TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Počítačová syntéza řeči pomocí umělých neuronových sítí

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. František Kynych**
Vedoucí práce: Ing. Petr Červa, Ph.D.

Liberec 2020

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání diplomové práce

Počítačová syntéza řeči pomocí umělých neuronových sítí

Jméno a příjmení: **Bc. František Kynych**
Osobní číslo: M18000148
Studijní program: N2612 Elektrotechnika a informatika
Studijní obor: Informační technologie
Zadávající katedra: Ústav informačních technologií a elektroniky
Akademický rok: **2019/2020**

Zásady pro vypracování:

1. Seznamte se s problematikou počítačové syntézy řeči, zejména s metodami využívajícími hluboké neuronové sítě.
2. Natrénujte na připravené databázi pomocí neuronových sítí model syntetické češtiny pro mužský a ženský hlas. Hyperparametry zvolené architektury neuronové sítě přitom optimalizujte s ohledem na co nejvyšší kvalitu syntetické řeči a rychlost trénování.
3. Porovnejte kvalitu vytvořeného syntezátoru s dostupnými systémy pro daný jazyk a s vybranými referenčními systémy pro další jazyky (např. pro angličtinu).
4. Vytvořte demonstrační webovou aplikaci, která umožní generovat řečový signál ze zadaného textu.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40-50
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

[1] NOUZA, Jan, ed., KOLDOVSKÝ, Zbyněk, ed. a VÍCH, Robert, ed. Řeč a počítač: principy hlasové komunikace, úlohy, metody a aplikace: sborník článků. Vyd. 1. Liberec: Technická univerzita v Liberci, 2009. 235 s. ISBN 978-80-7372-548-8.

[2] Shen, Jonathan et al. ?Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018, pp. 4779-4783.

[3] https://heartbeat.fritz.ai/a-2019-guide-to-speech-synthesis-with-deep-learning-630afcafb9dd

Vedoucí práce: Ing. Petr Červa, Ph.D.
Ústav informačních technologií a elektroniky

Datum zadání práce: 9. října 2019
Předpokládaný termín odevzdání: 18. května 2020

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 18. října 2019

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

18. května 2020 Bc. František Kynych

Počítačová syntéza řeči pomocí umělých neuronových sítí

Abstrakt

Tato diplomová práce se zabývá syntézou řeči pomocí neuronových sítí. Cílem bylo prozkoumání a ověření současných přístupů využívajících neuronové sítě a pomocí nejlepší architektury natrénování mužského a ženského hlasu. Dále porovnání s komerčními systémy a vytvoření demonstrační webové aplikace.

Pro experimenty byly vybrány DeepVoice 3, Tacotron 2 a WaveGlow architektury. Nejsrozumitelnější řeči dosahoval mužský hlas Tacotron 2 a WaveGlow architektury, proto byl vybrán pro porovnání s komerčními systémy. Porovnávání probíhalo prostřednictvím poslechových testů, pro které bylo vytvořeno prostředí v demonstrační webové aplikaci. Hodnocení se zúčastnilo 56 lidí a celkem bylo ohodnoceno 1060 nahrávek od každého systému. Výsledek této diplomové práce byl srovnatelný s komerčně používanými systémy a překonal standardní systém Googlu, který nevyužívá neuronové sítě. Nad rámec zadání byla řešena fonetická transkripce pro dosažení lepší kvality syntetizované řeči a dále byl Tacotron 2 model rozšířen o vektory mluvčího (tzv. X-Vektory), díky kterým se podařilo měnit hlas dle pohlaví osoby přivedeného vektoru.

Klíčová slova: syntéza řeči, neuronové sítě, syntéza řeči pro více mluvčí, Tacotron 2, WaveGlow

Computer Speech Synthesis Using Artificial Neural Networks

Abstract

This diploma thesis is focused on speech synthesis using neural networks. The goal was to explore current approaches using neural networks and to train male and female voices using the best architecture. Then compare it with commercial systems and create a web demo application.

DeepVoice 3, Tacotron 2 and WaveGlow architectures were selected for the experiments. The most intelligible speech was achieved by the male voice of the Tacotron 2 and WaveGlow architecture, so it was chosen for comparison with commercial systems. The comparison was performed through listening tests, for which an environment was created in a demonstration web application. The evaluation was attended by 56 people and a total of 1,060 recordings from each system were evaluated. The result of this diploma thesis was comparable to commercially used systems and surpassed the standard Google system, which does not use neural networks. In addition to the assignment, phonetic transcription was solved to achieve better quality of synthesized speech, and the Tacotron 2 model was extended by speaker vectors (so-called X-Vectors), thanks to which it was possible to change the voice according to the gender of the person of the input vector.

Keywords: speech synthesis, neural networks, speaker independent speech synthesis, Tacotron 2, WaveGlow

Poděkování

Rád bych poděkoval panu Ing. Petru Červovi, Ph.D. za odborné vedení, poskytnutá data a cenné rady během řešení této práce.

Obsah

Seznam tabulek

Seznam zkratek

ARSG	Attention-based Recurrent Sequence Generator
DCTTS	Deep Convolutional TTS
ELU	Exponential Linear Unit
G2P	Grapheme-to-Phoneme
GNMT	Google's Neural Machine Translation
GRU	Gated Recurrent Unit
GST	Global Style Token
LSTM	Long Short-Term Memory
MOS	Mean Opinion Score
ReLU	Rectified Linear Unit
Seq2Seq	Sequence-to-Sequence
SSRN	Spectrogram Super-Resolution Network
STFT	Short-time Fourier Transform
TDNN	Time Delay Neural Network
TTS	Text-to-Speech
WER	Word Error Rate

Úvod

Tato diplomová práce se zabývá syntézou řeči s využitím neuronových sítí. Syntéza řeči je důležitým prvkem pro hlasovou komunikaci počítače nebo jiného zařízení s uživatelem. Jedná se o uměle generovanou řeč z textu, která přenáší informace uživateli s cílem srozumitelného a přirozeného způsobu předání.

V současné době podporuje syntézu řeči velké množství elektronických zařízení, která používáme každý den. Většina operačních systémů obsahuje funkci předčítání, jež pomáhá například osobám se zrakovým postižením, a k tomu využívá text-to-speech (TTS) systém. Dále se často setkáváme se syntetizovaným hlasem například v různých mobilních aplikacích, u domácích hlasových asistentů, v multimediálních systémech moderních automobilů a na mnoha dalších místech. Neuronové sítě přinesly do této oblasti výrazné zlepšení, tyto hlasy se často stávají nerozeznatelnými od řeči skutečných lidí. V budoucnu bude pravděpodobně možné poslouchat audio knihu námi zvoleným syntetizovaným hlasem.

Nejvíce používané metody syntézy řeči využívaly parametrický nebo konkatenanční přístup. Řeč je složena z různých frekvencí a parametrický přístup (též označovaný jako formantový) využívá těchto vlastností. Uplatňuje například LPC, formantový nebo sinusový model. LPC syntéza modeluje přenosovou charakteristiku celého řečového traktu, formantový umožňuje měnit jednotlivé parametry a u sinusového modelu jsou vlastnosti řečového traktu zahrnuty v parametrech modelu [1]. Tento přístup umožňuje například jednodušší změnu hlasu na opačné pohlaví. Konkatenanční syntéza využívá velké databáze krátkých úseků řeči (difónů a trifónů), které poté řetězí, a tím vytvoří libovolnou řeč. Konkatenanční syntéza dosahuje lepší kvality výstupu než parametrická, ale stále je zde znát, že se jedná o uměle

vytvořenou řeč.

V současné době jsou aplikovány přístupy využívající neuronových sítí, které jsou využity i v této práci. Neuronové sítě překonaly dříve používané metody a také výrazně zvýšily kvalitu syntetizované řeči. U poslechových testů provedených Tacotron 2 týmem (viz [2]) se testovala podobnost mezi syntetizovanou řečí a řečí osoby z trénovacích dat. Většinou byly tyto hlasy hodnoceny jako téměř stejné a pouze s malým rozdílem byl více preferován hlas skutečné osoby. Nevýhodou oproti předchozím přístupům je vyšší výpočetní náročnost.

Syntézu řeči pro český jazyk poskytuje několik komerčních systémů, mezi které patří například sytém od společnosti Google, s nímž je možné syntetizovat ženský hlas s využitím neuronových sítí v Google Cloud konzoli [3]. Další systém poskytuje společnost Microsoft prostřednictvím služby Microsoft Azure Cognitive Services [4], kde je možné syntetizovat mužský hlas, ale pro český jazyk zatím stále používá parametrický a konkatenanční přístup. Dále je možné syntetizovat řeč využitím systému od společnosti SpeechTech [5], která nabízí více hlasů různé kvality a také další služby například pro vytvoření požadovaného hlasu nebo přirozenou syntézu pro konkrétní oblast použití. Všechny tyto služby budou porovnány s výsledkem této práce pomocí poslechových testů.

Motivací pro řešení práce bylo získání zkušeností s moderními metodami pro syntézu řeči. V Laboratoři počítačového zpracování řeči (SPEECHLAB) se zatím nikdo nezabýval využitím neuronových sítí pro tuto úlohu. Předchozí práce se věnovaly parametrické [6] a zřetězené [7] syntéze, ale výsledná řeč těchto prací nebyla přirozená.

Cílem je prozkoumat a ověřit architektury pro syntézu řeči využívající neuronové sítě a z těchto architektur vybrat tu s nejlepšími výsledky. Dále pro ni nalézt optimální hyperparametry a z dostupných dat pomocí ní natrénovat model mužského a ženského hlasu pro český jazyk. Pro tuto práci nejsou dostupné profesionálně zaznamenané nahrávky pro trénování vybraných systémů, neboť jejich vytvoření je nad rámec této práce. Nejlepší natrénovaný model bude porovnáván pomocí poslechových testů s komerčními systémy poskytujícími syntézu českého jazyka. Následně

budou výsledky práce demonstrovány prostřednictvím vytvořené webové aplikace.

Nad rámec zadání práce je kvůli lepší výslovnosti modelu řešena rovněž fonetická transkripce a syntéza pro více mluvčích, s níž je možné měnit výstupní hlas s použitím vektoru mluvčího.

První kapitola práce seznámí čtenáře se současně používanými architekturami neuronových sítí pro úlohu syntézy řeči a zároveň s přístupem syntézy pro více mluvčích. Další kapitola stručně popisuje vybranou architekturu pro fonetickou transkripci. Poté již práce popisuje použité řešení pro syntézu jednoho a více mluvčích, použitá data a experimenty provedené s vybranými architekturami neuronových sítí. Poslední kapitola popisuje vytvořenou webovou aplikaci pro demonstraci syntézy řeči a také pro porovnání nejlepšího systému této práce s komerčně používanými systémy pro syntézu českého jazyka, kde jsou i shrnuty výsledky hodnocení.

1 Architektury pro syntézu řeči využívající neuronové sítě

V této části jsou popsány současně používané a publlikované architektury využívající neuronové sítě pro syntézu řeči. Tyto systémy se často dělí na dva modely, které lze paralelně trénovat. První model většinou převádí vstupní text do mel spektogramu (viz 1.1.1) a druhý převádí mel spektogram do zvukového signálu. Tento přístup je popsán v kapitole 1.1. Kapitola 1.2 popisuje end-to-end systém, u kterého je vstupem text a výstupem zvukový signál.

Práce se primárně zabývá syntézou řeči hlasem jednoho mluvčího (speaker-dependent), ale nad rámec zadání je práce orientována i na syntézu pro více mluvčích (speaker-independent). Syntéza pro více mluvčích používá kromě vstupního textu i vektor mluvčího, který je získán ze systému trénovaného na úloze identifikace mluvčího. Použitý systém pro získání těchto vektorů je popsán v kapitole 1.3.

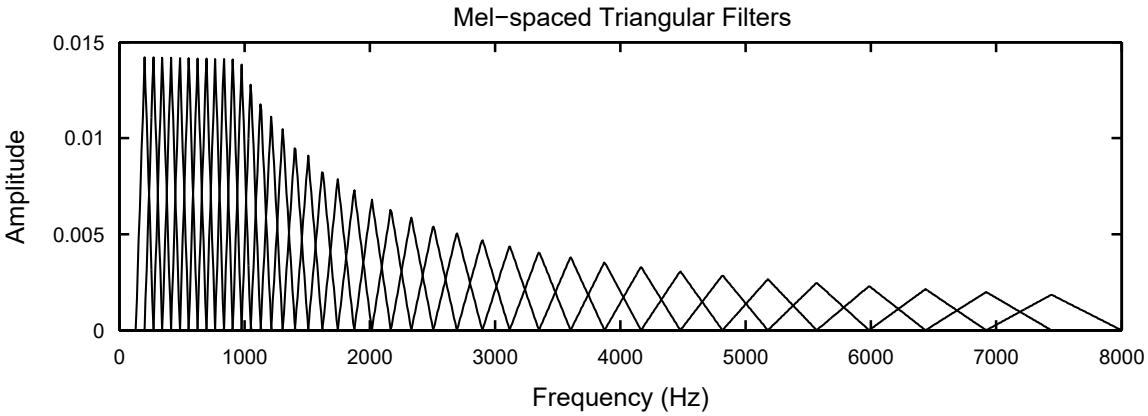
1.1 Použití dvou systému pro syntézu řeči

Současné state-of-the-art architektury jsou rozděleny na dvě části, kde jedna vytváří z textu mel spektogram a druhá predikuje z mel spektogramu zvůkový signál. V následující podkapitolách je popsán mel spektogram a jednotlivé současné přístupy dosahující nejlepších výsledků.

1.1.1 Mel spektogram

Mel spektogram je používán pro zdůraznění detailů v nižších frekvencích a naopak potlačení detailů ve vyšších frekvencích, které u syntézy řeči není potřeba dokonale predikovat. Melova stupnice je nelineární transformace frekvenční osy, která byla experimentálně definována vlastnostmi lidského sluchu. Jednotlivé filtry banky jsou tak rovnoměrně rozprostřeny v daném intervalu tak, že lidskému uchu připadá daná vzdálenost změny frekvece vždy stejná.

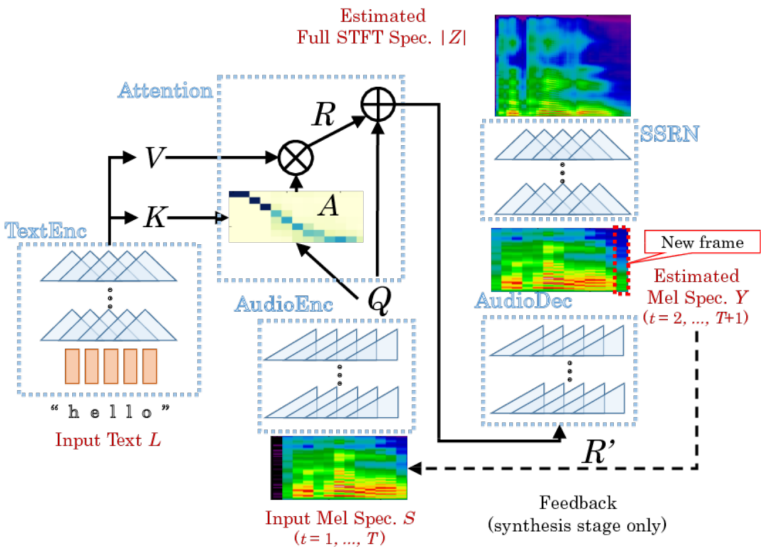
Mel spektogram je získán ze vstupního signálu provedením krátkodobé Fourierovy transformace (STFT) s použitím rámce a posunu o určené délce a použitím vhodné okénkovací funkce. Například u Tacotron 2 modelu (viz 1.1.4) je použit rámec o délce 50 ms, posun 12,5 ms a Hannovo okénko. Po převedení do frekvenční oblasti projde každý rámec bankou filtrů, která využívá melovy stupnice, se zvoleným počtem pásem (u Tacotron 2 modelu je základní hodnota 80 pásem). Na obrázku 1.1 je znázorněna banka mel filtrů.



Obrázek 1.1: Znázornění banky mel filtrů z [8].

1.1.2 Deep Convolutional TTS

Deep Convolutional TTS (DCTTS) [9] je konvoluční model využívající attention mechanismu pro predikci spektogramu z textu. Architektura tohoto modelu je zobrazena na obrázku 1.2. Jedná se o tzv. frontend model, který predikuje spektogram, a pro vytvoření zvukového signálu je potřeba navíc použít libovolný backend model.



Obrázek 1.2: DCTTS architektura z publikace [9].

První částí je textový kodér, ten převede znaky vstupního textu do vektorové reprezentace. Poté je použito několik konvolučních vrstev následovaných tzv. Highway sítí (viz [10]), která reguluje signál procházející touto vrstvou podobně jako LSTM neurony v rekurentních neuronových sítích. Výstupem tohoto bloku jsou matice key (K) a value (V), které jsou dále použity v attention mechanismu. Matice K a V si lze představit jako datovou strukturu obsahující klíč a k němu příslušnou hodnotu.

Druhou částí je audio kodér, který zpracovává mel spektogram vygenerovaný v předchozích krocích sítě. Tato část obsahuje několik konvolučních vrstev následovaných Highway sítí a výstupem je query matice (Q).

Třetí částí je audio dekodér, který využívá výstup attention mechanismu. Attention mechanismus se snaží určit relevanci matice Q, která při trénování vznikla v audio kodéru z požadovaného výstupního mel spektogramu, s maticí K, která vznikla v bloku pro zpracování vstupního textu. Na výstupu attention mechanismu má poté nejvyšší váhu matice V, která patřila k nejrelevatnějšímu klíči K pro dotaz Q v daném kroku generování. Audio dekodér je dále tvořen konvolučními a Highway vrstvami, které produkují mel spektogram.

Poslední částí je Spectrogram Super-resolution Network (SSRN) blok, jenž pomocí konvolučních vrstev a Highway sítí převádí mel spektogram do vylepšeného lineárního spektogramu.

1.1.2.1 Výsledky

K hodnocení syntézy řeči se nejčastěji používá mean opinion score (MOS) metrika, jejíž hodnota se získává z poslechových testů, při nichž uživatelé hodnotí kvalitu dané nahrávky hodnotou 1 až 5. MOS je poté vypočten jako průměr všech hodnocení a vyšší hodnota znamená lepší kvalitu systému.

Výsledky dosažené tímto modelem jsou uvedeny v tabulce 1.1. K hodnocení byla v publikaci použita služba Amazon Mechanical Turk, ve které hodnotilo celkem 31 lidí 20 syntetizovaných vět od každého systému. MOS byl vypočítán využitím crowdMOS toolkitu (viz [11]).

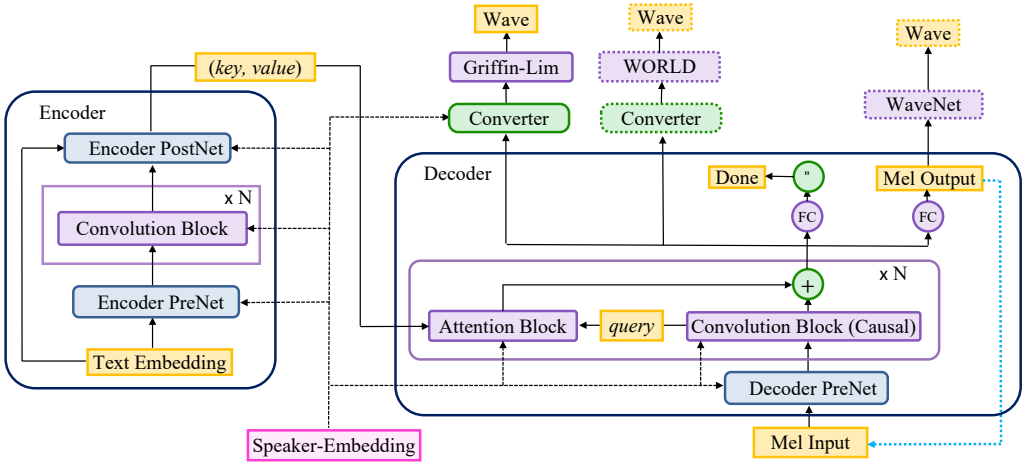
Systém	**MOS**
Tacotron 1	$2,07 \pm 0,62$
DCTTS	$2,71 \pm 0,66$

Tabulka 1.1: MOS ohodnocení DCTTS modelu s 95% intervalem spolehlivosti jednotlivých systémů z publikace [9].

DCTTS systém je porovnáván s první generací Tacotronu a bohužel zatím nebylo provedeno srovnání s druhou generací. Dle publikace [9] lze síť natrénovat zhruba za 15 hodin s použitím dvou běžných herních grafických karet (NVIDIA GeForce GTX 980 Ti), ale výsledná řeč je popsána tak, že není zdaleka dokonalá a je zde stále prostor pro vylepšení např. pomocí dalšího ladění hyperparametrů modelu.

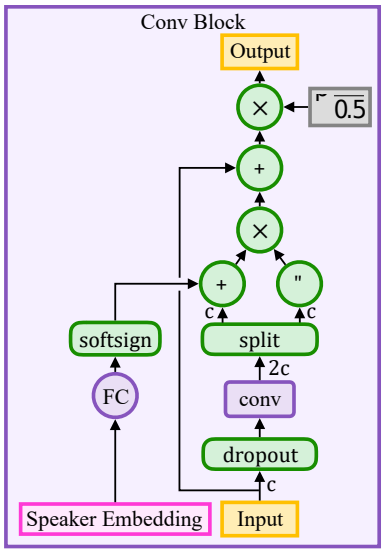
1.1.3 DeepVoice 3

DeepVoice 3 [12] je konvoluční sequence-to-sequence (Seq2Seq) model s attention mechanismem pro syntézu řeči. Tato architektura umožňuje syntézu řeči pro jednoho i pro více mluvčích. Jedná se o frontend model, jehož výstupem může být mel spektogram pro WaveNet [15] syntetizér (viz kapitola 1.1.5), nebo je možné za konvertor připojit Griffin-Lim [13] či WORLD [14] syntetizér. Architektura (viz Obrázek 1.3) se skládá z tří hlavních bloků – z kodéru, dekodéru a konvertoru.



Obrázek 1.3: DeepVoice 3 architektura z publikace [12].

Kodér je konvoluční síť, která vytváří, ze vstupní sekvence znaků, skrytý stav. První je použita embedding vrstva, převádějící znaky do vektorové reprezentace. Tyto vektory jsou vstupem do dopředné neuronové sítě, jejíž výstup je zpracován několika konvolučními bloky (viz Obrázek 1.4), a poté je výstup transformován do vektoru o stejné dimenzi jako embedding vektor, aby tvořil tzv. key attention vektor.



Obrázek 1.4: Struktura DeepVoice 3 konvolučního bloku z publikace [12].

Dekodér generuje výstup ve formě mel spektogramu, kde jsou jednotlivé budoucí kroky podmíněny minulým výstupem tohoto bloku. PreNet síť obsahuje několik vrstev dopředné neuronové sítě s ReLu aktivační funkcí pro zpracování mel spektogramu. Poté je použito několik konvolučních a attention bloků. Konvoluční bloky

vytvářejí tzv. query pro attention blok, který pracuje s výstupem kodéru. Attention mechanismus zde funguje obdobně, jako je popsáno u předchozího modelu. Pro predikci mel spektrogramu je použita dopředná neuronová síť a tento typ sítě je použit i pro binární klasifikátor, který predikuje konec generovaného výstupu.

Konvertor používá skrytý stav poslední skryté vrstvy dekodéru a z něj predikuje příznaky potřebné pro použitý syntetizér. Tento blok se skládá z konvolučeních bloků a dopředných neuronových sítí.

1.1.3.1 Výsledky

V tabulce 1.2 jsou uvedeny MOS hodnocení jednotlivých systémů natrénovaných na stejných datech. Model byl v publikaci natrénován pomocí 20 hodin nahrávek se vzorkovací frekvencí 48 kHz. Díky použití konvolučních vrstev místo rekurentních je možné využít paralelizace při trénování, a tím natrénovat model rychleji než Tacotron 2. Hodnocení probíhalo s využitím služby Amazon Mechanical Turk a bylo k němu využito 100 syntetizovaných vět. MOS byl poté vypočten použitím crowdMOS toolkitu.

Systém	**MOS**
Deep Voice 3 (Griffin-Lim)	$3,62 \pm 0,31$
Deep Voice 3 (WORLD)	$3,63 \pm 0,27$
Deep Voice 3 (WaveNet)	$3,78 \pm 0,30$
Tacotron 2 (WaveNet)	$3,78 \pm 0,34$

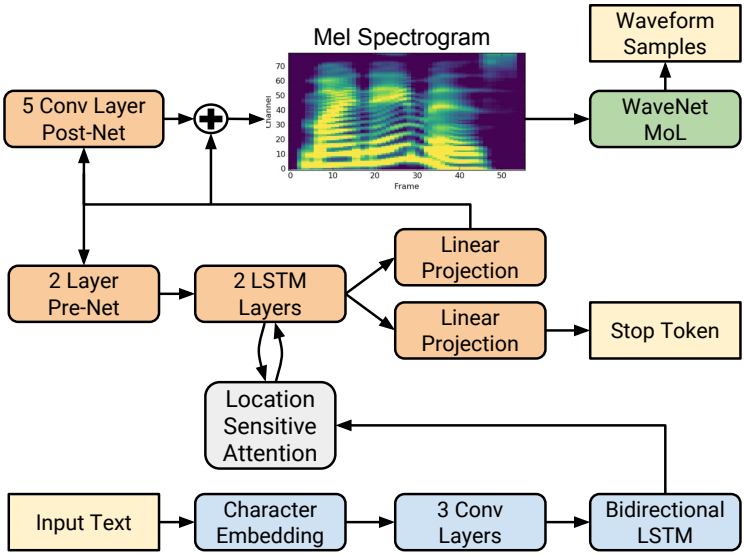
Tabulka 1.2: MOS ohodnocení s 95% intervalem spolehlivosti jednotlivých systémů z publikace [12].

1.1.4 Tacotron 2

Architektura Tacotron 2 [2] je rekurentní neuronová Seq2Seq síť, která z textu na vstupu vytvoří mel spektrogram. Za touto sítí se může následně nacházet libovolný syntentizér generující řečový signál z vytvořeného mel spektogramu. V publikaci [2] se pro generování řečového signálu používá WaveNet model a v této kombinaci

se dosahuje hodnoty 4,53 MOS, eventuálně 4,58 MOS s profesionálně zaznamenanou řečí. Výstup ve tvaru mel spektogramu byl vybrán z důvodu jeho lehkého získání z nahrávky a možnosti separátního trénování sítě Tacotron a WaveNet. Pro trénování jsou z trénovacích dat vytvořeny mel spektogramy s použitím okénka o délce 50 ms, posunem 12,5 ms a aplikováním Hannova okénka. Takto zvolené parametry umožňují nejenom zachycení výslovnosti, ale i hlasitosti, rychlosti a intonace řeči.

Tacotron 2 model (viz Obrázek 1.5) se skládá z kodéru a dekodéru s attention mechanismem. Kodér vytváří ze vstupní sekvence znaků skrytý stav, ten je poté předán do dekodéru, který z tohoto stavu vytváří mel spektogram.



Obrázek 1.5: Tacotron 2 architektura z publikace [2].

Vstupní znaky jsou přivedeny do embedding vrstvy, která vytvoří 512-dimenzionální vektor reprezentující znaky. Tento vektor je poté přiveden na vstup tří 1-dimenzionálních konvolučních vrstev, kde každá obsahuje 512 filtrů o délce 5, díky čemuž jsou do filtru zahrnuty i 2 předchozí a 2 následující znaky. Dále je použita bidirekční rekurentní neuronová síť s 256 neurony v každém směru.

Mezi kodérem a dekodérem se nachází tzv. soft attention mechanismus (viz [16]), který určuje, na kterou část kodéru by se měl dekodér v daném kroku soustředit. Attention mechanismus využívá předchozího skrytého stavu kodéru, předchozího výstupu kodéru a vah, které se v každém kroku generování přizpůsobují.

Dekodér se skládá z tzv. PreNet sítě, která obsahuje dvě vrstvy klasické dopředné neuronové sítě s 256 neurony v každé vrstvě, a použitým dropout algoritmem s pravděpodobností 0,5. Dropout se zde nepoužívá pouze u trénování, ale i během generování řeči a výstup je poté spojen s výstupem attention vrstvy (označovaným jako kontext vektor). Spojené výstupy vrstev jsou vstupem do LSTM rekurentní sítě s 1024 neurony v každé z nich. Výstup je spojen se stejným kontext vektorem a vede do klasické dopředné neuronové sítě s 80 neurony – predikce jednoho kroku mel spektogramu. Používá se také jeden rekurentní neuron, který určuje pravděpodobnost, zda se jednalo o poslední krok. Pro vylepšení kvality se zde nachází PostNet síť, která obsahuje 5 konvolučních vrstev s 512 filtry o délce 5 v každé vrstvě. Mezi vrstvami je použita batch normalizace a tanh aktivační funkce. Poté je použita dopředná neuronová síť s 80 neurony a výstup je přičten k původnímu výstupu dekodéru.

1.1.4.1 Výsledky

Srovnání Tacotron 2 systému s dalšími metodami syntézy řeči je uvedeno v tabulce 1.3. MOS byl v publikaci vypočítán z hodnocení 100 nahrávek, kde každá z nich musela být ohodnocena alespoň 8 lidmi. Z hodnocení je vidět, že se kvalita syntetizovaných nahrávek blíží lidské řeči a často je od ní nerozeznatelná.

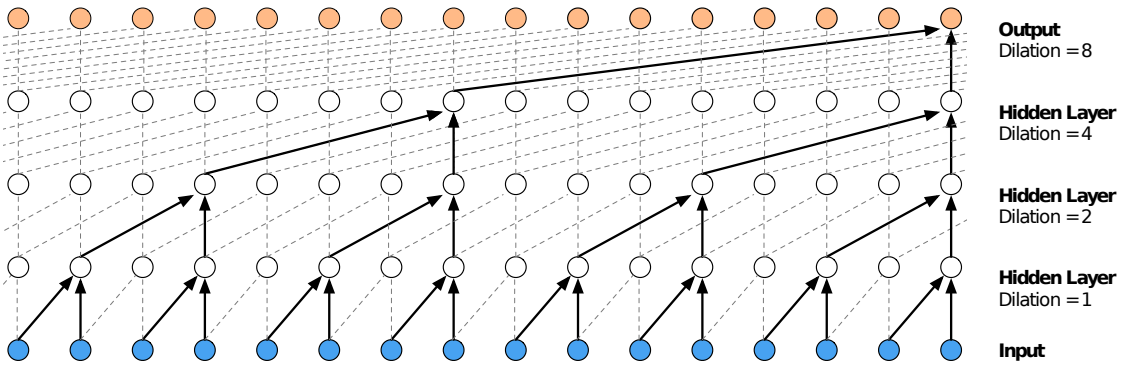
Systém	**MOS**
Parametrický	$3,492 \pm 0,096$
Tacotron 1 (Griffin-Lim)	$4,001 \pm 0,087$
Konkatenační	$4,166 \pm 0,091$
WaveNet	$4,341 \pm 0,051$
Tacotron 2 (WaveNet)	$4,526 \pm 0,066$
Lidská řeč	$4,582 \pm 0,053$

Tabulka 1.3: MOS ohodnocení s 95% intervalem spolehlivosti z publikace [2].

1.1.5 WaveNet

WaveNet model [15] se využívá u řady různých modelů pro zpětnou transformaci mel spektogramu do časové oblasti.

Pro zpracování sekvence dat využívá dilatační konvoluční síť, která se od standardní konvoluční sítě liší vynecháváním hodnot z předchozí vrstvy s určitým krokem (viz Obrázek 1.6), což umožňuje zpracování většího počtu předchozích hodnot ze vstupu do sítě. Použit byl 2× větší krok v každé další vrstvě do daného limitu a poté se krok opakoval (1, 2, 4, …, 512, 1, 2, …).



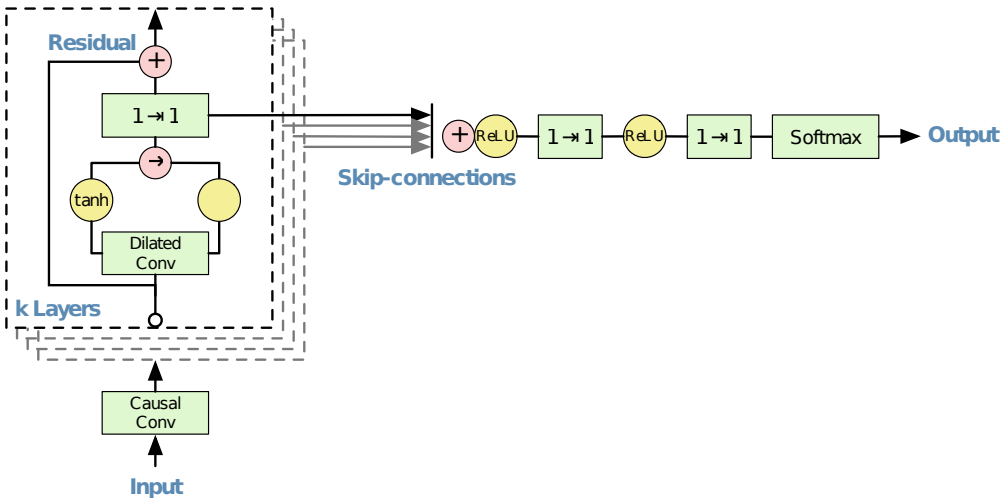
Obrázek 1.6: Vizualizace dilatačních konvolučních vrstev z publikace [15].

Jsou zde použity tzv. Gated Activation Units (viz 1.1), které jsou podobné např. LSTM nebo GRU neuronům v rekurentních neuronových sítích:

$$z = tanh(W_{f,k} * x) \odot \sigma(W_{g,k} * x), \tag{1.1}$$

kde z je výstupem, operátor $*$ zde označuje konvoluci a symbol $\odot$ označuje násobení po prvcích, $\sigma(\cdot)$ je sigmoid funkce, W je naučený konvoluční filtr, u nějž index f značí filter část, index g gate část matice a index k určuje index vrstvy.

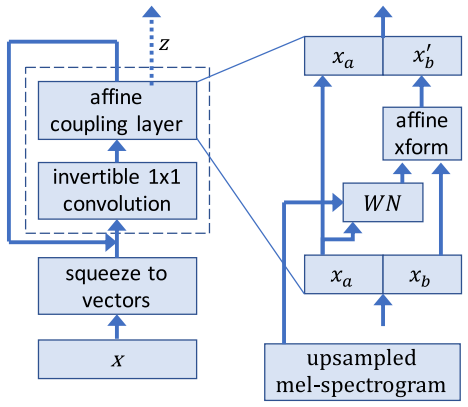
V této architektuře se dále používají tzv. residual (přeskočení jedné vrstvy) a parametrizované skip connections (viz Obrázek 1.7), které umožní přeskočit konvoluční vrstvy a mít přímý vliv na predikovaný výstup.



Obrázek 1.7: Architektura residuálního bloku z publikace [15].

1.1.6 WaveGlow

WaveGlow [17] model byl inspirován Glow [18] a WaveNet modelem. Umožňuje rychlé generování kvalitního zvukového signálu z mel spektogramu. Cílem bylo vytvoření modelu, který nebude příliš složitý a umožní rychlou syntézu paralelním zpracováním vstupu.



Obrázek 1.8: WaveGlow architektura z publikace [17].

V Obrázku 1.8 je zobrazena WaveGlow architektura, ta je založena na tzv. Flow-based modelech, které umožňují invertovat výstupní funkci, a je tak možné síť učit minimalizací záporné logaritmické věrohodnosti. Vstupem je mel spektogram, který je převeden do vektoru, a dále je aplikována invertovatelná 1×1 konvoluce. Poté následuje tzv. affine coupling vrsta (viz [19]), do které vstupuje i mel spektogram.

Tato vrstva zachovává invertibilitu a v bloku WN je možné použít libovolnou transformaci, která nemusí být invertibilní. V tomto modelu jsou ve WN bloku použity dilatační konvoluční vrstvy podobné architektuře WaveNetu.

1.1.6.1 Výsledky

V tabulce 1.4 je zobrazeno porovnání WaveGlow modelu s ostatními systémy, kde tento model překonal kvalitu WaveNet modelu a také rychlost predikování výstupu. Vylepšená architektura WaveNetu – Parallel WaveNet [20] dosahovala rychlosti generování 500 000 vzorků za sekundu na NVIDIA V100 GPU, WaveGlow dosahuje 520 000 vzorků za sekundu.

Systém	MOS
Griffin-Lim	$3,823 \pm 0,1349$
WaveNet	$3,885 \pm 0,1238$
WaveGlow	$3,961 \pm 0,1343$
Lidská řeč	$4,274 \pm 0,1340$

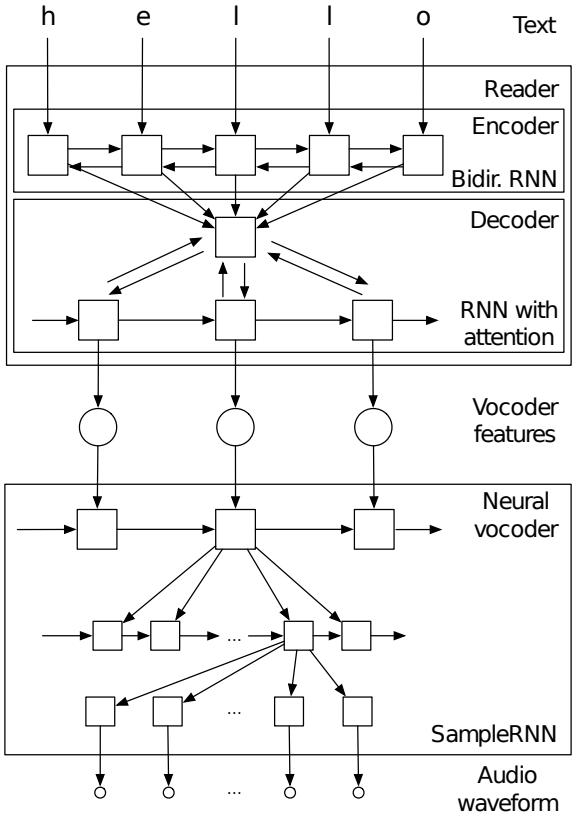
Tabulka 1.4: MOS ohodnocení syntetizérů s 95% intervalem spolehlivosti z publikace [17].

1.2 End-to-end systémy

End-to-end systémy využívají jeden model pro přímou syntézu řeči ze vstupního textu. Uvnitř modelu se i tak nachází fronted, který ze vstupu získá lingvistické příznaky, které jsou poté zpracovány backend částí pro generování signálu. Síť se ale trénuje jako jeden celek.

1.2.1 Char2Wav

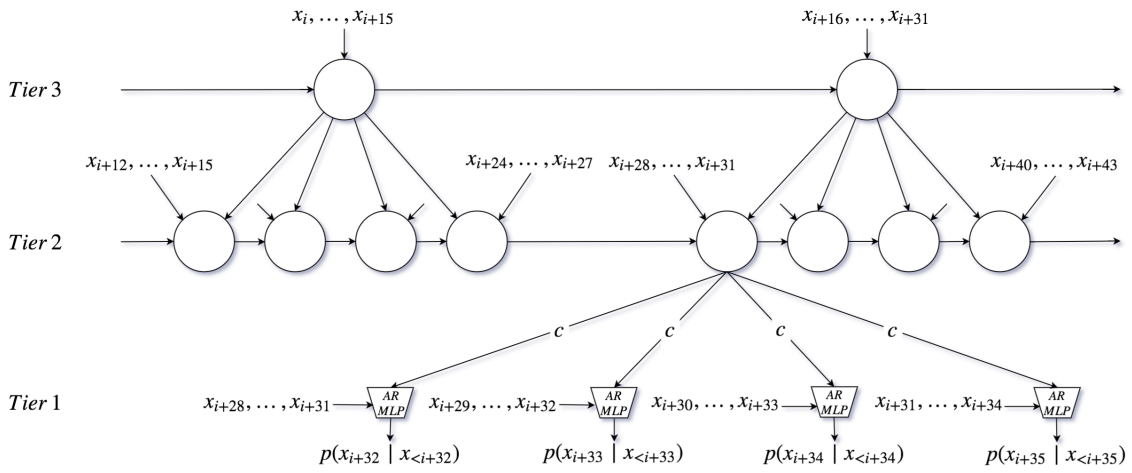
Char2Wav [21] je end-to-end model pro syntézu řeči. Tento model (viz Obrázek 1.9) se skládá ze Seq2Seq architektury s použitím attention mechanismu. Vstupem do modelu je text, případně fonetický přepis, a výstupem jsou příznaky pro syntetizér, jímž je SampleRNN model [22]. Tato architektura byla jedna z prvních, které dokázaly generovat řečový signál přímo z textu.



Obrázek 1.9: Char2Wav architektura z publikace [21].

V reader části je vstupní text převeden pomocí embedding vrstvy do vektorové reprezentace, která je vstupem do Seq2Seq rekurentní neuronové sítě. Použitá Seq2Seq síť využívá attention mechanismu a tento přístup je označován jako attention-based recurrent sequence generator (ARSG, viz [23]), který byl původně využíván pro zpracování řeči. Výstupem jsou příznaky pro backend část.

V backend části je použita SampleRNN síť, která je vhodná pro zpracování sekvenčních dat, jelikož zvládne zpracovávat dlouhodobé závislosti. Využívá hierarchii vrstev rekurentních neuronových sítí (viz Obrázek 1.10), kdy neurony v první vrstvě zpracovávají určité množství vstupních dat. Neurony v další vrstvě obdrží výstup neuronů předchozí vrstvy a navíc zpracovávají menší rozsah vstupních dat než neuron v předchozí vrstvě. Na výstupu je použit vícevrstvý perceptron a funkce softmax.



Obrázek 1.10: SampleRnn architektura z publikace [22].

1.3 Syntéza řeči pro více mluvčích

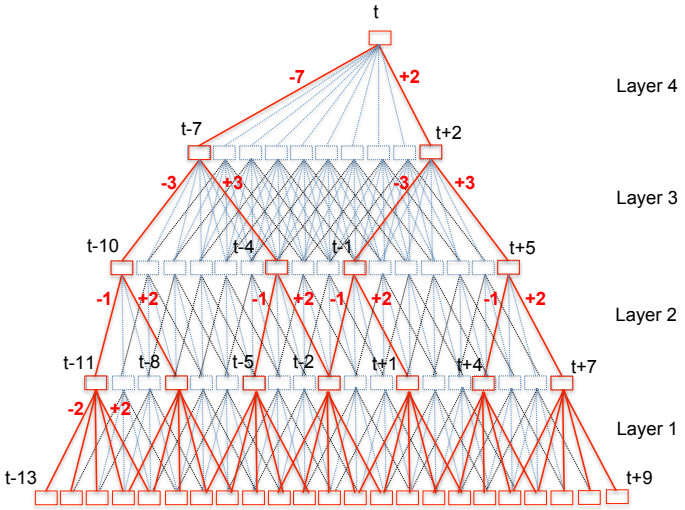
U syntézy řeči pro více mluvčích se do modelu přidává vektor, který specifikuje daného mluvčího. Tyto vektory mohou být předem vygenerovány a uloženy pro každou nahrávku, případně postačí jeden vektor pro každou osobu pokud všechny nahrávky obsahují stejný hlas a prostředí. Poté je vektor přiveden na vstup sítě společně s textem pro syntézu. Tento přístup umožní síti syntetizovat různé hlasy dle vstupního vektoru včetně těch, které model neviděl během trénování.

Tacotron 2 tým k této úloze použil síť, která generovala vektory pro úlohu verifikace mluvčího (viz [24]). K trénování tohoto systému byly použity nahrávky bez transkripce, které mohly obsahovat i ruchy. Tato síť poté umožňovala generovat vektory fixní délky, které již z několika sekundové nahrávky umožní verifikovat mluvčího. Pro generování těchto vektorů byla použita architektura z publikace Generalized end-to-end loss for speaker verification [25].

Pro účely práce byl zvolen stejný přístup, který byl použit Tacotron 2 týmem, ale byl zvolen jiný systém pro generování vektorů mluvčího.

1.3.0.1 Použitá architektura pro získání vektoru mluvčího

Pro získání vektoru mluvčího byl využit systém poskytnutý ústavem ITE (viz [26]). Tento systém je založen na Time delay neural network (TDNN) [27] architektuře, která je zobrazena na obrázku 1.11, a X-VECTORS architektuře [28].



Obrázek 1.11: TDNN architektura z publikace [27].

TDNN architektura pracuje v čase t s kontextem daného rámce. Například jednotlivé neurony první vrstvy pracují s kontextem 5 rámců vstupní sekvence a další vrstvy již s rozdílným kontextem z předchozí vrstvy. Díky tomuto přístupu pracují hlubší vrstvy s větším kontextem a kromě rámce v čase t je zpracován i kontext v intervalu $[t-13, t+9]$. V této architektuře se může využívat podzvorkování (červeně zvýrazněno v obrázku 1.11), které umožňuje až 5× rychlejší trénování.

X-VECTOR architektura převádí vstupní signál do vektorů charakterizujících mluvčího (tzv. X-Vektory). Využívá TDNN sítě pro postupné zpracování celé vstupní sekvence a na výstupy TDNN sítě je použita tzv. stats pooling vrstva, která ze vstupních framů vypočte průměr a směrodatnou odchylku, a tyto hodnoty jsou spojeny do jednoho vektoru. Následně jsou použity dvě tzv. segment vrstvy (dopředná neuronová síť) s ReLU aktivační vstvou a na výstupu je použita softmax vrstva. Tato síť je trénována pro klasifikaci N mluvčích a po natrénování je X-Vektor získán z výstupu první segment vrstvy bez použití aktivační funkce.

Používaná architektura v této práci obsahuje několik změn. Nepoužívá se zde podvzorkování, tudíž TDNN vrstva používá celý kontext té předchozí. Na vstupu

každé TDNN vrstvy jsou rámce v kontextu násobené po prvcích naučenou maticí a poté je proveden pooling přes časovou dimenzi, čímž se limituje počet vah pro trénování. Na výstupu TDNN a dopředných vrstev se používá ELU aktivační funkce pro rychlejší konvergenci a pomocí použité pooling vrstvy se vypočítá pouze rozptyl. Celkem architektura používá 6 TDNN vrstev, poté dvě dopředné neuronové vrstvy, mezi nimiž je pooling vrstva, a na výstupu se nachází softmax. Výstupní X-Vektor se získává z výstupu pooling vrstvy.

2 Fonetická transkripce

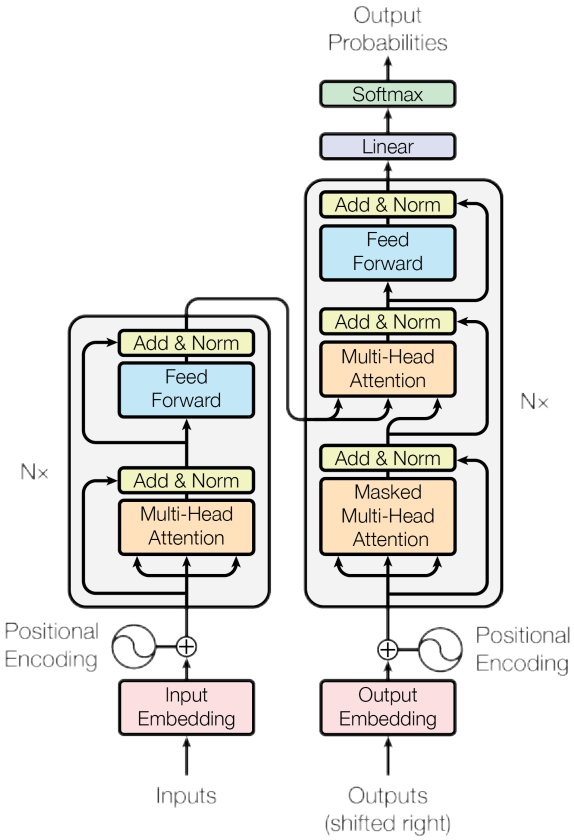
Použití fonetického přepisu (označováno jako Grapheme-to-Phoneme, G2P) pro trénování systémů na syntézu řeči je přínosnější než využití samotného textu. Fonetický přepis vyjadřuje výslovnost textu pomocí posloupnosti fonémů. Jeho využití místo standardního textu pro syntetizování hlasu přináší lepší srozumitelnost, a navíc je možné základní přepis rozšířit o modelování nádechu, pauzy v řeči či další různé zvukové projevy

Pro fonetickou transkripci byla vybrána Transformer architektura [29] využívající attention mechanismu. Cílem této práce nebylo vytvoření systému pro fonetickou transkripci, proto podrobnější popis a provedené práce v této oblasti pomocí Seq2Seq rekurentních neuronových sítí je možné nalézt v bakalářské práci autora [30]. Transformer architektura byla vybrána pro možnost srovnání s dosaženými výsledky z bakalářské práce. Srovnání proběhlo pouze na transkripci samostatných slov, poté je tato architektura použita pro fonetický přepis celých vět v této práci.

2.1 Transformer

Transformer architektura (viz Obrázek 2.1) vznikla v publikaci Attention is all you need [29]. Před touto publikací se pro zpracování sekvenčních dat používaly primárně Seq2Seq rekurentní neuronové sítě, kde kodér zpracoval vstupní sekvenci, z níž vytvořil skrytý stav, a poté dekodér začal postupně generovat výstup. Myšlenkou této publikace bylo, že není potřeba takto zpracovávat vstupní sekvenci, jelikož byla i náročnější zpětná propagace chyby. V Transformer modelu se používá attention mechanismus, jenž určuje, na který vstupní stav se má síť nejvíce sou-

středit. Na vstupu je při každém generovaném kroku vstupní sekvence a zároveň výstupní sekvence predikovaná v předchozích krocích. Je tak možné jednoduše určit chybu při každém predikovaném výstupu, rychleji učit tuto síť a paralelizovat výpočty, protože se nemusí čekat na dokončení předchozího kroku.

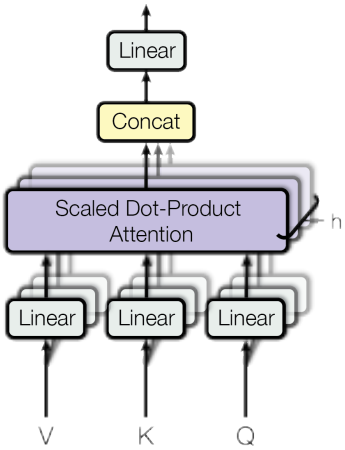


Obrázek 2.1: Transformer architektura z publikace [29].

V levé části obrázku 2.1 je zobrazen kodér, na jehož vstupu je v této práci přivedena sekvence znaků, která je pomocí embeddingu převedena do vektorové podoby. Position Encoding (PE, viz 2.1) se zde stará o zakódování pozice vstupních znaků, jinak by nebyla zachována (u rekurentních neuronových sítí se pozice zachová díky postupnému zpracování vstupní sekvence). V tomto výpočtu značí *pos* pozici, i dimenzi tokenu a d_{model} dimenzi PE vektoru, který je stejně velký jako dimenze embeddingu, aby je bylo možné sečíst.

$$\begin{aligned} PE[pos, 2i] &= sin(pos/10000^{2i/d_{model}}) \\ PE[pos, 2i+1] &= cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{2.1}$$

Dále je použit Multi-Head attention mechanismus (viz Obrázek 2.2) a dopředná neuronová síť. Attention mechanismus zde funguje stejně, jako je popsáno u DCTTS modelu (viz kapitola 1.1.2). Cílem je zde určit, na kterou část se má nyní síť zaměřit. U Multi-Head attention mechanismu je attention vrstev několik (většinou 8) a každá se snaží zaměřovat na jiný semantický význam. Poté je již použita dopředná neuronová síť a na závěr softmax funkce pro určení výstupních pravděpodobností.



Obrázek 2.2: Architektura Multi-Head Attention bloku z publikace [29].

2.2 Provedené experimenty

U fonetické transkripce bylo experimentováno s Transformer modelem, kde byla využita a upravena implementace [38] sloužící k překladu mezi jazyky.

V první části experimentů bylo cílem natrénovat systém pro fonetickou transkripci na slovníku poskytnutém Laboratoří počítačového zpracování řeči pro možnost porovnání s dříve dosaženými výsledky v předchozích pracích [30]. Data použitá pro optimalizaci a trénování jsou samostatná slova nebo slovní spojení s jejich fonetickým přepisem.

Po optimálním natrénování sítě se mohl výsledek porovnat s dříve použitými systémy pro fonetickou transkripci. Výsledky word error rate (WER) jsou uvedeny v tabulce 2.1, kde je porovnán základní Baseline systém založený na statistickém modelování, jenž se běžně používá v rámci laboratoře v různých nadstavbových

systémech, Seq2Seq rekurentní neuronová síť (G2P) [39], Google's Neural Machine Translation (GNMT) [40] síť a Transformer síť využívaná v této práci. Transformer architektura překonala Baseline a G2P přístupy. GNMT architektura dosahuje nižší WER, ale trénuje se zrhuba 2× déle než Transformer architektura a má větší nároky na paměť grafické karty.

Systém	**WER** [%]
Baseline	2,91
G2P	1,95
GNMT	1,72
Transformer	1,84

Tabulka 2.1: Porovnání chybovosti dříve použitých systémů v práci [30] s Transformer architekturou pro fonetickou transkripci českého jazyka.

Po porovnání fonetické transkripce samostatného slova s ostatními architekturami se pokračovalo trénováním Transformeru na transkripci celých vět. Vstupní text se převáděl do stejné fonetické podoby, jaká byla použita u poskytnutých nahrávek v trénovací sadě, např. fráze „Ahoj světe jak se máš“ se přepíše na „Ahoj svjete jakse máš“. Na těchto datech se podařilo dosáhnout 3,26 WER a tento natrénovaný model je dále použit v demonstrační aplikaci pro syntézu řeči.

3 Použité metody a provedené experimenty

Pro experimenty byl v této práci vybrán DeepVoice 3 a Tacotron 2 s WaveGlow modelem. DeepVoice 3 udává stejný MOS jako Tacotron 2 s mírně nižším rozptylem, proto byl také vybrán pro porovnání.

Pro Tacotron 2 a WaveGlow byla použita již implementovaná verze od NVIDIA s využitím PyTorch frameworku. Použit je Tacotron 2 [31] a WaveNet je zde nahrazen WaveGlow [32] modelem, který generuje výstupní signál rychleji a dosahuje lepší kvality než nejlepší veřejně dostupná implementace WaveNet modelu. Pro Deepvoice 3 model byla použita oficiální Deepvoice 3 implementace [33].

Rozšířením je druhá část experimentů s Tacotron 2 modelem pro více mluvčích, kde je na vstup přiveden kromě textu i vektor mluvčího. Vektor mluvčího by měl umožnit natrénování sítě pro více hlasů a po natrénování syntetizovat z tohoto vektoru i hlasy, které síť u trénování neviděla.

3.1 Provedené experimenty a dosažené výsledky

Experimenty byly rozděleny do dvou sekcí. První podkapitola se věnuje experimentům se syntézou řeči pro jednoho mluvčího a v druhé podkapitole jsou popsány pokusy o syntézu řeči s využitím vektorů mluvčího. Při použití vektorů mluvčího bylo cílem rozšířit Tacotron 2 síť o vektor identifikující osobu, díky kterému by bylo možné trénovat síť na více osobách a při syntéze řeči změnit hlas použitím vektoru libovolné osoby.

U experimentů se místo klasického textového vstupu využívala fonetická transkripce, která by již měla obsahovat správnou výslovnost, a tím umožnit kvalitnější natrénování modelů. Zároveň v použitých trénovacích datech obsahovala fonetická

transkripce speciální symboly pro pauzu v mluvě, nádech a další hlasové jevy.

K jednotlivým provedeným experimentům lze v příloze nalézt nahrávku a poslechnout si tak kvalitu trénovaného systému. Tyto nahrávky jsou členěny podobně jako kapitoly experimentů a v textovém souboru je u nich uveden popis.

3.1.1 Syntéza řeči s jedním mluvčím

Cílem bylo natrénovat funkční model syntézy řeči pro muže a ženu na českém jazyce s ohledem na nejvyšší možnou kvalitu syntetizované řeči a rychlé trénování modelů.

3.1.1.1 Použitá data

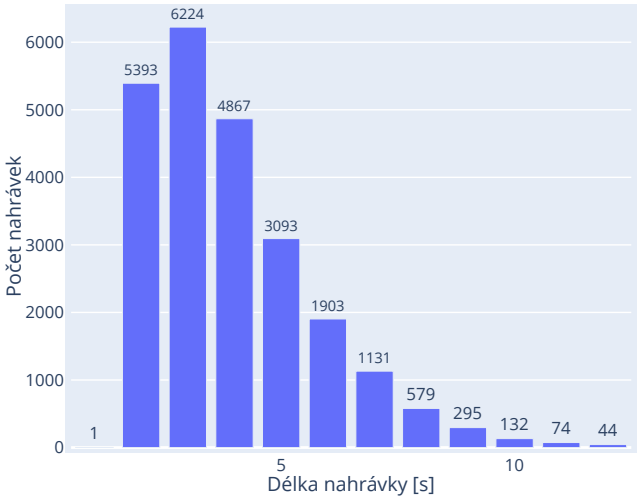
Modely se trénovaly s daty od jedné osoby, která byla rozdělena na trénovací a validační množinu dat. Data byla poskytnuta Laboratoří počítačového zpracování řeči a obsahovala WAV nahrávku, její obsah v textovém souboru a fonetický přepis v .phn souboru. Zvukové soubory jednotlivých osob byly pořízeny z rozhlasu, dále byly ustřiženy v části, kde bylo rozpoznáno ticho (pauza v mluvě osoby). Nacházelo se zde i několik nahrávek, které byly výrazně delší než většina ostatních (outlier), z důvodu optimalizace paměťových nároků na trénování byla tato data ostraněna z datasetu. Takto dlouhé nahrávky se nenacházely ani v trénovacích datech předtrénovaných modelů NVIDIE. WAV nahrávky byly ve formátu Mono 16bit PCM se vzorkovací frekvencí 16 kHz.

Byly použity 4 sady trénovacích dat pro 2 muže a 2 ženy. Celková délka nahrávek pro každou osobu je uvedena v tabulce 3.1.

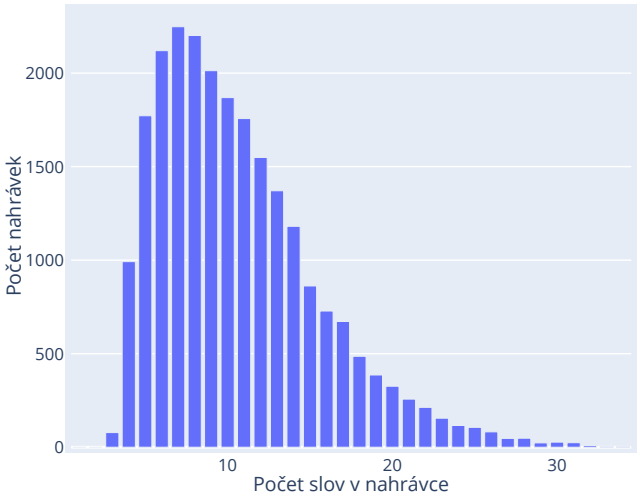
Osoba	**Délka nahrávek**
Muž 1	14 h 59 min 4 s
Muž 2	14 h 0 min 37 s
Žena 1	26 h 10 min 19 s
Žena 2	17 h 19 min 45 s

Tabulka 3.1: Délka nahrávek pro jednotlivé osoby.

Délka jednotlivých nahrávek pro ženu č. 1 je zobrazena v grafu 3.1 a počet nahrávek v grafu 3.2.

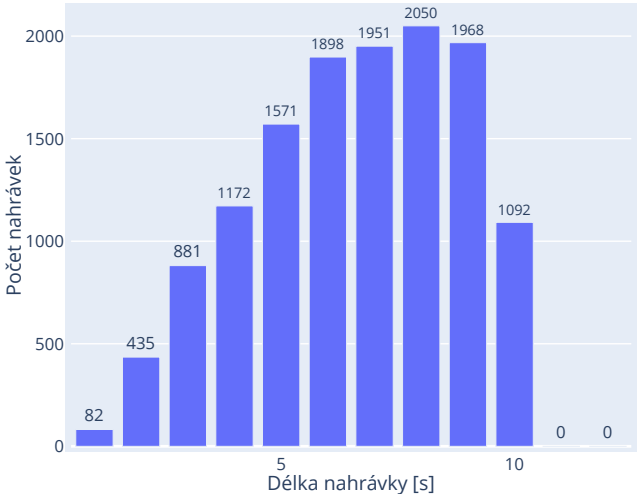


Graf 3.1: Délky nahrávek u osoby s označením žena 1.

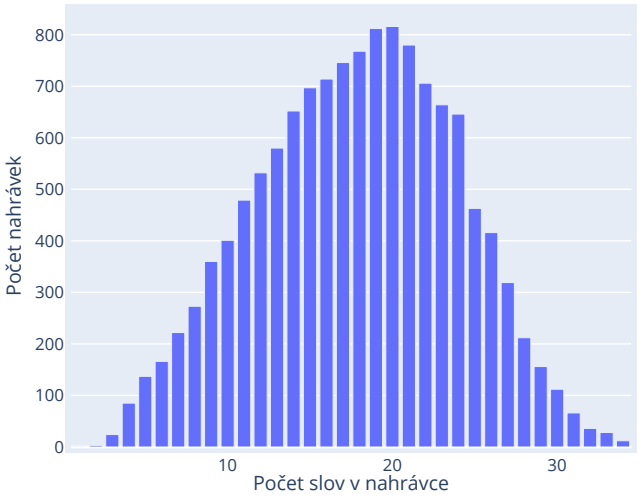


Graf 3.2: Počty slov v nahrávkách u osoby s označením žena 1.

V následující části je srovnání s nahrávkami, jež byly použity k natrénování předtrénovaných modelů. Tyto modely poskytované od společnosti NVIDIA byly trénovány na LJ Speech datasetu [41], který se skládá z 13 100 krátkých nahrávek mluvené řeči ve formátu Mono 16bit PCM se vzorkovací frekvencí 22 050 Hz. Nahrané byly jedním mluvčím, který předčítal 7 knih, a pro každou nahrávku je poskytnuta i transkripce. Nahrávky mají celkem cca 24 hodin a délka každé z nich je od 1 do 10 sekund. V grafu 3.3 je zobrazeno rozložení délek a v grafu 3.4 je znázorněný počet slov v každé nahrávce.



Graf 3.3: Délky nahrávek v LJ Speech datasetu



Graf 3.4: Počty slov v nahrávkách LJ Speech datasetu

Z histogramů je vidět, že se data použitá v této práci skládají z kratších nahrávek, než které byly použity u předtrénovaných modelů. Tím je v modelu způsobena horší kvalita syntetizovaných nahrávek, které obsahují velké množství vstupních slov.

3.1.1.2 Časová náročnost trénování

Před započetím trénování modelů pro syntézu řeči byly provedeny testy, díky kterým bylo možné určit optimální počet gpu pro trénování každého experimentu, aby se maximalizoval přínos využitých zdrojů a mohlo se trénovat paralelně. Uvedené výsledky jsou z clusteru Technické univerzity v Liberci, kde bylo možné využívat

4×Intel Xeon E5-2690, 512 GB RAM a 10×NVIDIA GTX 1080ti.

V tabulce 3.2 je zobrazeno, jak velká část epochy se projde za jednu minutu v závislosti na různém počtu GPU a potřebný čas pro natrénování modelu s 500 epochami.

Počet GPU	**Epochy/min**	**Čas pro natrénování modelu [h]**
1	0,0239	348,68
2	0,0402	207,29
3	0,0520	160,26
4	0,0657	126,84
5	0,0781	106,70

Tabulka 3.2: Výsledky měření časové náročnosti v závislosti na různém počtu GPU pro Tacontron 2 model.

Pro většinu experimentů se používaly dvě grafické karty pro možnost paralelního trénování několika modelů. U experimentů, které bylo potřeba rychleji ověřit, se využíval větší počet GPU.

Stejné měření bylo provedeno i pro WaveGlow model. Záměrem byla možnost paralelního trénování několika Tacotron 2 a WaveGlow modelů zároveň, aby se tyto modely mohly co nejdříve propojit a vyzkoušet jejich společnou funkčnost. Výsledky měření jsou uvedeny v tabulce 3.3, čas pro natrénování je uveden pro 300 epoch.

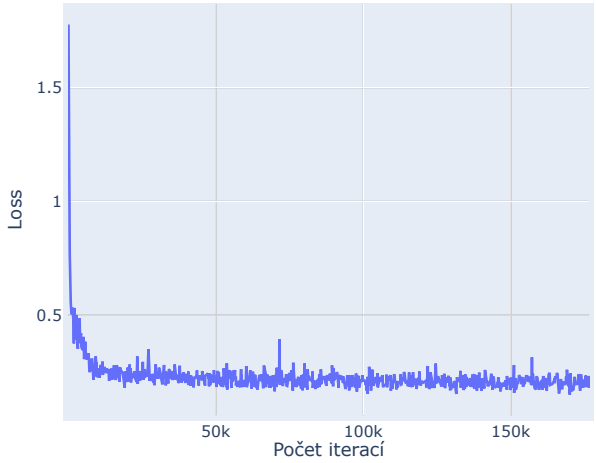
Počet GPU	**Epochy/min**	**Čas pro natrénování modelu [h]**
1	0,0158	316,46
2	0,0234	213,68
3	0,0262	190,84
4	0,0312	160,26
5	0,0357	140,06

Tabulka 3.3: Výsledky měření časové náročnosti v závislosti na různém počtu GPU pro WaveGlow model.

3.1.1.3 Tacotron 2 a WaveGlow

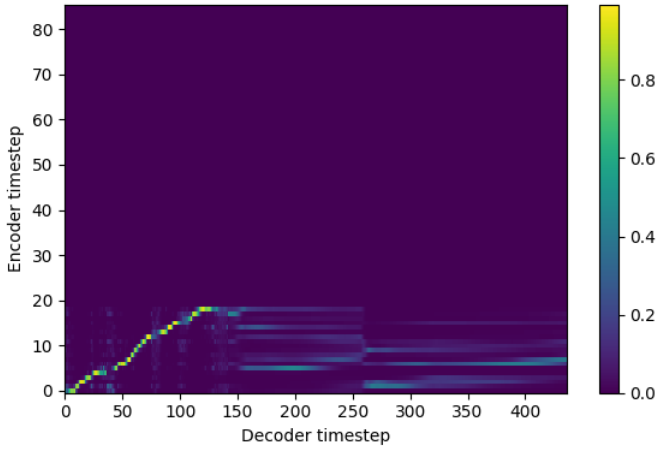
Před započetím experimentů byla přizpůsobena konfigurace obou modelů, kde bylo potřeba změnit parametr hop length na hodnotu 200. To umožnilo zachování posunu rámce o 12,5 ms při vzorkovací frekvenci 16 kHz u trénovacích dat.

Prvním experimentem bylo využití dat osoby žena 1, která měla nejvíce dostupných dat pro trénování Tacotron 2 modelu. Během trénování se snižoval trénovací loss sítě (viz Obrázek 3.5), ale lepším identifikátorem pro kvalitu generovaných nahrávek je tzv. attention zarovnání. To zobrazuje, na které kroky kodéru se soustředí attention mechanismus při predikování výstupního mel spektogramu.



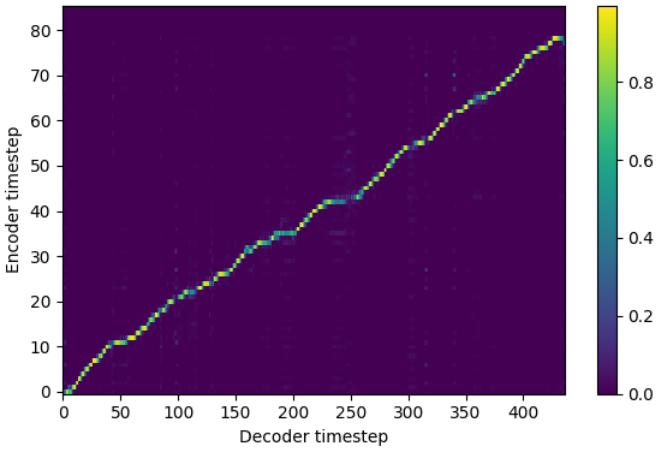
Graf 3.5: Průběh loss funkce v závislosti na počtu iterací.

Graf 3.6 zobrazuje zarovnání attention mechanismu po 20 000 iterací. Na tomto zarovnání je vidět, že se jedná z většiny o šum a attention mechanismus neví, na které části se soustředit, případně kdy ukončit predikování výstupu.



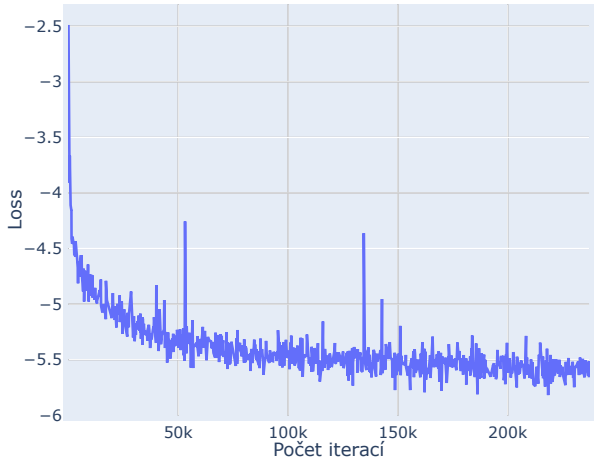
Graf 3.6: Zarovnání attention mechanismu po 20 000 iterací.

Graf 3.7 zobrazuje zarovnání attention mechanismu po delším trénování (80 000 iterací). Zde je vidět, že se attention mechanismus zaměřuje postupně na jednotlivé kroky kodéru. Část, ve které se dočasně zastavil růst tohoto zarovnání, zobrazuje pauzu v řeči, případně nádech mluvčího v syntetizované řeči. Stále se ale nejedná o optimálně natrénovaný attention mechanismus, což je viditelné z rozptylu v některých krocích dekodéru.



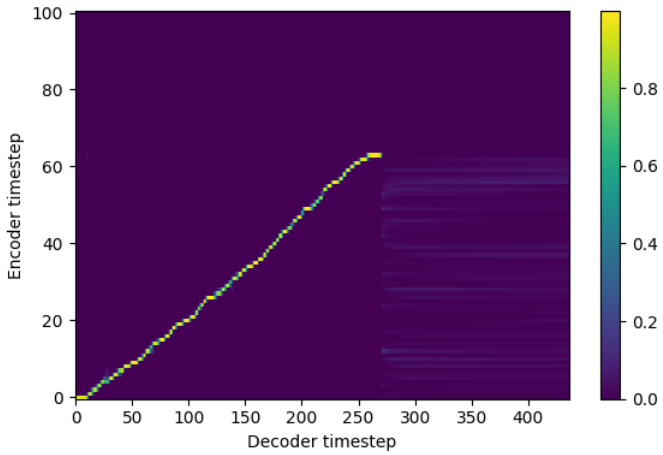
Graf 3.7: Zarovnání attention mechanismu po 80 000 iterací.

Paralelně s Tacotron 2 modelem se na hlase osoby žena 1 trénoval WaveGlow model, u kterého jediným ukazatelem trénování byl trénovací loss (viz Obrázek 3.8). Ten klesal, ale muselo se využívat průběžně ukládaných checkpointů a použít mel spektogramy z nahrávek ve validační sadě pro ohodnocení kvality výstupních nahrávek. Ačkoliv se někdy loss zvýšil nebo se zdálo, že stagnuje, tak se kvalita nahrávky často zlepšila.



Graf 3.8: Průběch loss funkce WaveGlow modelu.

Po zhruba týdnu trénování model umožnňoval generovat nahrávky se srozumitelným ženským hlasem. Vytvořené nahrávky často obsahovaly příliš rušivý šum nebo zněly, jako by někdo mluvil v akusticky špatném prostředí. Často se také po syntetizovaném textu vyskytoval několika sekundový šum, za který mohl Tacotron 2 model, když špatně predikoval konec generování výstupu (viz Graf 3.9).



Graf 3.9: Chybná detekce konce predikování výstupu u Tacotron 2 modelu.

V dalších experimentech byl porovnán model natrénovaný na fonetickém přepisu s modelem natrénovaném na standardním textu. Syntetizované nahrávky z obou systémů byly srozumitelné, ale systém s fonetickým přepisem zněl více přirozeně a u řeči v některých větách používal lepší intonaci hlasu. Natrénované modely obsahovaly stále vyšší šum, proto bylo potřeba přistoupit k ladění některých hyperparametrů modelu.

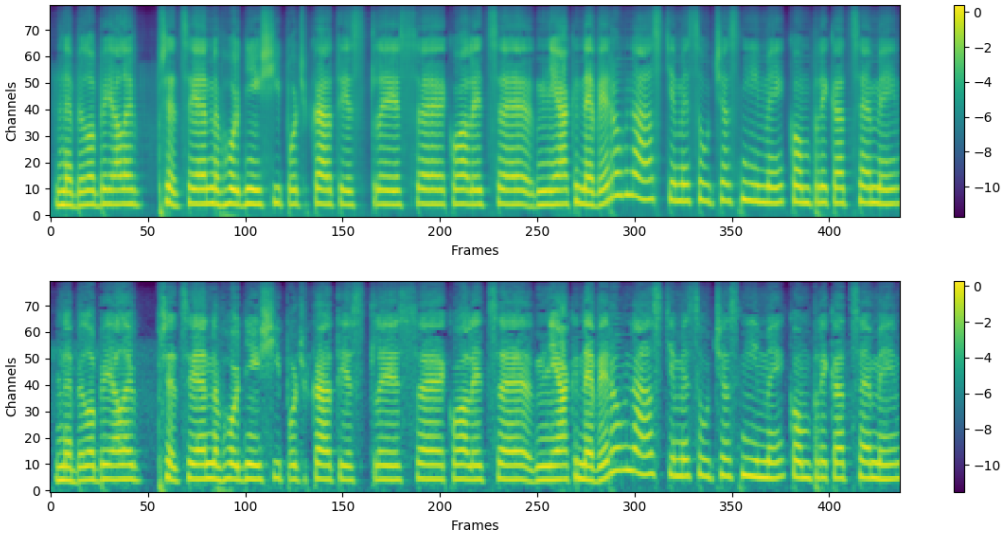
Během trénování WaveGlow modelu je výhodné průběžně sledovat loss funkci. Pokud delší dobu stagnuje, tak je vhodné snížit learning rate, což pomáhá k rychlejší změně v kvalitě nahrávky. U Tacotron 2 i WaveGlow modelu zůstaly dimenze jednotlivých vrstev stejné, zmenšení dimenze mělo často negativní vliv na kvalitu výstupu a naopak zvýšení dimenze pouze zvýšilo nároky na paměť gpu, zpomalilo trénování, ale nebylo viditelné zlepšení kvality výstupu. Batch size parametr byl u Tacotron 2 modelu zvýšen na 32. U WaveGlow modelu zůstal batch size nastaven na 8 a segment length byl nastaven na 6000, jelikož vyšší hodnota občas způsobila chybu u trénování kvůli nedostatku paměti.

Dále bylo potřeba projít trénovací data a odstranit některé nahrávky z důvodu špatného prostředí, ve kterém byly pořízeny. Tyto nahrávky měly špatný vliv na kvalitu trénování, která se po pročištění dat mírně zvýšila.

Při dalším experimentu se trénoval model pro osobu žena 1 celých 500 epoch a po dotrénování byla kvalita syntetizovaných nahrávek výrazně lepší. S použitím Denoiser modulu, který byl součástí WaveGlow implementace, bylo možné i odstranit většinu šumu.

V následujícím trénování byl použit předtrénovaný model od NVIDIA pro Tacotron 2 i WaveGlow, který byl trénován na anglickém jazyce. Během jednoho dne trénování se hlas změnil na požadovanou osobu a bylo možné syntetizovat nahrávky, které byly přirozenější než dříve natrénovaný model. Problémem předtrénovaného modelu byla výslovnost. Například při syntetizování nahrávky s písmenem "r" uprostřed slova model buď nedokázal slovo správně vyslovit, nebo ho při vyslovení vynechal. Při delším trénování se tyto nedostatky zmírnily, ale nezmizely úplně.

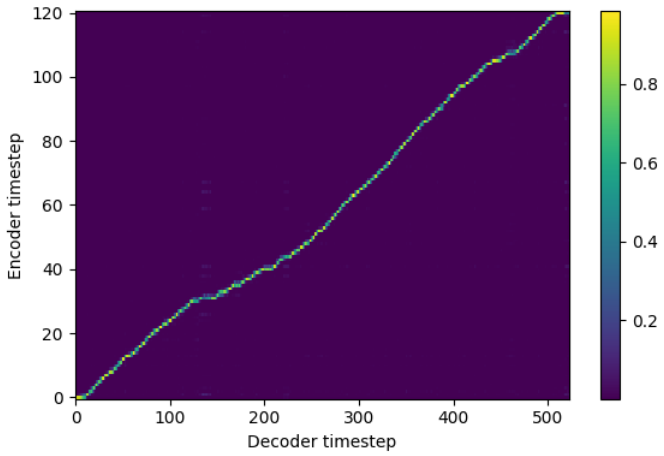
Graf 3.10 zobrazuje predikovaný mel spektogram pro text a mel spektogram získaný z cílové nahrávky ve validační sadě.



Graf 3.10: Predikovaný mel spektogram (nahoře) a cílový mel spektogram získaný z nahrávky (dole) Tacotron 2 modelu pro ženský hlas.

Poté byl trénován hlas pro osobu muž 1. Během předchozích experimentů se našel optimální postup a model byl natrénován od základu i s využitím předtrénovaného

modelu. Předtrénovaný model měl ženský hlas, ale opět se během jednoho dne změnil na mužský a s dalším trénováním měl lepší barvu a přízvuk. Použití předtrénovaného modelu přineslo lepší výsledky než trénování bez něj. Bez něj byl v nahrávkách větší šum a při delším trénování se v syntetizované řeči začala objevovat ozvěna. S využitím předtrénovaného modelu se dosáhlo přirozeného a srozumitelného hlasu, který byl pouze v některých případech rozeznatelný od lidského hlasu. V grafu 3.11 je zobrazeno zarovnání attention mechanismu u syntetizování nahrávky nejlepším systémem.



Graf 3.11: Zarovnání Tacotron 2 attention mechanismu nejlépe natrénovaného modelu s mužským hlasem.

Modely pro další osoby již byly natrénovány podobně jako předchozí dva hlasy. Hlas osoby muž 1 dosahoval nejlepších výsledků.

3.1.1.4 DeepVoice 3

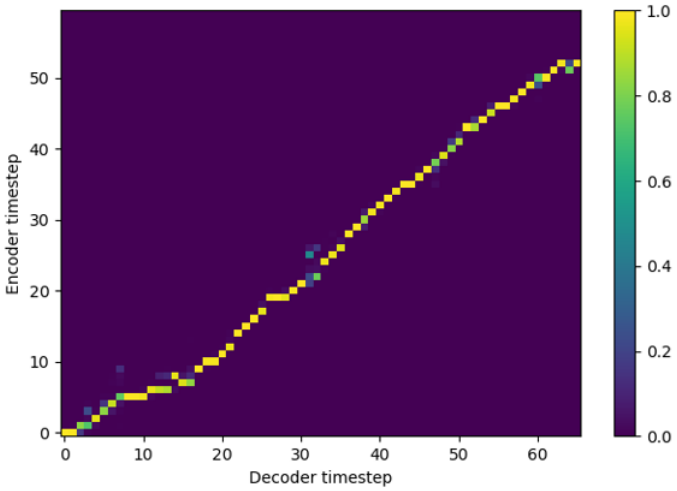
DeepVoice 3 byl trénován pro porovnání s ostatními natrénovanými modely. U této části proběhlo pár základních experimentů, jelikož ani oficiálně publikované nahrávky nepřekonávaly kvalitu Tacotronu 2.

Pro základní experiment byl použit mužský hlas, s nímž se dosahovalo nejlepších výsledků v předchozích experimentech. Během trénování byly průběžně ukládány nahrávky společně se zarovnáním attention alignmentu.

Pro experimenty byl vytvořen vlastní preset (konfigurace), která používala základní parametry. Upraveny byly pouze části, které se vztahovaly k použitým datům

(např. hop size a sample rate). Dále stačilo vytvořit data loader pro používaná data a upravit vstupní abecedu.

Proběhlo několik trénování, ale nepodařilo se z hlasu odstranit rušení, kvůli kterému syntetizovaný hlas nebyl přirozený. Pokud by v syntetizovaném hlasu nebylo toto rušení, tak by na datech použitých pro tuto práci pravděpodobně překonal Tacotron 2. Generované nahrávky jinak měly správnou výslovnost, hlas i správně naučený attention mechanismus (viz Graf 3.12).



Graf 3.12: Zarovnání DeepVoice 3 attention mechanismu u nejlépe natrénovaného systému.

3.1.1.5 Dosažené výsledky

V části pro syntézu hlasu jednoho mluvčího byl porovnán DeepVoice 3 model s Tacotron 2 a WaveGlow modelem. Deepvoice 3 model obsahoval v syntetizovaném hlase příliš velké rušení, proto se více experimentovalo s Tacotron 2 a WaveGlow modelem. S těmito systémy se podařilo natrénovat hlasy pro ženu a muže, které zněly přirozeně a srozumitelně. U ženského hlasu bylo občas více znatelné, že se jedná o hlas syntetizovaný počítačem, proto pro porovnání s komerčními systémy byl vybrán mužský hlas. Syntéza je často nerozeznatelná od lidské řeči, ale občas má problém s vyslovením písmene ”ch” nebo se syntetizováním dlouhého textu.

3.1.2 Syntéza řeči pro více mluvčích

U syntézy pro více mluvčích byl použit Tacotron 2 a WaveGlow model stejně jako u syntézy pro jednoho mluvčího. Tato implementace byla rozšířena o vektor mluvčího, o který byl rozšířen každý krok kodéru Tacotron 2 modelu. Po rozšíření výstupu kodéru bylo potřeba zvýšit vstupní dimenzi u částí využívajících tento vektor (attention mechanismus a některé vrstvy dekodéru). Dále bylo potřeba upravit moduly pro načítání dat, které kromě textu a mel spektogramu musely načítat i vektory mluvčího.

3.1.2.1 Použitá data

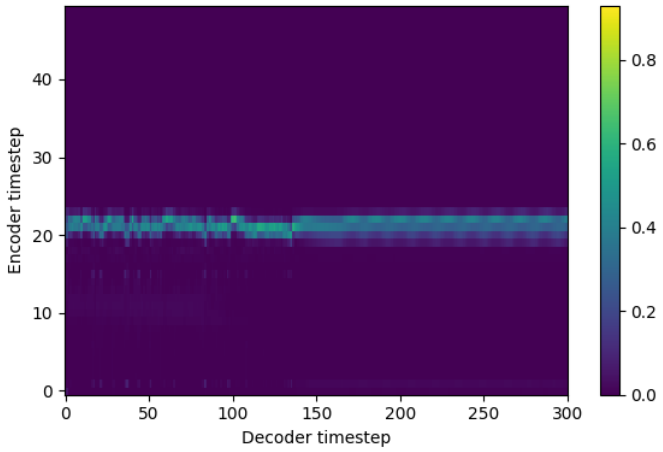
Pro tuto část experimentů byla rozšířena předchozí sada dat o nahrávky z rozhlasu od Laboratoře počítačového zpracování řeči. Bylo poskytnuto 299 821 nahrávek, které trvaly celkem téměř 287 hodin, od 4 290 mluvčí, což je průměrně 4 minuty na osobu a nejdelší data pro osobu trvala okolo 10 minut.

Poskytnutý systém pro xvektory byl napsán v Matlabu, proto v něm byl napsán skript, který pomocí tohoto systému vytvořil pro každou nahrávku vektor mluvčího a uložil ho na disk. V grafu 3.13 jsou zobrazeny vektory mluvčích pro 5 osob a pro 5 nahrávek od každé z nich. Redukce dimenzí pro zobrazení byla realizována pomocí t-SNE [42] metody, která pro možnost zobrazení zredukovala 128-dimenzionální vektor do dvourozměrného.

Graf 3.13: Zobrazení vektorů mluvčí pomocí t-SNE algoritmu.

3.1.2.2 Experimenty

V prvním experimentu byly použity všechny poskytnuté nahrávky a v každé sadě pro trénování bylo 32 nahrávek (batch size). Tento experiment první dva týdny nepřinesl žádné výsledky, až v třetím týdnu se v nahrávkách objevily příznaky hlasů. V grafu 3.14 je zobrazeno zarovnání attention mechanismu během trénování. Ani po delší době trénování se však model nezlepšoval a generoval pouze šum s nesrozumitelnými hlasy v pozadí.



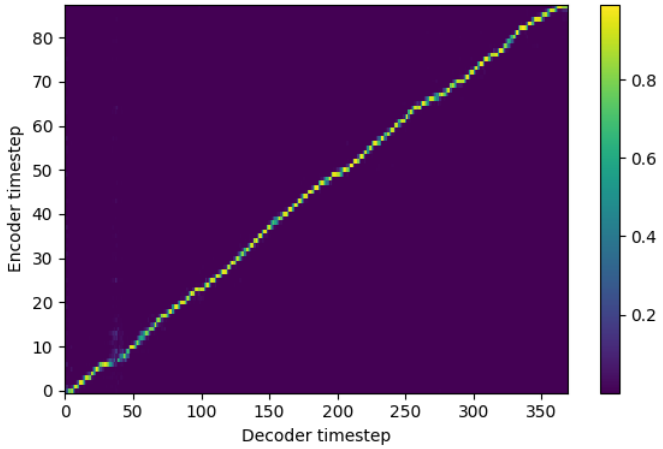
Graf 3.14: Zarovnání rozšířeného Tacotron 2 modelu pro více mluvčích v prvním experimentu.

V druhém experimentu byla ověřena funkčnost implementace pomocí využití dat pro syntézu jedné osoby. Byly tedy vytvořeny vektory mluvčího pro nahrávky jedné osoby a pouze na ní se tento systém trénoval. Po třech dnech trénování proběhla kontrola modelu a ten již vykazoval známky učení. V nahrávkách se dala rozpoznat řeč mužského hlasu, na kterém byl model trénován. Po dalších třech dnech model generoval již srozumitelné nahrávky ve špatné kvalitě, ale i tímto výsledkem bylo ověřeno, že se model rozšířený o vektory mluvčího učí. Bylo také možné, že se systém naučil ignorovat tyto vektory, proto se přistoupilo k dalšímu experimentu, v němž byla přidána osoba opačného pohlaví.

V dalším experimentu byly vytvořeny vektory mluvčího i pro ženský hlas a síť se trénovala na dvou hlasech opačného pohlaví. Po devíti dnech trénování síť dokázala generovat srozumitelný hlas, který stále nebyl kvalitní, ale podle zvoleného vektoru

mluvčího bylo možné syntetizovat text hlasem dané trénované osoby. Při použití vektoru mluvčího, který síť neviděla u trénování, byl změněn výstupní hlas dle pohlaví, ale přepínal se pouze mezi trénovanými hlasy. Síť při trénování se dvěma osobami měla problém se správným detekováním konce syntézy, proto vždy po syntetizované řeči generovala šum a generování zastavila maximální délka výstupní sekvence. Po delším trénování se hlas v nahrávkách mírně zlepšoval, ale šum v pozadí byl příliš rušivý a síť konvergovala pomalu.

Při dalším trénování byly využity všechny čtyři hlasy, které byly použity pro syntézu jednoho mluvčího. V natrénovaném systému opět fungoval výběr hlasu dle vektoru mluvčího na vstupu. Použitím vektoru různých osob bylo většinou vybráno správné pohlaví a výsledný hlas byl podobný těm, které síť používala k trénování, pouze byl mírně upraven např. v barvě hlasu či výslovnosti, ale tyto rozdíly byly minimální. V grafu 3.15 je zobrazeno natrénované zarovnání attention mechanismu u nejlépe natrénovaného modelu.



Graf 3.15: Zarovnání rozšířeného Tacotron 2 modelu pro více mluvčích v nejlépe natrénovaném modelu.

3.1.2.3 Dosažené výsledky

Experimenty syntézy hlasu s využitím vektoru mluvčího prokázaly možnost natrénovat model s daty, která byla využívána pro trénování syntézy pro jednoho mluvčího. Nejlepší model dokázal vybrat hlas správného pohlaví na základě vstupního vektoru mluvčího. Výstupní hlas byl mírně modifikovaný hlas osoby z trénovacích dat.

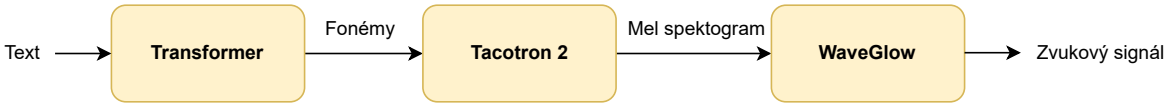
Tato část práce byla vypracována nad rámec zadání a bohužel nebyl dostatek času na provedení dalších experimentů. Jeden experiment se trénoval zhruba jeden až dva týdny, než začal přinášet příznivé výsledky.

Přidání většího množství dat pro více osob by umožnilo tento systém optimálně natrénovat. Ověření funkčnosti tohoto přístupu dává možnost budoucímu vylepšení a rozšíření v dalších pracích.

3.2 Výsledná architektura pro syntézu řeči

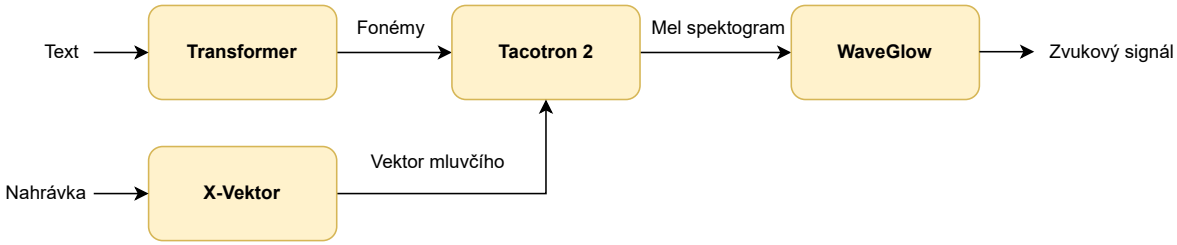
Výsledné řešení je realizováno pomocí několika systémů a dělí se na dvě části. První část využívá data pouze jednoho mluvčího a v druhé části jsou přidány vektory mluvčího pro identifikaci hlasu osoby, které umožní trénovat systém na více osobách.

Na obrázku 3.1 je znázorněn systém pro syntézu řeči jedné osoby (speaker-dependent). Na vstupu se vyskytuje text, který je pomocí Transformer sítě převeden do fonémů ve stejném formátu, v jakém byla trénovací data. Fonetický přepis je vstupem do Tacotron 2 sítě, která z něj predikuje mel spektogram, jenž je vstupem do WaveGlow sítě. WaveGlow následně vytvoří ze vstupního mel spektogramu zvukový signál.



Obrázek 3.1: Blokové schéma systému pro syntézu řeči jednoho mluvčího.

Rozdíl v druhém systému je v použití vektoru mluvčího (viz Obrázek 3.2). Blok X-Vektor vytváří vektor mluvčího z poskytnuté nahrávky na vstupu. Tato architektura umožňuje samostatné trénování každého z těchto modulů, a tak je možné všechny modely trénovat paralelně a nezávisle na sobě.



Obrázek 3.2: Blokové schéma systému pro syntézu řeči pro více mluvčí.

3.3 Další možná rozšíření

Tato kapitola seznamuje s existujícími experimenty v různých publikacích, o které by v budoucnu tato implementace mohla být potanciálně rozšířena.

V publikaci Towards Transfer Learning for End-to-End Speech Synthesis from Deep Pre-Trained Language Models [34] se autoři snažili rozšířit Tacotron 2 architekturu o BERT kodér [35] s attention mechanismem, který by pomohl vytvořit lepší vnitřní reprezentaci vstupní sekvence slov. BERT model získá kvalitní sémantické a syntaktické informace ze vstupního textu, které jsou poté přidány ke vstupu dekodéru Tacotronu. Tento přístup nevedl ke znatelnému zlepšení predikovaných nahrávek, ale pomohl k rychlejší konvergenci modelu a k přesnějšímu ukončení predikování výstupu.

Dalším možným rozšířením této architektury je použití tzv. global style tokenů (GST Tacotron) [36], kde je kodér Tacotron modelu rozšířen o vektor stylu. Použitím reference kodéru [37] je získán vektor prozódie (zvukové vlastnosti) ze vstupní sekvence. Ten je přiveden do attention mechanismu, který se učí měřit podobnost mezi vektorem prozódie a global style tokeny. Global style tokeny jsou na počátku náhodně inicializovány a učí se společně s Tacotron modelem. GST Tacotron v závěru umožňuje během syntézy řeči využít natrénované tokeny pro ovlivnění stylu výstupní řeči, tím je možné například ovládat emoce.

4 Vytvořená aplikace a evaluace systému

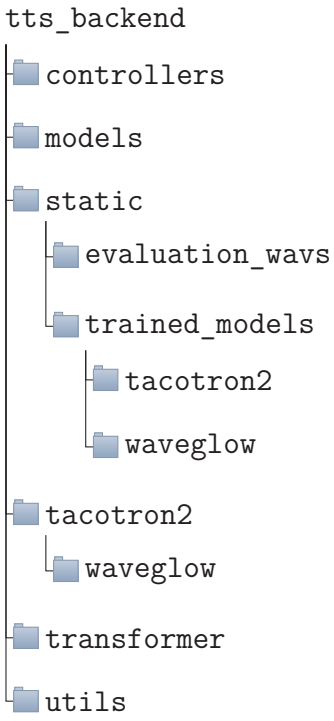
Pro systém byla vytvořena webová aplikace, která umožňuje syntézu řeči mužského nebo ženského hlasu ze zadaného textu. Dále aplikace umožňuje hodnocení natrénovaného systému a několika komerčních systémů pro český jazyk. Z hodnocení je možné určit MOS hodnoty a tím získat srovnání s komerčními systémy. Aplikace pro hodnocení je podobná aplikaci, kterou je možné vytvořit například pomocí služby Amazon Mechanical Turk, kde různí lidé hodnotí kvalitu nahrávek za finanční odměnu (podobnou službou byl určen MOS většiny modelů pro syntézu řeči). V této práci je hodnocení nahrávek dobrovolné.

První podkapitola popisuje implementaci jednotlivých částí vytvořené aplikace, jsou popsány uživatelské obrazovky a endpointy na backendu aplikace. V druhé podkapitole jsou popsány vybrané komerční systémy pro porovnání s nejlépe natrénovaným modelem této práce a na závěr jsou zde shrnuty výsledky hodnocení těcho systémů.

4.1 Implementace aplikace

Aplikace je rozdělena na backend, který poskytuje REST api, a frontend, který využívá uživatel. Pro backend část byl vybrán jazyk Python z důvodu jednoduchého vytvoření REST api s využitím Flask knihovny a snadného použití natrénovaných modelů, jejichž implementace je také v Pythonu. Pro ukládání dat z hodnocení systémů byla využita databáze MongoDB. Frontend byl napsán s využitím React frameworku.

Níže je znázorněna adresářová struktura backend aplikace:



4.1.1 Syntéza nahrávek

Webová aplikace umožňuje syntézu řeči hlasem vybraného mluvčího. Pro syntézu se používá nejlepší natrénovaný Tacotron 2 a WaveGlow model pro jednoho muže a jednu ženu. Využití systému pro syntézu hlasu jednoho mluvčího znamená, že při startu serveru je připraven v paměti GPU jeden Tacotron 2 model pro každý hlas. Dále stačí, aby byl připraven jeden WaveGlow model a stejně tak i systém pro fonetický přepis. Nároky na GPU paměť jsou celkem 1,9 GB.

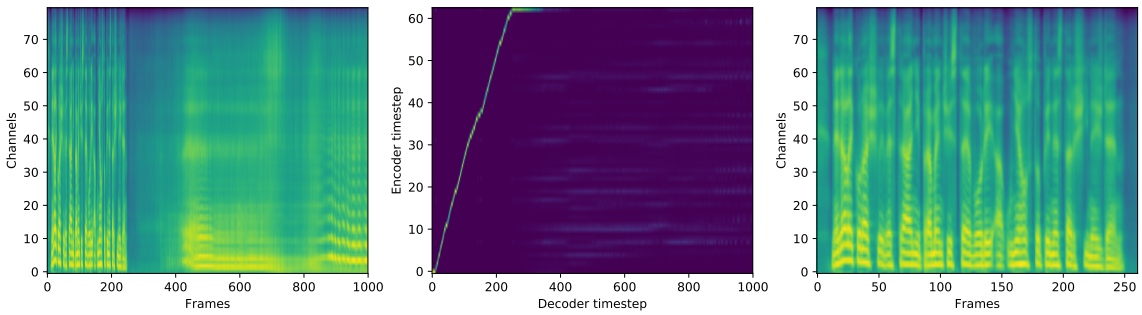
Frontend pro syntézu obsahuje v počátečním stavu jednoduchý formulář se vstupním polem pro zadání textu, který je po odeslání převeden do fonetické podoby a vrácen do druhého vstupního pole. V tomto poli je možné fonetickou transkripci upravit, nebo experimentovat s různými přepisy. Po odeslání textu se požadavek odešle na backend, který vrátí syntetizovanou nahrávku hlasem vybrané osoby ve formátu WAV, která se objeví pod formulářem s možností přehrání v prohlížeči. Výsledná obrazovka je zobrazena na obrázku 4.1.

Obrázek 4.1: Obrazovka pro syntézu řeči.

Backend umožňuje syntetizovat řeč pomocí GET nebo POST požadavku. GET požadavek je potřeba odeslat na `<BACKEND_URL>/tts/<SPEAKER>/<TEXT>`, kde speaker je hodnota 0 pro ženu nebo 1 pro muže, a fonetická transkripce je provedena na backendu. POST požadavek vyžaduje již získaný fonetický přepis, který může být získán z endpointu `<BACKEND_URL>/g2p`. Poté je možné využití endpointu `<BACKEND_URL>/tts`, který v těle musí obsahovat speaker a text (fonetický přepis) hodnoty. Přepis je přiveden na vstup Tacotron 2 modelu vybraného hlasu a výsledný mel spektogram jde na vstup WaveGlow modelu pro převedení do signálu. Na výstupní signál je použit WaveGlow Denoiser modul, který převede signál do frekvenční oblasti, odečte bias modelu vynásobený zadanou intenzitou a poté je převeden zpět do časové oblasti. Intenzita Denoiser modulu byla nastavena na hodnotu 0,05. Při příliš nízké intenzitě se ve výsledné nahrávce vykytoval rušivý šum, a naopak při příliš vysoké hodnotě tento modul sníží srozumitelnost řeči. Pro signál bez šumu byla v posledním kroku vytvořena WAV hlavička a výsledek byl předán zpět na frontend.

Pro syntézu byla mezi Tacotron 2 a WaveGlow přidána pomocná funkce, která detekuje chybné zastavení Tacotronu 2 při predikci mel spektogramu. Tacotron 2 model v některých případech nezastavil a po dokončení syntézy textu generoval chvilku ticha a šum s různými zvuky. Tento stav se dal detekovat z attention zarovnání, kdy se dekodér několik kroků zaměřoval na poslední krok kodéru a poté začal generovat šum. Pokud bylo detekováno takové chování, tak byl chybný konec mel

spektogramu odstraněn a poté předán dál do WaveGlow modelu. Na obrázku 4.2 je znázorněna oprava šumu na konci nahrávky pomocí attention zarovnání.

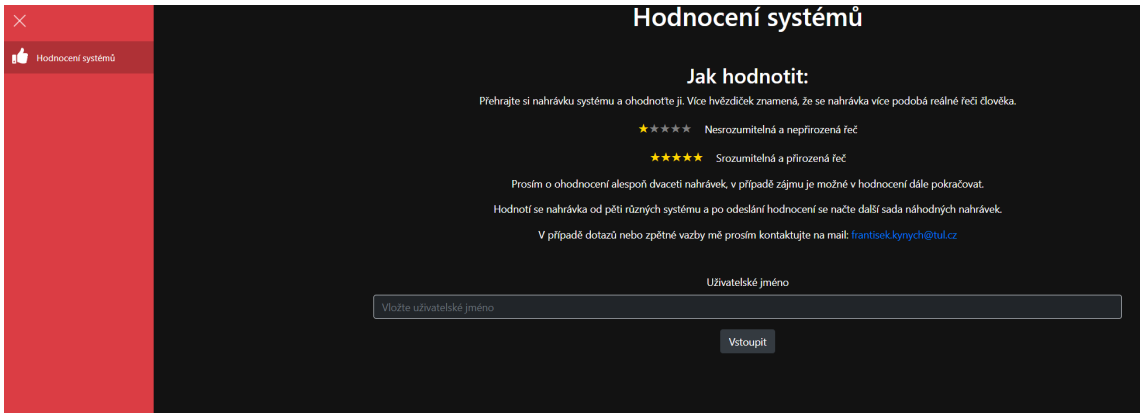


Obrázek 4.2: Výstupní mel spektogram Tacotron 2 modelu, attention zarovnání, s nímž je detekováno chybné zakončení, a opravený mel spektogram.

4.1.2 Hodnocení nahrávek

Tato část aplikace byla inspirována popisem hodnocení Tacotron 2 modelu, kde bylo připraveno 100 nahrávek od natrénovaného systému a tyto nahrávky byly poté hodnoceny lidmi, kteří mohli nahrávku hodnotit dle její srozumitelnosti a přirozenosti. Každá nahrávka byla ohodnocena alespoň 8 lidmi a poté bylo vytvořeno MOS hodnocení s 95% intervalem spolehlivosti.

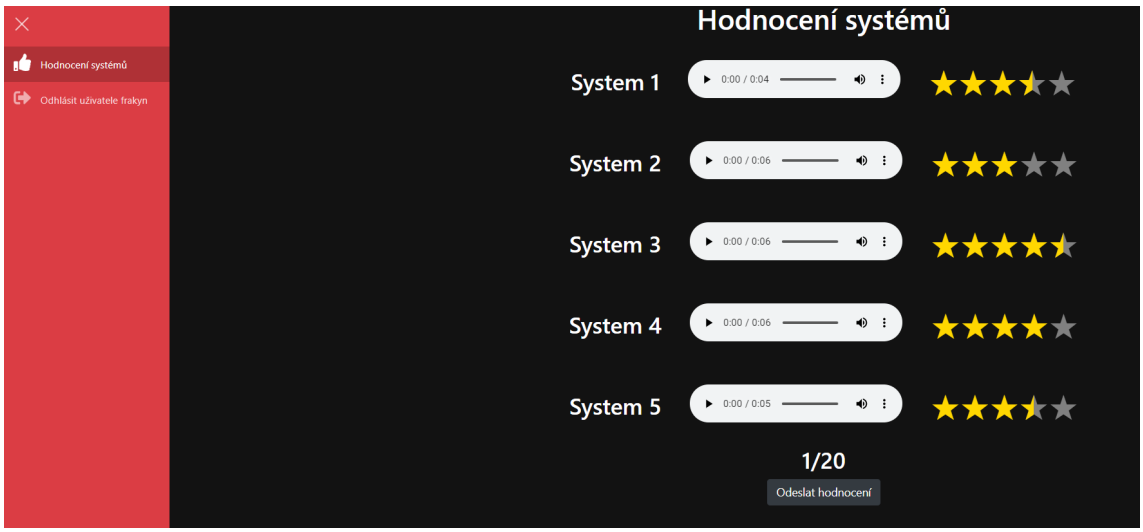
U frontend části byla vytvořena úvodní obrazovka, která uživatele stručně informuje o postupu hodnocení (viz Obrázek 4.3). Po zadání uživatelského jména je možné hodnotit nahrávky.



Obrázek 4.3: Úvodní obrazovka aplikace pro hodnocení systémů.

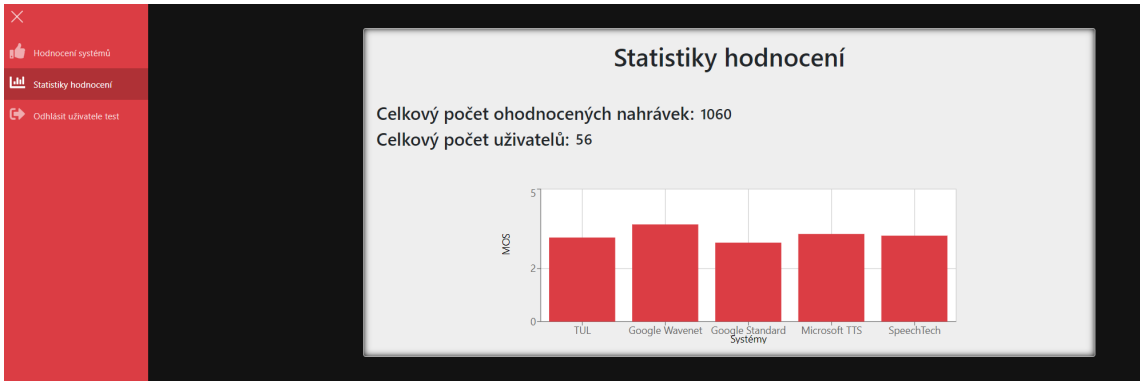
Pro hodnocení se načte jedna nahrávka od každého systému se stejným syntetizovaným textem a po poslechnutí je možné ji hodnotit dle kvality až pěti hvězdičkami

(nejkvalitnější). Krok hodnocení je půl hvězdičky. Po ohodnocení nahrávek je možné hodnocení odeslat a načte se dalších 5. Nahrávky jsou pokaždé náhodně rozmístěny, aby si uživatel musel vždy poslechnout všechny nahrávky a nehodnotil systémy například na základě několika prvních nahrávek. Hodnotící obrazovka je zobrazena na obrázku 4.4. Uživatelské jméno je uloženo v HTTP cookie, a je tak možné se kdykoliv k hodnocení vrátit a pokračovat v něm, případně po zadání stejného jména je uživateli umožněno pokračovat tam, kde přestal.



Obrázek 4.4: Obrazovka aplikace během hodnocení sytémů.

Po ohodnocení 20 nahrávek je uživateli v menu zpřístupněn odkaz na statistiky, kde je zobrazeno kolik je celkem ohodnocených nahrávek, kolik uživatelů celkem hodnotilo a graf se zobrazeným MOS hodnocením jednotlivých systémů (viz Obrázek 4.5).



Obrázek 4.5: Obrazovka aplikace se statistikami hodnocení.

Backend část obsahuje několik endpointů, které jsou využívány při hodnocení nahrávek. Prvním je `<BACKEND_URL>/evalutaion-init/<USERNAME>`, který slouží k získání statusu hodnocení daného uživatele a případně i nahrávek. Povinnou částí odpovědi je status pole, ve kterém je uvedeno, o jakého uživatele se jedná. Pro nového uživatele je vygenerován první set 20 nahrávek, který je uložen v databázi a navrácen na frontend. Pro uživatele, který již hodnotil, ale hodnocení nedokončil je navrácena rozpracovaná sada a pokračuje tam, kde skončil. Uživatel, který již splnil hodnocení základních 20 nahrávek, dostává status click-for-next, při němž se na frontedu zobrazí poděkování a tlačítko s možností hodnocení dalších 10 nahrávek. Posledním stavem je samostatné poděkování pro uživatele, který ohodnotil všech 100 nahrávek.

Získaná data z backendu obsahují pole s čísly nahrávek. Pro hodnocení je potřeba získat nahrávky z endpointu `<BACKEND_URL>/evaluation-wav/<SYSTEM>/<ID>`, který vrací danou nahrávku zadaného systému. Počet systémů je při startu aplikace uložen ve stavu frontend aplikace a je získán z endpointu `<BACKEND_URL>/evalutaion-system-count`. Počet systémů není nikde uložen v databázi a získá se zjištěním počtu složek systémů. Díky tomu je možné jednoduché rozšíření o další systém.

Hodnocení nahrávky od různých systémů je odesíláno POST požadavkem na endpoint `<BACKEND_URL>/rate` a v těle musí obsahovat jméno uživatele, hodnocení a nahrávky. Na backendu je přijaté hodnocení uloženo do databáze a také je aktualizován seznam nahrávek v dávce pro daného uživatele, tím je umožněno pokračování v případě pozdějšího návratu uživatele.

Poslední částí jsou statistiky, které se získávají z endpointu `<BACKEND_URL>/stats`, který navrací výsledek jednoduchých agregačních funkcí provedených nad hodnoceními v databázi.

4.2 Evaluace systému

Pro hodnocení byl vybrán model můžského hlasu (v datasetu označen jako Muž 1), který zněl nejpřirozeněji a nedělal téměř žádné chyby. Pro porovnání bylo synteti-

zováno 100 nahrávek, se kterými se model nesetkal při trénování, ani při validaci. Syntetizovaný text obsahoval běžné věty bez zkratek, jejichž syntetizování je limitováno G2P modelem, který byl připraven pouze pro fonetickou transkripci vět. Dále bylo vybráno několik komerčních systémů, které se v současné době používají pro syntézu českého jazyka, a pomocí nich byl syntetizován stejný text.

Hodnocení bylo postupně rozesíláno různým skupinám lidí na Technické univerzitě v Liberci i mimo ni. Cílem bylo získat celkem alespoň 800 ohodnocených nahrávek pro každý systém, stejně jako tomu bylo u hodnocení Tacotron 2 modelu. Na základě obdržených hodnocení se poté vypočítal MOS s 95% intervalem spolehlivosti, který umožnil porovnání všech modelů.

4.2.1 Vybrané komerční systémy pro porovnání syntézy češtiny

Natrénovaný systém v této práci byl porovnáván se službou Cloud Text-to-speech poskytovanou společností Google. Z této služby byly vybrány dva hlasy, jeden využívající neuronové sítě (Tacotron 2 + Wavenet, označován jako cs-CZ-Wavenet-A) a druhý hlas vznikal konkatenační syntézou (označován jako cs-CZ-Standard-A). Oba hlasy byly ženské a nahrávky byly generovány s 24 kHz vzorkovací frekvencí.

Další hlas byl od služby Microsoft Azure, v níž je poskytována syntéza pomocí neuronových sítí a standardní syntéza (parametrická s konkatenačními prvky). Pro český jazyk lze zatím použít pouze můžský standardní hlas se vzorkovací frekvencí 16 kHz.

Poslední hlas byl od společnosti SpeechTech, u které bylo možné v aplikaci SpeechTech TTS syntetizovat nahrávky mužským hlasem se vozkovací frekvencí 16 kHz (označován jako hlas Jan 2.10).

Všechny systémy syntetizovaly nahrávky pouze z obyčejného textu. Google a Microsoft systémy navíc umožňují použití vstupního formátu ve tvaru Speech Synthesis Markup Language (SSML). SSML umožňuje vkládat tagy pro další modifikaci výstupní řeči, čímž je možné např. změnit hlasitost a rychlost řeči.

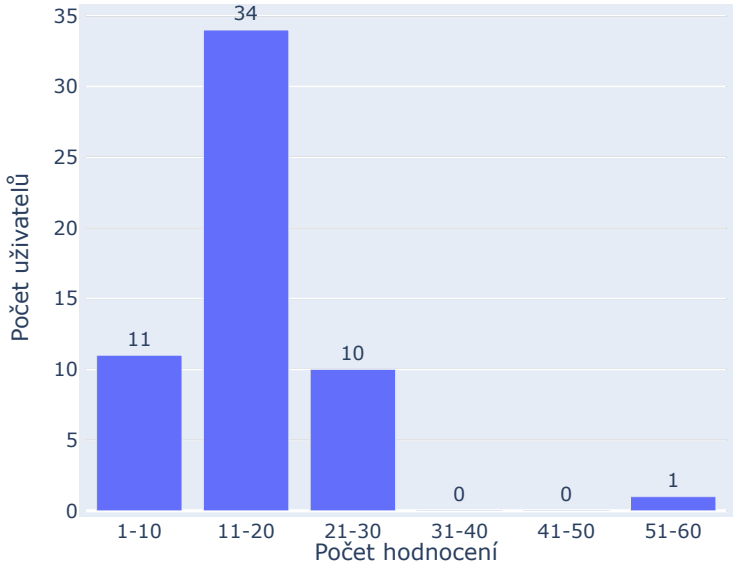
4.3 Dosažené výsledky

Hodnocení systémů se zúčastnilo celkem 56 lidí, kteří dohromady ohodnotili 1060 nahrávek od každého systému. Výsledné hodnocení je zobrazeno v tabulce 4.1.

Systém	MOS
Google (Tacotron 2 + WaveNet)	3.666 ± 0.074
Microsoft	3.306 ± 0.076
SpeechTech	3.242 ± 0.075
Diplomová práce	3.171 ± 0.075
Google (Standard)	2.978 ± 0.080

Tabulka 4.1: MOS jednotlivých systémů s 95% intervalem spolehlivosti.

V grafu 4.1 je zobrazen počet hodnocení různých uživatelů, ze kterého je vidět, že někteří honotící mohli mít vyšší váhu než ostatní. V tabulce 4.2 jsou výsledky nezatížené různým počtem hodnocení od každého uživatele. Z hodnocení každého uživatele byl vypočten průměr, čímž získal každý stejnou váhu.



Graf 4.1: Histogram ohodnocených nahrávek.

Systém	MOS
Google (Tacotron 2 + WaveNet)	3.475 ± 0.312
Diplomová práce	3.171 ± 0.294
Microsoft	3.123 ± 0.290
SpeechTech	3.102 ± 0.275
Google (Standard)	2.693 ± 0.294

Tabulka 4.2: MOS jednotlivých systémů s 95% intervalem spolehlivosti se stejnou váhou hodnotícího.

V případě, kdy mohl mít hodnotící vyšší váhu, kvůli většímu počtu ohodnocených nahrávek, dosáhla diplomová práce čtvrtého místa a překonala standardní systém Googlu. S MOS hodnocením, u kterého měl každý hodnotící stejnou váhu, dosáhla práce druhého místa.

Natrénovaný systém z této práce je srovnatelný s komerčně používanými systémy pro syntézu textu v českém jazyce a překonal standardní Google systém. Vzhledem k horší kvalitě použitých dat pro trénování v této diplomové práci je systém překvapivě lepší než výsledek, který byl s danými daty očekáván. Použitím profesionálně zaznamenaných nahrávek s vyšší vzorkovací frekvencí by bylo možné přiblížení v kvalitě ke Google systému využívajícímu Tacotron 2 a WaveNet model.

Systémy od firem Google, Microsoft a Amazon dosahují na anglickém jazyce stále lepší kvality než u českého jazyka a s použitím SSML formátu je možné v syntetizované řeči ovládat i další prvky jako např. emoce. Výhoda ostatních systémů je v možnosti syntetizování rozsáhlých textů a zkratek, ale tato funkce by se dala vyřešit rozšířením trénovacích dat pro použité modely.

5 Závěr

Cílem práce bylo prozkoumání a ověření současných přístupů k umělé syntéze řeči s využitím neuronových sítí a dále vybrání nejlepší architektury a její použití pro syntézu českého jazyka mužským a ženským hlasem. Po natrénování vybraných modelů se využily poslechové testy pro porovnání s komerčními systémy a byla vytvořena jednoduchá webová aplikace pro možnost demonstrace výsledků.

První kapitola se věnuje popisu současných řešení syntézy řeči s využitím neuronových sítí. Pro další experimenty byly vybrány DeepVoice 3 a Tacotron 2 architektury, které ze vstupního textu predikovaly mel spektogram. K převedení mel spektogramu do zvukového signálu se využíval model WaveGlow. Porovnání s komerčními systémy bylo založeno na mužském hlasu syntetizovaném Tacotron 2 modelem, který zněl přirozeně a srozumitelně.

Dále byla vytvořena webová aplikace umožňující syntézu řeči ze zadaného textu mužským nebo ženským hlasem. V této aplikaci bylo zároveň vytvořeno prostředí pro realizaci poslechových testů, kde uživatelé mohli porovnávat a hodnotit syntetizovanou řeč se stejným obsahem od různých systémů (diplomová práce, Google, Microsoft, SpeechTech). Hodnocení se účastnilo 56 lidí a celkem bylo ohodnoceno 1060 nahrávek od každého systému. Z těchto hodnocení byly vypočítány MOS hodnoty s 95% intervalem spolehlivosti. V první části mohl mít uživatel větší vliv na výsledek, díky většímu počtu ohodnocenách nahrávek. Zde byl nejlépe hodnocen Google systém využívající Tacotron 2 a WaveNet model, poté Microsoft a SpeechTech systémy. Následoval systém z této diplomové práce, který dokázal překonat standardní hlas Googlu nevyužívající neuronových sítí. V druhé části byl z uživatelských hodnocení vytvořen průměr a až poté se počítal MOS pro jednotlivé systémy,

tím měl každý hodnotící stejný vliv na výsledek. Po provedení těchto kroků zůstal neuronový hlas od Googlu na prvním místě, poté následoval výsledek diplomové práce, Microsoft, SpeechTech a nakonec standardní hlas Googlu. Vzhledem k charakteristice použitých dat pro trénování systému v této práci, kde se často měnilo prostředí, případně se vyskytoval i šum a ruch v pozadí, dosáhl natrénovaný model překvapivého výsledku. Výstup práce ukázal, že je kvalitou hlasu a srozumitelností srovnatelný se současně používanými komerčními systémy.

Zadání práce bylo tedy splněno a nad rámec tohoto zadání byla realizována fonetická transkripce pomocí Transformer architektury, která pomohla k lepšímu natrénování sítě a také se v ní modelovaly například pauzy v řeči nebo nádechy. Dále byl nad rámec zadání rozšířen Tacotron 2 model o vektory mluvčího. Tyto vektory bylo možné získat z nahrávek využitím modelu popsaného v kapitole 1.3, který byl poskytnut Laboratoří počítačového zpracování řeči. Záměrem byla možnost změny hlasu syntetizované řeči s přivedením vektoru cílové osoby. Tento systém se podařil natrénovat tak, že umožnil změnu hlasu dle pohlaví osoby přivedeného vektoru a také mírně modifikoval naučený hlas. Pro další experimenty nad rámec zadání již nezbyl čas.

Další pokračování v této práci by bylo možné například aplikováním rozšíření z kapitoly 3.3, čímž by bylo možné v syntetizované řeči ovládat i emoce. Výrazného zlepšení kvality syntetizované řeči by bylo možné dosáhnout profesionálním zaznamenáním mluvené řeči, tím by se výsledný MOS více přiblížil nebo i rovnal současně používaným komerčním systémům s neuronovými sítěmi. Dalším pokračováním by mohlo být rozšíření experimentů pro syntézu řeči s vektory mluvčího, k tomu je ale také potřeba více kvalitních dat od velkého množství osob.

Literatura

[1] NOUZA, Jan, ed., KOLDOVSKÝ, Zbyněk, ed. a VÍCH, Robert, ed. Řeč a počítač: principy hlasové komunikace, úlohy, metody a aplikace: sborník článků. Vyd. 1. Liberec: Technická univerzita v Liberci, 2009. 235 s. ISBN 978-80-7372-548-8.

[2] SHEN, Jonathan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018. p. 4779-4783.

[3] Cloud Text-to-Speech [online]. Google, 2020 [cit. 2020-05-18]. Dostupné z: https://cloud.google.com/text-to-speech

[4] Převod textu na řeč [online]. Microsoft, 2020 [cit. 2020-05-18]. Dostupné z: https://azure.microsoft.com/cs-cz/services/cognitive-services/text-to-speech/

[5] SpeechTech Text-to-speech [online]. SpeechTech, 2020 [cit. 2020-05-18]. Dostupné z: https://www.speechtech.cz/speechtech-text-to-speech/

[6] ŠILHÁN, Stanislav. Parametrická syntéza české řeči: Parametric synthesis of Czech speech. Liberec: Technická univerzita v Liberci, 2004. Diplomové práce.

[7] ŠKODA, Jan. Zřetězená syntéza řeči pracující s rozsáhlou databází promluv: Concatenation speech synthesis working with large speech databases. Liberec: Technická univerzita v Liberci, 2005. Diplomové práce.

[8] SCHMIDT, Erik M.; WEST, Kris; KIM, Youngmoo E. Efficient Acoustic Feature Extraction for Music Information Retrieval Using Programmable Gate Arrays. In: ISMIR. 2009. p. 273-278.

[9] TACHIBANA, Hideyuki; UENOYAMA, Katsuya; AIHARA, Shunsuke. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018. p. 4784-4788.

[10] SRIVASTAVA, Rupesh K.; GREFF, Klaus; SCHMIDHUBER, Jürgen. Training very deep networks. In: Advances in neural information processing systems. 2015. p. 2377-2385.

[11] RIBEIRO, Flávio, et al. Crowdmos: An approach for crowdsourcing mean opinion score studies. In: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2011. p. 2416-2419.

[12] PING, Wei, et al. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. arXiv preprint arXiv:1710.07654, 2017.

[13] GRIFFIN, Daniel; LIM, Jae. Signal estimation from modified short-time Fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1984, 32.2: 236-243.

[14] MORISE, Masanori; YOKOMORI, Fumiya; OZAWA, Kenji. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE TRANSACTIONS on Information and Systems, 2016, 99.7: 1877-1884.

[15] OORD, Aaron van den, et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[16] CHOROWSKI, Jan K., et al. Attention-based models for speech recognition. In: Advances in neural information processing systems. 2015. p. 577-585.

[17] PRENGER, Ryan; VALLE, Rafael; CATANZARO, Bryan. Waveglow: A flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019. p. 3617-3621.

[18] KINGMA, Durk P.; DHARIWAL, Prafulla. Glow: Generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems. 2018. p. 10215-10224.

[19] DINH, Laurent; SOHL-DICKSTEIN, Jascha; BENGIO, Samy. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

[20] OORD, Aaron van den, et al. Parallel wavenet: Fast high-fidelity speech synthesis. arXiv preprint arXiv:1711.10433, 2017.

[21] SOTELO, Jose, et al. Char2wav: End-to-end speech synthesis. 2017.

[22] MEHRI, Soroush, et al. SampleRNN: An unconditional end-to-end neural audio generation model. arXiv preprint arXiv:1612.07837, 2016.

[23] CHOROWSKI, Jan K., et al. Attention-based models for speech recognition. In: Advances in neural information processing systems. 2015. p. 577-585.

[24] JIA, Ye, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In: Advances in neural information processing systems. 2018. p. 4480-4490.

[25] WAN, Li, et al. Generalized end-to-end loss for speaker verification. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018. p. 4879-4883.

[26] JANSKY, Jakub, Jiri MALEK, Jaroslav CMEJLA, Tomas KOUNOVSKY, Zbynek KOLDOVSKY a Jindrich ZD'ANSKY. Adaptive Blind Audio Source Extraction Supervised By Dominant Speaker Identification Using X-Vectors. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, 2020, , 676-680. DOI: 10.1109/ICASSP40776.2020.9054693. ISBN 978-1-5090-6631-5. Dostupné také z: https://ieeexplore.ieee.org/document/9054693/

[27] PEDDINTI, Vijayaditya; POVEY, Daniel; KHUDANPUR, Sanjeev. A time delay neural network architecture for efficient modeling of long temporal contexts. In: Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[28] SNYDER, David, et al. X-vectors: Robust dnn embeddings for speaker recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018. p. 5329-5333.

[29] VASWANI, Ashish, et al. Attention is all you need. In: Advances in neural information processing systems. 2017. p. 5998-6008.

[30] KYNYCH, František : M15000036. Využití neuronových sítí pro automatickou fonetickou transkripci. Liberec: Technická univerzita v Liberci, 2018. Bakalářské práce. Technická univerzita v Liberci.

[31] Tacotron 2 (without wavenet). GitHub [online]. NVIDIA Corporation, 2018 [cit. 2020-03-30]. Dostupné z: https://github.com/NVIDIA/tacotron2

[32] WaveGlow: a Flow-based Generative Network for Speech Synthesis. GitHub [online]. NVIDIA Corporation, 2018 [cit. 2020-03-30]. Dostupné z: https://github.com/NVIDIA/waveglow

[33] YAMAMOTO, Ryuichi. Deepvoice3_pytorch. GitHub [online]. 2018 [cit. 2020--04-14]. Dostupné z: https://github.com/r9y9/deepvoice3_pytorch

[34] FANG, Wei; CHUNG, Yu-An; GLASS, James. Towards Transfer Learning for End-to-End Speech Synthesis from Deep Pre-Trained Language Models. arXiv preprint arXiv:1906.07307, 2019.

[35] DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[36] WANG, Yuxuan, et al. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. arXiv preprint arXiv:1803.09017, 2018.

[37] SKERRY-RYAN, R. J., et al. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. arXiv preprint arXiv:1803.09047, 2018.

[38] LYNN-EVANS, Sam. Transformer. GitHub [online]. 2018 [cit. 2020-04-15]. Dostupné z: https://github.com/SamLynnEvans/Transformer

[39] GitHub - cmusphinx/g2p-seq2seq: G2P with Tensorflow. GitHub [online]. Copyright © 2018 [cit. 14.04.2018]. Dostupné z: https://github.com/cmusphinx/g2p--seq2seq

[40] Luong, Minh-Thang, Eugene BREVDO a Rui ZHAO. Neural Machine Translation (seq2seq) Tutorial [online]. 2017 [cit. 2018-04-14]. Dostupné z: https://github.com/tensorflow/nmt

[41] ITO, Keith. The LJ Speech Dataset [online]. 2017 [cit. 2020-03-26]. Dostupné z: https://keithito.com/LJ-Speech-Dataset/

[42] MAATEN, Laurens van der; HINTON, Geoffrey. Visualizing data using t-SNE. Journal of machine learning research, 2008, 9.Nov: 2579-2605.

A Obsah přiloženého CD

- text diplomové práce
  - diplomova_prace_2020_Frantisek_Kynych.pdf
  - zadani_diplomova_prace_2020_Frantisek_Kynych.pdf
- experimenty
  - adresář obsahuje nahrávky jednotlivých systémů během prováděných experimentů a část nahrávek použitých pro poslechové testy
- zdrojové kódy
  - backend a fronted vytvořené aplikace
  - upravený Tacotron 2 pro více mluvčích