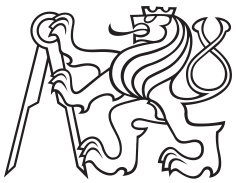
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Detektor řečové aktivity s pokročilými strukturami neuronových sítí

David Machát

Vedoucí: doc. Ing. Petr Pollák, CSc.
Obor: Komunikace a zpracování signálu
Studijní program: Otevřené Elektronické Systémy
Srpen 2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Machát** Jméno: **David** Osobní číslo: **434970**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Otevřené elektronické systémy**

Studijní obor: **Komunikace a zpracování signálu**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Detektor řečové aktivity s pokročilými strukturami neuronových sítí

Název diplomové práce anglicky:

Voice Activity Detector based on Neural Networks with Advanced Structures

Pokyny pro vypracování:

1. Seznamte se s problematikou použití neuronových sítí v úlohách zpracování řečového signálu.
2. Navrhněte detektor řečové aktivity realizovaný pomocí neuronové sítě. Zvažte použití různých struktur neuronových sítí, zejména DNN v různých variantách počtu vrstev či vstupních parametrů resp. CNN (konvoluční neuronové sítě).
3. Navržený detektor implementujte pomocí volně dostupných nástrojových sad, zejména KALDI (sada nástrojů pro implementaci rozpoznávání řeči).
4. V experimentální části navržené detektory srovnejte na datech z dostupných řečových databází. Srovnejte funkčnost za různých akustických podmínek, tj. pro různé typy a úrovně šumu prostředí.

Seznam doporučené literatury:

[1] J. Psutka, L. Müller, J. Matoušek, V. Radová. Mluvíme s počítačem česky. Academia 2006.
[2] X. Huang, A. Acero, H.-W. Hon. Spoken Language Processing. Prentice Hall, 2001.
[3] D. Yu, L. Deng. Automatic Speech Recognition A Deep Learning Approach. Springer-Verlag London. 2015
[4] D. Povey et al, The Kaldi Speech Recognition Toolkit. In Proc. of IEEE 2011 ASRU, Hawai, US, 2011.
[5] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza. A deep neural network approach for voice activity detection in multi-room domestic scenarios. In Proc. of IJCNN, Killarney, Ireland, Jul. 12-17 2015, pp. 1-8.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Petr Pollák, CSc., katedra teorie obvodů FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **24.01.2020** Termín odevzdání diplomové práce: **14.08.2020**

Platnost zadání diplomové práce: **30.09.2021**

doc. Ing. Petr Pollák, CSc. podpis vedoucí(ho) práce	doc. Ing. Josef Dobeš, CSc. podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu diplomové práce doc. Ing. Petru Pollákovi, CSc. za věnovaný čas a jeho cenné rady a připomínky.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací

V Praze dne 13. srpna 2020

I declare that I have completed the presented thesis independently and that I have mentioned all used sources in accordance with the metholodogical instruction on ethical principles in academic theses.

In Prague, 13. srpna 2020

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací detektoru řečové aktivity za pomoci hlubokých neuronových sítí a konvolučních neuronových sítí. V teoretické části se práce zaměřuje na shrnutí základních poznatků z oblastí zpracování řečového signálu a strojového učení. Dále obsahuje přehled využití metod strojového učení v oblasti automatického rozpoznávání řeči. Experimentální část práce obsahuje návrh detektoru řečové aktivity za pomoci konvoluční neuronové sítě a jeho implementaci pomocí sady nástrojů Kaldi. Dále je zkoumán vliv změn parametrů sítě a vstupních dat na efektivitu detektoru. Na závěr je zkoumána funkčnost detektoru při práci s různými typy a úrovněmi šumu nad databázemi TIMIT a QUT-TIMIT.

Klíčová slova: konvoluční neuronové sítě, automatické rozpoznávání řeči, detekce řečové aktivity, Kaldi

Vedoucí: doc. Ing. Petr Pollák, CSc.

Abstract

This thesis deals with the realization of voice activity detector based on deep neural networks and convolutional neural networks. Theoretical part focuses on general overview of speech processing and machine learning. There is also a survey of applications of machine learning in automatic speech recognition. Experimental part contains proposed voice activity detector based on convolutional neural networks and its implementation in Kaldi toolkit. Effect of changing network parameters and input data on the effectivity of the detector is examined. Effectivity of proposed detector is also evaluated based on data with various types and levels of noise from TIMIT and QUT-TIMIT databases.

Keywords: convolutional neural networks, automatic speech recognition, voice activity detection, Kaldi

Title translation: Voice Activity Detector based on Neural Networks with Advanced Structures

Obsah

Obrázky

Tabulky

Seznam zkratek

ASR - Automatic Speech Recognition
HHM - Hidden Markov Model
LMFB - Log Mel-Filter Bank
MFCC - Mel Frequency Cepstral Coefficients
VAD - Voice Activity Detection
GMM - Gaussian Mixture Model
ANN - Artificial Neural Network
DNN - Deep Neural Network
CNN - Convolutional Neural network
MLP - Multilayer Perceptron
E2E - End-to-End
ERR - ERRor decision
ERS - ERror in Speech
ERP - ERror in Pause

Kapitola 1

Úvod

Snaha naučit počítače komunikovat s lidmi formou komunikace, která je lidem vlastní se v dnešní, na elektronice tolik závislé době, jeví jako přirozený směr vývoje moderních technologií.

Systémy pro automatické rozpoznávání řeči (Automatic Speech Recognition - ASR) jsou technologiemi, se kterými se dnes mnozí každý den setkávají. Nejčastěji ve formě hlasového ovládání elektronických zařízení nebo u hlasových asistentů, jakými jsou Siri, Alexa nebo Cortana. Historicky prvním přístupem k úlohám rozpoznávání řeči byla snaha naučit počítač správně určit povel z omezené množiny slov na základě formantových kmitočtů. Tento přístup ovšem nalezl uplatnění pouze v několika velmi specifických aplikacích. Velkým krokem pro evoluci ASR systémů byla popularizace skrytých markovových modelů (Hidden Markov Model - HMM) v osmdesátých letech minulého století, která vedla k novému přístupu k modelování řeči, založenému především na statistických modelech. Příkladem takového statistického modelu je například n-gram model předpovídající nadcházející slovo na základě n-1 předcházejících slov. Další vývoj ASR systémů pak směřoval k využití umělé inteligence. Spolu s HMM se používaly umělé neuronové sítě. V posledních dvaceti letech pak bylo dosaženo dostatečného výpočetního výkonu pro využití metod hlubokého učení. Tyto metody také umožnily oddělit mluvčího od promluvy a odpadla tak nutnost přizpůsobovat systémy specifickým mluvčím. Moderní ASR systémy pak pracují především právě na bázi hlubokých neuronových sítí.

Strojové učení nalezlo využití v mnoha oblastech vědy a výzkumu, ASR nevyjímaje. Tato práce je zaměřena na návrh a realizaci detektoru řečové aktivity. Cílem tohoto detektoru bude rozpoznat časové úseky, ve kterých je mluvčí aktivní a oddělit je od úseků, ve kterých je ticho, nebo hluk v pozadí. Znalost, kdy je uživatel aktivní, nám umožňuje lépe komprimovat data v kodecích řeči, určit začátek a konec promluvy při rozpoznávání řeči, automaticky segmentovat dlouhé záznamy, nebo rozlišit hluk pozadí od povelů

pro rozhraní ovládaná hlasem. Detektory řečové aktivity mohou pracovat s různými druhy akustických příznaků. V této práci se zaměříme především na detektory používající Mel-frekvenční kepstrální koeficienty a banky filtrů.

Kapitola 2

Řečový signál a jeho charakteristiky

První část této práce popisuje vznik řečového signálu v lidském těle a vlastnosti řečového signálu důležité pro zpracování lidské řeči a její následné využití v ASR systémech. Z fyzikální hlediska je lidský hlas mechanickým vlněním s omezeným frekvenčním rozsahem vznikajícím při průchodu vzduchu vytlačeného z plic vokálním traktem. Lidským sluchem zase popisujeme proces příjmu tohoto vlnění a jeho interpretaci lidským mozkem.

2.1 vznik řečového signálu

V lidském těle je řeč vytvářena hlasovým traktem. Ten je tvořen dechovým ústrojím, hlasovým ústrojím a artikulačním ústrojím [1]. Proces tvorby řeči začíná v plicích. Proud vzduchu vytlačený z plic prochází průdušnicí (trachea) do hrtanu, ve kterém se nachází hlasové ústrojí. Proud vzduchu rozkmitává hlasivky a vytváří tak základní hlas o základní frekvenci f_0. Hodnoty základní frekvence se nejčastěji pohybují mezi 60-400 Hz. Konkrétní rozsahy se různí podle pohlaví a věku. U dětí nabývá výška hlasu nejvyšších hodnot. S postupujícím věkem dochází k mutování a výška hlasu se tak snižuje. U mužů nabývá základní frekvence nižších hodnot než u žen. Základní hlas poté postupuje dále do artikulačního ústrojí, jenž je tvořeno rezonančními dutinami, které transformuje základní hlas na lidskou řeč. Dutina hrdelní se nachází přímo nad hlasivkami a její vliv na tvorbu řeči je řízen pohybem jazyka a činností krčních svalů. Funkce dutiny nosní spočívá ve tvorbě nazálních hlásek. Při tvorbě jiných druhů hlásek bývá uzavřena měkkým patrem (velum). Třetí

dutinou je dutina ústní, ve které je tvorba řeči ovlivňována pohybem jazyka, pohybem rtů a pohybem čelistí.

2.2 Obecné charakteristiky řeči

Lidský hlas obecně klasifikujeme jako kvaziperiodický signál [1]. Jednotlivé hlásky mohou mít kvaziperiodický (znělé), nebo neperiodický (neznělé) charakter. Typický model řeči předpokládá změny charakteristik řečového signálu každých 50 až 100 ms. Abychom mohli pracovat s takto proměnným signálem, je třeba využívat metody krátkodobé analýzy. Signál je rozdělen na mikrosegmenty, typicky o délce 10 ms, ve kterých je možné považovat signál za stacionární. Před výpočtem samotných příznaků je třeba ještě aplikovat okenní funkci. Nejčastěji se při práci s hlasovým signálem používá Hammingovo nebo Hannovo okno. Výhodou použití těchto druhů oken jsou jejich vlastnosti, konkrétně tlumení vyšších, ne tolik zajímavých frekvencí.

2.3 Číslicová reprezentace řeči

Pro zpracování řečového signálu je nutné jej digitalizovat za účelem snadnější manipulace a komprese. Prvním krokem při digitalizaci hlasového signálu je vzorkování. Vzorkovací frekvence volíme podle použitého druhu média. V telefonii se pro přenos lidského hlasu využívá kmitočtů do 3 kHz. Pro tyto systémy je volena vzorkovací frekvence 8 kHz. Dále lze ve specifických případech využít i frekvence jako 16 kHz, 44.1 kHz nebo 48 kHz, které nesou více detailů. Frekvence 8 kHz je nejmenší běžně používanou vzorkovací frekvencí, u které je zajištěná srozumitelnost jednotlivých hlásek a dostatek informací pro určení základních hlasových charakteristik mluvčího [2]. Vzorkovací frekvence 16 kHz se využívá například v komunikacích využívajících VoIP (Voice over Internet Protocol). Na této frekvenci dochází k přenosu většiny informace obsahované v lidském hlase a je tedy vhodnější pro práci s charakteristikami řeči jednotlivých mluvčích.
Druhým krokem při digitalizaci řeči je kvantizace. Kvantizace pro práci s hlasovým signálem by měla efektivně pokrýt dynamiku řeči, která se běžně pohybuje okolo 50 dB [2]. Nejběžněji používanou metodou je 16 bitová pulse code modulation (PCM) s lineárním kvantováním, která je schopná pokrýt rozsah 90 dB.

2.3.1 Řečové příznaky

V této části jsou popsány vybrané příznaky využívané v různých aplikacích rozpoznávání řeči. Analýza řečového signálu nabízí mnoho druhů možných příznaků pro různé aplikace rozpoznávání řeči. Následující výčet popisuje příznaky, které jsou využity v této práci a několik dalších, nejrozšířeněji používaných příznaků.

- Energie
  Jedním z nejrozšířenějších příznaků je energie signálu. Velikou výhodou tohoto příznaku je jeho malá výpočetní náročnost. Mezi nevýhodami tohoto přístupu je zase jeho malá robustnost vůči šumu. Definice energie pro signál $x[n]$ je následující

$$E = \sum_{n=1}^{N} x^2[n] \tag{2.1}$$

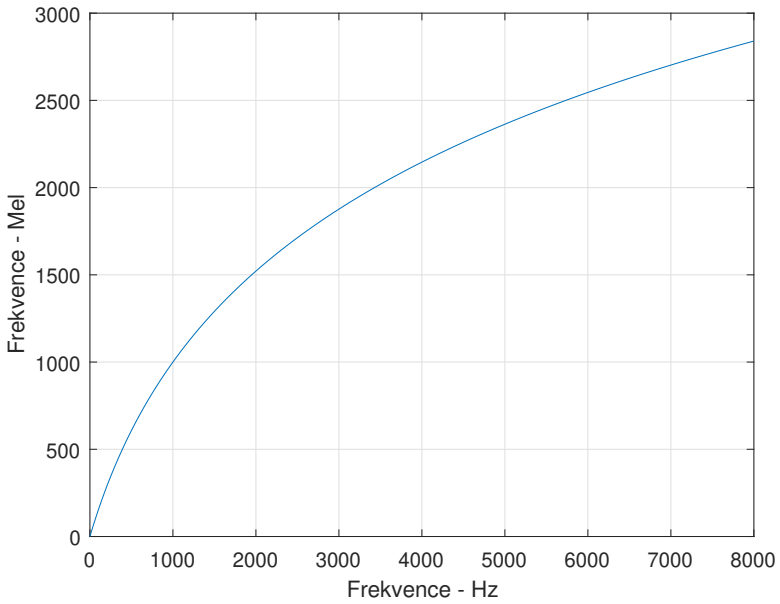
  kde $x[n]$ je n-tý vzorek signálu x.

- Spektrální koeficienty
  Dalším užitečným příznakem řečového signálu je jeho spektrum, vypočítané za pomoci krátkodobé Fourierovy transformace. Frekvenční rozlišení závisí počtu vzorků, tedy na zvolené délce časového segmentu a vzorkovací frekvenci. Její definice je následující:

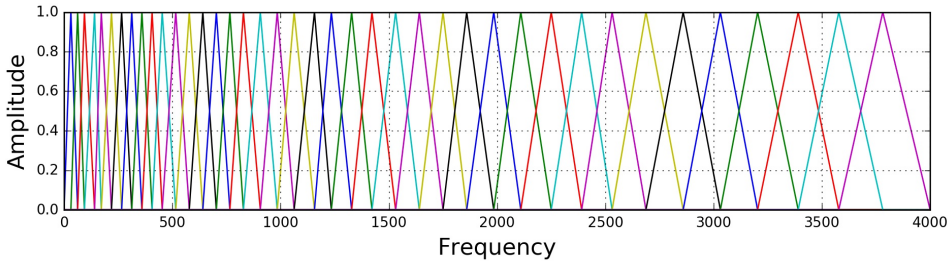
$$\mathcal{X}[k] = \sum_{x=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \tag{2.2}$$

  kde $x[n]$ je n-tý vzorek signálu x a N je délka okna se kterým se pracuje.

- Log mel banka filtrů
  Melovská banka filtrů (LMFB) je banka trojúhelníkových filtrů rovnoměrně rozmístěných na melovské frekvenční stupnici. Důvodem pro využití melovské stupnice je to, že lépe vystihuje nelineární lidské vnímání zvuku. Filtry jsou hustěji zastoupené na nižších frekvencích, kde je lidský sluch citlivější. Pro snazší manipulaci při zpracování jsou hodnoty logaritmovány.



Obrázek 2.1: Melovská frekvenční stupnice



Obrázek 2.2: Melovská Banka filtrů [3]

- Kepstrální koeficienty
  Kepstrum je výstupem inverzní fourierovy transformace logaritmu spektra signálu. Kepstrální koeficienty lze vypočítat jako:

$$c[n] = \mathcal{IDFT}\{\ln(\mathcal{DFT}\{x[n]\})\} \tag{2.3}$$

  Koeficienty reálného kepstra $c[n]$ nesou informace o tvaru spektra a o charakteru buzení. První část kepstra reprezentuje spektrální obálku (vyhlazené spektrum) signálu a druhá nese informace o periodicitě. Při práci s detektory řečové aktivity se často používá pouze omezený počet kepstrálních koeficientů nesoucích pouze informace o tvaru amplitudového spektra.
  Běžnou modifikaci výpočtu kepstra je použití nelineární melovské frekvenční stupnice, kvůli její schopnosti lépe modelovat lidské vnímání zvuku. Spektrální koeficienty jsou filtrovány melovskou bankou trojúhelníkových filtrů. Logaritmizací a následnou inverzní kosinovou transformací lze získat mel-frekvenční kepstrální koeficienty (Mel-frequency cepstral coefficients - MFCC).

2.3.2 Delta příznaky

Kvůli rychle se měnícím charakteristikám se občas ve zpracování řeči používají krom statických parametrů i parametry dynamické, nazývané Δ parametry. Jde o odhady první derivace statických příznaků [2].

$$\Delta_k[n] = \frac{\sum_{m=1}^{M} m(c_k[n+m] - c_k[n-m])}{\sum_{m=1}^{M} m^2} \tag{2.4}$$

kde c_k jsou příznaky segmentů sousedících se současným segmentem n a M je počet segmentů, přes které jsou příznaky z obou stran počítány. Dále se občas využívají i delta-delta příznaky, značené jako $\Delta\Delta$. Výpočet těchto parametrů je obdobný jako u delta příznaků, s využitím Δ příznaků na místě příznaků statických.
Ve specifických úlohách se občas využívají i delta příznaky třetího řádu, $\Delta\Delta\Delta$ [4].

2.4 Fonetika

Fonetika je věda zabývající se fyzikálními vlastnostmi řeči. Dělí se na tři podobory. Ty se zabývají fyziologickými vlastnostmi tvorby řeči (artikulace), akustickými vlastnostmi přenášené řeči a jejím vnímáním člověkem.
Základním stavebním kamenem řeči z hlediska fonetiky je hláska, také nazývaná fón. Jde o nejmenší řečovou jednotku popisující distinktivní zvuk.
Druhým významným pojmem je foném. Jde o základní stavební jednotkou fonologie - vědy zabývající se zkoumáním využití zvuků v jazyce. Jako foném je brána nejmenší lingvistická jednotka schopná měnit význam slova. Fonémy jsou specifické pro různé jazyky a bývají definovány pomocí párových slov - slov které se liší pouze jedním fonémem.
Pro zpracování řeči je často nezbytná segmentace nahrané lidské řeči spolu s odpovídajícím fonetickou transkripcí. Mapování poskytnuté touto transkripcí nachází uplatnění i v detektorech řečové aktivity.

2.5 Standardní algoritmy VAD

Detekce řečové aktivity (Voice Activity Detection - VAD) je proces rozlišení časových úseků s řečovou aktivitou od úseků s absencí řeči. Heuristické algoritmy VAD typicky pracují s příznaky extrahovanými ze segmentů signálu.

Extrahované příznaky jsou porovnány s příznaky získanými z pauz mezi řečovými úseky a na základě určeného prahu dojde k rozhodnutí [5].
Rozhodovací práh může pevný nebo adaptivní. Pevný práh se obvykle nastavuje podle dynamiky signálu. Hodnota adaptivního prahu se zase mění podle aktuálních charakteristik sledovaného signálu.
Heuristické detektory řečové aktivity lze dělit podle použitých příznaků na [2]:

- Energetické detektory,
  které sledují časový průběh energie signálu. Výhodou tohoto druhu detektoru je jeho snadná implementace a malá výpočetní náročnost. Nevýhodou je jeho nedostatečná robustnost vůči šumu, která se projevuje především u neznělé řeči.
- Spektrální detektory,
  které fungují na základě energií vázaných k jednotlivým kmitočtovým pásmům. Tyto algoritmy rozhodují na základě spektrální vzdálenosti mezi signály.
- Kepstrální detektory,
  které jsou podobné detektorům spektrálním. Používanými příznaky jsou aproximace vyhlazených spekter a podle nich vypočítaná kepstrální vzdálenost mezi současným segmentem a průměrným kepstrem pozadí.
- Koherenční detektory,
  které rozhodují na základě koherenční funkce. Předpokladem pro využití tohoto druhu detektoru je jeho využití pro signály s alespoň dvěma kanály a za přítomnosti nekoherentního šumu.

Standardní heuristické detektory

Chybovost standardních heuristických detektorů se obvykle pohybuje mezi 10 až 30 % v závislosti na typu a úrovni šumu pozadí. Příkladem realizovaných detektorů mohou být standardní detektory ITU-T G.729, ETSI AMR1 nebo ETSI AMR2 [6]. Detektor G.729 pracuje s rozdílem energií přes celé frekvenční pásmo, s počtem průchodů nulou a se spektrálním zkreslením [6]. Detektor AMR1 pracuje s hodnotami SNR v různých spektrální pásmech. Detektor AMR2 pracuje s odstupem signálu od šumu a hlasovými charakteristikami mluvčích. Při přítomnosti šumu s úrovní odstupu signálu od šumu 0 dB se pohybuje efektivita detektoru G.729 okolo 45 % a pro odstup signálu od šumu 15 dB dosahuje chybovosti 32.7 % a pro SNR = 30 dB okolo 30 % [7]. Detektory AMR1 a AMR2 jsou o něco efektivnější, AMR1 dosahuje pro stejné hodnoty odstupu signálu od šumu respektivních chybovostí 48.9, 26.4

a 11.1 %. Detektor AMR2 se pohybuje pro odstupy signálu od šumu 0 - 15 dB okolo chybovosti 20 % a pro SNR 30 dB okolo chybovosti 11 % [7].

Dalším druhem detektorů řečové aktivity jsou detektory stochastické. Tyto detektory rozhodují na základě statistických modelů řeči. Často pracují na základě skrytých markovovových model (Hidden Markov Model - HMM) a Gaussovských směsí (Gaussian Mixture Model - GMM). Nevýhodou tohoto druhu detektorů je to, že vyžadují velké množství trénovacích dat pro správné nastavení vnitřních parametrů, které jsou nezbytné pro správnou funkci detektoru [2].
Poslední skupinou detektorů řečové aktivity popsanou v této práci jsou detektory využívající neuronové sítě.

Kapitola 3

Použití strojového učení v hlasových technologiích

Tato část popisuje základy strojového učení nezbytné pro pochopení principu fungování umělých neuronových sítí. Strojové učení je souhrnný název pro počítačové algoritmy spadající pod umělou inteligenci, které jsou schopné se samostatně učit z předcházejících zkušeností. Algoritmy strojového učení vytvářejí na základě vstupní (trénovací) množiny dat matematické modely, podle kterých mohou systémy samostatně vykonávat předem definované úkoly. Pro popis metod strojového učení budu v této práci používat notaci převzatou převážně z [8].

3.1 Metody strojového učení

Algoritmy strojového učení se dělí do několika základních skupin podle své funkce, přístupu k datům a metodě učení:

- Učení s učitelem (supervised learning) - V učení s učitelem je ke každému prvku trénovací množiny přiřazen jeho předem připravený, očekávaný výstup. Na základě vztahu mezi vstupem a výstupem se pak systém učí jak k novým vstupům přiřazovat správné výstupy.
- Učení bez učitele (Unsupervised learning) - Učení bez učitele pracuje pouze s trénovací množinou, bez přidaných připravených odpovědí a

hledá v trénovací množině dat podobné elementy a vzory, kterými je trénovací množina charakterizována.

- Zpětnovazebné učení (Reinforcement learning) - Ve zpětnovazebném učení je využito zpětné vazby pro ovlivňování učící procesu po každém provedeném rozhodnutí s cílem maximalizovat dlouhodobý zisk.

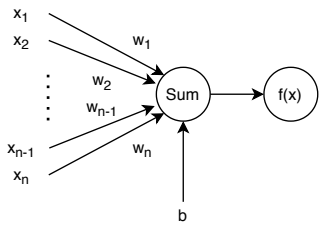
Tato práce se zabývá především metodami učení s učitelem. Následuje popis nejpopulárnějších druhů úloh pro tento způsob učení.

3.1.1 Úlohy učení s učitelem

Mezi nejpoužívanější typy úloh učení s učitelem patří úlohy klasifikace a regrese [8]. Cílem těchto úloh je k novému pozorování přiřadit odpovídající rozhodnutí podle získaného předpisu. Výstupem úlohy klasifikace je diskrétní rozhodnutí. Často do které z předem definovaných skupin patří nové pozorování. Úloha regrese se od tohoto přístupu liší tím, že nám místo diskrétního rozhodnutí dává k dispozici vypočítané spojité hodnoty. Může jít například o predikce teploty nebo atmosférického tlaku.
Úloha detekce řečové aktivity spadá pod úlohy učení s učitelem. Fonetické přepisy databází a připravené příznaky jsou pro nás dvojicí očekávaných výstupů a vstupních dat. Detekce řečové aktivity je úlohou klasifikace s rozhodnutím mezi dvěma vzájemně se vylučujícími stavy.

3.1.2 Perceptron

Perceptron je základní stavební buňkou neuronových sítí a strojového učení. Jde o model umělého neuronu, částečně modelovaného podle modelu fungování neuronů v lidském mozku.



Obrázek 3.1: Model perceptronu

Tento model můžeme matematicky popsat následovně [8] : Neuron přijímá vstupní hodnoty **x** a vytváří jejich lineární kombinaci za pomoci parametrů **w**. Tyto parametry jsou nazývány váhy neuronu.

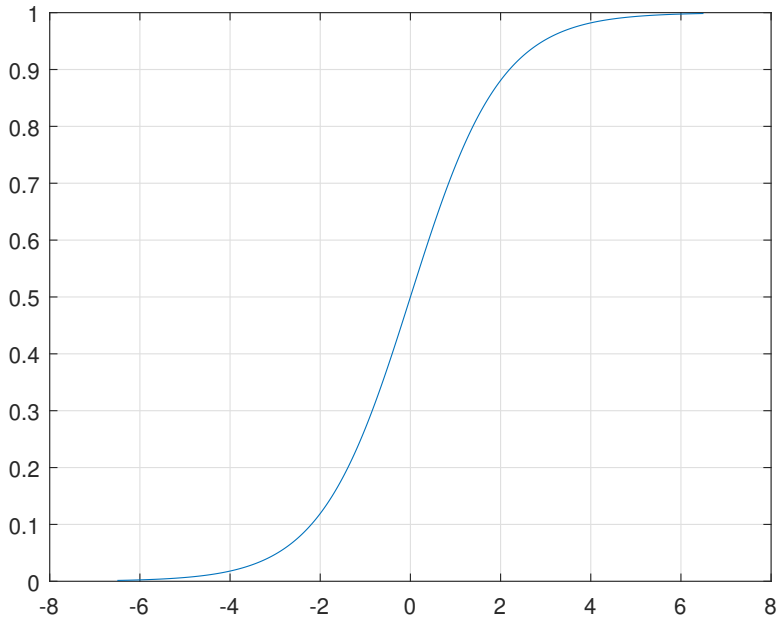
$$a = \sum_{i=1}^{D} w_i x_i + w_0 \tag{3.1}$$

kde w_n jsou váhy neuronu, w_0 je jeho práh, nebo také bias a a je aktivační hodnotou modelu. Aktivační hodnota je transformována nelineární aktivační funkcí σ.

$$z = \sigma(a) \tag{3.2}$$

Hodnota z je pak výstupem neuronu. Ve většině případů je výstup jedné vrstvy sítě vstupem neuronů vrstvy následující. Výjimkou je první a poslední vrstva sítě. Vstupem první vrstvy jsou vstupní data a výstupem poslední vrstvy je výstup sítě samotné.
Při tvorbě neuronových sítí lze vybrat z několika aktivačních funkcí. Jednou z nejrozšířenějších bázových funkcí je sigmoidní aktivační funkce

$$\sigma(a) = \frac{1}{1 + e^{-a}} \tag{3.3}$$



Obrázek 3.2: Sigmoidní aktivační funkce

Jedním z důvodů pro použití této funkce je její snadno použitelná derivace

$$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1 - \sigma(a)) \tag{3.4}$$

Příbuzným sigmoidní bázové funkce je hyperbolický tangens, daný následujícím vztahem.

$$tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{3.5}$$

Ten je svázán se sigmoidou přes následující vztah

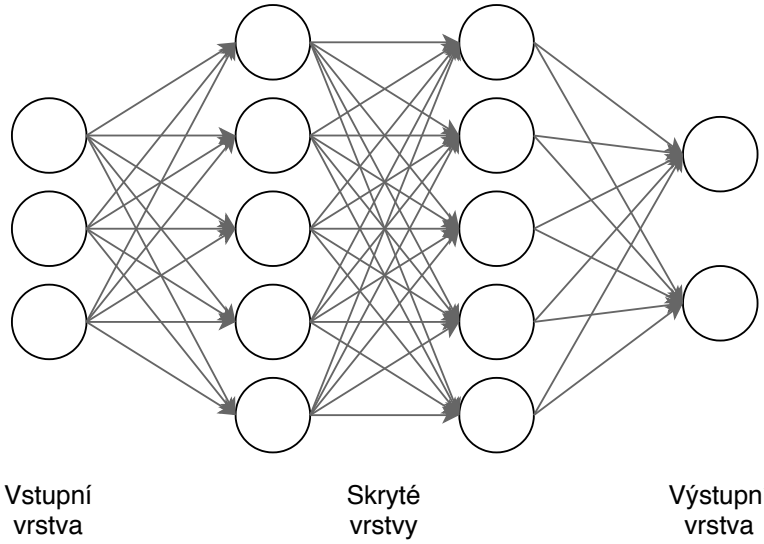
$$tanh(a) = 2\sigma(a) - 1 \tag{3.6}$$

Při použití hlubokých neuronových sítí při práci s klasifikátory řeči se právě tyto dvě funkce používají nejčastěji [9]. Obecně se však dají použít i další bázové funkce. Příkladem může být bázová funkce ReLU (rectified linear unit) [10]

$$ReLU(a) = max\{0, a\} \tag{3.7}$$

3.2 Neuronové sítě

Jedním z nejdůležitějších nástrojů strojového učení jsou umělé neuronové sítě (Artificial Neural Network - ANN). Standardní umělá neuronová síť je vícevrstvým perceptronem (Multilayer perceptron - MLP) [8]. Tento druh sítě je tvořen vrstvami neuronů, jejichž vzájemná propojení svou strukturou napodobují struktury neuronů v lidském mozku. Výstupy z jednotlivých neuronů jsou používány jako vstupy do neuronů v následujících vrstvách. Umělé neuronové sítě se dále rozdělují podle rozmístění neuronů a způsobu jakým mezi sebou komunikují.
Důležitým druhem neuronových sítí jsou hluboké neuronové sítě (Deep Neural Network - DNN). Hluboká neuronová síť má mezi vstupní a výstupní vrstvou několik vrstev vzájemně komunikujících neuronů, které se nazývají skryté vrstvy. Pokud výstupy těchto vrstev vstupují pouze do následujících vrstev, mluvíme o sítích dopředných (feedforward neural network). Pokud jsou však výstupy některých vrstev posílány zpátky do vrstev předcházejících, mluvíme o sítích zpětnovazebních (recurrent neural network - RNN). Vrstvy neuronů nacházející se mezi vstupní a výstupní vrstvou se nazývají skryté vrstvy.

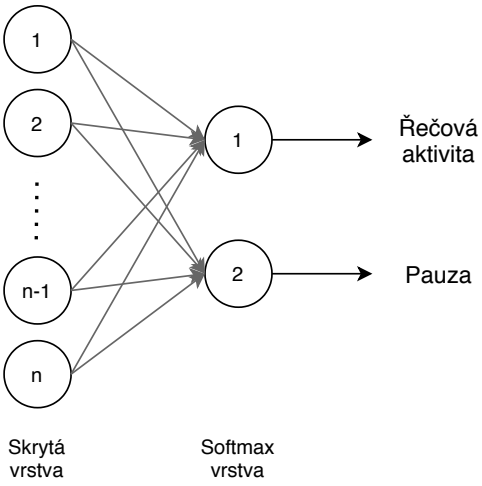


Obrázek 3.3: Hluboká neuronová síť

Ve výstupní vrstvě neuronových sítí se nejčastěji používá Softmax funkce. Cílem této funkce je normalizovat výstupní rozdělení pravděpodobnosti, které by podle použitých aktivačních funkcí mohlo nabývat hodnot mimo interval $(0, 1)$ do intervalu $(0, 1)$. Předpis softmax vrstvy vypadá následovně:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.8}$$

kde K je počet výstupních neuronů.
V detektoru řečové aktivity budeme pracovat pouze se dvěma výstupními neurony - s jedním pro řečovou aktivitou a jedním pro její absenci.



Obrázek 3.4: Softmax vrstva

3.2.1 Trénování hlubokých sítí

Cílem trénování neuronové sítě je najít takové parametry w, aby byla s co největší přesností schopná najít v novém pozorování charakteristiky trénovací množiny.

Chybové funkce

Jako metrika pro ověření efektivity sítě se nejčastěji využívají střední kvadratická chyba (MSE - Mean Squared Error) a křížová entropie (CE - Cross Entropy). Chyba typu MSE nachází uplatnění především v úlohách regrese a je dána následujícím předpisem [8]:

$$E(t, y) = \frac{1}{2} \sum_{d=1}^{D} (t_d - y_d)^2 \tag{3.9}$$

kde t_d je cílová trénovací hodnota, y_d je výstup neuronové sítě a D je množina trénovacích dat. Pro úlohy klasifikace se zase většinou používá chybová funkce křížová entropie [8] :

$$E(t, y) = -\sum_{d=1}^{D} t_d \log(y_d) \tag{3.10}$$

kde t_d je přiřazenou cílovou hodnotou, často zakódovanou pomocí one-hot kódování, y_d je výstup sítě a D je počet možných výstupů sítě.

Gradient descent

Pro nalezení optimálních vah w se nejčastěji používá metoda stochastického gradientního sestupu (stochastic gradient descent). Princip této metody spočívá ve sledování chování gradientu chybové funkce vůči parametrům neuronové sítě. Parametry sítě jsou pak modifikovány s cílem najít extrém cenové funkce.

$$\nabla E(w) = \frac{\partial E}{\partial w} \tag{3.11}$$

Nové váhy jsou pak určeny jako

$$w^{t+1} = w^t - \mu \nabla E(w^t) \tag{3.12}$$

kde w^t jsou váhy sítě pro trénovací epochu t a parametr μ označuje parametr nazývaný rychlost učení (learning rate). Rychlost učení nabývá hodnot mezi

nulou a jedničkou. Hodnota rychlosti učení se často nastavuje podle zvolené aktivační funkce.
Podle zvoleného přístupu k množině trénovacích dat může trénovaní pomocí algoritmů gradientního sestupu probíhat několika způsoby. Prvním přístupem je již zmíněný stochastický sestup gradientu (stochastic gradient descent). Tato metoda je nejjednodušším přístupem ke trénování. Proces trénování pracuje v každé iteraci se všemi prvky testovací množiny. Pro každý jeden vzorek testovací množiny jsou vypočítány gradienty chybové funkce a jsou upraveny váhy. Tento proces je v každém trénovacím cyklu (epoše) opakován pro všechny prvky testovací množiny. Další možností je využití dávkových metod (batch methods). V každé iteraci algoritmus pracuje s celou trénovací množinou najednou. Výsledná úprava vah je prováděna podle průměrných hodnot vypočítaných pro jednotlivé vzorky. Poslední možností je trénování s využitím malých dávek (mini-batches). Algoritmy této metody fungují na podobném principu jako metody dávkové. Velikost dávky se kterou pracují je ale omezená. Místo celé trénovací množiny pracují pouze s dávkami o velikosti v řádu desítek až stovek [11].
Volba metody závisí na několika faktorech jako dostupná paměť pro zachování mezivýsledků, požadavek na rychlost konvergence algoritmu nebo nutnost zpracování výsledků za chodu systému (online). Algoritmy pracující s malými dávkami pracují rychleji a jsou odolnější vůči fluktuacím. Algoritmy procházející celý trénovací soubor zase lépe konvergují[11].

Zpětné šíření chyb

Výše popsaný přístup se dá použít pro jednoduché systémy, ale je sám o sobě nedostatečný pro trénování složitějších struktur (eg. struktur obsahující skryté vrstvy neuronů). Pro výpočty gradientu těchto složitějších konstrukcí se využívá algoritmus zpětného šíření chyby (Backpropagation) [11]. Algoritmus zpětného šíření umožňuje cenové funkci procházet sítí od konce a studovat jak je ovlivňována jednotlivými parametry.
Výstup každého neuronu závisí na jeho vstupních vahách $\mathbf{w}$ a jeho vstupních hodnotách $\mathbf{x}$. Jeho vstupní hodnoty jsou zároveň výstupy neuronů předchozích vrstev a platí pro ně podobné závislosti. Algoritmus zpětného šíření je založen na využití řetězového pravidla pro derivace složené funkce.

Chceme-li například zjistit novou hodnotu váhy w_{ji} spojující neurony j a i. Výpočet nové hodnoty této váhy vyžaduje znalost derivace chybové funkce [8].

$$\Delta w_{ji} = -\mu \frac{\partial E}{\partial w_{ji}} \tag{3.13}$$

Aktivační hodnotou neuronu j je a_j, která je funkcí vah neuronu j. Za pomoci

řetízkového pravidla tedy můžeme psát:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} \tag{3.14}$$

Zde se často zavádí notace

$$\delta_j = \frac{\partial E}{\partial a_j} \tag{3.15}$$

Kde δ_j je chybou na úrovni aktivační hodnoty neuronu j. Dále můžeme psát :

$$\frac{\partial a_j}{\partial w_{ji}} = z_i \tag{3.16}$$

a dosazením do 3.15 a 3.16 do 3.14 dostaneme

$$\frac{\partial E}{\partial w_{ij}} = \delta_j z_i \tag{3.17}$$

Derivace chybové funkce podle váhy w_{ji} je tedy násobkem hodnoty δ a výstupní hodnotou neuronu i z_i [8].

3.2.2 Hluboké sítě v hlasových technologiích

Mezi využitím hlubokých neuronových sítí v hlasových technologiích převládají dva hlavní přístupy, hybridní kombinace hlubokých neuronových sítí a skrytých markovových modelů (HMM-DNN) a End-to-End (E2E) zpracování [12].
Přístup HMM-DNN vznikl evolucí do té doby používaných HMM-GMM modelů. Pravděpodobnosti jednotlivých stavů pro zpracování skrytými markovovými modely se místo gaussovských směsí získávají za pomoci neuronových sítí. Oba dva přístupy na bázi HMM však vyžadují velké množství práce věnované přípravě dat. Pro správnou funkci je nezbytné mít k dispozici fonetické transkripce trénovacích promluv. Pro úlohy rozpoznávání řeči je třeba mít navíc i statistické, jazykové a výslovnostní modely, které jsou často specifické pro jazyk, ve kterém je databáze nahrána. Za pomoci těchto modelů lze namapovat části promluv jako posloupnosti fónů k odpovídajícím časovým segmentům [13]. Sítě pracující na HMM-DNN principu určují za pomoci hlubokých neuronových sítí pravděpodobnosti výskytu fónů. HMM následně na základě znalosti jazykového modelu hledá nejpravděpodobnější sekvenci fónů, kterou je možné mapovat na známá slova.

Druhým rozšířeným způsobem využití DNN sítí v úlohách ASR je takzvané end-to-end (E2E) rozpoznávání. End-to-end ASR modely nevyžadují pro svojí funkci předem připravené příznaky. Stále však potřebují znát výslovnostní

a jazykové modely. Stejně tak nepoužívají skryté markovovy modely pro dekódování posloupností fónů.
End-to-end systémy nahrazují segmentovaný postup přípravy akustického systému jediným krokem, ve kterém pracují přímo se vstupním řečovým signálem místo předem napočítaných příznaků a převádějí jej na výstupní sekvence symbolů. Některé modifikace tohoto přístupu vyžadují použití konvolučních, nebo Long short-time memory (LSTM) sítí. E2E systémy se dále dělí do dvou hlavních skupin na attention based modely a na modely využívající connectionist temporal classification (CTC) [14]. CTC je v modelech využíváno pro mapování segmentů s řečovým signálem k rozpoznaným symbolům. Attention based modely, které pracují na bázi páru kodérů postavených na LSTM sítích, zase nacházejí uplatnění především v úlohách zpracování přirozeného jazyka (natural language processing) [15].

3.3 Konvoluční neuronové sítě

Konvoluční neuronové sítě (Convolutional neural networks) jsou specifickým typem neuronové sítě používající operaci konvoluce přes vstupní hodnoty v alespoň jedné vrstvě. Tyto sítě našly uplatnění především v oblasti počítačového vidění, ale jejich využití se postupně rozšířilo i do dalších oblastí, jako je zpracování řeči. Neuronové sítě nejčastěji využívají jednodimenzionální, nebo dvoudimenzionální konvoluci. Dvoudimenzionální konvoluce, která se originálně začala používat počítačovém vidění, pracuje na vstupu s obrazem, kde je poloha pixelů udávána ve dvou osách. Jednodimenzionální konvoluce pracuje s vektorem hodnot, který mohou tvořit po sobě jdoucí vzorky signálu nebo jeho spektrální koeficienty. Jako konvoluční vrstva je typicky označována vrstva, která se skládá z operace konvoluce, aplikace nelineární aktivační funkce a poolingové vrstvy [11]. Diskrétní jednodimenzionální konvoluci datového vektoru f a konvolučního jádra g píšeme jako

$$(f * g)[m] = \sum_{n} f[n]g[m-n] \tag{3.18}$$

Dvoudimenzionální konvoluce vstupních dat f a konvolučního jádra g je dána následujícím předpisem

$$(f * g)[m,n] = \sum_{k}\sum_{l} f[k,l]g[m-k,n-l] \tag{3.19}$$

Většina nástrojů pro neuronové sítě však místo konvoluce aplikuje vzájemnou korelaci [11], danou následujícím předpisem:

$$(f \star g)[m,n] = \sum_{k}\sum_{l} f[k,l]g[m+k,n+l] \tag{3.20}$$

Koeficienty konvolučních jader (filtrů) nazýváme jejich váhami. Při aplikaci konvoluce se konvoluční jádro pohybuje nad vstupními daty a na každé své pozici generuje novou hodnotu. V konvolučních vrstvách se typicky vyskytuje větší množství těchto filtrů s různými nastaveními vah filtrů a jejich hodnoty jsou předmětem procesu učení. Při posouvání konvolučního filtru přes vstupní data je aplikován princip sdílení vah (weight sharing), tedy že váhy filtru mají stejnou hodnotu na všech pozicích filtru.

Pooling

Další částí konvolučních vrstev je poolingová vrstva. Hlavním cílem této funkce je snížit rozlišení získaných příznaků a zredukovat množství dat, které putuje dále sítí [11]. Výsledkem této operace je zrobustnění rozpoznávací schopnosti sítě vůči malým translacím. Nejčastěji využívanými typy poolingu jsou maxpool a avgpool. Maxpool vrstva zachová z vybraného polohovacího okna pouze nejvyšší hodnotu, kterou pošle dál. Vrstva avgpool vypočítá průměrnou hodnotu vybraného okna a pošle ji sítí dál. Velkým rozdílem mezi Max a Avg vrstvou je ten, že avgpool posílá dál část informace ze všech vstupních hodnot, zatímco maxpool posílá informaci pouze o jedné hodnotě a zbytku se zbavuje.

3.3.1 Trénování konvolučních sítí

Konvoluční neuronové sítě se stejně jako hluboké neuronové sítě učí za pomoci algoritmu zpětného šíření chyb na základě sestupu gradientu. U plně propojených vrstev se aktualizují váhy a biasy jednotlivých neuronů. V konvolučních vrstvách se aktualizují váhy konvolučních jader, přes která byla prováděna konvoluce. Pokud je součástí konvoluční vrstvy poolingová vrstva typu Avg, dojde k aktualizaci všech relevantních hodnot. Je-li součástí konvoluční vrstvy poolingová vrstva typu maxpool, dochází k aktualizaci pouze v místě, odkud pochází největší hodnota.

3.3.2 Konvoluční sítě v hlasových technologiích

Stejně jako hluboké sítě, i konvoluční sítě nacházejí uplatnění v úlohách využívající HMM a v úlohách E2E zpracování. V úlohách typu CNN-HMM

převládá využití dvoudimenzionální konvoluce nad spektrogramy a jednodimenzionální konvoluce ve spektrální oblasti. V modelech E2E zase převládá využití jednodimenzionální konvoluce přes časovou doménu, kde jsou konvoluční neuronové sítě často využívány k extrakci příznaků přímo ze vzorků zkoumaného signálu.
Výhodou využití konvolučních neuronových sítí je jejich schopnost extrahovat příznaky z malých částí vstupu. Posuv filtrů a sdílení vah nám umožňují hledat stejné typy příznaků v různých částech frekvenčního spektra. Následná poolingová vrstva snižuje rozlišení získaných hodnot a zlepšuje tak odolnost sítě vůči drobným posuvům ve frekvenci, které mohou být způsobeny rozličnými délkami vokálních traktů mluvčích.

3.4 Architektury sítí

Tato část obsahuje obecné zásady tvorby modelů strojového učení, ukázky realizovaných modelů strojového učení v úlohách rozpoznávání řeči a popis architektury realizované v druhé části práce. Obecně neexistuje žádný předpis, nebo struktura, které by se daly použít jako univerzální řešení pro úlohy strojového učení. Struktury používané pro úlohy typu rozpoznávání spojité řeči s velkým slovníkem nemusí být vhodné pro úlohy rozpoznávání fonémů nebo detekce řečové aktivity. Při budování modelu je vždy nutné brát v potaz velikost výstupní vrstvy, velikost vstupních dat, konečnou hodnotící metriku, dostupnost dat nebo možnost rozšíření použitých databází [11].

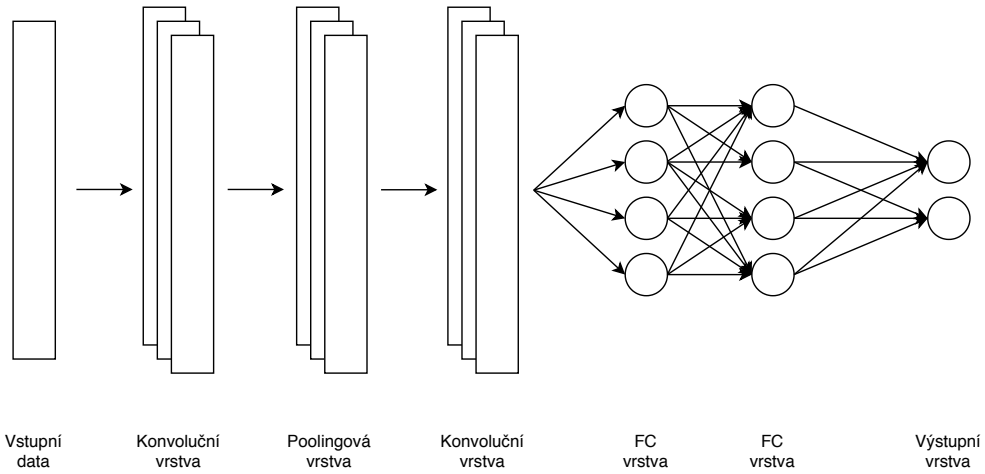
3.4.1 Příklady architektur realizovaných v rozpoznávání řeči

Tato sekce obsahuje několik vybraných realizovaných struktur rozpoznávačů řeči týkajících se hlubokých i neuronových sítí, se snahou ilustrovat rozmanitost používaných struktur a typů úloh. Mezi prvními modely využitými pro akustické modelování byl model navržený v [16], který se skládal z Deep belief network (DBN) s až osmi vrstvami a 2048 neurony na vrstvu. Podobné experimenty byly provedeny v [17] pro úlohu rozpoznávání s velkým slovníkem s jednou až pěti skrytými vrstvami o 2048 neuronech. Struktury konvolučních sítí nalezly uplatnění například v [18] pro úlohu detekce řečové aktivity s využitím dvoudimenzionální konvoluce. Konvoluční sítě se také uplatnily ve [19], kde dvě konvoluční vrstvy aplikují jednodimenzionální konvoluci přes frekvenční oblast. Novějším využitím konvolučních sítí je jejich aplikace přes časovou doménu přímo na vzorky signálu pro úlohu rozpoznávání fónů [20]. Tato realizace obsahuje tři konvoluční vrstvy následované jednou skrytou

vrstvou.

3.4.2 Realizovaná architektura

Struktura realizovaná v této práci je inspirovaná strukturami realizovanými v [19] nebo v [21] jako součásti řešení úloh rozpoznávání slov. Jde o síť se dvěma jednodimenzionálními konvolučními vrstvami. Operace konvoluce je prováděna ve spektrální oblasti nad log-mel bankami filtrů. Výstup z první konvoluční vrstvy prochází maxpoolingovou vrstvou, která snižuje rozlišení získaných hodnot. Výstup z poolingové vrstvy prochází sigmoidální aktivační funkci a a následně jde dál do druhé konvoluční vrstvy s větším počtem filtrů. Z druhé konvoluční vrstvy již data putují přímo do plně propojených vrstev se sigmoidálními aktivačními funkcemi a nakonec do softmax vrstvy na výstupu. Výstupní vrstva má pouze 2 neurony, jeden pro řečovou aktivitu a jeden pro její absenci.



Obrázek 3.5: Konvoluční neuronová síť

Abychom mohli využít vlastností konvolučních vrstev pro extrakci lokálních rysů dat, je třeba využít příznaky, které zachovávají korelaci v čase nebo ve frekvenci. Z tohoto důvodu je nevhodné použít klasické MFCC příznaky. Diskrétní kosinová transformace, přes kterou jsou tyto příznaky tvořeny, data dekoreluje. Místo toho jsou použity melovské spektrální příznaky, konkrétně log mel banka filtrů. Melovské příznaky zachovávají lokální korelovanost v čase i ve frekvenci. Místo využití Δ příznaků je časový kontext dodán splicingem, tedy zřetězením příznaků několika po sobě jdoucích segmentů.

Síť je trénována optimalizací křížové entropie přes stochastický sestup

gradientu s malými dávkami. Trénovací proces je ukončen při příliš malém zlepšení hodnocené funkce, nebo po provedení maximálního počtu 20 iterací. Ve většině realizovaných experimentů síť dokonvergovala mezi 10 až 15 iteracemi. Počáteční rychlost učení byla nastavena jako 0.008.

Kapitola 4

Implementace

4.1 Použité programové sady

V této části jsou popsány nástroje, knihovny a programy použité pro realizaci detektoru řeči.

4.1.1 Kaldi

Pro realizaci detektorů řečové aktivity využíváme sadu nástrojů pro Automatic Speech recognition s názvem Kaldi [22], licencovanou pod Apache 2.0 . Kaldi je sada nástrojů napsaných v jazyce C++ zaměřená na automatické rozpoznávání řeči s využitím strojového učení. Kaldi je vytvořeno a optimalizováno především pro prostředí typu Unix. Ideálním prostředím pro výpočet je Linux cluster libovolné distribuce využívající Oracle grid engine (dříve Sun grid engine - SGE).
S toolkitem se také dá operovat na pouze jednom stroji. To však přichází s podstatným prodloužením výpočetní doby. Pro zrychlení výpočtů umožňuje kaldi paralelizovat výpočty s využitím grafických jader(gpu) přes architekturu CUDA (Compute Unified Device Architecture). CUDA je rozhraní vyvinuté firmou Nvidia pro umožnění využití grafických karet pro obecné výpočetní účely v programech psaných v jazycích C, C++ nebo Fortran.

Příprava Kaldi

V rámci přípravy instalace Kaldi je potřeba pro stáhnutí toolkitu nainstalovat několik utilit jako Git, wget, perl, bash, grep a make.
Kaldi je pak třeba stáhnout pomocí příkazu

```
git clone https://github.com/kaldi-asr/kaldi.git
```

Prvním krokem v instalaci Kaldi je zajištění instalace nezbytných knihoven a nástrojů pro chod kaldi jako openFST, IRSTLM, ATLAS nebo OpenBLAS. Dostupnot těchto nástrojů se dá ověřit pomocí skriptu */extras/check_dependencies.sh*. Instalační skripty pro tyto nezbytnosti jsou většinou součástí distribucí Kaldi. Jsou-li všechny nutné soubory nainstalovány, je třeba ještě Kaldi zkompilovat pomocí příkazu `make`.

Struktura Kaldi

Na nejvyšší úrovni se Kaldi dělí do několika adresářů, z nichž pro uživatele nejdůležitější jsou:

- egs - adresář, ve kterém se skrývají předpřipravené ukázkové recepty Kaldi pro rozpoznávání řeči přizpůsobené pro různé řečové databáze.
- src - složka obsahující většinu skriptů spojených s chodem neuronových sítí a jimi prováděných výpočtů
- tools - jde především o soubory pro instalaci utilit nezbytných k chodu Kaldi
- windows - zde se nachází několik skriptů pro chod Kaldi pod operačním systémem Windows

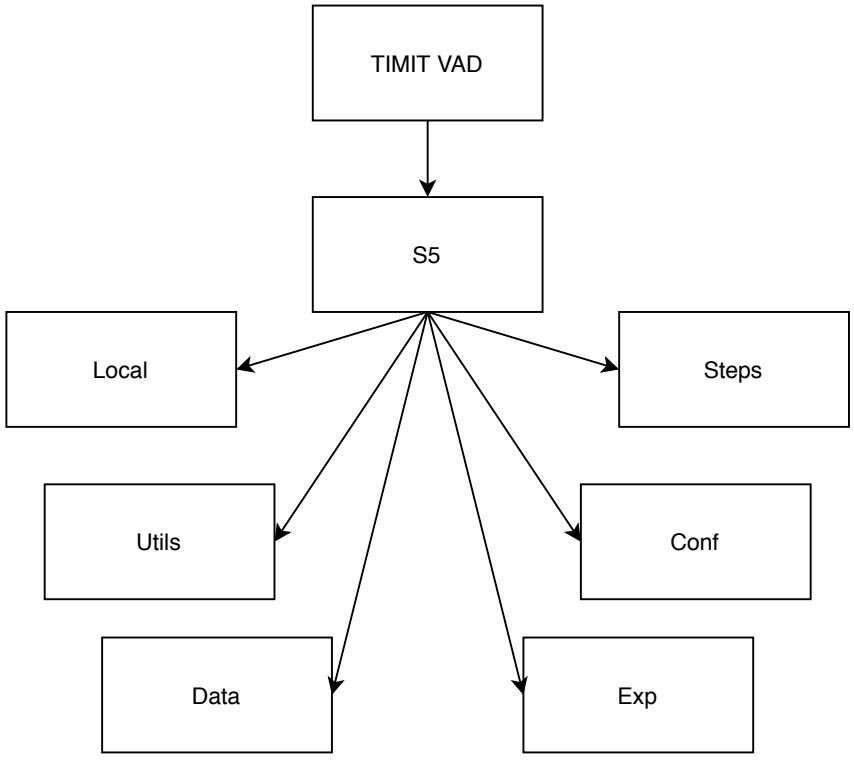
Práce s daty v Kaldi

Při práci s tabulkami kaldi pracujeme s řetězci, které udávají jakou formou se mají data ze souborů číst nebo do nich zapisovat. Řetězce s označením

`rspecifier` říkají programům čtoucím data, jestli mají očekávat archiv a jak k němu mají přistupovat. Označení `wspecifier` zase udává jak do souborů data zapisovat.
Mezi nejdůležitější možnosti rspecifieru, které budou v této práci využity jsou *s*, *t* a *ark*. Specifikátor *t* ukládá výstupní soubory v textové formě pro možnost úprav dalšími skripty. Pro *ark*, jsou soubory ukládány a čteny ve standardním módu pro práci v Kaldi. Specifikátor *s* značí, že data jsou v souboru seřazená.

Struktura Kaldi receptů

Připravené recepty Kaldi používají pevně danou adresářovou strukturu. Ve složce nesoucí název projektu, často zpracovávané databáze nebo specifického přístupu k datům, jsou k dispozici různé verze projektu. Nejaktuálnější verze projektu se zpravidla označuje jako "S5".
V této složce již lze naleznout skripty *run.sh*, *cmd.sh* a *path.sh*, které tvoří jádro receptu. Recept je koncipován tak, aby spuštěním skriptu *run.sh* byly provedeny veškeré úkony nezbytné pro přípravu dat, trénování rozpoznávače a vyhodnocení jeho efektivity. Dále zde lze nalézt několik složek, z nichž nejdůležitější jsou na následujícím diagramu.



Obrázek 4.1: Struktura Kaldi Receptů

- steps - obsahuje různé nástroje pro přípravu dat a trénování neuronových sítí.
- utils - obsahuje nástroje pro extrakci doprovodných statistik
- data - do této složky se ukládají data získaná během trénovacího procesu.
- local - v této složce lze nalézt programy a skripty přímo svázané s konkrétní řečovou databází.
- conf - obsahem této složky jsou konfigurační soubory pro skripty spuštěné během realizace rozpoznávače
- exp - do této složky se ukládají všechny soubory související se současnou natrénovanou sítí

Struktura a obsah adresářů steps a utils je standardizovaná pro všechny recepty Kaldi. Obsahy adresářů local,conf a data jsou relevantní pouze pro vybraný recept.

4.2 Použité řečové databáze

4.2.1 TIMIT

Pro realizaci detektoru byl vybrán řečový korpus TIMIT [23], jehož licenci ČVUT vlastní. Korpus TIMIT vznikl na základě spolupráce mezi Texas Instruments (TI), Massachusetts Institute of Technology (MIT) a SRI international. Distribuce databáze spadá pod National Institute of Standards and Technology (NIST), který korpus distribuuje buď online přes LDC (Linguistic Data Conortium) a nebo na CD-ROM. Jde o databázi jednovětných promluv nahraných se vzorkovací frekvencí 16 kHz. Korpus obsahuje 6300 promluv od 630 rodilých mluvčích v osmi nejrozšířenějších dialektech americké angličtiny. Mluvčí byli převážně vybíráni mezi zaměstnanci Texas Instruments. Každý mluvčí nahrál 10 stejných, foneticky bohatých vět o celkové délce okolo 30 vteřin. Součástí databáze TIMIT jsou ortografické a fonetické transkripce nahrávek pro další zpracování.
Databáze obsahuje 438 mužských mluvčí (70%) a 192 ženských mluvčích (30%)

Region	dr kód	Počet mužů	Počet žen	Celkový počet mluvčích
New England	1	31	18	49 (8%)
Northern	2	71	31	102 (16%)
North Midland	3	79	23	102 (16%)
South Midland	4	69	31	100 (16%)
Southern	5	62	36	98 (16%)
New York City	6	30	16	46 (7%)
Western	7	74	26	100 (16%)
Army brat	8	22	11	33 (5%)

Tabulka 4.1: Rozdělení databáze TIMIT

Příslušnost mluvčích k regionu se určuje podle oblasti, ve které mluvčí vyrůstal ve věku od dvou do deseti let [23].
Součástí databáze je doporučená separace promluv mezi testovací a trénovací množinou. Ta byla vytvořena podle následujících pravidel, doporučených i pro tvorbu vlastních separací testovací a trénovací části.

- Trénovací množina by měla obsahovat mezi 20 až 30 % z celkového počtu promluv a testovací množina by měla obsahovat zbylých 70 až 80 %.
- Žádný mluvčí by se neměl v objevit trénovací i v testovací části zároveň.
- Všechny dialekty by měly být zastoupeny v obou skupinách. Obě skupiny by měly obsahovat alespoň jednoho muže a jednu ženu od každého dialektu.
- Obě množiny neměly obsahovat stejné promluvy.
- Testovací část by měla obsahovat všechny dostupné fonémy, ideálně v různém kontextu.

4.2.2 QUT-NOISE

Pro otestování robustnosti detektoru vůči vlivu šumu prostředí byla vybrána databáze QUT-NOISE. Databáze QUT-NOISE obsahuje přes 10 hodin šumu pozadí z různých, běžně se vyskytujících prostředí [24]. Pro každé prostředí jsou data rozdělena mezi dva specifické scénáře. Mezi zahrnuté situace a k nim patřící scénáře patří:

- *CAFE* - Šum z prostředí typické venkovní restaurace (CAFE-CAFE) a z

restaurace umístěné uvnitř nákupního centra (CAFE-FOODCOURTB). V pozadí lze slyšet rozpravy mezi hosty nebo skřípot příborů.

- *HOME* - Domácí prostředí. Obsahuje zvuky typicky slyšitelné v kuchyni (HOME-KITCHEN) a v obývacím pokoji (HOME-LIVINGB). Zvuky spojené s částí obývací pokoj jsou například zapnutá televize nebo hrající si děti.
- *STREET* - Tato část korpu zahrnuje nahrávky pořízené u křižovatek. V jedné instanci jde o křižovatku nacházející se v centru města (STREET-CITY), druhá popisuje křižovatku ležící na předměstí (STREET-KG). Šum je zde tvořen především zvuky silniční dopravy.
- *CAR* - Část CAR obsahuje šum nahraný při jízdě na dálnici a při jízdě po městě a předměstí. Tento šum je dále dělen podle toho, jestli byla při nahrávání otevřena okna (CAR-WINDOWNB a CAR-WINUPB).
- *REVERB* - Poslední část databáze obsahuje zvuky nahrané v místech s dlouhou dobou dozvuku. Prvním z vybraných míst je uzavřený bazén (REVERB-POOL), druhým je částečně uzavřené parkoviště (REVERB-CARPARK). Šum v bazénu je tvořen zvuky jako je šplouchání lidí v bazénu. V části parkoviště je v pozadí slyšet převážně pohyb motorových vozidel.

Výhodou databáze QUT-NOISE jsou připravené MATLABovské kódy pro vytvoření databáze QUT-NOISE-TIMIT z již existující distribuce TIMITu. Díky strukturovanosti výsledné databáze lze snadno vytvářet datové sady libovolně kombinované z dostupných druhů prostředí a úrovní šumu.

4.3 Popis skriptů

Skripty použité v této práci byly připraveny na základě skriptů prezentovaných v diplomové práci [25], která se zabývá aplikací problematiky hlubokých neuronových sítí v detektorech řeči a na základě předpřipravených skriptů receptů Kaldi.
Výstup této práce je strukturovaný podle předpřipravených receptů Kaldi toolkitu.
Pro správnou funkci skriptů je potřeba korektně nastavit soubory *cmd.sh* a *run.sh*. Soubor *cmd.sh* umožňuje nastavit jakým způsobem probíhá paralelizace úloh na výpočetním clusteru. V této práci byla využita dvě hlavní nastavení, jedno pro práci na clusteru AMAGI a jedno pro práci lokálně. První nastavení, které vyžaduje pro svou správnou funkci nainstalovaný Sun grid engine.

```
export train_cmd="queue.pl -q cpu.q "
export decode_cmd="queue-pl -q cpu.q"
export mkgraph_cmd="queue.pl -q cpu.q"
export cuda_cmd="queue.pl -q gpu.q"
```

Pokud stroj, na kterém výpočty probíhají nemá k dipozici Sun grid engine, spuští se trénování s následujícími parametry

```
export train_cmd="run.pl"
export decode_cmd="run.pl"
export cuda_cmd="run.pl"
export mkgraph="run.pl"
```

Při tomto nastavení jsou všechny výpočty spouštěné lokálně.
Výše uvedená nastavení jsou standardní nastavení exemplárních receptů Kaldi. Tato práce pracuje s upraveným procesem, ve kterém není využit dekódovací proces. Proměnné `decode_cmd` a `mkgraph` tak nejsou nutné pro funkčnost realizovaného detektoru. Stejně tak není nutné používat proměnnou `cuda_cmd`, pokud není k dipozici grafická karta s její podporou. Soubor *path.sh* obsahuje cestu k instalaci Kaldi toolkitu a jeho rozličným sadám nástrojů.

4.3.1 run.sh

Po správném nastavení vnitřních skriptů jsou všechny části detektoru řečové aktivity od zpracování řečové databáze až po výsledné chybovosti realizovány pomocí jednoho skriptu *run.sh*. Tento skript obsahuje všechny níže popsané části trénovacího procesu.

Příprava databáze

Prvním krokem je příprava databáze pro extrakci příznaků a vypracování cílových vektorů. Pro případ databáze TIMIT je v Kaldi k dispozici shellový skript *timit_data_prep.sh*. Tento skript rozdělí řečovou databázi podle předem definovaného seznamu mluvčích na části Train, Test a Dev. Části train a dev obsahují promluvy používané pro učení sítě. Část test obsahuje promluvy testovacího souboru. Pro každou z těchto skupin dále vytvoří několik souborů,

které obsahují informace vázající jednotlivé promluvy k jejich unikátním identifikátorům, jejich ortografické transkripce nebo jejich fonetické transkripce. Důležitým výstupem jsou soubory *test_wav.scp*, *train_wav.scp* a *dev_wav.scp*, které obsahují seznamy promluv zařazených do dané skupiny a informace o umístění odpovídajících řečových signálů pro budoucí extrakci příznaků.

Extrakce příznaků

Druhým krokem je extrakce příznaků z řečové databáze a spárování příznaků s cílovými vektory.
Extrakce příznaků probíhá za pomoci skriptů, které jsou součástí instalace Kaldi. Pro příznaky využité v této práci jde o skripty *compute-fbank-feats* a *compute-mfcc-feats*. Volání obou skriptů je identické s výjimkou parametrů exkluzivních pro dané typy příznaků:

```
compute-fbank-feats [options...] <wav-rspecifier> <feats-wspecifier>
```

kde `options` je konfigurací příznaků. `Wav-rspecifier` ukazuje na místo odkud se čtou wav soubory a `feats-wspecifier` ukazuje, kam se zapisují extrahované příznaky.
Vstupními daty pro tento skript jsou soubory *_wav.scp* vzniklé v předchozím kroku. Výstupem je soubor typu .ark, v němž jsou uloženy napočítané příznaky. Jeho struktura je následující: za identifikátorem promluvy začíná hranatá závorka a na každém dalším řádku jsou příznaky pro jeden časový segment. Po posledním segmentu je promluva uzavřena hranatou závorkou.

4.3.2 Příprava cílových vektorů

Pro správnou funkci detektoru je třeba vytvořit soubor, který obsahuje pro každý časový segment informaci, jestli v něm je řeč, nebo její absence. Pro vytvoření tohoto souboru jsou využity soubory s příponou .phn, které vznikly jakou součást *timit_data_prep.sh* a obsahují fonetické transkripce nahrávek. Tyto soubory obsahují posloupnosti fónů spolu s časovými okamžiky, kdy každý fón začíná a končí.
Program napsaný v jazyce Python *label-prep.py* na základě těchto souborů a předem připravených seznamů tichých a hlasitých fónů vytvoří soubor obsahující identifikátory nahrávek a jim odpovídající cílové vektory s jedničkami pro

segmenty s řečovou aktivitou a nulami pro segmenty s její absencí. Vzniklé soubory jsou pojmenované *labels_test.ark* a *labels_train.ark*. Další program napsaný v jazyce Python, *feat-prep.py* prochází seznamy promluv přiřazených do oddílů test, train nebo dev a kontroluje, jestli jsou všechny promluvy přítomny a jestli jsou odpovídající příznaky a cílové vektory stejně dlouhé. Nevyhovující promluvy jsou ze souborů vyřazeny.
Pro databázi QUT-TIMIT jsou cílové vektory vytvořeny pomocí skriptu *qut-label-prep.py* napsaného v jazyce Python. Tento skript projde zvolené části databáze QUT-TIMIT a připraví soubory s cílovými vektory pro detektor řeči na základě souborů s příponou *.eventlab*, které obsahují časové značky značící trvání jednotlivých segmentů s řečí nebo s její absencí.

Trénování sítě

Kaldi obsahuje několik modelů pro trénování neuronových sítí, nnet1, nnet2 a nnet3. Pro realizaci tohoto detektoru byl vybrán Kaldi model nnet1. Tento model byl vybrán z důvodu možnosti práce s vlastními cílovými vektory a kvůli jeho kompatibilitě s konvolučními neuronovými sítěmi. Proces učení neuronové sítě se spouští příkazem

```
./steps/nnet/train.sh [options] <data-train> <data-dev> <lang-dir>
<ali-train> <ali-dev> <exp-dir>
```

kde `<data-train>` a `<data-dev>` ukazují na adresáře s daty připravenými pro trénování sítě. Možnosti `<lang-dir><ali-train><ali-dev>` se týkají rozpoznávačů pracujících s jazykovými a výslovnostními modely. Pro detektor řečové aktivity s modifikovanými cílovými vektory jsou nahrazeny prázdnými složkami `dummy-dir`. Poslední `<exp-dir>` značí adresář, kam se ukládá výsledná neuronová síť.
Pomocí možností `options` se nastavuje většina parametrů, které ovládají strukturu realizované neuronové sítě a učící proces. Ve finální formě tedy příkaz k trénování vypadá následovně.

```
  steps/nnet/train.sh --network-type cnn1d
--cnn-proto-opts "--pool-step 2 --pool-size 2"
--hid-layers 2 --num_tgt 2 --hid-dim 256
--splice 5 --copy-feats false
--labels "ark:ali-to-post ark:$dir/targets/targets.ark ark:-|"
--skip-phoneset-check true --skip-cuda-check true \
```

```
  $directory/data/train  $directory/data/dev dummy-dir dummy-dir
dummy-dir $directory/exp/nnet || exit 1;
```

Pro použití konvolučních vrstev a jejich správné nastavení jsou použity možnosti `network-type cnn1d` a `cnn-proto-opts`, které specifikují vlastnosti konvolučních vrstev. Možnosti `hid-layers` a `hid-dim` konfigurují podléhající hlubokou síť. Rychlost učení sítě lze nastavit pomocí parametru `learn-rate`. Zřetězení příznaků se nastavuje pomocí parametru `splice`.
Pro správnou funkci rozpoznávače detektoru řečové aktivity je nutné nastavit použití vlastních cílových stavů pomocí `num-tgt`, které určuje počet výstupních neuronů. Dále musí být nastavena i možnost `labels`, která ukazuje na předpřipravené cílové vektory, které jsou převedeny do formátu posterior pomocí Kaldi skriptu *ali-to-post* [22].

Detekce řečové aktivity

Klasifikace trénovací množiny je prováděna následující formou

```
nnet-forward [options] <nnet1-in> <feature-rspecifier>
<feature-wspecifier>
```

kde `<nnet1-in>` ukazuje na natrénovanou neuronovou síť vzniklou v předchozím kroku pod názvem *final.nnet*. Dále je nutné specifikovat soubor *final.feature_transform* který vznikl během trénování sítě. Možnost `<feature-rspecifier>` ukazuje, kde jsou uloženy příznaky připravené pro detekci. A `<feature-wspecifier>` ukazuje, kam se má zapsat výstup neuronové sítě.

Zhodnocení experimentu

Výstup neuronové sítě je vyhodnocen porovnáním s předem připravenými cílovými vektory. Pro tento účel byl vytvořen v Pythonu program *vad-prep.py*. Na základě rozdílu mezi cílovými vektory a výstupem sítě může požadované chybovosti. Zároveň upraví výstup neuronové sítě a připravené testovací cílové vektory do formy nezbytné pro jejich vyhodnocení programem Vadcrit [26], který provádí podrobnější analýzu výstupu.

Kapitola 5

Výsledky

V této části jsou specifikována hodnotící kritéria použitá pro vyhodnocení provedených experimentů. Dále zde jsou popsány výsledky a konfigurace provedených experimentů.

5.1 Chybové metriky

Tři nejdůležitější základní metriky chybovosti řečových detektorů, ERR, ERS a ERP, spolu s dalšími pokročilými metrikami efektivity řečových detektorů byly vyhodnocovány přes program vadcrit [26].

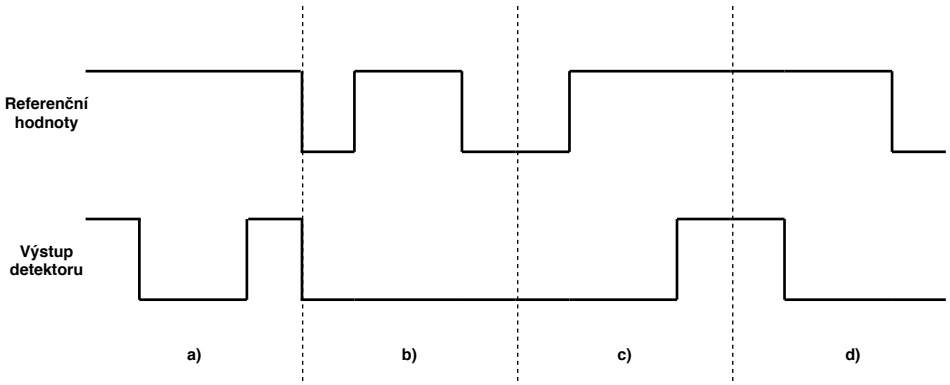
- ERR (ERRor decision) - Relativní množství všech typů chybných rozhodnutí. Jde o základní a často využívanou metriku pro určení efektivity detektoru řečové aktivity.

- ERS (ERror in Speech) - Chybová metrika zahrnující všechny případy, kdy je řečový signál chybně detekován jako jeho absence. Podle časového kontextu tuto metriku dále vyhodnocující program vadcrit dělí podle [27] na:

  - SDN (Speech Detected as Noise) - Pod chyby tohoto typu jsou zařazeny zašuměné úseky obklopené řečovou aktivitou nesprávně

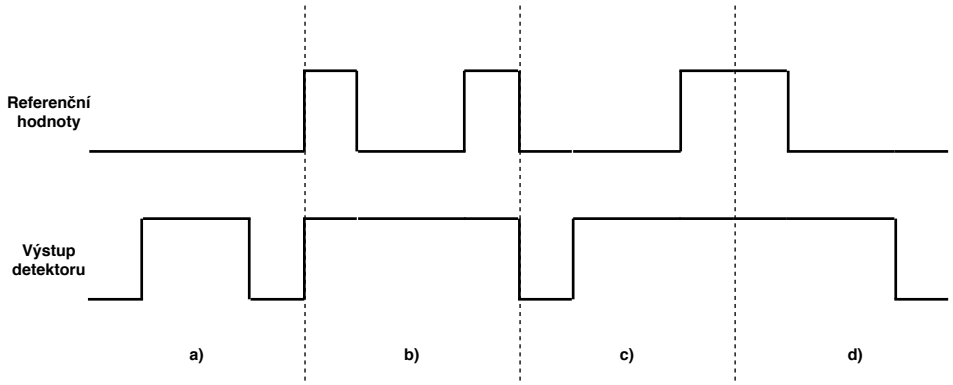
vyhodnocené jako úseky s řečovou aktivitou. Tento druh chyby je také někdy zařazován do skupiny chyb se souhrnným názvem *Mid Speech Clipping* [6].

- MIS (MIssed Speech) - Situace, ve kterých detektor vůbec nezachytne probíhající řečovou aktivitu. Tato chyba spadá do skupiny chyb MSC.
- TRF (TRuncation at the Front) - Začátek delšího úseku řeči je detekován se zpožděním. Tento druh chyby je také někdy označován jako Front End Clipping (FEC) [6].
- TRB (TRuncation at the Back) - Konec delšího úseku řeči je nesprávně detekován jako šum. Tato chyba spadá do skupiny chyb MSC.



Obrázek 5.1: chyby typu ERS - a) SDN b) MIS c) TRF d) TRB

- ERP (ERror in Pause) - Určuje s jakou přesností je detektor schopen klasifikovat úseky bez řečové aktivity. Podle časového kontextu lze tyto chyby dále dělit na [27]:
  - NDS (Noise Detected as Speech) - Uprostřed delší zašuměné části je je zašuměný segment chybně detekován jako segment řečové aktivity.
  - MIN (MIssed Noise) - Chyba, ve které detektor vůbec nezaregistruje pauzu mezi slovy.
  - OVF (OVerlap at the Front) - Jako tuto chybu označujeme, když šumový úsek předcházející řečové aktivitě je detekován jako její součást.
  - OVB (OVerlap at the Back) - Část šumu, která následuje úsek s řečovou aktivitou, je detekován jako její součást. Tato chyba je také občas označována jako OVER [28].



Obrázek 5.2: chyby typu ERP - a) NDS b) MIN c) OVF d) OVB

5.2 Výsledky provedených experimentů

V této části se nachází výsledky experimentů provedených nad databázemi TIMIT a QUT-TIMIT.

5.2.1 TIMIT

Tato část experimentů byla provedena nad databází TIMIT. Část experimentů se zabývá srovnáním CNN s DNN. V další částech je zkoumán vliv počtu zřetězených příznaků, nebo vliv struktury neuronové sítě na chybovost detektoru.
První provedený experiment byl proveden za účelem srovnání detektorů řečové aktivity fungujících na hlubokých neuronových sítích s detektory pracujícími s konvolučními neuronovými sítěmi.
Následující sekce popisuje nastavení hyperparametrů modelů a použité příznaky

- **DNN** - Při práci s hlubokou neuronovou sítí jsou použity příznaky typu MFCC o délce 13 koeficientů. Pro zachycení časového kontextu jsou zřetězeny příznaky 11 po sobě jdoucích segmentů (pět z každé strany). Síť se skládá z pěti skrytých vrstev po 1024 neuronech a výstupní Softmax vrstvy.
- **CNN** - U konvolučních neuronových sítí jsou použity LMFB příznaky. Pro srovnání jsou použity dva běžně používané rozměry LMFB příznaků,

23 a 40. Vstupem do konvolučních vrstev jsou opět zřetězené příznaky 11 po sobě jdoucích segmentů.
Síť má na vstupu dvě konvoluční vrstvy o rozdílných počtech filtrů, 128 a 256. Obě konvoluční vrstvy obsahují Maxpoolingovou část. Výstup z konvolučních vrstev putuje dále do plně propojené části. Do sítě skládající se ze pěti skrytých vrstev po 1024 neuronech, zakončených Softmax vrstvou. Tato část sítě má tedy stejnou konfigurací jako síť použitá u DNN modelu.

Druh sítě	Použitý druh příznaku	ERR	ERS	ERP
DNN	MFCC	2.350	1.018	1.332
CNN	LMFB - 23	2.224	1.049	1.175
CNN	LMFB - 40	2.255	1.055	1.198

Tabulka 5.1: Srovnání CNN a DNN sítě

Daná konfigurace vykazuje relativní zlepšení chybovosti o 5.4 % pro 23 dimenzionální banku filtrů a 4 % pro 40 dimenzionální banku filtrů oproti detektoru využívajícím MFCC příznaky. Tato úroveň relativního zlepšení nedosahuje úrovní dosažených v [4], kde se relativní zlepšení získané využitím CNN místo DNN pohybuje mezi 10 - 25 %. Cílem dalších experimentů tedy bude zkoumání hyperparametrů sítě s cílem dosáhnout co nejmenší chybovosti.

Dalším provedeným experimentem je studování vlivu dostupného časového kontextu na efektivitu detektoru. Tento experiment je prováděn nad konvoluční sítí, skládající se ze dvou konvolučních vrstev napojených na plně propojenou část se třemi skrytými vrstvami po 256 neuronech. Konvoluční vrstvy obsahují 128 a 256 filtrů a Maxpool vrstvy s velikostí okna a kroku dva. Kontrolovaným parametrem je parametr Splice, který udává, kolik příznaků sousedních segmentů je zřetězeno z obou stran s příznaky aktuálního segmentu.

Splice	ERR	ERS	ERP
0	7.089	3.880	3.209
1	4.167	2.102	2.065
2	3.333	1.452	1.881
3	2.732	1.218	1.514
4	2.364	1.037	1.328
5	2.170	0.966	1.204
6	1.978	0.859	1.119
7	1.815	0.832	0.983
8	1.797	0.799	0.997
9	1.764	0.807	0.956
10	1.712	0.779	0.933

Tabulka 5.2: VAD - Vliv počtu zřetězených příznaků

Výsledky ukazují, že chybovost detektoru je výrazně ovlivněna množstvím dostupného časového kontextu. V případě, kdy detektor měl k dispozici pouze velmi omezený počet příznaků, je vykazován větší počet chyb typu ERS. Se zvedajícím se množstvím časového kontextu však míra chybně detekovaných úseků řeči zlepšuje rychleji, než míra chybně detekovaných úseků pauz. To může být dáno strukturou použité databáze, ve které je kvůli charakteru jednotlivých nahrávek větší prostor pro chyby v částech s aktivním mluvčím.

Dalším parametrem, jehož vliv na efektivitu detektoru je zkoumán, je počet filtrů v konvolučních vrstvách. Počet filtrů v první vrstvě je označen jako N_1 a počet filtrů v druhé vrstvě jako N_2. V konvolučních vrstvách je aplikovaná Maxpool funkce s velikostí okna dva. Vstupem do konvoluční sítě je 11 zřetězených příznaků typu LMFB s dimenzí 23. Podléhající neuronová síť obsahuje tři skryté vrstvy po 256 neuronech a Softmax vrstvu na výstupu. Experiment byl rozdělen na dvě části,ve kterých je vždy zafixován počet filtrů v jedné vrstvě a je zkoumán vliv změny počtu filtrů ve druhé vrstvě.

N_1/N_2	ERR	ERS	ERP	N_1/N_2	ERR	ERS	ERP
16/256	2.325	1.026	1.299	256/16	2.228	1.014	1.214
32/256	2.305	1.033	1.272	256/32	2.170	1.004	1.167
64/256	2.197	0.983	1.214	256/64	2.154	0.973	1.181
128/256	2.170	0.966	1.204	256/256	2.168	0.979	1.189
512/256	2.214	1.030	1.183	256/512	2.255	1.035	1.220

Tabulka 5.3: VAD - Vliv počtu filtrů v konvolučních vrstvách

Z dosažených hodnot je vidět, že počet filtrů použitých v konvoluční neuronové síti má poměrně malý vliv na výsledné chybovosti. Pro detekci řečové aktivity bez přidaného šumu, úlohy s pouze dvěma výstupními třídami tedy není potřeba vysoký počet filtrů.

Následující experiment ukazuje vliv typu poolingové funkce na hodnocené metriky. Srovnány jsou dva nejčastěji používané typy poolingových vrstev. Maxpool a Avgpool. Zbytek sítě je nastaven podle předchozích experimentů. Tedy dvě konvoluční vrstvy, velikost poolingového okna 2, tři skryté vrstvy po 256 neuronech a Softmax vrstva na výstupu. Následující tabulka porovnává vliv poolingové funkce.

Pooling	ERR	ERS	ERP
Max	2.170	0.966	1.204
Avg	2.263	1.043	1.220

Tabulka 5.4: VAD - Vliv typu poolingové vrstvy

Výsledky potvrzují, že poolingová vrstva typu Maxpool je pro detekci

řečové aktivity vhodnější, než vrstva Avgpool.

Dále se v této práci zabýváme vlivem hyperparametrů hluboké neuronové sítě za konvolučními vrstvami. Následující experiment zkoumá vliv počtu skrytých vrstev na výsledné chybovosti. Součástí experimentu je i model konvoluční sítě bez skrytých vrstev. V tomto modelu po konvolučních vrstvách následuje pouze jedna afinní vrstva. Zbytek parametrů je nastaven obdobně jako v předchozích experimentech. Vstupem detektoru je 11 zřetězených příznaků. Každá přidaná skrytá vrstva obsahuje 256 neuronů.

Počet skrytých vrstev	ERR	ERS	ERP
0	2.245	1.033	1.212
1	2.253	1.086	1.167
2	2.177	0.999	1.178
3	2.170	0.966	1.204
4	2.213	1.008	1.206
5	2.174	1.022	1.152

Tabulka 5.5: VAD - Vliv počtu skrytých vrstev

Dosažené hodnoty ukazují, že pro detektor řečové aktivity s konvolučními neuronovými sítěmi nám stačí pracovat s omezeným počtem skrytých vrstev. Další zvedání počtu skrytých vrstev již efektivitu detektoru nezlepšuje.
Druhým zkoumaným hyperparametrem plně propojených vrstev je počet neuronů ve skryté vrstvě. Ostatní parametry sítě jsou pro všechny měření konstantní: 11 zřetězených příznaků a 3 skryté vrstvy.

Počet neuronů na skrytou vrstvu	ERR	ERS	ERP
64	2.197	1.022	1.175
128	2.199	1.033	1.167
256	2.170	0.966	1.204
512	2.269	1.053	1.216
1024	2.203	1.016	1.187
2048	2.261	1.033	1.229

Tabulka 5.6: VAD - Vliv počtu neuronů ve skrytých vrstvách

Ze zjištěných hodnot vyplývá, že vyšší počet neuronů efektivitu detektoru nijak nezvyšuje. Vyšší počet neuronů by pravděpodobně našel spíš uplatnění ve složitějších úlohách jako je klasifikace fónů, nebo detekce řečové aktivity v zašuměném prostředí.

5.2.2 QUT-NOISE-TIMIT

V této části se zabýváme schopností detektoru pracovat s řečovou databází QUT-NOISE-TIMIT vytvořenou z databáze TIMIT přidáním šumové databáze QUT.
Prvním provedeným experimentem s databází QUT-TIMIT je pozorování, jak si detektor natrénovaný na nezašuměném TIMITu poradí se zašuměnými daty.
Parametry sítě na které byl detektor trénován jsou následující: 2 konvoluční vrstvy s Maxpool o velikosti 2, dvě skryté vrstvy o 264 neuronech a pět segmentů zřetězených z obou stran.

Scénář	SNR [dB]	ERR	ERS	ERP
TIMIT	-	2.154	0.973	1.181
HOME-LIVINGB	15	54.540	0.018	54.522

Tabulka 5.7: VAD - robustnost detektoru vůči přidanému šumu

Při přítomnosti šumu považuje detektor většinu šumu za řečový signál a výsledná chybovost výrazně klesá.
Dalším provedeným experimentem s databází QUT-TIMIT je zkoumání efektivity detektorů natrénovaných a otestovaných na datech ze stejného prostředí, se stejnou úrovní přidaného šumu. Pro srovnání je tento experiment proveden na několika dostupných šumových úrovních pro několik druhů prostředí. Druhy prostředí pro tento experiment byly vybrány pro co největší předpokládanou rozmanitost charakteru přidaného šumu. Kvůli omezenému rozsahu šumové databáze pro specifické podmínky probíhá trénování na omezeném souboru 50 promluv a trénování na odlišném souboru 50 promluv nahraných při stejných podmínkách.
Použitá konvoluční síť obsahuje dvě konvoluční a dvě skryté vrstvy. Každá skrytá vrstva obsahuje 256 neuronů a vstupem do sítě je 11 zřetězených segmentů.

Scénář	SNR [dB]	ERR	ERS	ERP
HOME-LIVINGB	15	3.404	1.995	1.408
HOME-LIVINGB	5	11.440	4.284	7.155
HOME-LIVINGB	0	14.660	6.498	8.162
STREET-CITY	15	2.472	1.265	1.207
STREET-CITY	5	4.586	2.245	2.340
STREET-CITY	0	9.173	2.697	5.620
REVERB-CARPARK	15	2.309	0.742	1.658
REVERB-CARPARK	5	4.406	1.574	2.833
REVERB-CARPARK	0	10.524	2.315	8.208

Tabulka 5.8: Chybovost přizpůsobeného detektoru

Se zvyšujícíse úrovní šumu se snižuje efektivita detektoru. Nejhorších výsledků dosáhl detektor v prostředí HOME-LIVINGB, jenž se typicky skládá z přidaných zvuků podobných lidské řeči (dětský pláč, zapnutá televize atd.). To je i vidět na mnohem větších hodnotách chyb typu ERS, kde jsou části promluv nesprávně klasifikovány jako zvuky v pozadí. V prostředích, kde přidaný šum nemá tento specifický charakter si detektor počínal lépe, Vzhledem k malému počtu trénovacích promluv je při vyhodnocení také nutné brát v potaz vliv overfittingu.

5.2.3 Univerzální detektor

Druhou částí je experimentu je univerzální detektor natrénovaný na směsi dat vytvořených z několika různých druhů prostředí a úrovní šumu. Detektor byl trénován na směsi signálů z následujících prostředí

Prostředí	SNR
REVERB-POOL	0
REVERB-CARPARKB	0
CAFE-CAFE	5
STREET-KG	5
CAR-WINUPB	10
CAFE-FOODCOURTB	10
HOME-KITCHEN	15
STREET-CITY	15

Struktura sítě použitá pro tento detektor obsahuje dvě konvoluční vrstvy a dvě skryté vrstvy s 256 neurony. Vstupem do sítě je 11 zřetězených příznaků

(splice = 5). Tento detektor je pak aplikován na testovací soubor vytvořený z nezašuměné databáze TIMIT a na testovací soubory vytvořené z jednoho typu prostředí s různými úrovněmi SNR.

Scénář	SNR	ERR	ERS	ERP
TIMIT	-	9.574	8.231	1.342
HOME-LIVINGB	-10	38.627	3.214	35.413
HOME-LIVINGB	0	22.579	3.168	19.411
HOME-LIVINGB	5	15.568	2.911	12.656
HOME-LIVINGB	15	5.508	1.748	3.760

Tabulka 5.9: Chybovost univerzálního detektoru

Z hodnot v tabulce je vidět, že univerzální detektor při ekvivalentním SNR zaostává o zhruba 5 až 10 % za přizpůsobenými detektory. Chybovost zde také může být ovlivněna poměrně malou velikostí testovacích souborů. Efektivitu detektoru by jistě šlo zlepšit použitím mnohem většího testovacího souboru, který by zahrnoval i další typy a úrovně šumu, nebo využitím sítě se složitější strukturou.

5.3 Další realizované detektory

V této části jsou pro možnost porovnání prezentované efektivity dalších, dříve realizovaných detektorů řečové aktivity nad databází TIMIT. Pro srovnání může sloužit například detektor navržený v [29], pracující na bázi odhadu fraktální dimenze pro jednotlivé časové úseky. Na nezašuměném TIMITu dosahuje chybovostí okolo 10 %. Při přidání aditivního šumu se chybovost pomalu zvyšuje, až při šumu se SNR 5 dB dosahuje 15 %. Dalším příkladem může být detektor navržený v [30], který pracuje na bázi k-means algoritmu a dosahuje nad minimálním množstvím šumu chybovosti okolo 4 %. Pro úrovně SNR 15 a 5 dB dosahuje respektivních chybovostí 10 a 20 %.

Kapitola 6

Závěr

Cílem této práce bylo vybudovat detektor řečové aktivity s pokročilými strukturami neuronových sítí a ověřit jeho efektivitu pro různé druhy prostředí a úrovní šumu. Práce popisuje základní poznatky ze zpracování řečového signálu a strojového učení. Součástí práce je i souhrn využití metod strojového učení v úlohách rozpoznávání řeči a detekce řečové aktivity.
Za pomoci sady nástrojů pro rozpoznávání řeči Kaldi byl realizován detektor pracující na bázi konvolučních neuronových sítí. Realizace byla vedená jako jeden z ukázkových skriptů Kaldi.
Experimentálně byl zkoumán vliv hyperparametrů sítě na efektivitu detektoru. Zároveň byla efektivita navrženého detektoru srovnána s efektivitou detektoru pracujícím s hlubokými neuronovými sítěmi. Podle zvolené konfigurace a typu úlohy dosahoval navržený detektor 5 % relativního zlepšení oproti detektorům s hlubokými neuronovými sítěmi. Ve specifických konfiguracích se podařilo klesnout s chybovostí až na 1.7 %.
Navržený detektor byl dále aplikován na databázi QUT-TIMIT a jeho účinnost byla experimentálně ověřena v situacích, kde byl detektor přizpůsoben zvolenému prostředí nebo úrovni šumu pozadí. Dále byl vytvořen univerzální detektor natrénovaný a hodnoceny na kombinaci signálů s různými typy a úrovněmi šumu pozadí. Při otestování tohoto detektoru nad databází TIMIT bylo dosaženo chybovosti necelých 10 %.
V budoucnu by tento detektor mohl být rozšířen o další vhodně zpracované řečové databáze. Místo práce s příznaky je možné pracovat přímo se vzorky signálu a konvoluční síť aplikovat způsobem end-to-end. Zároveň by mohla být zkoumána efektivita detektoru řečové aktivity, který by byl postavený na rekurentních neuronových sítích.

Bibliografie

[1] J. Psutka, L. Müller, J. Matoušek a V. Radová. *Mluvíme s počítačem česky*. Prague: Academia, 2006, s. 752. ISBN: 80-200-1309-1.

[2] P. Pollak. *Slidy k předmětu Zpracování řeči*. ČVUT FEL. 2019.

[3] H. M. Fayek. *Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between*. 2016. URL: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html.

[4] S. Thomas, S. Ganapathy, G. Saon a H. Soltau. "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, s. 2519–2523.

[5] A. d. S. P. Soares, W. D. Parreira, E. G. Souza, S. J. M. de Almeida, C. M. Diniz, C. D. Nascimento a M. F. Stigger. "Energy-based voice activity detection algorithm using Gaussian and Cauchy kernels". In: *2018 IEEE 9th Latin American Symposium on Circuits Systems (LASCAS)*. 2018, s. 1–4. DOI: 10.1109/LASCAS.2018.8399936.

[6] F. Beritelli, S. Casale a A. Cavallaero. "A robust voice activity detector for wireless communications using soft computing". In: *IEEE Journal on Selected Areas in Communications* 16.9 (1998), s. 1818–1829.

[7] I.C. Yoo a D. Yook. "Robust Voice Activity Detection Using the Spectral Peaks of Vowel Sounds". In: *Etri Journal - ETRI J* 31 (2009), s. 451–453. DOI: 10.4218/etrij.09.0209.0104.

[8] Ch. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007. ISBN: 0387310738.

[9] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath a B. Kingsbury. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *Signal Processing Magazine, IEEE* 29 (2012). DOI: 10.1109/MSP.2012.2205597.

[10] D. Yu a L. Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Signals and Communication Technology. London: Springer, 2015. ISBN: 978-1-4471-5778-6. DOI: 10.1007/978-1-4471-5779-3.

[11] I: Goodfellow, Y. Bengio a Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[12] T. Parcollet, M. Morchid a G. Linarès. "E2E-SINCNET: Toward Fully End-To-End Speech Recognition". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, s. 7714–7718.

[13] H. Bourlard a N. Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. 1994. DOI: 10.1007/978-1-4615-3210-1.

[14] S. Watanabe, T. Hori, S. Kim, J. R. Hershey a T. Hayashi. "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition". In: *IEEE Journal of Selected Topics in Signal Processing* 11.8 (2017), s. 1240–1253.

[15] J. Cui, C. Weng, G. Wang, J. Wang, P. Wang, C. Yu, D. Su a D. Yu. "Improving Attention-Based End-to-End ASR Systems with Sequence-Based Loss Functions". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. 2018, s. 353–360.

[16] A. Mohamed, G. Dahl a G. Hinton. "Deep Belief Networks for phone recognition". In: *Science* 4 (2010).

[17] G. E. Dahl, D. Yu, L. Deng a A. Acero. "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1 (2012), s. 30–42.

[18] A. Sehgal a N. Kehtarnavaz. "A Convolutional Neural Network Smartphone App for Real-Time Voice Activity Detection". In: *IEEE Access* 6 (2018), s. 9017–9026.

[19] G. Saon, S. Thomas, H. Soltau, S. Ganapathy a B. Kingsbury. "The IBM speech activity detection system for the DARPA RATS program". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (2013), s. 3497–3501.

[20] D. Palaz, Magimai-Doss M. a Collobert R. "Analysis of CNN-based speech recognition system using raw speech as input". In: *INTERSPEECH*. 2015.

[21] H. Soltau, H. Kuo, L. Mangu, G. Saon a T. Beran. "Neural network acoustic models for the DARPA RATS program". In: (2013), s. 3092–3096.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y Qian, P. Schwarz, J. Silovsky, G. Stemmer a K. Vesely. "The Kaldi Speech Recognition Toolkit". In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Catalog No.: CFP11SRW-USB. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, 2011.

[23] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus a D. Pallett. "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1". In: *NASA STI/Recon Technical Report N* 93 (1993), s. 27403.

[24] D. Dean, S. Sridharan, R. Vogt a M. Mason. "The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms." In: 2010, s. 3110–3113.

[25] M. Lakosil. *Detektor řečové aktivity na bázi DNN - Diplomová práce*. Praha: ČVUT, 2017.

[26] M. Vaclavik. *Vadcrit*. ČVUT - FEL K331. 2002.

[27] J. Rosca, R. Balan, N. P. Fan, C. Beaugeant a V. Gilg. "Multichannel voice detection in adverse environments". In: *2002 11th European Signal Processing Conference*. 2002, s. 1–4.

[28] X. Li, R. Horaud, L. Girin a S. Gannot. "Voice activity detection based on statistical likelihood ratio with adaptive thresholding". In: *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2016, s. 1–5.

[29] Z. Ali a M. Talha. "Innovative Method for Unsupervised Voice Activity Detection and Classification of Audio Segments". In: *IEEE Access* 6 (2018), s. 15494–15504.

[30] M. H. Moattar, M. M. Homayounpour a N. Khademi Kalantari. "A new approach for robust realtime Voice Activity Detection using spectral pattern". In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010, s. 4478–4481.

Příloha A

Obsah CD

Na přiloženém CD se nachází:

- Text diplomové práce ve formátu pdf
- Realizace detektoru řečové aktivity pomocí nástrojů Kaldi