VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVÁNÍ ŘEČI PRO LETECKOU KOMUNIKACI

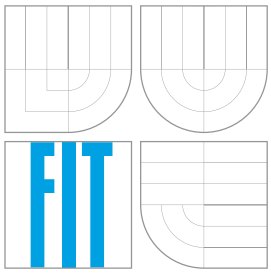
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

KATEŘINA ŽMOLÍKOVÁ

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVÁNÍ ŘEČI PRO LETECKOU KOMUNIKACI

SPEECH RECOGNITION FOR AIR TRAFFIC COMMUNICATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — KATEŘINA ŽMOLÍKOVÁ
AUTHOR

VEDOUCÍ PRÁCE — Ing. KAREL VESELÝ,
SUPERVISOR

BRNO 2014

Abstrakt

Tato bakalářská práce se zabývá rozpoznáváním řeči. Jejím cílem je postavit systém rozpoznávání řeči založený na neuronových sítích a otestovat jej na nahrávkách letecké komunikace. Výsledný akustický model bude použit v projektu A-PiMod. Postavený systém dosáhl na testovacích datech úspěšnost 29.5% WER. Dalším úkolem práce byly experimenty s neuronovými sítěmi, které jsou součástí akustického modelu. První experimenty zkoumaly možnost jejich zjednodušení a urychlení a dopad na úspěšnost rozpoznávání. Další se zabývaly aktivační funkcí rectifier a také konvolučními neuronovými sítěmi. V experimentech s konvolučními neuronovými sítěmi bylo dosáhnuto 1.5% zlepšení a dosáhly tak o 0.4% lepšího výsledku než plně propojená neuronová síť se stejnou architekturou.

Abstract

This thesis deals with speech recognition. The aim is to build a speech recognition system based on neural networks and test it on recordings of air traffic communication. Final acoustic model will be used in project A-PiMod. The system reached word error rate 29.5%. Next task of this thesis was to experiment with neural networks which are part of acoustic model. First experiments explored its simplification and accelaration and its impact on error rate. Next experiments dealt with activation function rectifier and convolutional neural networks. Experiments with convolutional neural networks achieved 1.5% improvement, so the final result was 0.4% better than fully connected network with the same architecture.

Klíčová slova

rozpoznávání řeči, neuronové sítě, letecká komunikace, konvoluční neuronové sítě

Keywords

speech recognition, neural networks, air traffic communication, convolutional neural networks

Citace

Kateřina Žmolíková: Rozpoznávání řeči pro leteckou
komunikaci, bakalářská práce, Brno, FIT VUT v Brně, 2014

Rozpoznávání řeči pro leteckou komunikaci

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Karla Veselého.

.....
Kateřina Žmolíková
21. května 2014

Poděkování

Ráda bych poděkovala Ing. Karlovi Veselému za jeho nekonečnou trpělivost, všechny rady a vedení celé práce. Dále také děkuji celé skupině Speech@FIT za uvedení do světa rozpoznávání řeči.

Obsah

Kapitola 1

Úvod

Rozpoznávání řeči, které je hlavním předmětem této práce, prošlo v minulosti dlouhým vývojem. Zatímco první systémy uměly rozpoznávat spíše izolovaná slova s malým slovníkem, dnešní systémy již dokáží s přijatelnou úspěšností rozpoznávat souvislé spontánní promluvy. Uplatnění takových systémů je široké, mohou být využity například pro ovládání různých přístrojů hlasem, pro online titulkování pořadů nebo automatický překlad jazyka z řeči do řeči.

Systém rozpoznávání řeči vybudovaný v rámci této práce bude využit v projektu A-PiMod, jehož cílem je zvýšit leteckou bezpečnost. Jeho součástí je vytvoření multimodálního kokpitu, který může být ovládán jak ručně, tak řečí nebo gesty. Skupina Speech@FIT se do tohoto projektu zapojuje právě v rámci rozpoznání řeči. Mým úkolem bylo natrénovat akustický model, který byl dále do projektu integrován Karlem Benešem.

V práci jsem se zaměřovala na neuronové sítě, které se dnes používají ve state-of-the-art systémech rozpoznávání řeči. Systém, který jsem vybudovala, je na nich založený. S neuronovými sítěmi jsem také provedla experiment mířící k jejich urychlení a několik experimentů s alternativními architekturami sítí.

Členění práce je následovné. Kapitola 2 se bude zabývat základy rozpoznávání řeči nutnými pro postavení rozpoznávače. Kapitola 3 probírá neuronové sítě, jejich definici, trénování a využití v rozpoznávání řeči. Kapitola 4 popíše systém rozpoznávání řeči, který jsem vytvořila, konkrétní postupy, které jsem použila a ukáže úspěšnost tohoto systému. Kapitola 5 popíše experimenty, které jsem prováděla s neuronovými sítěmi.

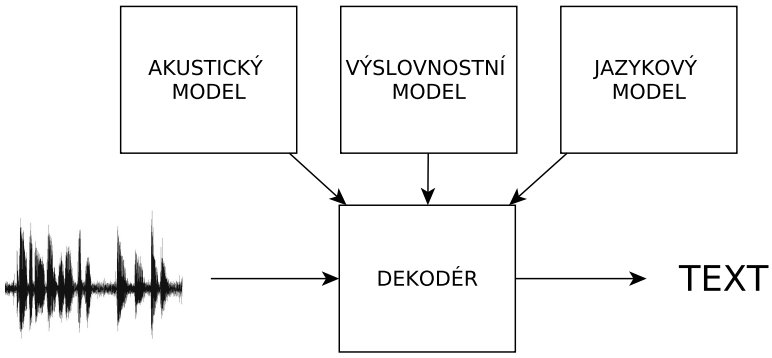
Kapitola 2

Základní principy rozpoznávání řeči

Podoba řečového signálu, který zachycuje plynulou lidskou řeč, je ovlivněna mnoha faktory jako jsou parametry hlasového ústrojí řečníka, situace, ve které je promluva pronášena nebo i okolní šum. To dělá z rozpoznávání řeči složitý úkol, který zasahuje do oborů zpracování signálu, strojového učení, lingvistiky a dalších. Systém, který dokáže řeč rozpoznat, je úzce spjatý s teorií z těchto oblastí. Tato kapitola se proto bude zabývat nutnými základy teorie pro vytvoření takového rozpoznávače.

Nejčastější přístup k rozpoznávání řeči je založený na metodách využívajících statistických modelů řeči a jazyka. Obecný graf rozpoznávače řeči a jeho částí je ukázán na obrázku 2.1. Hlavním modelem v systému je *akustický model*, který popisuje závislost řečového signálu na posloupnosti akustických jednotek, které byly vysloveny. V systémech s velkým slovníkem se akustické modely vytvářejí na úrovni fonémů, což jsou jednotky řeči přibližně odpovídající hláskám. Pro fonémy se na základě trénovacích dat vytvoří modely, jejichž zřetězením pak lze modelovat celá slova a promluvy. Podrobnější popis fungování akustického modelu bude následovat v části 2.2.

Další důležitou částí systému je *jazykový model*, který odráží zákonitosti jazyka a predikuje pravděpodobnost následujícího slova v promluvě v závislosti na historii. Jazykový model bude více popsán v části 2.4. Protože jazykový model pracuje se slovy a akustický s fonémy, je zapotřebí znát způsob, jakým slova z fonémů skládat. To popisuje *výslovnostní model* označovaný jako slovník nebo lexikon.



Obrázek 2.1: Obecný graf rozpoznávání řeči.

Matematicky lze celý problém rozpoznávání řeči vyjádřit jako nalezení posloupnosti slov $\hat{W}$ s největší posteriorní pravděpodobností $P(W|S)$ pro daný řečový signál S. Podle Bayesova pravidla lze tato pravděpodobnost vyjádřit jako

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|S) = \underset{W}{\operatorname{argmax}} \frac{P(W)P(S|W)}{P(S)}, \tag{2.1}$$

kde pravděpodobnost $P(S)$ nezávisí na posloupnosti slov, při hledání maxima ji tedy můžeme ignorovat a maximalizovat pouze součin apriorní pravděpodobnosti posloupnosti slov $P(W)$ a pravděpodobnosti signálu při dané posloupnosti slov $P(S|W)$

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W,S) = \underset{W}{\operatorname{argmax}} P(W)P(S|W). \tag{2.2}$$

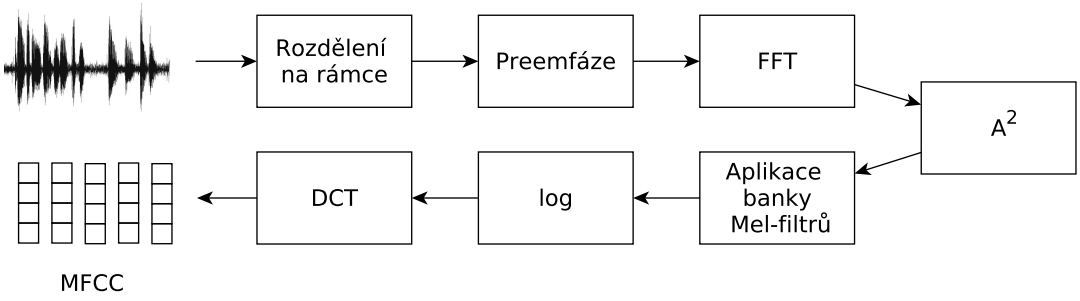
Proces určení posloupnosti W, pro kterou je tento součin maximální, se nazývá dekódování. Hledáme při něm pro daný vstupní signál optimální cestu skrz rozpoznávací síť, která je reprezentována konečným automatem s vahami, tzv. weighted transducerem. Více tuto strukturu celé rozpoznávací sítě popíši v sekci 2.5.

Následující sekce se kromě již zmíněného budou také věnovat extrakci příznaků 2.1 a hodnocení chybovosti rozpoznávače 2.6. Protože teorie rozpoznávání řeči je velmi rozsáhlá, u všech témat bude spíše stručné uvedení do problému potřebné pro tuto práci a odkazy na podrobnější zdroje.

2.1 Extrakce příznaků

Ještě před popisem jednotlivých komponent rozpoznávače je třeba zmínit, že uvnitř systému se nepracuje se samotným řečovým signálem, ale s příznaky, které jej reprezentují. Jde o posloupnost vektorů, která je ze signálu vyextrahována tak, aby bylo zachováno co nejvíce informace užitečné pro následné rozpoznávání. Hlavním důvodem pro extrakci příznaků je redukce dimenze vstupních dat. Postupy extrakce příznaků využívají metod zpracování signálu a jsou inspirované poznatky o procesu slyšení a řečové produkce. Nejčastěji používanými příznaky v rozpoznávání řeči jsou MFCC, PLP, FBank. Postup extrakce MFCC příznaků je naznačen an obrázku 2.2.

Více o extrakci příznaků je možné najít v knize *Mluvíme s počítačem česky* [12], kde jsou v úvodních kapitolách nejprve popsány vlastnosti mluvené řeči, poté podstata metod analýzy řečového signálu. Podobný úvod je také v knize *Spoken Language Processing* [8].



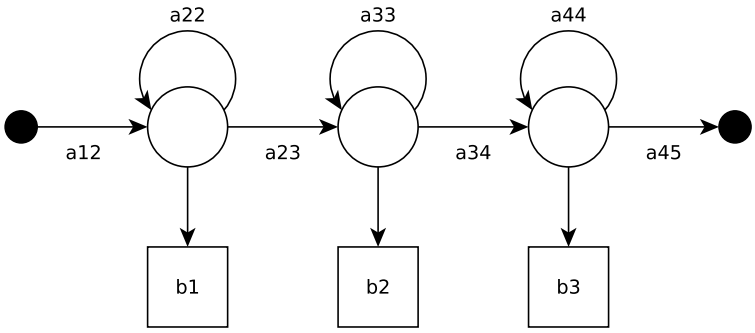
Obrázek 2.2: Příklad extrakce příznaků MFCC.

2.2 Akustický model

Jak již bylo řečeno v úvodu kapitoly, akustický model zachycuje podobu fonémů v prostoru vstupních příznaků. Pro každý foném je vytvořen model, který pro vstupní posloupnost příznaků určuje pravděpodobnost, že tyto příznaky náleží právě danému fonému. Pro akustické modelování se téměř výhradně používají *Skryté Markovovy modely* (neboli HMM z anglického Hidden Markov Model).

2.2.1 Skryté Markovovy modely

Skrytý Markovův model je pravděpodobnostní konečný automat. Každý přechod ze stavu i do stavu j je ohodnocen pravděpodobností přechodu a_{ij}. Každému stavu j navíc přísluší výstupní pravděpodobnost $b_j(o_t)$, které určuje s jakou pravděpodobností vektor o_t náleží danému stavu.



Obrázek 2.3: Skrytý Markovův model.

$$a_{ij} = P(s(t+1) = s_j | s(t) = s_i) \quad \sum_{j=1}^{N} a_{ij} = 1 \tag{2.3}$$

$$b_j(o_t) = P(o_t | s(t) = s_j) \tag{2.4}$$

Jako struktura modelu jednoho fonému je většinou využívána levo-pravá topologie se třemi stavy (viz obrázek 2.3). Pro určení hodnoty výstupní pravděpodobnosti $b_j(o_t)$ je typicky využívána *Směs Gaussovských rozložení* (neboli GMM z anglického Gaussian Mixture Model).

$$b_j(o_t) = \sum_{i=1}^{N} w_i \frac{1}{\sqrt{(2\pi)^P |\mathbf{\Sigma}_i|}} e^{-\frac{1}{2}(x-\mu_i)^T \mathbf{\Sigma}_i{}^{-1}(x-\mu_i)} \quad \sum_i w_i = 1, \tag{2.5}$$

kde parametry modelu jsou střední hodnoty jednotlivých gaussovských rozložení μ, jejich kovarianční matice Σ a také váhy w.

Alternativou jsou *neuronové sítě*, kterým se bude věnovat celá kapitola 3. Parametry modelu a_{ij} a modelů poskytujících výstupní pravděpodobnost $b_j(o_t)$ jsou odhadnuty statisticky trénováním na sadě nahrávek s referenčním přepisem.

Jakmile jsou parametry modelu M známy, je možné určit pravděpodobnost $P(O|M)$, se kterou tento model bude trénovací posloupnost O generovat. Její vyhodnocení ale není triviální z důvodu, že posloupnost stavů S, kterou model prochází je skrytá. Pravděpodobnost může být vyjádřena jako suma přes všechny možné posloupnosti stavů

$$P(O|M) = \sum_{S} P(O, S|M), \tag{2.6}$$

často je ale aproximována pomocí nejpravděpodobnější posloupnosti stavů

$$\hat{P}(O|M) = \max_{S} P(O, S|M), \tag{2.7}$$

která může být efektivně spočítána pomocí Viterbiho algoritmu.

Podrobnější vhled do použití Skrytých Markovových modelů v rozpoznávání řeči uvádí opět Psutka v [12] a Huang v [8]. Dalším zdrojem je také tutoriál *The HTK Book* [17], kde je vedle základů toolkitu HTK také teoretický úvod do HMM.

2.3 Kontextová závislost

Jako jednotky, se kterými pracuje akustický model, byly dosud uváděny fonémy. Jejich realizace při plynulé řeči jsou velmi rozdílné díky koartikulaci, proto je vhodné rozlišovat fonémy s různým kontextem. Nejčastěji bývají použity tzv. *trifony*, tedy fonémy uvažující jeden předchozí a jeden následující foném.

S kontextově závislými fonémy ale nastává problém s jejich velkým množstvím. Například při 38 různých fonémech by množství trifonů vzrostlo na $38^3 = 54872$. Mnoho z těchto trifonů se vůbec nebo velmi zřídka vyskytne v trénovacích datech. Pro vyřešení těchto problémů se používají techniky shlukování celých trifonů nebo jejich stavů. Přesný popis algoritmů shlukování uvádí Psutka v 5. kapitole [12].

2.4 Jazykový model

Jazykový model pomáhá rozpoznávači určit pravděpodobnost určité sekvence slov bez ohledu na zvuková data. Jeho úkolem je odhadovat apriorní pravděpodobnost poslouposti slov $P(W)$. Model by měl brát v úvahu nejen obecná pravidla daného jazyka, ale také cílovou oblast rozpoznávání – jiný jazykový model bude mít rozpoznávač odborných přednášek než rozpoznávač běžných konverzací.

V praxi se většinou používají *n-gramové modely*, kdy pravděpodobnost každého slova v promluvě závisí na posledních $n - 1$ slovech.

$$P(w_1^k) = \prod_{i-1}^{k} P(w_i|w_{i-n+1}^{i-1}) \tag{2.8}$$

Nejpoužívanější jsou unigramové ($n = 1$), bigramové ($n = 2$) a trigramové ($n = 3$) modely, které jsou natrénovány na textových korpusech.

Celou kapitolu různým jazykovým modelům, jejich vytváření a posouzení kvality věnuje Psutka [12].

2.5 Rozpoznávací síť

V minulých sekcích byly uvedeny jednotlivé komponenty rozpoznávače – Skryté Markovovy modely, kontextovou závislost, výslovnostní slovníky a jazykové modely. Všechny tyto komponenty mohou být reprezentovány pomocí *váhovaných konečných transducerů* (neboli WFST z anglického Weighted finite-state transducer). Díky této reprezentaci je možné všechny modely pomocí operace kompozice spojit do jedné rozpoznávací sítě a tu následně optimalizovat a použít ji pro hledání nejvhodnější posloupnosti slov.

Váhovaný konečný transducer je konečný automat, jehož přechodům jsou přiřazeny jak vstupní, tak i výstupní symboly. Reprezentuje tak relaci mezi dvěma formálními jazyky. Každý přechod má navíc také váhu, která může reprezentovat například pravděpodobnost nebo cenu přechodu. S váhovanými transducery lze s mírnými modifikacemi provádět klasické operace jako determinizace, minimalizace, kompozice. Další používanou operací je stlačení vah, které distribuuje váhy přechodů směrem k počátečním stavům a zefektivňuje tak vyhledávání v automatu.

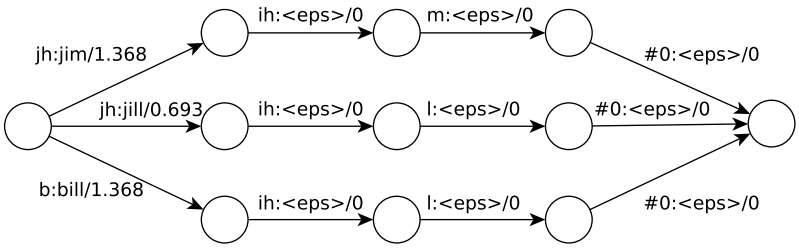
Typický automat používaný pro rozpoznávání řeči se skládá ze čtyř částí

$$H \circ C \circ L \circ G,$$

kde

- H reprezentuje sjednocení všech Skrytých Markovových modelů.
- C řeší kontextovou závislost. Jeho vstupními symboly jsou fonémy a výstupními kontextové fonémy.
- L zastupuje výslovnostní lexikon. Mapuje tedy kontextové fonémy na celá slova.
- G reprezentuje jazykový model.

Operátor $\circ$ zde značí kompozici, která se provádí zprava, ekvivalentní zápis je tedy $H \circ (C \circ (L \circ G))$. V jednotlivých krocích kompozice celého automatu probíhají operace determinizace a minimalizace pro efektivitu a redukci velikosti celého automatu. Na obrázku 2.4 je zobrazen automat vzniklý složením lexikonu a gramatiky $L \circ G$.



Obrázek 2.4: Ukázka váhovaného konečného transduceru. Jde o složení jednoduchého lexikonu a gramatiky $L \circ G$.

Pro podrobný popis váhovaných konečných transducerů doporučuji článek *Speech recognition with weighted finite-state transducers* [10], kde je uvedena jejich definice, také jsou zde prezentovány obecné algoritmy a detailně vysvětlena jejich aplikace na rozpoznávání řeči.

2.6 Hodnocení chybovosti rozpoznáváče

Pro zhodnocení kvality rozpoznávače řeči je důležité mít k dispozici nějakou metriku jeho chybovosti. Nejčastěji používanou je tzv. *Word Error Rate* (WER). WER počítá se třemi druhy chyb, které se při rozpoznávání mohou objevit:

- *Subs* Substituce – slovo bylo nahrazeno nesprávným slovem
- *Dels* Smazání – správné slovo bylo v rozpoznané větě vynecháno
- *Ins* Vložení – do rozpoznané věty bylo přidáno nesprávné slovo

WER je pak definováno jako

$$WER = 100\% * \frac{Subs + Dels + Ins}{počet\ slov\ ve\ správné\ větě} \tag{2.9}$$

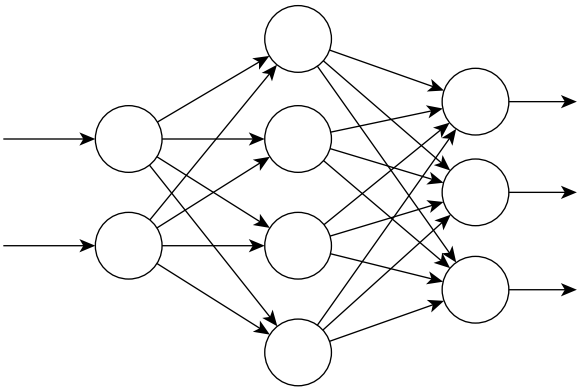
Tato definice je převzata z knihy *Spoken language processing* [8].

Kapitola 3

Neuronové sítě a jejich využití v rozpoznávání řeči

Jak již bylo řečeno v kapitole 2, jádrem akustického modelu v systému rozpoznávání řeči jsou Skryté Markovovy modely. Ke každému stavu těchto modelů je přiřazeno rozložení pravděpodobnosti, které vyjadřuje příslušnost vektoru příznaků danému stavu. Dříve byla pro toto rozložení používána téměř výhradně Směs Gaussovských rozložení. Lepších výsledků lze ale dosáhnout použitím neuronových sítí.

Neuronová síť je matematický model, který vyjadřuje určitou transformaci vstupů na výstupy. S dostatečným počtem parametrů je tato síť schopná s libovolnou přesností aproximovat jakoukoli spojitou funkci. Tato transformace probíhá pomocí základních jednotek sítě – *neuronů*, které jsou vzájemně propojeny a tvoří tak jednotlivé vrstvy sítě. Komunikují spolu posíláním signálů přes váhovaná spojení. Více o tom, jak pracují neurony v části 3.1.



Obrázek 3.1: Příklad neuronové sítě.

Neuronová síť na obrázku 3.1 je tzv. dopředná neuronová síť – na vstup neuronů jsou napojeny výstupy neuronů nižších vrstev. První vrstva je nazývána jako vstupní, poslední jako výstupní a vrstvy mezi nimi jako skryté.

Neuronovou síť lze využít pro úlohu klasifikace vstupů do tříd. Počet výstupů sítě odpovídá počtu tříd a hodnota na daném výstupu reprezentuje pravděpodobnost, že vstup náleží do dané třídy. Neuronová síť na obrázku 3.1 by tedy zařazovala dvoudimenzionální vstupní data do tří tříd. Chování neuronové sítě ovlivňují její parametry – váhy jednotlivých spojení, které lze natrénovat tak, aby prováděly požadovanou klasifikaci. Při trénování

síť zpracovává vstupy z trénovací sady a snaží se upravit své parametry tak, aby dosáhla požadovaných výstupů. Přesný postup trénování bude uveden v sekci 3.2.

V rozpoznávání řeči jsou vstupními daty neuronové sítě vektory příznaků a třídami, do kterých jsou vektory klasifikovány, jsou stavy HMM. Pro získání cílových tříd pro účely trénování se používají výstupy ze systému využívajících GMM. Použití neuronové sítě pro účel rozpoznávání řeči bude podrobněji popsán v části 3.3.

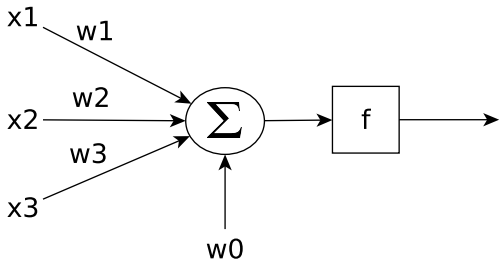
Tato kapitola vychází především z kapitoly o neuronových sítích z knihy *Pattern recognition and machine learning* [2]. V této knize je detailní teoretický rozbor neuronových sítí a jejich využití pro strojové učení. V této práci je z této knihy použit především postup trénování sítí. Další informace byly čerpány z diplomové práce Ing. Karla Veselého [15]. Informace o použití neuronových sítí v rozpoznávání řeči byly získány z článků *Deep Neural Networks for Acoustic Modeling in Speech Recognition* [7] a *Acoustic Modeling Using Deep Belief Networks* [9], ve kterých je tato problematika, kterou se bude zabývat část 3.3 podrobně rozebrána a jsou zde také uvedeny výsledky experimentů.

3.1 Neuron

Neuron je jednoduchá výpočetní jednotka, která je základním stavebním prvkem neuronových sítí. Stejně jako u biologického neuronu, kterým je inspirován, má několik vstupů, na které reaguje svým výstupem. Výstup je pak rozšířen k neuronům v následující vrstvě.

Výpočet probíhá tak, že všechny vstupy jsou vynásobeny váhami a následně sečteny. K tomuto součtu je pak ještě přičten tzv. bias (w_0), který není závislý na vstupu, a výsledek je transformován aktivační funkcí $f(\cdot)$.

$$y = f(\sum_{i=1}^{N} w_i x_i + w_0) \tag{3.1}$$



Obrázek 3.2: Neuron.

Aktivační funkce je v neuronové síti klíčovým prvkem, protože díky ní může síť provádět nelineární transformace. Protože při trénování pomocí backpropagation algoritmu, který bude popsán v 3.2, se používá gradient této funkce, je nutné, aby byla diferenciovatelná. Nyní budou uvedeny nejpoužívanější aktivační funkce.

Logistická sigmoida

Velmi často používanou aktivační funkcí je logistická sigmoida. Její derivace je velmi jednoduchá, což je užitečné při procesu trénování.

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{3.2}$$

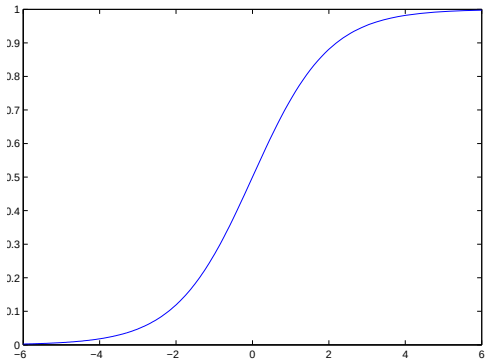
$$\frac{d}{dx}\sigma(x) = \sigma(x) \cdot (1 - \sigma(x)) \tag{3.3}$$

Logistická sigmoida je inverzní funkce k funkci *logit*

$$logit(y) = log(\frac{y}{1-y}), \tag{3.4}$$

z čehož plyne užitečná vlastnost a to, že převádí logaritmus poměru pravděpodobností dvou tříd na pravděpodobnost první z nich.

$$\sigma\left(log\frac{p(a)}{p(b)}\right) = \sigma(logit(p(a))) = p(a). \tag{3.5}$$



Obrázek 3.3: Graf průběhu sigmoidy.

Softmax

Aktivační funkce softmax se používá především na výstupní vrstvě neuronové sítě. Jejím výstupem je vektor, jehož j-tý prvek je definován jako

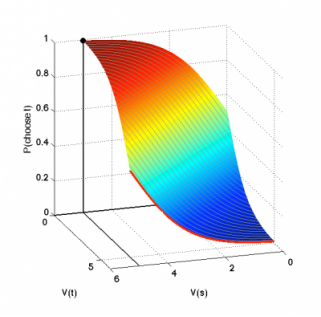
$$softmax_j(\mathbf{x}) = \frac{e^{x_j}}{\sum_{k=1}^{M} e^{x_k}}. \tag{3.6}$$

Součet prvků výstupního vektoru funkce je 1

$$\sum_{j=1}^{M} softmax_j(\mathbf{x}) = \frac{\sum_{j=1}^{M} e^{x_j}}{\sum_{k=1}^{M} e^{x_k}} = 1, \tag{3.7}$$

může být tedy intepretován jako rozložení pravděpodobnosti. Pokud je tedy $softmax$ použit na výstupní vrstvě neuronové sítě, je možné její výstupy intepretovat jako posteriorní pravděpodobnosti jednotlivých tříd. Obrázek 3.4[1] ukazuje průběh funkce.

[1] Obrázek 3.4 pochází z webu http://wagerlab.colorado.edu/wiki/doku.php/help/core/figure_gallery.



Obrázek 3.4: Graf průběhu softmax funkce.

3.2 Trénování neuronové sítě

Jak bylo naznačeno v úvodu, cílem trénování je upravit parametry neuronové sítě (váhy jednotlivých přechodů) tak, aby co nejlépe klasifikovala data z trénovací sady. Využívá se k tomu algoritmus *gradient descent*, který se snaží minimalizovat chybovou funkci vah tak, že od nich odečítá gradient chybové funkce - tedy směr, kde chyba nejvíce roste. Princip ilustruje obrázek 3.5[2].

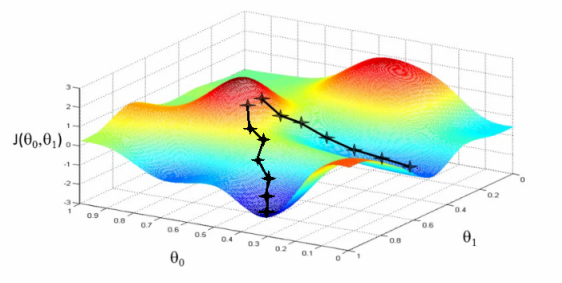
$$\mathbf{w}^{\mathcal{T}+1} = \mathbf{w}^{\mathcal{T}} - \eta \nabla E(\mathbf{w}^{\mathcal{T}}) \tag{3.8}$$

η je zde tzv. *learning rate* neboli zvolený krok učení, který udává, jak velká úprava vah se v každém kroku udělá.

Chybová funkce E v tomto případě závisí na celé trénovací sadě. Tato metoda, kdy se váhy aktualizují až po získání gradientu pro celou trénovací sadu, se nazývá batch trénování. Výhodnější a více používané je tzv. online trénování, kdy se aktualizace vah provádí vždy na základě jednoho data. Kompromisem mezi těmito dvěma variantami je trénování po blocích zvolené velikosti, které také umožňuje optimální využití cache.

$$\mathbf{w}^{\mathcal{T}+1} = \mathbf{w}^{\mathcal{T}} - \eta \nabla E_n(\mathbf{w}^{\mathcal{T}}) \tag{3.9}$$

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w}) \tag{3.10}$$



Obrázek 3.5: Algorimuts gradient descent.

[2]Obrázek je převzat ze slidů http://mlg.eng.cam.ac.uk/roger/doc/bigdata.pdf.

3.2.1 Chybové funkce

Mezi nejpoužívanější chybové funkce patří střední kvadratická odchylka a cross entropie.

Střední kvadratická odchylka

Střední kvadratická odchylka je definována jako

$$E_n = \frac{1}{2}\sum_{j=1}^{M}(y(x_n; w) - t_n)^2 \tag{3.11}$$

a nejčastěji se využívá pro problém lineární regrese. Její derivace je triviální

$$\frac{\partial E_n}{y_n} = y_n - t_n. \tag{3.12}$$

Cross entropie

Cross entropie se používá v kombinaci s aktivační funkcí softmax a je definována jako

$$E_n = -\sum_{j=1}^{M} t_{nj} ln(y_{nj}). \tag{3.13}$$

Její derivace nabývá tvaru

$$\frac{\partial E_n}{y_n} = \frac{t_{nj}}{y_{nj}}. \tag{3.14}$$

3.2.2 Postup trénování

Při trénování pomocí algoritmu gradient descent je potřeba v každém kroku vyhodnocovat gradient chybové funkce pro vstup a požadovaný výstup. Pro každý prvek tohoto gradientu je možné pomocí řetězového pravidla rozepsat derivaci jako

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} \tag{3.15}$$

$$a_j = \sum_i w_{ji} z_i, \tag{3.16}$$

kde z_i, je tzv. aktivace neuronu, tedy výstup z i-té jednotky minulé vrstvy a vstup do j-té jednotky aktuální vrstvy. Ze vzorců 3.15 a 3.16 přímo vychází

$$\frac{\partial a_j}{\partial w_{ji}} = z_i \tag{3.17}$$

Zlomek $\frac{\partial E_n}{\partial a_j}$ se často zapisuje jako δ_j a označuje se jako chyba. Celou derivaci lze tedy zapsat jako

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i. \tag{3.18}$$

Pokud je na výstupní vrstvě použitá aktivační funkce softmax a jako chybová funkce je uvažována cross entropie, chyba se nám zjednoduší a nabude tvaru

$$\delta_k = \frac{\partial E_n}{\partial a_k} = y_k - t_k. \tag{3.19}$$

U předchozích vrstev je opět možné použít řetězové pravidlo

$$\delta_j = \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j}, \tag{3.20}$$

kde zlomek $\frac{\partial E_n}{\partial a_k}$ je opět chyba δ_k a $\frac{\partial a_k}{\partial a_j}$ je součin derivace aktivační funkce a váhy spojení w_{kj}. Výsledný vzorec vypadá následovně

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k. \tag{3.21}$$

Tato rovnice vyjadřuje princip zpětné propagace, kdy se chyba v jednotlivých vrstvách počítá pomocí chyb ve vrstvě další.

Celý postup trénování se tedy dá shrnout do několika kroků. Nejprve se vstupní vektor z trénovacích dat propaguje přes síť pomocí 3.1. Na výstupních jednotkách se poté spočítá chyba pomocí 3.19. Tato chyba se pomocí zpětné propagace 3.21 použije k výpočtu chyby na všech jednotkách v síti. Pomocí chyby je poté vypočítán gradient 3.18, který se použije pro aktualizaci vah 3.9.

3.2.3 Overfitting

Při trénování může dojít k problému přetrénování, kdy neuronová síť sice čím dál lépe vyhodnocuje trénovací data, ale již negeneralizuje na nová. Tomuto problému se dá vyhnout použitím tzv. held-out sady, tedy vyčlenením malého množství trénovacích dat. Tato data se pak nepoužijí pro samotné trénování, ale po každé iteraci se na nich vyhodnotí chybová funkce. Pokud se chyba začně zvětšovat, neuronová síť se začíná přetrénovávat a trénování je zastaveno. Tato technika se nazývá *early stopping*.

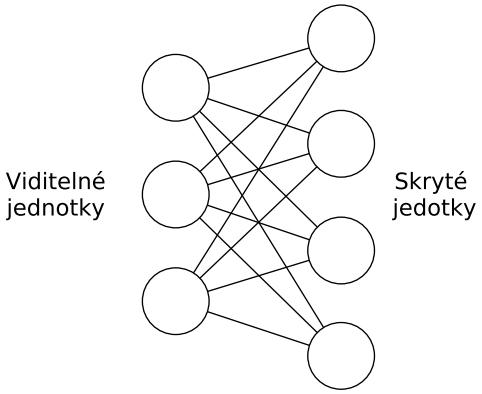
3.3 Použití v rozpoznávání řeči

Neuronové sítě v rozpoznávači řeči přebírají roli Směsi Gaussovských rozložení. Jejich vstupem jsou vektory příznaků společně s jejich kontextem a výstupem jsou posteriorní pravděpodobnosti jednotlivých stavů HMM $P(S|X)$. Ty jsou během dekódování pomocí Bayesova vzorce převáděny na likelihood $P(X|S)$. Apriorní pravděpodobnost $P(S)$ je odhadnuta z četnosti výskutu jednotlivých stavů a pravděpodobnost $P(X)$ je zanedbána.

$$P(X|S) \propto \frac{P(S|X)}{P(S)}.$$

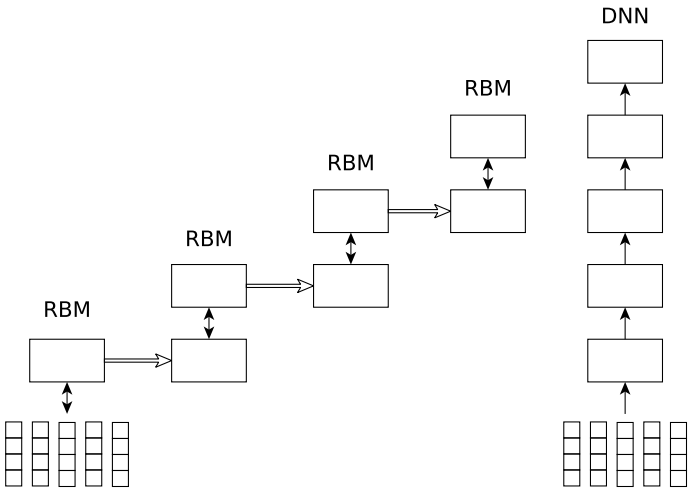
Pro úspěšné rozpoznávání řěči je třeba budovat neuronové sítě s více skrytými vrstvami a velkým počtem neuronů v jedné vrstvě. Takto velké sítě je velmi těžké vyladit a diskriminativní trénování, tak jak bylo popsáno v 3.2 často vede k přetrénování. Proto se využívá dvoufázové trénování, kdy se v první fázi nejdříve neuronová síť generativně předtrénuje. Takto předtrénovaná síť pak slouží jako lepší výchozí bod pro diskriminativní trénování.

Pro generativní natrénování neuronové sítě se používají tzv. *Restricted Boltzmann machines* (RBM). RBM je bipartitní graf, skládá se z viditelných a skrytých jednotek. Mezi viditelými a skrytými jednotkami existují neorientovaná spojení. K natrénování RBM se používá algoritmus *contrastive divergence*.



Obrázek 3.6: Restricted boltzman machine.

Skryté jednotky se natrénováním naučí jinou reprezentaci původních příznaků. Hodnoty těchto skrytých jednotek se poté mohou použít jako viděná data pro natrénování dalšího RBM. Opakováním tohoto postupu je možné vytvořit libovolný počet vrstev. Přidáním softmax vrstvy s jedním neuronem pro každý HMM stav vznikne finální síť, která je následně diskriminativně dotrénována.

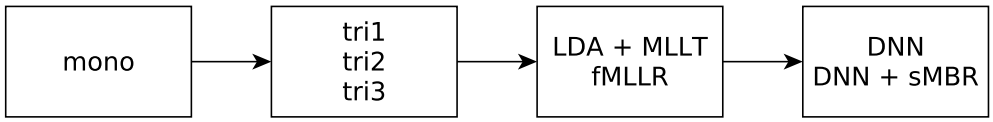


Obrázek 3.7: Pretraining neuronové sítě.

Kapitola 4

Popis vytvořeného systému rozpoznávání řeči

Hlavním cílem této práce bylo natrénování state-of-the-art rozpoznávače řeči pro angličtinu založeného na neuronových sítích. Celkem bylo vytvořeno 8 verzí akustického modelu. První jednoduchý model využíval monofony, poté byl rozšířen na trifonový model. V dalších verzích byly přidány metody lineárních transformací příznaků a poslední dvě verze využívaly samotných neuronových sítí. Na obrázku 4.1 je zobrazen tento postup. Všechny fáze budou postupně uvedeny v této kapitole. Budou zde popsány konkrétní techniky použité při tvorbě tohoto systému. U každé verze systému bude uvedena také jeho úspěšnost na testovacích datech.



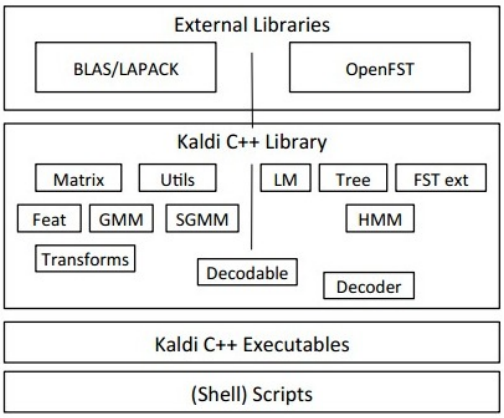
Obrázek 4.1: Graf postupu vytváření systému.

Dalším úkolem práce bylo systém otestovat na nahrávkách letecké komunikace. Podrobnosti o tomto testování a diskuze úspěšnosti bude uvedena v části 4.7.

4.1 Toolkit Kaldi

Celý systém byl implemetován v toolkitu Kaldi [11]. Kaldi je open-source toolkit určený pro výzkumníky v oblasti rozpoznávání řeči. Skládá se z nástrojů pro příkazový řadek napsaných v C++, které jsou pak volány ve skriptovacích jazycích. Celý systém ja založený na konečných tranducerech a využívá pro ně volně dostupné knihovny OpenFST [1]. Obrázek 4.2[2] ukazuje jednotlivé moduly Kaldi.

[1] Knihovna OpenFST http://www.openfst.org
[2] Obrázek 4.2 byl převzat z článku [11].



Obrázek 4.2: Graf toolkitu Kaldi.

4.2 Příprava dat

Pro úspěšnost systému rozpoznávání řeči je klíčové, aby byl natrénovaný na vhodných datech. Protože cílovou aplikací systému v rámci této práce byla letecká komunikace, nejvhodějšími daty pro natrénování by byly právě nahrávky pilotů. Protože taková trénovací sada nebyla k dispozici, bylo nutné zvolit takovou, která k ní má co nejblíže. Jednou z charakteristik, která se objevuje v letecké komunikaci a je podstatná pro rozpoznávání, je přítomnost různých akcentů angličtiny. Z tohoto důvodu byla zvolena sada AMIDA[3], která různé přízvuky angličtiny také obsahuje.

Sada AMIDA obsahuje nahrávky meetingů. Angličtina je na nich velmi různorodá, což sice ztěžuje úkol natrénování systému, systém je ale ve výsledku více robustní. V trénovací sadě se vyskytovalo celkem 1409 různých mluvčích, z nichž u každého bylo v průměru asi 8 minut řeči. Podrobnější údaje o trénovací i testovací sadě následují v tabulce.

	trénovací data	testovací data
počet mluvčích	1374	35
počet promluv	282433	4527
počet hodin	cca 176	cca 3
průměrná délka promluvy	cca 2.2 sekund	cca 2.5 sekund

Tabulka 4.1: Vlastnosti sady AMIDA.

Použitý jazykový model byl bigramový a obsahoval celkem 58788 slov.

Všechna data bylo třeba připravit do formátu, který je vhodný jako vstup pro toolkit Kaldi. Jazykový model byl z formátu ARPA[4] převeden pomocí nástrojů OpenFST do formátu konečného transduceru. Do stejné formy byl převeden i výslovnostní lexikon.

Data potřebná k vytvoření akustického modelu se převádí do formátu jednoduchých tabulek. Jde konkrétně o seznam samotných nahrávek, jejich transkripce, segmentace nahrávek a mapy mluvčích.

[3] AMIDA corpus http://corpus.amidaproject.org

[4] Formát jazykového modelu ARPA http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html

4.3 Extrakce příznaků

Ze zvukových dat je potřeba vyextrahovat příznaky, které se používají v dalších částech systému. Pro tento systém byly zvoleny PLP příznaky. Postup extrakce PLP příznaků je možné nalézt například v [6]. V tomto případě bylo použito 13 koeficientů, společně s jejich prvními a druhými derivacemi, celkově byl tedy použit 39 rozměrný vektor příznaků.

Dále byla použita normalizace střední hodnoty, od příznaků byla tedy odečtena střední hodnota vypočítana ze všech promluv daného řečníka.

4.4 Systém se směsí Gaussovských rozložení

Přestože cílový systém má být založený na neuronových sítích, k jejich natrénování je potřeba již znát příslušnost vektorů příznaků z trénovacích dat ke stavům Skrytého Markovova modelu. Pro získání tohoto zarovnání příznaků byl natrénován model využívající Směsi Gaussovských rozložení. V této sekci bude popsán postup, jak byl tento model natrénován a uvedu výsledky dekódování testovacích dat pomocí tohoto systému.

Monofonový systém

Prvním krokem bylo vytvoření modelu `mono0`, který pracuje s monofony, tedy fonémy bez kontextové informace. Takovýto model je sice velmi jednoduchý, nedosahuje však velmi dobrých výsledků při dekódování. Nicméně zarovnáme-li s ním trénovací data, můžeme natrénovat složitější trifonový model.

Celý model byl napřed inicializován, byl tedy vytvořen HMM pro každý foném, kde ke každému stavu bylo přiřazeno pouze jedno Gaussovské rozložení s globálně odhadnutou střední hodnotou a rozptylem. Tyto modely byly zřetězeny na základě transkripcí trénovacích dat a trénovány pomocí EM algoritmu s Viterbi zarovnáním. Po několika iteracích tohoto algoritmu vždy také dochází ke štěpení gaussovských rozložení, aby bylo dosáhnuto jejich požadovaného množství, v tomto případě celkem 1000. Model byl natrénován pomocí celkem 40 iterací.

Trifonové systémy

Po dokončení trénování monofonového modelu se pomocí něj získalo zarovnání příznaků z trénovacích dat do stavů HMM. V Kaldi je toto zarovnání reprezentováno posloupností čísel vyjadřujících HMM stavy.

```
AMD-S1002B-R1-H01_f3178-AD_0000268_0000404
[ 44358 44357 44357 44428 44427 44427 44427 44427 44427 44427 ... 44517 44517 ]
[ 12402 12401 12401 12498 12497 12497 12746 12745 12745 12745 ]
[ 35996 35995 35995 35995 35995 35995 35995 35995 35995 35995 ... 36379 36379 ]
AMD-S1002B-R1-H01_f3178-AD_0000268_0000404 y_B eh_I s_E
```

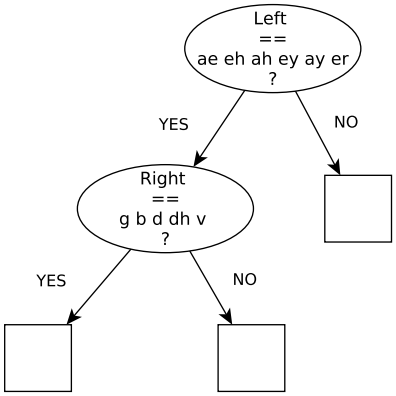
Zde je ukázka zarovnání promluvy, na které je řečeno slovo yes. První řádek je identifikátor této promluvy, následují posloupnosti stavů, náležící postupně fonémům `y_B`, `eh_I` a `s_E` (suffixy `_B`, `_I` a `_E` vyjadřují pozici fonému ve slově).

Z tohoto zarovnání byly pro každý stav každého trifonu spočítány statistiky, konkrétně počet příznaků náležící tomuto stavu, jejich součet a součet druhých mocnin. Pomocí slučovacích technik se z těchto statistik vytvořily tzv. questions, což jsou skupiny fonémů

s podobnými akustickými vlastnostmi. Zde je ukázka několika řádků z automatický vytvořených questions pro finální trifonový model.

```
ae eh ah ey ay er
ch zh f s sh
z g b d dh
v
```

Nakonec se za pomocí těchto questions vytvořil rozhodovací strom, který již zajišťuje samotné slučování stavů trifonů. Následující trénování pak probíhalo stejně jako v případě monofonů, jen s využitím tohoto stromu.



Obrázek 4.3: Ukázka rozhodovacího stromu.

Systémů s trifony bylo vytvořeno několik, kdy v každém dalším byly využity pro inicializaci výstupy získané ze systému minulého. Další systémy byly postupně komplexejší a přesnější.

Výsledky

Tabulka 4.2 uvádí úspěšnost vytvořených systémů na testovacích datech.

Iterace systému	WER [%]	Subs [%]	Ins[%]	Dels [%]
mono0 – monofony	69.5	41.6	2.2	25.8
tri1 – trifony	48.1	29.6	2.7	15.8
tri2 – trifony	46.6	28.8	2.8	15.1
tri3 – trifony	45.4	27.6	3.0	14.7

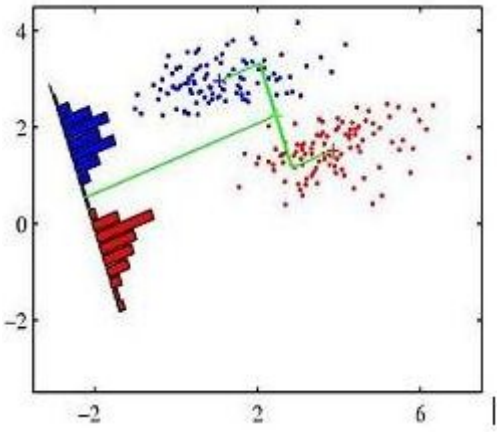
Tabulka 4.2: Úspěšnost systémů se Směsí Gaussovských rozložení.

4.5 Lineární transformace příznaků

Pro další zlepšení úspěšnosti systému byly využity lineární transformace LDA a MLLT a adaptace na řečníka fMLLR, které více popíšu v této sekci.

Lineární diskriminační analýza

LDA neboli Linear discriminant analysis je metoda pro nalezení lineární tranformace určená k redukci dimenze a dekorelaci příznaků. Promítá data do takového směru, ve kterém jsou jednotlivé třídy co nejlépe odděleny a zachovává veškerou informaci nutnou pro diskriminaci mezi třídami. Tento princip zachycuje obrázek 4.4[5]. Tato metoda předpokládá, že příznaky jednotlivých tříd jsou gaussovské a že všechny mají stejnou kovarianční matici. Počet dimenzí, do kterých LDA může data promítnout je omezený počtem tříd, maximálně $N - 1$ dimenzí pro N tříd. Směry, do kterých LDA data promítá jsou dány vlastními vektory matice $\Sigma_{ac}\Sigma_{wc}^{-1}$, kde Σ_{ac} je kovarianční matice váhovaných středních hodnot všech tříd a Σ_{wc} je průměrá kovarianční matice všech tříd váhovaná počtem dat v jednotlivých třídách.



Obrázek 4.4: Lineární diskriminační analýza.

Maximum likelihood linear transform

MLLT neboli Maximum likelihood linear transform je technika, která umožňuje sdílení kovariančních matic několika směsí gaussovských rozložení a tudíž snížení množství parametrů pro jejich reprezentaci. Každá kovarianční matice je reprezentována dvěma parametry – lineární transformací W, která je sdílená a diagonální maticí Λ_j, která je pro každou kovarianční matici unikátní. Kovarianční matice je pak vyjádřena jako

$$\Sigma_j = (W\Lambda_j W^T)^{-1} \tag{4.1}$$

Parametry modelu jsou odhadovány pomocí metody Maximum likelihood. Podrobnější popis této techniky je uveden v článku [4].

Feature-space maximum likelihood linear regression

fMLLR neboli Feature-space maximum likelihood linear regression je metoda adaptace na řečníka. Metody adaptace na řečníka obecně umožňují model nezávislý na řečníkovi přetrénovat na hlas konkrétního řečníka s využitím malého množství trénovacích dat. Metody se dají rozdělit na metody adaptace na úrovni parametrů modelu a na úrovni příznaků, což je méně výpočetně náročná varianta a fMLLR patří právě do této skupiny metod. Transformace střední hodnoty a kovarianční matice příznaků má tvar

[5]Obrázek 4.4 byl převzat z knihy *Pattern recognition and machine learning* [2], kapitola 4.

$$\hat{\mu} = A'\mu + b' \tag{4.2}$$

$$\hat{\Sigma} = A'\Sigma A'^{T} \tag{4.3}$$

Parametry A a b této transformace jsou nalezeny pomocí Expectation maximization algoritmu. Vypočítané transformace jsou pak použity při trénování akustického modelu pro potlačení rozdílu mezi jednotlivými řečníky.

Podrobnější popis této metody je uveden v článku [3].

Výsledky

Následující tabulka ukazuje výsledky systémů s aplikovanými transformacemi.

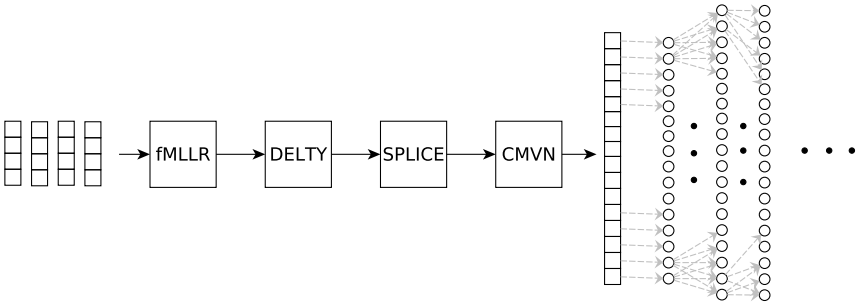
Iterace systému	WER [%]	Subs [%]	Ins[%]	Dels [%]
tri3 – trifony	45.4	27.6	3.0	14.7
tri4 – trifony + LDA + MLLT	38.9	23.9	2.4	12.7
tri5 – trifony + LDA + MLLT + fMLLR	38.0	23.0	2.4	12.6

Tabulka 4.3: Úspěšnost systému s LDA, MLLR a fMLLR.

4.6 Systém s hlubokými neuronovými sítěmi

Posledním krokem bylo nahradit Směsi Gaussovských rozložení z minulých verzí systémů za neuronovou síť, která bude odhadovat příslušnosti vektorů příznaků k jednotlivým stavům. Jako cíle pro natrénování této sítě byly využity výstupy `tri5` systému. Výsledná neuronová síť tedy má celkem 7014 výstupů, což odpovídá počtu stavů tohoto systému.

Před samotným trénováním neuronové sítě je nutné oddělit z trénovácích dat část pro cross-validaci, jak je popsáno v 3.2.3. V tomto případě bylo 90% trénovacích dat využito pro samotné trénování a 10% pro cross-validaci. Dále je potřeba upravit příznaky pro zpracování neuronovou sítí. Příznaky podobně jako u GMM-HMM systému prošly fMLLR transformací. Byly také globálně normalizovány, aby měly nulovou střední hodnotu a jednotkový rozptyl.



Obrázek 4.5: Feature transformace před neuronovou sítí.

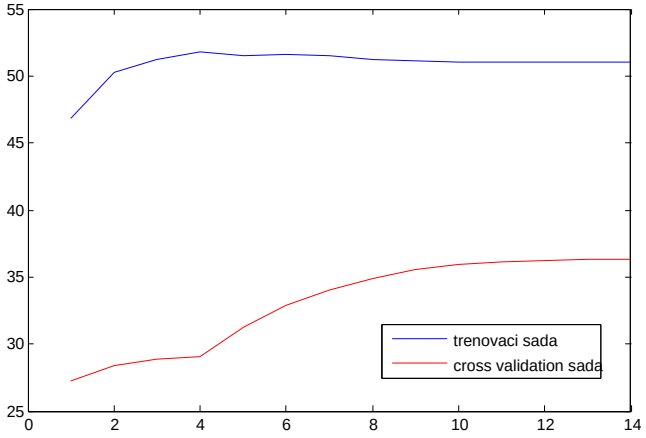
Generativní předtrénování

V první fázi byly takto transformované příznaky použity pro generativní trénování RBM. Bylo natrénováno 6 RBM po 1048 neuronech pomocí contrastive divergence algoritmu. Počáteční hodnoty spojení viditelných a skrytých jednotek RBM byly inicializovány pomocí normálního rozložení se střední hodnotou 0 a rozptylem 0.1.

Diskriminativní trénování pro optimalizaci cross entropie

Po natrénování RBM následovalo samotné diskrimitivní trénování. Celá síť byla inicializována pomocí hodnot získaných při generativním pretrainingu. Při diskriminativním trénování byl použit backpropagation algoritmus a jako objektivní funkce byla použita cross entropie. V průběhu trénování byla vyhodnocována frame accuracy (procento rámců, ke kterým byla přiřazena správná třída) na trénovací i na cross-validační sadě. Trénování bylo zastaveno, jakmile bylo její relativní zlepšení na cross-validační sadě menší než 0.1. Proběhlo celkem 14 iterací, kdy při poslední bylo zlepšení frame accuracy pouze 0.0037. Vývoj frame accuracy v průběhu trénování je ukázán v grafu 4.6.

Na průběh trénování má velký vliv hodnota learning rate. Pokud by byla zvolena příliš nízká, síť by se učila velmi pomalu. Naopak velmi vysoká learning rate může způsobit, že algoritmus nebude vůbec konvergovat. Při trénování neuronové sítě v Kaldi je proto zvolena strategie, kdy počáteční learning rate je ponechána tak dlouho, dokud je relativní zlepšení frame accuracy na cross-validační sadě větší než 0.5. Jakmile toto přestane platit, learning rate se začne v každé epoše trénování zmešovat na polovinu. Počáteční learning rate v tomto případě byla zvolena 0.008.



Obrázek 4.6: Vývoj frame accuracy v průběhu trénování.

Sekvenční trénování pro optimalizaci sMBR

Přestože optimalizace cross entropie použitá při diskriminativním trénování je odůvodněná technika, je možné najít kritéria, které vykazují lepší výsledky. Protože konečným cílem rozpoznávače je co nejmenší WER, je lepší použít více vysokoúrovňová kritéria, které se WER více blíží. Mezi nejčastější patří například Minimum Phone Error (MPE), Maximum Mutual Information (MMI) nebo Minimum Bayes Risk (MBR).

Právě MBR bylo v této práci použito a to na úrovni stavů (state-level MBR neboli sMBR). Optimalizováno bylo tedy

$$\mathcal{F}_{MBR} = \sum_u \frac{\sum_W p(\mathbf{O}_u|S)^{\mathcal{K}} P(W) A(W, W_u)}{\sum_{W'} p(\mathbf{O}_u|S)^{\mathcal{K}} P(W')}, \tag{4.4}$$

kde u značí jednotlivé promluvy (utterance), $A(W, W_u)$ je počet správně rozpoznaných stavů v sekvenci slov W a $\mathcal{K}$ je faktor scalingu akustického modelu. Pro trénování pomocí tohoto kritéria je zapotřebí pro každou promluvu vygenerovat lattici, která obsahuje všechny posloupnosti slov s nezanedbatelnou pravděpodobností, a zarovnání transkripce na sekvenci stavů, která se použije jako referenční sekvence W_u.

Podrobější popis sMBR i dalších kritérií je v článku *Sequence-discriminative training of deep neural networks* [16].

Výsledky

Postupně vytvořené systémy s neuronovými sítěmi na testovací sadě dosáhly následujících výsledků.

Iterace systému	WER [%]	Subs	Ins	Dels
tri5 – trifony + LDA + MLLT + fMLLR	38.0	23.0	2.4	12.6
tri5_pretrain_dbn_dnn	31.5	19.0	2.1	10.4
tri5_pretrain_dbn_dnn_smbr	29.5	18.8	2.1	8.8

Tabulka 4.4: Úspěšnost systémů s neuronovými sítěmi.

4.7 Vyhodnocení úspěšnosti systému

V předchozích sekcích byly uvedeny úspěšnosti jednotlivých iterací systému na testovacích datech. Výsledných 29.5% WER je uspokojivý výsledek srovnatelný s jinými systémy natrénovanými na stejných datech.

Systém byl následně otestován na dvou nahrávkách letecké komunikace. K těmto nahrávkám bohužel nebyly k dispozici transkripce, úspěšnost systému tedy nebylo možné nijak vyčíslit. Délka nahrávek byla 24:59 minut a 25:24 minut. Systém tyto nahrávky nerozpoznal příliš úspěšně. Na vině je to, že pro trénování nebyly k dispozici žádná data letecké komunikace a také se na testovacích nahrávkách vyskytovalo velké množství němčiny, na což nebyl model připraven.

Pro lepší výsledek dekódování těchto nahrávek, byly vytvořeny dvě alternativní verze jazykového modelu, které zvýhodňovaly často používaná slova v letecké komunikaci.

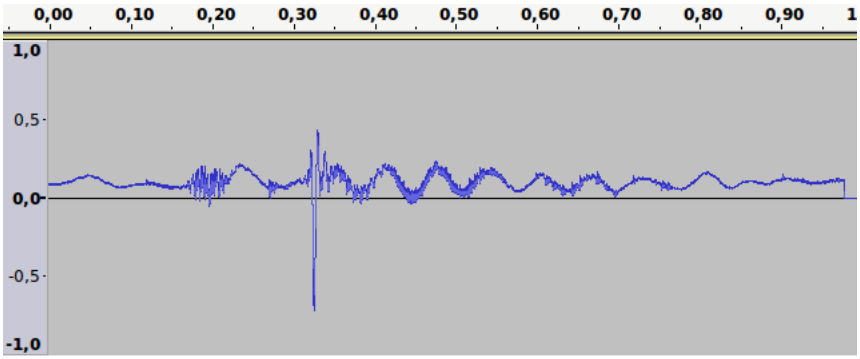
- V první verzi byl jazykový model převeden na unigramový a zvýšila pravděpodobnosti slov často se vyskytujících v letecké komunikaci. Jsou to především číslovky, dále také kódová slova, která se používají pro hláskování (např. alfa, beta, charlie, ...[6]) a další související výrazy (např. departure, tower, approach). Tato verze jazykového modelu sloužila spíše pro rychlé otestování, jestli akustický model s jeho pomocí dokáže tyto výrazy rozpoznat.

- Při další verzi byl využit soubor textových přepisů letecké komunikace, který obsahoval 1300 krátkých promluv o průměrně 10 slovech. Z tohoto textu byl pomocí

[6]ICAO abeceda http://legacy.icao.int/icao/en/trivia/alphabet.htm

nástrojů z *The SRI Language Modeling Toolkit*[7] vytvořen bigramový jazykový model a interpolovala jej s původním jazykovým modelem. Toto řešení dosahovalo nejlepších výsledků.

Úprava jazykového modelu rozpoznání nahrávek zlepšila, přesto úspěšnost stále není ideální. Na obrázku 4.7 je ukázka části nahrávky, která byla po úpravě modelu úspěšně rozpoznána.



1. REQUEST | START | UP
2. BECAUSE | STUFF
3. REQUEST | UP
4. REQUEST | START | UP

Obrázek 4.7: Ukázka rozpoznání části nahrávky.
1. je skutečná vyslovená posloupnost slov.
2. rozpoznaná posloupnost neupraveným systémem.
3. rozpoznaná posloupnost s modelem se zvýšenou pravděpodobností leteckých slov.
4. rozpoznaná posloupnost systémem s interpolovaným jazykovým modelem.

[7]The SRI Language Modeling Toolkit http://www.speech.sri.com/projects/srilm/

Kapitola 5

Možnosti rozšíření rozpoznávače

V rámci této práce jsme experimentovali s několika cestami, jak rozpoznávač zpřesnit či urychlit. Protože se práce zaměřovala na neuronové sítě, bylo experimentováno právě s nimi. O provedení a výsledcích těchto experimentů bude pojednávat tato kapitola.

5.1 Zjednodušení neuronových sítí

Prvním provedením experimentem bylo prozkoumání možnosti zjednodušení a zrychlení neuronové sítě a jeho vlivu na zhoršení přesnosti rozpoznávání. Testovaná neuronová síť měla 4 vrstvy o 1024 neuronech, byla diskriminativně natrénovaná a jejím vstupem byly FBank příznaky. Porovnání bylo provedeno s neuronovou sítí s 6 vrstvami s 2048 neurony, která byla natrénována ve dvou fázích – generativní pretraining a diskriminativní trénování pro optimalizaci cross etropie. Příznaky na vstupu byly MFCC transformováné pomocí fMLLR.

Díky FBank příznakům je možné provést pouze jedno dekódování, protože není potřeba počítat fMLLR transformace. Menší síť zase umožní rychlejší propagaci dat.

Obě sítě byly natrénovány a otestovány na datové sadě Switchboard.

	trénovací data	testovací data
počet mluvčích	4870	80
počet promluv	262509	4447
počet hodin	cca 317	cca 3.6
průměrná délka promluvy	cca 4.3 sekundy	cca 2.9 sekundy

Tabulka 5.1: Vlastnosti datové sady Switchboard.

Úspěšnost a rychlost rozpoznávání testovacích dat je uvedena v následující tabulce

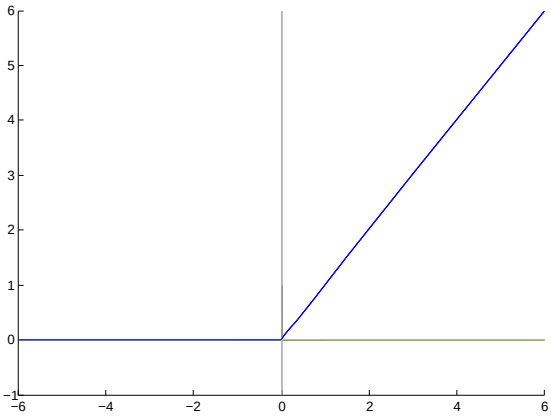
	WER[%]	rychlost dekódování [s]
dnn_4L_1024_FBANK	24.5	22308
dnn_6L_2048_pretrain_MFCC_fmllr	21.3	15377 + 31778

Tabulka 5.2: Výsledky rozpoznávání zjednodušené neuronové sítě na sadě Switchboard. Rychlost dekódování u složitější neuronové sítě se skládá z dekódování GMM pro získání fMLLR transformací a samotného dekódování pomocí DNN.

5.2 Aktivační funkce rectifier

Dalším experimentem bylo nahrazení logistické sigmoidy aktivační funkcí rectifier, která je definována jako

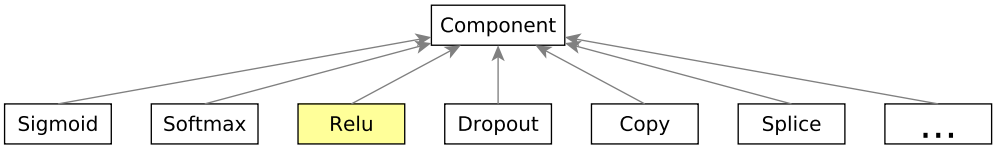
$$f(x) = \max(0, x). \tag{5.1}$$



Obrázek 5.1: Graf průběhu rectifier funkce.

Tento experiment vycházel z článku *Deep Sparse Rectifier Neural Networks* [5], kde byly s touto aktivační funkcí dosáhnuty dobré výsledky v oblasti rozpoznávání obrazu a klasifikaci textů.

V Kaldi jsou aktivační funkce potomci třídy `Component`, která reprezentuje jednotlivé části neuronové sítě. Pro vyzkoušení rectifier funkce, bylo tedy třeba implementovat novou třídu `Relu` a její metody `PropagateFnc` a `BackpropagateFnc`, které obsahují výpočet výstupů vrstvy při propagaci a výpočet chyby při zpětné propagaci.



Obrázek 5.2: Třída Component.

Experimenty byly nejprve prováděny na datové sadě RM. Jedná se o menší sadu, obsahující čistou řeč, konkrétně příkazy imaginárnímu systému. Informace o datové sadě jsou uvedeny v tabulce 5.3.

	trénovací data	testovací data
počet mluvčích	109	59
počet promluv	3990	1460
počet hodin	cca 3.7	cca 1.3
průměrná délka promluvy	cca 3.4 sekundy	cca 3.3 sekundy

Tabulka 5.3: Vlastnosti datové sady RM.

Natrénovaná neuronová síť obsahovala tři skryté vrstvy s 500 neurony. Tato síť byla natrénována jak s rectifier tak i se sigmoid jednotkami. U rectifier jednotek bylo nutné výrazně snížit learning rate z 0.008 na 0.001, trénování sítě jinak nekonvergovalo.

Aktivační funkce rectifier přinesla srovnatelný výsledek s logistickou sigmoidou, což odpovídá výsledklm pozorovaným v jiných laboratořích (Microsoft, JHU).

Aktivační funkce	WER [%]
sigmoida	1.84
rectifier	1.87

Tabulka 5.4: Porovnání úspěšnosti aktivačních funkcí sigmoida a rectifier.

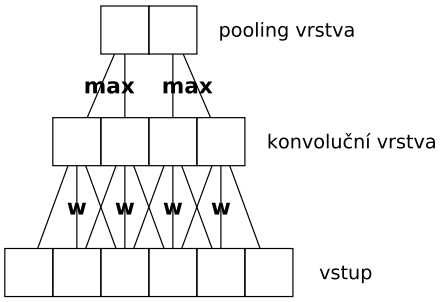
5.3 Konvoluční neuronové sítě

Tato část vychází především z článků *Improvements to deep convolutional neural networks for LVCSR* [13], *Deep convolutional neural networks for LVCSR* [14] a *Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition* [1], kde je popsána aplikace konvolučních neuronových sítí na rozpoznávání řeči a provedeny úspěšné experimenty. Obecný teoretický úvod do konvolučních neuronových sítí uvádí i Bishop [2].

Konvoluční neuronové sítě jsou alternativní typ neuronových sítí, které se často používají v rozpoznávání obrazu a v nedávné době s nimi byly zaznamenány úspěchy v rozpoznávání řeči. Díky jejich architektuře se dokáží přizpůsobit drobným změnům ve spektru ve směru frekvenční osy, která v řečových signálech vzniká díky různým stylům projevu a rozdíly mezi mluvčími. Tuto variabilitu dokáží modelovat i klasické neuronové sítě, ovšem pomocí mnohem více parametrů.

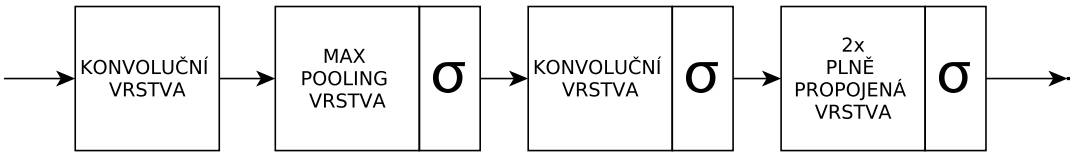
Konvoluční neuronová síť se skládá z jedné nebo více konvolučních a tzv. pooling vrstev. Neurony v konvoluční vrstvě nejsou narozdíl od plně propojených vrstev spojeny se všemi neurony předchozí vrstvy. Místo toho každý neuron zpracovává pouze malý lokální vstup. Váhy jednotlivých neuronů jsou pak sdíleny přes celý vstupní prostor. Neurony tak detekují lokální vzory v datech.

Za konvoluční vrstvou následuje pooling vrstva, která vždy slučuje výstupy několika neuronů do jednoho a snižuje tak dimenzionalitu výstupu. Nejčasteji používaný je tzv. max-pooling, který vybírá ze vstupů ten s největší hodnotu. Další možností je vypočítat ze vstupů průměr. Díky poolingu jsou konvoluční neuronové sítě invariantní vůči malým změnám ve spektru.



Obrázek 5.3: Konvoluční neuronová síť.

V rámci této práce bylo experimentováno s parametry konvolučních sítí a vytvořila skripty pro jejich sestavení pomocí Kaldi. Počáteční architektura konvolučních sítí vychází z článku [14]. Obsahuje dvě konvoluční vrstvy, mezi nimiž je jedna pooling vrstva a následně 2 plně propojené vrstvy po 1024 neuronech. První konvoluční vrstva se skládá ze 128 filtrů, druhá obsahuje 256 filtrů. Neurony pooling vrstvy mají na vstup napojené vždy tři neurony z vrstvy minulé a jejich vstupy se nepřekrývají. Jako aktivační funkce v celé síti je použita logistická sigmoida. Příznaky na vstupu sítě jsou FBANK, společně s jejich prvními a druhými derivacemi. Celá architektura je zobrazena na obrázku 5.4.



Obrázek 5.4: Architektura konvoluční neuronová sítě.

Všechny experimenty probíhaly na datové sadě z projektu BABEL pro jazyk Zulu. Parametry datové sady jsou uvedeny v tabulce.

	trénovací data	testovací data
počet mluvčích	120	119
počet promluv	10429	14092
počet hodin	cca 10	cca 10
průměrná délka promluvy	cca 3.5 sekundy	cca 2.5 sekundy

Tabulka 5.5: Vlastnosti datové sady jazyka Zulu.

Jako baseline, se kterou byly výsledky porovnány, byla plně propojená síť natrénovaná pomocí generativního pretrainingu a následně diskriminativního tréninku s 6 skrytými vrstvami po 2048 neuronech.

	WER[%]
baseline dnn_FBANK_pretrain_dbn_dnn	71.1

Tabulka 5.6: Baseline systém pro porovnání úspěšnosti s konvoluční neuronovou sítí.

Zpracování F0

První experimenty se zabývaly vyřešením problému, jak v konvoluční síti zpracovávat základní tón neboli příznak f0. Příznak f0 bývá často přidáván k ostatním příznakům a zvyšuje úspěšnost rozpoznávání řeči. Protože se nejedná o spektrální příznak, nemá smysl přes něj provádět konvoluci. Bylo vyzkoušeno několik způsobů jak f0 přidat do systému.

1. Prvním nejjednodušším způsobem bylo začít pracovat s f0 až za konvolučními vrstvami, tedy připojit je k jejich výstupu. Tento postup sice vykazoval dobré výsledky, po přidání pretraingu, o kterém bude řeč dále, se ale ukázal být nefunkčním. RBM totiž předpokládá na vstupu Bernoulliho uzly, které jsou i na výstupu konvolučních vrstev. F0 ale odpovídá uzlům Gaussovským, jejich konkatenace tedy není vhodná.

2. Dalším způsobem bylo přidat dvě plně propojené vrstvy s 200 neurony, které F0 ztransformují na příznaky s Bernoulliho rozložením. Výstupy těchto vrstev jsou pak připojeny k výstupům konvolučních vrstev stejně jako u bodu 1. Tento postup se ukázal být nejvhodnějším.

3. Ke vstupním příznakům byly připojovány první a druhé derivace (neboli delta příznaky). Další test ověřil, jestli má smysl delty používat i u F0 příznaku. Zkusila jsem tedy delty F0 odstranit.

4. Poslední experiment byl kombinací dvou předchozích. F0 opět procházely dvěma plně propojenými vrstvami, ovšem bez delt.

	1.	**2.**	3.	4.
WER [%]	72.2	**72.2**	72.4	72.4

Tabulka 5.7: Výsledky experimentů se zpracováním příznaků F0.

Přidáním dvou plně propojených vrstev se skóre nijak nezměnilo, ale jak již bylo zmíněno v dalších fázích se ukázalo být jako nejlepší řešení. Odstranění delta příznaků způsobilo úbytek -0.2% WER, nebylo tedy použito.

Parametry sítě

Další experimenty měly za cíl najít optimální nastavení parametrů sítě. Experimenty probíhaly s počtem plně propojených vrstev, počty filtrů, velikostí a typem poolingu.

Počet plně propojených vrstev	**2**	3	4	6
WER [%]	**72.2**	72.2	72.2	72.3

Tabulka 5.8: Výsledky experimentů s počtem propojených vrstev.

Zvětšování počtu plně propojených vrstev nevedlo ke zlepšení. U velkého počtu vrstev se výsledek nepatrně zhoršil.

Počet filtrů v 1./2. konvoluční vrstvě	128/128	**128/256**	256/256	256/512	512/512
WER [%]	72.8	**72.2**	72.4	72.2	72.3

Tabulka 5.9: Výsledky experimentů s počtem filtrů.

Zvyšování počtu filtrů tedy také nevedlo ke zlepšení.

Velikost poolingu	**3/3**	4/4	5/5	6/3	6/6	8/4
WER [%]	**72.2**	72.4	72.5	72.6	72.4	72.9

Tabulka 5.10: Výsledky experimentů s velikostí poolingu.

První číslo vždy značí samotnou velikost poolingu, tedy kolik neuronů poslední vrstvy je napojeno na jeden neuron pooling vrstvy. Číslo za lomítkem udává počet neuronů, o který jsou mezi sebou jednotlivé skupiny posunuty. Pokud se tedy tyto dvě čísla liší, pooling

Typ poolingu	**Max**	Avg
WER [%]	**72.2**	72.7

Tabulka 5.11: Výsledky experimentů s typem poolingu.

se překrývá. Zvětšování poolingu opět nevedlo k lepším výsledkům, zvlášť experimenty s překrýváním vedly ke zhoršení.

Použití průměru u poolingu místo výběru maximální hodnoty také vedlo k horšímu výsledku.

Pomocí experimentů s parametry sítě se mi nepodařilo dosáhnout zlepšení, což bylo pravděpodobně způsobeno tím, že počáteční architektura sítě navrhnutá článkem [14] byla již dobře zvolena.

Pretraining

Po experimetování s parametry sítě byl nejlepší výsledek stále 72.2% WER, což je o 1% horší než plně propojená síť se stejným nastavením. Dalším krokem, který mohl vést ke zlepšení byl generativní pretraining plně propojených vrstev stejně jako u klasické neuronové sítě. Celý postup se tedy rozšířil na tři části:

1. diskriminativní natrénování konvoluční sítě, tak jako bylo prováděno doteď
2. generativní pretraining plně propojené části
3. diskriminativní trénování celé sítě

Před generativním pretrainingem se tedy odtrhly dříve natrénované plně propojené části a konvoluční část se použila jako tranformace příznaků na vstupu RBM. Pretraining znovu otevřel cestu k použití většího množství plně propojených vrstev, nejlepší finální konfigurace měla plně propojených vrstev 6.

	WER [%]
Baseline DNN	71.1
CNN bez pretrainingu	72.2
CNN s pretrainingem	**70.7**

Tabulka 5.12: Výsledky experimentů s pretrainingem konvolučních neuronových sítí.

Ve výsledku se tedy podařilo překonat klasickou plně propojenou neuronovou síť o 0.4% WER. Výsledná konfigurace byla také otestována na sadě RM, kde bylo také dosáhnuto zlepšení.

	WER [%]
Baseline DNN	1.82
CNN s pretrainingem	**1.77**

Tabulka 5.13: Výsledky experimentů s pretrainingem konvolučních neuronových sítí.

Kapitola 6

Závěr

V této bakalalářské práci jsem se zabývala vytvořením systému rozpoznávání řeči založeném na neuronových sítích. V práci jsem popsala postup vytvoření tohoto systému a jeho úspěšnost na testovacích datech. Dosažená úspěšnost 29.5% WER je srovnatelná se state-of-the-art systémy.

Vytvořený akustický model bude použit v projektu A-PiMod, který se zabývá leteckou bezpečností, systém jsem proto otestovala i na nahrávkách letecké komunikace. Úspěšnost tohoto testování bohužel nebyla vyčíslitelná kvůli nedostatku dat s transkripcemi. Systém byl pouze odzkoušen na dvou nahrávkách a pro účely rozpoznání těchto nahrávek jsem upravila jazykový model celého systému.

V další části práce jsem provedla experimenty s neuronovými sítěmi. V prvním experimentu jsem zkoumala vliv zjednodušení a zrychlení neuronových sítí na úspěšnost rozpoznávání. Dekódování se díky zjednodušení neuronové sítě zrychlilo více než 2x, přičemž WER na testovacích datech se zvýšila o 3%. Dalším experimentem bylo použití aktivační funkce rectifier, se kterou jsem na datové sadě RM dosáhla WER srovnatelné s aktivační funkcí sigmoida.

Další experimenty proběhly s konvolučními neuronovými sítěmi. Vyzkoušela jsem několik způsobů jak s nimi zpracovávat příznak F0. Experimentovala jsem také s jejich parametry a přidala jsem pretraining, což vedlo k jejich zlepšení o 1.5% WER. Ve výsledku tedy konvoluční neuronové sítě dosáhly o 0.4% WER lepšího výsledku než DNN se stejnou architekturou.

Literatura

[1] Abdel-Hamid, O.; Mohamed, A.; Jiang, H.; aj.: Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, ISSN 1520-6149, s. 4277–4280, doi:10.1109/ICASSP.2012.6288864.

[2] Bishop, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006, ISBN 0387310738.

[3] Gales, M. J.: Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech & language*, ročník 12, č. 2, 1998: s. 75–98.

[4] Gales, M. J. F.: Semi-tied covariance matrices for hidden Markov models. *Speech and Audio Processing, IEEE Transactions on*, ročník 7, č. 3, May 1999: s. 272–281, ISSN 1063-6676, doi:10.1109/89.759034.

[5] Glorot, X.; Bordes, A.; Bengio, Y.: Deep Sparse Rectifier Neural Networks. In *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, Duben 2011.

[6] Hermansky, H.: Perceptual Linear Predictive (PLP) Analysis of Speech. *J. Acoust. Soc. Am.*, ročník 57, č. 4, Duben 1990: s. 1738–52.

[7] Hinton, G.; Deng, L.; Yu, D.; aj.: Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Signal Processing Magazine*, 2012.

[8] Huang, X.; Acero, A.; Hon, H.: *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, 2001, ISBN 9780130226167.
URL http://books.google.cz/books?id=reZQAAAAMAAJ

[9] Mohamed, A.; Dahl, G.; Hinton, G.: Acoustic Modeling Using Deep Belief Networks. *Audio, Speech, and Language Processing, IEEE Transactions on*, ročník 20, č. 1, jan. 2012: s. 14 –22, ISSN 1558-7916, doi:10.1109/TASL.2011.2109382.

[10] Mohri, M.; Pereira, F.; Riley, M.: Speech Recognition with Weighted Finite-State Transducers. In *Springer Handbook of Speech Processing*, editace J. Benesty; M. Sondhi; Y. Huang, Springer Berlin Heidelberg, 2008, ISBN 978-3-540-49125-5, s. 559–584, doi:10.1007/978-3-540-49127-9˙28.
URL http://dx.doi.org/10.1007/978-3-540-49127-9_28

[11] Povey, D.; Ghoshal, A.; Boulianne, G.; aj.: The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*,

Idiap-RR-04-2012, Rue Marconi 19, Martigny: IEEE Signal Processing Society, Prosinec 2011, ISBN 978-1-4673-0366-8, iEEE Catalog No.: CFP11SRW-USB.

[12] Psutka, J.; Müller, L.; Matoušek, J.; aj.: *Mluvíme s počítačem česky*. Prague: Academia, 2006, ISBN 80-200-1309-1, 752 s.
URL http://www.kky.zcu.cz/en/publications/PsutkaJ_2006_Mluvimes

[13] Sainath, T. N.; Kingsbury, B.; rahman Mohamed, A.; aj.: Improvements to deep convolutional neural networks for LVCSR. *CoRR*, ročník abs/1309.1501, 2013.

[14] Sainath, T. N.; rahman Mohamed, A.; Kingsbury, B.; aj.: Deep convolutional neural networks for LVCSR. In *ICASSP*, IEEE, 2013, s. 8614–8618.
URL
http://dblp.uni-trier.de/db/conf/icassp/icassp2013.html#SainathMKR13

[15] Veselý, K.: *Paralelní trénování neuronových sítí pro rozpoznávání řeči*. Diplomová práce, Vysoké učení technické v Brně, 2010.

[16] Veselý, K.; Ghoshal, A.; Burget, L.; aj.: Sequence-discriminative training of deep neural networks. In *INTERSPEECH*, August 2013.
URL
http://www.cstr.ed.ac.uk/downloads/publications/2013/is13-dnn_seq.pdf

[17] Young, S. J.; Evermann, G.; Gales, M. J. F.; aj.: *The HTK Book, version 3.4*. Cambridge, UK: Cambridge University Engineering Department, 2006.

Příloha A

Obsah CD

Přiložené CD obsahuje

- natrénované modely
- použité skripty
- výsledky dekódování a skórování
- soubor README, ve kterém je podrobě popsána struktura CD