VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVÁNÍ TEXTU S VYUŽITÍM INFORMACE O PISATELI

TEXT RECOGNITION ENHANCED BY WRITER IDENTITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MATĚJ TRNĚNÝ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. JAN KOHÚT**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Trněný Matěj**
Program: Informační technologie
Název: **Rozpoznávání textu s využitím informace o pisateli**
Text Recognition Enhanced by Writer Identity
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy neuronových sítí pro rozpoznávání textu.
2. Vytvořte si přehled o metodách, které umožňují učení neuronových sítí s využitím informace o pisateli.
3. Vyberte nejvhodnější metody a navrhněte metody vlastní.
4. Navrhněte experimenty.
5. Obstarejte si datovou sadu.
6. Implementujte metody a proveďte experimenty nad datovou sadou.
7. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
8. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Meng, Zhong, et al. "Speaker-invariant training via adversarial learning." *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.
- Wang, Shuai, et al. "On the Usage of Phonetic Information for Text-Independent Speaker Embedding Extraction." *INTERSPEECH*. 2019.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Kohút Jan, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 30. července 2021
Datum schválení: 30. října 2020

Abstrakt

Cílem práce je vytvořit neuronovou síť pro rozpoznání textu s využitím informace o pisateli. Pro tento účel byla vybrána metoda adversarial learning. Účinost teto metody byla ověřena experimentálně. Vytvořená síť by měla díky použité metodě adversarial learning dosahovat lepších výsledků na datech, která nejsou podobná datům obsaženým v trénovací sadě oproti stávající metodě single-task learning. Výsledná síť dosažená pomocí uvedené metody byla porovnána se současnou metodou rozpoznávání textu metodou single-task learning a multi-task learning. Síť implementující single-task learning dosahuje průměrné chyby při rozpoznávání znaku $7,995\%$, síť implementující multi-task learning dosahuje průměrné chyby $7,565\%$ v porovnání se sítí využívající adversarial learning, která dosahuje úspěšnosti $7,573\%$. V porovnání single-task learning dosahuje multi-task learning $5,38\%$ zlepšení a adversarial learning $5,28\%$.

Abstract

The objective of this theses was to implement a neural network for text recognition enhanced by writers identity. Adversarial learning method was selected for this purpose. Usefulness of this method was verified by experiments. This net should yield better results on data which are not similar to data contained in training data set. Accuracy of the resulting net was compared to method single-task learning and method multi-task learning. Net implementing single-task learning method has reached average character recognition error of $7,995\%$, net implementing multi-task learning method has reached average error of $7,565\%$ and net implementing adversarial learning method has reached average error of $7,573\%$. In comparison to the net implementing single-task learning multi-task learning has improvement of $5,38\%$ and adversarial learning has reached improvement of $5,28\%$.

Klíčová slova

adversarial learning, rozpoznávání textu, neuronové sítě, LSTM, CNN

Keywords

adversarial learning, text recognition, neural networks, LSTM, CNN

Citace

TRNĚNÝ, Matěj. *Rozpoznávání textu s využitím informace o pisateli.* Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Kohút

Rozpoznávání textu s využitím informace o pisateli

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Kohúta. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Matěj Trněný
29. července 2021

Poděkování

Rád bych tímto poděkoval vedoucímu své práce Ing. Janu Kohútovi za odborné vedení a pomoc při zpracování práce.

Obsah

Kapitola 1

Úvod

Cílem práce je vytvořit neuronovou síť pro rozpoznání textu s využitím informace o pisateli. K tomuto účelu byla vybrána metoda adversarial learning, jejíž účinnost má být experimentálně ověřena. Předpokladem a také důvodem tohoto zadání je domněnka, že díky použité metodě adversarial learning bude síť dosahovat lepších výsledků na datech, která nejsou podobná datům obsaženým v trénovací sadě.

Další částí práce je ověření výslednosti tohoto experimentu a jeho porovnání se současnými metodami rekurentní sítě a multi-task learningu.

Trénovací metoda adversarial learning je detailně popsána v podkapitole 3.4. Technologie, které tato síť využívá a jsou s ní spojeny, jsou popsány v kapitole 2. Zvolený přístup dle studií Speaker-Invariant Training Via Adversarial Learning[11] a On the Usage of Phonetic Information for Text-independent Speaker Embedding Extraction[18] přináší zlepšení v porovnání s ostatními metodami v případě, kdy síť dostává vstupy, které nebyly reprezentovány v trénovací datové sadě, což je v praxi, kdy používáme neuronové sítě velmi častý případ.

Tento přístup je také potřeba porovnat s ostatními přístupy, které jsou používány pro rozpoznávání textu, aby bylo možné ukázat, že adversariální sítě jsou schopné přinést snížení znakové chyby při správném použití. Jedná se o přístupy single-task a multi-task learning, které jsou podrobněji dále popsány v kapitole 2. Výsledky srovnání testování těchto přístupů a zvoleného přístupu jsou uvedeny v kapitole 5. Metoda adversarial learning na rozdíl od výše uvedených metod/přístupů využívá rozpoznávání domény, ze které daný řádek pochází. Dále pomocí gradient reversal layer (GRL) odnaučuje znakový klasifikátor rozpoznávat tyto domény, což vede k snížení závislosti sítě na doménách, na kterých byla síť trénována.

Zmíněné sítě byly trénovány na datové sadě historických dokumentů IMPACT, která poskytuje dostatečně velký počet různorodých dokumentů, které lze rozdělit do skupin pro trénování domén, na které se bude adversariální síť adaptovat.

Popis datové sady a její rozdělení na skupiny je popsáno v kapitole 4. Adaptace na domény způsobí, že se klasifikátor bude odnaučovat charakteristiky daných domén. Tato nově získaná vlastnost povede k tomu, že když síť dostane jako vstup jinou doménu, než která byla v trénovací sadě, bude schopna přesněji rozpoznávat znaky než síť, která byla trénována pouze jako single-task nebo multi-task learning. Metoda adversarial learning byla již vyzkoušena na rozpoznávání řeči ve studii, která je prezentována v článku [11]. Podle studie dosahuje model speaker-invariant training (SIT) 4,99 % zlepšení oproti konvenčnímu modelu speaker-independent (SI). V případě této práce je cílem vyzkoušet tuto metodu na trénování sítě pro rozpoznávání psaného textu.

Kapitola 2

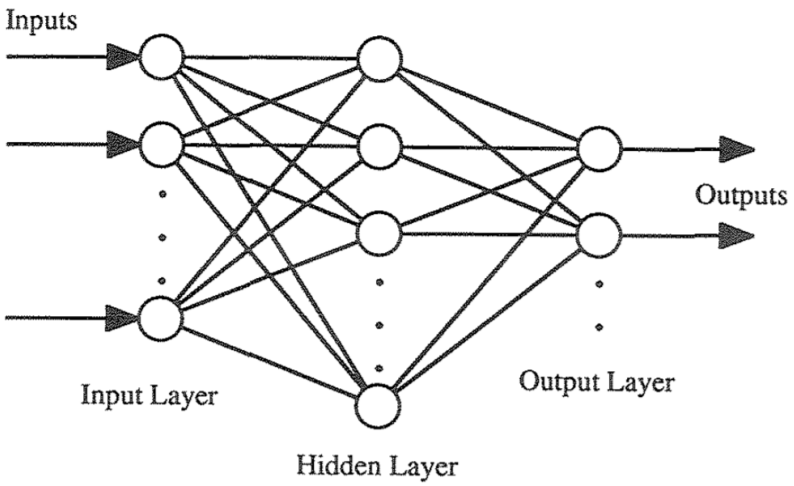
Neuronové sítě

Tato podkapitola se zabývá základními koncepty a prvky neuronových sítí. Umělé neuronové sítě (Artificial Neural Network - ANN) jsou inspirovány biologickými neurony. Podobně jako biologické neurony jsou ANN založeny na propojených umělých neuronech. Aktivace jednotlivých neuronů jsou spočítány jako suma výstupů nějaké nelineární funkce. Neurony a propoje mají váhy, které se postupně upravují v průběhu učení neuronové sítě. Váhy zesilují nebo oslabují sílu propojů.

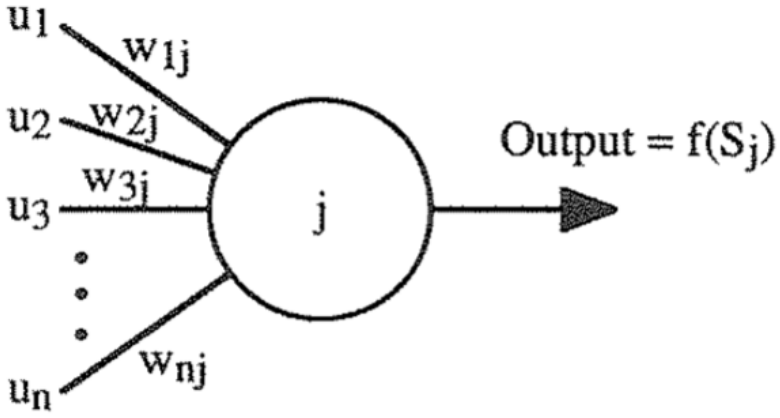
2.1 Umělá neuronová síť

ANN je pak uspořádaná do tří základních vrstev. Jedná se o následující vrstvy vstupní, skrytá a výstupní. Vrstvy se skládají z prvků, které názýváme neurony. Neurony vrstev, které následují za sebou, za sebou následující neurony jsou mezi sebou spojeny pomocí propojů. Takovému uspořádání sítě říkáme feed forward neural network (FFNN) na rozdíl od rekurentních sítí (podkapitola 2.3).

V těchto sítích jsou výstupy sítě dále používány i jako součást vstupu pro další iteraci, vstup je použit pouze pro získání daného výstupu. Vstupní vrstva neprovádí žádné výpočty, slouží pouze pro distribuci vstupních parametrů do sítě.



Obrázek 2.1: Plně propojená síť a její jednotlivé části, vstupní a výstupní vrstvu. Mezi nimi je pak jedna skrytá vrstva. Převzato z [4]



Obrázek 2.2: Umělý neuron. Převzato z [4]

Obrázek 2.2 znázorňuje, jak vypadá jeden neuron. Má n propojů, přičemž každý z těchto propojů má váhu w a vstup u. Tyto propoje potom dohromady dávají rovnici pro výpočet výstupu Output. Výstup neuronu j pak spočítáme jako sumu vah a vstupů následovně:

$$S_j = \sum_{i=1}^{n} w_{ij} u_j + w_{0j}, \tag{2.1}$$

kde j je index neuronu, w_{ij} je váha daného propoje i neuronu j, u_1 až u_n jsou vstupy neuronu, w_{0j} je bias daného neuronu j. Potom je na neuron aplikována aktivační funkce, typicky funkce ReLU nebo sigmoid. Aktivační funkce může být lineární, diskrétní nebo jiná spojitá distribuční funkce. Podmínkou pro trénování sítě pomocí algoritmu zpětné propagace je, aby tato funkce byla diferenciovatelná.

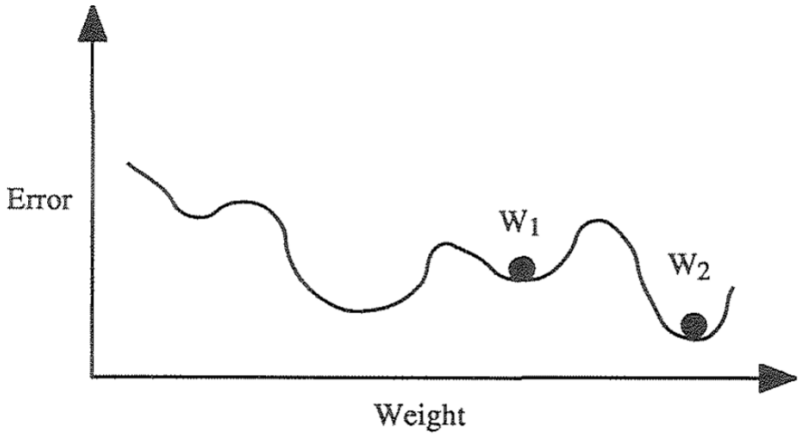
2.1.1 Trénování umělé neuronové sítě

Nezbytnou součástí neuronových sítí je proces, kterým se učí/trénují. ANN se učí pomocí postupného upravování váh a bias, které propojují neurony. Na začátku jsou tyto hodnoty nastaveny na náhodné malé hodnoty. Po nastavení počátečních hodnot je síť trénována pomocí série příkladů, které ukážou síti, jak by se měla pro dané vstupy chovat. Každý takovýto příklad je pár, který obsahuje vstupní hodnotu (případně hodnoty, záleží na počtu vstupních neuronů) a očekávaný výstup, který by měla síť poskytnout na výstupu. Tato data jsou potom síti opakovaně prezentována na vstup, dokud síť není schopná úspěšně spojit dané vstupy s jejich očekávanými výstupy, nebo dokud síť nedosahuje dostatečné úspěšnosti v tomto úkolu.

Algorismus, kterým je toto učení prováděno, se nazývá zpětná propagace. Základní myšlenka toho algoritmu je postup, při kterém je vybrán trénovací pár, který je aplikován na síť. Síť spočítá výstup pomocí současného vnitřního stavu a poskytnutého vstupu. Tento výstup je následně porovnán s očekávaným výstupem pro daný vstup z trénovacího páru. Váhy a bias každého neuronu jsou potom upraveny na základě derivace použité aktivační funkce (např. sigmoid), rozdílem mezi výstupem, který síť vyprodukovala a očekáváným výstupem, výstupy daných neuronů. Pomocí takových úprav je možné vylepšit správnost výstupů sítě, což považujeme za učení sítě.

Jak moc se váhy a bias mění v průběhu zpětné propagace také závisí na parametru learning rate, což je hodnota, kterou jsou všechny změny násobeny. Příliš vysoký learning

rate může způsobit oscilaci sítě mezi různými špatnými výsledky. Na druhou stranu příliš malý learning rate může vést k tomu, že síť se zastaví v lokálním minimu zabraňujícím síti použít větší krok, který by mohl vést k lepšímu stavu vah.



Obrázek 2.3: Funkce zobrazuje graf chyby a vah. Převzato z [4]

Na obrázku 2.3 vidíme příklad, kde v případě W_1 síť uvázne jako důsledek volby příliš malého learning rate. Síť už nebude schopná nalézt lepší minimum. Ideálně by síť dosáhla stavu W_2, ve kterém by potom zůstala. Jsou různé způsoby, jak se s tímto problémem vypořádat. Jedním z nich je úprava algoritmu zpětné propagace, kdy se bude learning rate dynamicky měnit. Další takový způsob by mohlo být začít trénování sítě od znovu s jinými počátečními hodnotami pro váhy a bias, které by náhodou mohly být blíže W_2 [4]. Úprava parametrů sítě W probíhá v iteracích podle následujícího vzorce:

$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W}, \tag{2.2}$$

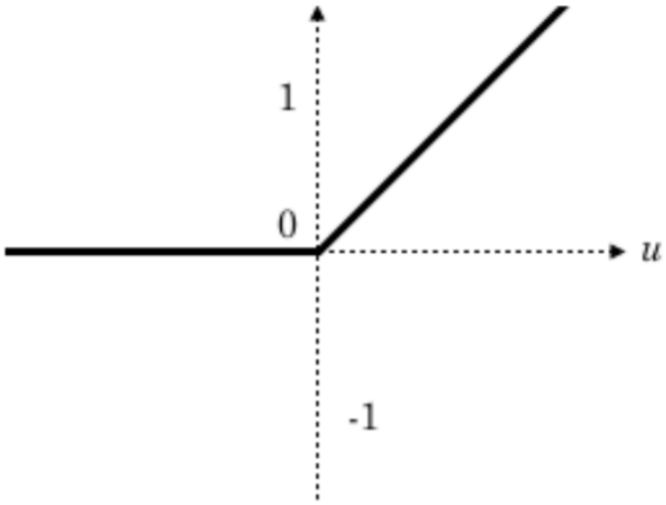
kde W je vektor váh, ϵ je konstanta learning rate (reguluje rychlost s jakou se síť učí), $E(W)$ je chybová fukce pro daný vektor vstupů a váh W [10].

2.1.2 ReLU

Rectified Linear Units (ReLU) je populární aktivační funce. ReLU je definována funkcí $R(x) = max(0, x)$, kde x je vstup neuronu. Výhodou ReLU je menší výpočetní náročnost oproti ostatním aktivačním funkcím (např. sigmoid, tanh), protože používá výpočetně méně náročné operace. Také vede k rychlejší konvergenci SGD (stochastic gradient descent) a předchází problémům, jako je vanishing gradient.

$$R(x) = \left\{ \begin{array}{ll} x & x > 0 \\ 0 & x \leq 0 \end{array} \right\} \tag{2.3}$$

$$R'(x) = \left\{ \begin{array}{ll} 1 & x > 0 \\ 0 & x < 0 \end{array} \right\} \tag{2.4}$$



Obrázek 2.4: Graf funkce ReLU. Převzato z [14]

Na obrázku 2.4 je zobrazen předpis funkce ReLU a její derivace. V podstatě ReLU všechny hodnoty, které jsou menší než 0 změní na 0 a větší čísla nechá projít nezměněná[1].

2.1.3 Sigmoid

Je nelineární aktivační funkce, která je často používána v rekurentních neuronových sítích. Předpis této funkce je:

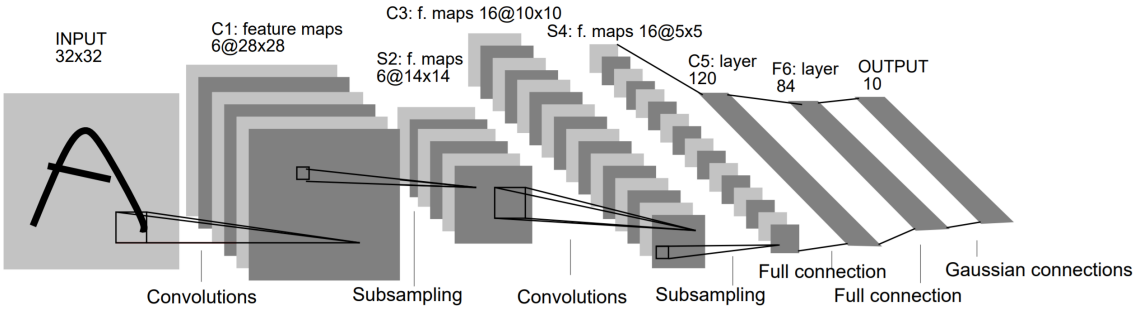
$$\sigma(u) = \frac{1}{1 + e^{-u}}, \tag{2.5}$$

sigmoid převádí všechny vstupy na hodnoty mezi 0 až 1 [19].

2.2 Konvoluční síť

Konvoluční sítě kombinují myšlenky, které se vypořádávají s problémy posunů, zvětšování nebo zmenšování měřítka a zkreslení vstupních dat. Toho konvoluční sítě dosahují za pomoci lokálních vnímavých polí, sdílené váhy a prostorového podvzorkování (sub-sampling).

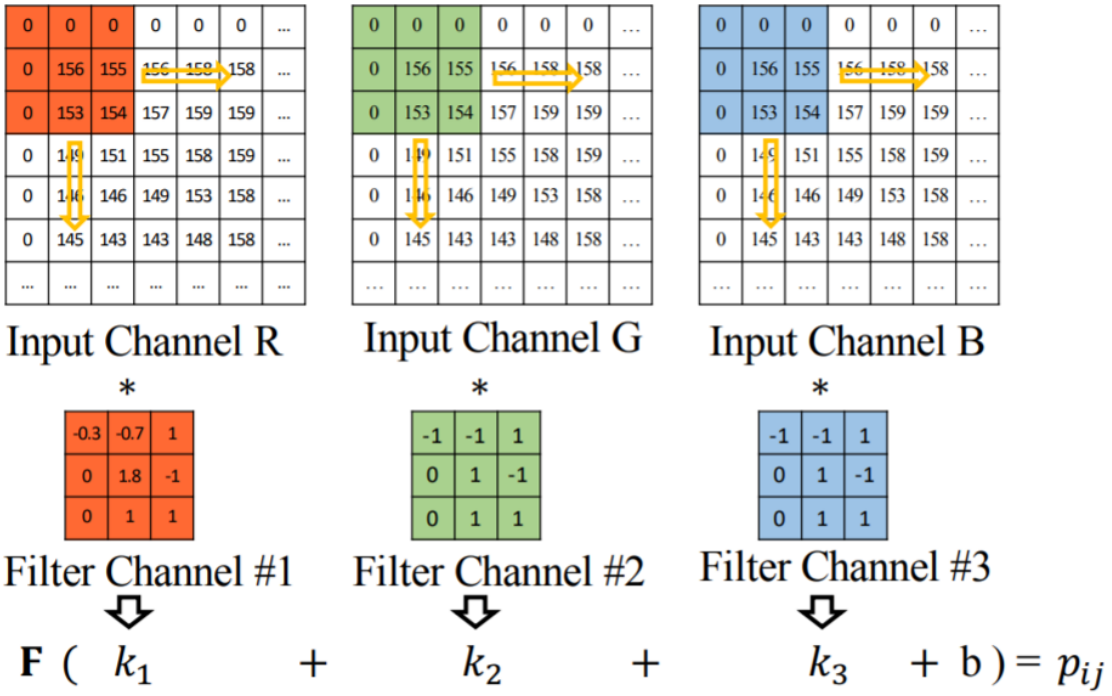
Neurony lokálních vnímavých polí se mohou naučit rozpoznávat elementární vizuální vlastnosti, jako jsou orientované hrany, koncové body a okraje (také podobné vlastnosti v jiných signálech jako např. rozpoznávání řeči). Tyto vlastnosti jsou potom zkombinovány v následujících za účelem rozpoznávání charakteristik vyššího řádu. Jednotky v dané vrstvě jsou organizované do rovin uvnitř, všechny jednotky v této rovině sdílí stejnou množinu vah. Množině výstupů těchto jednotek v dané rovině se říká feature map. Jednotky ve feature map jsou podmíněny k vykonávání stejné operace na různých částech obrázku. Potom úplná konvoluční vrstva je složená z několika feature mapami s různými váhovými vektory.



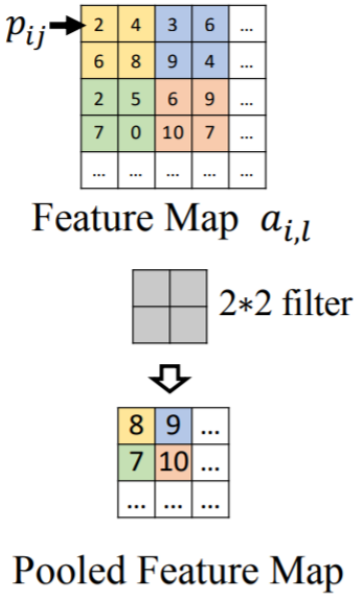
Obrázek 2.5: Ukázka konvoluční sítě. Převzato z [10]

Na obrázku 2.5 je zobrazena ukázka konvoluční sítě na rozpoznávání tvarů LeNet. Zde pro rozpoznávání číslic. Každá rovina představuje feature map (nebo také jednotky, jejichž váhy byly podmíněny, aby byly identické). Výstupy z konvolučních vrstev jsou potom zploštěny a předány plně propojeným vrstvám a na konci jsou předány výstupní vrstvě s klasifikátorem [10].

Feature mapa
Každý filtrový kanál počítá skalární součin v malém regionu ze vstupních dat (konvoluce) a vytváří tak specifický feature kanál (Obrázek 2.6). Počet filtrů na obrázku odpovídá počtu kanálů RGB, když síť očekává jako vstup obrázky.



Obrázek 2.6: Vizuální ukázka počítaní feature mapy. Převzato z [15]



Obrázek 2.7: Pooling vrstva zmenšuje prostorovou velikos feature mapy. Převzato z [15]

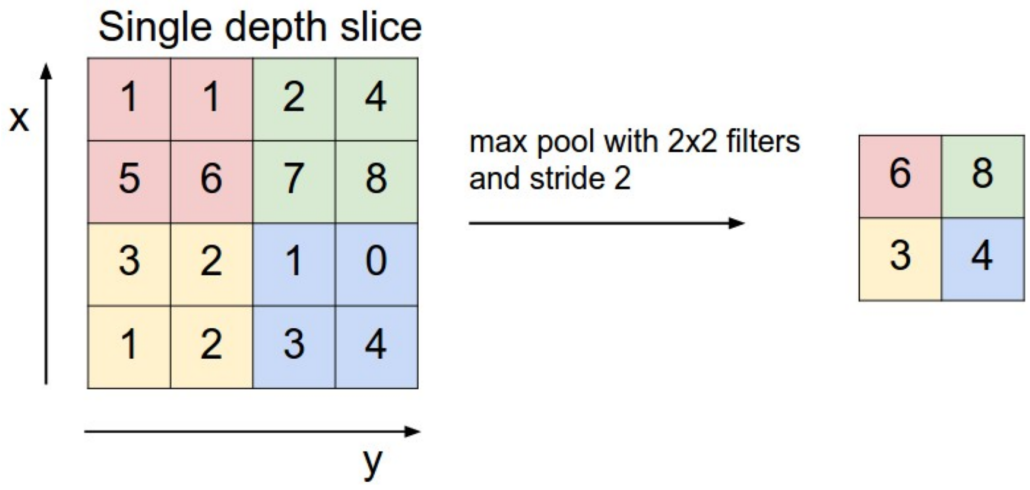
Po sečtení výsledků konvolucí je aplikována aktivační funkce F a je získána feature mapa, její prvek je na obrázku 2.7 označen p_{ij}. Proces počítání feature mapy $a_{i,l+1}$ lze popsat rovnicí:

$$a_{i,l+1} = F(\sum w_{i,l} a_{i,l} + b_{i,l}), \tag{2.6}$$

kde w a b reprezentují parametry sítě, váhy a bias resp., F je aktivační funkce (obvykle ReLU). Tento proces slouží pro získání charakteristik, které jsou pak užitečné pro klasifikaci [15].

2.2.1 Max pooling

Častým problémem konvolučních sítí je jejich prostorová náročnost dat, proto je výhodné používat pooling vrstvy, které slouží k prostorovému zmenšování objemu dat. Jedním z případů pooling vrstev je max pooling, která rozdělí vrstvy na kvadranty. Z každého kvadrantu pak vybere největší hodnotu (nejvíce aktivní neuron). Zmenšení objemu dat závisí na velikosti těchto kvadrantů (např. 2x2), tyto kvadranty se nazývají jádro max pooling vrstvy [3].

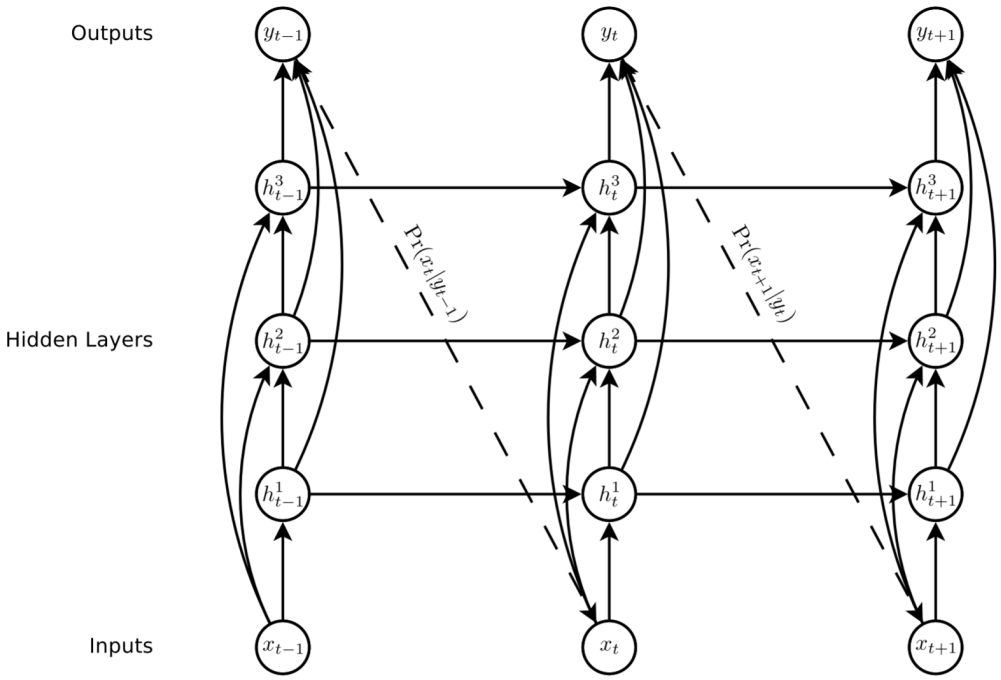


Obrázek 2.8: Ukázka max pool 2x2. Převzato z [2]

Na obrázku 2.8 je zobrazen max pooling s jádrem 2x2 a velikostí kroku 2, z každého kvadrantu je vybrána největší hodnota a ta je potom předána dál.

2.3 Rekurentní síť

Na rozdíl od feed forward neuronových sítí (FFNN) rekurentní sítě (RNN) používají cyklické propojení, která jsou užitečná pro modelování sekvencí. Rekurentní sítě cyklicky předávají výsledky aktivací z předchozích průchodů jako vstupy pro získaní dalších výstupů. Aktivace z předchozích kroků jsou uloženy ve vnitřním stavu sítě, které pak poskytují temporální kontextuální informaci, zatímco FFNN mají pevné kontextuální rámce.



Obrázek 2.9: Ukázka schématu rekuretní neuronové sítě. Převzato z [6]

Buňky v obrázku 2.9 reprezentují vrstvy sítě, plné šipky zobrazují váhované propoje sítě a přerušované šipky jsou predikce sítě. Daná rekurentní síť má vstup ve formě vektoru $x = (x_1, ..., x_T)$, který prochází váhovanými propoji přes N rekurentních skrytých vrstev $h^n = (h^n_1, ..., h^n_T)$ a pak vychází ven v podobě vektorové sekvence $y = (y_1, ..., y_T)$ Aktivace skryté vrstvy jsou spočítány v iteracích pomocí následujících rovnic:

$$h^1_t = \mathcal{H}(W_{ih^1}x_t + W_{h^1h^1}h^1_{t-1} + b^1_h) \tag{2.7}$$

$$h^1_t = \mathcal{H}(W_{ih^n}x_t + W_{h^{n-1}h^n}h^{n-1}_t + W_{h^nh^n}h^n_{t-1} + b^n_h) \tag{2.8}$$

Pro $t = 1$ až T a $n = 2$ až N. W představuje matice s váhami, b jsou vektory s bias, $\mathcal{H}$ je funkce skryté vrstvy (pro většinu rekurentních sítí bývá tato funkce aktivační funkce sigmoid).

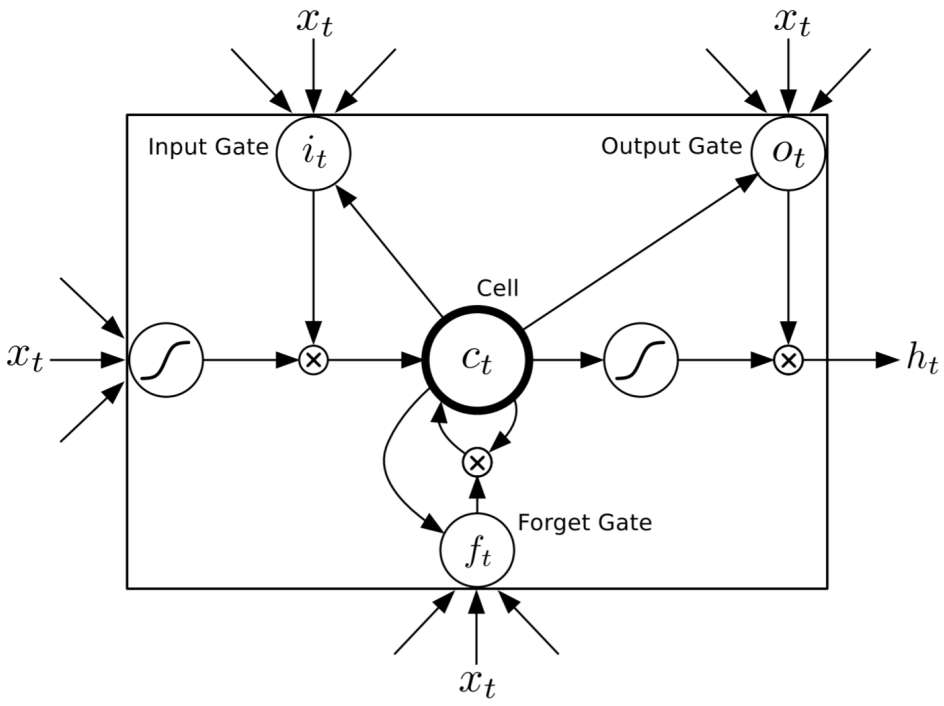
$$\hat{y}_t = b_y + \sum_{n=1}^{N} W_{h^n y}h^n_t \tag{2.9}$$

$$y_t = \mathcal{Y}(\hat{y}_t), \tag{2.10}$$

kde $\mathcal{Y}$ je funkce výstupní vrstvy. Celý síť potom definuje funkci, která je parametrizována váhovými maticemi a vstupem historií $x_{1:t}$ vstupů a výstupem vektorů y_t [6].

2.3.1 LSTM

Long Short-Term Memory (LSTM) je rekurentní síť, která byla navržena pro vyřešení problému vanishing a exploding gradient, který se objevoval u rekurentních sítí. Architektura LSTM využívá paměťové buňky na ukládání informací a vzdálených závislostí obsažených v datech.



Obrázek 2.10: Ukázka jedné buňky LSTM. Převzato z [6]

LSTM síť počítá mapování ze sekvence vstupů $x = (x_1, ..., x_T)$ na sekvenci výstupů $y = (y_1, ..., y_T)$ tak, že spočítá jednotlivé aktivace pomocí následujících rovnic iterativně od $t = 1$ do T:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{2.11}$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{2.12}$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \tag{2.13}$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{2.14}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.15}$$

$$h_t = o_t \odot \tanh(c_t), \tag{2.16}$$

kde W jsou váhy matic, b jsou váhové vektory, σ je funce sigmoid a i, f, o, c jsou resp. input gate, forget gate, output gate a cell activation vectors, které jsou všechny stejné velikosti jako cell output activation vector m. $\odot$ je skalární součin vektorů a g a h jsou cell input a cell output aktivačních funkcí, běžně tanh [6]. Předpis aktivační funkce pro LSTM tanh:

$$\tanh(u) = \frac{2}{1 + e^{-2u}} - 1, \tag{2.17}$$

která převádí vstupní hodnoty do rozmezí -1 až 1 [19].

2.4 Chybové funkce

V této kapitole jsou popsány chybové funkce, které byly použity pro trénování sítí v rámci této práce. V případě rozpoznávání textu je často používána vrstva CTC. Chybové funkce slouží pro srovnávání výstupů sítí a očekáváného správného výstupu (ground truth).

2.4.1 Connectionist temporal classification

Connectionist temporal classification [7] (CTC) je způsob trénování rekurentních neuronových sítí. CTC síť má softmax jako výstupní vrstvu s jednou jednotkou navíc, než je počet štítků v L. Aktivace prvních $|L|$ jednotek je interpretována jako pravděpodobnosti pozorování odpovídajících štítků v daný čas. Aktivace jednotky, která je navíc, je pravděpodobnost pozorovat „blank“ nebo také žádný štítek. Dohromady tyto výstupy definují pravděpodobnosti pro všechny možné sekvence s danou vstupní sekvencí. Celková pravděpodobnost daného jednoho štítku sekvence lze spočítat jako suma pravděpodobností různých rozložení.
Pro vstupní sekvenci x o délce T, definuje rekurentní neuronová síť s m vstupy, n výstupy a váhovým vektorem w jako spojitou mapu $\mathcal{N}_w$: $(\mathbb{R}^m)^T \mapsto \mathbb{R}^n)^T$. Nechť $y = \mathcal{N}_w(x)$ je sekvence výstupů, a denotován y_k^t aktivace výstupů jednotky k v čase t. Potom y_k^t je interpretováno jako pravděpodobnost pozorování štítku k v čase t, která definuje distribuci pro množinu sekvencí o délce T pro abecedu $L' = L \cup \{blank\}$:

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in {L'}^T. \tag{2.18}$$

Elementům L^T říkáme cesty a označují se π.
Dále je potřeba definovat jedna ku mnoho mapa $\mathcal{B} : {L'}^T \mapsto L^{\leq T}$, kde $L^{\leq T}$ je množina možných označení.
$\mathcal{B}$ definuje podmíněnou pravděpodobnost daných štítků $l \in L^{\leq T}$ jako sumu pravděpodobností všech cest, které korespondují:

$$p(l|x) = \sum_{\pi \in \mathcal{B}^{-1}(l)} p(\pi|x). \tag{2.19}$$

Výstupem klasifikátoru by mělo být nejvíce pravděpodobné oštítkování pro vstupní sekvenci:

$$h(x) = \arg\max_{l \in L^{\leq T}} p(\pi|x). \tag{2.20}$$

Tento proces hledání štítku se nazývá dekódování. Aby bylo dosaženo trénování největší pravděpodobnosti a trénování správných klasifikací je potřeba minimalizovat následující funkci:

$$O^{ML}(S, \mathcal{N}_w) = -\sum_{(x,y) \in S} \ln(p(z|x)). \tag{2.21}$$

2.4.2 Cross-entropy loss

Cross-entropy loss je hodnotící funkce, která slouží pro trénování neuronových sítí. Při trénování se používá pro měření přesnosti klasifikačního modelu, který dává výstupy jako

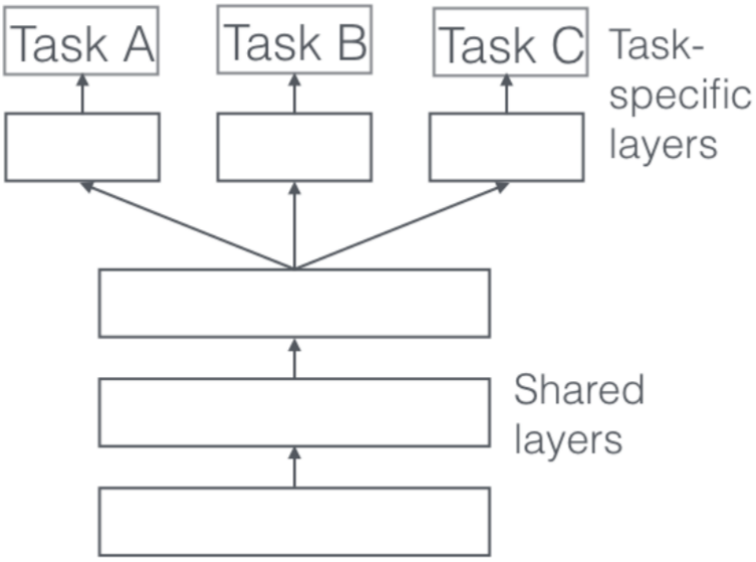
pravděpodobnosti v rozmezí od 0 do 1. Chyba se zvětšuje v závislosti na tom, jak moc se liší výstupy pravděpodobností modelu od správných výstupů (ground truth). Výpočet chyby pro cross-entropy je definován následovně:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}), \tag{2.22}$$

kde M je počet tříd, na které je síť trénována, ln je přirozený logaritmus, y je binární indikátor, jestli je třída c správná predikce pro observaci o, p, je predikce pravděpodobnosti, s jakou je o třídou c. Počet tříd, se kterými model pracuje, musí být větší než 2.

2.5 Multi-task learning

Multi-task learning (MTL) se ukázalo jako efektivní metoda v mnoha problémech počítačového vidění. Hloubkové neuronové sítě jsou vhodné pro využití této metody vzhledem k tomu, že charakteristiky které se učí z jednoho úkolu mohou být užitečné pro rozpoznávání dalšího.



Obrázek 2.11: Síť sdílející parametry pro multi-task learning. Převzato z [16]

Tradiční multi-task learning se snaží zlepšit generalizaci učení se souvisejících úkolů, tak že se síť učí všechny tyto úkoly zároveň. Předpokládá se, že síť má celkem T úkolů a trénovací data pro t-tou úlohu jsou zapsány jako (x_i^t, y_i^t), kde $t = \{1, ..., T\}$, $i = \{1, ..., N\}$, $x_i^t \in \mathbb{R}^d$ je vektor charakteristic a $y_i^t \in \mathbb{R}$ je vektor štítků. Cílem multi-task learning je potom minimalizovat:

$$\underset{\{w^t\}_{t=1}^T}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{i=1}^{N} \ell(y_i^t, f(x_i^t; w^t)) + \Phi(w^t), \tag{2.23}$$

kde $f(x^t; w^t)$ je funkce x^t a je parametrizována výhovým vektorem w^t, chybová funkce je označena ℓ, $\Phi(w^t)$ slouží k penalizaci za komplexitu váh. Cílem použití metody multi-task

learning v této práce je optimalizovat hlavní úlohu r (rozpoznávání textu), na rozdíl od konvenčního multi-task learning, které se snaží optimalizaci všech úloh. Tato síť se tedy snaží optimalizovat:

$$\underset{W^r,\{W^a\}_{a\in A}}{\operatorname{argmin}} \sum_{i=1}^{N} \ell^r(y_i^r, f(x_i; W^r)) + \sum_{i=1}^{N} \sum_{a\in A} \lambda^a \ell^a(y_i^a, f(x_i; W^a)), \tag{2.24}$$

kde λ^a značí důležitostostní koeficient pro chybu a-té úlohy, regularizační term je zanedbán z důvodu zjednodušení. Použití vzorce 2.24 má oproti 2.23, že je možné v průběhu optimalizace použít více různých chybových funkcí zároveň [20].

2.6 Adversarial multi-task learning

Účel adversarial multi-task learning[11] je optimalizovat hlavní úkol sítě a zaroveň minimaximalizovat vedlejší úkol sítě.
Síť využívající adversarial learning je v základu rozdělena na tři části $\theta = \{\theta_x, \theta_y, \theta_z\}$, θ_x, θ_y a θ_z značí parametry sítě, vstup a výstupní podsítě resp. Pro tyto parametry jsou definovány dvě chybové funkce $\mathcal{L}_y(\theta_x, \theta_y)$ a $\mathcal{L}_z(\theta_x, \theta_z)$. Pro tyto funkce je celková chyba spočítána pomocí rovnice:

$$\mathcal{L}_{total}(\theta_x, \theta_y, \theta_z) = \mathcal{L}_y(\theta_x, \theta_y) - \lambda \mathcal{L}_z(\theta_z, \theta_y) \tag{2.25}$$

Cílem této sítě je optimalizovat parametry následujícím způsobem:

$$(\hat{\theta}_x, \hat{\theta}_y) = \underset{\theta_x,\theta_y}{\operatorname{argmin}} \mathcal{L}_{total}(\theta_x, \theta_y, \hat{\theta}_z) \tag{2.26}$$

$$\hat{\theta}_z = \underset{\theta_z}{\operatorname{argmax}} \mathcal{L}_{total}(\hat{\theta}_x, \hat{\theta}_y, \theta_z), \tag{2.27}$$

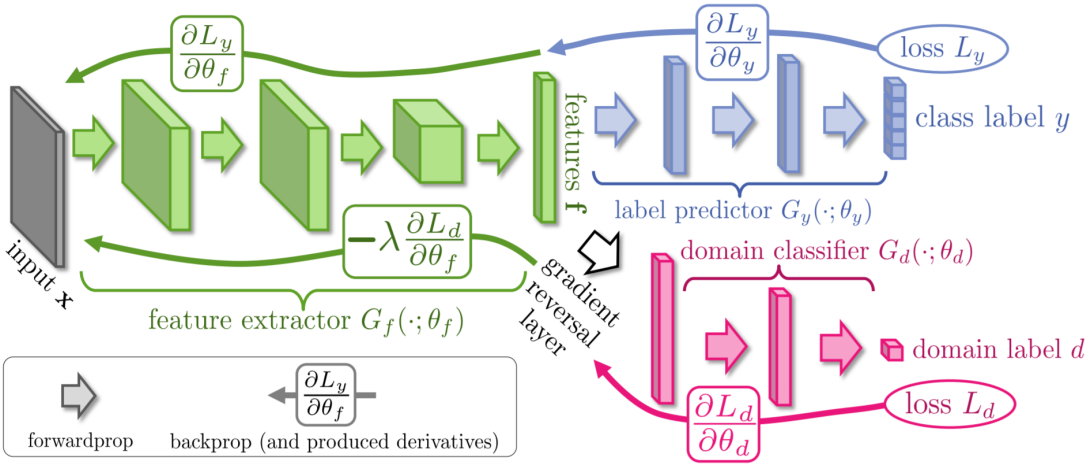
potom úprava parametrů sítě probíhá následovně:

$$\theta_y \longleftarrow \theta_y - \epsilon \frac{\partial \mathcal{L}_y}{\partial \theta_y}, \tag{2.28}$$

$$\theta_z \longleftarrow \theta_z - \epsilon \frac{\partial \mathcal{L}_z}{\partial \theta_z}, \tag{2.29}$$

$$\theta_x \longleftarrow \theta_x - \epsilon \left(\frac{\partial \mathcal{L}_y}{\partial \theta_x} - \lambda \frac{\partial \mathcal{L}_z}{\partial \theta_z} \right), \tag{2.30}$$

kde ϵ je learning rate sítě, λ je learning rate adversarial learning podsítě [17].

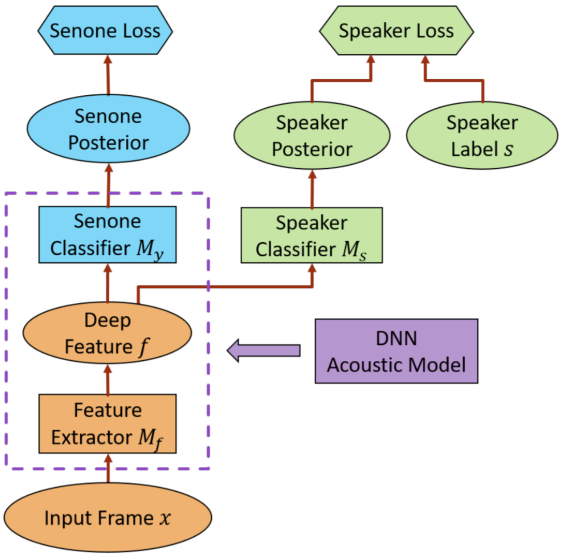


Obrázek 2.12: Schéma sítě pro adversarial multi-task learning. Převzato z [5]

Na obrázku 2.12 zelená část je exktraktor charakteristik vstupu. Modrou je vyznačena část, která rozpoznává jednotlivé třídy. Podobně jako u multi-task learning je k základní síti připojen klasifikátor domény (červená část obrázku), který rozpoznává, do které domény daný vstup patří. Šipky zobrazují průchod gradientu při zpětné propagaci, při průchodu gradient reversal layer násobí gradient $-\lambda$.

2.6.1 Gradient Reversal Layer

Vrstva Gradient Reversal Layer (GRL) je hlavní komponenta, která odlišuje multi-task learning od adversarial learning. GRL slouží pro otáčení znaménka gradientu, který prochází touto vrstvou při zpětné propagaci. Součástí této vrstvy je také learning rate λ, který reguluje vliv adversarialní vrsty na ostatní vrsty.



Obrázek 2.13: Ukázka adversariální sítě na rozpoznání řeči. Převzato z [11]

Kapitola 3

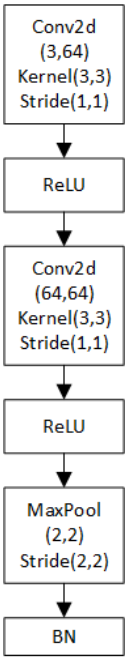
Architektura sítí

V této kapitole jsou popsány architektury sítí, které byly použity. První síť je základní síť pro rozpoznávání textu. Tato síť byla použita jako výchozí bod pro porovnání, jestli za použití adversarial learning došlo ke zlepšení. Dále je popsána v kapitole 3.4. Podobně byla použita i síť z kapitoly 3.1. MTL je principiálně velmi podobný adversarial multi-task learning, takže je také vhodné porovnat výsledky s tímto přístupem, proto byla implementována síť v podkapitole 3.3. Sítě byly implementovány pomocí knihovny PyTorch [13] a programy na zpracování datové sady byly poskytnuty od projektu PERO [1].

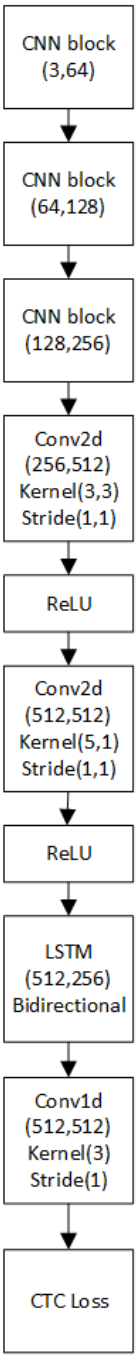
3.1 Základní síť

Základní síť je rekurentní síť, která používá single-task learning pro rozpoznávání znaků. Architektura této sítě (obrázek 3.2) se skládá z bloků, každý blok (obrázek 3.1) obsahuje 2 konvoluční vrstvy, každá tato vrstva je následována nelinearitou ReLU a celý blok má na konci batch normalizaci[8] a max pooling. Potom za konvolučními bloky je rekurentní blok, za kterým následuje další konvoluční vrstva, která už se mapuje na klasifikátor. Síť obsahuje 3 takovéto konvoluční bloky, jeden rekurentní a jednu výstupovou vrstvu a CTC vrstvu.

[1] pero.fit.vutbr.cz



Obrázek 3.1: Jeden blok základní sítě

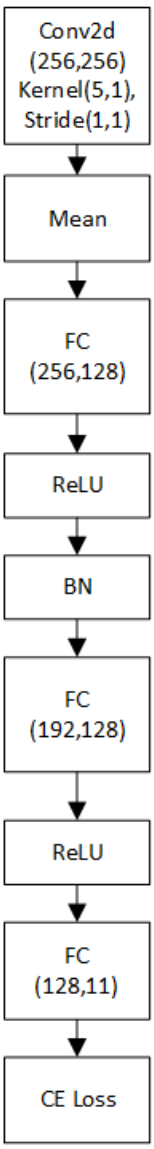


Obrázek 3.2: Schéma základní sítě

Na obrázcích 3.1 a 3.2 Conv2d a Conv1d označuje konvoluční síť s parametry filtrů uvedenými v závorkách. Ve stejném obdélníku je také uvedena velikost konvolučního jádra (Kernel) a velikost kroku (Stride). MaxPool označuje max pooling s velikostí rozměry a krokem (Stride) v závorkách. BN označuje batch normalizaci, ReLu je aktivační funkce, LSTM je obousměrná rekurentní síť LSTM s velikostí vstupu a skryté vrstvy v závorkách, CTC Loss je chybová funce. CNN block je část sítě zobrazená na 3.1 a v závorkách je uvedena velikost vstupu a výstupu daných bloků.

3.2 NetWriter

NetWriter (obrázek 3.3) se skládá z agregační vrstvy do které přicházejí charakteristiky z rekurentní sítě na rozpoznávání textu. Za agregační vrstvou jsou 3 plně propojené vrstvy. Každá vrstva je následována aktivační funkcí ReLU až na poslední výstupovou vrstvu, za kterou ReLU není. NetWriter používá chybovou funkci cross-entropy loss.

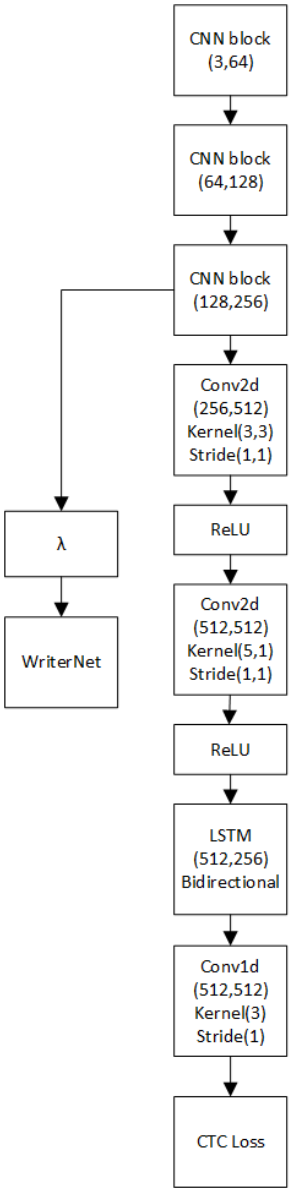


Obrázek 3.3: Schéma NetWriter

Na obrázku 3.3 FC označuje plně propojenou vrstvu s velikostí vstupu a výstupu uvedenou v závorkách, Mean označuje zprůměrování výstupu konvoluční sítě před tím než je předán plně propojené vrstvě, ostatní prvky jsou stejné jako v 3.1.

3.3 Multi-task learning síť

Základ sítě je společný se základní sítí, která byla popsána v podkapitole 3.1. K základní síti je připojena síť NetWriter pro rozpoznávání, ze které skupiny je daný řádek, který je právě zpracováván na vstupu. Síť také používá learning rate λ pro gradient přichazející z sítě NetWriter, kterým je regulován vliv této sítě na síť pro rozpoznávání textu. Schéma této sítě je znázorněno na obrázku 3.4.

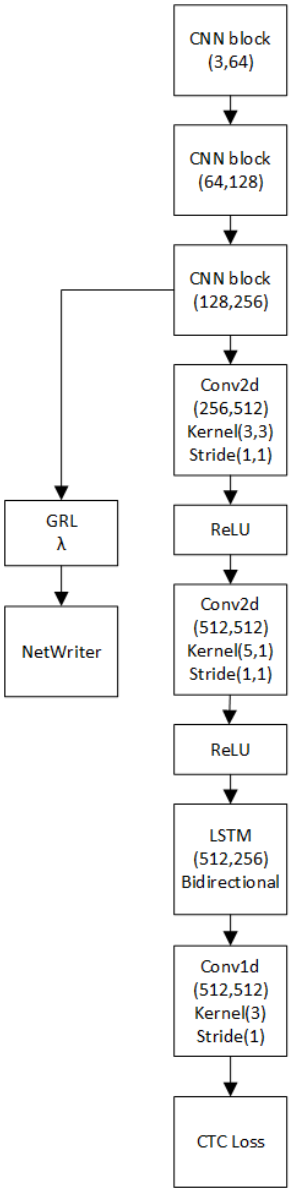


Obrázek 3.4: Schéma multi-task sítě

Na obrázku 3.5 λ označuje learning rate pro multi-task learning, ostatní prvky jsou stejné jako v 3.1.

3.4 Adversarial learning síť

Podobně jako síť z podkapitoly 3.3 využívá síť již definovanou v podkapitole 3.1, stejně tak používá síť NetWriter s tím rozdílem, že adversarial learning využívá vrstvu GRL. GRL je připojena před NetWriter, takže při zpětné propagaci je gradient, který prochází přes tuto vrstvu násoben záporným learning rate λ. Schéma této sítě je znázorněno na obrázku 3.5.



Obrázek 3.5: Schéma adversarial learning sítě

Na obrázku 3.5 GRL λ označuje learning rate pro adversarial learning, ostatní prvky jsou stejné jako v 3.1.

Kapitola 4

Datová sada

Tato kapitola se zabývá informacemi o datové sadě IMPACT[12] a informacemi o tom, jak byla rozdělena pro trénování implementovaných neuronových sítí. Při trénování neuronových sítí, je nezbytnou součástí trénovací sada, která zaručí, že síť bude mít při trénování k dispozici velký výběr pro adversarial learning. Rovněž je potřeba dostatečně rozmanitá datová sada, která bude rozdělena do skupin. Proto na trénování implementovaných sítí byla použita datová sada IMPACT, ze které je možné takovéto skupiny vytvořit (např. historická písma, která datová sada obsahuje, jsou vhodná pro vytváření takových skupin).

4.1 IMPACT

IMPACT je anotovaná datová sada, která je dostatečně velká a různorodá, aby bylo možné ji rozdělit do rozdílných skupin.

Obsahuje obrázky stránek z 10 různých knihoven. Každá knihovna poskytnula zhruba 50000 obrázků, celkem datová sada obsahuje přes 600000 obrázků historických dokumentů. Dokumenty jsou převážně z knih a novin, přesné rozdělení podle typu dokumentů je zobrazeno v tabulce 4.1.

Typ	**Počet dokumentů**
Stran knih	335640
Stran novin	142748
Stran právních dokumentů	80289
Stran deníků	19573
Stran časopisů	18957
Nezařazené strany	5423
Celkem stran	602630

Tabulka 4.1: Tabulka typů knih. Převzato z [12]

Téměř 80% těchto dokumentů je z devatenáctého století nebo začátku dvacátého, dalších 17% je ze sedmnáctého nebo osmnáctého století. Podrobnější rozdělení je v tabulce 4.2.

Století	Stran knih	Stran novin	Stran právních dokumentů	Stran časopisů	Stran ostatních dokumentů	Nezařazené strany	Celkem
15	338	0	0	0	1	0	339
16	17384	119	0	0	5	0	17508
17	58633	1119	0	0	8	280	60040
18	39139	2707	1297	0	144	132	43419
19	194682	64642	29821	10.216	13556	4061	316978
20	23038	73745	49171	9357	5243	234	160788
?	2426	416	0	0	0	716	3558

Tabulka 4.2: Rozdělení datové sady podle století. Převzato z [12]

Datová sada má dostatečnou rozmanitost díky tomu, že obsahuje různé jazyky. Je potvrzeno, že obsahuje 18 různých jazyků.

bulharština	němčina	polština
katalánština	řečtina	portugalština
čeština	hebrejština	ruština
nizozemština	latina	slovinština
angličtina	norština	španělština
francouzština	staroslověnština	

Tabulka 4.3: Rozdělení datové sady podle století. Převzato z [12]

Tabulka 4.3 obsahuje výčet jazyků, které jsou v datové sadě obsaženy. Další vhodnou vlastností z hlediska rozmanitosti jsou druhy písma.

Bohoričica	Hebrew
Cyrillic	Latin
French	Latin/Gothic
Gaj Old	Cyrillic
Greek	Serif

Tabulka 4.4: Druhy písma, které IMPACT obsahuje. Převzato z [12]

Tabulka 4.4 obsahuje výčet druhů písma, které jsou obsaženy v datové sadě.

4.2 Rozdělení pro testování

Na trénování sítí byla datová sada rozdělena na celkem 16 skupin. Tyto skupiny jsou vzhledově navzájem dostatečně odlišné, aby bylo možné je použít pro trénování klasifikátoru adversariální sítě. Z těchto skupin bylo použito 10 na trénování jednotlivých sítí, následně zbývajících 6 skupin bylo použito pro testování.

Id	Název skupiny	Počet řádků	Vlastnosti
trénovací skupiny			
1	Cluster1	24038	slovinština, noviny, normální písmo
2	Cluster2	25119	polština, kniha, normální písmo
3	Cluster3	30000	nizozemština, právní dokument, normální písmo
4	Cluster4	17060	bulharština, noviny, cyrilice
5	Cluster5	30000	španělština, kniha, normální písmo
6	Cluster6	14666	francouzština, kniha, normální písmo
7	Cluster7	11037	nizozemština, noviny, historické písmo
8	Cluster8	9625	slovinština, kniha, normální písmo
9	Cluster9	11249	italština, kniha, normální písmo
10	Cluster10	10463	angličtina, kniha, normální písmo
testovací skupiny			
-	Cluster11	14373	angličtina, noviny, normální písmo
-	Cluster12	10918	polština, kniha, historické písmo
-	Cluster13	2273	čeština, kniha, normální písmo
-	Cluster14	6098	bulharština, kniha, cyrilice
-	Cluster15	2645	nizozemština, kniha, historické písmo
-	Cluster16	26674	bulharština, noviny, cyrilice

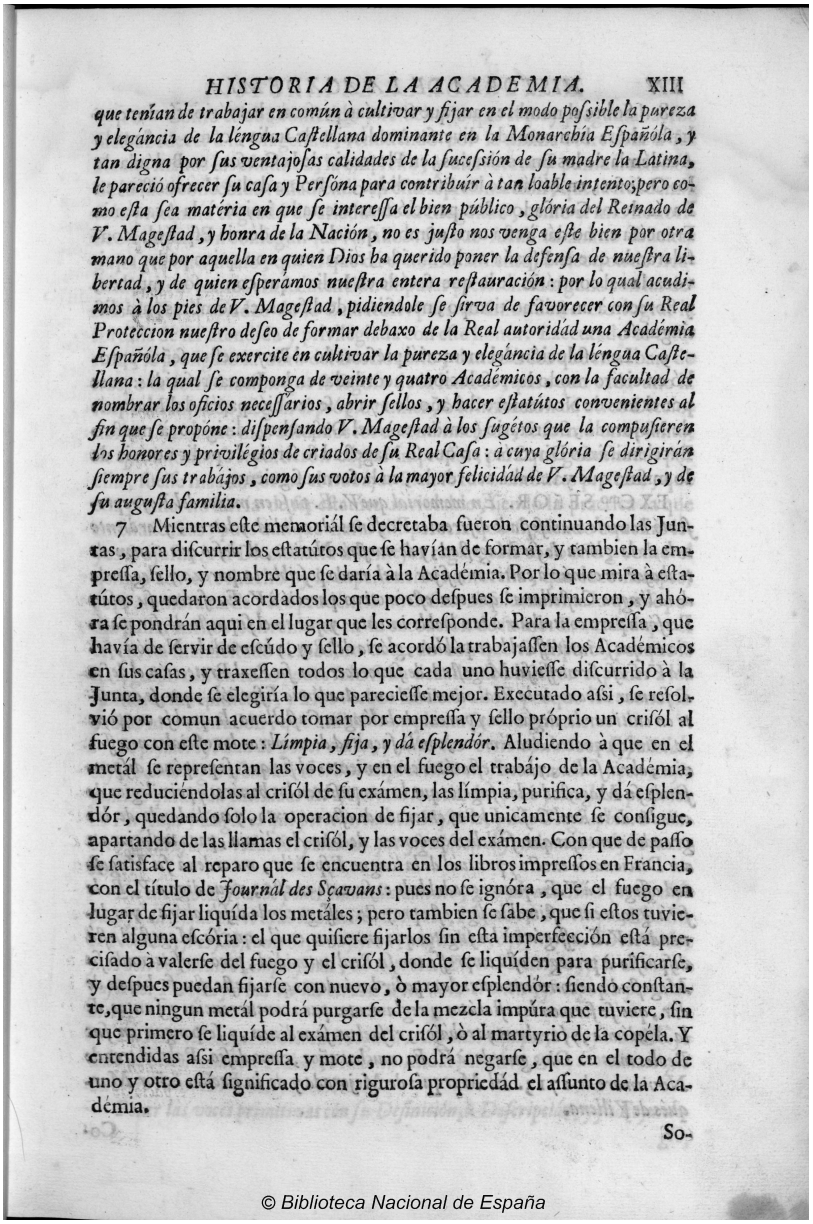
Tabulka 4.5: Použité části datové sady na jednotlivé skupiny

Každá z trénovacích skupin má alespoň 9000 řádků přepisů s tím, že bylo rezervování 10% řádků na testování i těchto skupin. Testovací skupiny nemají identifikátor, protože nebyly použity při trénování sítí. Dále byly tyto skupiny vybrány podle jazyků, ve kterých jsou napsány, typů písma (cyrilice, latina, atd.) a typu dokumentu (knihy, noviny, atd.).

Pro co největší podobnost řádků v rámci jedné skupiny, jsou řádky některých těchto skupin přebírány pouze z jednoho dokumentu. To známená, že jedna skupina odpovídá jednomu dokumentu.

Ostatní skupiny jsou založeny na spojení řádků více dokumentů, které jsou vzhledově dostatečně podobné, aby bylo možné je použít jako jednu trénovací skupinu.

Cílem tohoto rozdělení je, aby bylo možné trénovat klasifikátor sítě NetWriter na rozpoznávání jednotlivých skupin (identifikátorů skupin) a potom při testování, aby byly k dispozici skupiny, které nebyly použity pro trénování sítě. Účel těchto skupin, které byly vybrány pouze pro testování bylo to, že na nich potom bude ukázáno zlepšení oproti ostatním přístupům.



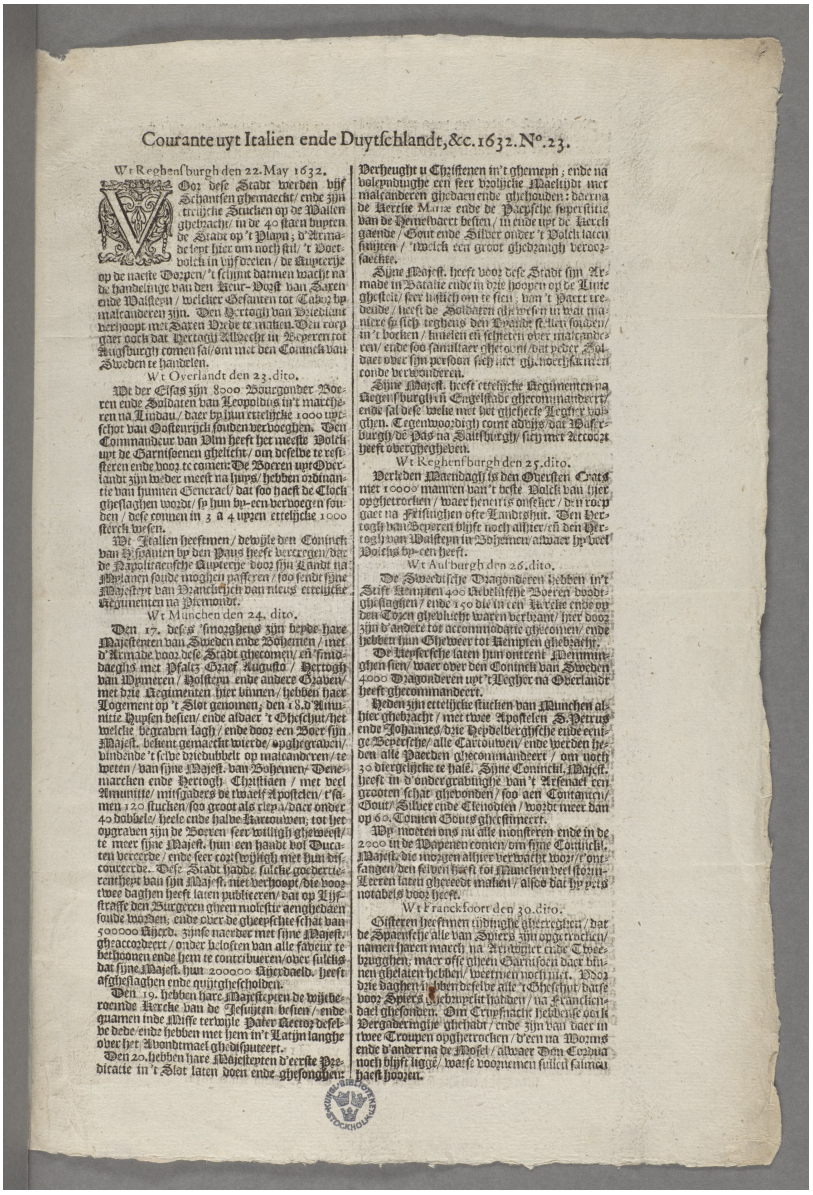
HISTORIA DE LA ACADEMIA. XIII

que tenían de trabajar en común à cultivar y fijar en el modo posible la pureza y elegáncia de la léngua Castellana dominante en la Monarchía Española, y tan digna por sus ventajosas calidades de la sucesión de su madre la Latina, le pareció ofrecer su casa y Persóna para contribuír à tan loable intento; pero como esta sea matéria en que se interessa el bien público, glória del Reinado de V. Magestad, y honra de la Nación, no es justo nos venga este bien por otra mano que por aquella en quien Dios ha querido poner la defensa de nuestra libertad, y de quien esperámos nuestra entera restauración: por lo qual acudimos à los pies de V. Magestad, pidiendole se sirva de favorecer con su Real Proteccion nuestro deseo de formar debaxo de la Real autoridád una Académia Española, que se exercite en cultivar la pureza y elegáncia de la léngua Castellana: la qual se componga de veinte y quatro Académicos, con la facultad de nombrar los oficios necessarios, abrir sellos, y hacer estatútos convenientes al fin que se propóne: dispensando V. Magestad à los sugétos que la compusieren los honores y privilégios de criados de su Real Casa: à cuya glória se dirigirán siempre sus trabájos, como sus votos à la mayor felicidád de V. Magestad, y de su augusta familia.

7 Mientras este memoriál se decretaba fueron continuando las Juntas, para discurrir los estatútos que se havían de formar, y tambien la empressa, sello, y nombre que se daría à la Académia. Por lo que mira à estatútos, quedaron acordados los que poco despues se imprimieron, y ahóra se pondrán aqui en el lugar que les corresponde. Para la empressa, que havía de servir de escúdo y sello, se acordó la trabajassen los Académicos en sus casas, y traxessen todos lo que cada uno huviesse discurrido à la Junta, donde se elegiría lo que pareciesse mejor. Executado assi, se resolvió por comun acuerdo tomar por empressa y sello próprio un crisól al fuego con este mote: *Límpia, fija, y dá esplendór.* Aludiendo à que en el metál se representan las voces, y en el fuego el trabájo de la Académia, que reduciéndolas al crisól de su exámen, las límpia, purifica, y dá esplendór, quedando solo la operacion de fijar, que unicamente se consigue, apartando de las llamas el crisól, y las voces del exámen. Con que de passo se satisface al reparo que se encuentra en los libros impressos en Francia, con el título de *Fournál des Sçavans*: pues no se ignóra, que el fuego en lugar de fijar liquída los metáles; pero tambien se sabe, que si estos tuvieren alguna escória: el que quisiere fijarlos sin esta imperfeccion está precisado à valerse del fuego y el crisól, donde se liquíden para purificarse, y despues puedan fijarse con nuevo, ò mayor esplendór: siendo constante, que ningun metál podrá purgarse de la mezcla impúra que tuviere, sin que primero se liquíde al exámen del crisól, ò al martyrio de la copéla. Y entendidas assi empressa y mote, no podrá negarse, que en el todo de uno y otro está significado con riguróla propriedád el assunto de la Académia.

So-

Obrázek 4.1: Ukázka stránky z datové sady



Courante uyt Italien ende Duytschlandt, &c. 1632. No. 23.

Wt Reghensburgh den 22. May 1632.

Van Godt deser Stadt werden dese Schansen ghemaeckt, ende zijn dryerley Stucken op de Wallen ghebracht, in de 40 staen buyten de Stadt op 't Platy; d'Armade leyt hier om noch stil, 't Boerevolck is in vyf oorden, de Kuyerije op de naeste Dorpen, 't schijnt datmen wacht na de handelinge van den Keur-Vorst van Saxen ende Wallestyn, welcker Gesanten tot Tabor by malcanderen zijn. Den Hertogh van Wurtenberch verhoopt met Saxen Vrede te maken. Den Paep gaet nae Landt dat Hertogh Albrecht in Beyeren tot Augspurgh comen sal, om met den Coninck van Sweden te handelen.

Wt Overlandt den 23. dito.

Wt der Elsas zijn 8000 Bourgonder Boeren ende Soldaten van Leopoldus in't marcheren na Lindau, daer by hun eenighe 1000 uytschot van Oostenrijck souden verhoogen. Den Commandeur van Ulm heeft het meeste Volck uyt de Garnisoenen ghelicht, om deselve te versterken ende voort te comen. De Boeren uyt Overlandt zijn weder meest na huys, hebben ordinantie van hunnen Generael, dat soo haest de Clock ghesleghen wordt, sy hun by-een verhoegen souden, dese comen in 3 à 4 uyren eenighe 1000 sterck wesen.

Wt Italien heerstmen, dewijle den Coninck van Hispanien by den Paus heeft vercregen, dat de Napolitaensche Kuyerije voor syn Landt na Italyanen soude mogen passeren, soo sendt syne Majesteyt van Spanjen by den nieuw eenighe Regimenten na Piemondt.

Wt München den 24. dito.

Den 17. des s'morghens zijn beyde hare Majesteiten van Sweden ende Bohemen, met d'Armade voor dese Stadt ghecomen, en s'middaghs met Pfaltz Graef Augusto, Hertogh van Wyneren, Holsteyn ende andere Graven, met drie Regimenten hier binnen, hebben haer Logement op 't Slot genomen, den 18. d'Annuntie Dupsen besien, ende alhaer 't Ghestut, het welcke begraven lagh, ende voor een Boer syn Majest. beleent gemaeckt wierde, opghegraven, bindende 't selve driedubbelt op malcanderen, te weten, van syne Majest. van Bohemen, Denemarcken ende Hertogh Christiaen, met veel Ammunitie, musquaders de twaelf Apostelen, t'samen 120 stucken, soo groot als clep, daer onder 40 dobbele, heele ende halve Karrouwen, tot het opgraven zijn de Boeren seer vrolijck ghebleven, te meer syne Majest. hun een handt vol Ducaten verceerde, ende seer coortelijck met hun discourreerde. Dese Stadt hadde sulcke goederen thept van syn Majest. niet verhoopt, die voor twee daghen heeft laten publiceren, dat op Lijfstraffe den Burgeren ghen molestie aenghedaen soude worden, ende voor de ghevonden schat van 500000 Rijcxd. zijnse naerder met syne Majest. ghecaccordeert, onder beloften van alle faveur te bethoonen ende hem te contribueren, voor sulcks dat syne Majest. hun 200000 Rijxdaelt. heeft afghehaghen ende quijtgheschouden.

Den 19. hebben hare Majesteyten de wijtberoemde Kercke van de Jesuijten besien, ende quamen inde Misse terwijle Pater Keetop deselve dede, ende hebben met hem in't Latijn langhe over het Avondtmael ghedisputeert.

Den 20. hebben hare Majesteyten d'eerste Predicatie in't Slot laten doen ende ghesonghen: Verheught u Christenen in't ghemeyn, ende na volcpuchte een seer vrolijcke Maeltyt met malcanderen ghehouden ende ghehouden: daerna de Heerlijcke Missa ende de Papsche superstitie van de Heemelvaert besien, in ende uyt de Kercke gaende, Gout ende Silver onder 't Volck laten smijten, twelck een groot ghedrangh veroorsaeckte.

Sijne Majest. heeft voor dese Stadt syn Armade in Battalie ende in drie hoopen op de Linie ghestelt, seer luystelijck om te sien, dan 't waert treedende, heeft de Soldaten ghelesen in welk manieren sy slesh tegheus den Vyandt sullen houden, in 't boecken, knuselen en schieten ober malcanderen, ende soo sammitaer ghestoort, dat yeder Soldaet ober syn persoon soch niet ghevroetsaken en conde verwonderen.

Sijne Majest. heeft eenighe Regimenten na Augenspurgh en Engelstadt ghecommandeert, ende sal dese welke met het gheheele Legher volghen. Tegenwoordigh comt advijs dat Wasserburgh de Pas na Saltzburgh sterck met Accoort heeft obergheghehen.

Wt Reghensburgh den 25. dito.

Verleden Maendaghs den Obersten Cratz met 10000 mannen van't beste Volck dan hier, opgetrocken, waer henen dit onseker, den roep gaet na Feistnighen ofte Landtshut. Den Hertogh van Beyeren blijft noch altijt, en den Hertogh van Wallestyn in Bohemen, alwaer hy veel Volcks by-een heeft.

Wt Aulburgh den 26. dito.

De Swaedtsche Vranzonderen hebben in't Reiss 14 persoonen 400 Oostenrijcke Boeren doodt ghestaghen, ende 140 die in een Kercke ende op den Tozen ghebleven waren verbrant, het voor zijn d'andere tot accommodatie ghecomen, ende hebben hun Ghewer tot Kempten ghebracht.

De Keyserselse laten hun ontrent Memmingen sien, waer over den Coninck van Sweden 4000 Vagonderen uyt't Legher na Oberlandt heeft ghecommandeert.

Heden zijn eenighe stucken van München alhier ghebracht, met twee Apostelen S. Petrus ende Johannes, drie Heydelbergsche ende eenige Keyserlijcke, alle Carrouwen, ende werden heden alle daeghen ghecommandeert, om voort 30 diergelijcke te halen. Sijne Coninckl. Majest. heeft in d'onder grafnighe van 't Arsenael een grooten schat ghevonden, soo aen Contanten, Gout, Silver ende Clenodien, wordt meer dan op 60. Tonnen Gouts ghestimeert.

Wy moeten ons nu alle monsteren ende in de 2000 tot de Wapenen comen, om syne Coninckl. Majest. die nu zijn alhier verwacht wort, t'ontfangen, den selven heeft tot Munchen drie Vorsten Heeren laten ghereedt maken, alsoo dat hy zijn notabels voor heeft.

Wt Franckfoort den 30. dito.

Gisteren heerstmen tijdinghe ghecregheen, dat de Spaensche alle van Spiers zijn opgetrocken, namen haren marsch na Arentuyer ende Tweebruyghen, maer offe ghen Garnisoen daer binnen ghelaten hebben, weetmen noch niet. Voor dese daghen hebben deselve alle 10 Oberste hun voor Spiers ghehouden, na Franckendael ghesonden. Den Cruysmacht schijnt ons te Vergaderinghe ghehadt, ende zijn daer in twee Troupen opgetrocken, d'een na Worms ende d'ander na de Mosel, alwaer Den Cardinal noch blijft liggen, waerse voornemen sullen saemen haest te voegen.

Obrázek 4.2: Ukázka stránky z datové sady

Na obrázcích 4.1 a 4.2 jsou ukázky stránek z daných skupin, které byly vybrány pro trénování sítí. Z obrázků je možné vidět, že se vizuálně navzájem liší. Jedná se o různé typy dokumentů, jazyků a typů písma.

Obrázek 4.3: Ukázka výstupu sítě

Na obrázku 4.3 je vidět jak vypadá výstup sítě (popsané v podkapitole 3.1) na testovacích datech po 5000 iteracích trénování. Obrázek je rozdělen do tří sloupů, první sloupec jsou řádky z dokumentů, které síť dostala jako vstup, prostřední sloupec obsahuje výstup z sítě (červenou jsou označeny chyby), na pravé straně jsou přepisy k daným řádkům.

Kapitola 5

Experimenty

V této kapitole je popsán postup experimentální ověření metody adversarial learning.

Každý experiment byl navržen na trénování jednoho typu sítě, přičemž pro některé metody trénování bylo natrénováno více sítí s různými hodnotami learning rate lambda (multi-task learning, adversarial learning). Pro základní síť je použita hodnota 0, protože pro tuto síť je použitá stejná struktura sítě, ale je potřeba, aby gradient, který přichází z NetWriter byl nulový. Díky své nulové hodnotě nebude mít žádný vliv na trénování hlavní sítě. Celkem byly provedeny 3 experimenty.

V rámci prvního experimentu 5.1 byla natrénována základní síť, která je potřeba pro stanovení přesnosti sítě bez použití trénovacích metod multi-task nebo adversarial learning.

V druhém experimentu byly natrénovány dvě sítě pro různé hodnoty lambda, které slouží ke srovnání adversarial learning s podobnou metodou.

V posledním experimentu byly natrénovány dvě sítě metodou adversarial learning, tyto dvě sítě použijí stejnou hodnotu learning rate lamba jako v druhém experimentu, ale zápornou.

Kromě části obsahující trénování, má každý experiment také testovací část. V ní byl proveden test sítí na vybraných testovacích skupinách, které se liší od trénovací skupiny.

Na začátku experimentů byl learning rate trénovacího scriptu nastaven na 0,0003. Poté byla každá ze sítí trénována 35 epoch (35000 iterací), přičemž výsledky každých 5000 iterací byly použity pro srovnání úspěšnosti sítí. Pro optimalizaci byl použit optimalizační algoritmus Adam[9]. Velikost dávky byla nastavena na 32 řádků. Trénování sítí bylo provedeno na grafické kartě Nvidia GeForce RTX 2080.

5.1 Experiment 1

V rámci prvního experimentu byla trénována základní rekurentní síť, takže lambda pro gradient reversal layer byla nastaven na 0. V případě, že je lambda nastavená na nula, tak veškerý gradient, který prochází přes GRL, bude nulován. Výsledkem je, že NetWriter nebude mít žádný vliv na trénování sítě na rozpoznávání znaků. Výsledky průběžného testování chyby sítě na trénovací sadě jsou uvedeny v tabulce 5.1 a na testování na skupinách Cluster1 až Cluster10 je v tabulce 5.2.

Číslo iterace	Chyba znaku
5000	3, 38%
10000	1, 60%
15000	1, 34%
20000	1, 09%
25000	0, 95%
30000	1, 12%
35000	0, 99%

Tabulka 5.1: Chyba sítě pro každých 5000 iterací při trénování

Název skupiny	Chyba znaku
Cluster1	5, 05%
Cluster2	4, 09%
Cluster3	4, 92%
Cluster4	5, 39%
Cluster5	4, 80%
Cluster6	4, 11%
Cluster7	5, 46%
Cluster8	4, 45%
Cluster9	6, 94%
Cluster10	7, 95%

Tabulka 5.2: Chyba základní sítě na testovacích skupinách z Cluster1 až Cluster10

5.2 Experiment 2

V rámci druhého exprimentu byla trénována síť metodou multi-task learning, proto byla lambda nastavena 1, protože je potřeba kladná hodnota a síť pak byla trénována 35000 iterací. V další části experimentu pak byla lambda nastavena na 2 a síť byla natrénována metodou multi-task learning. Tato síť byla také trénována 35000 iterací. Výsledky průběžného testování chyby sítě na trénovací sadě jsou uvedeny v tabulce 5.3 a na testování na skupinách Cluster1 až Cluster10 je v tabulce 5.4, sítě jsou od sebe v tabulkách odlišeny hodnotou λ.

Číslo iterace	Chyba znaku	Chyba domény	Chyba znaku	Chyba domény
λ	1		2	
5000	3, 08%	3, 22%	3, 38%	8, 13%
10000	1, 63%	2, 28%	1, 58%	1, 09%
15000	1, 31%	1, 39%	1, 24%	0, 84%
20000	1, 09%	0, 50%	1, 18%	0, 64%
25000	2, 38%	2, 38%	0, 93%	0, 74%
30000	0, 95%	1, 39%	0, 94%	0, 30%
35000	0, 93%	0, 40%	0, 97%	0, 84%

Tabulka 5.3: Chyba sití pro každých 5000 iterací při trénování

Číslo iterace	**Chyba znaku**	
λ	1	2
Cluster1	$5,00\%$	$5,02\%$
Cluster2	$4,08\%$	$4,09\%$
Cluster3	$4,91\%$	$4,89\%$
Cluster4	$5,36\%$	$5,37\%$
Cluster5	$4,81\%$	$4,81\%$
Cluster6	$4,07\%$	$4,12\%$
Cluster7	$5,50\%$	$5,49\%$
Cluster8	$4,48\%$	$4,46\%$
Cluster9	$6,89\%$	$6,97\%$
Cluster10	$8,01\%$	$7,96\%$

Tabulka 5.4: Chyba MTL sítí na testovacích skupinách z Cluster1 až Cluster10

5.3 Experiment 3

Jako poslední experiment byla natrénována síť metodou adversarial learning, v tomto případě je potřeba nastavit hodnotu lambda pro gradient reversal layer na zápornou hodnotu. Proto v první části byla hodnota lambda nastavena na -1, síť byla potom trénována 35000 iterací. Pak byla natrénována druhá síť taktéž pomocí adversarial learning, přičemž lambda byla tentokrát nastavena na -2. Tato síť byla opět trénována po dobu 35000 iterací. . Výsledky průběžného testování chyby sítě na trénovací sadě jsou uvedeny v tabulce 5.5 a na testování na skupinách Cluster1 až Cluster10 je v tabulce 5.6, sítě jsou od sebe v tabulkách odlišeny hodnotou λ.

Číslo iterace	**Chyba znaku**	**Chyba domény**	**Chyba znaku**	**Chyba domény**
λ	-1		-2	
5000	$3,35\%$	$61,95\%$	$4,25\%$	$77,93\%$
10000	$1,75\%$	$82,74\%$	$1,67\%$	$76,14\%$
15000	$1,45\%$	$76,88\%$	$1,54\%$	$88,99\%$
20000	$1,11\%$	$84,23\%$	$1,10\%$	$82,34\%$
25000	$0,94\%$	$81,65\%$	$1,04\%$	$88,99\%$
30000	$1,03\%$	$83,48\%$	$1,14\%$	$85,81\%$
35000	$0,99\%$	$88,10\%$	$0,97\%$	$82,49\%$

Tabulka 5.5: Chyba sití pro každých 5000 iterací při trénování

Číslo iterace	Chyba znaku	
λ	-1	-2
Cluster1	$5,03\%$	$5,06\%$
Cluster2	$4,12\%$	$4,17\%$
Cluster3	$4,87\%$	$4,90\%$
Cluster4	$5,42\%$	$5,42\%$
Cluster5	$4,81\%$	$4,83\%$
Cluster6	$4,15\%$	$4,07\%$
Cluster7	$5,50\%$	$5,50\%$
Cluster8	$4,46\%$	$4,50\%$
Cluster9	$6,90\%$	$6,90\%$
Cluster10	$7,95\%$	$7,97\%$

Tabulka 5.6: Chyba adversarial learning sítí na testovacích skupinách z Cluster1 až Cluster10

5.4 Shrnutí experimentů

Potom, co byly natrénovány všechny sítě, bylo provedeno testování úspěšnosti jednotlivých sítí na testovacích skupinách. Výsledky tohoto testování jsou zaznamenány v následující tabulce 5.7.

Typ sítě	λ	C11	C12	C13	C14	C15	C16
Základní síť	0	$1,51\%$	$18,09\%$	$3,10\%$	$1,81\%$	$5,22\%$	$18,24\%$
Multi-task learning	1	$1,57\%$	$19,01\%$	$3,38\%$	$1,65\%$	$4,98\%$	$16,99\%$
Multi-task learning	2	$1,39\%$	$17,02\%$	$3,46\%$	$1,55\%$	$4,98\%$	$16,99\%$
Adversarial multi-task learning	-1	$1,64\%$	$19,15\%$	$2,99\%$	$1,73\%$	$5,64\%$	$14,29\%$
Adversarial multi-task learning	-2	$1,43\%$	$18,84\%$	$3,70\%$	$1,68\%$	$5,64\%$	$16,84\%$

Tabulka 5.7: Chyba sítí na jednotlivých testovacích skupinách

Tabulka zobrazuje chybu při rozpoznávání znaku pro jednotlivé skupiny pro danou síť (Cluster11 až Cluster16), sítě jsou také od sebe odlišeny pomocí lambda gradient reversal layer se kterým byla daná síť trénována. V následující tabulce jsou shrnuty průměrné chyby.

Typ sítě	λ	Chyba trénovací data	Chyba testovací data
Základní síť	0	$5,316\%$	$7,995\%$
Multi-task learning	1	$5,311\%$	$7,930\%$
Multi-task learning	2	$5,318\%$	$7,565\%$
Adversarial multi-task learning	-1	$5,321\%$	$7,573\%$
Adversarial multi-task learning	-2	$5,332\%$	$8,022\%$

Tabulka 5.8: Průměrná chyba sítí na trénovacích a testovacích datech

Tabulka 5.8 shrnuje průměrnou chybu sítí na trénovacích a testovacích datech. Learning rate λ potom od sebe odlišuje stejné sítě, které byly trénovány s různou λ.
Z hodnot v tabulce 5.8 je vidět, že všechny metody mají skoro stejnou úspěšnost na trénovacích datech, zatímco na testovacích datech už k rozdílům v přesnosti sítí dochází. Hodnota $\lambda = -2$ byla jediná, která nemá zlepšení oproti základní síti. Největší zlepšení na testovací

datové sadě přinesly sítě s hodnotou $\lambda = -1$ a $\lambda = 2$, kde multi-task learning ($\lambda = 2$) mělo o $5,38\%$ a adversarial multi-task learning ($\lambda = -1$) o $5,28\%$ oproti základní síti. Multi-task learning dosahoval v průměru lepších výsledků než adversarial learning, ale adversarial learning vykazoval v některých skupinách lepší výsledky, o $0,47\%$ na Cluster13 a o $2,7\%$ na Cluster16. K ověření lepších výsledků adversarial learning v některých skupinách by bylo nutné provést další experimenty s rozsáhlejším počtem testovacích skupin.

Adversarial learning síť měla lepší výsledky pro λ blíže 0, je možné že při zvolení λ větší než -1, ale menší než 0 síť by mohla přinést lepší výsledky.

Kapitola 6

Závěr

Cílem této práce bylo vyzkoušet metodu zvýšení přesnosti sítě pro rozpoznávání textu s využitím informace o pisateli. Za tímto účelem byla vybrána metoda učení adversarial learning, která byla v rámci této práce implementována a porovnána s ostatními metodami, které jsou za tímto účelem v současnosti používány.

V rámci experimentálního ověřování účinnosti zvolené metody byla implementována síť na rozpoznávání textu. Následně pak byly implementovány i sítě, které využívají metody učení multi-task learning a adversarial learning za účelem zvýšení přesnosti základní sítě. V experimentech byly testovány všechny tyto sítě pro porovnání, jestli tento přístup přináší nějaké zlepšení oproti ostatním metodám.

Experimenty byly provedeny na vybraných skupinách z datové sady IMPACT s cílem získat skupiny, které jsou reprezentovány pouze v trénovací sadě a taktéž pro testovací sadu. Důvodem tohoto postupu je předpoklad, že hlavní přínos metody adversarial learning se projeví na datech, která nejsou reprezentována v trénovací sadě.

Skupiny pro účely této práce reprezentují informaci o pisateli, která je obsažena v daném dokumentu. V trénovací sadě je v každé skupině jedna kniha, což běžně odpovídá jednomu pisateli.

Při experimentech bylo zjištěno, že adversarial learning dosahuje průměrné chyby znaku $5,321\%$ na trénovací sadě a $7,573\%$ na testovací sadě. Metodu multi-task learning v případě trénovací sady dosahuje chyby $5,318\%$ a pro testovací sadu $7,565\%$. Jako poslední základní síť dosahuje na trénovací sadě úspěšnosti $5,316\%$ a na testovací sadě $7,995\%$.

Metoda adversarial learning měla srovnatelné zlepšení s multi-task learning $5,28\%$ a $5,38\%$ resp. oproti základní síti na rozpoznávání textu. Síť používající metodu adversarial learning vykazovala lepší výsledky pro learning rate λ blíže 0. Domnívám se na základě provedených experimentů, že by mohla volba hodnoty pro learning rate λ větší než -1, ale menší než 0 síť přinést lepší výsledky v rozpoznávání textu.

Literatura

[1] *Activation Functions - ML Glossary documentation* [online]. [cit. 2021-26-07]. Dostupné z: https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html.

[2] *CS231n Convolutional Neural Networks for Visual Recognition* [online]. [cit. 2021-26-07]. Dostupné z: https://cs231n.github.io/convolutional-networks/.

[3] Cireşan, D., Meier, U. a Schmidhuber, J. *Multi-column Deep Neural Networks for Image Classification*. 2012.

[4] DAWSON, C. W. a WILBY, R. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*. Taylor & Francis. 1998, sv. 43, č. 1, s. 47–66. DOI: 10.1080/02626669809492102. Dostupné z: https://doi.org/10.1080/02626669809492102.

[5] Ganin, Y. a Lempitsky, V. *Unsupervised Domain Adaptation by Backpropagation*. 2015.

[6] Graves, A. *Generating Sequences With Recurrent Neural Networks*. 2014.

[7] Graves, A., Fernández, S., Gomez, F. a Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2006, s. 369–376. ICML '06. DOI: 10.1145/1143844.1143891. ISBN 1595933832. Dostupné z: https://doi.org/10.1145/1143844.1143891.

[8] Ioffe, S. a Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015.

[9] Kingma, D. P. a Ba, J. *Adam: A Method for Stochastic Optimization*. 2017.

[10] Lecun, Y., Bottou, L., Bengio, Y. a Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, sv. 86, č. 11, s. 2278–2324. DOI: 10.1109/5.726791.

[11] Meng, Z., Li, J., Chen, Z., Zhao, Y., Mazalov, V. et al. Speaker-Invariant Training Via Adversarial Learning. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. Apr 2018. DOI: 10.1109/icassp.2018.8461932. Dostupné z: http://dx.doi.org/10.1109/ICASSP.2018.8461932.

[12] Papadopoulos, C., Pletschacher, S., Clausner, C. a Antonacopoulos, A. The IMPACT Dataset of Historical Document Images. In: *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. New York, NY, USA: Association for Computing Machinery, 2013, s. 123–130. HIP '13. DOI: 10.1145/2501115.2501130. ISBN 9781450321150. Dostupné z: https://doi.org/10.1145/2501115.2501130.

[13] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché Buc, F. d', Fox, E. et al., ed. *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, s. 8024–8035. Dostupné z: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[14] Pauly, L., Peel, H., Luo, S., Hogg, D. a Fuentes, R. Deeper Networks for Pavement Crack Detection. In: Cheng, M.-Y. N. T. U. o. S., Technology), Chen, H.-M. N. T. U. o. S., Technology), Chiu, K. C. N. T. U. o. S. et al., ed. *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*. Taipei, Taiwan: Tribun EU, s.r.o., Brno, July 2017, s. 479–485. DOI: 10.22260/ISARC2017/0066. ISBN 978-80-263-1371-7.

[15] Qin, Z., Yu, F., Liu, C. a Chen, X. *How convolutional neural network see the world - A survey of convolutional neural network visualization methods*. 2018.

[16] Ruder, S. *An Overview of Multi-Task Learning in Deep Neural Networks*. 2017.

[17] Shinohara, Y. Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition. In: *Interspeech 2016*. 2016, s. 2369–2372. DOI: 10.21437/Interspeech.2016-879. Dostupné z: http://dx.doi.org/10.21437/Interspeech.2016-879.

[18] Wang, S., Rohdin, J., Burget, L., Plchot, O., Qian, Y. et al. On the Usage of Phonetic Information for Text-Independent Speaker Embedding Extraction. In: *Proc. Interspeech 2019*. 2019, s. 1148–1152. DOI: 10.21437/Interspeech.2019-3036. Dostupné z: http://dx.doi.org/10.21437/Interspeech.2019-3036.

[19] Yonaba, H., Anctil, F. a Fortin, V. Comparing Sigmoid Transfer Functions for Neural Network Multistep Ahead Streamflow Forecasting. *Journal of Hydrologic Engineering*. 2010, sv. 15, č. 4, s. 275–283. DOI: 10.1061/(ASCE)HE.1943-5584.0000188. Dostupné z: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29HE.1943-5584.0000188.

[20] Zhang, Z., Luo, P., Loy, C. C. a Tang, X. Facial Landmark Detection by Deep Multi-task Learning. In: Fleet, D., Pajdla, T., Schiele, B. a Tuytelaars, T., ed. *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, s. 94–108. ISBN 978-3-319-10599-4.

Příloha A

Složky na paměťovém mediu

- src – zdrojové kódy
- text – text bakalářské práce
- other – video