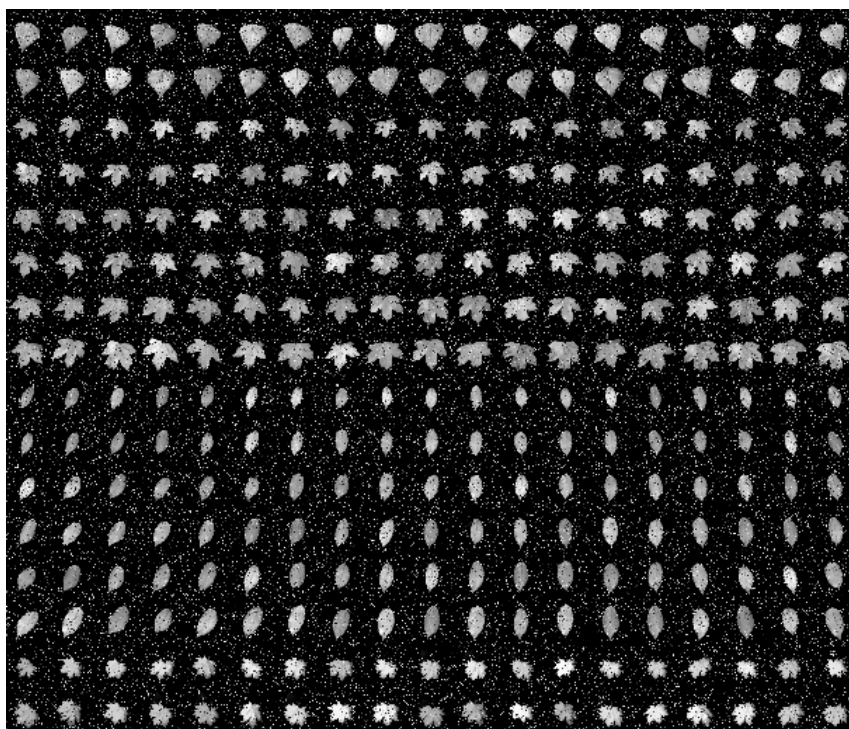
Obrázok 15: Ukážka datasetu 2

Parametre datasetu:**Rotácia:** náhodne, $x(-30, 30)$, $y(-30, 30)$, z s krokom 2**Veľkosť:** $(-4, -2.8)$, krok 0.06

Šum:	bez šumu
Jas:	(0.7, 1.0)
Počet vzorov:	8000 trénovacích, 1504 testovacích

Dataset 3

V tomto datasete sú použité ako zdroj na jeho vygenerovanie všetky listy. Ukážka tohto datasetu je na obr. 16.



Obrázok 16: Ukážka datasetu 3

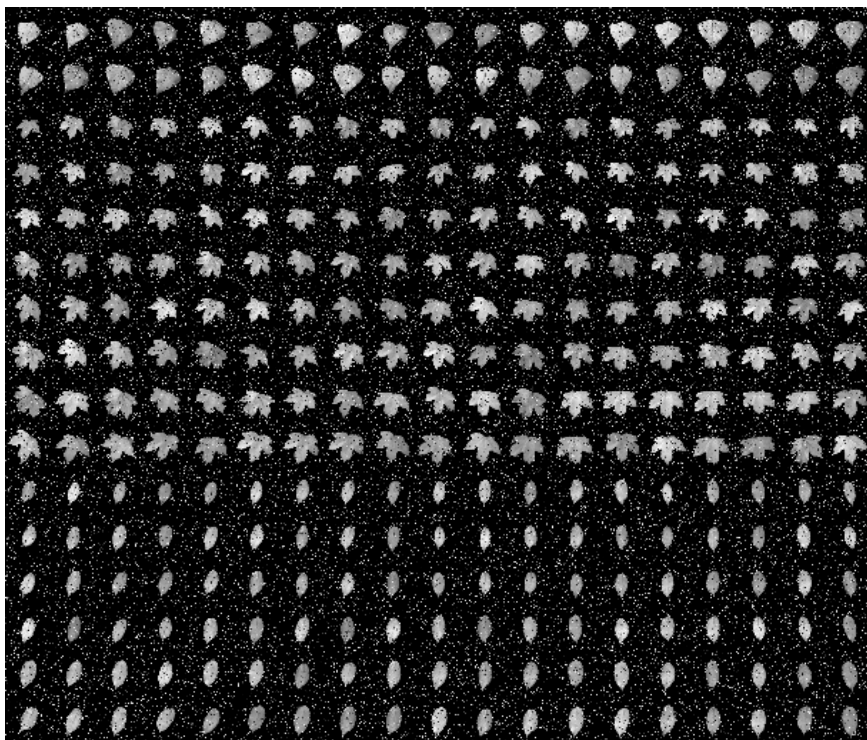
Parametre datasetu:

Rotácia:	náhodne, $x(-30, 30)$, $y(-30, 30)$, z s krokom 2
Veľkosť:	$(-4, -2.8)$, krok 0.06
Šum:	farba náhodná z $(0.0, 1.0)$, úroveň šumu 10%
Jas:	(0.7, 1.0)

Počet vzorov: 8000 trénovacích, 1504 testovacích

Dataset 4

V tomto datasete sú použité ako zdroj na jeho vygenerovanie všetky listy. Ukážka tohto datasetu je na obr. 17.



Obrázok 17: Ukážka datasetu 4

Parametre datasetu:

Rotácia: náhodne, $x(-30, 30)$, $y(-30, 30)$, z s krokom 1

Veľkosť: $(-4, -3)$, krok 0.04

Šum: farba náhodná z $(0.0, 1.0)$, úroveň šumu 10%

Jas: $(0.7, 1.0)$

Počet vzorov: 17000 trénovacích, 4120 testovacích

Dataset 5

V tomto datasete sú použité listy takto: na generovanie trénovacích vzorov boli použité 4 listy z každého druhu stromu. Na generovanie testovacích vzorov bol použitý zvyšný 1 list.

Parametre trénovacieho datasetu:

Rotácia: náhodne, $x(-30, 30)$, $y(-30, 30)$, z s krokom 1

Veľkosť: $(-4, -3)$, krok 0.04

Šum: farba náhodná z $(0.0, 1.0)$, úroveň šumu 5%

Jas: $(0.7, 1.0)$

Parametre testovacieho datasetu:

Rotácia: náhodne, $x(-30, 30)$, $y(-30, 30)$, z s krokom 5

Veľkosť: $(-4, -3)$, krok 0.04

Šum: farba náhodná z $(0.0, 1.0)$, úroveň šumu 5%

Jas: $(0.7, 1.0)$

Počet vzorov: 21120 trénovacích, 5808 testovacích

4.1.3 Výsledky – menšie konfigurácie**Dataset 1**

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.01	11.615	53
6	12	120	60	0.01	10.73	50
6	18	60	60	0.01	11.726	44
6	18	120	60	0.01	11.394	36
12	18	60	60	0.01	7.965	50
12	18	120	60	0.01	8.186	53
12	24	60	60	0.01	9.735	56
12	24	120	60	0.01	9.624	29
18	40	60	60	0.01	8.96	59
18	40	120	60	0.01	7.522	37

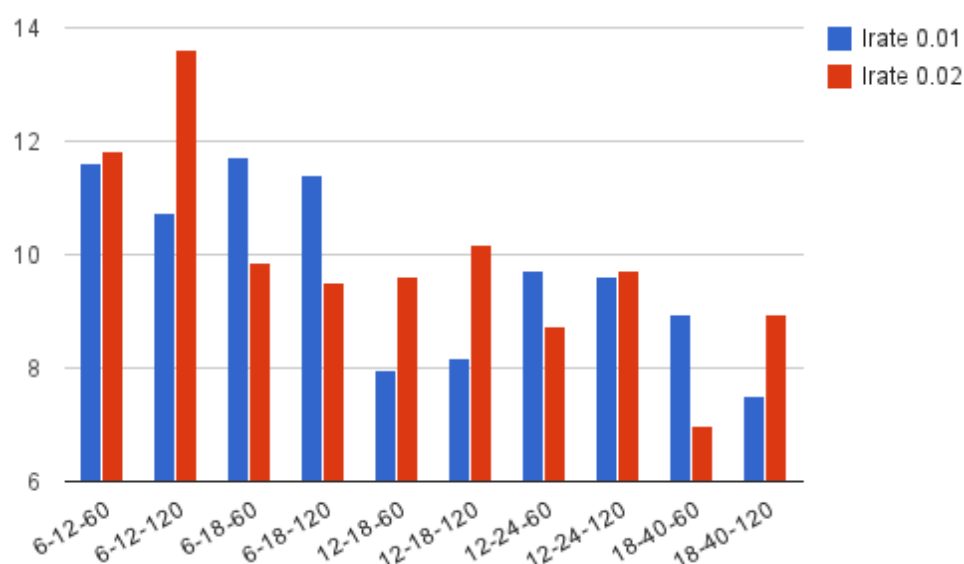
Tabuľka 1: Dataset 1 - konfigurácie s rýchlosťou učenia 0.01

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.02	11.836	59
6	12	120	60	0.02	13.606	59
6	18	60	60	0.02	9.845	46
6	18	120	60	0.02	9.513	54
12	18	60	60	0.02	9.624	28
12	18	120	60	0.02	10.177	60
12	24	60	60	0.02	8.739	33
12	24	120	60	0.02	9.735	7
18	40	60	60	0.02	6.969	57
18	40	120	60	0.02	8.96	55

Tabuľka 2: Dataset 1 - konfigurácie s rýchlosťou učenia 0.02

Poznámka: v nasledujúcom texte bude ako chyba myslené percento chybné klasifikovaných obrazov z testovacej množiny, pokiaľ nebude uvedené inak. Najlepšia chyba je v tabuľkách vyznačená.

Najlepší výsledok siete na tomto datasete je 7.522% pri konfigurácii 18-40-120 a rýchlosti učenia 0.01. DBN dosiahla najlepší výsledok 16.4%.



Obrázok 18: Dataset 1 - závislosť testovacej chyby od rýchlosti učenia

Pri tejto konfigurácii nemôže byť jednoznačne určené, ako vplýva rýchlosť učenia na

výslednú chybu. Je to pravdepodobne spôsobené malým počtom tréovacích vzorov, čo sa prejavuje aj na výsledných chybách, ktoré sa pohybujú od 7 do 12%. Tento výsledok nie je úplne uspokojivý pre praktické použitie.

Dataset 2

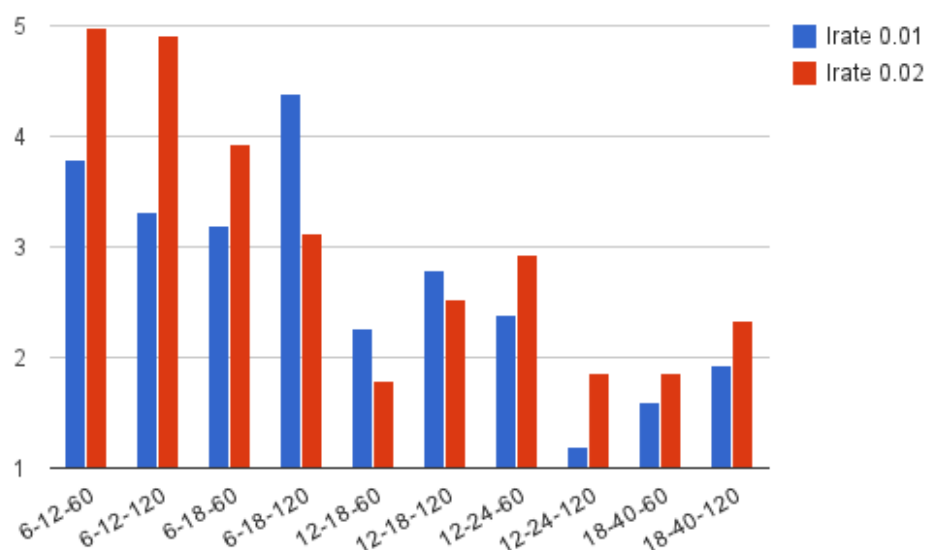
Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.01	3.79	33
6	12	120	60	0.01	3.324	59
6	18	60	60	0.01	3.191	48
6	18	120	60	0.01	4.388	54
12	18	60	60	0.01	2.261	58
12	18	120	60	0.01	2.793	50
12	24	60	60	0.01	2.394	30
12	24	120	60	0.01	1.197	43
18	40	60	60	0.01	1.596	59
18	40	120	60	0.01	1.928	60

Tabuľka 3: Dataset 2 - konfigurácie s rýchlosťou učenia 0.01

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.02	4.987	57
6	12	120	60	0.02	4.92	38
6	18	60	60	0.02	3.923	59
6	18	120	60	0.02	3.125	56
12	18	60	60	0.02	1.795	58
12	18	120	60	0.02	2.527	58
12	24	60	60	0.02	2.926	56
12	24	120	60	0.02	1.862	14
18	40	60	60	0.02	1.862	50
18	40	120	60	0.02	2.327	21

Tabuľka 4: Dataset 2 - konfigurácie s rýchlosťou učenia 0.02

Najlepší výsledok siete na tomto datasete je 1.197% pri konfigurácii 12-24-120 a rýchlosti učenia 0.01. DBN dosiahla najlepší výsledok 4.7%.



Obrázok 19: Dataset 2 - závislosť testovacej chyby od rýchlosti učenia

Dataset 3

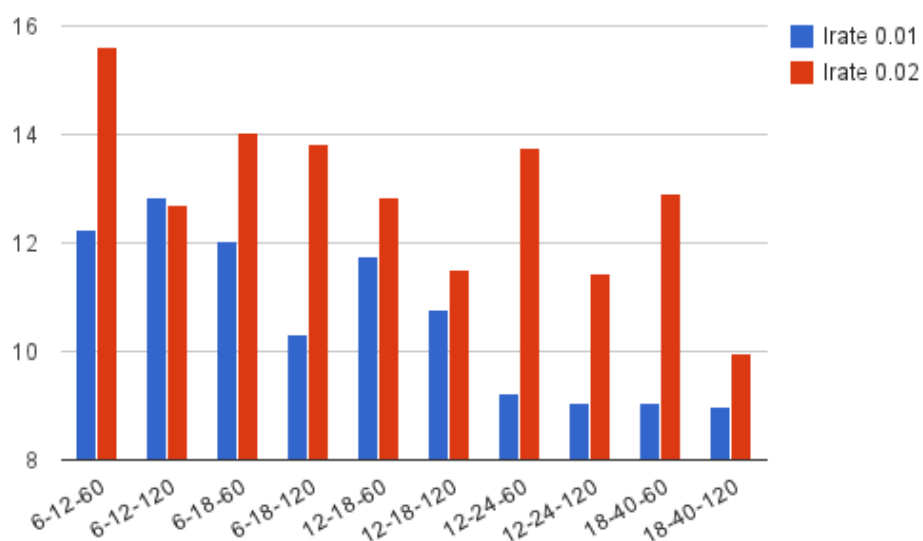
Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.01	12.234	31
6	12	120	60	0.01	11.835	16
6	18	60	60	0.01	12.035	18
6	18	120	60	0.01	10.306	21
12	18	60	60	0.01	11.769	48
12	18	120	60	0.01	10.771	10
12	24	60	60	0.01	9.242	12
12	24	120	60	0.01	9.043	20
18	40	60	60	0.01	9.043	54
18	40	120	60	0.01	8.976	33

Tabuľka 5: Dataset 3 - konfigurácie s rýchlosťou učenia 0.01

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.02	15.625	36
6	12	120	60	0.02	12.699	55
6	18	60	60	0.02	14.029	53
6	18	120	60	0.02	13.83	45
12	18	60	60	0.02	12.832	51
12	18	120	60	0.02	11.503	41
12	24	60	60	0.02	13.763	38
12	24	120	60	0.02	11.436	20
18	40	60	60	0.02	12.899	32
18	40	120	60	0.02	9.973	56

Tabuľka 6: Dataset 3 - konfigurácie s rýchlosťou učenia 0.02

Najlepší výsledok siete na tomto datasete je 8.976% pri konfigurácii 18-40-120 a rýchlosti učenia 0.01. DBN dosiahla najlepší výsledok 19.6%.



Obrázok 20: Dataset 3 - závislosť testovacej chyby od rýchlosti učenia

Dataset 4

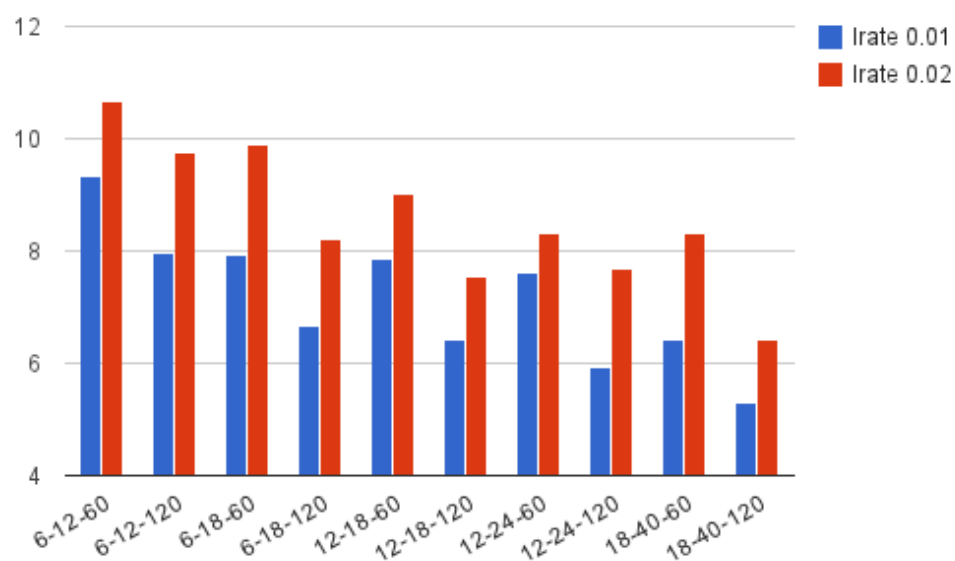
Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.01	9.345	18
6	12	120	60	0.01	7.961	60
6	18	60	60	0.01	7.937	60
6	18	120	60	0.01	6.675	48
12	18	60	60	0.01	7.864	46
12	18	120	60	0.01	6.432	53
12	24	60	60	0.01	7.597	58
12	24	120	60	0.01	5.922	55
18	40	60	60	0.01	6.432	58
18	40	120	60	0.01	5.291	51

Tabuľka 7: Dataset 4 - konfigurácie s rýchlosťou učenia 0.01

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.02	10.655	47
6	12	120	60	0.02	9.757	60
6	18	60	60	0.02	9.879	57
6	18	120	60	0.02	8.228	43
12	18	60	60	0.02	9.029	55
12	18	120	60	0.02	7.549	60
12	24	60	60	0.02	8.301	54
12	24	120	60	0.02	7.67	59
18	40	60	60	0.02	8.301	60
18	40	120	60	0.02	6.432	60

Tabuľka 8: Dataset 4 - konfigurácie s rýchlosťou učenia 0.02

Najlepší výsledok siete na tomto datasete je 5.291% pri konfigurácii 18-40-120 a rýchlosti učenia 0.01. DBN dosiahla najlepší výsledok 15%.



Obrázok 21: Dataset 4 - závislosť testovacej chyby od rýchlosti učenia

Dataset 5

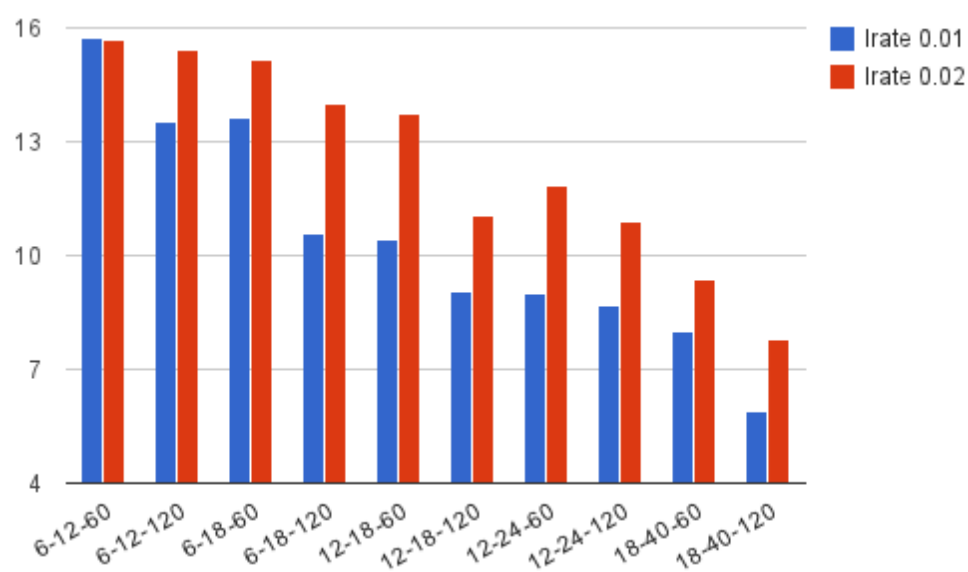
Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.01	15.479	55
6	12	120	60	0.01	13.55	60
6	18	60	60	0.01	13.619	60
6	18	120	60	0.01	10.554	60
12	18	60	60	0.01	10.417	57
12	18	120	60	0.01	9.074	60
12	24	60	60	0.01	8.988	60
12	24	120	60	0.01	8.66	60
18	40	60	60	0.01	8.023	60
18	40	120	60	0.01	5.906	60

Tabuľka 9: Dataset 5 - konfigurácie s rýchlosťou učenia 0.01

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	60	0.02	15.668	60
6	12	120	60	0.02	15.341	60
6	18	60	60	0.02	15.152	60
6	18	120	60	0.02	13.981	53
12	18	60	60	0.02	13.757	55
12	18	120	60	0.02	11.071	59
12	24	60	60	0.02	11.863	60
12	24	120	60	0.02	10.899	60
18	40	60	60	0.02	9.349	60
18	40	120	60	0.02	7.782	60

Tabuľka 10: Dataset 5 - konfigurácie s rýchlosťou učenia 0.02

Najlepší výsledok siete na tomto datasete je 5.906% pri konfigurácii 18-40-120 a rýchlosti učenia 0.01. DBN dosiahla najlepší výsledok 19.6%.



Obrázok 22: Dataset 5 - závislosť testovacej chyby od rýchlosti učenia

4.1.4 Výsledky - komplexnejšie konfigurácie

Keďže uvedené výsledky neboli úplne uspokojivé, skúsil som datasety otestovať na konfiguráciách siete, ktoré majú výrazne väčšie počty konvolučných rovín a neurónov v

výstupnej vrstve. Tentokrát som testoval pre maximálny počet epoch 30. Nasledujú výsledky:

Dataset 1

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
32	96	180	30	0.01	6.969	25
64	128	240	30	0.01	8.85	19
128	256	400	30	0.01	7.301	29
32	96	180	30	0.02	8.628	18
64	128	240	30	0.02	8.85	26
128	256	400	30	0.02	8.628	14

Tabuľka 11: Výsledky siete na datasete 1 (množina konfigurácií 2)

Dataset 2

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
32	96	180	30	0.01	1.729	18
64	128	240	30	0.01	1.263	29
128	256	400	30	0.01	1.596	29
32	96	180	30	0.02	2.128	17
64	128	240	30	0.02	1.33	29
128	256	400	30	0.02	1.529	24

Tabuľka 12: Výsledky siete na datasete 2 (množina konfigurácií 2)

Dataset 3

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
32	96	180	30	0.01	8.311	10
64	128	240	30	0.01	8.777	29
128	256	400	30	0.01	8.976	13
32	96	180	30	0.02	10.971	21
64	128	240	30	0.02	10.439	3
128	256	400	30	0.02	13.364	5

Tabuľka 13: Výsledky siete na datasete 3 (množina konfigurácií 2)

Dataset 4

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
32	96	180	30	0.01	4.49	22
64	128	240	30	0.01	4.636	17
128	256	400	30	0.01	5.583	11
32	96	180	30	0.02	4.636	17
64	128	240	30	0.02	7.888	30
128	256	400	30	0.02	11.189	30

Tabuľka 14: Výsledky siete na datasete 4 (množina konfigurácií 2)

Dataset 5

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
32	96	180	30	0.01	6.457	30
64	128	240	30	0.01	5.699	29
128	256	400	30	0.01	5.596	29
32	96	180	30	0.02	8.833	29
64	128	240	30	0.02	8.936	27
128	256	400	30	0.02	8.209	3

Tabuľka 15: Výsledky siete na datasete 5 (množina konfigurácií 2)

4.2 Experiment s datasetom MNIST

Toto testovanie siete som sa rozhodol robiť z dvoch dôvodov. Prvým je porovnanie samotnej siete s ostatnými technikami na rozpoznávanie vzorov. MNIST je na tento účel vhodný dataset, keďže je rozšírený za týmto účelom a existuje množstvo výsledkov, s ktorými je možné porovnávať.

Druhý dôvod je porovnanie obtiažnosti datasetu s nejakým, ktorého obtiažnosť je dobre zmapovaná. MNIST bol vybraný práve z vyššie uvedených dôvodov. Tu sú výsledky:

Konv. roviny 1	Konv. roviny 2	Neuróny v MLP	Epochy	Rýchlosť učenia	Najlepšia chyba v %	Epocha naj. chyby
6	12	60	10	0.01	0.7	7
6	12	120	10	0.01	0.77	6
6	18	60	10	0.01	0.76	6
6	18	120	10	0.01	0.7	10
12	18	60	10	0.01	0.7	9
12	18	120	10	0.01	0.76	6
12	24	60	10	0.01	0.76	5
12	24	120	10	0.01	0.71	4
18	40	60	10	0.01	0.78	5
18	40	120	10	0.01	0.71	5
6	12	60	10	0.02	0.71	8
6	12	120	10	0.02	0.72	8
6	18	60	10	0.02	0.74	10
6	18	120	10	0.02	0.8	10
12	18	60	10	0.02	0.81	5
12	18	120	10	0.02	0.86	7
12	24	60	10	0.02	0.8	7
12	24	120	10	0.02	0.68	10
18	40	60	10	0.02	0.71	10
18	40	120	10	0.02	0.83	8

Tabuľka 16: Výsledky siete na datasete MNIST

4.3 Diskusia

Pre lepšiu prehľadnosť sú najlepšie výsledky DBN a prvého a druhého behu CNN porovnané v tabuľke 17.

Dataset	CNN – množina konfigurácií 1			CNN – množina konfigurácií 2			DBN
	chyba	konfig.	rýchlosť učenia	chyba	konfig.	rýchlosť učenia	chyba
1	7.522	18-40-120	0.01	6.969	32-96-180	0.01	16.4
2	1.197	12-24-120	0.01	1.263	64-128-240	0.01	4.7
3	8.976	18-40-120	0.01	8.311	32-96-180	0.01	19.6
4	5.291	18-40-120	0.01	4.49	32-96-180	0.01	15
5	5.906	18-40-120	0.01	5.596	128-256-400	0.01	19.6

Tabuľka 17: Porovnanie najlepších výsledkov CNN a DBN

Z nameraných výsledkov môžeme vidieť, že sieť dosahuje významne lepšiu testovaciu chybu ako DBN, čím sa stala dobrým kandidátom na kategorizáciu obrazových dát podobných, ako v testovacom datasete.

Počet epoch je pravdepodobne pri druhej množine konfigurácií nie úplne dostatočný, na dobré natrénovanie veľmi komplexnej siete by bolo odhadom potrebných vyše 60, možno až 100 epoch. Pri prvej množine konfigurácií obvykle testovacia chyba už významne neklesala po prekročení epochy 50.

Testovacia chyba zjavne závisí od veľkosti datasetu. Pri MNISTe, na ktorom väčšina dnešných rozpoznávacích techník dosahuje chybu pod 1%, čím sa blížila k hranici možnosti ich rozpoznávania, je nízka chyba umožnená aj veľkosťou datasetu 60000 vzorov. Dataset listov je obtiažnejší, čo je dané nízkym počtom vstupných vzorov, v extrémnom prípade len 2000 a, najmä, pridaním šumu do jednotlivých obrazov. Môžeme však povedať, že týmto je dataset viac priblížený reálnym podmienkam, keďže často nie je možné vo vhodnej dobe nazbierať požadovaný počet dát, navyše v takpovediac vysokej kvalite.

Pri počte 8000 trénovacích vzorov bez šumu bola sieť schopná rozpoznať testovacie dáta s chybou zhruba 1.2%, čo môžeme považovať za veľmi dobrý výsledok. Je predpoklad, že s

d'alším vhodným zvýšením počtu vstupov by táto chyba mohla dosiahnuť hodnotu pod 1%. V zašumených datasetoch má sieť prijateľné výsledky. Znova môžeme vidieť, že zvýšením počtu tréovacích vzorov je možné chybu významne znížiť (viď datasety 3 a 4).

Najzaujímavejší je z tohto pohľadu dataset 5, v ktorom je tréovacia množina vytvorená iným spôsobom, ako testovacia, čo ešte sťažuje úlohu klasifikácie. Na druhú stranu, počet vzorov je najväčší z všetkých datasetov listov, čo ho obtiažnosťou stavia niekde medzi dataset 1 a dataset 3.

Čo sa týka rýchlosti učenia, z grafov môžeme vidieť, že pri menších datasetoch (1 a 2) nie je možné jednoznačne určiť, ktorá z dvoch možností dosahuje lepšie výsledky. Určitou stopou môže byť fakt, že všetky najlepšie testovacie chyby boli dosiahnuté s rýchlosťou učenia 0.01. Pri ostatných datasetoch je rýchlosť učenia 0.01 jednoznačne prospešnejšia výslednej testovacej chybe.

Posledným aspektom sú topológie sietí. Okrem druhého datasetu, kde je chyba už značne nízka, pomohlo zvýšenie konvolučných rovín a pridanie neurónov do výstupnej vrstvy pri všetkých ostatných datasetoch.

4.4 Možné vylepšenia

Ako vylepšenie rozpoznávania datasetu listov uvediem prácu [35], v ktorej boli použité Gabor filtre ako detektory príznakov v prvej vrstve. Táto práca používa aj vyššie spomínaný boosting, ktorý je takisto jedným z možných vylepšení. Komisia týchto sietí dosiahla na MNISTe výsledok 0.58%, je teda možné očakávať aj mierne zlepšenie testovacej chyby na datasete listov.

5. Zhrnutie

Cieľom tejto práce bolo ukázať schopnosti konvolučnej neurónovej siete v spojení s datasetom prírodných objektov, konkrétne listov a porovnať tento experiment v dvoch aspektoch. Prvým z nich je výkon samotnej techniky konvolučnej siete, ktorá bola porovnávaná s výkonom Deep Belief Network, pričom konvolučná sieť dosahuje značne lepšie výsledky. Druhým je približné stanovenie obtiažnosti datasetu v porovnaní s niektorým zo známych datasetov. Pre porovnanie som zvolil MNIST, ktorý je tvorený číslicami písanými rukou, pričom ako obtiažnejší sa ukázal byť dataset listov, hlavne kvôli menšiemu počtu tréovacích vzorov a šumu pridaného do obrazu.

Z uvedeného môžeme uzavrieť, že konvolučná sieť má vcelku dobrú odolnosť voči šumu v obraze, čo môže byť spôsobené jej inšpiráciou vizuálnou kôrou. Môže tým pádom byť nasadená aj mimo akademického prostredia - ako príklad môžeme uviesť rozpoznávanie captcha, kde jednotlivé obrázky, obsahujúce číslice a písmená, sú značne skreslené (rotácia, šum, zakrivenie).

6. Príloha

Prílohou k tejto práci je CD, na ktorom sa nachádzajú:

- adresár `datasety_listov`:
obsahuje datasety listov v textovej podobe spolu s ich obrazovými podobami
- adresár `siet`:
obsahuje implementáciu siete použítú v experimentoch
- adresár `Torch7`:
obsahuje zdrojové kódy knižníc Torch7, ktoré sú použité v implementácii siete
- elektronická verzia tejto práce

7. Zoznam použitých zdrojov

- [1] Kvasnička V., Beňušková L., Pospíchal J., Farkaš I., Tiňo P., Král' A. (1997). Úvod do teórie neurónových sietí. IRIS, Bratislava.
- [2] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4), 115-133. Springer.
- [3] Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. (Erlbaum, Eds.) *Science Education* (Vol. 44, p. 335). Wiley.
- [4] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. London.
- [5] Behnke, S. (2003). *Hierarchical Neural Networks for Image Interpretation*. *Lecture Notes in Computer Science* (Vol. 2766, p. 7+217). Springer-Verlag New York Inc.
- [6] Hrycej, T. (1992). *Modular learning in neural networks: a modularized approach to neural network classification*. Wiley.
- [7] Sewell, M. (2008). *Ensemble Learning*. (C Sammut, Eds.) *IEEEACM transactions on computational biology and bioinformatics* IEEE ACM, 8(2), 605-624. Springer New York.
- [8] Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12), 3207-3220. MIT Press, Cambridge, USA.
- [9] Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3), 574-591. Physiological Soc.
- [10] Boureau Y., Bach1 F., LeCun2 Y., & Ponce J. (2010). *Analysis of Feature Learning and Feature Pooling for Image Recognition*. *Learning Workshop*, Cliff Lodge, Utah.
- [11] Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), 500-544. Blackwell Publishing.

- [12] Izhikevich, E. M. (2003). Simple model of spiking neurons. IEEE Transactions on Neural Networks. IEEE.
- [13] Minsky, M., & Papert, S. (1969). Perceptrons: An Introduction to Computational Geometry. (M. L. Minsky & Seymour Papert, Eds.) Handbook of attachment Theory research and clinical (Vol. 1988, p. 258). MIT Press.
- [14] Orr, M. J. L. (1996). Introduction to Radial Basis Function Networks. Time, 302(5652), 1-67. Citeseer.
- [15] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities Biophysics : Hopfield I T ., V . Biophysics, 79(April), 2554-2558. National Acad Sciences.
- [16] Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks. submitted for publication, 148(GMD Report 148), 43. GMD-Forschungszentrum Informationstechnik.
- [17] Martinetz, T., & Schulten, K. (1991). A neural gas network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, & J. Kangas (Eds.), Artificial Neural Networks (Vol. I, pp. 397-402). Elsevier.
- [18] Nguyen, D., & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. International Joint Conference on Neural Networks. IEEE.
- [19] Johansson, R. (2011). Sorting batteries with deep neural networks : Using deep belief networks and convolutionary neural networks for image classification. Master thesis, Chalmers University, Göteborg
- [20] LeCun, Y., Cortes C., online source. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>
- [21] Cireşan, D., Meier, U., & Schmidhuber J. (2012). Multi-column Deep Neural Networks for Image Classification. Applied Sciences, (February), 20.
- [22] Jolliffe, I. T. (2002). Principal Component Analysis. (K. Diamantaras & S. Kung, Eds.) Springer Series in Statistics (Vol. 2, p. 487). Springer

- [23] Cortes, C., & Vapnik, V. (1995). Support-vector networks. (L. Saitta, Ed.)Machine Learning, 20(3), 273-297. Springer.
- [24] Halabuk, M. (2012). Testing the hierarchical neural network DBN in invariant object recognition. Master thesis, Comenius University, Bratislava
- [25] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. (S. Haykin & B. Kosko, Eds.)Proceedings of the IEEE, 86(11), 2278-2324. IEEE.
- [26] Russell, S. J., & Norvig, P. (2010). Artificial Intelligence, A Modern Approach - 3rd Edition.pdf. (S. Russell & P. Norvig, Eds.)Artificial Intelligence. Prentice Hall.
- [27] Hubel, D. H. (1963). The visual cortex of the brain. Scientific American, 209(5), 54-62.
- [28] Weisstein, E. W., online resource. Convolution. MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Convolution.html>
- [29] Online resource. Generic filters – Convolution matrix. GIMP Documentation. <http://docs.gimp.org/en/plugin-convmatrix.html>
- [30] Eddins, S. (2006), online resource. Separable Convolution. MathWorks blog. <http://blogs.mathworks.com/steve/2006/10/04/separable-convolution/>
- [31] Online resource. Torch7 Manual. Torch7 Documentation. <http://www.torch.ch/manual/index>
- [32] Lecun, Y., Farabet, C., & Kavukcuoglu, K. (2010). Convolutional Networks and Applications in Vision. Computer, 253-256. IEEE.
- [33] Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. (P. Vitányi, Ed.)Journal of Computer and System Sciences, 55(1), 119-139. Springer.
- [34] Botton, L., online resources. Stochastic Gradient Descent (version 2). <http://leon.bottou.org/projects/sgd>
- [35] Calderón, A., Roa, S., & Victorino, J. (2003). Handwritten Digit Recognition using Convolutional Neural Networks and Gabor filters. Proc Int Congr Comput Intell. Citeseer.