

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Lenka Polášková



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

**POUŽITÍ METOD HLUBOKÉHO UČENÍ V ÚLOHÁCH
ZPRACOVÁNÍ OBRAZU**

METHODS OF DEEP LEARNING IN IMAGE PROCESSING TASKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Lenka Polášková

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jan Mikulka, Ph.D.

BRNO 2016



Diplomová práce

magisterský navazující studijní obor **Biomedicínské inženýrství a bioinformatika**

Ústav biomedicínského inženýrství

Studentka: Bc. Lenka Polášková

ID: 147517

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Použití metod hlubokého učení v úlohách zpracování obrazu

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši v oblasti metod strojového učení a jejich nasazení při řešení úloh zpracování obrazů. Zaměřte se na metody hlubokého učení (deep learning). 2) Navrhněte postup trojrozměrného zpracování MR tomografických obrazů s cílem segmentace vybrané tkáně. 3) Navrhněte způsob vyhodnocení kvality segmentace. 4) Z dostupných klinických vyšetření vytvořte obrazovou databázi vhodnou ke zpracování. 5) Navržené řešení realizujte v prostředí Matlab, C, Python nebo jiném dostupném jazyce. 6) Proveďte analýzu obrazových dat pomocí navrženého řešení i pomocí referenčních metod, případně manuálním zpracováním. 7) Proveďte vyhodnocení kvality zpracování obrazové databáze navrženými metodami a porovnejte výsledky s odpovídajícími výstupy publikovaných prací. 8) Proveďte diskuzi získaných výsledků a zhodnoťte účinnost a využitelnost metod hlubokého učení v praxi.

DOPORUČENÁ LITERATURA:

[1] DUDA, R.O., HART, P.E., STORK, D.G. Pattern Classification. Wiley, 2001.

[2] ŠONKA, M., HLAVÁČ, V., BOYLE, R. Image Processing, Analysis, and Machine Vision. Thomson, 2008.

Termín zadání: 8.2.2016

Termín odevzdání: 20.5.2016

Vedoucí práce: doc. Ing. Jan Mikulka, Ph.D.

Konzultant diplomové práce:

prof. Ing. Ivo Provazník, Ph.D., předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Učení rozpoznávání objektů pomocí neuronové sítě spočívá v napodobení chování živočišné neuronové sítě. Přestože nejsou ani zdaleka známy detaily fungování mozku, týmy složené z vědců z různých oborů medicíny a techniky se snaží po těchto detailech pátrat. Díky velikánům jako je Geoffrey Hinton věda v tomto oboru velice pokročila. Konvoluční neuronové sítě, které se vycházejí z živočišného modelu optického systému, lze s výhodou použít pro segmentování obrazu, a proto byly vybrány pro segmentaci nádorů a edémů z obrazů magnetické resonance. Modely neuronové sítě, použité v této práci, dosáhly úspěšnosti 41% procent v segmentaci edémů a 79% v segmentaci nádorů z mozkové tkáně.

KLÍČOVÁ SLOVA

strojové učení, neuronová síť, hluboké učení, konvoluční sítě, segmentace obrazů, umělá inteligence

ABSTRACT

The clue of learning to recognize objects using neural network lies in imitation of animal neural network's behavior. In spite the details of how brain works is not known yet, the teams consisting of scientists from various medical or technical professions are trying to search for them. Thanks to giants like Geoffrey Hinton science made a big progress in this domain. The convolutional networks which are based on animal model of optical system can be advantageously used for image segmentation and therefore they were chosen for segmentation of tumor and edema from images of magnetic resonance. The models of artificial neural networks used in this work had achieved the 41% of success in edema segmentation and 79% in segmentation of tumor from brain issue.

KEYWORDS

machine learning, neural network, deep learning, convolutional networks, image segmentation, artificial intelligence

POLÁŠKOVÁ, Lenka *Použití metod hlubokého učení v úlohách zpracování obrazu*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2016. 77 s. Vedoucí práce byl doc. Ing. Jan Mikulka, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Použití metod hlubokého učení v úlohách zpracování obrazu“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Děkuji všem, kteří mě během diplomové práce podporovali, zejména pak vedoucímu práce panu Janu Mikulkovi za přínosné konzultace, korekci textu a přístup na školní server a poskytnutí databáze obrazů, bez čehož bych práci nemohla dokončit a nemalé poděkování patří paní Marině Ronhzině za praktický úvod do neuronových sítí v předmětu AUIN a také přínosné konzultace. Dále bych ráda věnovala poděkování „otcům“ hlubokého učení za sdílení všech znalostí, vědomostí a demo projektů široké veřejnosti.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	12
1 Základní pojmy	13
2 Úvod do algoritmu neuronových sítí	17
2.1 Trénování sítě	17
2.2 Chyba učení	17
2.3 Algoritmus zpětné propagace	18
2.4 <i>Wake-sleep</i> algoritmus	18
2.5 Přeučení sítě	18
3 Učení s učitelem	20
3.1 Vícevrstvý perceptron	20
3.2 Konvoluční sítě	21
3.2.1 Historie konvolučních sítí	22
3.2.2 Sada konvolučních sítí pro výřezy obrazu	25
3.2.3 Plně-konvoluční/Dekonvoluční sítě	26
3.2.4 Problémy CNN	28
3.2.5 Použití CNN	28
4 Učení bez učitele	29
4.1 RBM - <i>Restricted Boltzmann Machines</i>	29
4.1.1 Úprava parametrů sítě	33
4.1.2 Počáteční parametry učení	33
4.1.3 Vzorkování	33
4.1.4 Kontrastivní divergence (<i>KL divergence</i>)	34
4.1.5 Persistentní kontrastivní divergence	34
4.1.6 Kontrastivní divergence založená na minimalizaci volné energie	34
4.1.7 Gibbsovo vzorkování	35
4.2 <i>Deep Belief Networks</i>	35
4.3 Autoenkodery (AE - <i>AutoEncoders</i>)	36
4.4 Stakované autoenkodery (SAE - <i>Stacked AutoEncoders</i>)	39
5 Výběr a přizpůsobení vhodného vývojového prostředí	40
5.1 Optimalizace výpočetních nároků	40
5.2 Toolbox pro MATLAB	40
5.3 Knihovna Theano	40
5.4 Knihovna Torch	41

6	Vyhodnocení kvality segmentace	43
6.1	Dice-Sørensen koeficient	43
6.2	Jaccard koeficient	43
6.3	Senzitivita a specificita	43
7	Návrh trojrozměrné segmentace vybrané mozkové struktury	45
7.1	Sada 2D řezů	45
7.2	Triplanární řezy	45
7.3	Sada 3D řezů	45
7.4	Sada 2D řezů ve třech kontextech	46
8	Charakter dat použitých pro trénování a testování	47
8.1	Databáze BRATS 2015	47
9	Volba modelu CNN, zpracování sady dat, trénování a segmentace	50
9.1	Příprava sady dat pro trénování	50
9.2	Zpracování sady dat pro trénování	50
9.3	Zpracování sady dat pro validaci a segmentaci	51
9.4	Příprava dat po spuštění programu	51
9.5	Model neuronové sítě	52
9.6	Proces trénování a validace	52
9.7	Princip segmentace	54
9.8	Výběr referenční metody pro srovnání výsledků segmentace	54
9.9	Srovnání výsledků segmentace	54
10	Výsledky a diskuze	56
10.1	Segmentace edémů	56
10.2	Segmentace nádorů	58
10.3	Spojení výsledků segmentace nádorů a edémů	59
	Závěr	63
	Literatura	65
	Seznam příloh	68
A	Pozitivní a negativní fáze učení u <i>Restricted Boltzmann machines</i>	69
A.0.1	Pozitivní fáze	69
A.0.2	Negativní fáze	69

B	Hodnoty Dice-Sørensenova, Jaccarova indexu, sensitivity a specificity pro všechny měřené obrazy	71
C	Obsah přiloženého DVD	74
D	Schéma konvoluční sítě LeNet	77

SEZNAM OBRÁZKŮ

2.1	Algoritmus zpětné propagace	19
3.1	Vícevrstevný perceptron	20
3.2	Odezva na změnu parametrů Gaborovy funkce	23
3.3	Ukázka ručně psaných číslic z databáze MNIST	24
3.4	Z obrazu jsou vyřezána tři okna o stejných rozměrech, každé okno podstupuje vlastní CNN a klasifikuje středový pixel okna znázorněný červeným křížkem. Malý kontext (nahore), střední kontext (upro- střed), velký kontext (dole).	26
3.5	Nadvzorkování a dekonvoluce (převzato z: [4])	28
4.1	Příklad s třesoucím se domem	30
4.2	Mezi viditelnou a skrytou vrstvou jsou váhy a prahy sdíleny pro vi- ditelnou a skrytou vrstvu	31
4.3	Učení <i>Restricted Boltzmann</i> sítě (tzv. vzorkování vrstev viz. kap. 4.1.3)	32
4.4	Trénování DBN pro generování dat a klasifikaci	37
8.1	Struktury nádoru a edému klasifikované do čtyř tříd podle [24], pro srovnání je na obr. 8.2 uveden tentýž anotovaný obrázek klasifikovaný pouze do tří tříd - edém, nádor a pozadí - pro segmentaci pomocí CNN	48
8.2	T1-vážené kontrastní MRI zobrazení - lze vidět světlejší lem nádoru (vlevo nahore), T2-vážené MRI zobrazení mozku - lze vidět světlejší homogenní nepravidelnou strukturu obsahující edém (vpravo nahore), anotovaný obraz originál (vlevo dole), anotovaný obraz vytvoření pro rozhodování neuronové sítě (vpravo dole)	49
9.1	Ukázka trénovací sady pro segmentaci nádorů - velký kontext (vlevo nahore), středí kontext (vpravo nahore), malý kontext (dole). Ob- rázky v trénovací sadě obsahují snímky nádorů i pozadí a jsou ná- hodně popřeházeny.	55
10.1	Srovnání úspěšnosti segmentace, anotované obrazy (nahore), segmen- tace metodou jednoduchého prahování (uprostřed), pomocí CNN (dole)	56
10.2	Metodou CNN byla falešně vysegmentována pravá mozková komora	57
10.3	Srovnání úspěšnosti segmentace, anotované obrazy (nahore), segmen- tace metodou jednoduchého prahování (uprostřed), pomocí CNN (dole)	59
10.4	Použití větší trénovací sady (vpravo) v porovnání s menší trénovací sadou (vlevo).	60
10.5	Anotovaný obrázek (vlevo nahore) Metoda CNN (dole uprostřed), metoda kombinovaného prahování(vpravo nahore)	61
10.6	62

10.7	Výsledky spojení segmentace nádorů a edémů. Anotovaný obrázek (vlevo), metoda CNN (uprostřed), metoda kombinovaného praho- vání(vpravo)	62
D.1	Konvoluční síť LeNet	77

SEZNAM TABULEK

3.1	Jednotlivé vrstvy sítě LeNet (topologie je pro jednoduchost demonstrována pro jeden trénovací obraz	25
9.1	Jednotlivé vrstvy sítě pro segmentaci nádorů	53
9.2	Jednotlivé vrstvy sítě pro segmentaci edémů	53
10.1	Srovnání průměrné hodnoty a směrodatné odchylky koeficientů podobnosti u segmentovaných edémů	57
10.2	Srovnání průměrné hodnoty a směrodatné odchylky koeficientů podobnosti u segmentovaných nádorů u prvního pacienta	58
10.3	Srovnání průměrné hodnoty a směrodatné odchylky koeficientů podobnosti u segmentovaných nádorů u druhého pacienta	60
B.1	Srovnání úspěšnosti segmentace edémů metodou kombinovaného prahování a referenčních obrazů	71
B.2	Srovnání úspěšnosti segmentace edémů metodou CNN a referenčních obrazů	71
B.3	Srovnání úspěšnosti segmentace nádorů metodou kombinovaného prahování a referenčních obrazů - první pacient	72
B.4	Srovnání úspěšnosti segmentace nádorů metodou CNN a referenčních obrazů - první pacient	72
B.5	Srovnání úspěšnosti segmentace nádorů metodou kombinovaného prahování a referenčních obrazů - druhý pacient	72
B.6	Srovnání úspěšnosti segmentace nádorů metodou kombinovaného prahování a referenčních obrazů - druhý pacient - pokračování	73
B.7	Srovnání úspěšnosti segmentace nádorů metodou CNN a referenčních obrazů - druhý pacient	73
B.8	Srovnání úspěšnosti segmentace nádorů metodou CNN a referenčních obrazů - druhý pacient - pokračování	73

ÚVOD

Hluboké učení patří do oboru umělé inteligence a je to velmi mladá vědní disciplína. Vychází z principu práce umělých perceptronů spojených do řady a propojených navzájem v několika vrstvách. Ve své podstatě jsou inspirovány biologickým modelem.

Perceptrony jako takové byly objeveny v šedesátých letech, mají ale velmi omezené možnosti. Kolem roku 1985 byly perceptronové sítě upraveny pro potřeby korekce chyb při učení. V této době byl založen princip zpětné propagace, která jak bude vysvětleno dále, má klíčový význam ve funkci sítě. V tomto roce ale ještě nebyly příliš rozvinuté počítačové techniky pro zpracovávání velkého množství dat, což vědce limitovalo. V 90. letech V. Vapnik a spol. vyvinuli novou techniku učení a nazvali ji *Support Vector Machine* (SVM). Tato technika byla dlouho velmi populární a pracovala lépe než tehdejší neuronové sítě a mnoho vědců proto vývoj neuronových sítí opustilo.[1]

V roce 2004 začal klíčový projekt Neuronové algoritmy a adaptivní percepcie (*Neural Computation and Adaptive Perception*) vedený Geoffrey Hintonem z Univerzity v Torontu ve spolupráci s Yann LeCunem, Yoshuou Bengiem a Kanadským Institutem pro pokročilý výzkum (CIRAF). V programu bylo zahrnuto seskupení vybraných vědců z celého světa přes počítačové vědy, biologii, elektroinženýrství, neurologické vědy a psychologii. Cílem projektu je porozumění biologického fungování mozku a imitace principu fungování ve strojovém učení. Výsledkem byl obrovský rozmach v tomto směru a vznik dalších programů hlubokého učení jako například projekt profesora Standfordské univerzity Andrewa Ng, který pro Google+ vytváří hluboké neuronové sítě pro rozpoznávání hlasu, obrazu a významu klíčových slov. Nedávno se k této skupině přidal také Geoffrey Hinton. Yann LeCun spolupracuje na tomtéž pro Facebook. Také Microsoft a nespočet dalších organizací od zmíněných gigantů až po malé firmy či jednotlivce těží z výzkumu, který začali Hinton, LeCun a Bengio. Mnoho studentů z University v Torontu následuje svého mistra. Všichni tito představitelé se stali nejznámějšími osobnostmi mezi vědci zabývajícími se metodami hlubokého učení .[1].

Ještě ani v dnešní době není jasné, jak vlastně mozek funguje. Vědci nedokážou vysvětlit jak spojení mezi biliony neuronů dává vznik inteligenci. Dokážou vysvětlit, jak se šíří elektrický signál mezi neurony, ale stále zůstává otázkou popsat, jak se systém dokáže učit a generovat nové objekty. Umělá inteligence dokáže napodobit chování alespoň některých aspektů mozku, jako je například vizuální kortex, o němž bude řeč u konvolučních neuronových sítí a napodobení tohoto chování se využívá například při zpracovávání obrazu pro takové účely, jako je detekce, rozpoznávání a segmentace objektů [1].

1 ZÁKLADNÍ POJMY

V této kapitole budou vysvětleny pojmy, které jsou v následné práci často používány. Pojmy jsou frekventovaně používány v oboru neuronových sítí, u některých je vhodnější použít anglický výraz než český překlad.

Parametry sítě

Pod pojmem parametry sítě je vždy myšleno váhy a prahy spojující vrstvy neuronů. Cílem úspěšné neuronové sítě jsou správně nastavené parametry sítě tak, že síť na nový, ještě neviděný příklad reaguje podle očekávání.

Trénovací sada dat

Zpravidla nejrozšířenější sada dat, která se používá pro trénování neuronové sítě. Platí, že trénovací sada by měla být dostatečně velká a reprezentativní a zároveň by měla zahrnovat určitý stupeň variability.

Validační sada dat

Část trénovací sady, která se používá během trénování k validaci modelu. Tato sada není použita ani jako trénovací ani jako testovací.

Testovací sada dat

Po natrénování sítě je použita pro testování modelu, tedy určení přesnosti a kvality modelu.

Minidávky

Jak se rozšiřovaly datasety ukázalo se, že pro výpočetní výkon počítače je vhodné a praktické, aby se při trénování, testování a validaci iterovalo přes minidávky trénovací sady namísto práce s celou trénovací sadou najednou. Tomu se pak musí přizpůsobit také úprava parametrů sítě - výsledek je normalizován pro velikost minidávky. Pro jednoduchost bude v teoretickém úvodu do jednotlivých sítí vždy uvažována práce s celým souborem.

Gradientní sestup

Jednoduchý algoritmus, jehož principem je „malými kroky scházet dolů podél chybového prostoru“ definovaného kritériální funkcí. Kritériální funkce jsou funkce klasifikátoru vyjadřující jeho chybu pomocí parametrů. Budou uvedeny dvě.

Zero-One Loss Funkce vyjadřující počet případů kdy se nejvyšší posteriorní pravděpodobnost vstupní proměnné neshoduje se skutečnou hodnotou třídy dané proměnné. Tato funkce ale není diferenciable.

Negative-Log Likelihood (NLL) Její podstatou je záporně vzatá suma logaritmů predikce dané třídy. Stochastický gradientní spád narozdíl od ordinárního gradientního spádu proběhne rychleji, protože odhaduje gradient pouze z několika případů v trénovacího souboru a ne z celého trénovacího souboru. V jednom kroku se vyhodnocuje jeden trénovací vzorek. Pseudokód pro výpočet NLL by vypadal takto:

```
1 for (x_batch, y_batch in train_batches)
2     % pro davku vstupnich dat a jejich stitku
3     % kriterialni funkce je funkci parametru modelu,
4     % vstupnich dat a jejich stitku
5     loss = f(params, x_batch, y_batch)
6     % gradient kriterialni funkce
7     d_loss_wrt_params = T.grad(loss, params)
8     % uprava parametru ve smeru tohoto gradientu
9     params -= learning_rate * d_loss_wrt_params
10    % pokud je dosazeno potrebné kvality uceni,
11    % uceni se ukonci
12    if <stopping condition is met>:
13        return params
14 end
```

Regularizace

Model je třeba trénovat tak, aby fungoval dle očekávání i na nových datech nejen na již viděných a najít takovou síť, která by dobře vystihovala trénovací množinu a zároveň byla dostatečně obecná, aby správně odhalila i neviděná data. Je třeba předcházet přeučení modelu. To se provádí pomocí regularizátorů, které se přidávají jako aditivní parametr ke kritériální funkci. Tyto parametry vyjadřují relativní důležitost daných parametrů učení. Používá se pro ně pojem penalta vah, nebo anglicky také *weight decay*. Pojem penalta vystihuje funkci regularizátorů - z abstraktního

pohledu regularizátory penalizují vysoké hodnoty parametrů, kvůli kterým by síť modelovala příliš vysoké nelinearity a docházelo by také k přeučování.

Brzké ukončení

Pojem brzké ukončení (*early stopping*) se používá k zamezení přeučení modelu. Při trénování sítě může být podmínkou ukončení trénování sítě dosažení maximálního počtu epoch. Nejen pro hluboké sítě to ale není dostatečná podmínka. Jako další podmínka se používá tzv. brzké zastavení učení. Model je při trénování průběžně testován na validačních datech. Brzké zastavení může vypadat tak, že trénování je skončeno, pokud se na validačních datech neprokáže významný rozdíl v chybové funkci po nejméně dvojnásobku iterací od minulého významného rozdílu nebo pokud je chyba učení menší než maximální přípustná chyba učení. Pokud síť pracuje s minidávkami, pak je vhodné po určitém množství natrénovaných minidávek nechat síť validovat a měřit významnost chybového odhadu porovnáním s předchozím odhadem.

Stride

Pojem *stride* vyjadřuje krok, se kterým jsou váhy násobeny se vstupními daty. *Stride* 2 například znamená násobení vah s každým druhým vzorkem vstupních dat.

Softmax

Funkce, která převádí obecně M-dimenzionální prostor do N-dimenzionálního prostoru, kde $N < M$. Používá se pro klasifikaci ve výstupní vrstvě, kdy vítěznou třídou se stává ta, která má nejvyšší posteriorní pravděpodobnost. Její funkční předpis je následující:[2]

$$P(Y = i|x, W, b) = \text{softmax}(Wx + b) = \frac{e^{(W_{x_i} + b_i)}}{\sum_j e^{(W_{x_j} + b_j)}}, \quad (1.1)$$

kde Y je množina tříd dat, W je matice vah sítě, b je vektor prahů sítě, x označuje trénovací množinu. Model pak predikuje výstupní proměnnou klasifikace jako třídu, jejíž pravděpodobnost je nejvyšší:[2]

$$y_{pred} = \text{argmax}_i P(Y = i|x, W, b). \quad (1.2)$$

Dropout

Dropout je technika, kterou vymyslel zakladatel skupiny pro výzkum hlubokého učení Geoffrey Hinton. Tato technika spočívá v tom, že se během učení deaktivuje polovina neuronů každé vrstvy. Druhá polovina je aktivní a v další epoše je tomu naopak. Je to jedna z možností, jak síti zabránit, aby se síť „přeučila“. [31]

Ground truth

Tento termín se používá pro označení modelového výsledku metody. Pro metody učení s učitelem je často modelový výsledek množina anotovaných dat, vůči kterým se provádí výpočet chyby. Tento pojem se používá při hodnocení kvality modelu, kdy se srovnávají modelové výsledky, poskytnuté například lidským expertem, a výsledky testované metody.

2 ÚVOD DO ALGORITMU NEURONOVÝCH SÍTÍ

Cílem této kapitoly je popsat podstatu neuronových sítí pro metody hlubokého učení, nebudou zde však dopodrobna rozebírány základní pravidla neuronových sítí. Kvalitní zdroj pro tato základní pravidla je možno najít zde [16].

Teprve relativně nedávno se začaly masivně rozšiřovat metody hlubokého učení. Název vyplývá z podstaty, že neuronová síť musí mít implementovanu alespoň jednu skrytou vrstvu. Jednotlivé vrstvy sítě extrahují z dat tvořených trénovací sadou charakteristické rysy těchto dat a to od nejjednodušších, jako jsou hranové reprezentace, po nejsložitější jako je význam položek trénovací sady. Kromě klasifikace se tyto neuronové sítě dají využít i pro segmentaci obrazu, protože segmentace obrazu je zvláštní případ klasifikace, kdy se klasifikuje každý pixel zvlášť.

2.1 Trénování sítě

Trénování sítě je založeno na maticovém násobení vstupů jednotlivých vrstev, odečtení prahového parametru a aplikace nelineární funkce a tyto kroky se opakují po definované množství vrstev. Co se trénování s datasetem obrazů týče, k trénování se často používají minidávky obrazů. Minidávky jsou nejprve náhodně seřazeny. U každé minidávky je provedeno dopředné trénování, dále je změřena chyba učení a tato chyba je propagována zpět skrz celou topologii sítě a během této zpětné cesty se upravují parametry sítě. K vylepšení parametrů modelu bylo vyvinuto několik přístupů. Tím nejznámějším a nejběžnějším je algoritmus zpětné propagace. Poté, co tyto kroky absolvuje každá minidávka obrazů z datasetu, skočila jedna epocha učení. Minidávky jsou opět náhodně seřazeny a následuje nová epocha učení. Po určitém množství epoch je vhodné provést validaci modelu s validačními daty.

2.2 Chyba učení

Chyba učení se vypočítává jako rozdíl mezi současným výstupem sítě a požadovaným výstupem. Síť se „učí“ vygenerovat takovou reprezentaci vstupních dat, aby co nejlépe odpovídala požadovanému výstupu. Pokud budou vstupním datasetem obrázky ručně psaných číslic a výstupem vektor hodnot od nuly po devítku, kde každému obrázku odpovídá jedna hodnota, síť se „naučí“, které charakteristiky na obrázku odpovídají které číslici a když se pak objeví nový obrázek ručně psané číslice, síť je schopna tomuto obrázku dodat význam a zařadit jej do jedné ze skupin hodnot od nuly do devíti. Požadovaný výstup sítě může být dán explicitně, nejčastěji

přiřazením významu od lidského experta, v tom případě se jedná o učení s učitelem. V mnoha praktických aplikacích ale nejsou k dispozici požadované výsledky a pak se jedná o učení bez učitele.

Jedním z jednodušších způsobů jak vypočítat chybu učení je suma kvadrátů odchylek přes všechny sledované učební příklady, podle níže uvedené rovnice 2.1.

$$\delta_i = \sum_i (y_i - d_i)^2, \quad (2.1)$$

kde y_i je i -tý výstup sítě a d_i je i -tý požadovaný výstup sítě. Dalším způsobem stanovení chyby vyžaduje přístup k datům jako k distribuci náhodných proměnných s modelovým rozdělením. Často se ke stanovení chyby tímto způsobem využívá Kullback-Leiblerovy divergence definované pro diskretní proměnnou podle rov. 2.2.

$$D_{KL}(P||Q) = \sum_i P_i \log \frac{P_i}{Q_i}, \quad (2.2)$$

kde P_i je i -tá hodnota rozdělení výstupní proměnné P a Q_i je i -tá hodnota požadované proměnné Q [19]. Chyba učení se normalizuje vzhledem k počtu minidávek a k počtu epoch.

2.3 Algoritmus zpětné propagace

Během zpětné propagace se upravují parametry sítě podle rovnice 2.3

$$w_{k,j}(i+1) = w_{k,j}(i) + \mu \cdot \delta_{k,j} \cdot \frac{\partial f(\alpha)}{\partial \alpha} \cdot y_{k-1,j}, \quad (2.3)$$

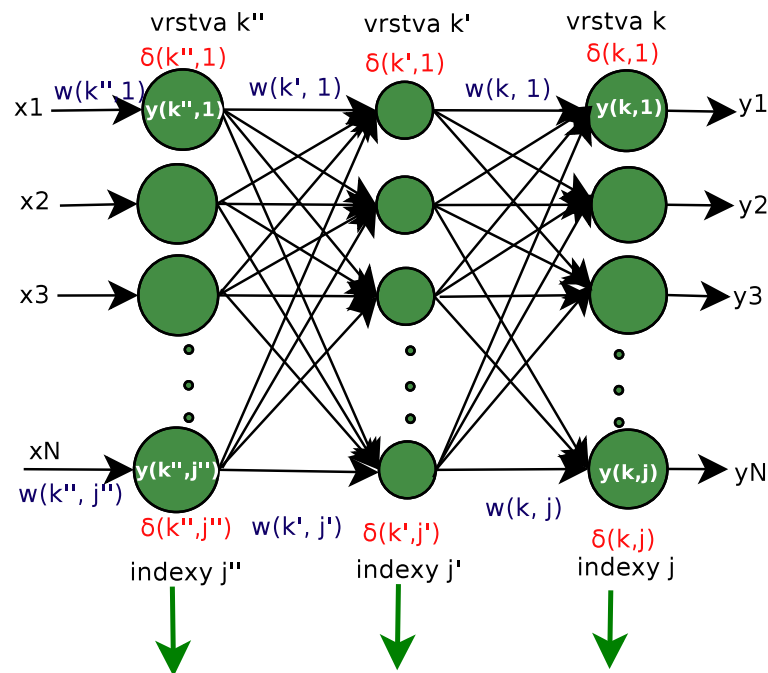
kde w jsou váhy mezi neuronem vrstvy k a neuronem vrstvy $k-1$, μ je parametr učení, j je pořadí neuronu ve vrstvě, $\delta_{k,j}$ symbolizuje chybu spočtenou pro neurony vrstvy k , $\frac{\partial f(\alpha)}{\partial \alpha}$ je parciální derivace aktivační funkce $f(\alpha)$ a $y_{k-1,j}$ je výstup neuronu j ve vrstvě $k-1$. [16] Lépe je tento příklad ukázán na obrázku 2.1

2.4 Wake-sleep algoritmus

Tento algoritmus se používá u trénování pravděpodobnostních modelů a bude rozebrán v kap. 4.2.

2.5 Přeučení sítě

Během učení může chyba neuronové sítě kolísat a při správné implementaci a konfiguraci sítě, vhodné volbě modelu a pečlivě sestaveném datasetu chyba v určitém



Obr. 2.1: Algoritmus zpětné propagace

čase dosáhne minima. Chyba sítě po natrénování je malá, avšak po shlédnutí nových dat je chybovost sítě vysoká. [29] Pro eliminaci přeučování sítě se doporučuje vyzkoušet tyto postupy:

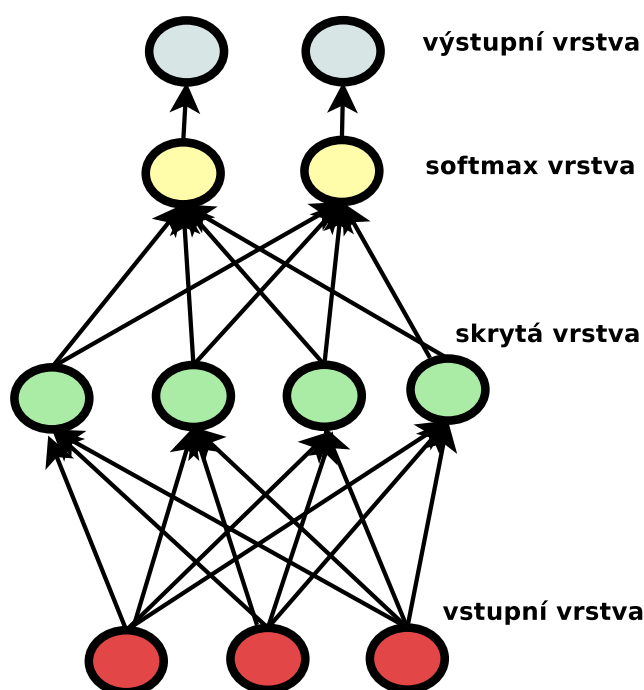
- rozšířit testovací sadu (např. pootočení obrázků o malý úhel) nebo generalizace dat (např. přidání lehkého šumu), [32]
- změnit parametry učení (inicializace prahů a vah, rychlost učení apod.), [32]
- přidat penializaci vah (více v kapitole 1) [29],
- zmenšit počet parametrů učení, [32]
- moninotorování procesu učení a brzké zastavení učení (více v kapitole 1) [29] [32],
- použít techniku Dropout (více v kapitole 1) [31] [32]
- provést tzv. předučení, tj několik epoch učení bez učitele pro inicializaci vah [30].

3 UČENÍ S UČITELEM

Tato kapitola bude věnována učení s učitelem. Do této kategorie spadají vícevrstvé perceptrony a konvoluční sítě.

3.1 Vícevrstvý perceptron

Vícevrstvý perceptron (MLP - *MultiLayer Perceptron*) je nejjednodušší model neuronových sítí pro hluboké učení. Vstup je nejprve transformován nelineární transformační funkcí. V literatuře se uvádí použití sigmoidální aktivační funkce nebo funkce \tanh - funkce symetrické kolem nuly, a to proto, že jejich výstupy (vstupy do další vrstvy) mají střední hodnotu kolem nuly, což je užitečná vlastnost. Empiricky se prokázalo, že trénování s funkcí \tanh rychleji konverguje k minimu chybové funkce. Tato transformace vstupních dat se projektuje do prostoru kde jsou nové faktory z původních proměnných lineárně separabilní. Po grafické stránce může být MLP reprezentována například jako na obr. 3.1 [2]



Obr. 3.1: Vícevrstevný perceptron

Formálně MLP s jednou skrytou vrstvou je funkce $f : D \rightarrow L$ kde D je velikost vstupního vektoru a L je velikost výstupního vektoru $f(x)$, kde transformační funkce vstupu je vyjádřena rov. 3.1:

$$f(x) = \phi(b_2 + W_2(\psi(b_1 + W_1 \cdot x))) \quad (3.1)$$

kde ψ je aktivační funkcí skryté vrstvy a ϕ je aktivační funkcí softmax vrstvy.[2]

3.2 Konvoluční síť

Konvoluční síť (CNN - *Convolutional Neural Network*) jsou jednoduché neuronové sítě, které využívají princip konvoluce místo obecného násobení matic alespoň v jedné ze svých skrytých vrstev.

Pro obrazy se ale používá principu dvourozměrné diskétní konvoluce zadané takto:

$$s[i, j] = K * I[i, j] = \sum_m \sum_n I[m, n] K[i - m, j - n] \quad (3.2)$$

kde I je vstupní obraz a K je konvoluční maska. Pro využití existuje několik výhod. První výhodou je řídkost v jednotlivých vrstvách neuronové sítě a sdílení vah. Když totiž probíhá konvoluce, pouze několik pixelů z okolí vstupu se podílí na výstupu jednoho neuronu skryté vrstvy a to díky tomu, že konvoluční maska (filtr) - váhy je obvykle menší než vstupní obraz, zatímco u klasické neuronové sítě je každý neuron výstupem interakce se všemi pixely vstupu. Tato menší oblast obrazu, ve které probíhá konvoluce se nazývá **receptivní pole**. Každý obraz jedné vrstvy neuronové sítě využívá jednu matici vah, která se po obraze posunuje. Je-li například konvoluční maskou filtr pro detekci hran (často bývá v první konvoluční vrstvě) pak je hrana detekována v každém bodě obrazu. Použití matice vah menší než matice vstupního obrazu je také výhodnější z hlediska paměťových nároků.

Při trénování jedné CNN se postupuje po několika krocích, které se zpravidla opakují. V prvním kroku je provedena **konvoluce obrazů s filtry** pro obdržení preaktivačních výstupů. Preaktivační výstupy konvolučních vrstev se nazývají **mapy vlastností** (*feature maps*) Tato vrstva sítě zde bude nazývána C s indexem označujícím o kolikátou vrstvu v pořadí se jedná. Maska se po obraze může pohybovat i s větším krokem než 1 (přeskakuje některé pixely), čehož se využívá hlavně pro velké obrazy, kde se funkční hodnota nemění prudce. Nevýhodou ale je ztráta podstatné části informace. Konvoluce s krokem 2 (*stride* 2) ztrácí polovinu informace ze vstupních dat. Další informace se ztratí tzv. okrajovým efektem. Konvoluce nemůže být vypočítána pro $m/2 - 1$ okrajových pixelů pro každý okraj výstupního obrazu, kde m je velikost masky. V MATLABu je tato konvoluce denotována přívlastkem „valid“. Pokud je nutné, aby výsledný obraz měl stejný počet pixelů jako vstupní, okraje jsou doplněny nulami, nebo se hodnoty okrajových pixelů opakují, pak je tato konvoluce v MATLABu denotována přívlastkem v „same“.

V dalším kroku se aplikuje **nelineární aktivační funkce**. Nejčastěji používaná je logistická aktivační funkce, kterou lze vyjádřit vztahem:

$$f(x) = \frac{1}{1 + e^{\left(\frac{x}{T}\right)}}, \quad (3.3)$$

kde parametr T určuje strmost funkce. Také se používá funkce *tanh* nebo *ReLU* (Rectified Linear Unit), danou vztahem

$$f(x) = \max(x, 0), \quad (3.4)$$

V dalším kroku se provádí podvzorkování získaných map vlastností (*pooling*). Tato vrstva sítě zde bude nazývána S s indexem označujícím o kolikátou vrstvu v pořadí se jedná. Podvzorkování dává konvolučním sítím jednu podstatnou vlastnost a totiž, že činí detekovaný znak (například na portrétu obličeje jsou to oči, nos, ústa atd.) invariantní vůči lokálnímu posunu. To je velmi užitečné pokud je v centru zájmu určit, zda se daný znak v obraze nachází a ne v jakém místě se nachází. Podvzorkování však není povinné a může se vynechat. Ačkoliv s podvzorkováním se také ztrácí z obrazu informace, využívá se také proto, že nová matice dává původnímu obrazu nový charakter. Pro podvzorkování se používá několik strategií. Výsledný pixel může být například průměrem nebo maximální hodnotou 4 nebo více pixelů.

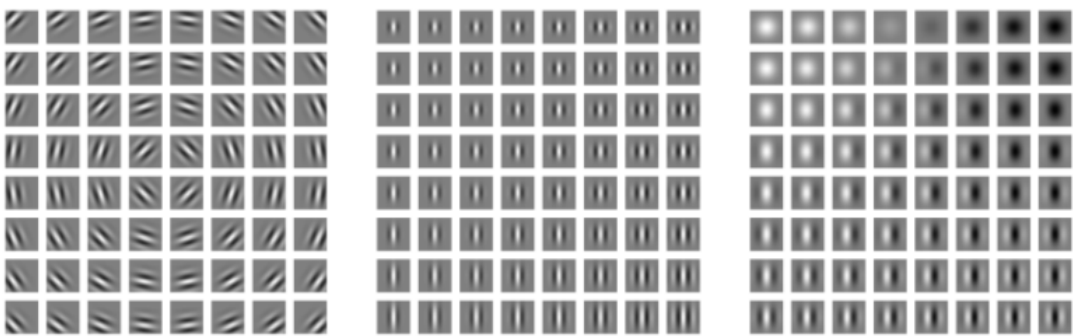
Stejně jako jiné neuronové sítě, výstupy konvoluční nebo podvzorkovací vrstvy slouží v dalším kroku jako vstupy. Procesy konvoluce, aplikace nelineární funkce a podvzorkování se opakují. Podle charakteru aplikace ne vždy je nutno dosáhnout plně-konektované vrstvy. Plně-konektovaná vrstva vznikne rozložením matic poslední vrstvy do vektorů a spojením těchto vektorů. Pro klasifikaci se jako poslední vrstva před výstupní vrstvou vkládá softmax operátor, který vyjádří pravděpodobnost náležitosti obrazu do jedné z klasifikačních tříd. Konvoluční jádra, tvořící váhy sítě, nejsou sdílené mezi konvolučními vrstvami $C_1, C_2 \dots C_n$ v hierarchii sítě. [11]

3.2.1 Historie konvolučních sítí

Počátky konvolučních sítí jsou v pokusech Davida Hubela a Torstena Wieselova na kočkách. Ti podávali kočkám lehká sedativa a znehybňovali jim oči a nahrávali aktivitu individuálních neuronů, aby sledovali, jak neurony mozku kočky odpovídají na projekce obrazů na plátně. První pokusy byly dlouho bez vypovídajícího výsledku. Avšak jednoho dne projektovali na plátno obraz obdélníka a otáčeli s ním. Všimli si, že odezva ve vizuální kortexu kočky koresponduje s touto rotací. Tito dva vědci zjistili, že vizuální kortex se skládá ze seskupení jednoduchých i komplexních buněk.

Obraz se projektuje tím způsobem, že je vytvořen světelným paprskem na retině, kde je převeden na elektrický signál. Odtud putuje optickými nervy do temporálního laloku, kde se projektuje do vrstvy vizuálního kortexu V1. V1 je dvoudimenzionální vrstva neuronů, na které se elektrická aktivita projektuje stejným způsobem, jako světlo na retinu. Čili stimulací spodní poloviny retiny je stimulována odpovídající polovina kortexu. Jednoduché buňky se chovají jako lineární filtry a jsou citlivé na malé subregiony vizuálního pole, zvané receptivní pole. Tyto subregiony jsou spojeny tak, aby pokrývaly celé zorné pole. Buňky subregionů se chovají jako lokální filtry přes celé zorné pole a jsou dobře přizpůsobené, aby využívaly silné prostorové korelace vyskytující se v přírodních obrazech. Jednoduché buňky maximálně reagují na hranové znaky uvnitř receptivního pole. Jednoduché buňky inspirovaly vznik detektorových vrstev. Komplexní buňky mají větší receptivní pole, ale jsou lokálně invariantní vůči pozici objektu v prostoru a také vůči nasvětlení obrazu. Tyto vrstvy se chovají jako podvzorkovací. Vrstvy detektorů následovaných podvzorkováním se opakují do hloubi mozkové tkáně a předpokládá se, že stejná strategie zpracování se aplikuje i v jiných mozkových regionech.

Aby tito vědci změřili váhy neuronů V1 odpovídající vahám v konvoluční vrstvě, změřili nejprve nervovou aktivitu po projekci bílého šumu a následně změřili nervovou aktivitu po projekci modelu s lineárním rozložením jasové funkce podél prostoru. Tento postup se nazývá reverzní korelace. Tímto způsobem se zjistilo, že váhy neuronu odpovídají vahám Gaborových funkcí a mění se ve směru od středu zorného pole k periferiím. Protože běžně se váhy inicializují jako náhodná čísla a je potřeba je „učit“, pokud jsou inicializovány jako koeficienty Gaborovy filtrů, je učení efektivnější.



Obr. 3.2: Odezva na změnu parametrů Gaborovy funkce

Odezvy na změnu vah jsou znázorněny na obrázku obr. 3.2 (převzato z [11]). Objevili dále mnoho aspektů, které přesahují rámec této práce. Za své objevy dostali Nobelovu cenu.[11]

Jelikož je zvířecí visual kortex nejmocnější nástroj vizuálního zpracování jaký známe, je tedy přirozené snažit se ho napadobit. V devadesátých letech byly konvoluční sítě používány firmou AT&T na čtení šeků. Na konci 90. let sloužily pro čtení 90 % všech šeků v USA. Typický příklad využití CNN pak vyvinula firma Microsoft, její software pro rozpoznávání ručně psaných číslic - LeNet je velmi slavná topologie konvoluční neuronové sítě, proto zde bude zmíněna do větší hloubky. Využívá dataset MNIST (obr. 3.3), což je dataset ručně psaných číslic a jejich klasifikačních štítků. Obsahuje 60 000 trénovacích obrazů, 10 000 testovacích obrazů, z této množiny je pak možno ještě oddělit validační data. První vrstva je viditelná vrstva, kde se vstupuje jedna minidávka obrazů ručně psaných číslic, například o velikosti 50 obrázků. První skrytá vrstva je konvoluční, druhá skrytá vrstva je podvzorkovací, třetí je opět konvoluční, čtvrtá opět podvzorkovací, pátá vrstva vyrovnává matici do vektoru a spojí tyto vektory do jednoho, šestá vrstva je softmax a poslední vrstva je výstupní. Schematicky je uvedena v tab. 3.1



Obr. 3.3: Ukázka ručně psaných číslic z databáze MNIST

Schematicky je tato síť znázorněna v příloze (v tomto schématu je znázorněná práce s minidávkou trénovacích obrazů).

CNN kromě klasifikace může provádět také segmentaci obrazu nebo vyhledávání objektů ve scéně. Dva základní přístupy budou podrobněji popsány:

- Sada konvolučních sítí zpracovávajících paralelně výřezy obrazu,
- Plně-konvoluční síť zpracovávající jeden celý obraz.

Oba přístupy budou podrobněji popsány níže.

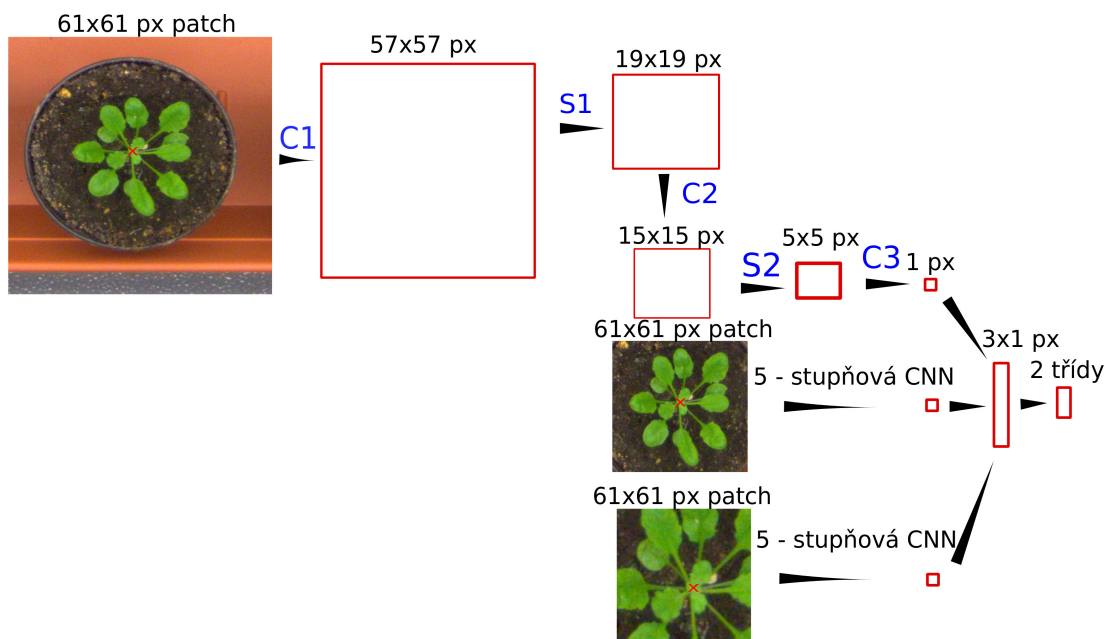
Tab. 3.1: Jednotlivé vrstvy sítě LeNet (topologie je pro jednoduchoost demonstrována pro jeden trénovací obraz)

vrstva	velikost masky nebo podvzorkovací faktor	počet masek	výstup
viditelná	-	-	28x28
C1	5x5	6	24x24x6
S1	2	-	12x12x6
C2	5x5	12	8x8x12
S2	2	-	4x4x12
plně-konektovaná	-	-	192x1
sotmax	-	-	1x1
výstupní	-	-	1x1

3.2.2 Sada konvolučních sítí pro výřezy obrazu

Princip sítě je znázorněn na obr. 3.4 a bude schematicky popsán níže. Z obrazů se postupně po řádcích a po sloupcích obrazové matice vyřezávají okna a podstupují CNN. Na výstupu CNN je třída objektu nacházejícím se ve středovém pixelu vstupního okna. Aby byl algoritmus efektivnější vyřezávají se také jiná okna. Například zde [3] se provádí podvzorkování vstupního obrazu a vyřiznutí okna se stejnými rozměry jako mělo největší okno. Obraz se opět podvzorkuje a vyřizne se další okno. Takto vzniknou tři okna reprezentující objekt v několika přiblíženích. Okna s nejmenším přiblížením by měla reprezentovat interakci objektu s okolím, okna s největším přiblížením zase lokální charakteristiky daného objektu. Každé okno pak podstupuje vlastní CNN. Parametry jednotlivých CNN nejsou sdílené. Poté jsou matice vyrovnány do vektorů, které jsou spojeny do jednoho. Dále následuje opět použití softmax funkce, která predikuje třídu středového pixelu podle jeho maximální pravděpodobnosti. Výsledek lze porovnat s třídou objektu na středovém pixelu anotovaného obrazu a provést zpětnou propagaci. [3]

V tomto odstavci bude stručně popsána podstata použití tří úrovní pohledu na strukturu, jejíž prostřední pixel má být klasifikován. V nejmenším kontextu („small“ kontext) se struktury cíleného objektu zobrazují celé nebo téměř celé a neuronová síť nad tímto kontextem rozhoduje, zda daná struktura na obrázku, odpovídá struktuře, která má být segmentována nebo ne. Tedy na této úrovni provádí neuronová síť pouze rozpoznání celých objektů, avšak není schopna je vysegmentovat s rozlišením několika pixelů. Střední kontext a největší kontext („medium“ a „large“ kontext) tedy hrají roli v rozlišovací schopnosti neuronové sítě. Střední kontext má dvojí roli. Poskytuje bližší pohled na objekt, než malý kontext, tedy na této úrovni lze většinou ještě poznat, o jakou strukturu se jedná, avšak hranici této struktury určuje přesněji, než malý kon-



Obr. 3.4: Z obrazu jsou vyřezána tři okna o stejných rozměrech, každé okno podstupuje vlastní CNN a klasifikuje středový pixel okna znázorněný červeným křížkem. Malý kontext (nahore), střední kontext (uprostřed), velký kontext (dole).

text. Úlohou velkého kontextu je tedy pouze rozlišení hranice struktury s přesností na několik pixelů, ale nemá vůbec žádnou šanci poznat, o jakou strukturu se jedná. Pokud struktura, která má být segmentována, vyplňuje nadpoloviční plochu velkého kontextu, pak se na středovém pixelu tohoto kontextu jistě nachází pixel příslušící do jedné z tříd. Teoreticky vzato, čím více kontextů, tím spolehlivěji může být síť natrénována a čím bližší pohled poskytuje největší kontext, tím větší bude rozlišovací schopnost neuronové sítě. Neplatí to ale zcela absolutně, protože čím vyšší je počet parametrů, které se musí neuronová síť učit, tím pozorněji musí být sledována kvůli artefaktu přeučení. Z důvodů uvedených výše vylývá, že se tento princip se hodí k segmentaci objektů, kdy je zapotřebí klasifikovat jednotlivé pixely s určitým rozlišením.

3.2.3 Plně-konvoluční/Dekonvoluční síť

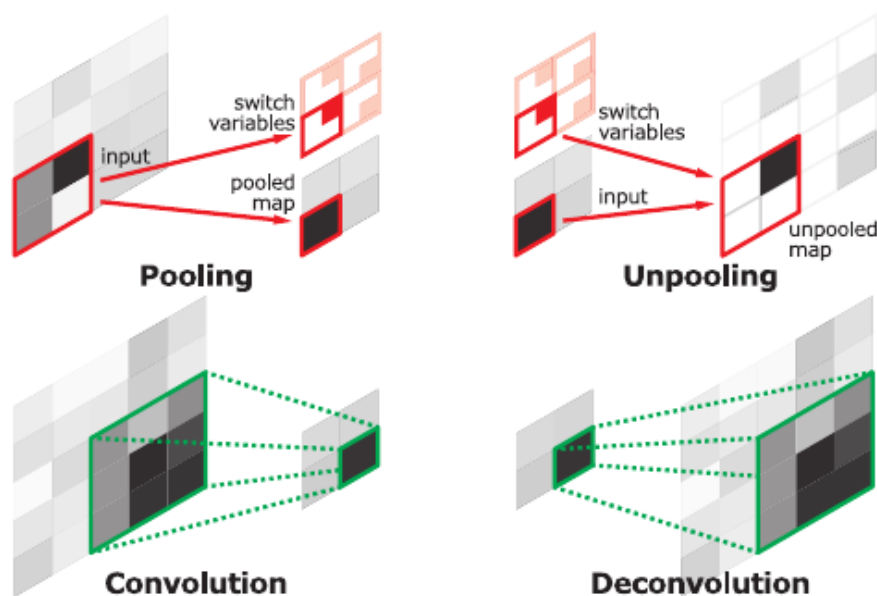
U plně-konvolučních sítí (FCNN - *Fully Convolutional Neural Networks*) postupuje do algoritmu celý obraz. Po několika iteracích konvoluce a podvzorkování je získána plně-konektovaná vrstva. Nyní se postup otočí. Operace proběhnou v obráceném pořadí, místo konvolučních operátorů jsou aplikovány dekonvoluční operátory a z podvzorkovacích vrstev se stanou nadvzorkovací vrstvy. Výsledkem je pravděpodob-

nostní mapa příslušnosti pixelů obrazu do některé třídy. Zde jsou nejzajímavější části operace nadvzorkování a dekonvoluce znázorněné na obr. 3.5. Pro správné provedení nadvzorkování je nutné při podvzorkování trasovat lokalizaci pixelu postupujícího do další vrstvy. K tomu se používají tzv. přepínací proměnné. Při nadvzorkování se použije těchto proměnných ke zjištění původní pozice pixelu a na tuto pozici se také vloží nová hodnota pixelu. Použitím nadvzorkování vznikne řídká matice. Ostatní hodnoty matice se musí doplnit dekonvolucí. Dekonvoluce je proces opačný k procesu konvoluce, kdy aplikací filtrů použitých v dopředném běhu na získané mapy vlastností je získána rekonstrukce původních vstupů konvolučních vrstev. Konvoluce map vlastností s jednotlivými filtry jsou penalizovány parametrem řídkosti (*sparsity term*) a poté se počítají, aby vytvořily mapu vlastností pro vyšší vrstvu. [5]

Výsledný obraz je pravděpodobnostní mapa zařazení pixelu do jedné z predefinovaných tříd a má stejnou velikost jako původní obraz.[4]

Z popsaného principu plně-konvoluční/dekonvoluční sítě je na první pohled zřejmá jedna velká výhoda a naproti tomu dvě velké nevýhody oproti předešlému přístupu. Výhodou zcela určitě je, že obrazy vstupují do algoritmu celé, takže připadá jedna CNN síť na jeden obraz (minidávku obrazů), což urychluje a zefektivňuje výpočet. Nevýhoda je ovšem ta, že nadvzorkování a dekonvoluce způsobují vysoké nepřesnosti ve výstupní mapě. Výstupní obraz trpí příliš vysokým šumem a také nevyhovujícím množstvím tříd - výstupní obraz připomíná něco jako teplotní mapu po zobrazení termovizí. Druhou velkou nevýhodou je limitace týkající se velikosti segmentovaného objektu. Z principu konvoluce, která probíhá v rámci receptivního pole obrazu lze rozpoznávat pouze objekty, které jsou přiměřeně velké vzhledem k receptivnímu poli. Příliš velké objekty mohou být fragmentovány a špatně zařazeny, příliš malé objekty mohou zaniknout v pozadí.[4]

Tento princip se hodí k rozpoznávání objektů ve scéně (*scene parsing*). Scénou může být například ulice. Na ulici stojí domy, stromy, auta a pohybují se lidé, což jsou objekty zájmu. Poté, co je proces konvoluce zastaven v určitém bodě, jsou objekty ve scéně zastoupeny určitým počtem pixelů v mapách vlastností. Pak se provádí zpětný průchod. Objekty, které nejsou předmětem detekce (zpravidla příliš malé a/nebo nevýrazné) jsou během průchodu scény konvoluční sítí vyfiltrovány a ve výsledné mapě zůstávají jen objekty zájmu. Použití pro segmentaci by bylo nevhodné, protože tento typ sítě má limitovanou schopnost dodržet tvar struktury, která má být segmentována.



Obr. 3.5: Nadvzorkování a dekonvoluce (převzato z: [4])

3.2.4 Problémy CNN

Typický problém nastává, jestliže použitím operace konvoluce v důsledku okrajového efektu nebude mít obraz stejnou velikost. Pokud se objekty nacházejí dále od okrajů a je použita malá konvoluční maska, je tento problém zanedbatelný, v opačném případě musí být obraz doplněn nulami nebo opakováním hodnot na okrajích. Dalším velkým problémem se stává situace, kdy nemají všechny obrazy stejné rozměry. Nejjednodušší způsob jak toto ošetřit je rozdělit obrazy na menší díly se stejnou velikostí. Jedním z nejhorších problémů pro trénování je malá trénovací sada. Trénovací sadu je možno rozšířit přidáním stejných obrazů, které jsou rotovány pod malým úhlem.[3]

3.2.5 Použití CNN

Autoři sítí pro segmentaci obrazu nebo parsování scény si algoritmus pro efektivnost upravili podle potřeby. Například k segmentaci pomocí CNN přidali ještě segmentaci pomocí grafové prezentace nebo například shlukovací analýzu [20].

Použití CNN díky typickým výhodám konvoluce a relativní jednoduchosti v použití slibuje dobrý přístup pro segmentaci MRI obrazů. Například zde byla CNN použita pro segmentaci hipokampu.[6]

4 UČENÍ BEZ UČITELE

Tato kapitola bude věnována učení bez učitele. Do této kategorie spadají *Restricted Boltzmann Machines*, *Deep Belief Networks*, autoenkodery a stakované autoenkodery.

Učení je založené pouze na znalostech, které vyplývají z charakteristik trénovacích dat. Hlavním klíčem je tedy najít ve vstupních datech společné charakteristiky. První dva modely jsou tzv. generativní modely. Jsou to modely, které využívají obousměrný postup ve vrstvách a mohou generovat vstupní data. S velkou opatrností by tedy mohly být používány nejen k rozpoznávání ručně psaných textů, ale i k jejich generování, podobně jako to funguje v síti biologické nervové soustavy. [9] [10]

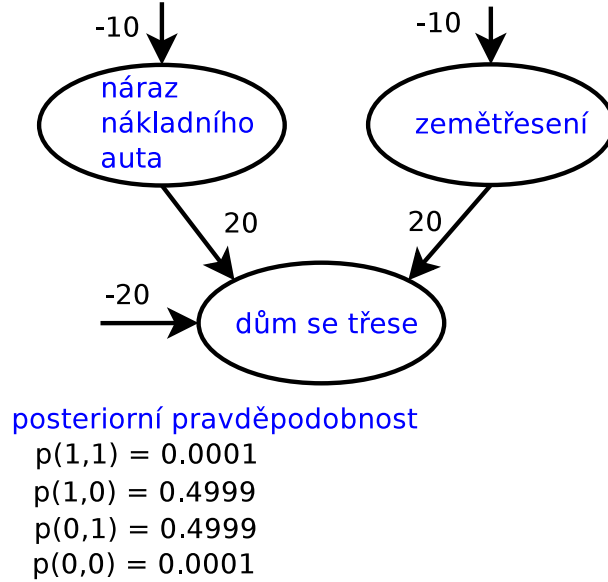
4.1 RBM - *Restricted Boltzmann Machines*

RBM jsou energeticky založené stochastické neuronové sítě. Jejich vstupy a výstupy jsou často binární a parametry nesměrové, model využívá učební algoritmus bez učitele. Do jisté míry jsou podobné Hopfieldově síti, protože model je založen také na optimalizaci chyby dané energetickou funkcí proměnných modelu. Odlišnost však spočívá v tom, že jsou dvou a vícevrstvé, tedy narozdíl od Hopfieldových sítí patří do skupiny hlubokého učení a přívlastek *restricted* znamená, že žádný neuron jedné vrstvy není spojen s jiným neuronem stejné vrstvy. [21]).

Z obecného hlediska patří mezi speciálně uspořádaná Markovova pole, protože cílem jejich použití je odhad posteriorní pravděpodobnosti rozdělení výstupních dat na základě pozorování pravděpodobnostního rozdělení vstupních dat a stavy viditelné a skryté vrstvy závisí pouze na jejich stavech v předchozím kroku. Jiná definice může být i taková, že RBM provádí faktorovou analýzu, tedy snahu vysvětlit rozdělení K vstupních proměnných pomocí N latentních faktorů. Předpokladem správného fungování sítě je, že mezi vstupními proměnnými existuje souvislost, která je patrná právě z rozložení dat. Aktivace neuronů skryté vrstvy závisí na pravděpodobnosti, že budou v jednu chvíli aktivovány určité neurony tj. že vstupní proměnné zaujmou nějakou konfiguraci. Pokud se zvyšuje pravděpodobnost jedné konfigurace vstupních proměnných, zároveň se snižuje pravděpodobnost ostatních konfigurací vstupních proměnných. Také platí, že pokud jsou dvě skryté události nezávislé, mohou se stát závislými, jestliže pozorujeme efekt, který je obojí spojuje. Typicky se tato situace demonstruje na příkladu s třesoucím se domem znázorněná na (obr. 4.1 převzato z [21]).

Pokud pozorujeme třesoucí se dům, příčinou může být náraz nákladního vozu do

domu nebo zemětřesení. Pokud se síť naučí, že třesení domu způsobuje zemětřesení, sníží pravděpodobnost toho, že zemětřesení způsobil náraz nákladního vozu.



Obr. 4.1: Příklad s třesoucím se domem

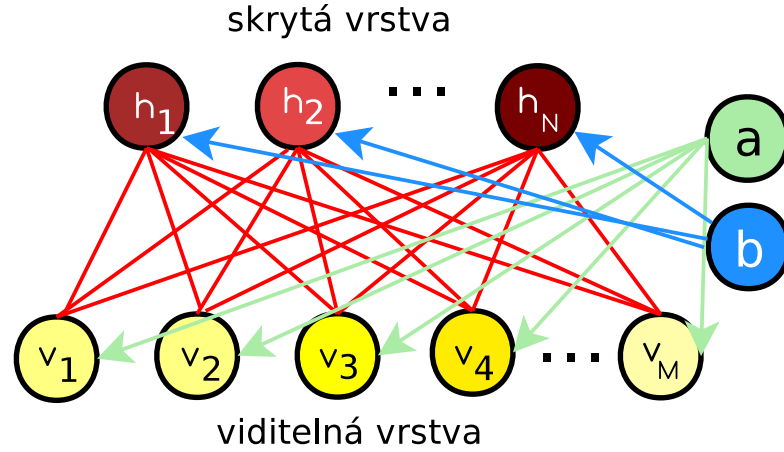
Většinou není známa informace o pravděpodobnosti výskytu jevu, proto trénování začíná s náhodně inicializovanými vahami a učení sítě se tvaruje tzv. energetická funkce modelu. Tu si lze představit jako reliéf krajiny, po které se valí sněhová koule. Tvarování kopců krajiny uvádí kouli do pohybu směrem do údolí. Čím větší je kopec, tím těžší se koule stává, tím má vyšší energii a tím větší je pravděpodobnost, že skončí v nějakém globálním údolí. [7] Energie celé sítě je dána rovnicí:

$$E(v, h) = \sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{i,j}, \quad (4.1)$$

kde E , je energetická funkce modelu, v_i a h_j jsou binární stavy viditelné vrstvy i a skryté vrstvy j , a_i, b_j jsou jejich prahy a $w_{i,j}$ jsou váhy mezi nimi. Tyto váhy jsou sdílené oběma vrstvami (topologie sítě je prezentována na obrázku 4.2 níže).

Učebním algoritmem má být dosaženo snížením této energetické funkce pro určité konfigurace parametrů. K nalezení této energetické funkce slouží nalezení informací o pravděpodobnostním rozdělení proměnných modelu v, h . Pravděpodobnost, že se systém nachází v určité konfiguraci parametrů v, h , se dá vyjádřit jako:

$$p(v, h) = \frac{e^{-E(v,h)}}{Z}, \quad (4.2)$$



Obr. 4.2: Mezi viditelnou a skrytou vrstvou jsou váhy a prahy sdíleny pro viditelnou a skrytou vrstvu

kde Z je tzv. parcitní nebo normalizační funkce, daná vztahem:

$$Z = \sum_{v,h} e^{(-E(v,h))}, \quad (4.3)$$

Jedná se tedy o podíl energie jedné konfigurace vstupních proměnných ku všem ostatním konfiguracím. Nalezení alespoň přibližného řešení parcitní funkce je v praxi NP-těžký problém, protože zahrnuje enumeraci přes všechny konfigurace systému.

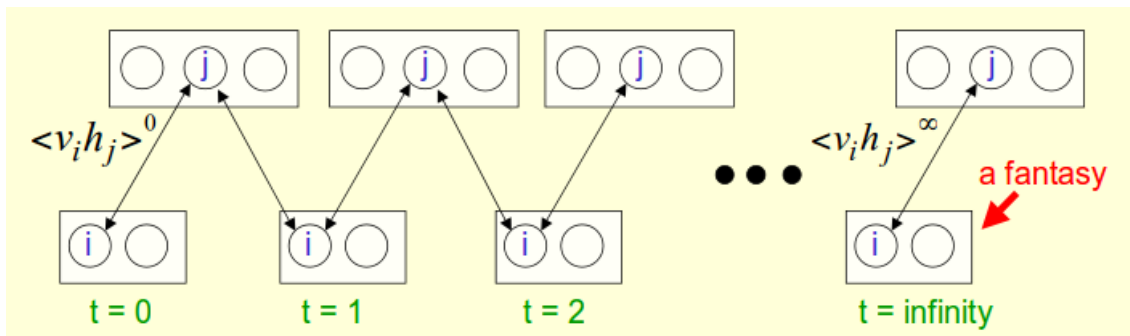
Tedy pro fungování sítě je důležité získat pravděpodobnosti, za jakých se aktivují příslušné neurony skryté vrstvy a naopak pravděpodobnosti, se kterými se aktivují neurony viditelné vrstvy pouze na základě znalostí o aktivaci neuronů skryté vrstvy. Pravděpodobnost, že výstup neuronu j skryté vrstvy na základě znalostí o vstupních datech je dána rovnicí:

$$p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}), \quad (4.4)$$

dále pravděpodobnost, že se aktivuje neuron viditelné vrstvy na pouze základě znalosti o parametrech modelu (bez jakékoliv znalosti o vstupních datech) je dána vztahem:

$$p(v_i = 1|h) = \sigma(a_i + \sum_j h_j w_{ij}), \quad (4.5)$$

Odvození těchto rovnic je demonstrováno například v [8]. Úprava vah a prahů je založena na porovnání výstupu první skryté vrstvy (rekonstrukci dat) a výstupu teoretické nekonečné skryté vrstvy (obr. 4.3, převzato z: [21])



Obr. 4.3: Učení *Restricted Boltzmann* sítí (tzv. vzorkování vrstev viz. kap. 4.1.3)

Váhy jsou pak upravovány podle rovnice:

$$\Delta w_{i,j} = \mu(\langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^{inf}), \quad (4.6)$$

kde μ je rychlost učení, $\langle v_i h_j \rangle^0$ a $\langle v_i h_j \rangle^{inf}$ jsou výstupy první a teoretické nekonečné skryté vrstvy podle obr. 4.3. Tato metoda zkoumá rozdíly mezi dvěma rozděleními a výsledek je tím bližší nule, čím podobnější jsou si daná rozdělení. Implementaci této metody se věnuje kapitola vzorkování. [8] [2]

Postup trénování sítě se skládá ze tří kroků:

- pozitivní (dopředná) fáze učení - vycházející z rovnice 4.4,
- negativní (zpětná) fáze učení - vycházející z rovnice 4.5,
- úprava vah a prahů - vycházející z rovnice 4.6.

Vstupní data jsou v podobě binárního vektoru a při trénování se v praxi využívá toho, že prahy mohou být brány jako speciální váhy s indexem 0. Prahý určují preferenci aktivace neuronů vrstvy v_i resp. h_j . Kladné prahy jsou spojeny s tím, že proměnná vrstvy v_i resp. h_j byla s větší četností nastavena na jedna než na nulu a záporné prahy naopak znamenají, že proměnná vrstvy v_i resp. h_j byla s větší četností nastavena na nulu než na jedna.

Jak už bylo uvedeno, v algoritmizaci lze snadno využít faktu, že práh může být reprezentován jako váha s indexem 0. Ve schématickém znázornění fungování sítě níže bude dodržena tato úvaha. Vstupních dat pak bude $i - 1$ a jednotek skryté vrstvy bude $j - 1$. Dále zde bude využita vlastnost, že žádné dva neurony vstupní ani skryté vrstvy nejsou propojeny navzájem, a proto v pozitivní i negativní fázi mohou být stavy neuronů obnoveny paralelně, tedy během jedné epochy. Zde budou vstupní data anotována jako stavy viditelné vrstvy a data ve skryté vrstvě jako stavy skryté vrstvy. [7]

Jedna epocha učení se skládá z pozitivní fáze, tedy propagace parametrů viditelné vrstvy do neuronů vrstvy, a z negativní fáze, tedy propagace parametrů skryté vrstvy do neuronů viditelné vrstvy. Podrobnější schéma je přidáno do přílohy zde A. Během pozitivní fáze se vypočítají tzv. pozitivní asociace, tj. vztahy mezi parametry viditelné vrstvy navzájem a tzv. pozitivní stavy skryté vrstvy. Během negativní fáze se vypočítají tzv. negativní asociace, které vyjadřují totéž co pozitivní asociace vzhledem k neuronům skryté vrstvy a také negativní stavy viditelné vrstvy.

4.1.1 Úprava parametrů sítě

Úprava vah a prahů probíhá s pomocí vypočtených asociací:

$$\Delta w = \mu(\text{pozitivni_asociace} - \text{negativni_asociace})$$

nebo přesněji:

$$\Delta w = \text{momentum} \cdot \Delta w + \mu(\text{pozitivni_asociace} - \text{negativni_asociace} - \text{weight_decay})$$

$$w = w + \Delta w$$

úprava prahů viditelné vrstvy je odvozena z původních dat a stavů viditelné vrstvy po rekonstrukci:

$$\Delta a = \text{momentum} \cdot \Delta a + \mu \sum (\text{data} - \text{negativni_viditelne_stavy})^T$$

$$a = a + \Delta a$$

úprava prahů skryté vrstvy vychází ze stavů skryté vrstvy v pozitivní fázi a pravděpodobností v negativní fázi:

$$\Delta b = \text{momentum} \cdot \Delta b + \mu \sum (\text{pozitivni_skryte_stavy} - \text{negativni_skryte_pravdepodobnosti})^T$$

$$b = b + \Delta b$$

kde parametr μ je rychlost učení. [12] [15]

4.1.2 Počáteční parametry učení

Váhy se nastavují na náhodné hodnoty blízké nule, prahy se nastavují na nulu, a podle charakteru dat je vhodné přizpůsobit rychlost učení. Rychlost učení se volí jako malá konstanta typicky 0.3. Do algoritmu úpravy vah se může připojit jako parametr momentum a dále penalta pro příliš vysoké váhy, která se typicky volí jako kladná hodnota blízká nule. [9]

4.1.3 Vzorkování

Metody vzorkování rozhodují o tom, jaká data vstupují do pozitivní a negativní fáze a kolikrát se tato fáze provede. Bude zde uvedeno několik metod vzorkování:

- 1) kontrastivní divergence,
- 2) persistentní kontrastivní divergence,
- 3) kontrastivní divergence založená na minimalizaci volné energie,

4) Gibbsovo vzorkování.

4.1.4 Kontrastivní divergence (*KL divergence*)

Tato metoda obecně zkoumá rozdíly mezi dvěma rozděleními. Implementuje se tak, že jako pozitivní stavy viditelné vrstvy v pozitivní fázi učení vstupuje rovnou trénovací vektor. Proto stačí provést pouze jediný krok tohoto vzorkování. [15]

4.1.5 Persistentní kontrastivní divergence

Nejprve se provede pozitivní fáze učení na vstupních datech pro získání pravděpodobností skryté vrstvy, které pak slouží k úpravě vah. Pak je opět provedena pozitivní fáze, ale nyní s náhodnými daty jako pozitivní stavy viditelné vrstvy. Místo toho, aby se prováděla pozitivní a negativní fáze učení na celém souboru najednou se provede na podsouborech tzv. řetězcích. Výstupy těchto podsouborů se pak řadí do matice za sebou, aby ve výsledku tvořily matici správných rozměrů jako by se pracovalo s celým souborem. Rekonstruované negativní viditelné stavy jsou ukládány, takže při dalším běhu vzorkování se jako pozitivních stavů viditelné vrstvy místo náhodně vygenerované matice použije těchto rekonstruovaných dat. Proto má tato metoda velice dobré výsledky. [15]

4.1.6 Kontrastivní divergence založená na minimalizaci volné energie

Vychází z metody persistentní kontrastivní divergence. Rozdíl je v tom, že po každém běhu pozitivní a negativní fáze se data se do výsledné matice seřazují podle velikosti volné energie. Volná energie rekonstruovaných dat je definována následujícím vztahem:

$$F(v) = - \sum_i a_i v_i - \sum_j \log(e^{(x_j+1)}), \quad (4.7)$$

kde

$$x_j = b_j + \sum_j v_i w_{ij}, \quad (4.8)$$

tato metoda se používá u monitorování procesu učení a jako pozitivní stavy viditelné vrstvy vstupují validační data. Pokud začne volná energie validačních dat neproporcionálně růst vzhledem k trénovacím datům, pak se model přeučuje. [9] [12] [15]

4.1.7 Gibbsovo vzorkování

Jako stavy viditelné vrstvy v pozitivní fázi učení vstupuje náhodně vygenerovaná matice dat. Učení pak probíhá jako u kontrastivní divergence. Vzhledem k tomu, že vstupní data u této metody vzorkování vůbec nefigurují (figurují až při úpravě vah) toto vzorkování pro dvouvrtovou RBM síť má význam pouze při snaze vygenerovat naprosto nová data u naučené sítě. [15]

Pro lepší představu o tom, jak tyto sítě fungují, doporučuji například tento zdroj [10]. RBM jsou elementárními jednotkami složitějších sítí jako je *Deep Belief Networks* popsaná v další kapitole.

4.2 *Deep Belief Networks*

Deep belief networks (DBN) jsou několikavrstvé sítě, jejichž poslední dvě vrstvy vždy tvoří RBM. Další vrstvy tvoří Bayesovu síť, kde pravděpodobnosti konfigurace parametrů jedné vrstvy závisí na stavu předešlé vrstvy. Trénování této sítě je velmi náročné. Techniky učení vyžadují značné množství praktických zkušeností s tím, jak nastavit parametry učení a meta-parametry modelu jako je rychlost učení, momentum, inicializační hodnoty vah, počet skrytých vrstev a velikost učebního vzorku pro jeden krok trénování. Také jsou tu rozhodnutí o tom, jaké typy vrstev použít, zda jejich stavy obnovovat stochasticky nebo deterministicky, kolikrát je obnovovat během každé učební fáze apod. Velice záleží na inicializaci modelu a nastavení vzorkování. Dále je užitečné vědět jak monitorovat proces učení a kdy ho zastavit. Tato síť může být použita při klasifikaci dat, při generování nových vzorků a při rekonstrukci poškozených dat. [9]

Trénování sítě probíhá v několika krocích:

- 1) Nejprve je natrénována jednovrstvevná RBM. Jejím výstupem je aproximace vstupních dat.
- 2) Poté se přidá druhá RBM vrstva, jako vstupní vektor slouží výstupy první RBM a jako její váhy jsou použity transponované váhy z prvního trénování. Při trénování této vrstvy se parametry nižší vrstvy nemění.
- 3) Dále je přidána třetí RBM vrstva, která je trénována s parametry a výstupy druhé RBM vrstvy. Opět se parametry nižších vrstev nemění.

Dále je možno přidávat další vrstvy. Další postup je přizpůsoben účelu sítě.

Pro diskriminační analýzu je přidána poslední vrstva softmax, která provádí klasifikaci. Topologie DBN je pak převedena na topologii normální neuronové sítě. Parametry DBN sítě se použijí pro tuto neuronovou síť, takže jednotlivé vrstvy NN přejímají naučené váhy a prahy od vrstev DBN. Poté je možno provádět zpětnou propagaci.

Pro generování nových dat, je potřeba síť natrénovat s daty, jež mají být generovány. Po trénování se na horních dvou vrstvách, které tvoří RBM provádí Gibbsovo vzorkování pro velmi dlouhou dobu a s užitím rovnic:

$$p(h_j^1 = 1|h_2) = \sigma(b^1 + W^{(2)^T} \cdot h^{(2)}), \quad (4.9)$$

$$p(v_i^1 = 1|h_1) = \sigma(b^0 + W^{(1)^T} \cdot h^{(1)}), \quad (4.10)$$

lze ve vrstvě v vygenerovat naprosto nová vstupní data, která budou podobná trénovacímu vektoru. Horní indexy parametrů odpovídají vrstvě, ke které se parametry vztahují. Příklad tohoto využití je uveden zde [13].

Wake-sleep algoritmus

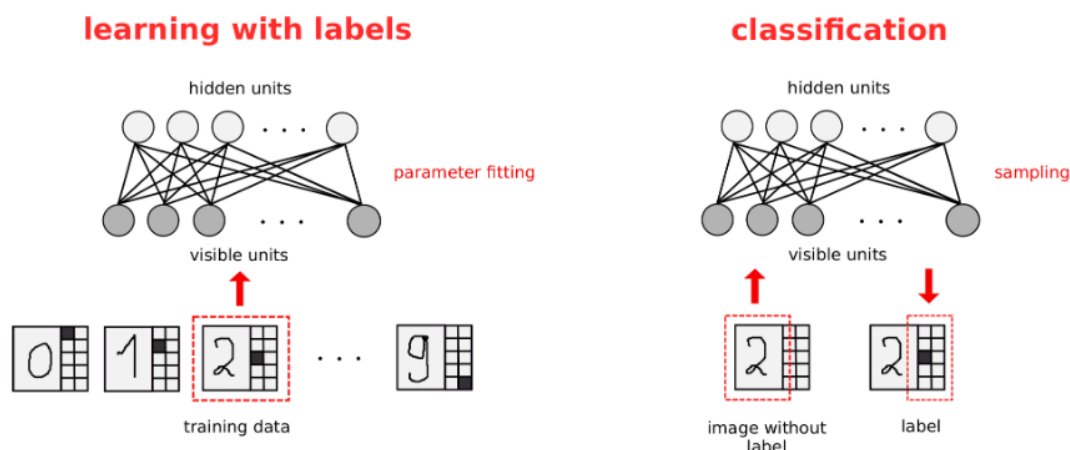
Ke trénování takovéto sítě se využívá tzv. *wake-sleep algoritmus*. U této metody se uvažuje, že mezi jednotlivými vrstvami jsou váhy rozpoznávací a generační. Během *wake* fáze učení sítě směřuje k nejabstraktnějším (nejvrchnějším) vrstvám sítě a síť se učí s použitím rozpoznávacích vah. Přitom adaptuje váhy generačních vah tak, aby se zvýšila pravděpodobnost, že tyto váhy rekonstruují očekávaný výsledek v nižší vrstvě při cestě směrem od nejabstraktnějších k nejnižším vrstvám. Během *sleep* fáze se tyto předpoklady otočí. Postupuje se od nejabstraktnějších vrstev směrem dolů k nejnižším vrstvám s použitím generačních vah a během této cesty jsou upravovány váhy rozpoznávací tak, aby se zvýšila pravděpodobnost, že při cestě nahoru tyto váhy správně zareagují na předložený vstup. [18]

Hinton a kol. vytvořili speciální metodu která v sobě skrývá generování nových dat spjatých s klasifikací. Do viditelné vrstvy mohou vstupovat vstupní data kombinovaná se označením třídy a pro klasifikaci je vložen obraz bez označení třídy a výstupem je vygenerovaný obraz i s označením třídy jak je znázorněno na obr. 4.4 (převzato z: [14])

Dále se DBN využívá pro rekonstrukci poškozených dat například těch zašuměných nebo těch se ztrátou hodnot. Více o implementaci RBM a DBN je možné najít na tomto odkazu [15].

4.3 Autoenkodery (AE - *AutoEncoders*)

Autoenkodery jsou jednoduché neuronové sítě s jednou skrytou vrstvou, které se snaží natrénovat parametry tak, aby výstup sítě co nejvěrohodněji odpovídal vstupu. Skládají se ze dvou částí: endkodér a dekodér. Cílem enkoderu je zakódovat vstupní



Obr. 4.4: Trénování DBN pro generování dat a klasifikaci

data podle kódové abecedy do určité šifry. Cílem dekodéru je rozšifrovat data s pomocí kódové abecedy. Kódovací abecedou se rozumí parametry sítě. Jestliže enkodér zakóduje data s váhami W pak dekodér může k rozšifrování použít vah $W' = W^T$ (vázané váhy - *tied weights*). Je zřejmé, že výstupní vrstva autoenkodéru bude mít stejný počet neuronů jako vstupní vrstva. Chyba rekonstrukce se pak stanovuje pomocí sumy čtverců mezi skutečnou a rekonstruovanou proměnnou nebo pomocí KL divergence. [18]

Existují rozdílné přístupy k práci s autoenkodery. U prvního přístupu se implementuje počet neuronů skryté vrstvy menší než je počet neuronů viditelné vrstvy a k získání neuronů skryté vrstvy se využívají lineární operace. Operace nad zcela nezávislými daty by nevedly k žádnému výsledku, kdežto operace nad daty, která sdílejí určité faktory (nesou společné informace) vede k tomu, že tyto faktory budou rozpoznány a budou se projevovat ve skryté vrstvě. V podstatě tak tento druh autoenkodéru provádí analýzu hlavních komponent (PCA), kde neurony ve skryté vrstvě reprezentují proměnné na prvních osách PCA. Na vektor skryté vrstvy lze také pohlížet jako na kompresi dat. Model bude velmi dobře fungovat na podobných datech (s podobným pravděpodobnostním rozdělením), ale nebude fungovat vůbec na datech s rozdílným rozdělením. [18]

Druhý přístup je takový, že počet neuronů skryté vrstvy je větší než počet neuronů viditelné vrstvy a k získání rekonstruovaného výstupu se používají nelineární operace. S tímto přístupem se pojí problém toho, že model vytvoří funkci identity, která bude pouze kopírovat výstup na vstup, což by v mnoha případech bylo bez užitku. Tomu lze zabránit s použitím koncepce řídkosti, nebo se vytvoří nové vstupní vzorky zašuměním vstupních dat. [18]

Řídké autoenkodery (*Sparsed Autoencoders*)

Koncepce řídkosti v reprezentaci dat je momentálně v oblasti velkého zájmu nejen pro strojové učení, ale také ve statistice a zpracování signálů. Koncept řídkosti byl inspirován analýzou vizuálního kortexu zvířat. Zjistilo se, že jen asi 1 – 4% neuronů je aktivních ve stejný okamžik. Tento koncept vznikl pro účely zpracování dat proto, aby umožnil proměnnou velikost komprese dat *per exempla*. V mnoha odvětvích zájmu vyžadují různá data pro svou kompresi různý počet bitů, které je budou kódovat. Volba fixní délky pro kompresní vektor se pak většinou volí tak, aby poskytovala dostatečný prostor pro kompresi většiny dat. Řídká kompresní matice poskytuje prostor pro kompresi dat s různým počtem stupňů volnosti. Pokud je řídkost vektoru komprese vysoká, lze tento vektor dále komprimovat a tím ušetřit pamětní prostor. Cílem řídkých autoenkoderů je co nejvíce ponechat neurony skryté vrstvy neaktivní.[18] [17]

Řídkosti je dosaženo dvěma způsoby. Prvním způsobem je upravená logistická funkce (*sparsifying logictic*). Tato funkce sleduje průměrnou hodnotu aktivace neuronu i ve vstupních datech a její ofset je nastaven tak, aby ve skryté vrstvě aktivovala neuron j pouze tehdy, zaznamená-li významnou průměrnou aktivaci ve vstupních datech. Dalším způsobem je regularizace parametrů sítě. Je stanoven parametr řídkosti ρ (hodnota blízká nule, typicky 0.05). Po každé epoše se spočte průměrná aktivace neuronu j skryté vrstvy (tedy střední hodnota $\hat{\rho}_j$) a porovná se s hodnotou parametru řídkosti. Úprava parametrů sítě je pak penalizována tímto rozdílem. [18] [17]

Řídkosti se stejně jako u autoendokerů využívá také u RBM a DBN.

Filtrující autoenkodery (*Denoising Autoencoders*)

Princip těchto autoenkoderů je velmi prostý. Uvnitř enkoderu jsou vstupní data uměle zašuměna vhodně vybraným typem a stupněm šumu a cílem dekoderu je rekonstruovat vstupní data s nalezením funkce $\Phi(\bullet)$, kterou je možno charakterizovat statistické rozdělení vstupních dat. „Zašumnění“ lze provést také náhodným nastavením některých neuronů vstupní vrstvy na nulu a pak je cílem rekonstrukce vyplnit tyto prázdné pozice skutečnou hodnotou. Chybu učení u těchto autoenkoderů určuje rozdíl mezi výsledkem rekonstrukce a vstupními daty. [18] Je zcela vhodné tento typ autoenkoderů použít k filtraci signálů a ke generování nových dat, avšak pro účely této diplomové práce se nehodí.

Autoenkodery se obvykle používají k předtrénování složitějších modelů jako jsou RBM a DBN.

4.4 Stakované autoenkodery (SAE - *Stacked AutoEncoders*)

Hierarchie stakovaných autoenkoderů je velmi podobná hierarchii DBN. Stakované autoenkodery se skládají z vrstev autoenkoderů, které jsou stejně jako u DBN postupně přidávány. Také zde se začíná přidáním prvního autoenkoderu a jeho natrénováním. Poté jsou parametry tohoto autoenkoderu fixovány a je přidán další autoenkoder. Jako vstupní vrstva mu slouží výstupy trénování předchozího autoenkoderu. Po každém přidání autoenkoderu jsou parametry předchozích autoenkoderů fixovány a trénovány jsou pouze parametry aktuálně přidaného autoenkoderu. Jako poslední vrstvu je možno zařadit softmax pro klasifikaci. Také u tohoto typu stakované sítě jsou parametry optimalizovány převedením sítě na obyčejnou MLP, natrénováním této MLP s celou trénovací sadou a následnou zpětnou propagací. [2]

5 VÝBĚR A PŘIZPŮSOBENÍ VHODNÉHO VÝVOJOVÉHO PROSTŘEDÍ

Metody hlubokého učení vyžadují použití rozsáhlých databází a vícevrstevných sítí. Tomu je vhodné přizpůsobit vývojové prostředí jak po stránce výpočetních nároků, tak po stránce výběru vhodného softwaru. Vhodný software by měl mít integrované funkce potřebné pro zpracování neuronových sítí tak, aby tyto funkce mohly být jednoduše použity pro vývoj vlastního programu. Tento požadavek splňují volně dostupné knihovny a toolboxy.

5.1 Optimalizace výpočetních nároků

Pro kvalitní výsledky klasifikace a zejména segmentace pomocí neuronových sítí je potřeba použít obrovský dataset a nejspíš několik skrytých vrstev. Navíc je potřeba mezivýsledky během trénování a testování sítě uchovávat v dočasných proměnných, pomocí kterých probíhají výpočty jako je maticové násobení, podvzorkování, interpolace a další. To vyžaduje velký paměťový prostor. Jeden způsob šetření paměti již byl zmíněn v sekci Použité pojmy a to jsou minidávky. Další způsob jak šetřit operační paměť, je distribuce výpočtů na grafickou kartu. Je potřeba, aby grafická karta distribuci podporovala. K těmto účelům se hodí například grafické karty výrobce NVidia, který dokáže distribuovat data, nad kterými probíhají výpočty, mezi operační pamětí a grafickou kartou pomocí knihovny CUDA. Ještě efektivnější se zpracování může stát s použitím knihovny Theano.

5.2 Toolbox pro MATLAB

Matlab má k dispozici vlastní toolbox pro neuronové sítě a volně ke stažení je také DeepLearningToolbox vyvinutý uživatelem Rasmus Berg Palm. V tomto toolboxu jsou vcelku jednoduše i poctivě pokryty všechny metody hlubokého učení, o kterých se tato diplomová práce zmiňuje. Nevýhodou této knihovny je, že její verze je zastaralá a sám autor ji nedoporučuje používat. [12]

5.3 Knihovna Theano

Theano je matematická knihovna, která umožňuje používání symbolických proměnných pro matematické operace. Vícedimenzionální data jsou deklarována datovým typem *Tensor* (v matematice tensor = obecně vícerozměrný vektor). Dále používá tzv. sdílené proměnné, používají se kvůli práci s GPU. GPU je velmi zatížená při

přesouvání dat do paměti „na požádání“, tedy velké objemy dat přesouvané přesně ve chvíli, kdy jsou k výpočtům potřeba. Pokud jsou data definována jako sdílená, k jejich přesunu do GPU dojde pouze jednou a to při prvním volání metody pro přesun těchto dat. Bez použití sdílených proměnných by nebyl kód o moc rychlejší než s použitím CPU nebo dokonce ještě pomalejší. Velká nevýhoda použití této knihovny se zdá být ve velmi nestandardní architektuře kódu a omezených možnostech ladění. Při vykonávání kódu se nejprve nalezne volání první funkce. V deklaraci této funkce jsou stanoveny její argumenty. Výstupní argumenty jsou často dány jako výpočty s použitím vstupních a jiných argumentů této funkce. V těchto výpočtech mohou být použity proměnné deklarované výše v kódu, které je potřeba před použitím inicializovat hodnotou. Tato inicializace se také může skládat z výpočtu pomocí dalších neznámých proměnných deklarovaných v kódu výše atd. Z tohoto důvodu ladící program například nedovoluje vstoupit do funkce a projít ji krok po kroku a dále není možné zobrazit hodnoty symbolických nebo více než 3-dimenzionálních proměnných.

5.4 Knihovna Torch

Knihovna torch nabízí optimální prostředí pro zpracování vícedimenzionálních dat pomocí neuronových sítí. Je napsána v jazyce Lua, což je odlehčený, interpretovaný jazyk syntakticky podobný Pythonu. Lua v sobě kombinuje jednoduchý procedurální syntax s širokým rozmezím procedur nad daty, založených na dynamickém programování. Lua poskytuje automatický management paměti s inkrementálním garbage collectorem. Díky srozumitelnému a nenáročnému syntaxu velmi vhodný na rychlé skriptování. [28]

V knihovně torch se nachází demo pro rozpoznávání objektů, z nichž lze vyjít při psaní vlastního programu, včetně demo pro rozpoznávání ručně psaných číslic. Stejně jako u knihovny Theano i tato knihovna definuje datový typ *Tensor* a umožňuje dynamickou alokaci tenzorů. Vlastní modul se do hlavního spouštěcího modulu přidá příkazem „require“. Tímto příkazem se vykoná veškerý kód v modulu, který není zapouzdřený do funkce. Funkce, definované v modulu, jsou přístupné navrácením ukazatele na funkci příkazem „return“ na konci příslušného modulu. K návratovým hodnotám jednoho modulu se z jiného modulu přistupuje přes instanci přiřazenou při vložení modulu. Po instalaci základního balíčku Torch lze přidávat další moduly (například pro práci s obrazy, normalizaci dat apod.). Jedním z těchto modulů je také výše zmíněná knihovna CUDA.

Knihovnu lze využívat s některými vývojovými prostředími například ZeroBrane Studio, umožňující kontrolu syntaxe, kompletaci kódu a ladění, nebo s LTD plugi-

nem pro Eclipse. Další výhodou knihovny je možnost ukládání modelu neuronové sítě na disk ve formátu `.net`. Z disku jej lze načíst a znovu použít bez nutnosti opětovného trénování. Využití knihovny má také jednu nevýhodu a to tu, že je přístupná pouze pro Mac OS X a Ubuntu 12+.[28]

6 VYHODNOCENÍ KVALITY SEGMENTACE

Nejjednodušším intuitivním způsobem, jak vyhodnotit kvalitu segmentace, je použít koeficienty míry podobnosti obrazů po segmentaci a *ground-truth* obrazů. Prvním koeficientem míry podobnosti je Dice-Sørensen koeficient dalším může být například Jaccardův koeficient. Oba koeficienty se pohybují v rozsahu od nuly pro absolutní minutí množin až do jedné pro absolutní průnik množin. Tedy čím vyšší Dice a Jaccard index, tím větší je podobnost mezi segmentovanými daty a *ground-truth*. Dále se pro vyhodnocení využívá míra senzitivity a specificity.

6.1 Dice-Sørensen koeficient

Tento koeficient byl definován přírodovědcem L.Dicem a nezávisle na něm i botanikem T. Sørensenem. Jestliže definujeme a jako počet pixelů, které jsou nasegmentovány do třídy A v segmentovaném obraze a jako b počet pixelů, které jsou nasegmentovány do třídy A v *ground-truth* obraze, dále h jako pixelů, které jsou nasegmentovány do třídy A v obou obrazech (skutečně pozitivní případy), pak výpočet Dice-Sørensen koeficientu je následující [25] [24]:

$$Dice_index_podobnosti = \frac{2h}{a + b} \quad (6.1)$$

6.2 Jaccard koeficient

Výpočet Jaccardova koeficientu je velice podobný [26]:

$$Jaccard_index_podobnosti = \frac{h}{a + b - h} \quad (6.2)$$

6.3 Senzitivita a specificita

Výpočty senzitivity a specificity se zakládají na sledování pravděpodobnosti, zda je za učitě okolnosti nastal nebo ne. Nech jsou dány pojmy:

TP = počet skutečně pozitivních případů, jako průnik mezi segmentovanou částí a *ground-truth*,

FP = počet falešně pozitivních případů, segmentované části v testovacím obraze se neprotínají se segmentovanými částmi v *ground-truth*,

FN = počet falešně negativních případů, segmentované části v *ground-truth*, které nejsou segmentovány v testovacím obraze

TN = počet skutečně negativních případů, části, které nejsou segmentovány ani v

jednom testu.

Senzitivita testu obecně vyjadřuje pravděpodobnost, že pokud sledovaný jev nastane, test tento případ označí za pozitivní [27]. Tedy podle rov. 6.3:

$$\textit{senzitivita} = \frac{TP}{TP + FN}, \quad (6.3)$$

Specificita testu obecně naopak vyjadřuje pravděpodobnost, že pokud sledovaný jev nenastane, test tento případ označí za negativní [27]. Tedy podle rov. 6.4:

$$\textit{senzitivita} = \frac{TN}{TN + FP}. \quad (6.4)$$

V ideálním případě by se všechny čtyři koeficienty rovnaly jedné.

7 NÁVRH TROJROZMĚRNÉ SEGMENTACE VYBRANÉ MOZKOVÉ STRUKTURY

Trojrozměrná segmentace vybrané struktury ze snímků magnetické rezonance může být provedena několika způsoby:

- Sada 2D řezů,
- Tri-planární řezy,
- Sada 3D řezů,
- Sada 2D řezů se třemi kontexty.

7.1 Sada 2D řezů

Princip této segmentace spočívá v tom, že je použita sada koplanárních 2D řezů kolem každého voxelu, který má být segmentován. Počet a tloušťka vrstev, které jsou pro segmentaci použity pak určuje velikost segmentovaného voxelu. Pro každý voxel, který má být segmentován, je zapotřebí vybrat alespoň 3 vrstvy - jednu v rovině voxelu, jednu paralelně pod a jednu paralelně nad tímto voxelu, které do neuronové sítě vstupují jako jednotlivé kanály. Tento způsob vytvoří v neuronové síti 3D kontext daného voxelu. Výsledkem jednoho průchodu tří vrstev neuronovou sítí je 2D řez, který obsahuje útvar segmentovaný v rovině žádaného voxelu. Útvar segmentovaný ve 3D vznikne uměle vytvořením výsledných 2D segmentovaných vrstev dohromady nebo může být výsledek každé segmentované vrstvy ukládán přímo do výsledné matice za běhu neuronové sítě. [6]

7.2 Triplanární řezy

Druhá použitelná metoda využívá sadu tří perpendikulárních řezů pro každý voxel, který má být segmentován. Uvedená metoda může být například implementována se dvěma nezávislými konvolučními sítěmi, každá segmentuje 3 navzájem kolmé řezy a jejich výstupy se spojí, aby vytvořily lineární plně-konvoluční vrstvu pro finální segmentaci.

7.3 Sada 3D řezů

Tato metoda je intuitivním rozšířením přístupu uvedeného zde 7.1 a spočívá v tom, že neuronová síť neklasifikuje pouze ve 2D, ale ve 3D. Každá vrstva včetně konvolučních filtrů je rozšířena o jeden rozměr. [6]

7.4 Sada 2D řezů ve třech kontextech

Všechny zmíněné přístupy přinášejí komplikaci ve formě implementace neuronové sítě, která musí být přizpůsobena pro práci se 3D vstupy nebo dokonce je sama implementována ve 3D (s trojrozměrnou konvolucí). Předchozí přístupy byly použity ve článku „Deep learning for Medial Image Segmentation“ pro segmentaci hipokampu z mozkové tkáně zde [6]. Výsledky uvedené v tomto článku nepůsobí dobrým dojemem, procenta úspěšnosti jsou nízká oproti relativně vysoké výpočetní náročnosti. Naproti tomu, postup zvolený ve článku „Early hierarchical Contexts Learned by Convolutional Networks for Image Segmentation“ zde [3] je také zajímavý a přesnost této metody na validačních datech uvedená v tomto článku je slibná. Pro tyto důvody byl zvolen přístup, který v sobě kombinuje oba postupy popsané zde 7.1 a zde 3.2.2. Jednotlivé 2D řezy stejné roviny jsou segmentovány pomocí tří konvolučních sítí. Každá konvoluční síť rozpoznává objekt segmentace v různém kontextu. Poslední vrstvy těchto třech neuronových sítí jsou sloučeny do plně-konvoluční vrstvy a na ni navazující *softmax* vrstva pak provádí výslednou segmentaci pixelu. Jak bylo uvedeno v kapitole 3.2.2, použití tří kontextů by mělo vést ke zvýšení úspěšnosti klasifikace. Na segmentované 2D řezy se nakonec použijí ke stavbě 3D objektu.

8 CHARAKTER DAT POUŽITÝCH PRO TRÉNOVÁNÍ A TESTOVÁNÍ

Hlavním cílem této diplomové práce je trojrozměrná segmentace vybrané tkáně ve snímcích magnetické rezonance mozku a vyhodnocení kvality segmentace. Pro tento účel byla vybrána tkáň nádorová. Zobrazování magnetickou rezonancí je pro diagnostiku tohoto onemocnění mozkové tkáně velmi vhodné, protože při použití specifických modalit zobrazení lze nádorovou masu relativně dobře odlišit od okolní tkáně. Dále byla provedena i segmentace okolního edému pro demonstrační ukázkou. V této diplomové práci byla vybrána data z databáze sestavené pro účely BRATS (Multimodal Brain Tumor Image Segmentation Challenge) 2015 ve spojení s konferencí MICCAI 2015.[24] Této databázi bude věnována další sekce.

8.1 Databáze BRATS 2015

Databáze vytvořená pro soutěž BRATS 2015 byla vytvořena pro srovnání mnoha přístupů k segmentaci nádoru a relevantních struktur v mozkové tkáni. V tomto ročníku soutěže bylo srovnáváno 20 algoritmů segmentace, celková úspěšnost se pohybovala mezi 75% a 85% Dice koeficientu. Tato databáze obsahuje 300 případů HGG (*high-grade glioma*, anaplastické astrocytomy a multiformní glioblastomy) a LGG (*low-grade glioma*, astrocytomy nebo oligoastrocytomy) nádorů. Každý případ byl zobrazován čtyřmi modalitami MRI zobrazení:

- T1-vážené zobrazení - nativní snímky, axiální nebo sagitální 2D akvizice,
- T1-vážené kontrastní zobrazení - 3D akvizice, izotropické voxely o velikosti 1mm,
- T2-vážené zobrazení - axiální 2D akvizice, tloušťka řezu 2-6 mm,
- FLAIR zobrazení - axiální, koronární nebo sagitální, tloušťka řezu 2-6 mm.

Rozlišení obrazů se pohybuje v rozmezí 200×200 až 400×400 pixelů, u každého snímaného pacienta je rozlišení napříč řezy stejné. Počet řezů se pohybuje v rozmezí 100 - 150 na jednoho pacienta. Všechny řezy jednoho pacienta jsou registrované (i napříč modalitami, registrace bylo provedena vzhledem k T1-kontrastnímu zobrazení) a normalizované. Každý řez byl manuálně anotován čtyřmi experty, třídy nádor a edém jsou v těchto obrazech odlišeny ve stupních šedi, avšak nachází se zde ještě více tříd.

Podle referenčního manuálu k databázi BRATS se v anotovaném obraze nacházejí celkově 4 třídy znázorněné na obr. 8.1 a jsou to tyto: edém (žlutá), neexpandující solidní část jádra nádoru (červená), nekrotické/cystické jádro nádoru (zelená), expandující jádro nádoru (modrá). Všechny čtyři třídy nemusí být rozlišitelné u všech

modalit. Edém byl anotován primárně z T2-vážených obrazů, proto byly tyto obrazy vybrány pro trénování a segmentaci edému. Modalitou FLAIR jsou zobrazeny struktury nádoru i edému, celá patologická jednotka je dobře rozlišitelná od okolní nervové tkáně, ale v těchto snímcích je hůře rozpoznatelná hranice jednotlivých podstruktur. T1-kontrastní i T2-vážené zobrazení budou popsány níže. [24]



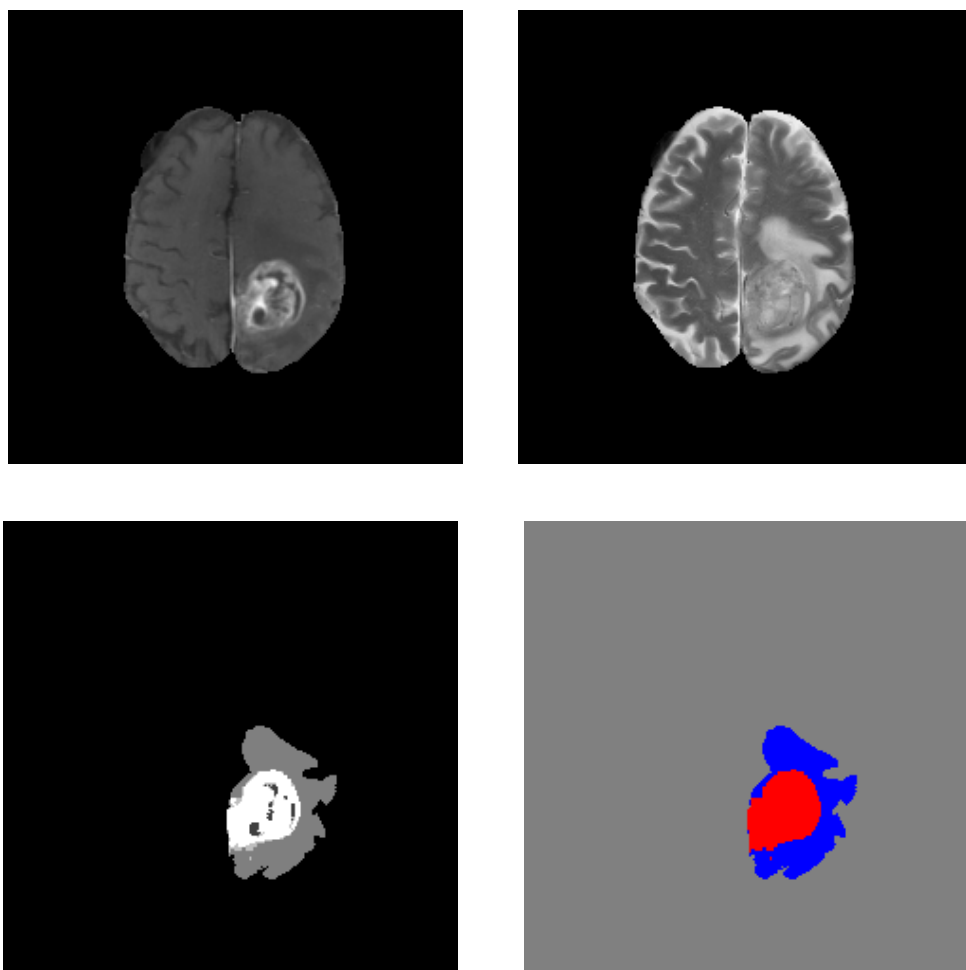
Obr. 8.1: Struktury nádoru a edému klasifikované do čtyř tříd podle [24], pro srovnání je na obr. 8.2 uveden tentýž anotovaný obrázek klasifikovaný pouze do tří tříd - edém, nádor a pozadí - pro segmentaci pomocí CNN

Pro praktickou práci byly některé třídy spojeny dohromady, aby vytvořily výslednou třídu pixelů. Na obr. 8.2 vpravo dole lze vidět anotovaný obraz z databáze BRATS, ve kterém jsou červeně obarveny pixely, které byly klasifikovány jako nádor a modře obarveny pixely, které byly klasifikovány jako edém.

Pro segmentaci nádorů byly vybrány pouze případy segmentované metodou T1-váženého kontrastního a pro segmentaci edému byly vybrány případy T2-váženého zobrazení. Pro ilustraci jsou uvedeny řezy T1-váženého kontrastního zobrazení na obr. 8.2 vlevo nahoře a T2-váženého zobrazení na obr. 8.2 vpravo nahoře, přičemž tyto řezy jsou snímány u jednoho pacienta v téže rovině.

Při T1-váženém kontrastním zobrazování se využívá injekce kontrastní látky, např. Gadolinia pro prodloužení relaxačního času T1 tkání, které tuto kontrastní látku vychytávají.[23] Ty se pak na výsledném obraze jeví kontrastnější. T1-vážené kontrastní zobrazení umožňuje velmi dobré zobrazení tumoru, především jeho okrajových struktur jak lze vidět na obr. 8.2 vlevo. [24]

T2-vážený obraz je obrazem rozdílných relaxačních časů T2, tedy příčné neboli spin-spin relaxace. Tuto relaxaci mají delší struktury obsahující vysoké procento



Obr. 8.2: T1-vážené kontrastní MRI zobrazení - lze vidět světlejší lem nádoru (vlevo nahoře), T2-vážené MRI zobrazení mozku - lze vidět světlejší homogenní nepravidelnou strukturu obsahující edém (vpravo nahoře), anotovaný obraz originál (vlevo dole), anotovaný obraz vytvoření pro rozhodování neuronové sítě (vpravo dole)

vody, tedy edém, ale i například mozkomíšní mok v mozkových komorách. Edém se typicky vyskytuje v okolí nádoru jako reakce na jeho rozpínání. Struktura edému má nepravidelný, individuální tvar a je vidět v okolí nádoru na obr. 8.2 vpravo.[22] [24]

Anotované obrazy jsou v databázi k dispozici u každého pacienta a následná segmentace softwarem byla prováděna pouze v axiální rovině. Jednotlivé sekvence případů jsou uloženy ve formátu .mha, který je možné načíst například programem 3DSlicer. Tento program navíc nabízí možnost sestavení DICOM databáze ze všech řezů jednoho pacienta. DICOM je standardní formát pro ukládání obrazové informace v medicíně a je možno jej načíst například pomocí programu MATLAB.

9 VOLBA MODELU CNN, ZPRACOVÁNÍ SADY DAT, TRÉNOVÁNÍ A SEGMENTACE

Prvotním cílem diplomové práce bylo zjistit, zda je segmentace obrazu pomocí neuronové sítě vůbec možná. V teoretické části byla uvedena řada zdrojů, které odkazují na funkční postupy ověřené v praxi. Princip neuronové sítě, použité v této diplomové práci, je inspirován postupem uvedeným v kapitole 3.2.2 a tato kapitola bude věnována tomuto postupu.

9.1 Příprava sady dat pro trénování

Jak je popsáno v kapitole 8.1, obrazy jsou uloženy ve formátu .mha. Během načítání těchto obrazů do programu 3DSlicer proběhla první selekce nereprezentativních případů výskytu tumoru či edému. Šlo například o případy, kdy nádor nebyl dobře rozpoznatelný. Pomocí programu 3DSlicer byly jednotlivé případy ukládány ve formátu DICOM sérií. Pro trénování neuronové sítě byla využita data pouze axiálních řezů a to celkově od 52 pacientů. V programu MATLAB jsou obrazy DICOM sérií načteny a uloženy ve formátu PNG v datovém typu *uint16*. Po tomto kroku proběhla další selekce. Ze 100 až 150 řezů jednoho pacienta je k trénování použito maximálně deset, protože řezy, které segmentovaný útvar neobsahují, nebo ty, které snímají sousedící roviny, nejsou k trénování potřeba. Je lépe použít méně řezů od vícero pacientů než naopak, protože trénovací sada musí postihnout co možná nejširší spektrum případů, jinak může docházet k přeučování sítě (viz. kap. 2.5). V dalším kroku bylo provedeno *parsování* jednotlivých řezů, tedy vytváření receptivních polí, která postupují do neuronové sítě.

9.2 Zpracování sady dat pro trénování

Axiální řezy, které byly vybrány, protože obsahovaly sledovaný útvar dobře rozlišitelný od okolní tkáně, byly tedy nejprve dvakrát a čtyřikrát nadvzorkovány a na okrajích doplněny nulami. T1, T2 zobrazení i anotované obrazy daného řezu byly v programu procházeny klouzajícím oknem s určitým krokem a na daných pozicích byly vybírány oblasti zájmu. Z nenadvzorkovaných řezů byly vybrány oblasti zájmu o dvojnásobné velikosti receptivního pole, které tvořily nejmenší kontext (v programu označovaný výrazem „small“). Ze dvakrát nadvzorkovaných obrazů byly vybrány oblasti také o dvojnásobné velikosti receptivního pole, aby tvořily střední kontext (v programu označovaný výrazem „medium“). Ze čtyřikrát nadvzorkovaných obrazů byly vybrány oblasti také o dvojnásobné velikosti receptivního pole, aby tvořily velký

kontext (v programu označovaný výrazem „large“), avšak u T2-vážených obrazů byly oblasti o velikosti 180 na 180 pixelů, tedy více, než je dvojnásobek receptivního pole. To je z toho důvodu, že se vyskytují v okolí nádoru a je pro jejich celistvé zachycení tudíž nutno použít větší okno. O tom, do které třídy kontext spadá, rozhodovaly pixely v okolí centrálního pixelu okna anotovaného obrazu. Jestliže se na většině pixelů v okolí středu nacházela třída nádor nebo edém, byl tento vzorek zařazen do dané třídy a uložen na disk. V tomto případě byla zvolena velikost okolí 7x7 pixelů. Naopak, jestliže se na většině pixelů v okolí středu nenacházela třída nádor nebo edém, byl tento vzorek zařazen do třídy pozadí a uložen na disk. V tomto případě byla zvolena velikost okolí 21x21 pixelů. Zvolení různého okolí má důležitou podstatu. Pro trénování neuronové sítě je potřeba, aby vzorky patřící do jiných tříd byly tzv. exkluzivní. Čím více se vzorky spadající do různých tříd od sebe liší, tím větší úspěšnost bude mít neuronová síť při klasifikaci. Toto rozhodnutí ale přináší také omezení. Případy, které spadají do nezařazené kategorie bude muset neuronová síť klasifikovat bez apriorní znalosti.

Vybrané oblasti pak byly podvzorkovány na velikost receptivního pole z důvodu šetření paměti. Software prováděl sestavování sady dat automaticky tak, že načítal jednotlivé řezy a po každém řezu se se zvoleným krokem posouvalo okno, které vyřezávalo malé, střední a velké kontexty. Avšak tímto automatickým postupem se ve výsledné databázi kontextů objevily i nereprezentativní případy, proto byla vytvořená databáze následně ještě manuálně profiltrována.

9.3 Zpracování sady dat pro validaci a segmentaci

Příprava i zpracování sady dat pro validaci a segmentaci probíhala stejně jako příprava dat pro trénování. Rozdíl byl pouze ve výběru receptivních polí pro vytvoření výsledné sady dat. Na rozdíl od trénovací sady nyní by bylo nežádoucí vybírat kontextů pouze některé, které správně reprezentují danou třídu. Pro účely validace byly MRI řezy procházeny s krokem 6 pixelů a pro účely segmentace pak s krokem 1 pixel. Krok 1 pixel zajistí, že bude segmentován každý pixel MRI řezu.

9.4 Příprava dat po spuštění programu

Pro sestavení modelu, trénování a testování byl použit volně dostupný software Torch, o němž byla řeč v kapitole 5.4. Po spuštění inicializačního skriptu je prvním krokem načtení T1-vážené a T2-vážené trénovací a testovací sady. Trénovací sada je načtena z disku, kam byla uložena pomocí skriptu pro program MATLAB. Pro validaci bylo použito 4 axiálních řezů od různých pacientů a segmentace nádorů se

pak prováděla na dvou pacientech, zatímco segmentace edému na 5 pacientech, u každého pacienta byly vybrány 2-4 axiální řezy z různých rovin. Validační a testovací sada je vytvářena za běhu programu pomocí skriptů knihovny Torch. Použití dvou různých prostředí pro vytváření sady dat má několik důvodů. Jedním z důvodů je způsob klasifikace kontextů do tříd. U trénovací sady probíhá klasifikace nejprve automaticky, ale poté je revidována a klasifikované obrazy jsou uloženy na disku. U testovací sady probíhá klasifikace tak, že se pro každý aktuální výřez určuje, do které třídy spadá pixel na jeho prostřední pozici. Zde není nutná revize, tudíž klasifikace může probíhat za běhu programu. Testovací sada dat obsahuje velké množství výřezů, a proto je lépe ji načíst až po ukončení trénování, kdy je možno uvolnit paměť proměnných spjatých s trénováním a validací. Další důvod je čistě historický. Skript pro vytvoření trénovací databáze vznikl jako první a to ještě v době, kdy nebyla vybrána knihovna implementující neuronovou síť. Validační a testovací sada byly přizpůsobeny potřebě klasifikace za běhu programu.

Obrazy kontextů jsou následně popřeházeny, avšak napříč všech tří kontextů jsou obrazy popřeházeny ve stejném pořadí. Jedna z možností, kterou program nabízí je globální normalizace těchto T1-vážených a T2-vážených MRI výřezů pro potřeby korekce průměru a směrodatné odchylky v sadě testovacích obrazů. Globální normalizace spočívá ve výpočtu průměrné hodnoty a směrodatné odchylky napříč celou trénovací a následné odečtení těchto hodnot od hodnot intenzit jednotlivých obrazů testovací sady. Část sady může být zobrazena pro vizuální kontrolu. Na obr. 9.1 je pro ilustraci ukázka části trénovací sady pro segmentaci nádorů.

9.5 Model neuronové sítě

Dalším krokem po načtení trénovací a validační sady je sestavení modelů neuronových sítí. Zkonstruovaná neuronová síť pro trénování a segmentaci nádoru je uvedena v tab. 9.1 a síť pro trénování a segmentaci nádorů je uvedena v tab. 9.2.

První vrstva neuronové sítě načítá obrazy o velikosti receptivního pole. Receptivní pole pro segmentaci nádorů má velikost 60×60 pixelů a receptivní pole pro segmentaci edémů má 74×74 pixelů. Tato velikost receptivních polí byla zvolena s ohledem na průměrnou velikost daného útvaru v obrazech a s ohledem na změnu dimenzí receptivního pole během jeho průchodu neuronovou sítí.

9.6 Proces trénování a validace

Trénování a validace T1-vážených a posléze T2-vážených dat probíhalo naprosto separátně, neboť trénování rozpoznávání různých struktur může vyžadovat rozdílné

Tab. 9.1: Jednotlivé vrstvy sítě pro segmentaci nádorů

vrstva	velikost masky nebo podvzorkovací faktor	počet masek	výstup
viditelná	-	-	60x60
C1	9x9	30	52x52x30
C2	9x9	36	44x44x36
S1	2	-	22x22
C3	9x9	48	14x14x48
S2	2	-	7x7
plně-konekovaná	7x7	60	60x1

Tab. 9.2: Jednotlivé vrstvy sítě pro segmentaci edémů

vrstva	velikost masky nebo podvzorkovací faktor	počet masek	výstup
viditelná	-	-	74x74
C1	11x11	30	64x64x30
C2	11x11	36	54x54x36
C3	11x11	36	44x44x36
S1	2	-	22x22
C4	9x9	48	14x14x48
S2	2	-	7x7
plně-konekovaná	7x7	74	74x1

parametry, jako je počet epoch, rychlost učení, množství a rozměry testovacích obrazů. Dále je možno trénovat data v minidávkách o různé velikosti s hraniční hodnotou 1 obraz. Stejně jako v kapitole 3.2.2 plně-konektované vrstvy modelů „small“, „medium“ a „large“ se spojí do jedné lineární vrstvy tvořenou 180 resp. 222 neurony pro segmentaci nádoru, resp. edému. Teprve tato vrstva postupuje do *softmax* operátoru, jehož výstupem jsou dva neurony vyjadřující hodnotu predikce nádoru nebo pozadí, resp. edému nebo pozadí. Během trénování je dále vypočtena chyba klasifikace pomocí kritériální funkce *NLL* (viz. kapitola 1). Zpětná propagace chyby učení je obráceným postupem dopředné propagace a tedy nejprve se chyba propaguje z výstupní vrstvy do softmax vrstvy, zde se vektor chybových složek rozloží do tří lineárních vrstev o rozměrech 60×1 , resp. 74×1 neuronů a příslušný chybový vektor je propagován modely „small“, „medium“ a „large“ kontextů až do viditelné vrstvy.

Po každé epoše trénování probíhá validace na validačních datech. V této fázi se vypočítává přesnost modelu, tj. procento správně zařazených pixelů v obou třídách. Pokud toto procento přesáhne stanovenou mez přesnosti nebo pokud se významně

nemění po 3 epochy, je zahájena segmentace. Významným rozdílem je myšlena odchylka o 3 procenta a více.

9.7 Princip segmentace

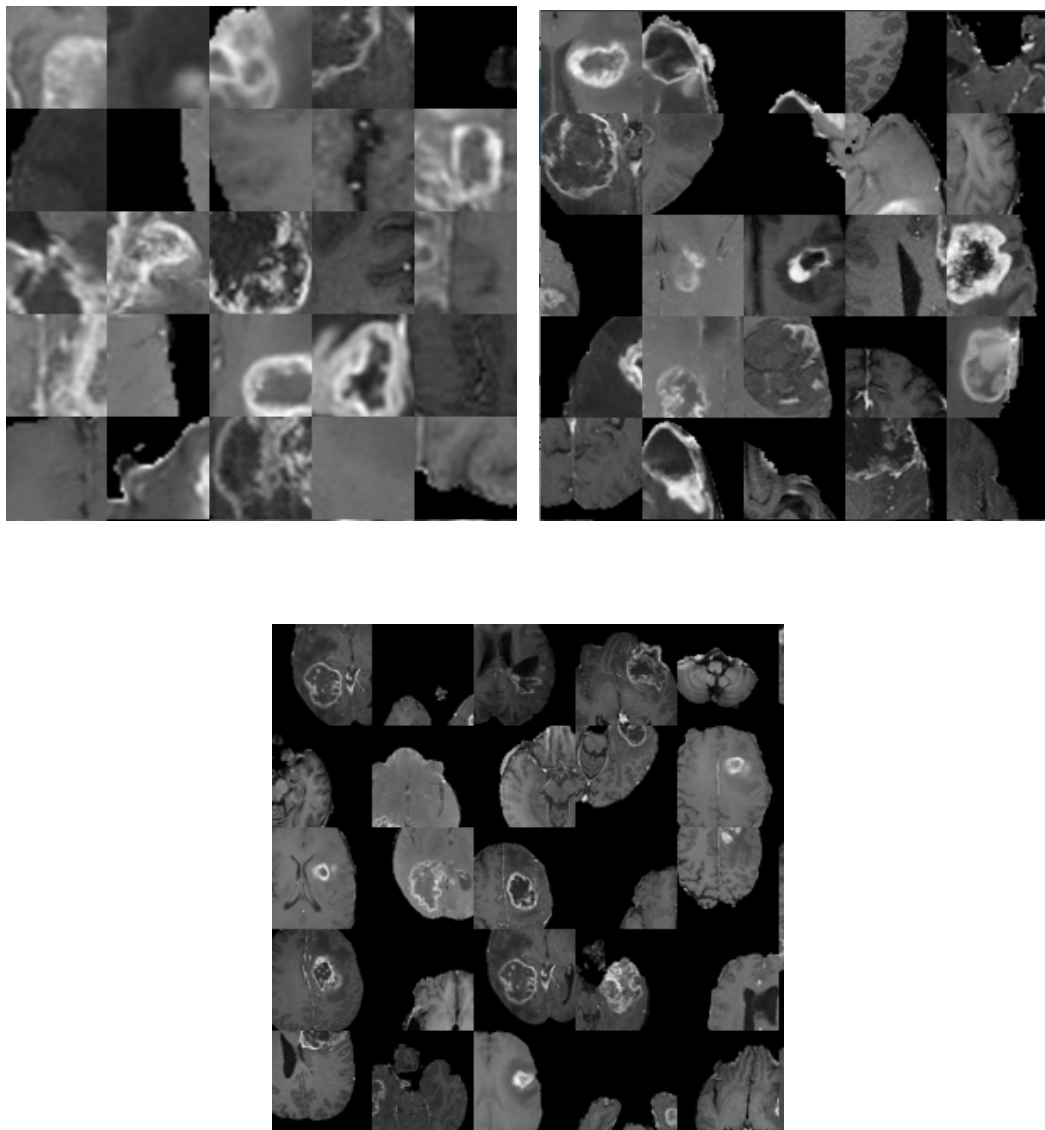
Po dokončení trénování je zahájena segmentace. Segmentace využívá natrénovaného modelu, do kterého vstupují postupně kontexty testovacích axiálních řezů jednoho pacienta. Okno, které provádí výřezy se pohybuje po jednom pixelu, takže ve výsledku je nasegmentován každý pixel obrazu. Aby to však platilo zcela, je nutné axiální řez před zahájením segmentace doplnit nulami po stranách nejméně o polovinu receptivního pole pro danou klasifikační třídu. Na základě hodnot predikce modelu jsou na příslušnou pozici trojrozměrné matice uloženy pixely obrazu odlišené barvou: červená - nádor, modrá - edém, šedá - pozadí. Po ukončení segmentace je matice uložena jako obraz na disk.

9.8 Výběr referenční metody pro srovnání výsledků segmentace

Jedním z cílů diplomové práce je srovnání přesnosti segmentace pomocí neuronové sítě a pomocí jiné metody určené pro segmentaci. Pro tuto variantu byla vybrána metoda kombinovaného prahování a fuzzy K-means shlukování. Základem je úprava kontrastu původního obrazu a použití morfologických operátorů. Poté obraz postupuje do K-means shlukování s předdefinovaným počtem shluků. Algoritmus K-means přiřadí každý pixel do určitého shluku podle jeho intenzity. Těžiště shluku se během algoritmu mění. Shluky jsou přiřazovány s určitou neurčitostí danou *fuzzy* konstantou, což je malé, náhodně vygenerované číslo. Každý shluk je odlišen barvou. Nakonec je vybrán shluk, který odpovídá segmentovanému útvaru. Algoritmus byl vyzkoušen na sadě 14 pacientů a nakonec sestaven tak, aby co možná nejlépe vyhovoval segmentaci dané struktury a nepřipustil mnoho falešně zařazených pixelů.

9.9 Srovnání výsledků segmentace

Pro srovnání výsledků segmentace jsou použity anotované obrazy z databáze BRATS 2015, které slouží jako *ground-truth*. Pomocí neuronové sítě a pomocí referenční metody vznikají pro každého pacienta dva obrazy - segmentace nádorů a edémů. Oba tyto obrazy se porovnávají s *ground-truth* obrazy, pro stanovení koeficientů podobnosti, senzitivity a specificity segmentace nádorů. Další podrobnosti budou uvedeny v kapitole 10.



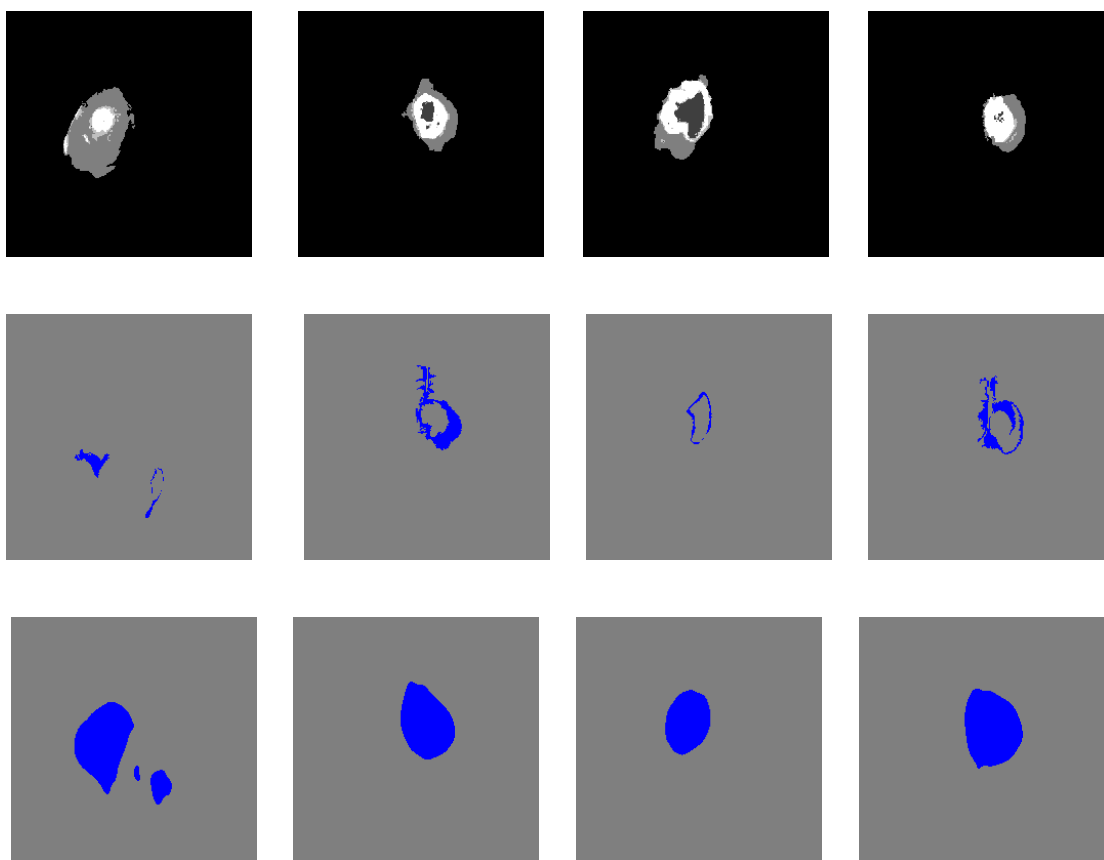
Obr. 9.1: Ukázka trénovací sady pro segmentaci nádorů - velký kontext (vlevo nahoře), středí kontext (vpravo nahoře), malý kontext (dole). Obrázky v trénovací sadě obsahují snímky nádorů i pozadí a jsou náhodně popřeházeny.

10 VÝSLEDKY A DISKUZE

V této části práce budou shrnuty a zhodnoceny dosažené výsledky.

10.1 Segmentace edémů

Na obr. 10.1 je ukázka několika výsledků segmentace edému jak pomocí konvoluční neuronové sítě, tak pomocí metody kombinovaného prahování a fuzzy k-means ve srovnání s anotovanými obrazy poskytnutými v databázi BRATS 2015. Výsledky pro demonstraci byly vybrány od čtyř různých pacientů. Na tomto obrázku je ukázáno,



Obr. 10.1: Srovnání úspěšnosti segmentace, anotované obrazy (nahore), segmentace metodou jednoduchého prahování (uprostřed), pomocí CNN (dole)

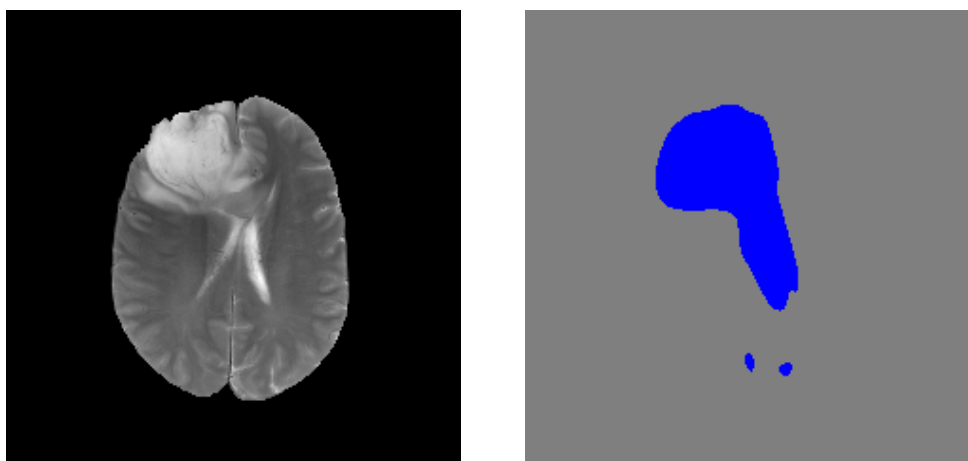
že obě metody segmentace jsou velice neúspěšné. V obou případech se na neúspěchu podílí velká biologická variabilita edému, která ve snímcích MRI způsobuje značný rozptyl nejen v intenzitě, ale i ve tvaru edému. Navíc může edém na snímcích MRI zanikat v okolní mozkové tkáni, zvláště poblíž mozkových komor, které tak mohou být falešně segmentovány, jak je ukázáno na obr. 10.2. Metoda CNN nesegmentovala jen edém, ale také nádor, který se na T2-vážených snímcích projevuje v různých

způsobem, nemusí být vůbec rozpoznatelný nebo může jeho struktura splynout se strukturou edému. V případě segmentace neuronovou sítí byly struktury edému spíše přesegmentovány. Dle hodnot Dice-Sørensenových a Jaccardových indexů uvedených v tab. 10.1 byla úspěšnost segmentace vyšší než v případě metody kombinovaného prahování. Pokud by se spojily výsledky segmentace nádorů a edému pro tyto případy, mohla by být přesnost určení edému ještě vyšší. V případě metody kombinovaného prahování byly edémy spíše podsegmentovány. Hodnoty Dice-Sørensenových a Jaccardových indexů jsou velmi nízké, avšak většinou zde nedochází k nežádoucímu vysegmentování nádoru. Koeficienty podobnosti, senzitivity a specifity všech testovaných obrazů jsou uvedené v příloze, k nalezení zde: B.

Tab. 10.1: Srovnání průměrné hodnoty a směrodatné odchylky koeficientů podobnosti u segmentovaných edémů

-	průměr (prahování)	sm. odchylka (prahování)	průměr (cnn)	sm. odchylka (cnn)
Dice index	0,268	0,185	0,416	0,147
Jaccard index	0,112	0,073	0,169	0,051
Senzitivita index	0,261	0,215	0,794	0,233
Specifita index	0,990	0,005	0,952	0,035

Na obr. 10.2 je případ selhání metody segmentace pomocí CNN. V tomto případě došlo k tomu, že neuronová síť chybně vysegmentovala i mozkové komory. Objevily se však i případy, kdy neuronová síť vysegmentovala chybně celý útvar mozkové tkáně.



Obr. 10.2: Metodou CNN byla falešně vysegmentována pravá mozková komora

10.2 Segmentace nádorů

Pokus o segmentaci nádorů byl nejprve proveden na prvním vybraném pacientovi. Pro tento pokus bylo vybráno 14 řezů obsahujících nádorovou tkáň. V tab. 10.2 jsou uvedeny výsledky Dice, Jaccardova indexu, sensitivity a specificity, použité pro srovnání obrazů segmentovaných jednoduchým prahováním a referenčních obrazů pro prvního pacienta.

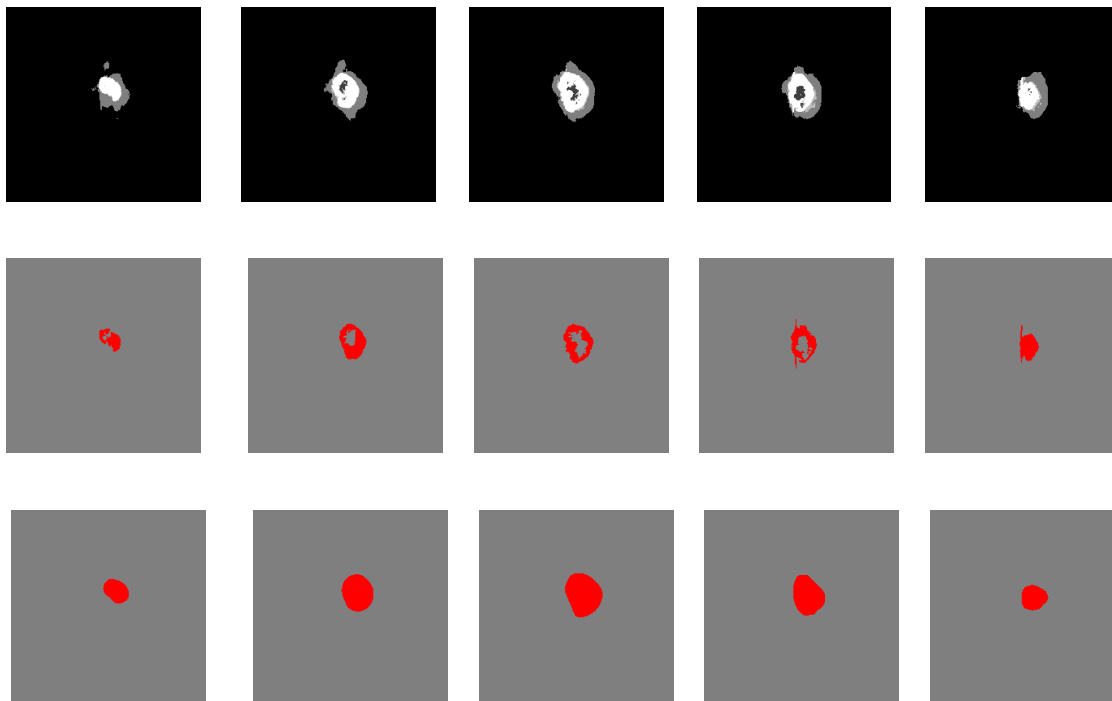
Tab. 10.2: Srovnání průměrné hodnoty a směrodatné odchylky koeficientů podobnosti u segmentovaných nádorů u prvního pacienta

-	průměr (prahování)	sm. odchylka (prahování)	průměr (cnn)	sm. odchylka (cnn)
Dice index	0,474	0,120	0,552	0,242
Jaccard index	0,189	0,038	0,209	0,078
Senzitivita index	0,360	0,138	0,678	0,319
Specificita index	0,995	0,012	0,973	0,023

Na obr. 10.3 bylo vybráno několik MRI řezů druhého testovaného pacienta a lze vidět, jak se chová segmentace pomocí neuronové sítě, vůči segmentaci pomocí referenční metody. Referenční metoda vysegmentovala vcelku přesně světlý okraj nádoru avšak vnitřní jádro zde vidět není. Neuronová síť dodržela integritu tohoto útvaru, avšak nedodržela přesně jeho tvar a také jej po okrajích přesegmentovala.

Na obr. 10.4 je pro srovnání výsledek použití větší trénovací sady pro segmentaci nádorů. Trénovací sada byla rozšířena o vzorky stejných pacientů, pro získání receptivních polí byl však použit menší krok, se kterým okno prochází jednotlivé řezy a byly použity téměř sousedící řezy. Výsledek je takový, že pravděpodobně došlo k přeučení modelu. Model sice klasifikoval trénovací sadu s vyšší přesností, avšak na testovací sadě následně segmentoval i útvary nesouvisející morfologicky se strukturou nádoru.

Pro vytvoření 3D modelu segmentovaného nádoru byla použita testovací sada pro druhého vybraného pacienta. Hotové modely jsou uvedeny na obr. 10.5. Ukázka segmentace pomocí neuronové sítě je vpravo nahoře, ukázka segmentace pomocí referenční metody je dole uprostřed a anotovaný 3D model se nachází vlevo nahoře. Dice a Jaccardovy indexy podobnosti mezi obrazy segmentovanými metodou CNN a referenčními obrazy byly vyšší než mezi obrazy segmentovanými kombinovaným prahováním. Diceho index podobnosti, který vyjadřuje, jak velký průnik tvoří dvě množiny u obrazů segmentovaných metodou CNN je více 55% u prvního pacienta a 79% u druhého pacienta, tedy více než polovina pixelů patřících do třídy nádor se shodovala. Celkově jsou procenta sensitivity u metody CNN výrazně vyšší, což



Obr. 10.3: Srovnání úspěšnosti segmentace, anotované obrazy (nahore), segmentace metodou jednoduchého prahování (uprostřed), pomocí CNN (dole)

znamená, že síť lépe rozpoznávala nádor než prahovací metoda. Procenta specifity jsou však typicky nižší, což znamená, že síť nasegmentovala více falešně pozitivních případů (často také přesegmentovala) než metoda prahování, kde se specifita pohybuje těsně pod hranicí 1.

Na obr. 10.6 vytvořily T1-vážené MRI řezy 3D strukturu, ve které je uložen vysegmentovaný nádor a je tak patrné, kde je nádor lokalizován.

10.3 Spojení výsledků segmentace nádorů a edémů

Při spojování výsledků segmentace nádoru a edému se postupuje tak, že pixely segmentovaného nádoru se ve výsledném obraze vždy projeví - dochází k nahrazení pixelů klasifikovaných jako edém nebo pozadí pixely klasifikovanými jako nádor. Výsledky všech spojených obrázků jsou k nalezení v příloženém DVD. V této formě dosahuje segmentace nádoru a edému pomocí CNN viditelně lepších výsledků, než pomocí metody kombinovaného prahování a je takto lépe použitelná pro praxi.

Na obrázcích segmentace metodou CNN i metodou prahování lze vidět, že segmentace pomocí neuronové sítě má menší rozlišovací schopnost než segmentace pomocí jednoduchého prahování, proto nekopíruje přesně tvar segmentované struktury, a proto se také objevuje přesegmentování. Metoda segmentace prahováním sice tvar

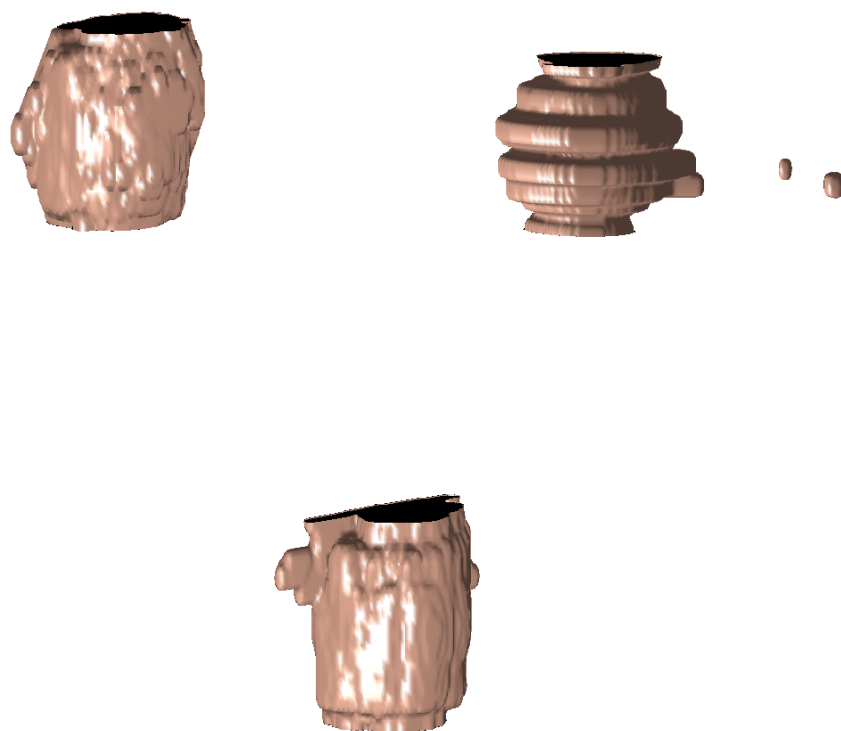


Obr. 10.4: Použití větší trénovací sady (vpravo) v porovnání s menší trénovací sadou (vlevo).

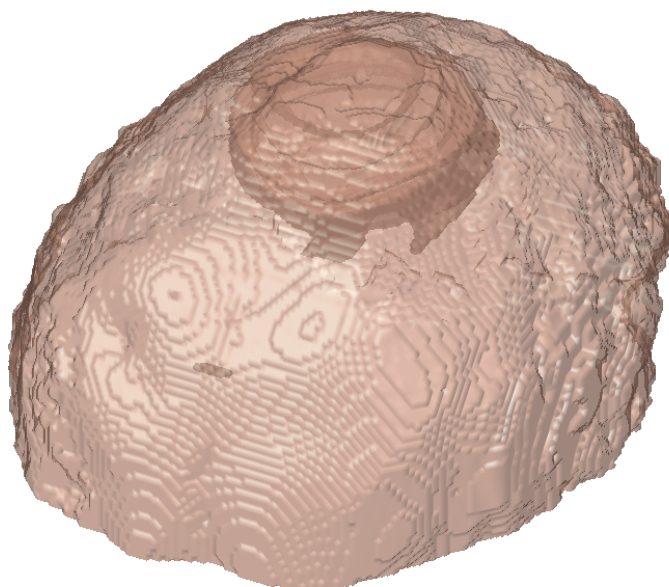
Tab. 10.3: Srovnání průměrné hodnoty a směrodatné odchylky koeficientů podobnosti u segmentovaných nádorů u druhého pacienta

-	průměr (prahování)	sm. odchylka (prahování)	průměr (cnn)	sm. odchylka (cnn)
Dice index	0,812	0,074	0,795	0,100
Jaccard index	0,288	0,018	0,283	0,002
Senzitivita index	0,707	0,122	0,954	0,088
Specificita index	0,999	0,001	0,994	0,008

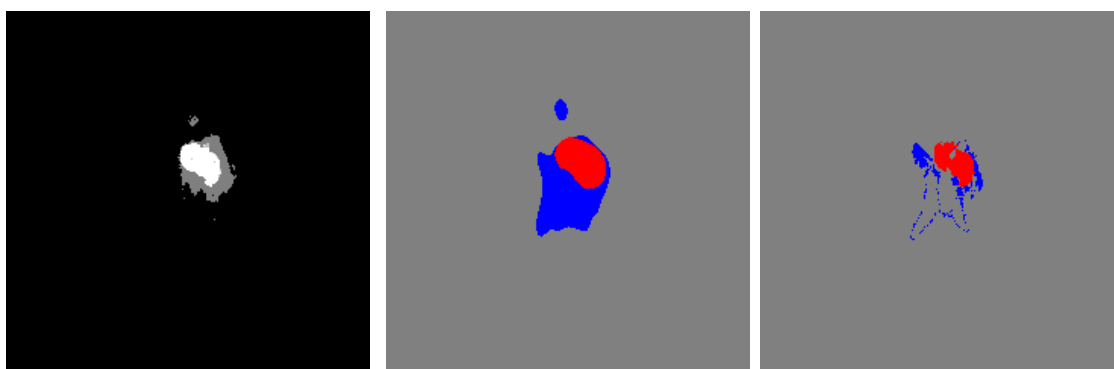
kopíruje vcelku věrně, ale selhává jakmile intenzita pixelu na dané pozici nepřesahuje stanovenou mez. Tento práh byl stanoven fixně odhadem po slédnutí několika MRI snímků, zatímco u metody CNN se práh neuronů stanovuje adaptivně během slédnutí velkého počtu MRI snímků.



Obr. 10.5: Anotovaný obrázek (vlevo nahoře) Metoda CNN (vpravo nahoře), metoda kombinovaného prahování(dole uprostřed)



Obr. 10.6:



Obr. 10.7: Výsledky spojení segmentace nádorů a edémů. Anotovaný obrázek (vlevo), metoda CNN (uprostřed), metoda kombinovaného prahování(vpravo)

ZÁVĚR

Automatická segmentace mozkových struktur ze snímků MRI pomocí výpočetní techniky není triviální záležitost. Především proto, že struktur, nacházejících se v napříč celou mozkovou tkání, je mnoho a jsou ve snímcích MRI odlišeny pouze v odstínech šedi. Díky kontrastním látkám se jejich intenzita v obraze výrazně liší oproti intenzitě okolních struktur, což je pro zpracování výpočetní technikou velká výhoda. Avšak kontrast je pouze to jediné, co je od okolní tkáně může odlišovat. Ve tvaru a lokalizaci struktur nádoru a především edému mezi jednotlivými pacienty je velká variabilita, vyplývající z variability biologické. Během segmentace struktur dochází k přesahování okrajových částí a to z důvodu omezené rozlišovací schopnosti implementovaných neuronových sítí. Celková chybovost sítě může být způsobena výskytem dominantních struktur (struktury s mnohem vyšším jasnem vůči svému okolí, nebo struktury s výraznými hranami). Takové struktury mohou být například mozkové komory nebo velké tepny. Dále také platí, že učení neuronové sítě je multifaktoriální proces. Účinnost sítí by se mohla zvýšit vhodnou změnou parametrů sítě, změnou konfigurace sítě (např. použití jiných metod uvedených v kapitole 7) nebo volbou jiného principu segmentace pomocí neuronové sítě. Jedním z těchto principů, které by mohly být účinné je víceparametrická segmentace, kdy se k segmentaci použije kombinace modalit MRI zobrazení.

Trénovací sada pro segmentaci nádoru prvního pacienta byla tvořena 7500 vzorky, z toho bylo 3500 vzorků nádoru a 4000 vzorků pozadí, trénování trvalo přibližně 4 hodiny (5 epoch) a úspěšnost algoritmu na validační sadě byla přes 94%, což byla nejvyšší dosažená úspěšnost. Trénovací sada pro segmentaci nádoru druhého pacienta byla původně tvořena 8500 vzorky, z toho bylo 4000 vzorků nádoru a 4500 vzorků pozadí, trénování trvalo přibližně 5 hodin a úspěšnost algoritmu na stejné validační sadě byla přes 93%. Protože však výsledky této segmentace nebyly dostatečně kvalitní, byly pouze uvedeny pro ilustraci v kapitole 10, ale nebyly k segmentaci použity. Místo toho byl použit původní model natrénovaný pro prvního pacienta. Samotná segmentace každého T1-váženého obrazu trvala hodinu a půl. Trénovací sada pro segmentaci edémů byla tvořena 5390 vzorky, z toho bylo 2690 vzorků nádoru a 2700 vzorků pozadí, trénování trvalo 1,5 dne, úspěšnost algoritmu na validační sadě byla 89%. Segmentace každého T2-váženého obrazu trvala 4,5 hodiny. Tyto parametry byly změřeny na CPU se čtyřmi jádry, kdy procesy běžely celkem v osmi vláknech, s 8GB RAM a rychlostí učení 0.0005. Vytížení každého jádra bylo přibližně 25%.

Hlavní výhoda použití neuronové sítě, vůči jednoduchému prahování by měla spočívat v tom, že neuronová síť se „naučí“ rozpoznávat pouze určitou strukturu a ostatní struktury rozpoznává jako pozadí, ačkoliv tyto struktury mohou mít stejnou

hodnotu jasů. Nevýhodou použití konvoluční neuronové sítě je fixní velikost receptivního pole, která musí být přizpůsobena velikosti objektu, jinak dochází k fragmentaci objektu nebo není objekt vůbec detekován. Trénovací sada, která se vytváří *parsováním* T1-vážených a T2-vážených obrazů musí být sestavena tak, aby dokázala segmentovat s dostatečnou správností a rozlišením, což jsou dva protichůdné parametry. Dostatečnou správnost klasifikace zajistí sestavení trénovací sady tak, aby vzorky spadající do různých tříd byly co nejvíce exkluzivní, naopak dostatečné rozlišení je dáno počtem pixelů v okolí středového pixelu vzorku, které v součtu klasifikují obraz do dané třídy. Jestliže pro sestavení trénovací sady bylo zvoleno okolí 7x7 pixelů pro klasifikaci do třídy nádor, bude teoreticky minimální odchylka pixelu na středové pozici segmentovaného vzorku 3 pixely každým směrem.

Přestože jsou metody segmentace použité v této práci založené na odlišných principech segmentace, výsledky jsou podobné. Na obrázcích v kapitole 10) lze vidět, jak se od sebe liší výsledky dvou metod, z nichž první metoda - konvoluční sítě - působí jako lokální operátor, tedy o hodnotě pixelu rozhodují hodnoty okolních pixelů, zatímco druhá metoda - metoda prahování působí jako bodový operátor, tedy okolní pixely nemohou ovlivnit hodnotu výsledného pixelu. I přes relativně vysoká procenta úspěšnosti na validačních datech nebyla úspěšnost segmentace na testovacích datech příliš vysoká. Důvodů lze najít hned několik. Samotná validační sada nebyla ani zdaleka tvořena tolika vzorky, kolik bylo po ukončení trénování testováno. V demo projektech pro rozpoznávání ručně psaných číslic vytvořených v MATLABu nebo Torch se původní sada dat rozdělila v poměru 5:1 na trénovací a validační data. Tento poměr by teoreticky byl dostatečný.

K účelům této práce bylo vyzkoušeno více knihoven, tyto knihovny jsou stručně popsány v kapitole 5). Nejprve byl testován Toolbox pro MATLAB dostupný ze serveru github.com (viz. [12]). Zdrojové kódy jsou napsány velmi srozumitelně, proto je vhodný pro pochopení hlubší podstaty a principu metod hlubokého učení. V tomto Toolboxu je k dispozici demo pro klasifikaci ručně psaných číslic, které vykazovalo dobré výsledky. Vlastní implementace neuronové sítě pro segmentaci byla však z neznámých důvodů neúspěšná. Dále byla testováno použití knihovny Theano ve vývojovém prostředí Spyder. Použití této knihovny je kvůli již zmíněným omezeným možnostem ladění opravdu nepraktické a implementace vlastní knihovny by vyžadovala enormní úsilí. Nakonec byla vyzkoušena knihovna Torch, její skripty byly spouštěny z příkazového řádku. Tato knihovna nakonec díky přívětivému uživatelskému rozhraní byla v práci použita.

V této práci byla vytvořena trénovací, validační a testovací sada pro segmentaci vybrané mozkové tkáně a sestavení 3D modelu segmentovaného útvaru. Celkově bylo dosaženo Dice koeficientů 79,5% pro segmentaci nádorů a 41% pro segmentaci edémů.

LITERATURA

- [1] HERNANDEZ, D.: *Meet the man Google hired to make AI a reality.* [online] <http://www.wired.com/2014/01/geoffrey-hinton-deep-learning>, www.wired.com
- [11] GOODFELLOW, I., COURVILLE, A. and YOSHUA, B.: *Deep Learning.* [online] <http://goodfeli.github.io/dlbook> Book in preparation for MIT Press, 2015
- [3] CRIPAC, NLPR, CASIA, BAIDU, ZIFENG, W., YONGZHEN, H., YINAN, Y., LIANG, W. and TIENIU, T.: *Early Hierarchical Context Learned by Convolutional Networks for Image Segmentation.* 2014
- [4] HYEONWOO, N., SEUNGHOON, H. and BOHYUNG, H.: *Learning Deconvolution Network for Semantic Segmentation.* 2015
- [5] ZEILER, D. M., TAYLOR, W. G., FERGUS, R., : *Adaptive Deconvolutional Networks for Mid and High Level Feature Learning.* Dept. of Computer Science Courant institute, New York, 2011
- [6] LAI, M.: *Deep learning for medical image segmentation.* 2015
- [7] HINTON, G., CICAR: *Deep Belief Nets.* Department of Computer Science, University of Toronto, 2007
- [8] LAROCHELLE, H.: *Restricted Boltzmann machine.* [online]<https://www.youtube.com/watch?v=MD8qXWucJBY>, videolectures, 2013
- [9] HINTON, G.: *A practical Guide to Training Restricted Boltzmann Machines.* Dept. of Computer Science, University of Toronto, Version 1, 2010-2013
- [10] CHEN, E.: *Introduction to Restricted Boltzmann Machines.* [online]<http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines>, 2011
- [2] Theano Development Team: *Deep Learning Tutorial.* [online]<http://deeplearning.net/tutorial/index.html>, 2008-2013
- [12] PALM, R., B.: *Prediction as a candidate for learning deep hierarchical models of data.* [online]<https://github.com/rasmusbergpalm/DeepLearnToolbox>, 2012
- [13] EILSCHOU, A.: *Deep Learning.* 2014
- [14] FISHER, A. and IGEL, Ch.: *Training Restricted Boltzmann Machines: An Introduction.* Department of Computer Science, University of Copenhagen, 2014

- [15] KEYVANRAD, A., M.: *DeeBNet (Deep Belief Network) toolbox*. [online] <http://ceit.aut.ac.ir/~keyvanrad/index.html> University of Technology, Tehran, Iran, V2.2, 9/7/2015
- [16] KOZUMPLIK, J., PROVAZNÍK, I.: *Umělá inteligence v medicíně*. Fakulta elektrotechniky a komunikačních technologií Vysoké učení technické v Brně, Brno, 2007
- [17] Ng, A.: *Sparse autoencoder*. [staženo z:] <https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf>
- [18] BENGIO, Y.: *Learning Deep Architectures for AI*. [staženo z:] <http://www.iro.umontreal.ca/~bengioy/papers/ftml.pdf> Foundations and Trends in Machine Learning, 2009
- [18] HINTON, G., DAYAN, P., FREY, J. B., NEAL, R. : *The wake-sleep algorithm for unsupervised neural network*. [staženo z:] <http://www.gatsby.ucl.ac.uk/dayan/papers/hdfn95.pdf> Dept. of Computer Science, University of Toronto, 1
- [19] OSBORNE, M.: *Entropy and Kullback-Leibler Divergence*. School of Informatics, University of Edinburg, 2008
- [20] FARABET, C., COUPRIE, C., NAJMAN, L., LECUN, Y.: *Scene parsing with multiscale feature learning, Purity trees and optimal Covers*. Courant Institute of Mathematical Sciences and Université Paris-Est, 2008
- [21] HINTON, G., CIRAF: *Deep learning with multiplicative interactions*. Dept. of Computer Science, University of Toronto, 2010
- [22] CZARNIECKI, M., JONES, J., et al.: *T2 weighted image*. [online] <http://radiopaedia.org/articles/t2-weighted-image>
- [23] GAI, D., JONES, J., et al.: *T1 weighted image*. [online] <http://radiopaedia.org/articles/t1-weighted-image>
- [24] MENZE, H. B., JAKAB, A., BAUER, S., KLAPATHZ-CRAMER, J., FARAHANI, K., et al.: *The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)*. [open-access] <http://braintumorsegmentation.org/>, IEEE transactions on medical imaging, vol. 34, no. 10., p. 1993, 2015
- [25] DICE, L. R.: *Measures of the Amount of Ecologic Association Between Species*. Ecology, vol. 26, pp.297-302, 1945

- [26] JACCARD, P.: *The distribution of the flora in the alpine zone*. The new phytologist, vol. 11, no. 2, 1912
- [27] ŠKUTKOVÁ, H.: *Rozhodování v biomedicíně*. [přednáška] Úvod do medicínské informatiky, 2016
- [28] Lua *Documentation*. [online] <https://www.lua.org/docs.html>, [Cit.: 7.5.2016], 2016
- [29] MathWorks *Documentation*. [online] <http://www.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html>, [Cit.: 8.5.2016], 2016
- [30] ERHAN, D., COURVILLE, A., Bengio, Y., VINCENT, P.: *Why does unsupervised pre-training help deep learning*. Journal of Machine Learning Research, vol.11, pp. 625-660, 2010
- [31] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., SALAKHUTDINOV, R.: *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research, vol.15, pp. 1929-1958, 2014
- [32] NIELSEN, A., M.: *Neural networks and deep learning*. Determination Press, 2015

SEZNAM PŘÍLOH

A	Pozitivní a negativní fáze učení u <i>Restricted Boltzmann machines</i>	69
	A.0.1 Pozitivní fáze	69
	A.0.2 Negativní fáze	69
B	Hodnoty Dice-Sørensenova, Jaccarova indexu, sensitivity a specificity pro všechny měřené obrazy	71
C	Obsah přiloženého DVD	74
D	Schéma konvoluční sítě LeNet	77

A POZITIVNÍ A NEGATIVNÍ FÁZE UČENÍ U *RESTRICTED BOLTZMANN MACHINES*

A.0.1 Pozitivní fáze

Vyjadřuje propagaci pozitivních stavů viditelné vrstvy do skryté vrstvy.

Nejprve se vypočtou preaktivační výstupy viditelné vrstvy.

$$positive_hidden_activations = \sum weights \cdot positive_visible_states$$

aplikuje se nelinearita

$$positive_hidden_probability = \sigma(positive_hidden_activations)$$

vygenerování náhodného čísla a porovnání s pravděpodobností vyjadřuje, zda jsou pravděpodobnosti daných konfigurací neuronů viditelné a skryté vrstvy dostatečně vysoké nebo ne. Stav skryté vrstvy se tedy nastaví s pravděpodobností $p(h_j = 1|v)$ na jedničku a na nulu s pravděpodobností $1 - p(h_j = 1|v)$.

$$positive_hidden_states = positive_hidden_probability < U\langle 0\ 1 \rangle$$

následně se vypočítají asociace s použitím pravděpodobností nebo stavů. Tato proměnná vyjadřuje vztahy mezi parametry viditelné vrstvy navzájem. Pokud měla vstupní proměnná velkou četnost, pak jsou tyto pozitivní asociace vysoké. Nebo naopak pokud měla nízkou četnost pak budou tyto asociace blízké nule.

$$positive_associations = \sum_i (positive_hidden_states)^T \cdot data$$

eventuálně

$$positive_associations = \sum_i (positive_hidden_probability)^T \cdot data$$

A.0.2 Negativní fáze

Negativní fáze je složitější než pozitivní fáze. V této fázi se jedná o rekonstrukci vstupních dat bez znalosti jejich rozložení pouze z nastavení vstupních neuronů a znalosti o parametrech modelu.

výpočet preaktivačních výstupů z neuronů skryté vrstvy:

$$negative_visible_activations = \sum_i positive_hidden_states \cdot weights^T$$

aplikace nelinearity:

$$negative_visible_probability = \sigma(negative_visible_activations)$$

a výpočet negativních stavů viditelné vrstvy - rekonstruovaná data podobně jako byly u pozitivní fáze vyjádřeny stavy skryté vrstvy:

$$negative_visible_states = negative_visible_probability < U\langle 0\ 1 \rangle$$

a opět následuje propagace rekonstruovaných dat do skryté vrstvy.

$$negative_hidden_activations = \sum_j negative_visible_states \cdot weights^T$$

dále se vypočítá pravděpodobnostní rozložení

$$negative_hidden_probability = \sigma(negative_hidden_activations)$$

a asociace mezi neurony vrstev

$$negative_associations = \sum_i (negative_hidden_probabilities)^T \cdot negative_visible_states$$

poslední tři kroky odpovídají prvním třem krokům v pozitivní fázi s tím rozdílem, že se provádějí nad daty rekonstruovanými modelem. Chyba sítě je určena rozdílem mezi vstupními daty a jejich rekonstrukcí:

$$e = (data - negative_visible_states).$$

Pozitivní a negativní fáze se může opakovat v závislosti na zvoleném typu vzorkování - viz. kap. 4.1.3.

Toto schéma bylo převzato z algoritmů pro *Restricted Boltzmann Machines* zde:[12]

B HODNOTY DICE-SØRENSENNOVA, JACCAROVA INDEXU, SENSITIVITY A SPECIFICITY PRO VŠECHNY MĚŘENÉ OBRAZY

Tab. B.1: Srovnání úspěšnosti segmentace edémů metodou kombinovaného prahování a referenčních obrazů

číslo obrazu	49	63	71	76	77	87	88	94	95	97	98	99	103	110	112	průměr	směr. odchylka
Dice index	0,50	0,30	0,45	0,0	0,18	0,31	0,15	0,12	0,43	0,0	0,54	0,29	0,28	0,0	0,40	0,26	0,18
Jaccard index	0,20	0,13	0,18	0,0	0,08	0,13	0,07	0,05	0,17	0,0	0,21	0,12	0,12	0,0	0,16	0,11	0,07
Senzitivita	0,74	0,24	0,38	0,0	0,10	0,28	0,13	0,09	0,45	0,0	0,49	0,24	0,29	0,0	0,44	0,26	0,21
Specificita	0,97	0,98	0,99	0,99	0,99	0,99	0,98	0,99	0,98	0,99	0,99	0,99	0,98	0,99	0,98	0,99	0,01

Tab. B.2: Srovnání úspěšnosti segmentace edémů metodou CNN a referenčních obrazů

číslo obrazu	49	63	71	76	77	87	88	94	95	97	98	99	103	110	112	průměr	směr. odchylka
Dice index	0,19	0,44	0,41	0,43	0,70	0,33	0,37	0,52	0,53	0,34	0,57	0,33	0,51	0,12	0,39	0,41	0,14
Jaccard index	0,08	0,18	0,17	0,17	0,26	0,14	0,15	0,20	0,21	0,14	0,22	0,14	0,20	0,05	0,16	0,16	0,05
Senzitivita	1,0	0,47	0,93	0,61	0,77	0,81	0,92	0,90	0,98	0,57	0,95	0,77	0,97	0,20	1,0	0,79	0,23
Specificita	0,83	0,97	0,94	0,96	0,97	0,96	0,94	0,96	0,96	0,94	0,97	0,95	0,96	0,96	0,95	0,95	0,03

Tab. B.3: Srovnání úspěšnosti segmentace nádorů metodou kombinovaného prahování a referenčních obrazů - první pacient

číslo obrazu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	průměr	směr. odchylka
Dice index	0,62	0,49	0,46	0,29	0,41	0,41	0,38	0,44	0,46	0,33	0,41	0,51	0,67	0,42	0,47	0,12
Jaccard index	0,23	0,19	0,19	0,12	0,17	0,17	0,16	0,18	0,18	0,14	0,17	0,20	0,25	0,17	0,18	0,03
Senzitivita	0,45	0,33	0,31	0,34	0,26	0,26	0,24	0,28	0,30	0,20	0,26	0,34	0,54	0,65	0,36	0,13
Specificita	0,99	0,99	0,99	0,95	0,99	0,99	1,0	1,0	0,99	0,99	1,0	0,99	0,99	0,97	0,99	0,01

Tab. B.4: Srovnání úspěšnosti segmentace nádorů metodou CNN a referenčních obrazů - první pacient

číslo obrazu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	průměr	směr. odchylka
Dice index	0,26	0,24	0,19	0,26	0,72	0,77	0,73	0,70	0,73	0,79	0,72	0,71	0,59	0,25	0,55	0,24
Jaccard index	0,11	0,11	0,08	0,11	0,26	0,27	0,26	0,26	0,26	0,28	0,26	0,26	0,22	0,11	0,20	0,07
Senzitivita	0,25	0,26	0,15	0,17	0,82	0,78	0,72	0,74	0,75	0,91	1,00	0,88	1,0	1,0	0,67	0,31
Specificita	0,98	0,97	0,98	0,99	0,97	0,98	0,98	0,97	0,98	0,98	0,96	0,97	0,95	0,89	0,97	0,02

Tab. B.5: Srovnání úspěšnosti segmentace nádorů metodou kombinovaného prahování a referenčních obrazů - druhý pacient

[illegible]

Tab. B.6: Srovnání úspěšnosti segmentace nádorů metodou kombinovaného prahování a referenčních obrazů - druhý pacient - pokračování

číslo obrazu	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	průměr	směr. odchylka
Dice index	0,76	0,72	0,75	0,75	0,73	0,71	0,74	0,90	0,91	0,91	0,85	0,84	0,86	0,88	0,87	0,86	0,81	0,07
Jaccard index	0,27	0,26	0,27	0,27	0,26	0,26	0,27	0,31	0,31	0,31	0,30	0,29	0,30	0,30	0,30	0,30	0,28	0,01
Senzitivita	0,62	0,56	0,60	0,60	0,58	0,56	0,60	0,85	0,88	0,89	0,78	0,74	0,78	0,83	0,85	0,86	0,70	0,12
Specifická	1,0	1,0	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,0

Tab. B.7: Srovnání úspěšnosti segmentace nádorů metodou CNN a referenčních obrazů - druhý pacient

číslo obrazu	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102
Dice index	0,65	0,77	0,88	0,87	0,72	0,87	0,89	0,87	0,58	0,80	0,81	0,80	0,56	0,74	0,83	0,87	0,85
Jaccard index	0,24	0,27	0,30	0,30	0,26	0,30	0,30	0,30	0,22	0,28	0,29	0,28	0,22	0,27	0,29	0,30	0,29
Senzitivita	1,0	1,0	0,98	0,89	0,56	0,78	0,98	0,99	1,0	1,0	1,0	1,0	1,0	1,0	0,99	0,99	0,98
Specifická	0,99	0,99	0,99	0,99	1,0	0,99	0,99	0,99	0,97	0,98	0,99	0,98	0,96	0,98	0,99	0,99	0,99

Tab. B.8: Srovnání úspěšnosti segmentace nádorů metodou CNN a referenčních obrazů - druhý pacient - pokračování

číslo obrazu	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	průměr	směr. odchylka
Dice index	0,85	0,7	0,86	0,75	0,86	0,91	0,86	0,79	0,78	0,90	0,87	0,85	0,84	0,81	0,78	0,53	0,79	0,100
Jaccard index	0,29	0,25	0,24	0,27	0,30	0,31	0,30	0,28	0,28	0,31	0,30	0,29	0,29	0,29	0,28	0,21	0,28	0,02
Senzitivita	0,99	1,0	1,0	1,0	0,98	0,96	0,95	0,99	0,99	0,96	0,87	0,84	0,85	0,88	0,96	1,0	0,95	0,08
Specifická	0,99	0,97	0,97	0,98	0,98	0,99	0,99	0,98	0,98	0,99	0,99	0,99	0,99	0,99	0,99	0,98	0,99	0,0

C OBSAH PŘÍLOŽENÉHO DVD

K této diplomové práci je přiložen DVD nosič na němž se nachází složky:

- složka s výsledky - zde jsou k nalezení výsledky segmentace nádorů, edému jak s pomocí konvoluční sítě, tak s pomocí modely kombinovaného prahování a natrénované modely, ukázka 3D modelů nádorů,
- složka s anotovanými obrazy,
- složka s T1-váženými MRI obrazy pro segmentaci nádorů a trénovací sada,
- složka s T2-váženými MRI obrazy pro segmentaci edému a trénovací sada,
- složka *BRATS* obsahující složku *SlicerDatabase* - série DICOM obrazů pro jednoho pacienta, složka *PNG* - do ní mohou být uloženy výsledky konverze z DICOM do PNG formátu.
- složka *test* ve které jsou T1-vážené/T2-vážené a anotační obrazy k validaci a testování T1/T2 modelu modelu

dále se zde nachází zdrojové kódy pro knihovnu Torch i pro pre- a post-processing v MATLABu.

1. skripty pro pre- a post- processing v MATLABu

- **loadDicoms.m** - načtení DICOM sérií z určené složky (defaultně *SlicerDatabase* složka) a převedení do formátu PNG a uložení do zvolené složky (defaultně *PNG*).
- **load_MRI_data.m** - načtení řezů ze složky *PNG*.
- **parse_T1_MRI_image.m** - funkce volaná ve skriptu *load_MRI_data.m* provede vytvoření receptivních polí pro třídu. nádor/pozadí. Přijímá jako argumenty T1-vážené obrazy a anotované obrazy ve formátu PNG. Výsledky jsou uloženy do složky *BRATS/T1dir* pod názvem uvedeným ve třetím argumentu funkce.
- **parse_T2_MRI_image.m** - provádí vytvoření receptivních polí pro třídu edém/pozadí. Přijímá jako argumenty T2-vážené obrazy a anotované obrazy ve formátu PNG. Výsledky jsou uloženy do složky *BRATS/T2dir* pod názvem uvedeným ve třetím argumentu funkce. Není volána ve skriptu *load_MRI_data.m*.
- **segmentT1.m** - provede načtení obrázků ze složky *PNG* a pomocí funkce *appliedThresholdKmeans* provede metodou kombinovaného prahování a fuzzy shlukování segmentaci nádorové tkáně z T1-vážených obrazů.
- **appliedThresholdKmeans.m** - kombinuje prahování, morfologické operátory a fuzzy Kmeans shlukování pro segmentaci nádorové tkáně. Přijímá jako argumenty T1 nebo T2-vážený obraz a počet shluků pro Kmeans shlukovací analýzu. Vrací mapu rozdělení do shluků.
- **FastCMeans.m**, **FastFCMeans**, **FM2map.m**, **LU2label.m** - skripty přidru-

žené k fuzzy kmeans shlukování.

- **JaccAndDice.m** - výpočet Jaccard, Dicce koeficientů podobnosti, sensitivity a specificity mezi dvěma obrazy. Vrací pole indexů postupně: Jaccardův index, Diceho index, senzitivitu a specificitu. První položka v poli je průměrná odchylka, druhá položka je směrodatná odchylka měřené veličiny.

2. skripty pro Torch:

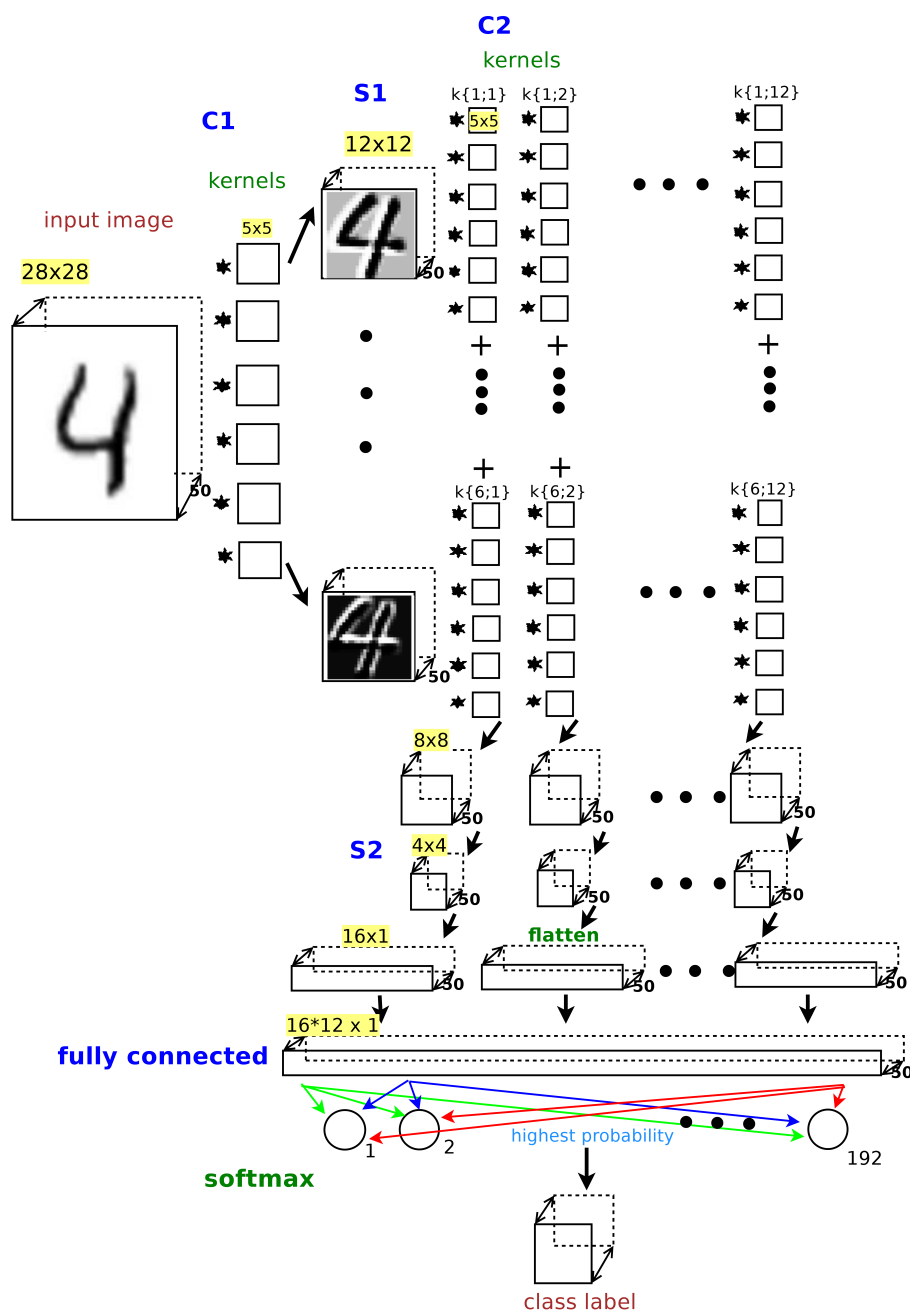
- **classifiers.lua** - definuje klasifikační Softmax a chybovou funkci pro T1 a T2 modely.
- **load_T1_data.lua** - načtení receptivních polí nádorů a pozadí z určené složky (defaultně *T1obrazy/T1_trenovaci_sada*). Vrací struktury obsahující trénovací a validační sadu náhodně přeházených receptivních polí obou tříd (nádor/pozadí).
- **load_T2_data.lua** - načtení receptivních polí edémů a pozadí z určené složky (defaultně *T2obrazy/T2_trenovaci_sada*). Vrací struktury obsahující trénovací a validační sadu náhodně přeházených receptivních polí obou tříd (edém/pozadí).
- **parse_T1_test_images.lua** - modul volaný ve skriptu **load_T1_data.lua** provede vytvoření receptivních polí pro třídu nádor/pozadí.
- **parse_T2_test_images.lua** - modul volaný ve skriptu **load_T2_data.lua** provede vytvoření receptivních polí pro třídu edém/pozadí.
- **parse_segm_data.lua** - funkce volaná ve skriptu **run_cnn.lua** provede vytvoření receptivních polí pro testovací data. Přijímá jako argument cestu k T1-váženým a T2-váženým testovacím snímkům a vrací *zparsovanou* strukturu receptivních polí pro obě modality.
- **run_cnn.lua** - hlavní modul, spouští trénování, validaci a testování CNN.
- **run_segmentation.lua** - provede segmentaci T1-vážených a T2-vážených obrazů pomocí struktur receptivních polí, navrácených funkcí **parse_segm_data.lua**. Výslednou segmentaci ukládá na disk.
- **sgd.lua** - upravená funkce knihovny Torch, patří přidat do cesty *torch/install/share/lua/<lua version>/optim*.
- **test_T1_data.lua** - implementuje funkci pro validaci T1-vážených obrazů, vrací ukazatel na funkci.
- **test_T2_data.lua** - implementuje funkci pro validaci T2-vážených obrazů, vrací ukazatel na funkci.
- **train_T1.lua** - implementuje funkci pro trénování T1-vážených obrazů, přijímá jako argument sadu T1-vážených obrazů rozdělených do receptivních polí, vrací ukazatel na funkci, umožňuje ukládat modely na disk.
- **train_T2.lua** - implementuje funkci pro trénování T2-vážených obrazů,

přijímá jako argument sadu T2-vážených obrazů rozdělených do receptivních polí, vrací ukazatel na funkci, umožňuje ukládat modely na disk.

- `T1_model.lua` - vytvoří strukturu modelu pro T1-vážené obrazy.
- `T2_model.lua` - vytvoří strukturu modelu pro T2-vážené obrazy.

Tyto informace jsou také uvedeny v souboru README.txt.

D SCHÉMA KONVOLUČNÍ SÍTĚ LENET



Obr. D.1: Konvoluční síť LeNet