FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Generování české řeči pomocí neuronových sítí

Ing. Jakub Vít

DISERTAČNÍ PRÁCE

k získání akademického titulu doktor v oboru

Kybernetika

Školitel: Doc. Ing. Jindřich Matoušek, Ph.D.

Plzeň, 2023

FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Czech Speech Generation Using Neural Networks

Ing. Jakub Vít

DISSERTATION THESIS

submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the field of

Cybernetics

Supervisor: Doc. Ing. Jindřich Matoušek, Ph.D.

Plzeň, 2023

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této práce.

. . . . . .

Ing. Jakub Vít

Abstrakt

Disertační práce se zaměřuje na nové architektury pro počítačové generování řeči pomocí neuronových sítí. S jejich příchodem došlo k velmi bouřlivému rozvoji nových metod, které umožnily generovat řeč s vyšší kvalitou a přirozeností, než umožňovaly tradiční metody. V teoretické části se uvádí souhrn běžných postupů a důležitých pojmů týkajících se syntézy řeči, jako je například zpracování textu, fonetická abeceda, poslechové testy, anotace a melovský spektrogram. Představeny jsou zde tradiční metody syntézy řeči: konkatenační metoda a statistická parametrická metoda. Teoretická část zároveň popisuje nové architektury neuronových sítí pro syntézu řeči vysoké kvality, a to převážně architektury *WaveNet* a *WaveRNN*. Dále je zde představen podpůrný webový nástroj pro vývoj a výzkum syntézy řeči. Experimentální část práce popisuje výstupy, kterých bylo dosaženo vlastní implementací těchto metod na syntézu českého jazyka, a také experimenty, jejichž cílem bylo navrhnout a vyvinout nový systém TTS pro syntézu řeči s vyšší kvalitou než v té době stávající systém, který byl založen na konkatenační metodě. Poslechový test ukázal, že nový systém dosáhl na českém jazyce lepších výsledků. Práce obsahuje i pokusy s trénováním jedné sítě pro více řečníků a také s vícejazyčnou syntézou. Experimenty dále obsahují analýzu trénovacích dat pro nové modely ve srovnání s tradičními metodami. V posledních letech se objevilo značné množství nových architektur, poslední část proto obsahuje jejich ucelený přehled a popisuje podrobněji několik z nich. Jsou zde představeny architektury *LPCNet*, *MelNet*, *Tacotron*, *MelGAN*, *VITS* a další. Je zde i diskuse o stávajícím trendu v podobě *end-to-end* architektur.

Klíčová slova: syntéza řeči, TTS, neuronové sítě, WaveNet, WaveRNN

Abstract

This dissertation focuses on new architectures for computational speech generation using neural networks. With their advent, there has been a very vigorous development of new methods that have enabled the generation of speech with higher quality and naturalness than traditional methods have allowed. In the theoretical part, a summary of common procedures and important concepts related to speech synthesis, such as text processing, phonetic alphabet, listening tests, annotation and mel spectrogram, is presented. The traditional methods of speech synthesis are introduced: the concatenation method and the statistical parametric method. The theoretical part also describes new neural network architectures for high quality speech synthesis, mainly the WaveNet and WaveRNN architectures. Furthermore, a web-based support tool for speech synthesis development and research is presented. The experimental part of the thesis describes the outputs achieved by the actual implementation of these methods on Czech language synthesis, as well as the experiments aimed at designing and developing a new TTS system for speech synthesis with higher quality than the then existing system, which was based on the concatenation method. The listening test showed that the new system achieved better results on the Czech language. The paper also includes experiments on training a single network for multiple speakers as well as multilingual synthesis. The experiments also include an analysis of the training data for the new models compared to traditional methods. A significant number of new architectures have emerged in recent years, so the last section provides a comprehensive overview and describes several of them in more detail. LPCNet, MelNet, Tacotron, MelGAN, VITS and other architectures are introduced. There is also a discussion of the current trend towards end-to-end architectures.

Keywords: speech synthesis, TTS, neural networks, WaveNet, WaveRNN

Poděkování

Tímto bych rád poděkoval Doc. Ing. Jindřichu Matouškovi, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

Obsah

Seznam zkratek

ASR	Automatic speech recognition
CCR	Comparison category rating
DTW	Dynamic time warping
F0	Fundamental frequency
FFT	Fast Fourier transform
FST	Finite-state transducers
G2P	Grapheme to phoneme
GAN	Generative adversarial networks
GRU	Gated recurrent unit
HMM	Hidden Markov model
IPA	International phonetic alphabet
LSTM	Long short-term memory
MCD	Mel cepstral distortion
MFCC	Mel-frequency cepstral coefficients
MOS	Mean opinion score
MRT	Modified rhyme test
MSE	Mean squared error
MUSHRA	Multiple stimuli with hidden reference and anchor
PCM	Pulse code modulation
PESQ	Perceptual evaluation of speech quality
POS	Part of speech
RELU	Rectified linear unit
RNN	Recurrent neural network
SUS	Semantically unpredictable sentences
TTS	Text-to-speech
VAD	Voice activity detection
VAE	Variational autoencoder
WER	Word error rate

Kapitola 1

Úvod

Úloha syntézy řeči se v posledních letech těší velkému zájmu. Důvodem je příchod hlasových asistentů a mobilních zařízeních, které využívají komunikaci pomocí hlasu. Zároveň v dnešní době zažívají rozkvět algoritmy strojového učení, zvláště pak neuronové sítě, které dokážou těžit z obrovského výkonnostního potenciálu, který nabízejí dnešní moderní procesory a grafické karty.

V poslední době se objevily nové architektury neuronových sítí, které jsou schopné generovat umělou řeč ve vysoké kvalitě. Vedle tradičních metod syntézy řeči tak dnes existují například architektury WaveNet, WaveRNN a Tacotron. Jejich potenciál je značný. Přinášejí mnoho výhod, které odsouvají tradiční metody na druhou kolej.

1.1 Cíle disertační práce

Cílem mého úsilí bylo navrhnout a vyvinout metodu syntézy řeči, která by dosáhla lepších výsledků než stávající desítky let vyvíjený systém, který je postavený na konkatenační metodě. V praktické části jsem navrhl, naprogramoval a provedl experimenty s novou metodou syntézy řeči postavenou na *WaveNET*u a *LSTM parametrické syntéze s neurálním vokodérem* založeném na *WaveRNN*. Srovnání metod na základě poslechových testů je součástí práce.

Práce rovněž popisuje experimenty, které jsem v průběhu tvorby disertační práce provedl a které pomáhaly udávat směr výzkumu, který byl zaměřen na syntézu řeči pomocí neuronových sítí co nejvyšší kvality.

Vzhledem k tomu, že neuronové sítě typu WaveNet a WaveRNN jsou velmi mladé, byly tyto experimenty nezbytné, neboť dosud nebyly provedeny buď vůbec, anebo jen na anglickém jazyce v zahraniční literatuře.

Jedním z výstupů práce je nový TTS framework postavený na neuronových sítích, který je možné použít i v praktickém nasazení. Dalším výstupem je nástroj SpeechLab, který usnadňuje práce související s výzkumem a nasazením systému syntézy řeči, jako je například ruční segmentace, anotace, nahrávání a sestavování trénovacích dat.

1.2 Struktura práce

Práce je členěna následovně. Po úvodu následuje kapitola popisující obecný problém převodu textu na řeč. Jsou zde popsány základní pojmy z úlohy syntézy řeči jako je zpracování textu, fonetická transkripce, poslechové testy a reprezentace řeči pomocí melovských spektrogramů. V další kapitole jsou popsány tradiční metody syntézy řeči, tj. konkatenační a statistická parametrická syntéza.

Další část se zabývá neuronovou sítí WaveNet. Tato architektura byla první, která dokázala generovat řeč vysoké kvality a odstartovala revoluci v úloze syntézy řeči. Je zde představena její architektura a principy, díky kterým je schopná generovat kvalitní řeč. Následující část popisuje další velmi úspěšnou neuronovou architekturu WaveRNN a problematiku neurálních vokodérů.

V praktické části je jako první představen podpůrný nástroj SpeechLab, který byl vyvinut v rámci práce. Dále je zde popsána realizace nového systému TTS s použitím architektury WaveNet a jeho výsledky pro český jazyk. Následuje experiment týkající se analýzy trénovacích dat. Další kapitola představuje systém TTS založený na neurálním vokodéru WaveRNN. Ta obsahuje i poslechový test porovnávající nový systém s tradičními metodami. Dále práce pokračuje experimenty týkajícími se vícejazyčného TTS a multi-speaker trénování.

Poslední kapitola představuje přehled nových architektur, které se v posledních letech objevily. Práce je zakončena závěrem a seznamem literatury.

1.3 Časový kontext práce

Ve světě syntézy řeči a umělé inteligence se vývoj posouvá tak rychle, že to, co se na začátku doktorského studia jevilo jako nejnovější trend, se po ukončení studia a odevzdání práce může jevit jako téměř samozřejmé a málo podnětné.

Dnes je snadné natrénovat modely architektur, které jsou použity v této práci, pomocí volně dostupných nástrojů a skriptů na internetu a výkonných počítačů v cloudu. Existují také předpřipravené modely pro angličtinu i pro češtinu, které mohou používat i laici bez hlubších kybernetických znalostí.

Tato disertační práce byla vypracována v období mezi roky 2015 a 2023. Experimenty a publikované články jsou v souladu s dobou, ve které vznikly. Některé části této práce, které v čase napsání byly velmi aktuální a zkoumaly nejnovější trendy, mohly kvůli tomu ztratit na unikátnosti.

Například experimenty s neuronovou sítí WaveNet včetně publikovaných článků pocházejí z roku 2018, což je jen dva roky po tom, co byla architektura představena. Publikované články tak byly velmi aktuální a jedny z prvních, které se na konferencích objevily. Experiment na českém jazyku včetně vlastní implementace sítě WaveNet popsaný v této práci byl vůbec první publikace, co se týče českého jazyka.

Článek představující WaveNet byl zveřejněn na konci roku 2016 a započal novou éru syntézy řeči. Do té doby bylo nemyslitelné, aby neuronová síť byla schopná generovat tak složitý signál jako je lidská řeč. Toto období se vyznačuje boomem nových architektur neuronových sítí a překotným vývojem v oboru. Situace se stala hektickou, jelikož došlo ke změně paradigmatu. Bylo možné se vrátit ke generativní syntéze, která generuje řeč z modelů, což bylo dříve limitováno neexistencí tak mocných modelů jako jsou právě neuronové sítě.

Chvíli trvalo, než se akademický svět přizpůsobil nové situaci. Dramatické zvýšení kvality, které nové neuronové architektury přinesly, znamenal postupné přeorientování pozornosti vědecké komunity a opuštění tradičních metod. Znamenalo to ústup expertních znalostí a nástup nových architektur učených výhradně z trénovacích dat (i typu end-to-end, viz kapitola 12).

Kapitola 2

Úvod do syntézy řeči

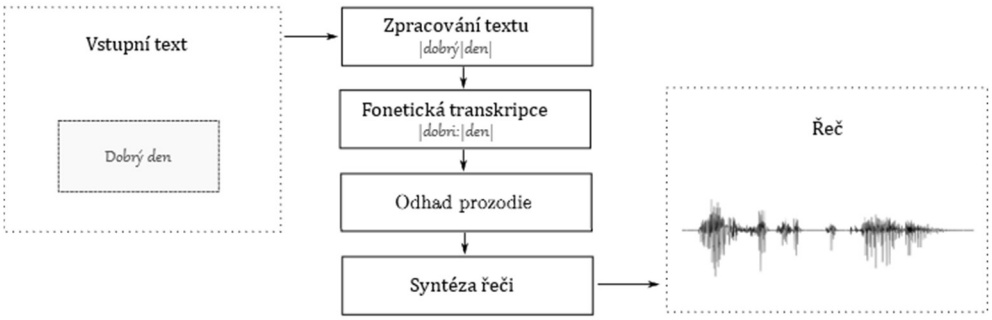
Syntéza řeči je proces, při němž se pomocí počítačových algoritmů uměle vytváří lidská řeč. Cílem je vytvořit řeč co nejpřirozenější a nejsrozumitelnější. Dnes je již problém srozumitelnosti považován za vyřešený, a tak se hlavní směr výzkumu ubírá k přirozenosti.

Počátky syntézy řeči sahají už do 30. let 20. století [1]. Před příchodem neuronových sítí existovaly dvě ustálené metody syntézy řeči: *konkatenační syntéza* s výběrem jednotek a *statistická parametrická syntéza*. Obě měly své klady a zápory, což umožňovalo jejich koexistenci. Zároveň se používala i hybridní metoda, která kombinovala obě metody dohromady. V průběhu času docházelo v obou metodách k inkrementálním zlepšením, které většinou zvyšovali kvalitu pouze nepatrně.

Do těchto klidných vod nyní přichází čerstvý vítr. V posledních letech se totiž situace začíná dramaticky měnit a objevují se nové přístupy a metody syntézy řeči založené na neuronových sítích a strojovém učení. Ty mají potenciál nahradit stávající tradiční po mnoho let vyvíjené metody. Použití metod strojového učení má mnoho výhod. Pro vytvoření modelu nejsou třeba podrobné expertní znalosti, lze využít potenciál velkých dat a výsledný model může dosahovat lepších výsledků.

Syntéza řeči je součástí úlohy zvané *text-to-speech* (TTS). Jejím úkolem je vygenerovat zvukový signál obsahující řeč odpovídající zadanému vstupnímu textu. Tento proces je plně automatický. Tradičně se dělí na tři úlohy, jak naznačuje obrázek 1.

Nutno podotknout, že výše zmíněný popis platí pro tradiční metody syntézy řeči. Některé nové metody využívají tzv. *end-to-end* přístup, ve kterém takový popis nemusí platit. U něj je snaha vše zahrnout do jednoho modelu a na celý problém nahlížet jako na jedinou úlohu. To je velká výhoda při tvorbě hlasu pro nové jazyky, neboť například odpadá problém s fonetickou transkripcí neznámého jazyka.



Obrázek 1: Schéma převodu textu na řeč.

2.1 Zpracování textu

Zpracování textu, někdy také označováno jako *normalizace*, je proces, který se zaměřuje na standardizaci textu, aby byl jednoznačně zpracovatelný syntetizérem. Nahrazuje ve vstupním textu elementy, které mají jinou formu zápisu psaní a výslovnosti. Jedná se například o číslovky, zkratky, data, časy, matematické symboly či jinak uživatelem definované zápisy. Několik příkladů uvádí tabulka 1 a obrázek 2.

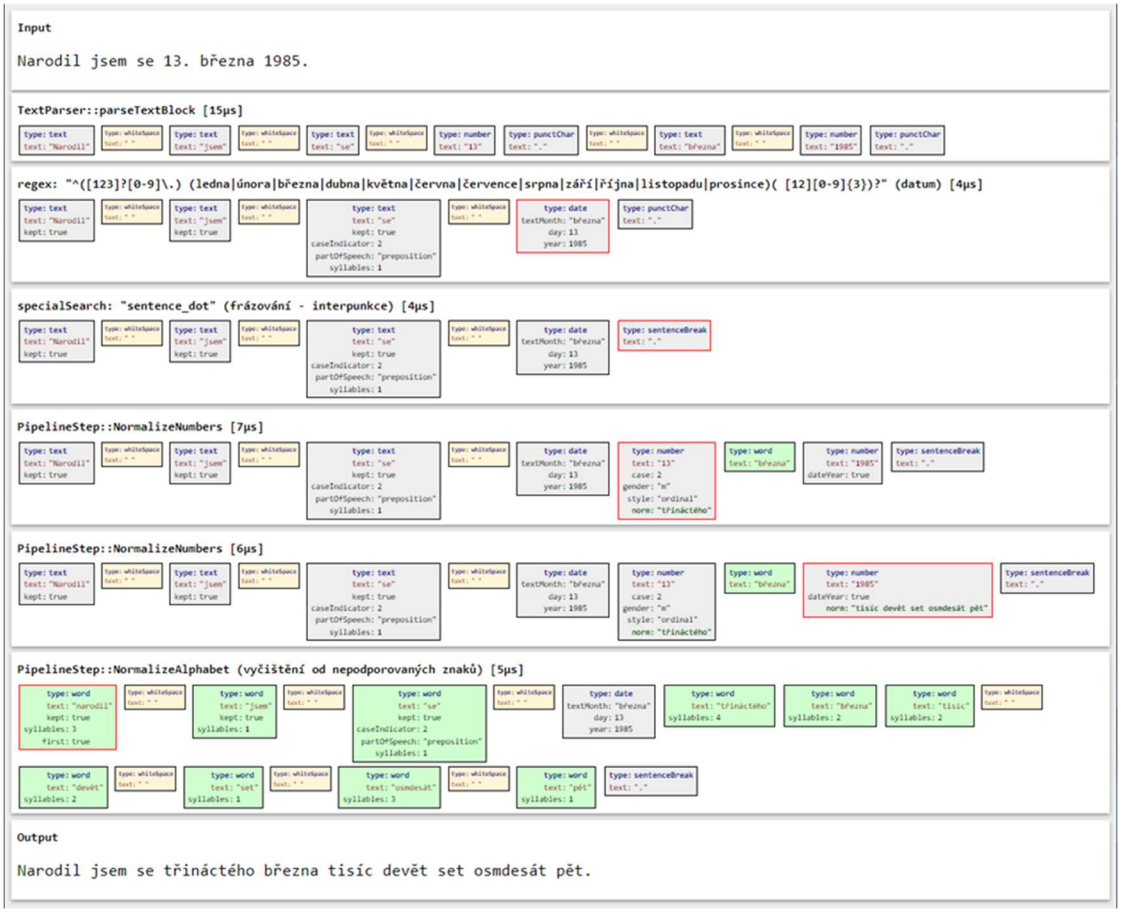
21	→	dvacet jedna
12:00	→	dvanáct nula nula
HIV	→	há í vé
prof. Novák	→	profesor novák

Tabulka 1: Příklady normalizace textu.

Chyby v normalizaci textu jsou snadno postřehnutelné a velmi rušivé. Dokáže je odhalit každý posluchač. Zároveň je velmi obtížné vyřešit všechny případy a výjimky, které mohou nastat. Normalizace textu je většinou zkoumána samostatně. Vědecké práce zabývající se syntézou řeči tak většinou pracují již s předzpracovaným textem.

Dříve bylo použití normalizace textu jako samostatného modulu běžnou praktikou. To znamená, že se text nejdříve normalizoval, a pak se s ním dále pracovalo. S nástupem neuronových sítí a jejich rychlým rozvojem se však objevily tzv. end-to-end systémy, které přistupují k syntéze řeči jako k jednomu celku. Tyto systémy nevyužívají normalizaci textu jako samostatný modul, ale dokážou se naučit normalizaci implicitně na základě trénovacích dat. Tyto systémy se tak mohou vypořádat s různými formami textu, aniž by bylo nutné text předem normalizovat.

Výstupem normalizace textu je čistý text obsahující pouze slova. Rovněž se řeší detekce konců vět a hranic frází. Správné frázování je důležité pro *prozodii*, tj. intonaci a rychlost čtení jednotlivých písmen. Zároveň normalizace doplňuje text o další informace, například o slovní druhy (POS tag), které mohou být později využity jako lingvistický příznak pro neuronovou síť.



Obrázek 2: Ukázka aplikace pravidel pro normalizaci textu. Pro přehlednost je zobrazeno jen několik pravidel. Celkový proces je mnohem komplexnější. Obrázek pochází z programu *SpeechLab* (viz kapitola 6).

2.2 Fonetická transkripce

Fonetická transkripce je proces, který převádí text ze psané formy (písmen) na formu fonetickou (hlásky). V angličtině se tento proces označuje jako G2P *(grapheme-to-phoneme)*. V češtině se většinou text čte stejně, jak se píše. Výjimkou je například jev *spodoba znělosti*. Při něm dochází k ovlivnění výslovnosti slova v závislosti na jeho okolí, například znělost může regresivně způsobit, že předchozí neznělá hláska se vyslovuje zněle apod. Při zpracování textu je tedy nutné zohlednit okolí hlásek.

Příklad fonetické transkripce:

```
Dnes bude zataženo, v některých oblastech přeháňky,
Dnez bude zataženo, vněkterých  oblastech přeháňky,
… po šesté hodině očekáváme sněžení.
… pošesté  hoďyňe očekáváme sňeženýː.
```

V předchozím příkladu bylo zvýrazněno několik fonetických jevů, které se vyskytují při výslovnosti češtiny. Například ve slově „Dnes“ dochází k nahrazení fonému /s/ fonémem /z/ z důvodu spodoby znělosti. Dalším příkladem jsou jednoslabičné předložky, které se vyslovují společně s následujícím slovem v jednom prozodickém taktu. Proto se například předložky „v“ a „po“ připojují k následujícímu slovu.

Přestože podle pravidel českého pravopisu se píše „ně“, změkčení při mluvení se provádí na souhlásce, tedy /ňe/. V mluvené češtině existuje pouze jedna výslovnost „i/y“, není tedy nutné rozlišovat mezi měkkým a tvrdým „i“. V této práci se v zápisech používá varianta /y/. Psané měkké „i“ totiž změkčuje některé předchozí souhlásky, například „di“ se ve výslovnosti změní na /ďy/. Tyto jevy je důležité zohlednit při zpracování textu, aby byl text co nejpřesněji převáděn do fonetické podoby.

2.2.1 Fonetická abeceda

Pro zápis fonetické výslovnosti se používá *fonetická abeceda*, což je znakový systém navržený k fonetickému zápisu a popisu řeči. Cílem fonetické abecedy je co nejpřesněji zaznamenat výslovnost jazyka, aby bylo možné jazyk převést do fonetické podoby. Fonetická abeceda se skládá ze symbolů, které představují jednotlivé fonetické jednotky. Těmi mohou být hlásky (fóny, z anglického phones) nebo fonémy [1]. Tyto symboly se používají pro zápis jednotlivých slov.

Fonetická abeceda se liší v závislosti na jazyce, protože každý jazyk má jinou výslovnost. Existuje několik různých fonetických abeced. Nejpoužívanější je IPA (*International Phonetic Alphabet*), která obsahuje znaky většiny jazyků a je mezinárodně uznávaná.

Obrázek 3 obsahuje podmnožinu fonetické abecedy IPA, která byla použita v experimentech s vícejazyčným systémem syntézy řeči (kapitola 10). V nich bylo velmi důležité zachovat jednotné značení napříč jazyky a zároveň správně popsat fonémy, které mají stejnou výslovnost ve více jazycích.

Abeceda IPA měla dříve problémy s počítačovým kódováním. Dříve se totiž v různých zemích používalo různé kódování, což způsobovalo, že se stejné kódy zobrazovaly různě v různých jazykových nastaveních. S příchodem kódování *Unicode* se tento problém vyřešil.

Pro syntézu řeči se také používá fonetická abeceda SAMPA (*Speech Assessment Methods Phonetic Alphabet*), která byla navržena pro počítačové systémy. Výhodou je, že používá pouze znaky ASCII (viz tabulka 2), což usnadňuje práci a vývoj systémů, které s touto abecedou pracují. Fonetický přepis lze vytisknout pomocí standardních písmen anglické abecedy. Abeceda je lépe čitelná a lze bez problému zobrazit v počítačových terminálech. Některé fonémy se zapisují pomocí více znaků.

V příkladu na začátku kapitoly 2.2 byla použita fonetická abeceda založená na českých grafémech. V ní jsou fonémům přirazeny grafémy tak, aby co nejvíce odpovídaly psané variantě (např. grafém „í“ -> foném /í/). Pouze tam, kde to není jednoznačné, jsou voleny zástupné znaky. Tato abeceda je jednodušší na zápis a čtení, není ale přenositelná na jiné jazyky.

Přehled fonetických abeced je podrobněji popsán v [1] na straně 42.

Vokály	`i, e, a, o, u, i:, e:, a:, o: u:`
Diftongy	`o_u, a_u, e_u`
Frikativy	`f, v, s, z, S, Z, x, h\, l, r, P\, j`
Plozivy	`P, b, t, d, c, J\, k, g`
Nazály	`m, n, J`
Afrikáty	`t_s, t_S, d_z, d_Z`
Významné alofony	`N, F, G, Q\, r=, l=, m=, @, ?`

Tabulka 2: Fonetická abeceda SAMPA pro češtinu.

IPA	Jazyky	IPA	Jazyky	IPA	Jazyky
p (101)	cz, de, gb, ru, sk, us	pf	de	p^j	ru
b (102)	cz, de, gb, ru, sk, us	b^j	ru	t (103)	cz, de, gb, ru, sk, u
ts	cz, de, ru, sk	tʃ	cz, de, gb, sk, us	t^j	ru
d (104)	cz, de, gb, ru, sk, us	dz	cz, sk	dʒ	cz, de, gb, sk, us
d^j	ru	c (107)	cz, sk	ɟ (108)	cz
ɟ	sk	k (109)	cz, de, gb, ru, sk, us	k^j	ru
g (110)	cz, de, gb, ru, sk, us	g^j	ru	ʔ (113)	cz, de, sk, us
m (114)	cz, de, gb, ru, sk, us	m^j	ru	m̩	cz, us
ŋ (115)	cz, de, gb, sk	n (116)	cz, de, gb, ru, sk, us	n̩	sk
n^j	ru	ɳ	us	ɲ (118)	cz, sk
ɲ	sk	ŋ (119)	cz, de, gb, sk, us	ŋː	de
r (122)	cz, de, ru, sk	r̥	cz	r^j	ru
r̝	cz	r̩	cz, sk	rː	sk
f (128)	cz, de, gb, ru, sk, us	f^j	ru	v (129)	cz, de, gb, ru, sk, u
v^j	ru	θ (130)	de*, gb, us	ð (131)	de*, gb, us
s (132)	cz, de, gb, ru, sk, us	s^j	ru	z (133)	cz, de, gb, ru, sk, u
z^j	ru	ʃ (134)	cz, de, gb, sk, us	ʒ (135)	cz, de, gb, sk, us
ʂ (136)	ru	ʐ (137)	ru	ç (138)	de
x (140)	cz, de, ru, sk	x^j	ru	ɣ (141)	cz, sk
ʁ (143)	de	h (146)	de, us	ɦ (147)	cz, gb, sk
ʊ (150)	sk	ɹ (151)	gb, us	j (153)	cz, de, gb, ru, sk, u
ɪ (155)	cz, de, gb, sk, us	l^j	ru	l̩	cz, sk, us
l̩ː	sk	ʎ (157)	de*, sk	w (170)	de*, gb, us
ɕ (182)	de*	ɕː	ru	ʑː	ru
ɬ (209)	ru	tɕ (215)	ru	i (301)	de, ru
iː	cz, de, gb, sk, us	ĭ	ru	e (302)	de, ru
eɪ	gb, us	eː	de	ĕ	ru
ɛ (303)	cz, de, gb, ru, sk, us	ɛɪ	de*	ɛʊ	cz
ɛə	gb	ɛ̃	de	ɛ̃ː	de
ɛː	cz, de, sk	a (304)	cz, de, ru, sk	aɪ	de, gb, us
aʊ	cz, de, gb, us	ã	de	ãː	de
aː	cz, de, sk	ɑ (305)	de*, us	ɑː	de*, gb
ɔ (306)	de, sk	ɔɪ	gb, us	ɔʏ	de
ɔ̃	de*	ɔ̃ː	de*	ɔː	de*, gb, us
o (307)	cz, de, ru	oʊ	cz, us	õː	de
oː	cz, de, sk	u (308)	de, ru, sk	uː	cz, de, gb, sk, us
y (309)	de	yː	de	ø (310)	de
øː	de	œ (311)	de	œ̃	de*
œː	de	œ̃ː	de*	ɒ (313)	de*, gb
ʌ (314)	de*, gb, us	ɨ (317)	de*, ru, us	ʉ (318)	ru, us
ɪ (319)	cz, de, gb, ru, sk, us	ɪɛ	sk	ɪa	sk
ɪu	sk	ɪə	gb	ɪ̯	sk
ʏ (320)	de	ʊ (321)	cz, de, gb, ru, us	ʊɔ	sk
ʊə	gb	ʊ̯	sk	ə (322)	cz, de, gb, ru, us
əʊ	gb	əː	de*	ɵ (323)	ru
ɐ (324)	de, ru	æ (325)	de, gb, ru, sk, us	ɜ (326)	us
ɜː	de*, gb				

Obrázek 3: Fonetická abeceda IPA, která byla použita v experimentech. Ke každému symbolu IPA jsou uvedeny i jazyky, se kterými se v experimentech pracovalo. Hvězdička značí, že se foném vyskytoval v cizích slovech daného jazyka.

Realizace transkripce v systému TTS

V případě češtiny se transkripce většinou řeší pomocí kontextových pravidel a slovníků pro cizí slova [2]. Kontextová pravidla se zaměřují na vliv okolních slov na výslovnost konkrétního slova (viz tabulka 3) a slovníky pro cizí slova zahrnují přepis cizích slov do češtiny. Naproti tomu transkripce angličtiny je více založena na slovnících, protože výslovnost anglických slov je méně předvídatelná a těžko se dají použít kontextová pravidla.

Angličtina	**Čeština**
Have you read[**rEd**] the book?	Most[**most**] k[**k**] věži[**vjeZi**].
You don't like to read[**ri:d**].	Most[**mozd**] k[**g**] dolu[**dolu**].
Don't touch live[**laIv**] wires.	Bez[**bez**] sdružení[**zdruZeJi:**].
I can't live[**lIv**] alone.	Bez[**bes**] vzpírání[**fspi:ra:Ji:**].
I leave[**lif**] today[**t@deI:**].	Jásot[**ja:sot**] politiků[**politiku:**].
Those[**D@Us**] shoes[**Su:z**].	Hvizd[**hvist**] politiků[**politiku:**].

Tabulka 3: Ovlivnění transkripce na základě kontextu. Výslovnost je napsaná v abecedě SAMPA.

Slovníky i kontextová pravidla jsou zástupci expertního přístupu (pravidla jsou navržená člověkem – expertem). Alternativně lze použít trénovatelné přístupy, jakými jsou například systémy založené na FST (*finite-state transducers*, například nástroj Phonetisaurus [3]), anebo neuronové sítě. Neuronové sítě pro tuto úlohu řeší tzv. sequence-to-sequence problém. Pro něj se používají architektury *encoder-decoder* [4], anebo modernější *transformer* [5].

V *end-to-end* systémech může být krok fonetické transkripce úplně vynechán, neboť jeho modely jsou trénovány na dvojicích <text, audio>. To znamená na jednu stranu zjednodušení pro tvorbu hlasového systému, zároveň to však dělá úlohu složitější a zvyšuje to nároky na komplexitu modelu a jeho kapacitu, neboť se musí učit navíc vztah mezi textem a jeho fonetickou podobou.

2.3 Syntéza řeči

Poslední úlohou je již samotná *syntéza řeči*, která vytváří akustickou reprezentaci řeči pro daný fonetický zápis v podobě audio signálu. Audio signál je obvykle reprezentován pomocí pulzně kódové modulace (PCM). Ta spočívá v pravidelném vzorkování hodnoty zvukového signálu (mechanického vlnění) v čase. Na hudebních nosičích se obvykle používají vzorkovací frekvence 48 kHz či 44,1 kHz. Na rozdíl od hudby nejsou u řeči nutné tak vysoké frekvence. Obvykle se používá vzorkovací frekvence 16 kHz či dnes častěji 24 kHz (popř. 22050 Hz).

Způsob tvorby řečového signálu závisí na použité metodě. Může být poskládán z již existujících vzorků z řečové databáze (*konkatenační metoda*), vygenerován pomocí signálových metod (*parametrická syntéza*), anebo vygenerován přímo jako výstup neuronových sítí (např. *WaveNet*). Poslední přístup je předmětem této práce.

2.4 Hodnocení kvality syntézy řeči

V úlohách syntézy řeči často chceme porovnat kvalitu výsledné řeči mezi více systémy syntézy řeči (nebo vyhodnotit různá nastavení, případně „vylepšení“ stejného systému). Toto porovnání nám umožňuje vybrat nejlepší systém nebo nastavení parametrů pro danou aplikaci. Lze to udělat objektivními způsoby nebo subjektivními. Objektivní metody se zaměřují na měření konkrétních parametrů řeči, zatímco subjektivní metody se zaměřují na poslechové hodnocení kvality řeči posluchači.

Objektivní metody zahrnují měření různých parametrů jako jsou například hlasitost, intonace, rychlost řeči a akustická rozdílnost od reálných nahrávek.

Pro porovnání vůči reálným nahrávkám je možné použít frekvenční melovský spektrogram pro výpočet akustické vzdálenosti. Této metrice se říká MCD (*mel cepstral distortion*) [6],

$$MCD(C, \hat{C}) = \frac{10\sqrt{2}}{ln10} \frac{1}{T} \sum_{t=1}^{T} \sqrt{\sum_i (C_{ti} - \hat{C}_{ti})^2}, \qquad (1)$$

kde C jsou kepstrální koeficienty reálného řečového signálu, $\hat{C}$ jsou kepstrální koeficienty syntetizovaného řečového signálu a T je počet framů.

Další metrikou, kterou lze použít pro porovnání s reálnými nahrávkami, je PESQ (*Perceptual Evaluation of Speech Quality*) [7], což je algoritmus, který analyzuje audio signál a vypočítává skóre podobné MOS (1-špatné, 5-výborné, viz dále).

Objektivní metody jsou výhodné, protože jsou nezávislé na posluchači a jsou snadno měřitelné. Avšak nedokážou poskytnout informace o tom, jak jsou nahrávky přirozené, jak budou znít, jestli bude člověk schopen porozumět a podobné jevy, které souvisí s lidským mozkem a vědomím.

Některé pokročilé objektivní metody používají rozpoznání řeči (*ASR*) [8] jako nástroj pro objektivní hodnocení srozumitelnosti syntetizované řeči. Tato metoda se zaměřuje

na měření schopnosti rozpoznat text v syntetizované řeči. Většinou se užívá metoda WER (*Word Error Rate*), která počítá počet chybně rozpoznaných slov v porovnání s původním textem. I když tato metoda poskytuje objektivní hodnocení srozumitelnosti, není tak vypovídající jako skutečný poslechový test. Lidský mozek totiž může kompenzovat chyby v rozpoznání slov a porozumět řeči, kterou by automatický rozpoznávač nebyl schopen rozpoznat (nebo obráceně).

Subjektivní metody zahrnují poslechové testy. Tyto metody poskytují informace o tom, jak se výsledná řeč jeví posluchačům, což může být důležité pro hodnocení kvality řeči.

Je vhodné kombinovat objektivní a subjektivní metody při hodnocení kvality syntetizované řeči.

2.4.1 Poslechové testy pro měření přirozenosti

Test průměrného názoru (MOS)

Účastníci poslouchají vzorky syntetizované řeči a hodnotí je na stupnici na základě celkové kvality. Většinou se používá pěti bodová Likertova stupnice [9]. Tato metoda [10] umožňuje zjistit subjektivní hodnocení kvality řeči, které se může lišit v závislosti na posluchači. Tato metoda má i další varianty například CMOS (porovnání, viz CCR dále), DMOS (degradace), MOS_c (konverzační), SMOS (podoba barvy hlasu s referenčním řečníkem).

ABX test

Účastníci poslouchají tři vzorky řeči A, B a X (které je náhodně vybrané z A a B) a musí rozhodnout, zda X bylo vytvořeno systémem A nebo B. Tato metoda měří schopnost posluchačů rozlišit mezi dvěma vzorky řeči a je také užitečná při srovnávání různých metod syntézy řeči.

Test CCR

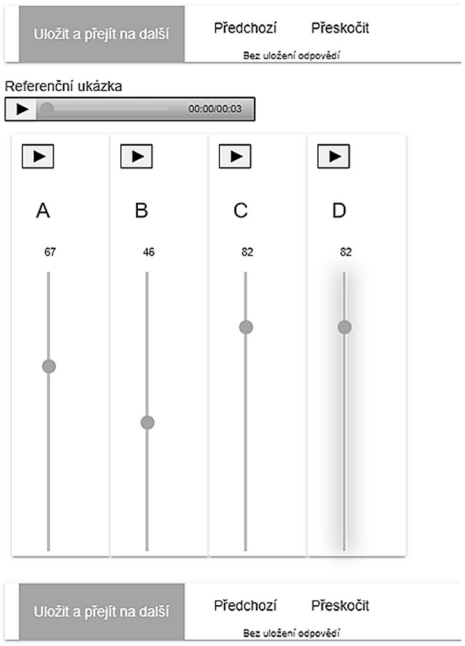
Někdy také značeno jako AB test nebo CMOS. V testu CCR (*Comparison Category Rating*) účastníci poslouchají různé vzorky syntetizované řeči a udávají, který z nich upřednostňují a jak moc. Tato metoda se používá k hodnocení, jak posluchači preferují jednotlivé vzorky řeči. Používají se různé stupnice pro vyjádření podobnosti. Nejzákladnější (lepší, stejné, horší) ale i vícestupňové (například pětibodové).

Test MUSHRA

Posluchači poslouchají různé vzorky zvuku, například skutečnou řeč a syntetizovanou řeč, a hodnotí je na stupnici od 0 do 100. V této stupnici se hodnota 100 interpretuje jako naprosto nejvyšší kvalita (přirozená řeč), 0 jako neakceptovatelná kvalita. V rámci testu mají posluchači k dispozici referenční nahrávku jako tzv. „horní kotvu“ kvality. V některých případech se používá i kotva spodní jako ukázka nejnižší kvality. Kotvy se používají, aby všechny uživatelské odpovědi měly stejný rozsah a byly vzájemně porovnatelné.

MUSHRA [11] test umožňuje srovnávat vygenerované vzorky syntetické řeči s reálnou řečí a získat subjektivní hodnocení od posluchačů. Tato metoda je často používána pro hodnocení kvality syntetizované řeči a dalších audio systémů.

Hlavní výhodou testu MUSHRA (obrázek 4) oproti testu průměrného názoru (MOS) je, že má jednodušší realizaci, protože vyžaduje méně účastníků k získání statisticky významných výsledků [12]. Studie [13] se zabývá potenciální zaujatostí ve výsledcích poslechových testů MUSHRA. Tento test byl použit například v kapitole 8.2.3.



Obrázek 4: Příklad poslechového testu MUSHRA.

2.4.2 Poslechové testy pro měření srozumitelnosti

MRT

Modified Rhyme Test (MRT) [14] je test, který se používá pro hodnocení schopnosti rozponat a přepsat slova ve skupině velmi podobných slov, které navzájem tvoří rým.

SUS

Tato metoda se používá k hodnocení, jak dobře posluchači rozumějí syntetizované řeči. Tento test byl použit například v kapitole 10. Při testu podle metodologie SUS (*Semantically Unpredictable Sentences)* [15] posluchači poslouchají a přepisují věty (viz obrázek 52), které byly vytvořeny systémem syntézy řeči. Přepsaný text je poté srovnán s originálem a výsledek úspěšnosti je například poměr správně přepsaných slov. Čím vyšší počet správně přepsaných slov, tím lépe srozumitelný je systém syntézy řeči.

Lidský mozek má schopnosti domýšlet slova, která přeslechnul (například z důvodu šumu nebo neporozumění). Domýšlení probíhá z lingvistického kontextu a je velmi

přesné, neboť lidská řeč a jazyk obecně mají v sobě mnoho redundance (jako příklad lze zmínit fakt, že lidé dokážou bez problému číst text bez diakritických znamének anebo text, jehož horní/spodní polovina není vidět). Tento jev komplikuje provedení testu. Proto se věty v SUS testu skládají ze slov, které se nedají vyvodit z kontextu a musí být rozpoznány nezávisle na sobě. Tím je zajištěno, že schopnost rozpoznat slovo je závislá pouze na schopnosti systému syntézy řeči produkovat dané slovo a ničím jiným. Přestože věty obsahují náhodná slova, jejich gramatická struktura odpovídá normálnímu textu (například obsahují podmět a přísudek).

2.5 Příprava řečového inventáře

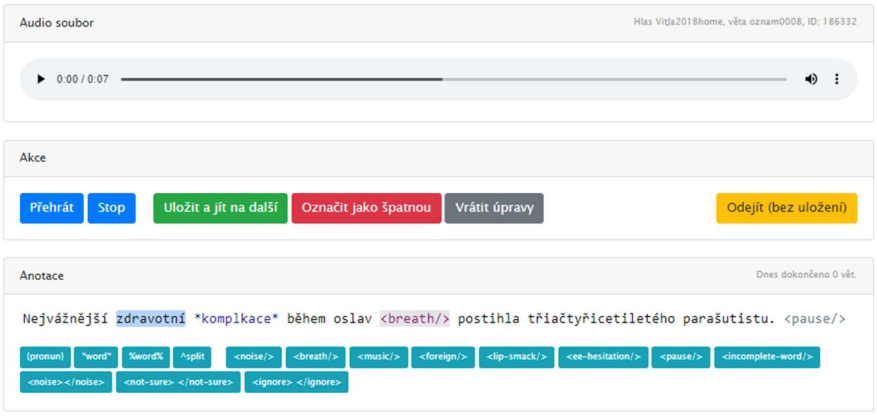
Pro vytvoření kvalitní syntézy řeči je důležité mít k dispozici kvalitní zdrojová data. Ty je možné získat několika způsoby. Buď lze použít veřejné dostupné řečové databáze (např. [16]), nebo je možné použít data, která byla původně určena k jinému účelu (např. audioknihy). Zde se ale může vyskytovat mnoho problémů, neboť mohou obsahovat i jiné zvuky než řeč, styl řeči nemusí odpovídat tomu, co požadujeme, mohou chybět fonetické kontexty atd. Nejlepší volbou (ale i nejnáročnější) je audio nahrávky nahrát v kontrolovaném prostředí (zvuková komora) pomocí dobře připravených textů.

Při přípravě řečového inventáře pro syntézu řeči je důležité si uvědomit, že kvalitní nahrávky nejsou všechno. Je nutné mít k nim adekvátní textový přepis, který umožní další analýzu a segmentaci řeči. Při nahrávání se může stát, že se řečníci přeřeknou nebo vysloví jiné slovo, což může mít negativní vliv na kvalitu dat. Pokud chceme mít opravdu kvalitní data pro syntézu řeči, je vhodné navrhnout a realizovat proces anotace.

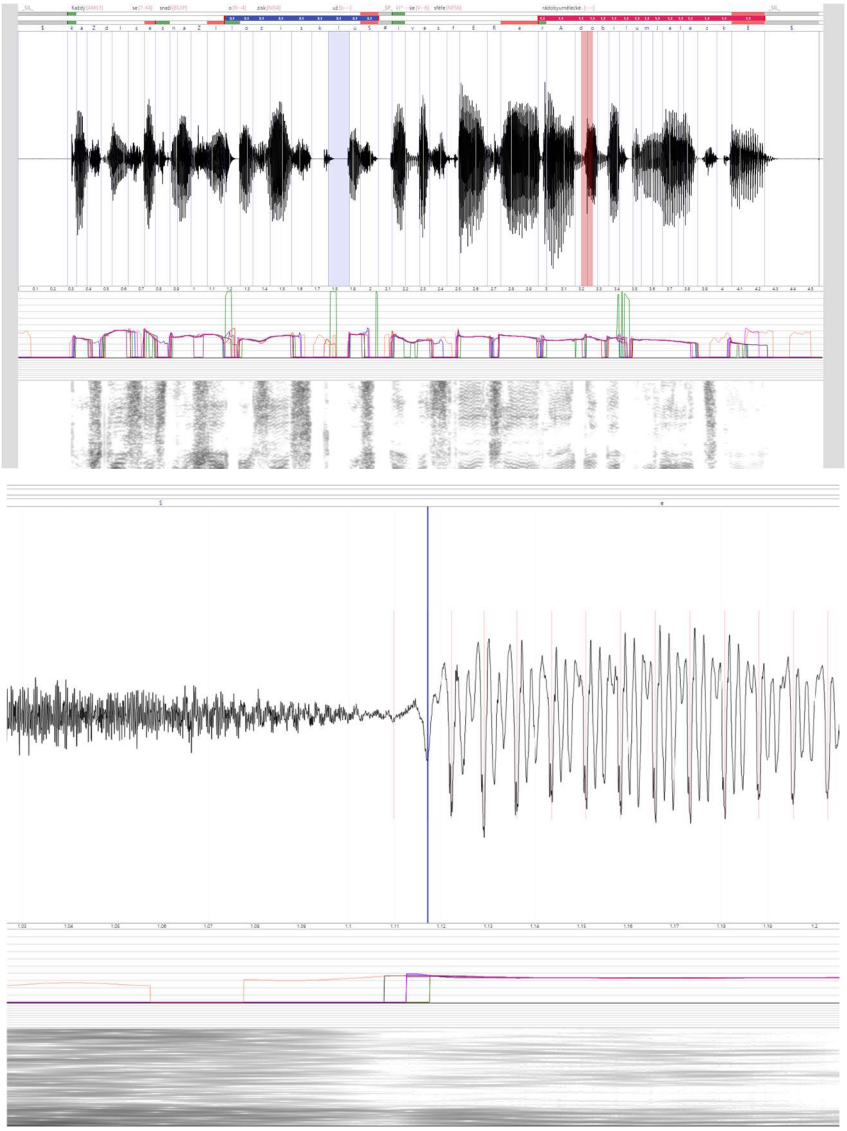
Anotace spočívá v tom, že lidští anotátoři poslouchají nahrávky řeči a kontrolují texty, přičemž označují a opravují různé jevy, jako například přeřeknutí, jinou výslovnost, pauzu a nadechnutí. Příklad z anotačního programu ukazuje obrázek 5. Tímto způsobem se zajistí, že data jsou co nejkvalitnější a odpovídají požadavkům experimentu.

Pokud je vstupem syntetizéru řeči fonetický text, je dále nutné přepsat řečová data do fonetické abecedy. Tuto úlohu lze řešit automaticky. Je ale důležité si uvědomit, že tyto automatické nástroje nemusí být vždy přesné a mohou vytvořit chyby. Proto je v průběhu anotace vhodné zároveň opravit i tento fonetický přepis.

U tradičních algoritmů syntézy řeči bylo nutné mít fonetický text zarovnaný s audio signálem (tj. pro každý foném znát čas, od kdy do kdy se v dané větě vyskytuje). Stejně jako u předchozích kroků, i zde platí, že automatické systémy mohou dělat chyby a manuální kontrola (obrázek 6) vede ke kvalitnějším datům.



Obrázek 5: Ukázka z programu pro anotaci dat. Obrázek pochází z programu *SpeechLab*. Hvězdičky označují „přeřek“, <breath> označuje slyšitelný nádech.



Obrázek 6: Ukázka manuální opravy segmentace. Obrázek pochází z programu *SpeechLab*.

2.6 Melovský spektrogram

Frekvenční spektrogram zobrazuje intenzitu jednotlivých frekvencí zvukového signálu v čase. Jedná se o graf, kde na vodorovné ose je čas a na svislé ose je frekvence. Intenzita každé frekvenční složky je dána barvou daného bodu. Jde o názornou reprezentaci řeči v obrázkové podobě a umožňuje analyzovat řeč mnohem lépe než graf samotného audio signálu.

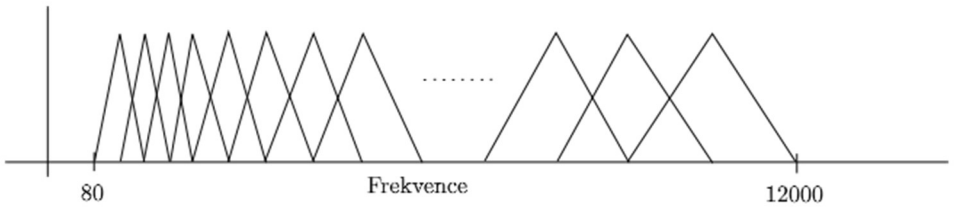
K výpočtu frekvenčního spektrogramu se používá *Diskrétní Fourierova transformace* (rovnice 2) obvykle realizovaná algoritmem FFT. Ten se provádí v posuvném okénku postupně na celý signál. Spektrogram nezachovává všechny informace – jedná se o ztrátové zobrazení, neboť neobsahuje informace o fázi jednotlivých frekvencí.

$$\hat{x}(k) = \sum_{n=0}^{n-1} x(n) e^{-i\frac{2\pi k}{N}n}, \quad k = 0, \ldots, N-1 \qquad (2)$$

Motivací pro použití spektrogramu je, že lépe zobrazuje dlouhodobé závislosti, neboť jsou informace reprezentovány kompaktněji ve 2D matici. Zobrazení řeči ve spektrogramu blíže odpovídá tomu, jak člověk vnímá řeč – tj. jako frekvence v čase. V lidském uchu rovněž nejsou receptory, které vnímají zvuk jako signál v čase, ale naopak jako úrovně jednotlivých frekvencí v čase. Spektrogram tak má blíže k fungovaní lidského sluchu.

Spektrogram je často používán jako akustická reprezentace řeči v neuronových sítích. Úlohu syntézy řeči lze rozdělit na dvě dobře oddělitelné části: akustický model, který vygeneruje řeč do formy spektrogramu, a vokodér, který jej převede do řečového signálu.

Často používaný je také *melovský spektrogram*, což je spektrogram, který byl pomocí melovských filtrů zredukován na nižší dimenzi. Tyto trojúhelníkové filtry byly navrženy tak, aby co nejlépe pokryly frekvence důležité pro řeč.



Obrázek 7: Rozložení mel filtrů ve frekvenční oblasti. Filtry jsou trojúhelníkové.

Obrázek 7 naznačuje, jak je spektrum pokryto filtry – nižší frekvence jsou pokryty hustěji, zatímco vyšší frekvence řídčeji. To umožňuje obsáhnout celé spektrum v menší dimenzi. Nižší frekvence jsou mnohem důležitější z hlediska porozumění řeči. Hlasivková frekvence a formanty jsou obsaženy právě v nižší oblasti frekvenčního spektra. Jako

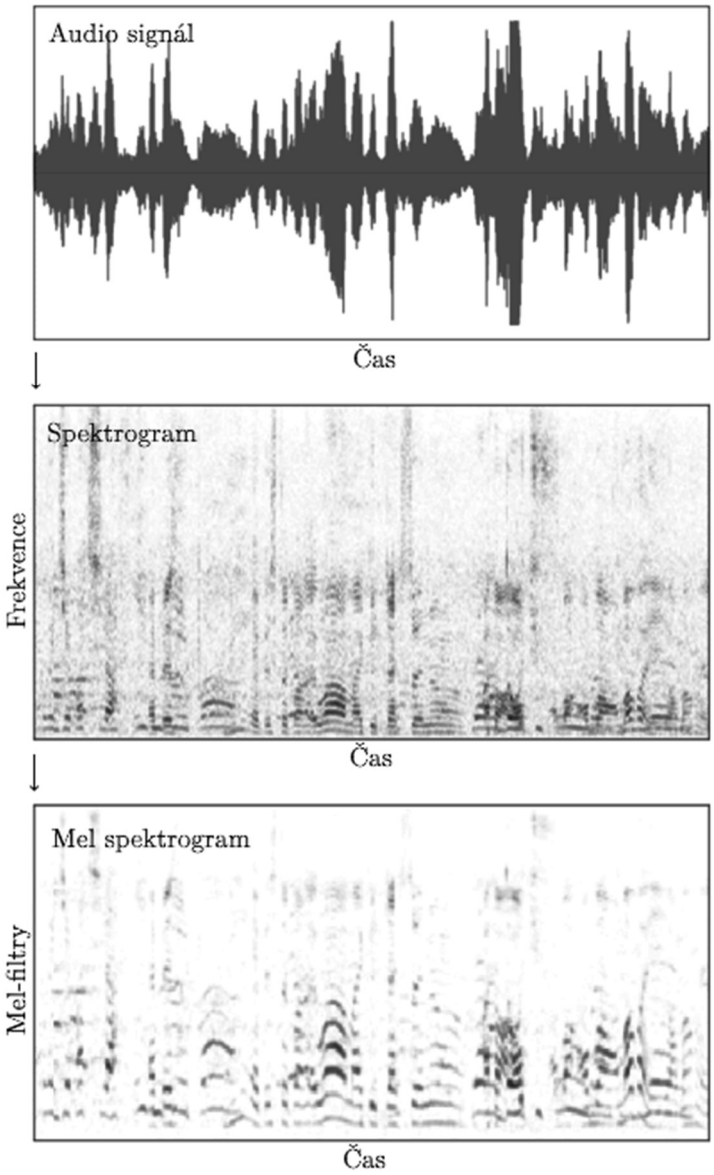
ilustraci toho lze zmínit, že staré telefonní linky používaly vzorkování 8 kHz, tj. maximální slyšitelná frekvence byla 4 kHz, a přesto nebyl problém se srozumitelností.

Spektrogram má obvykle dimenzi 512 či 1024. Melovský spektrogram (viz obrázek 8) se většinou používá v dimenzi 80 – redukce je značná, ale ztráta důležité informace je zanedbatelná. Pro použití v úloze syntézy řeči nebo automatického rozpoznávání řeči je tato forma vhodná, neboť umožňuje pracovat s nižší dimenzí.

Rovnice pro převod frekvence z melovského prostoru do lineárního prostoru a zpět:

$$F_{mel} = 2595 * \log_{10}\ (1 + \frac{F_{lin}}{700}) \qquad (3)$$

$$F_{lin} = 700 * (10^{\frac{F_{mel}}{2595}} - 1) \qquad (4)$$



Obrázek 8: Ukázka tvorby melovského spektrogramu.

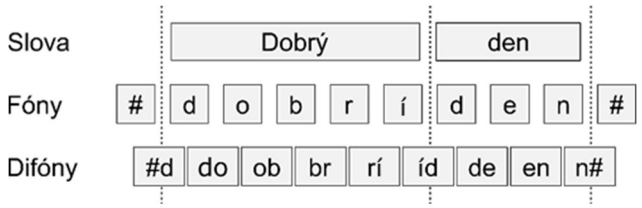
Kapitola 3

Tradiční metody syntézy řeči

Tato kapitola se věnuje tradičním metodám syntézy řeči. Novým metodám jsou věnovány další kapitoly práce. Kromě dvou dále zmíněných existovaly v minulosti další metody syntézy řeči, které se dnes již téměř nepoužívají, a proto zde nebudou zmíněny. Podrobněji o těchto metodách hovoří například [17], [1].

3.1 Konkatenační syntéza

Syntéza výběrem jednotek (*unit selection*) je hlavním zástupcem konkatenačních metod. Principem je spojování úseků, které pochází z již nahraných vět od řečníka, jehož hlas má syntéza napodobovat. Tyto úseky se vhodně spojí a vzniká tak syntetická řeč. Jako úsek se obvykle používá hláska nebo difón (od středu fónu ke středu dalšího fónu, viz obrázek 9 a 10).



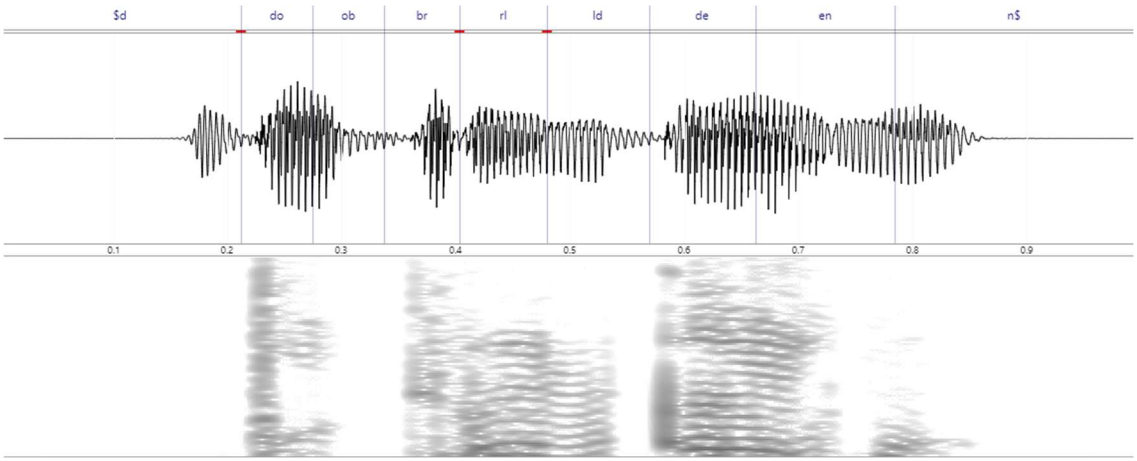
Obrázek 9: Příklad rozdělení slov na fóny a difóny.

Jednotky jsou uloženy v databázi jednotek (řečovém korpusu), odkud se v čase syntézy vybírají nejlepší kandidáti, kteří se potom spojí do výstupní věty. Je zde kladen velký důraz na správný výběr jednotek. Při spojování se provádí minimum signálových modifikací. Signál proto zachovává přirozenost a původní parametry řečníka. Schéma syntézy touto metodou ukazuje obrázek 11.

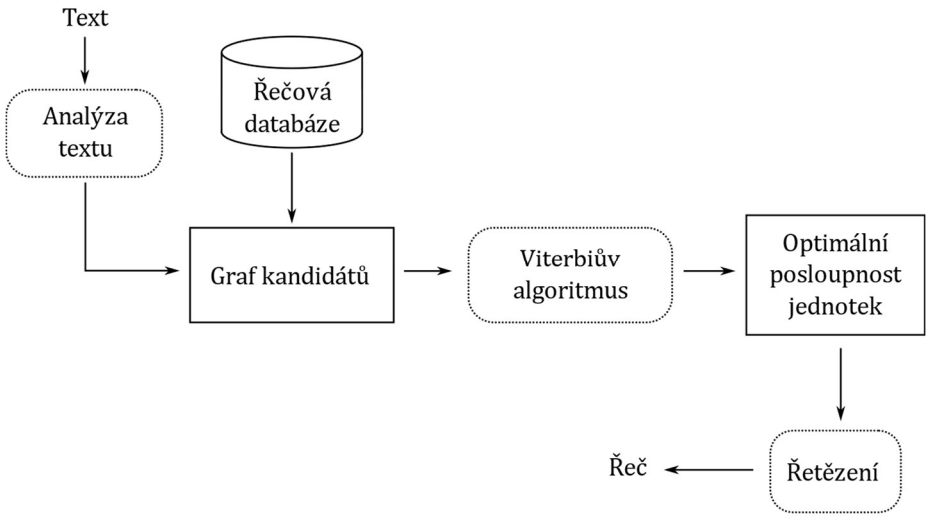
Pro vyjádření míry vhodnosti spojení dvou jednotek je nadefinována tzv. hodnoticí funkce (*cost function*), která se obvykle skládá ze dvou částí. Cena cíle (*target cost*) je veličina udávající kontextuální shodnost jednotky s konkrétním kandidátem. Správné kontextové okolí a pozice v původní větě vedou k nízké ceně cíle. Druhou složkou hodnoticí funkce je cena spojení (*join cost*). Na tu lze nahlížet jako na vzdálenost akustických parametrů signálu v místě spojení. V ideálním spojení se shoduje základní

frekvence hlasu F_0 i tendence jejích průběhů v okolí a také spektrum signálu v daném místě spojení. Pro popis spektra se používají například koeficienty *MFCC* (mel frequency cepstrum coefficients).

Algoritmus výběru jednotek hledá nejlepší posloupnost jednotek z řečové databáze tak, aby kumulativní součet hodnoticí funkce na jednotlivých spojích byl co nejmenší. Prohledávání stavového prostoru všech kombinací jednotek je časově náročná úloha. Pomocí určitých optimalizací a heuristik lze však náročnost značně snížit. Jako základ řešení prohledávání prostoru kombinací se obvykle používá *Viterbiův algoritmus* [18].



Obrázek 10: Příklad rozdělení slov na difóny v reálné větě vytvořené pomocí syntézy řeči unit selection. V horní polovině obrázku je výsledný signál řeči. Dolní polovina zobrazuje frekvenční spektrogram. Obrázek pochází z programu *SpeechLab*.



Obrázek 11: Schéma syntézy řeči pomocí metody unit selection.

Řečové jednotky jsou uloženy v inventáři. V něm jsou uloženy zdrojové promluvy a z nich vypočtené parametry: průběhy F_0, energie, doba trvání a spektrální charakteristiky (např. MFCC). Zdrojové promluvy jsou rovněž nasegmentované, tj. jsou rozděleny podle jednotlivých hlásek pomocí časových značek, které udávají začátky jednotlivých fonémů. Vytváření inventáře řečových jednotek probíhá zcela automaticky. Díky tomu lze vytvářet obrovské inventáře a zajistit dostatečný počet kandidátů pro každou jednotku. Pro kvalitní hlas je vhodné mít řečová data v řádech desítek hodin záznamu.

Řeč produkovaná metodou unit selection vyniká vysokou kvalitou a přirozeností. Je to dáno tím, že se signál nijak nemodifikuje, ale pouze se vhodně skládá. Generování neznělých a šumivých tónů zde není problémem, tak jako je tomu u modelových metod. V místě spojování vzniká bohužel místo potenciálních problémů. Při špatném zřetězení jednotek se v jinak bezchybné řeči vyskytne lokální propad v kvalitě (řečový artefakt), což působí velmi rušivě a kazí to celkový dojem z celé věty.

Občasné chyby v řetězení jsou velkou nevýhodou metody unit selection. Tímto problémem se například zabývá práce [19] a [20]. Mezi další nevýhody patří velké paměťové, datové a výpočetní nároky. Pomocí unit selection se také obtížně vytváří řeč z jiné domény, než ke které jsou k dispozici řečové nahrávky (obvykle se používá „neutrální“ řeč, resp. čtená řeč „zpravodajského charakteru“). Obtížná je i změna stylu řeči.

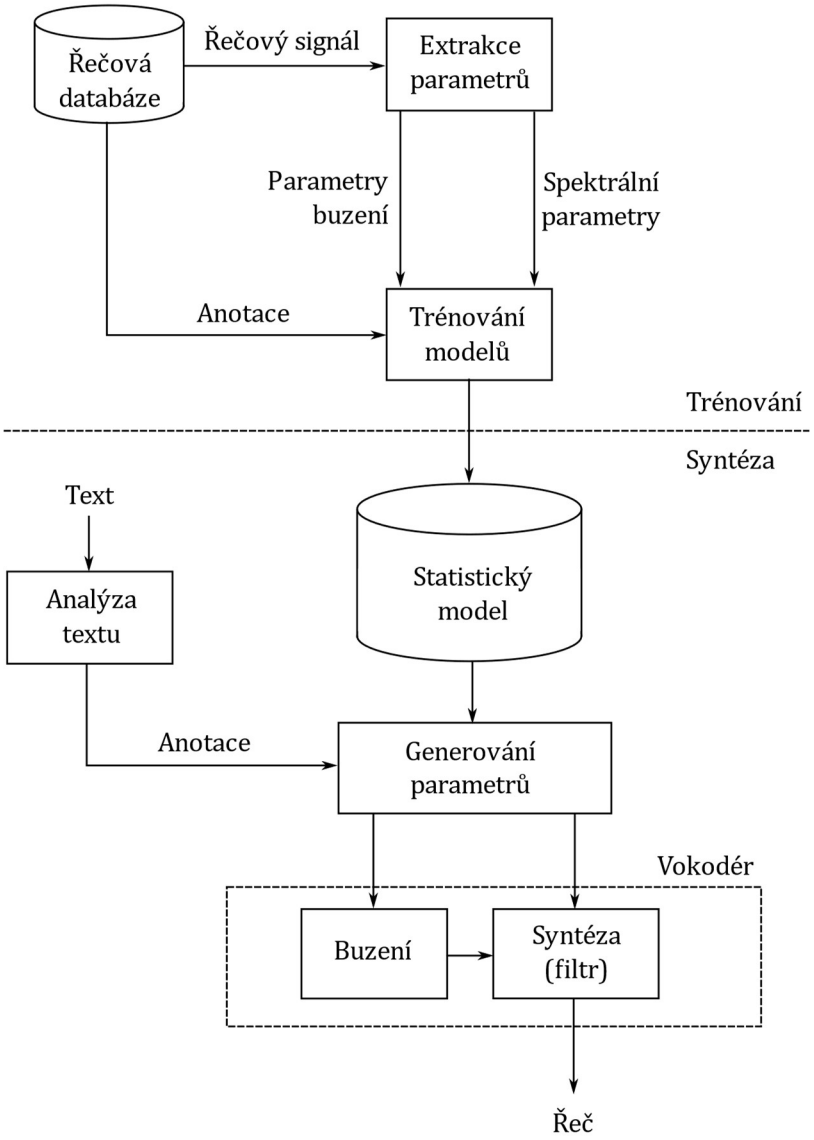
Paměťové a výpočetní nároky v dnešní době stále rychlejších počítačů nehrají tak významnou roli. Další nevýhodou je, že vytvoření velkého řečového inventáře, který tato metoda vyžaduje, je velmi časově náročné a drahé, protože je nutné provádět ruční korekce anotací, platit zvukovou komoru a honorář řečníkovi.

Metoda unit selection se dnes používá v případech, kde je požadavek na neutrální, emotivně nezabarvenou, co nejvíce kvalitní a přirozenou řeč. A také tam, kde není problém zajistit vyšší výpočetní a paměťové zdroje.

3.2 Statistická parametrická syntéza HMM

Parametrická syntéza [21] je postavena na modelování parametrů řeči namísto spojování již existujících úseků. Řeč je převedena do posloupnosti parametrů popisujících frekvenční charakteristiku signálu. Z těchto parametrů lze řeč zpětně zrekonstruovat, ideálně s co nejnižší ztrátou kvality. Schéma trénování syntézy pomocí parametrické metody ukazuje obrázek 12.

Řečový signál je generován pomocí buzení a filtru. Parametry buzení (hlasivková frekvence, míra periodicity) a filtru (frekvenční charakteristika) jsou pro každou jednotku jiné.



Obrázek 12: Schéma trénování a produkce řeči pomocí parametrické syntézy.

Pro generování výsledné řeči je z posloupnosti stavů, které jsou dány vstupní posloupností fonémů, generován akustický signál na základě parametrů konkrétního stavu. Tento proces se provádí přes vokodér. Kvalita vokodéru má velmi významný dopad na kvalitu řeči, proto je jeho výzkumu věnována velká pozornost. Mezi nejpoužívanější tradiční vokodéry patřily např. STRAIGHT [22], Vocaine [23], AhoCoder [24] a WORLD [25]. Kromě tradičních vokodérů, které používají signálové algoritmy, se dnes používají tzv. *neurální vokódéry*, které místo toho používají neuronovou síť, která generuje akustický signál.

Parametrická syntéza se dříve používala velice často. Jejími výhodami jsou konstantní a dobrá kvalita řeči, možnost modelovat různé emotivní či jiné změny hlasu („pouhou“ změnou parametrů modelu) a plynule měnit parametry řeči (rychlost, výška, tempo, barva hlasu).

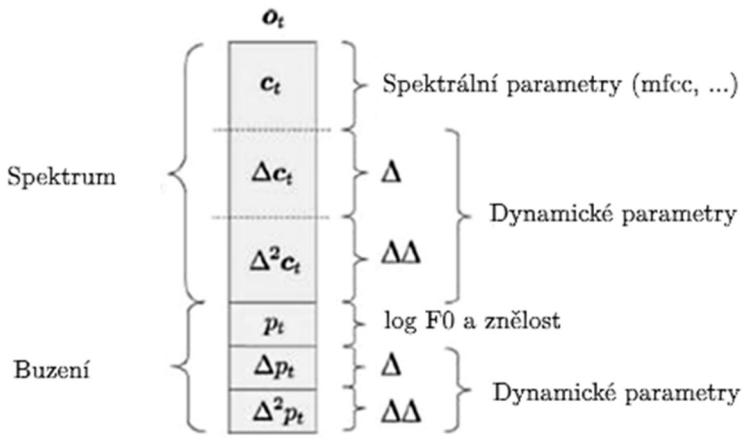
Parametrická syntéza také umožňuje adaptovat hlas na konkrétního řečníka. K tomu je potřeba pouze malé množství nahraných promluv od nového řečníka. Lze tak snadno rozšiřovat databázi hlasů. Mezi další výhody patří nízká výpočetní náročnost (při syntéze) a malé paměťové nároky. Toho lze využít v zařízeních s omezenou výpočetní kapacitou, jako jsou například mobilní telefon či vestavěný mikropočítač.

Nevýhodou je nižší kvalita akustického signálu, který pak zní velmi monotónně až roboticky. Metoda se dnes dá považovat za mezikrok k neurální syntéze.

Dříve se pro modelování používalo HMM (*Hidden Markov model*). Každá hláska je rozdělena na několik stavů a každý stav popisuje statistické rozdělení hodnot parametrů. Parametry jsou statické (průměr a odchylka) a dynamické (*delta* a *delta-delta* rozdíly hodnot). Trajektorie parametrů se generují tak, aby co nejlépe odpovídaly statistickým hodnotám jednotlivých HMM stavů a zároveň respektovaly dynamické parametry. Obrázek 12 platí i pro HMM syntézu, přičemž modelem jsou v tomto případě kontextově závislé markovské modely pro akustiku a trvání.

Vstupem HMM syntézy jsou lingvistické příznaky. Vzhledem k jejich velkému množství se modely seskupují pomocí rozhodovacích stromů. Výstupní vektor (obrázek 13) se skládá ze spektrálních parametrů a parametrů buzení.

HMM syntéza trpí neduhem přílišného vyhlazení signálu, neboť parametry jsou reprezentovány statistickým průměrem z natrénovaných dat. Kvůli tomu zní výstup HMM syntézy trochu „plechově“ a nepřirozeně. Problém vyhlazení signálu částečně řeší techniky jako například normalizace *globální variance* [26].



Obrázek 13: Složení výstupního vektoru HMM syntézy.

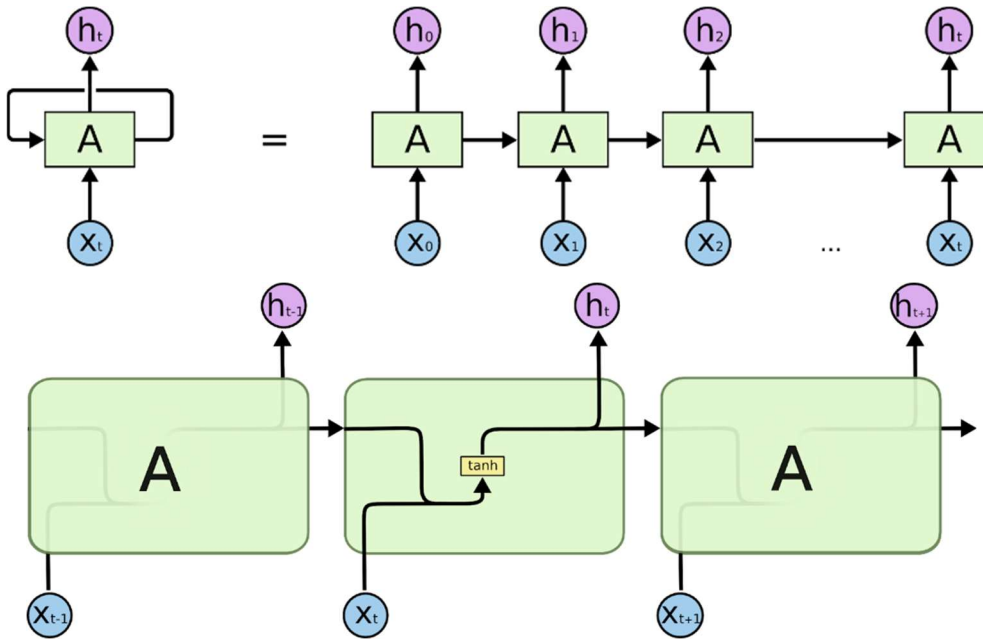
3.3 Parametrická syntéza pomocí neuronových sítí

HMM modely jsou dnes nahrazeny neuronovými sítěmi. Princip je stejný, akorát místo HMM modelů generujících posloupnost parametrů použijeme neuronovou síť. Neuronové sítě jsou schopny se naučit komplexnější a obtížnější úlohy a závislosti mezi vstupem a výstupem. Dokážou tak generovat posloupnost parametrů, které mají po převedení na řečový signál lepší akustickou kvalitu, než měly HMM modely.

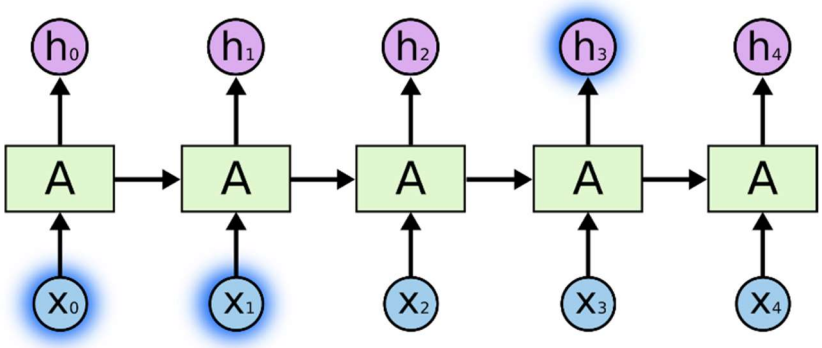
Nejvíce používanou náhradou HMM syntézy je parametrická metoda s využitím rekurentních neuronových sítí (obrázek 14). Ty jsou mocnější než modely HMM. Rekurentní neuronové sítě se dokážou efektivně učit z trénovacích dat akustické příznaky používané vokodérem, což vede k produkci kvalitnější umělé řeči. Rekurentní sítě používají vnitřní stav pro uchování znalostí z historie. Pracují tedy nad sekvencemi, nikoliv nad každým časovým okamžikem zvlášť.

Při trénování neuronové sítě je informace o událostech, které vznikly ve velmi vzdálené minulosti, utlumena mnoha iteracemi rekurentní vrstvy. Každá iterace snižuje vliv dřívějších událostí ve prospěch pozdějších. Gradient pro trénování je značně utlumen, anebo má naopak tendenci explodovat do extrémních hodnot, které rozbijí trénování.

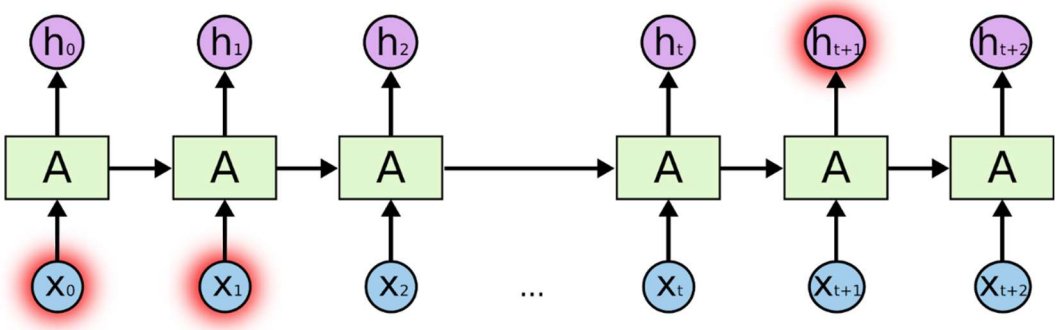
Důsledek problému je ten, že rekurentní sítě fungují dobře pro problémy, kde jsou příčina a následek velmi blízko po sobě (obrázek 15). Naopak velmi problematické jsou úlohy, kde je třeba do stavu uložit informace na dlouhou dobu (obrázek 16).



Obrázek 14: Schéma základní rekurentní sítě.



Obrázek 15: Uložení informace o nedávné události do stavu rekurentní sítě.



Obrázek 16: Problematický scénář, výstup sítě je ovlivněn dávnou událostí. Rekurentní sítě se tak musí naučit uchovávat informaci velmi dlouhou dobu.

LSTM sítě

LSTM (Long short-term memory) [27] je architektura, která byla navržena speciálně tak, aby řešila problém *mizejícího gradientu*. Jednodušší varianty rekurentních sítí jsou na tento problém velmi náchylné.

LSTM síť je schopna uchovávat dlouhodobé závislosti mezi vstupem a výstupem. Uvnitř neuronu je využíván systém bran a hradel (obrázek 17), které regulují tok informací do a vně buňky. Zároveň usnadňují tok gradientu skrze stav do minulosti a usnadňují trénování.

$$i_t = sigmoid\ (w_i[h_{t-1}, x_t] + b_i) \tag{5}$$

$$f_t = sigmoid\ (w_f[h_{t-1}, x_t] + b_f) \tag{6}$$

$$o_t = sigmoid\ (w_o[h_{t-1}, x_t] + b_o) \tag{7}$$

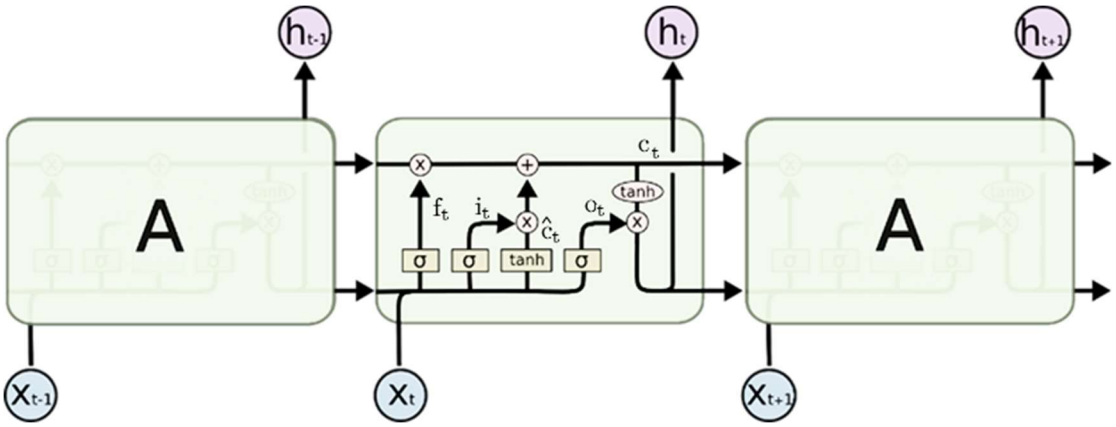
$$\widetilde{c_t} = tanh\ (w_c[h_{t-1}, x_t] + b_c) \tag{8}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \quad (9)$$

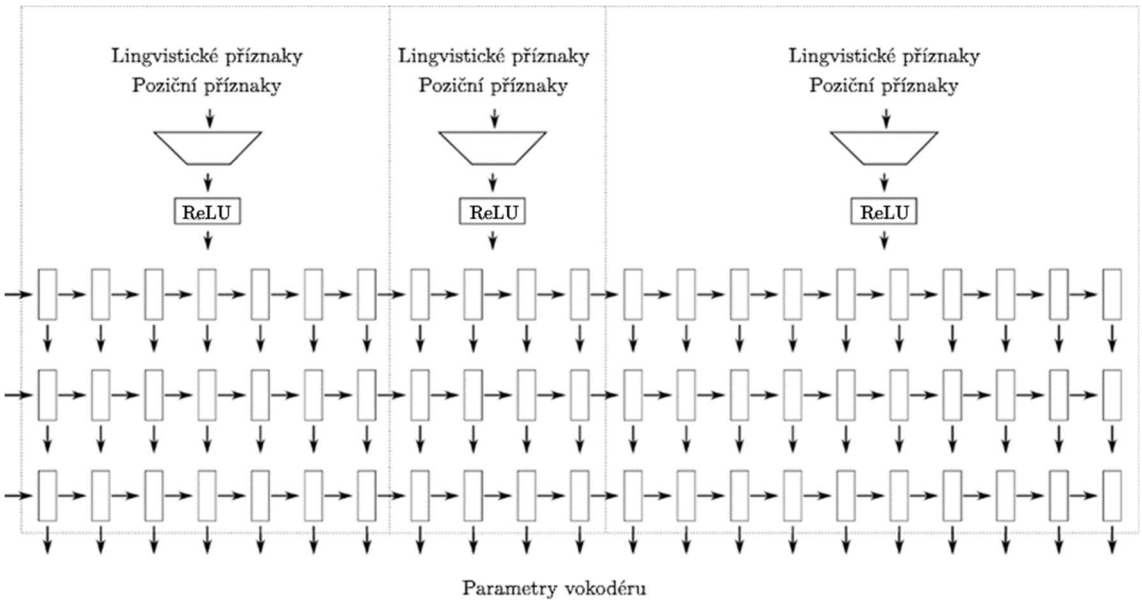
$$h_t = o_t * \tanh(c_t) \quad (10)$$

Rovnice LSTM buňky. i_t, f_t, o_t reprezentuje vstupní, zapomínací a výstupní hradlo (gate), w jsou váhy, b bias, h_{t-1} je předchozí výstup, x_t je vstup, c_t je aktuální stav, $\widetilde{c_t}$ je kandidát na nový stav, h_t je výstup buňky.

Kapacita sítě je dána počtem LSTM buněk ve vrstvě. Zvýšit se dá rovněž použitím více vrstev za sebou. Obvykle se pro úlohu syntézy řeči používají dvě až tři vrstvy. Obrázek 18 zobrazuje architekturu parametrické syntézy řeči pomocí třívrstvé jednosměrné rekurentní neuronové sítě.



Obrázek 17: Schéma LSTM sítě.



Obrázek 18: Schéma neuronové sítě pro parametrickou syntézu řeči. Aktivační funkce ReLU je popsána níže.

GRU sítě

Kromě LSTM existuje i podobně oblíbená varianta GRU (*Gated recurrent unit*) [28]. Její princip (obrázek 19) je obdobný s tím rozdílem, že má mírně zjednodušenou architekturu. Oproti LSTM, která má tři hradla, má GRU buňka pouze hradla dvě: *update* a *reset*. Pro každou vrstvu LSTM je nutné trénovat 4x více parametrů, než kolik by potřebovala základní RNN vrstva. Jelikož má GRU vrstva méně hradel, stačí trénovat pouze 3x tolik, co základní RNN. Díky tomu je její trénování a vyhodnocení (inference) nepatrně rychlejší a model uložený na disku počítače je o 25 % menší.

V některých úlohách je stejně úspěšná jako LSTM. Další výhodou je, že její stav je zároveň i výstupem (oproti LSTM, která má stav oddělený). Vrstvy GRU jsou použity i v architektuře WaveRNN, která bude zmíněna v dalších kapitolách.

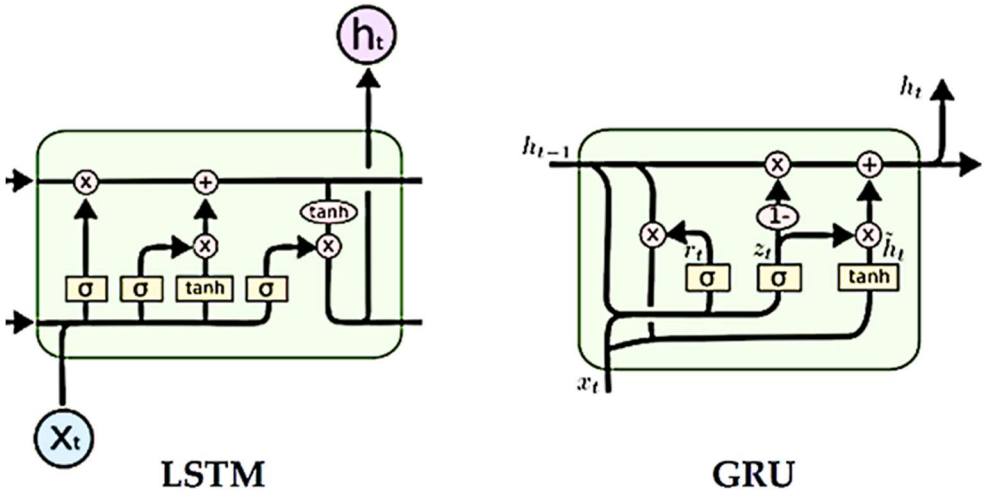
$$z_t = sigmoid\ (w_z[h_{t-1}, x_t] + b_z) \tag{11}$$

$$r_t = sigmoid\ (w_r[h_{t-1}, x_t] + b_r) \tag{12}$$

$$\widetilde{h_t} = tanh\ (w_h[r_t * h_{t-1}, x_t] + b_h) \tag{13}$$

$$h_t = \ z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t} \tag{14}$$

Rovnice GRU buňky. z_t, r_t reprezentuje *update* a *reset* hradlo, w jsou váhy, b bias, h_{t-1} je předchozí výstup, x_t je vstup, c_t je aktuální stav, $\widetilde{h_t}$ je kandidát na nový výstup, h_t je výstup buňky.



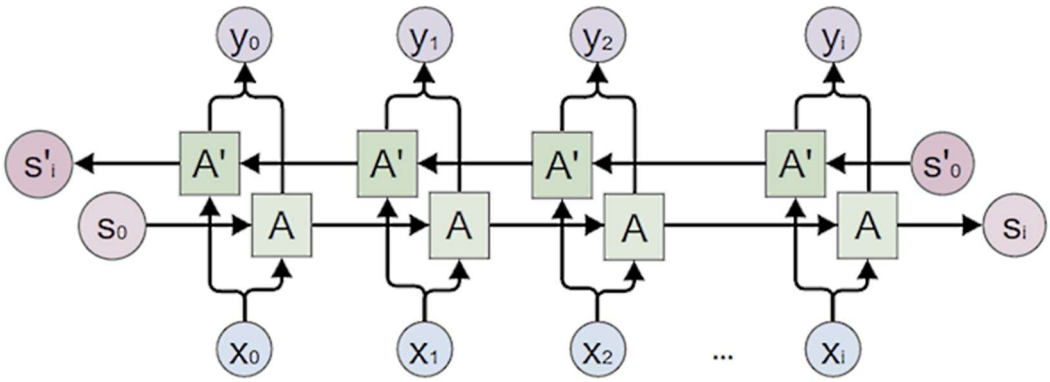
Obrázek 19: Rozdíl mezi architekturou LSTM a GRU.

Obousměrné rekurentní sítě

Rekurentní sítě zpracovávají informaci sekvenčně. V jejich stavu je proto uložena pouze historie. LSTM buňka proto „nevidí“ budoucí události ve vstupních datech.

Pokud máme k dispozici celý vstup, je vhodné použít obousměrnou rekurentní síť (*bi-directional,* viz obrázek 20). Ta je realizována jako dvě rekurentní sítě, kde jedna prochází sekvenci v jednom směru a druhá v opačném. Výstupy obou sítí se spojí a vznikne tak neuronová síť, která dokáže využít jak minulé, tak budoucí události ve vstupních datech pro svůj výpočet.

V systémech TTS se vstup obvykle zpracovává po větách (od pauzy k pauze), není problém spočítat vstupní parametry pro celou větu a využít obousměrné rekurentní vrstvy. Tento přístup však nelze aplikovat v inkrementální syntéze (např. [29]). To je metoda, kdy se řeč generuje kousek po kousku místo jedné velké části. Umožňuje začít generovat řeč okamžitě, i když celý text ještě není úplně připraven, tj. když neznáme budoucí text věty (pravý kontext). Využívá se pro aplikace v reálném čase, jako jsou například hlasoví asistenti.



Obrázek 20: Obousměrná rekurentní síť.

Lingvistické příznaky

Metody syntézy řeči využívají *lingvistický vektor příznaků*. Ten tvoří jejich vstupní data, která jsou ve vrstvách transformována do sekvence akustických příznaků pomocí natrénovaných vah. Vektor obsahuje informace extrahované z textu a umožňuje podmínit výstup modelu tak, aby generoval požadovanou řeč.

Lingvistické příznaky obsahují informace o hierarchii věty, typicky např:

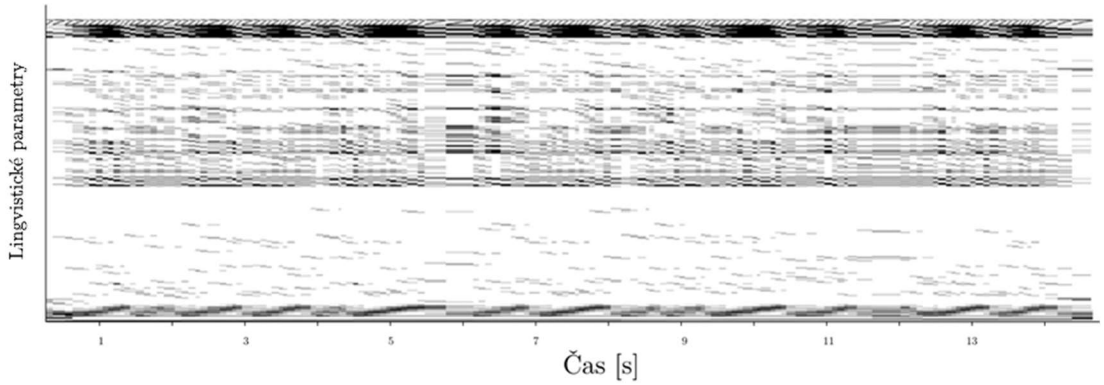
- Věta
  - Typ věty
  - Délka věty

- Fráze
    - Pozice fráze ve větě
    - Intonace
- Slovo
    - POS tag (*part of speech*)
    - Okolní závislosti (*dependency*)
    - Pozice slova ve frázi
- Slabika
    - Přízvuk
    - Tón (tonální jazyky)
- Fón
    - Identita
    - Znělost
- Časový úsek (*frame*)
    - Pozice (časový signál)

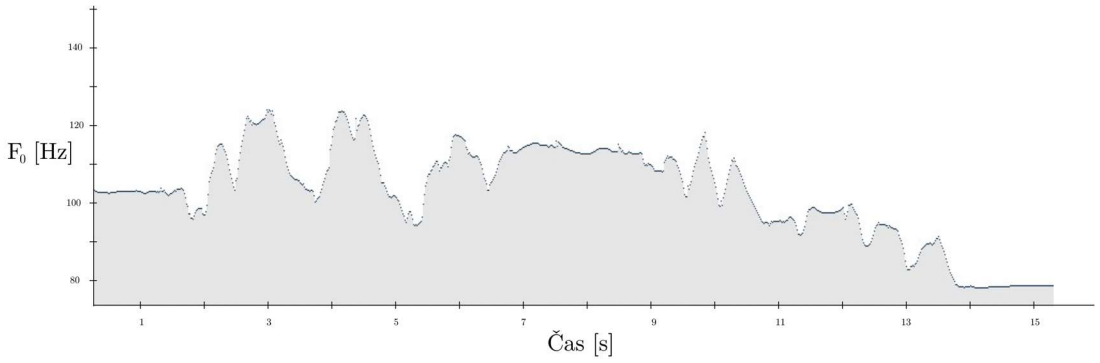
Tyto informace mohou být dále doplněny o kontext (např. předchozí fón, následující fón). Rekurentní sítě mají schopnost pracovat bez kontextu, neboť ten je možné uložit do vnitřního stavu. Jeho předáním na vstup se může zjednodušit problém a snížit tak počet neuronů.

Příznaky jsou obvykle unikátní pro každý fón ve větě. Poté, co jsou určena trvání fonémů, jsou tyto příznaky zarovnány do mřížky fixních časových úseků – *framů*, které mají stejné trvání (např. 5 ms). Příklad lingvistických příznaků ukazuje obrázek 21.

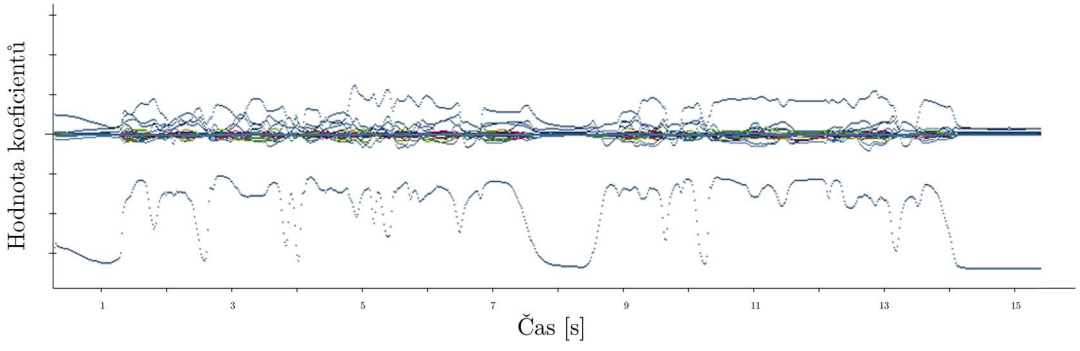
Výstupem modelu jsou parametry vokodéru. Jedná se zejména o hlasivkovou frekvenci a spektrální obálku. Vokodér tyto parametry použije pro vygenerování audio signálu řeči. Vokodér může být buď tradiční algoritmus založený většinou na principu *zdroj-filtr* využívající tradiční signálové přístupy a filtrace pro tvarování budícího signálu filtrem popisující hlasový trakt [30], anebo to může být neuronová síť natrénována v režimu pro vokodér. Obrázek 22 a obrázek 23 ukazují příklady průběhů vygenerovaných parametrů.



Obrázek 21: Průběh lingvistických příznaků. Intenzita jasu každého pixelu zobrazuje hodnotu jednotlivého parametru v rozsahu <-1,1>.



Obrázek 22: Ukázka průběhu trajektorie hlasivkové frekvence F_0 ve větě. Trajektorie má klesavou tendenci, neboť se jedná o oznamovací větu.



Obrázek 23: Průběh spektrálních parametrů. Rozsah hodnot je pro první koeficienty větší a postupně se snižuje. První koeficient v sobě nese energii, proto má jiný vertikální posun než ostatní.

3.3.1 Základní komponenty neuronových sítí

Hyperparametry

Neuronové sítě se skládají z trénovatelných parametrů (váhy). Kromě nich také obsahují hyperparametry. To jsou nastavení, která ovlivňují chování a účinnost neuronové sítě při učení a testování. Patří sem například počet vrstev a neuronů v síti, velikost trénovacího vektoru (*batch size*), rychlost učení, metoda optimalizace a poměr testovacích a trénovacích dat. Volba správných hodnot pro hyperparametry může mít vliv na úspěšnost modelu při řešení konkrétního úkolu. Tyto parametry jsou v průběhu trénování neměnné, na rozdíl od parametrů je není možné natrénovat.

Aktivační funkce

Aktivační funkce je nelineární matematická funkce, kterou každý neuron používá k výpočtu své výstupní hodnoty. Vstupní hodnoty neuronu jsou nejprve spočítány pomocí vah (*weights*) a přičtením konstanty (*bias*), a poté jsou předány do aktivační funkce. Aktivační funkce zpracovává vstupní hodnoty a vrací výstupní hodnoty, které jsou použity jako vstup pro další neurony v síti. Bez aktivační funkce by všechny neurony v síti pouze prováděly lineární kombinaci vstupů, což by značně omezilo schopnost sítě rozlišovat mezi různými vstupy a schopnost sítě se učit.

Základní aktivační funkce neuronových sítí jsou:

- **Sigmoidní funkce**: Používá se hlavně pro klasifikaci do dvou tříd. Vrací hodnotu v rozmezí od 0 do 1, což ji činí vhodnou pro použití jako funkci pro výstup neuronu.

$$f(x) = \frac{1}{1+e^{-x}} \qquad (15)$$

- **ReLU** (Rectified linear unit) [31]: Tato funkce se často používá jako aktivační funkce pro skryté vrstvy neuronové sítě. Funguje tak, že pro vstupní hodnoty větší než nula vrací hodnotu vstupu, zatímco pro záporné hodnoty vrací nulu.

$$f(x) = \max(0, x) \qquad (16)$$

- **Tanh** (hyperbolic tangent): Funkce, která se používá pro normalizaci vstupních dat do rozmezí (-1, 1).

$$f(x) = \tanh(x) \qquad (17)$$

- **Softmax** [32]: Používá se pro klasifikaci více tříd. Vrací hodnoty v rozmezí od 0 do 1, které se součtem rovnají 1, díky tomu lze jejich výstup použít pro odhad pravděpodobnostního rozdělení.

$$f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \qquad (18)$$

Ztrátové funkce

Ztrátová funkce (anglicky *loss function*) je matematická funkce, která měří, jak moc se výstup modelu liší od očekávaného výstupu. Používají se při trénování neuronové sítě, kdy se jejich hodnota minimalizuje pomocí optimalizačního algoritmu. Nejčastěji používané ztrátové funkce jsou:

- MSE (mean squared error) – průměrná kvadratická chyba, sečte všechny kvadráty rozdílů mezi skutečnou hodnotou a predikovanou hodnotou. Používá se v regresivních úlohách.

$$L(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (19)$$

- Categorical Cross-Entropy [33] – používá se pro klasifikaci do více tříd, například pokud máme tři třídy A, B, C, vypočítá entropii kategorií.

$$L(y) = -\sum_{i=1}^{N}\hat{y}_i \log\,(y_i) \quad (20)$$

- Hinge Loss – používá se pro lineární klasifikaci, například pro rozlišení mezi pozitivními a negativními příklady.

$$L(y) = \max\,(0, 1 - \hat{y}_i y_i) \quad (21)$$

Embedding vektor

Embedding vektor [34] je vektorová reprezentace slova nebo jiného symbolu. Tyto vektory se používají v jazykových modelech, kde slouží k reprezentaci slov nebo symbolů v numerické podobě, která umožňuje modelu pracovat s těmito daty. Obecně se používají pro reprezentaci kategorických vstupů do neuronové sítě. Tyto vektory mohou být buď předem vypočítané pomocí jiného (již natrénovaného) modelu, anebo se mohou trénovat společně s ostatními parametry.

Dense vrstva

Dense (fully connected) vrstva je nejzákladnější druh vrstvy [35] v neuronové síti, kde jsou všechny neurony v této vrstvě připojeny ke všem neuronům v předchozí a následující vrstvě. To znamená, že každý neuron v této vrstvě má váhu pro každý vstup z předchozí vrstvy a také bias.

Tento typ vrstvy se často používá jako skrytá vrstva nebo také jako vrstva pro projekci do jiné dimenze (např. na výstup). Jedná se o základní stavební blok v architektuře neuronové sítě.

Kapitola 4

WaveNet

WaveNet je neuronová síť pro generování syntetické řeči s vysokou kvalitou. Architektura byla představena v [36], kde v poslechových testech kvalitou překonala tradiční signálové a parametrické metody.

WaveNet je velmi mocná hluboká neuronová sít se schopností generovat přímo řečový signál vzorek po vzorku. Tato vlastnost je velmi unikátní, neboť běžné parametrické metody nepracují nad signálem samotným, ale reprezentují řeč pomocí *framů* akustických příznaků, které pak pomocí *vokodéru* převádějí na signál. To s sebou nese mírnou ztrátu kvality.

WaveNet modeluje podmíněnou pravděpodobnost každého vzorku v řečovém signálu. Podmíněnost je určena předchozími hodnotami generovaného signálu. Předchozí výstup sítě je tak zároveň jejím vstupem v dalším kroku. Tento model je proto autoregresivní.

4.1 Základní architektura

Řečový signál můžeme chápat jako posloupnost vzorků

$$x = x_1, x_2, \dots, x_T. \tag{22}$$

Ideální model, na kterém staví WaveNet, popisuje pravděpodobnost vzorku x_t pomocí všech předchozích hodnot:

$$p(x) = \prod_{t=1}^{T} p\left(x_t | x_1, \dots x_{t-1}\right). \tag{23}$$

V praxi je však tento model příliš komplikovaný. Proto se používá zjednodušený model, kdy je každý vzorek podmíněn fixním počtem předchozích vzorků (zahrnujících kolem 300 ms). Všimněme si, že WaveNet neodhaduje samotnou hodnotu následujícího signálu, ale pouze její pravděpodobnostní rozdělení (v základní verzi je to histogram). Hodnotu vzorku lze pak vybrat například náhodným vzorkováním. Příští vzorek je však podmíněn hodnotou vzorku, nikoliv jejím rozdělením. Tento detail je velmi důležitý (viz kapitola 4.2).

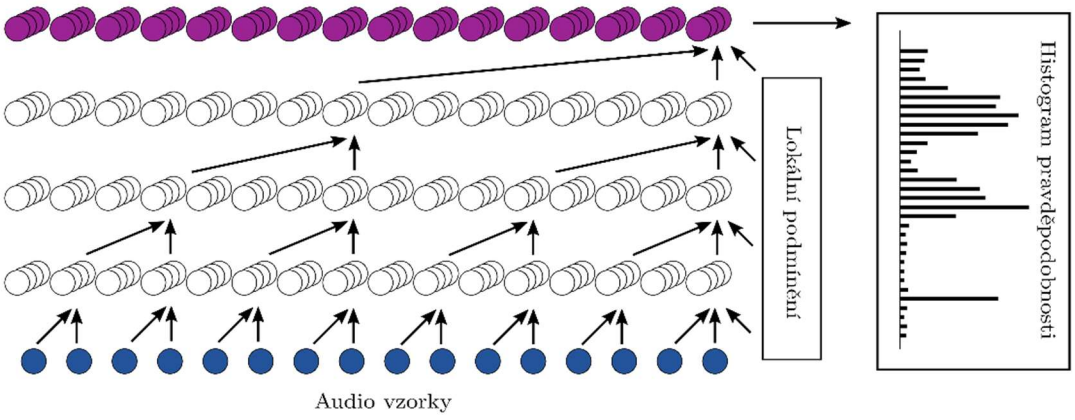
Pravděpodobnostní rozdělení $p(x)$ popisuje obecný řečový signál. Pro to, aby model generoval námi požadovanou řeč, je nutné model dále podmínit charakteristikami řeči h:

$$p(x) = \prod_{t=1}^{T} p\left(x_t | x_1, \dots x_{t-1}, h\right). \qquad (24)$$

Podmíněnost pomocí řečových charakteristik může být globální a lokální. Globální podmíněnost je konstantní pro všechny vzorky, například pokud je model trénován z více řečníků, může být globální podmíněnost aktuální identita řečníka. Lokální podmíněnost se mění v čase. Může se jednat například o lingvistické parametry (identita aktuálního fonému), prozodické parametry (hlasivková frekvence) nebo akustické parametry (spektrální obálka).

Modelovat výše definovanou podmíněnou pravděpodobnost je velmi složité. Proto je vhodné použít dostatečně komplexní model jako například hluboké neuronové sítě, které mají dostatečnou kapacitu pro takto složitý problém. Architektura WaveNet je postavená na bázi velmi hluboké konvoluční sítě. Je zde však jeden detail, kterým se odlišuje.

Autoři zvolili *dilatované konvoluce* s exponenciálně zvětšujícím se okénkem. Dilatovaná konvoluce (obrázek 24) řeší problém s omezeným dosahem vidění (*receptive field*) standardních konvolučních vrstev. Pokud budeme uvažovat ořezanou historii 300 ms, vychází to při vzorkovací frekvenci 16 kHz na 4800 vzorků, které musí vrstvy pokrýt. To by znamenalo nutnost použití stovek až tisíců vrstev. To není dost dobře proveditelné. *Dilatované konvoluce* však dokážou pokrýt takto velký počet vzorků pomocí několika desítek vrstev.



Obrázek 24: Architektura sítě WaveNet. Na obrázku je konfigurace dilatací 1, 2, 4, 8. První vrstva je dole. Dilatace 1 je vlastně tradiční konvoluce (dva vzorky vedle sebe), dilatace 2 znamená přeskočení ob jeden vzorek, 4 je každý čtvrtý atd. Reálná síť obsahuje desítky vrstev.

Vstupem první vrstvy jsou dva po sobě jdoucí vzorky. Vstupem druhé vrstvy je výstup předchozí vrstvy v čase t_n a t_{n-2}, vstupem třetí vrstvy jsou výstupy druhé vrstvy v čase t_n a t_{n-4} a tak dále. Originální architektura WaveNetu obsahuje 30 vrstev dilatovaných konvolucí s následující konfigurací:

1, 2, 4, 8, 16, …, 256, 512, 1, 2, 4, 8, 16, …, 256, 512, 1, 2, 4, 8, 16, …, 256, 512.

Obsahuje tři skupiny deseti vrstev, kde velikost dilatace je mocnina dvou a začíná od jedné. Autoři experimentálně nastavili tuto konfiguraci, aby dosáhli požadovaného dosahu vidění (*receptive field*) a zároveň měli dostatečný počet vrstev.

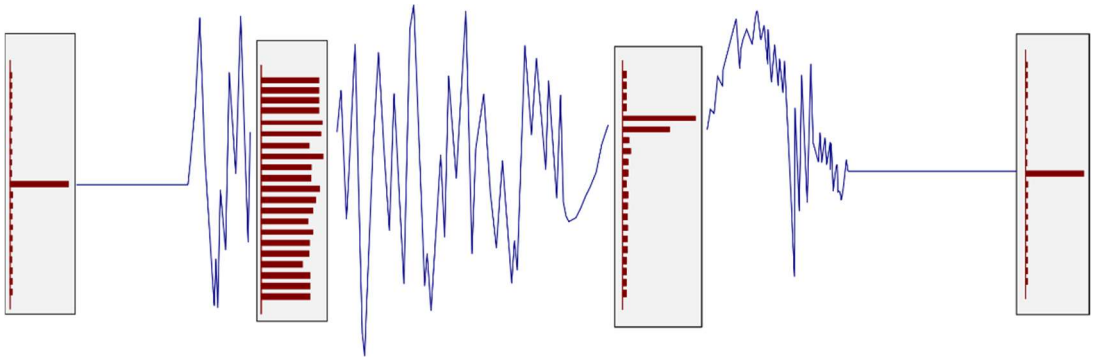
4.2 Generativní model

Výstupem WaveNetu není hodnota následujícího vzorku, nýbrž pouze jeho pravděpodobnostní rozdělení (ve formě histogramu, viz obrázek 25). Ač to může vypadat jako malý detail, ve skutečnosti je to jedna z nejdůležitějších myšlenek nové architektury a pravděpodobně také důvod, proč tato síť funguje. Stávající parametrické metody totiž nejsou schopné generovat přímo řečových signál. Musí proto predikovat řeč pomocí spektrálních charakteristik.

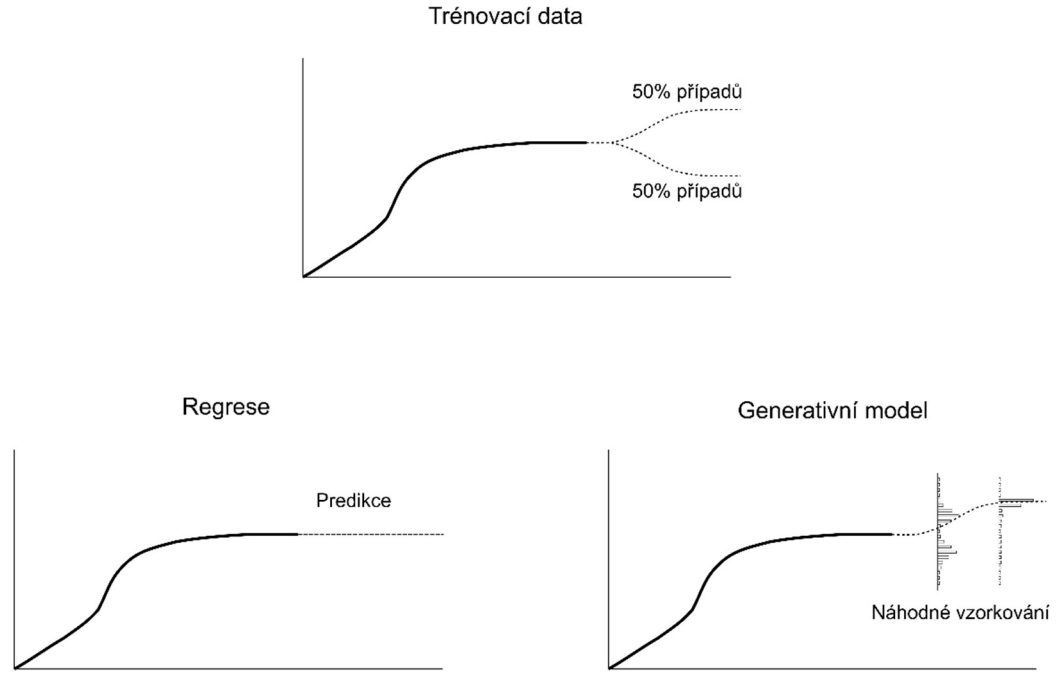
Průběh řečového signálu je velmi složitý proces, ve kterém se odehrává mnoho jevů současně. Každý jev probíhá zároveň v jiném přiblížení. Některé jevy se odehrávají mezi jednotlivými vzorky, některé jevy můžou probíhat na úseku několika stovek milisekund, což mohou být tisíce vzorků. Některé jevy, jako například šumy v sykavkách, jsou čistě stochastické.

Stávající parametrické modely (ať už jde o HMM, nebo novější neuronové sítě založené na LSTM) provádějí regresi, tj. predikují hodnoty. Na stejném základě vznikaly i první pokusy generovat řečový signál, avšak ty byly neúspěšné [37]. Autoři WaveNetu proto přeformulovali problém a použili místo toho generativní přístup. Ten skloubili s autoregresivním přístupem. Přeformulování problému generovaní řeči z regrese na generativní model byl geniální tah, který otevřel cestu k novým přístupům k syntéze řeči.

Jak již bylo zmíněno, WaveNet predikuje histogram pravděpodobnostního rozdělení. Z tohoto rozdělení lze pak hodnotu následujícího vzorku vybrat náhodným vzorkováním. Tento způsob přidává modelu cestu, jak generovat náhodné veličiny. Vzorkování je tak přímý kanál náhody, který model může nebo nemusí využít v závislosti na situaci. Pokud bude chtít model generovat náhodný šum, stačí aby histogram obsahoval ve všech kategoriích stejné hodnoty. Pokud naopak šum použít nechce, může histogram být nenulový pouze v jedné kategorii, která pak bude ve většině případů vybrána.



Obrázek 25: Příklad výstupu generativního modelu v čase.



Obrázek 26: Porovnání tradiční regrese oproti generativnímu modelu pro situace, kde trénovací data obsahují velmi odlišné hodnoty.

Uvažujme úlohu, na které lze předvést sílu generativního modelu. Řekněme, že se budeme snažit modelovat průběh hlasivkové frekvence věty na základě sekvence fonémů, které jsou v dané větě obsaženy. Můžeme natrénovat regresivní model, který bude fungovat obstojně. Co když ale pro stejné textové příznaky je více průběhů realizací, kde každý je úplně jiný? Můžeme si představit, že například řečník v otázkách náhodně intonuje hlasem nahoru, nebo dolů. Regresivní model nemá žádnou jinou možnost než zprůměrovat trénovací data. Predikce tudíž bude neutrální. Zde vzniká paradox. Řečník vždy intonoval buď nahoru, nebo dolů. Regresivní model však predikuje konstantní hodnotu, která se v datech nevyskytuje.

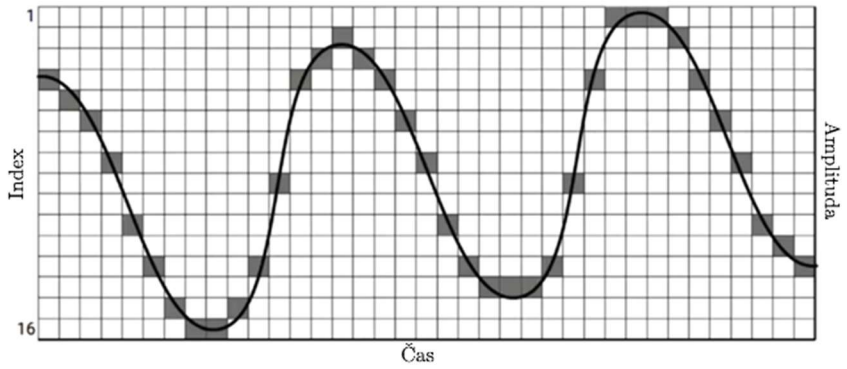
Generativní model může tento jev modelovat mnohem lépe (viz obrázek 26). V místě rozdvojení predikuje histogram, ze kterého se náhodným vzorkování vybere buď jedna cesta, nebo druhá. Následující hodnoty již budou sledovat onu trajektorii.

4.3 Diskretizace signálu

Síť WaveNet predikuje výstupní signál v kvantizované podobě (obrázek 27) jako histogram pravděpodobností jednotlivých úrovní signálu. Ve výchozím nastavení je kategorií 256, tj. 8 bitů přesnosti.

Poslední vrstva modelu je *softmax* aktivační funkce, jejíž výstup lze interpretovat jako pravděpodobnostní rozdělení pro jednotlivé úrovně signálu. Tím, že je úroveň signálu kategorizována, je možné pohlížet na úlohu jako na *klasifikaci*. Predikce signálu, kde výstupem sítě je přímo hodnota, je úloha *regrese*.

Tím, že je na úrovně přihlíženo jako na kategorie, je ztracena informace o jejich uspořádání. Neuronová síť tak proto netuší, že např. úroveň 0,56 je blízko úrovni 0,55, ale naopak velmi vzdálená úrovni -0,77. Tato zdánlivá nevýhoda umožňuje síti si vytvořit vlastní představu o tom, co která úroveň znamená.



Obrázek 27: Kvantizace audio signálu.

Nutno podotknout, že alternativou diskretizace signálu je predikce směsi gausovských rozděleních (gaussian mixture model), které rovněž dokáží popsat pravděpodobnostní rozdělení následujícího vzorku s dostatečnou přesností. Predikce diskretizovaného signálu je ale výchozí nastavení.

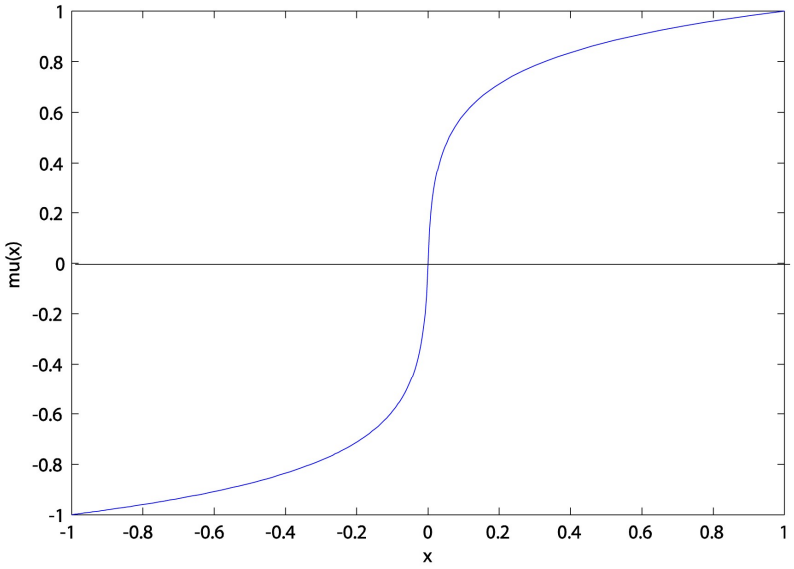
4.4 Mu-law kompander

Kvantizace do 8 bitů (256 úrovní) zanáší do výsledného zvuku kvantizační šum, který je slyšitelný. Obvykle se totiž signál kvantizuje do 16 bitů (65536 úrovní).

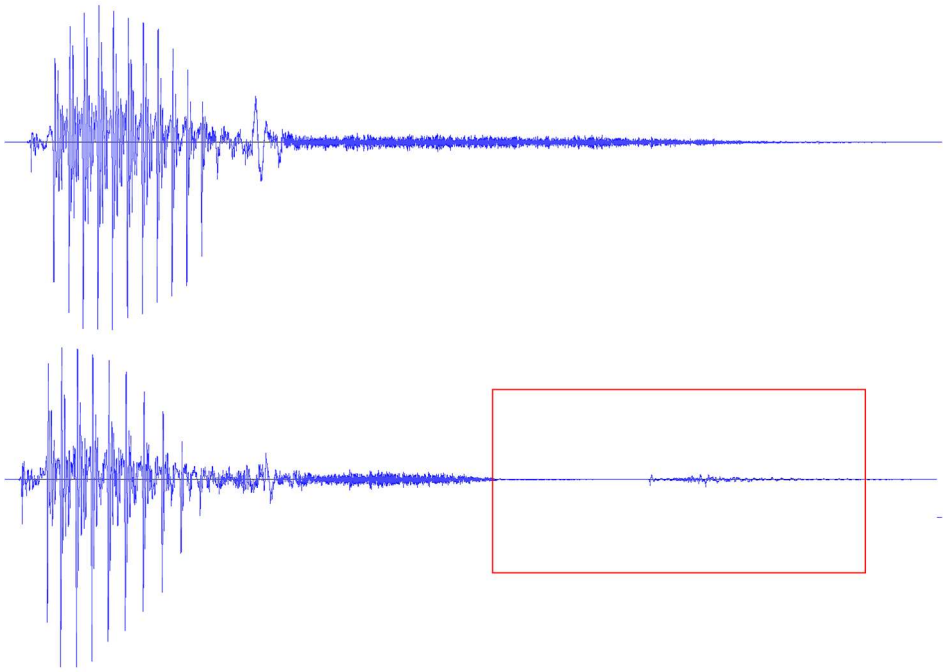
Pro minimalizaci kvantizačního šumu je signál nejdřív transformován pomocí mu-law (μ-*law*) kompanderu [38] následující rovnicí

$$F(x) = sgn(x)\frac{\ln(1+256|x|)}{\ln(1+256)}, \qquad (\ 25\ )$$

kterou vykresluje obrázek 28.



Obrázek 28: Mu-law kompander. Pro hodnoty blízko nuly je alokován mnohem větší rozsah na výstupu. To vede k zvětšení přesnosti pro hodnoty blízko nuly na úkor rozsahu blízko hodnot 1 a -1.



Obrázek 29: Porovnání signálu slova „pas“ a „past“.

Tento velmi starý algoritmus byl využíván v telekomunikačních technologiích, protože se dal jednoduše sestrojit pomocí elektronického obvodu. V síti WaveNet je však využit z jiných důvodů, využívá totiž vlastnosti lidkého ucha, které je citlivější na nižší úrovně signálu.

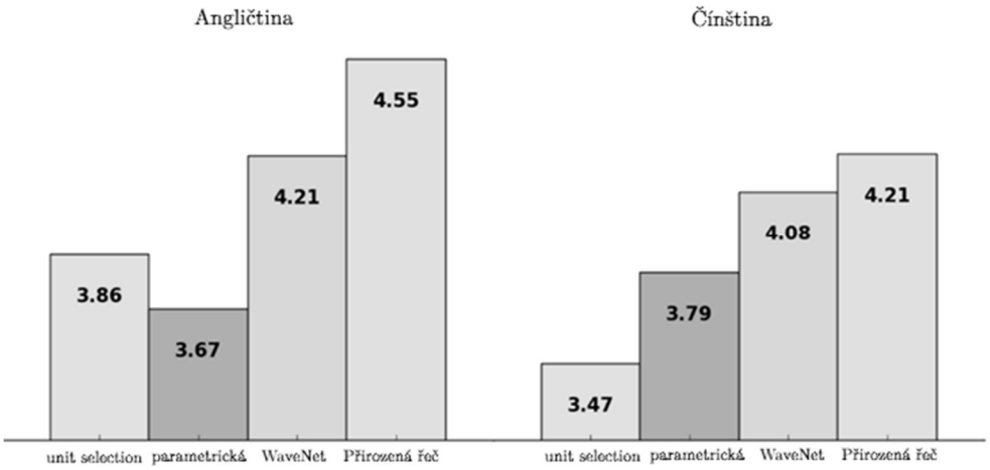
Obrázek 29 porovnává dvě velmi podobná slova. Z obrázku je patrné, že rozdíl těchto slov není v hlasitých úsecích, ale určuje ho velmi tichá část signálu. To je jedním z důvodů, proč se lidský mozek a ucho vyvíjely evolucí tak, aby byly mnohem citlivější na velmi tiché ruchy. V těchto úrovních intenzity je tak výhodné co nejvíce snížit kvantizační šum (zvýšit rozlišení a přesnost). Naopak, čím je hlasitější zvuk, tím je citlivost ucha nižší a stačí menší přesnost.

4.5 Publikované výsledky

Autoři článku [36] provedli poslechový test. WaveNet byl první model, který dokázal porazit jak unit selection pro angličtinu, tak statistickou parametrickou metodu pro čínštinu (pro kterou unit selection nedosahuje tak dobrých výsledků). Výsledky testu ukazuje obrázek 30.

4.6 Optimalizace trénování

Síť WaveNet používá mnoho důmyslných triků. Některé již byly zmíněny (dilatovaná konvoluce, kvantizace signálu, mu-law). Pro urychlení konvergence využívá síť několik dalších triků, které se v posledních letech ukázaly jako účinné. Tyto triky byly většinou odhaleny experimentálně. První výskyt je většinou u neuronových sítí pro zpracování obrazu, neboť na tuto vědeckou oblast je tradičně upřena velká pozornost. Jinak tomu není ani v tomto případě. Síť WaveNet je ve skutečnosti velmi inspirovaná sítí PixelCNN [39].



Obrázek 30: Výsledky poslechového testu z [23].

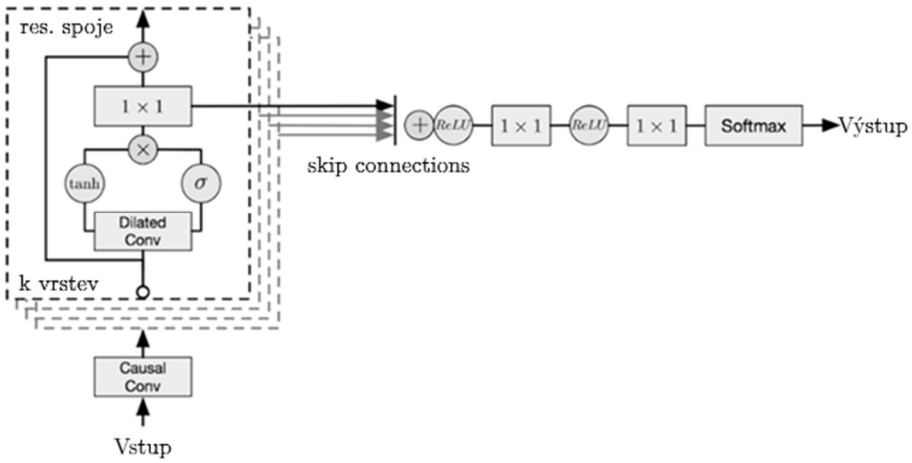
Jedním z dalších triků jsou tzv. residuální spoje [40]. Princip je jednoduchý. Tradičně je výstup vrstvy vstupem vrstvy následující. U residuálních spojení je výstup vrstvy součet jejího vstupu a výstupu (residuum). Residuální spoje urychlují konvergenci při trénování. Spojení popisuje rovnice:

$$y = x + f(x). \tag{26}$$

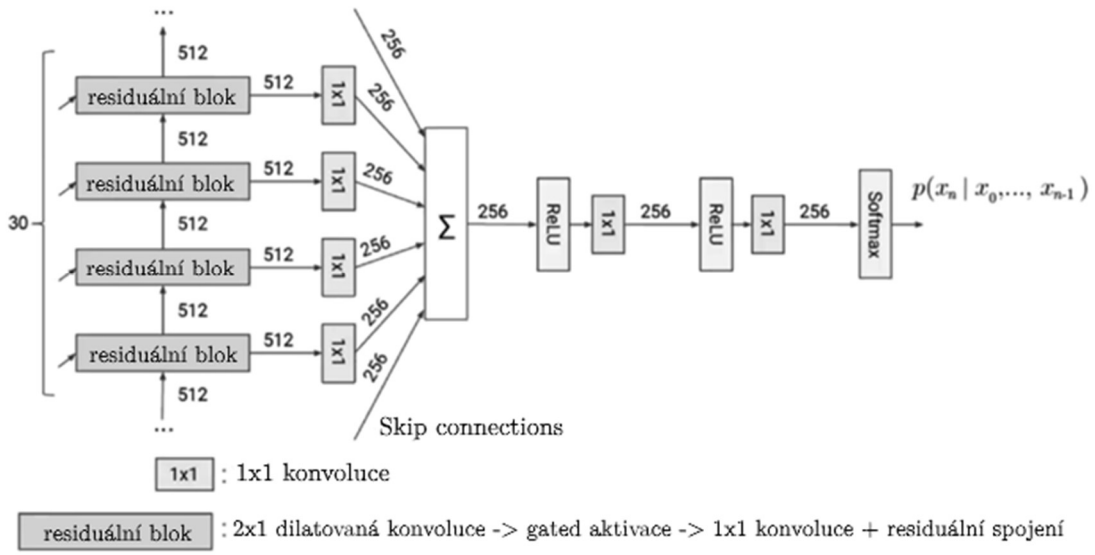
Dalším trikem je použití propracovanější aktivační vrstvy tzv. *gated activation*. Ta se v praxi ukázala jako mocnější, než je tradiční aktivace pomocí jedné funkce. Díky tomu je možné použít méně vrstev při zachování stejné komplexnosti sítě. *Gated activation* (obrázek 31) používá součin dvou aktivačních funkcí. Výstup se rozdělí na dvě poloviny: filtr a hradlo. Hradlo používá *sigmoid* aktivaci, která mapuje na rozsah (0, 1). Na filtr je použita aktivace *tanh*, která mapuje na rozsah (-1, 1). Celkový výstup má rovněž rozsah (-1, 1). Aktivace je provedena pomocí vzorce:

$$y = \tanh(W_f x)\,\mathrm{sigmoid}(W_g x). \tag{27}$$

Posledním zmíněným trikem jsou tzv. *skip connections*. Tradičně při použití mnoha vrstev je výstupem modelu výstup poslední vrstvy. Pro hluboké sítě ale vzniká problém, že cesta gradientu k prvním vrstvám je příliš dlouhá. Při použití *skip connections* je výstupem modelu součet výstupů všech vrstev (viz obrázek 32). Cesta gradientu je potom ke každé vrstvě velmi krátká.



Obrázek 31: Architektura sítě WaveNet – reziduální blok.



Obrázek 32: Architektura sítě WaveNet.

Kapitola 5

WaveRNN a neurální vokodér

V roce 2018, tedy dva roky po představení neuronové sítě WaveNet, byla představena architektura WaveRNN, která přinesla několik výhod. WaveRNN [41] je architektura neuronové sítě velmi podobná síti WaveNet. Jedná se o autoregresivní model schopný generovat řečový signál po vzorcích. Namísto konvolučních vrstev používá WaveRNN jednu obří (512–1024 buněk, což odpovídá cca 4 miliónům parametrů) rekurentní vrstvu.

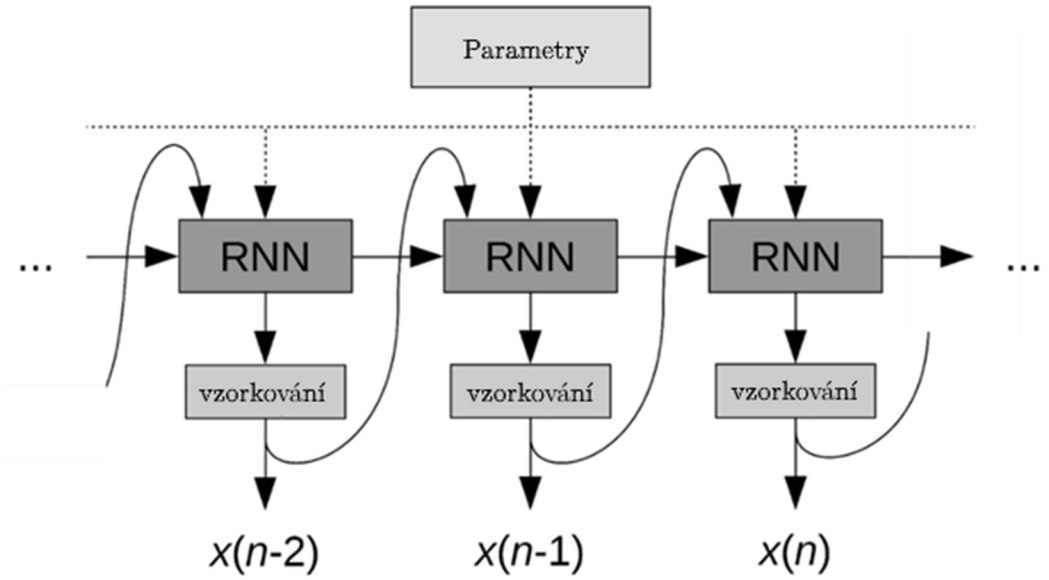
5.1 Architektura

Jádro WaveRNN tvoří jedna velká GRU (*gated recurrent unit*) vrstva. Na jejím vstupu je hodnota předchozího vzorku a podmiňující vektory pro daný 5ms časový úsek (*frame)*. Architekturu zobrazuje obrázek 33. Díky použití rekurentní vrstvy je mnohem jednodušší než architektura WaveNet.

Výhodou GRU vrstvy (viz kapitola 3.3) oproti LSTM je menší počet parametrů a také fakt, že její výstup je zároveň jejím stavem. To je rozdíl oproti LSTM, která má vlastní vnitřní stav, oddělený od výstupního stavu. Při vyhodnocení LSTM je výstup reprezentován dvojící vektorů (h, c). V GRU je vše obsaženo v jednom vektoru h. Tento fakt zjednodušuje generování ve WaveRNN a umožňuje efektivněji používat paměť.

5.1.1 Vstup sítě

Podmiňujícími vektory mohou být akustické nebo lingvistické příznaky. Příznaky je možné i kombinovat. Většinou jsou tyto vektory předzpracovány několika plně propojenými (*fully connected,* viz kapitola 3.3.1) vrstvami. Při trénování je jako hodnota předchozího vzorku použita skutečná hodnota (tzv. *teacher-forced* mód). Po GRU obsahuje síť jednu plně propojenou projekční vrstvu s ReLU aktivací a jednu projekční vrstvu se *softmax* aktivací.



Obrázek 33: Architektura WaveRNN.

5.1.2 Výstup sítě – Dual softmax

Audio signál je obvykle kvantizován do 16 bitů (2^{16} hodnot). Pomocí mu-law kompandingu lze použít i 8bitovou kvantizaci, tj. 256 hodnot. To je důležité pro kategorickou predikci WaveNetu a WaveRNN. Nevýhodou však je, že výstupní signál trpí kvantizačním šumem, což snižuje kvalitu výsledné řeči.

Jedním z možných řešení je místo kategorické klasifikace predikovat hodnotu audio signálu například pomocí Gaussovské směsi. Hustota pravděpodobnosti je v takovém případě spojitá a model netrpí kvantizačním šumem. Natrénování sítě je ale mnohem složitější a doprovází ho celá řada problémů. Trénovací proces je mnohem citlivější na nastavení hyperparametrů.

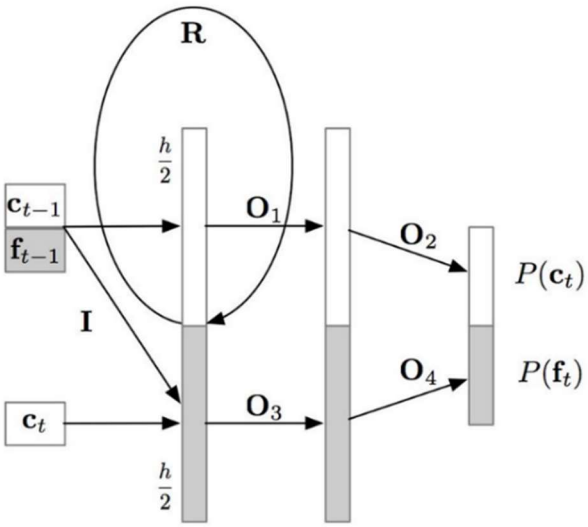
Autoři WaveRNN chtěli zachovat kategorický přístup k predikci výstupu. Zvolili proto tzv. *dual-softmax*. V tomto režimu predikuje neuronová síť dvě rozdělení. Audio signál je ponechán v 16bitovém formátu a je rozdělen na horních 8 bitů (*coarse-bits*) a spodních 8 bitů (*fine-bits*). První výstup neuronové sítě predikuje horních 8 bitů. Druhý pak predikuje spodních 8 bitů, avšak s již známou hodnotou horních 8 bitů. To je zajištěno pomocí maskování a odmaskování části výstupní matice (viz obrázek 34).

Neuronová sít pomocí techniky dual-softmax dokáže predikovat audio signál bez kvantizačního šumu v plné bitové hloubce. Výstupní vektor má dimenzi 2x256.

5.1.3 Trénování

Použití rekurentní sítě přináší mnohem pomalejší trénování sítě. WaveNet je postaven z konvolučních vrstev, které lze při trénování snadno paralelizovat. Rekurentní vrstva

GRU obsahuje stav a její průchod je vyhodnocován sekvenčně. Při trénování je proto nutné používat kratší sekvence a spíše zvětšovat počet reprezentací v dávce (batch size).



Obrázek 34: Dual softmax. V prvním průchodu (bílá část) je vypočteno rozdělení pravděpodobnosti $P(c_t)$, z ní je vzorkováním vybráno prvních 8 bitů. Při druhém průchodu (šedivá část) se počítá $P(f_t)$ a přitom se už používá prvních 8 bitů (f_{t-1}) jako vstup.

5.2 Výhody

5.2.1 Rychlost generování

Co se týče počtu matematických operací (FLOPS) pro generování, tak model WaveRNN má stejnou náročnost jako model WaveNet. Výpočet je ale realizován pouze pomocí násobení několika velkých matic. WaveNet na druhou stranu obsahuje mnoho menších vrstev, které je nutné vyhodnotit postupně za sebou. To s sebou přináší režii, která například na grafických akcelerátorech může způsobit znatelný propad výkonu.

Architektura WaveRNN je tak lépe uzpůsobená pro efektivní generování a dokáže generovat řeč mnohem rychleji než WaveNet, zvláště pak na grafickém procesoru. V [41] zmiňují autoři generování v rychlejším čase, než je trvání řeči.

Pomalostí WaveNetu se zabývá navazující práce [42], problém řeší pomocí upravené architektury *Paralelní WaveNet*. Jeho složitost a komplikovanost však je značná. Použitím WaveRNN je problém vyřešen mnohem jednodušeji [41].

5.2.2 Kvalita

Dle [41] dosahuje kvalita syntetické řeči stejných kvalit jako síť WaveNet (tj. velmi vysokou). To je očekávané, neboť tyto dvě sítě sdílí stejnou myšlenku autoregresivního

generování a jak konvoluční vrstvy, tak rekurentní vrstvy jsou dost mocné na to, aby daný problém vyřešily.

Pro generování 16bitového signálu je možné natrénovat WaveNet, který predikuje místo softmax histogramu logistickou směs. To s sebou přináší obtížnější trénování. WaveRNN tento problém řeší elegantněji použitím dual softmaxu, je tak co se týče kvality na stejné úrovni.

5.3 Neurální vokodér

Režim, ve kterém jsou jako podmínění sítě WaveNet nebo WaveRNN použity akustické příznaky, se označuje jako *neurální vokodér*. Tento název vychází z analogie s tradičními vokodéry, což jsou algoritmy pro převod akustických parametrů do řečového signálu. Tradiční vokodéry používají signálové algoritmy. Neuronové sítě se namísto toho učí generovat signál z trénovacích dat bez expertního matematického algoritmu.

Pro připomenutí, ve výchozím stavu se obvykle používají lingvistické příznaky extrahované z textu. Výchozí stav se dá považovat za end-to-end systém. Výchozí režim byl použit v kapitole 7 a 8, v této kapitole je WaveRNN použito jako neurální vokodér. Kapitola 12 se věnuje více architekturám plnícím funkci neurálního vokodéru a konceptu end-to-end.

Neurální vokodér se používá v kombinaci s jiným modelem, který akustické příznaky generuje. Výchozí stav je tak rozdělen na dva modely, kde každý řeší pouze svou dobře specifikovanou část úlohy. To usnadňuje rychlost trénování, protože se vždy trénuje jen jeden model. Rovněž to umožňuje kombinovat různé architektury pro generování akustických parametrů a pro neurální vokodér.

Pro generování akustických příznaků lze použít například model parametrické syntézy LSTM. V takovém případě jsou jako akustické příznaky brány výstupní spektrální obálky a F_0. Jiné modely (např. *Tacotron* [35]) jsou schopné generovat melovské spektrogramy (viz kapitola 2.6), které také mohou sloužit jako akustické příznaky.

Úkolem neurálního vokodéru je převést akustické příznaky do audio signálu. Jejich funkce je tedy stejná jako u tradičních vokodérů (např. WORLD [14] nebo STRAIGHT [22]). Signálové algoritmy jsou nahrazeny neuronovou sítí natrénovanou z reálných dat.

5.3.1 Režim trénování

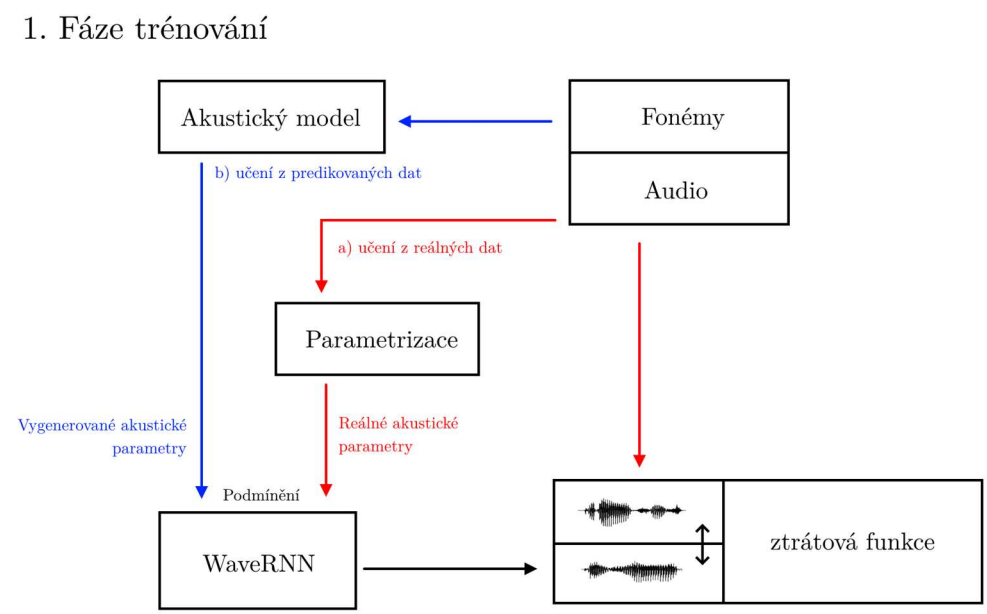
Pro trénování neurálního vokodéru je možné použít dva přístupy (viz obrázek 35). Lze použít buď reálné akustické parametry, nebo parametry vygenerované akustickým modelem. V prvním případě jsou podmíněním sítě akustické parametry ze stejného audio signálu, na kterém se trénuje (učení z reálných dat). V druhém případě je podmíněním sítě výstup z natrénovaného akustického modelu, který tu samou větu vygeneroval z jejího textu (učení z predikovaných dat).

Oba přístupy mají výhody i nevýhody. Při trénování z reálných dat odpovídají vstupní parametry lépe požadovanému výstupnímu signálu. Učení je proto jednodušší, neboť si vstupní a výstupní data odpovídají. Problém je, že ve fázi syntézy (generování) je síť vždy nucena použít predikovaná data z akustického modelu (reálná nejsou k dispozici). Vokodér tak ve fázi syntézy dostává jiná vstupní data, než na která byl naučen. Data vygenerovaná akustickým modelem jsou sice ve stejném rozsahu, ale nemusí být tak variabilní a bohatá na informace, jako byla skutečná data ve fázi trénování. Přestože se akustický model „snaží“ generovat data co nejvíc podobná reálným datům, určitý rozdíl zde bude vždycky. Jelikož si při trénování vstupní a výstupní data odpovídala, úloha, kterou se učil neurální vokodér, byla vlastně jen replikace toho, co dříve dělaly tradiční vokodéry – převést akustickou reprezentaci do audio signálu. Síť byla trénována tak, že neměla motivaci přidávat novou informaci do audio signálu či jinak audio signál „vylepšovat“. Ve fázi generování se může stát, že neurální vokodér bude generovat málo kvalitní řečový signál, který bude znít příliš „vyhlazeně“. Mohou se v něm vyskytovat řečové artefakty a šumění.

V druhém případě, tj. trénování z predikovaných dat, není problém v odlišnosti vstupních parametrů – jak ve fázi trénování, tak ve fázi generování jsou vstupem parametry generované stejným akustickým modelem. Vokodér je tak přesně naučený na to, co bude ve fázi generování jeho vstupem. Problém je v tom, že vstupní příznaky úplně neodpovídají charakteristice trénovacího řečového signálu. Dochází zde k nekonzistenci mezi vstupem a výstupem (řečový signál může mít jinou trajektorii F_0, energii, frekvenční spektrum je vyhlazené atd.). Trénovací řečový signál je mnohem bohatší, než by měl podle vstupního podmínění být. Neuronová síť tak plní složitější úlohu, kdy musí „domýšlet“ nové informace a suplovat tak funkci akustického modelu. Vstupní podmínění je tak pouze vodicí informace a realizace je tedy na samotné neuronové síti. V extrémních případech se může stát, že se síť naučí v některých případech ignorovat vstupní podmínění a vygeneruje jiná slova, než která vygeneroval akustický model.

Jak je patrné, ani jedna metoda není ideální. Při trénování je možné střídat oba přístupy. Například lze použít strategii, že úvodní trénování je provedeno na reálných datech a pro doladění (*fine tuning*) je použito trénování z predikovaných dat. Alternativní strategií je náhodně střídat vstupní data z obou variant a nutit tak síť, aby byla dostatečně robustní, aby zvládla oba typy vstupů (je možné použít vstupní příznak, zda jsou vstupní data reálná či generovaná).

V případě příznaku F_0 je vhodné použít vždy skutečné parametry, neboť trajektorie predikované F_0 se značně liší od trajektorie skutečné F_0 (nemá tak velkou variabilitu).



Obrázek 35: Režimy trénování neurální vokodéru. Červená linka značí první variantu – učení z reálných dat. Modrá pak učení z predikovaných dat.

Kapitola 6

SpeechLab – podpůrný nástroj pro vývoj syntézy řeči

Tento program byl navržen a naprogramován v rámci mého doktorského studia. Původně se jednalo o program pro manuální opravy anotací řečového inventáře [43]. Postupem času se z něj ale stal komplexní nástroj pro většinu činností souvisejících se syntézou řeči na katedře kybernetiky FAV ZČU[1].

V rámci různých projektů bylo postupně do tohoto webového nástroje naprogramováno mnoho funkcí. Pomocí něj bylo možné vytvořit a navrhnout nový systém TTS a zlepšit kvalitu trénovacích dat.

Samotná implementace nástroje sestává z webového frontendu a serverového backendu. Backend je naprogramován v jazyce `C++` a obsahuje potřebné algoritmy pro funkci nástroje SpeechLab a syntézy řeči. Obsahuje i vlastní implementaci pro inferenci neuronových sítí včetně architektury WaveNet a WaveRNN. Výhodou použití nativního jazyka je, že lze kód použít jako knihovnu a lze tak snadno zakomponovat do ostatních nástrojů a komerčních produktů.

6.1 Úprava anotací a segmentací

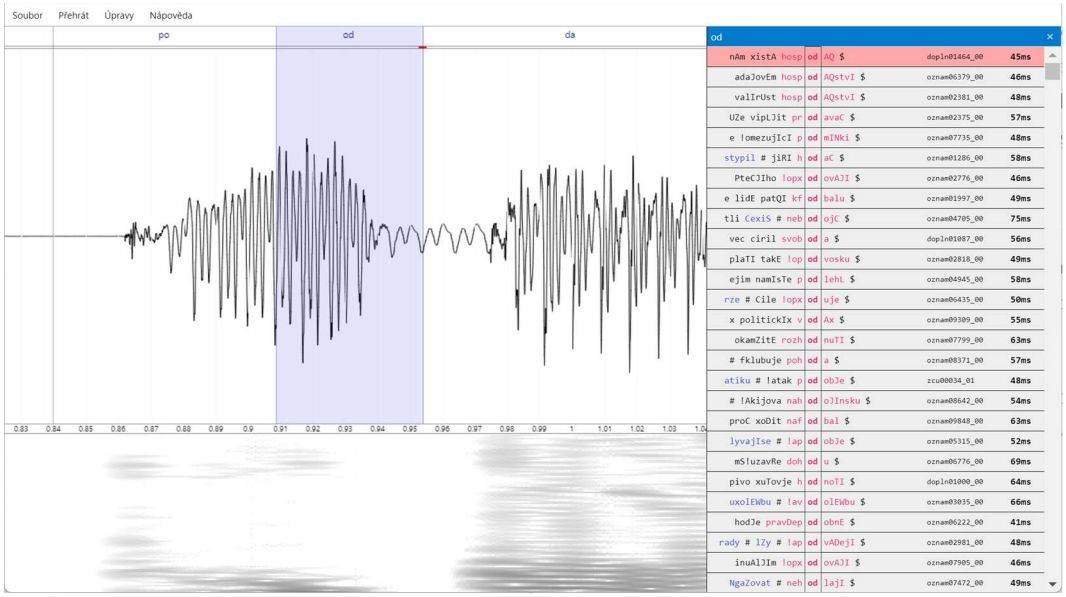
Nástroj SpeechLab obsahuje interaktivní editor, který umožňuje opravovat textové anotace a časové pozice jednotlivých fonémů přímo v audio signálu (viz obrázek 6 v kapitole 2). Uživatelé mohou provádět vizuální kontrolu trénovacích dat a opravovat případné chyby. To je důležité zejména pro systém TTS založený na algoritmu unit selection, který je náchylný k těmto chybám.

6.2 Interaktivní syntetizér unit selection

Tento speciální režim syntézy unit selection umožňuje trasování zdrojových jednotek zpět do řečového inventáře. Po vygenerování nahrávky lze v interaktivním editoru zobrazit pozice zřetězených jednotek. Po kliknutí na jednotku se lze snadno přenést do zdrojové promluvy z řečového inventáře. Lze tak snadno vyřešit případné chybné

[1] Webový program je dostupný na adrese https://speechlab.zcu.cz

anotace či segmentace. Editor zároveň zobrazuje ceny cíle a řetězení – to je užitečné při ladění chyb v samotném algoritmu unit selection. Uživatel si může zároveň zobrazit u každé jednotky její alternativy v grafu Viterbiho algoritmu (viz obrázek 36). Tím lze vygenerovat různé varianty stejné promluvy a opravit řečové artefakty ve vygenerované řeči.



Obrázek 36: Alternativy jednotek v algoritmu unit selection.

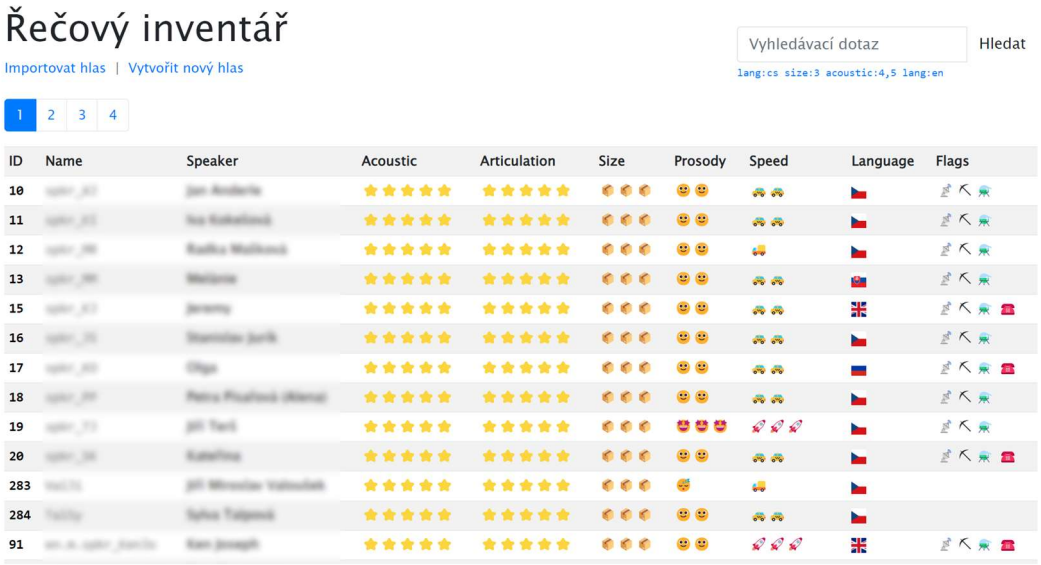
6.3 Automatická segmentace řečových nahrávek

Program obsahuje modul pro automatickou segmentaci řečového signálu pomocí LSTM neuronových sítí [44], která je uzpůsobená pro úlohu syntézy řeči. Tu je možné použít tam, kde není k dispozici ruční segmentace (například při nahrání nového hlasu). Tento model lze také použít pro prvotní nařezání velkých audio souborů (audioknih). Model pro segmentaci lze přetrénovat z ručně vytvořených segmentací. Manuální opravou segmentací u jednoho hlasu se tak zvyšuje přesnost segmentace na ostatních hlasech, které jsou segmentovány automaticky.

6.4 Správa a katalogizace řečové databáze

Tím, jak se nahrávají nová data a zvětšuje počet hlasů v řečovém inventáři, roste důležitost jejich správy. Postupem času se počet hlasů v hlasovém inventáři rozrostl na vyšší stovky. Nástroj SpeechLab umožňuje jednoduše procházet (viz obrázek 37) velké množství hlasů a přiřazovat jim různé atributy (například kvalita audio signálu, kvalita řečníka, počet vět, stav anotací atd.). Lze tak snadno vybrat hlasy, které odpovídají požadavkům experimentu. Po vybrání hlasů lze snadno vyexportovat potřebná audio a segmentační data.

Program disponuje systémem pro automatickou přípravu dat pro trénování neuronových sítí nebo pro použití při tvorbě hlasového balíčku unit selection. Tento proces obnáší řadů kroků: čištění audio nahrávek, změna vzorkovací frekvence, automatická segmentace [37], filtrování nahrávek a konverze do správného audio formátu. Tento proces je spuštěn automaticky při jakékoliv změně anotace nebo segmentace. To značně usnadňuje provádění experimentů a umožňuje udržovat aktuální verze všech modelů pro syntézu řeči.



Obrázek 37: Správa řečového inventáře. Jména řečníku byla rozmazána.

6.5 Nahrávání nového hlasu

Nástroj umožňuje nahrávat nové hlasy. Pomocí mikrofonu a webového prohlížeče může řečník buď doma, nebo ve zvukové komoře postupně nahrát věty (viz obrázek 38 a obrázek 39), ze kterých je poté možné vytvořit datový balíček pro syntetizér řeči. Lze zvolit různé jazyky a větné sady, které byly vytvořeny pro konkrétní typy nahrávání (řečník amatér nebo profesionální řečník).

Přestože se problém nahrávání zdá triviální, technická realizace obnáší mnoho problémů [45]. Je nutné zajistit, aby nahrávky byly dostatečně hlasité, ale zároveň aby hlasitost nepřetékala maximální hodnotu, kterou mikrofon dokáže zaznamenat. Dále je nutné zajistit dostatečnou délku pauzy na začátku a konci věty. Největším problémem je zajistit konzistenci dat. Občas totiž i na kvalitním hardwaru dochází k výpadkům řečových vzorků. Tento jev není častý (může se vyskytnout například jednou za hodinu), může však poškodit trénovací data a zanášet artefakty do trénovacích dat, což může vést k horší konvergenci při trénování.

Obrázek 38: Kalibrace mikrofonu před nahráváním nového hlasu.

Obrázek 39: Nahrávání nové věty.

6.6 Syntéza velkých dokumentů

Nástroj umožňuje syntézu velkých dokumentů se zpětným zvýrazněním špatně znormalizovaných slov v původním dokumentu. Původní dokument lze stáhnout zpět ve formátu *.rtf* a ve vlastním textovém editoru provést korekci před tím, než je dokument znovu vysyntetizován. Syntéza se provádí na pozadí kvůli dlouhému času potřebnému pro syntézu dlouhých textů, zejména při použití neuronových sítí. Je zde i nástroj pro interaktivní ladění pravidel pro normalizaci textu.

6.7 Neurální syntéza

Nástroj zároveň obsahuje samotný algoritmus a modely pro neurální syntézu pomocí sítě WaveNet a parametrické LSTM syntézy s neurálním vokodérem WaveRNN. Ty jsou jedním z výstupů této práce. Tyto modely jsou zakomponovány spolu s ostatními metodami do jednoho společného syntetizéru. Experimenty z dalších kapitol využívaly právě tento syntetizér pro syntézu řeči a porovnání s ostatními metodami.

Pro trénování modelů byl použit framework *TensorFlow*. Trénovací část kódu je naprogramovaná v jazyce Python. Kód pro fázi generování (inference) je zkomponován v samotném `C++` backendu aplikace.

Kapitola 7

Návrh systému syntézy řeči založeného na architektuře WaveNet

Cílem experimentu popisovaného v této kapitole bylo navrhnout, vyvinout[2] a otestovat syntézu české řeči založenou na neuronové architektuře WaveNet. Autoři [36] zmiňují vynikající výsledky na angličtině a čínštině. Syntézu českého jazyka v době provádění experimentu nikdo zatím nevyzkoušel.

Neuronová síť WaveNet v originální podobě pracuje v tzv. TTS režimu. To znamená, že vstupem sítě jsou lingvistické příznaky, které odpovídají vstupnímu textu, a celý proces syntézy řeči je pak součástí modelu sítě. I v tomto režimu jsou výstupem sítě přímo řečové vzorky audio signálu.

Alternativní přístup, který se dnes používá častěji, je použití WaveNetu jako vokodéru, kde vstupem jsou akustické příznaky odhadnuté jiným modelem. Nicméně jak již bylo zmíněno, původní článek popisuje WaveNet v TTS režimu. V rámci zachování co největší podobnosti byl i v experimentu použit TTS režim.

Nutno podotknout, že autoři [36] měli k dispozici data vynikající kvality a významné velikosti. Experiment v této kapitole může odpovědět na otázku, jak bude síť fungovat pro data standardní velikosti původně navržená pro tehdy dominantní *unit selection*.

Tato kapitola byla publikována v [46][3].

7.1 Detaily implementace

Pro experiment byla použita vlastní implementace. Síť obsahovala 20 vrstev dilatovaných konvolucí se stejným dilatačním vzorem jako uvádí autoři WaveNetu. Každá vrstva měla 128 spojů do vrstvy následující (residuální spoj) a 128 spojů v rámci optimalizace *skip connections*. Model dále obsahuje dvě ReLU postprocessing vrstvy a softmax aktivaci.

[2] V době psaní této práce nebyly veřejně dostupné žádné zdrojové kódy ani žádné další informace k síti WaveNet vyjma originálního článku. Vzhledem k tomu, že v článku nebyly dostupné všechny implementační detaily, vyžadoval tento experiment mnoho neúspěšných pokusů a iterací, než bylo možné úspěšně danou architekturu natrénovat a použít pro generování řeči.

[3] Článek je společné dílo více autorů. Má role byla v realizaci experimentu popsaného v této kapitole.

Generové audio vzorky byly kvantizované do 256 hodnot pomocí algoritmu mu-law. Pro konzistenci s ostatními metodami byla vzorkovací frekvence zachována na 16 kHz. Z toho vyplývá dosah vidění 120 ms předchozích vzorků.

Lokální podmínění odpovídá požadovaným charakteristikám produkované řeči a je aplikováno tak, aby vedlo síť WaveNet při generování řečového signálu. V experimentu byla použita následující lokální podmínění:

- Fonetická informace: identita aktuálního fónu a jeho levý a pravý kontext.
- Pozice audio vzorku v rámci fónu (zakódovaná pozičním vektorem, s experimentálně nastavenou dimenzí 100).
- Logaritmus hlasivkové frekvence (v neznělých segmentech je hodnota interpolována).
- Binární příznak pro znělost.

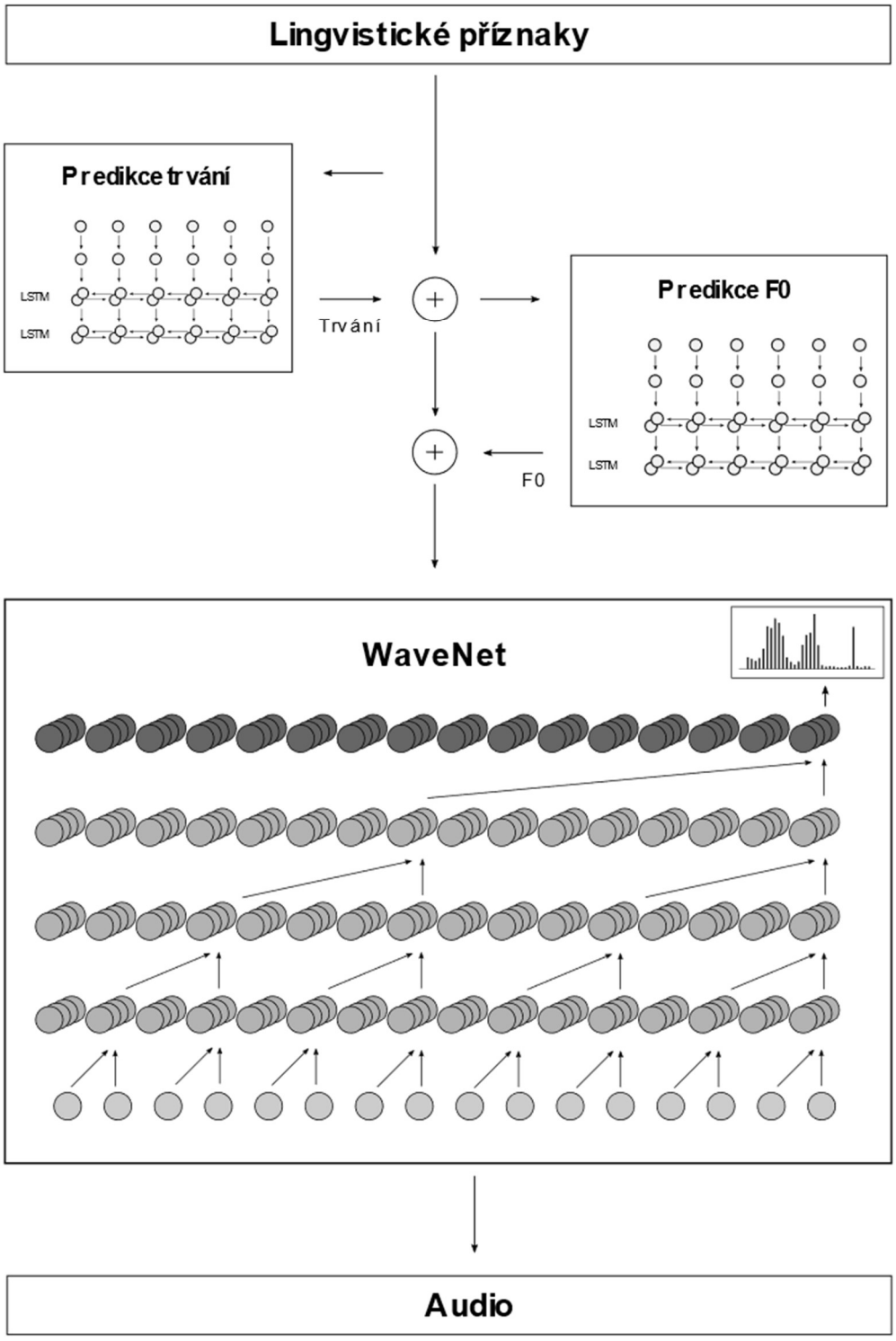
Pro generování řeči je třeba rozložit lingvistické příznaky v čase, k tomu je nutné znát trvání jednotlivých fonémů. K tomu byl použit model trvání založený na neuronové síti. Ta obsahovala dvě obousměrné LSTM vrstvy, každou s 64 neurony. Vstupem sítě jsou lingvistické příznaky jednotlivých fonémů a výstupem je jejich trvání. Délka vstupní a výstupní sekvence je počet fonémů ve větě, což je většinou několik desítek. Vyhodnocení sítě pro predikci trvání je velmi rychlé, stejně jako její trénování.

Pro predikci hlasivkové frekvence byl použit obdobný model – dvě obousměrné LSTM vrstvy, každá s 256 neurony. Vstupem byly opět lingvistické příznaky, ale byly rozloženy do 5ms úseků, dle odhadnutého trvání z předchozího modelu. Zároveň byl přidán nový příznak, a to pozice v rámci fonému. Ta byla zakódována do pozičního vektoru dimenze 4. Délka vstupní sekvence je počet úseků ve větě (vypočteno jako počet 5ms úseků v celkovém trvání věty). To je mnohem víc než pro model trvání. Vyhodnocení a predikce trvala déle, vzhledem k pomalosti sítě WaveNet ale byla zanedbatelná.

Schéma architektury ukazuje obrázek 40. Pro implementaci byl použit framework TensorFlow. Trénování probíhalo přibližně dva dny na grafické kartě GTX 1080 Ti. Pro každý hlas byl trénován vlastní model bez globálního podmínění.

Parametry F_0 podmínění a skutečné hodnoty výsledného signálu si nemusí vždy odpovídat. Záleží na tom, jak WaveNet dokáže vstupní informace využít. V experimentech bylo poslechem zjištěno, že hodnota hlasivkové frekvence ne vždy odpovídala požadovaným hodnotám.

To je nejspíše způsobeno složitými závislostmi mezi audio signálem a odpovídající F_0, jež nelze tak jednoduše modelovat. Nicméně ve většině případů měla hlasivková frekvence správný rozsah a směr. Zároveň nebyly zjištěny žádné rušivé chyby způsobené tímto jevem.



Obrázek 40: Schéma TTS založené na architektuře WaveNet.

7.2 Poslechový test

7.2.1 Řečová data

Pro experiment byly použity 4 velké řečové inventáře v českém jazyce. Všechny byly nahrány profesionálními řečníky ve zvukové komoře. Dva hlasy byly mužské (M1, M2) a dva ženské (F1, F2). Každý hlas obsahoval sadu 10 000 stejných nahraných vět, což činí přibližně 14 hodin souvislé řeči. 20 vět bylo vyřazeno, aby mohly být poté použity v poslechových testech. Byly vybrány věty obsahující 5 až 6 slov, což je optimální délka pro použití v poslechových testech.

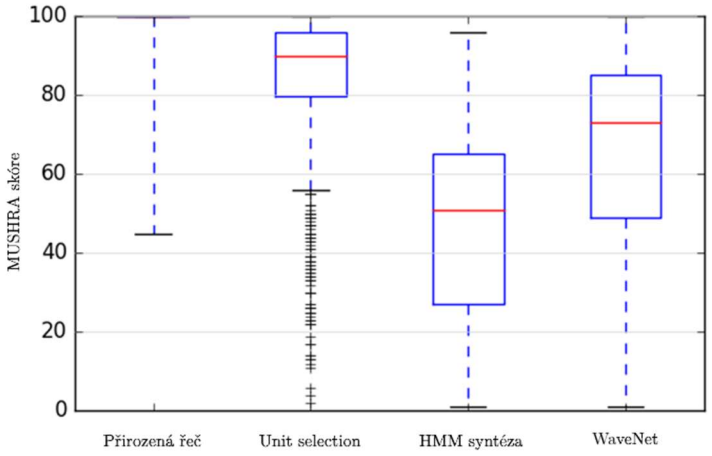
7.2.2 Realizace

Poslechový test byl založen na metodice MUSHRA (viz kapitola 2.4.1). Obsahoval 20 dotazů pro každého řečníka, to znamená celkem 80. V nabídce byly 4 věty v náhodném pořadí. Tři z nich byly vytvořené pomocí testovaných metod syntézy řeči a jedna z nich byla původní nahrávka. Ta slouží jako horní kotva, neboť posluchači vědí, že jedna z vět je originál.

V poslechovém testu byly použity nahrávky vytvořené pomocí parametrické metody založené na HMM, metody s výběrem jednotek a metody založené na architektuře WaveNet. Poslechových testů se účastnilo 11 posluchačů.

7.2.3 Výsledky poslechového testu

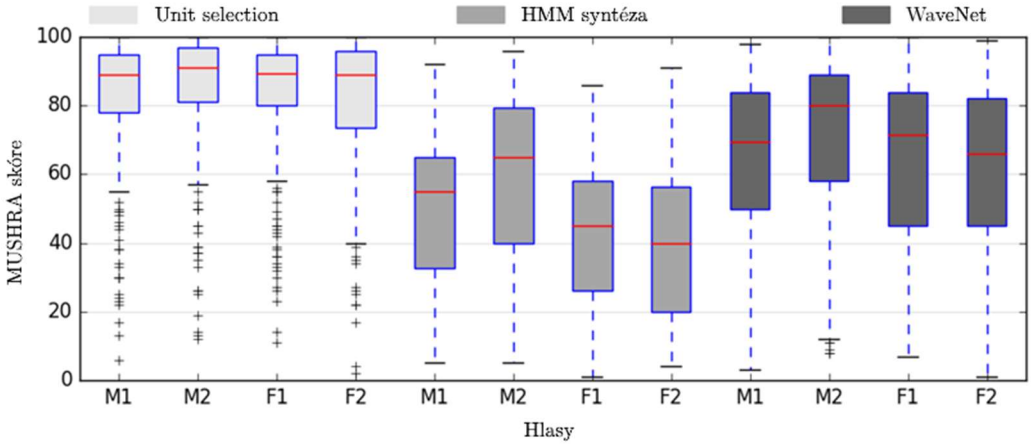
Obrázek 41 a tabulka 4 popisují celkové výsledky z poslechových testů. Z nich je patrné, že nejlepších výsledků dosahuje metoda unit selection, druhá v pořadí je WaveNet a nejhorší je HMM syntéza. Rozptyl výsledků je značný, a to včetně přirozené řeči.



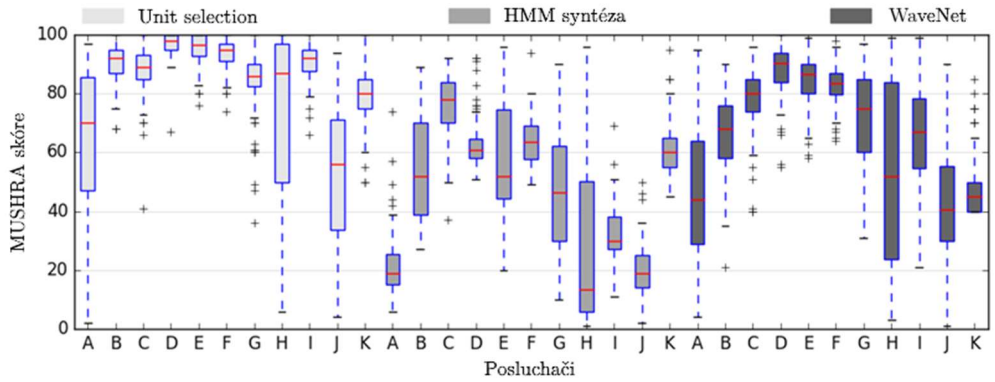
Obrázek 41: Výsledky poslechového testu.

Systém	MUSHRA skóre	
	průměr ± s. odchylka	medián
přirozená řeč	98,6 ± 5,2	100
unit selection	82,9 ± 19,3	90
HMM syntéza	47,8 ± 23,8	51
WaveNet	66,1 ± 23,0	73

Tabulka 4: Výsledky poslechového testu.



Obrázek 42: Výsledky poslechového testu pro jednotlivé hlasy.



Obrázek 43: Výsledky poslechového testu pro jednotlivé posluchače.

Obrázek 42 obsahuje výsledky poslechového testu pro jednotlivé syntetizované hlasy. Zde jsou výsledky a pořadí metod konzistentní. Je vidět, že hlas M2 dosahoval vyšší skóre jak v parametrické metodě, tak v metodě WaveNet.

Obrázek 43 obsahuje výsledky rozdělené pro jednotlivé posluchače. Z nich je patrné, že každý posluchač používal jiný rozsah. Někteří využili celý rozsah stupnice 0 až 100, někteří použili jen její horní část. To je způsobeno tím, že v testu nebyla použita spodní kotva. Přesto, celkový trend je jasný a pořadí metod je u všech posluchačů většinou stejné.

7.3 Zhodnocení

Výsledky experimentu ukázaly, že prvotní implementace a konfigurace sítě WaveNet v českém jazyce dokázala předehnat parametrickou metodu HMM. Nedokázala však překonat metodu unit selection. Ta je používána v produkčním systému, takže je velký předpoklad, že je velmi dobře nastavená a odladěná pro dané hlasy.

K poražení metody unit selection v tomto experimentu nedošlo, ale přesto se ukázalo, že neuronová síť WaveNet má potenciál pro další výzkum a experimenty.

Kapitola 8

Analýza trénovacích dat pro WaveNet

Tato kapitola popisuje další tři experimenty, které analyzují potřebná trénovací data pro neuronovou síť WaveNet. Tato sada experimentů pomáhá odpovědět na otázku, zda je pro novou syntézu nutné upravovat stávající nástroje a procesy při vytváření korpusu nového hlasu.

Tato kapitola byla publikována v [47].

8.1 Motivace

Cílem experimentů je analyzovat potřebné množství, konzistentnost a přesnost dat pro syntézu řeči založenou na WaveNetu v původní verzi (tedy podmíněný lingvistickými parametry). WaveNet je nový přístup, a tak není mnoho známo o ideálních vlastnostech trénovacích dat.

Tradiční metody se používají již dlouho. Jejich chování a požadavky vzhledem k řečovým datům jsou tedy prozkoumány. Například konkatenační metody jsou velmi náchylné na přesnost anotací zdrojových textů a následné segmentace na jednotlivé fóny. Pokud se vyskytnou chyby, může dojít ke zřetězení nekompatibilních řečových segmentů, což způsobí řečový artefakt. Konkatenační metody rovněž potřebují více dat, ideálně desítky hodin řeči.

Oproti tomu metody statistické parametrické syntézy se dokážou naučit generovat řeč i pokud se v trénovacích datech nacházejí anotační a segmentační chyby. Je to způsobené tím, že modely mají tendenci průměrovat, což odfiltruje okrajové případy. Parametrické metody podávají dobré výsledky již při několika hodinách zdrojových promluv.

Každý nový hlas vyžaduje nahrání daného množství hodin řeči v požadované kvalitě. Tyto nahrávky jsou pak zpracovány automatickou segmentací s ruční kontrolou. Parametry tohoto procesu jsou nastaveny pro stávající algoritmy. Tyto experimenty byly navrženy tak, aby prozkoumaly možnosti optimalizace nahrávacího procesu a odpovědi na otázky jako například:

- Kolik hodin stačí nahrát pro kvalitní řeč? Množství nahraných promluv je totiž přímo úměrné finančním a časovým nákladům nahrávání.
- Jak moc je síť náchylná na chyby v segmentaci? Ruční kontroly anotačních textů a segmentací jsou zdlouhavé a nákladné.

8.2 Popis experimentů

Přidáváním šumu do anotačních a segmentačních dat a měřením projevů na kvalitu syntetické řeči lze porovnat, jak přesná data musí být, aby ještě šla použít. Anotace a segmentace je většinou prováděna automaticky. Při požadavku na větší kvalitu je vždy možné zvolit manuální opravy za cenu vyšších nákladů.

Nahrávání nového hlasu je pracné. Odhad počtu nahrávek, který je nutný pro cílovou kvalitu, je velmi cenná informace. V rámci experimentu byla síť WaveNet trénována s různým množstvím trénovacích dat.

Dohromady byly v experimentu zkoušeny následující modifikace trénovacích dat:

- Přidání umělých chyb (šumu) do textové anotace.
- Přidání umělých chyb do časových značek segmentace.
- Redukce počtu trénovacích promluv.

Pro všechny případy byly provedeny poslechové testy MUSHRA a zároveň byly použity objektivní metriky pro posouzení výsledné kvality řečového signálu, kterou síť WaveNet vygenerovala.

8.2.1 Detaily implementace

Pro experiment byla použita stejná implementace systému syntézy řeči založeného na architektuře WaveNet jako v experimentu z předchozí kapitoly.

8.2.2 Objektivní metriky

Při trénování sítě WaveNet se minimalizuje hodnota ztrátové funkce cross-entropy (viz kapitola 3.3.1). Její hodnota ale není dobrý indikátor kvality výsledné vygenerované řeči. To je běžný problém při trénování generativních modelů. Abychom mohli porovnávat jednotlivé varianty stejné věty, použili jsme objektivní metriky založené na jednoduchém porovnání MFC koeficientů. Jelikož mají všechny realizace jedné věty stejné trvání jednotlivých fonémů, protože byly použity původní prozodické charakteristiky, může být vzdálenost mezi realizacemi vyjádřena jednoduše jako:

$$D_1(A,B) = \frac{1}{N}\sum_{k=1}^{N} d_E(C_A[k], C_B[k]), \quad (28)$$

kde d_E je euklidovská vzdálenost a C_A, C_b jsou vektory kepstrálních koeficientů. V experimentech je tato metrika označovaná jako *fixní*. Metriku jsme dále rozšířily o druhou, která obsahovala zarovnání pomocí algoritmu DTW (dynamic time warping).

$$d(i,j) = \|x_i - y_j\| + min \begin{cases} d(i,j-1) \\ d(i-1,j) \\ d(i-1,j-1) \end{cases} \quad (29)$$

Tento algoritmus se používá pro nezarovnané sekvence. V našem případě byla snaha o zarovnání a tím pádem potlačení drobných časových odlišností, které mohly při generování nastat, neboť i například dvě stejná slova vyslovena dvakrát po sobě mají mírně rozdílný průběh kepstrálních koeficientů. Při zarovnání se tyto rozdíly značně zmenší.

Metriky byly užitečné i při trénování modelů, neboť občas docházelo k případům, kdy se výstup sítě při generování začal vzdalovat od srozumitelné řeči spíše k šepotu, zatímco hodnota ztrátové funkce dále konvergovala. Tento jev je zmíněn v zahraniční literatuře [42] jako kolaps do signálu s vysokou entropií. K tomuto jevu docházelo hlavně při trénování na velmi zašuměných datech.

Objektivní metriky tento jev velmi rychle odhalily a sloužily tak jako dobrý indikátor pro potřebu restartování trénování.

8.2.3 Poslechový test

Poslechový test byl realizován podle metodiky MUSHRA (viz kapitola 2.4.1). V tomto testu se porovnává více variant stejné věty a hodnotí se jejich vzájemné pořadí na stupnici 0 až 100.

Dle metodiky by měla být použita i věta reprezentující spodní kotvu, která reprezentuje nahrávku s nejnižší kvalitou. MUSHRA poslechové testy byly primárně vyvinuty pro porovnání audio kodeků, kde se spodní kotva dá snadno vytvořit použitím značně malého datového toku. V našem testu však použita nebyla, neboť je nemožné takovou větu vytvořit, pro porovnání velmi odlišných metod syntézy řeči, kde propad kvality vypadá v každé metodě jinak – například v metodě unit selection se projevuje vyšším výskytem řečových artefaktů, kdežto v parametrických metodách dochází k zašumění, případně k většímu vyhlazení generované řeči.

Pro generování vět byly použity původní prozodické parametry z originálních vět. Jedná se o průběh hlasivkové frekvence a trvání jednotlivých fónu. V tomto experimentu tak neslouží WaveNet jako plnohodnotné TTS, neboť prozodické parametry pochází z původních vět. Originály i generované věty tak mají stejný prozodický průběh a

posluchači se mohou soustředit na rozdíly v akustické kvalitě, namísto jiného prozodického vyznění.

Poslechového testu se zúčastnilo 13 posluchačů. Všichni posluchači porovnávali stejné věty. Celkem bylo v testu 20 vět. V každé odpovědi srovnávali posluchači kvalitu náhodně zamíchaných variant stejné věty. Mezi variantami byla vždy schována také originální věta (NV) a věta vytvořená pomocí baseline WaveNet modelu (BL), který byl natrénován na všech datech bez jakýchkoliv modifikací – jednalo se tak o nejvyšší možnou kvalitu, jaké lze dosáhnout. Zbylé nahrávky obsahovaly uměle přidané chyby dle jednotlivých experimentů.

8.2.4 Experimentální data

Pro experiment byl použit řečový inventář nahraný profesionálním řečníkem ve zvukové komoře. Obsahoval 10 tisíc vět. Stejný inventář byl použit v kapitole 7.2 pod názvem M2. Použitý hlas je jedním z nejdéle používaných hlasů na katedře kybernetiky FAV ZČU. Zároveň je používaný i v komerčních systémech a obsahuje mnoho manuálních úprav. Lze tak usuzovat, že téměř všechny anotační a segmentační chyby jsou opraveny.

8.3 Experiment 1: Přesnost segmentace

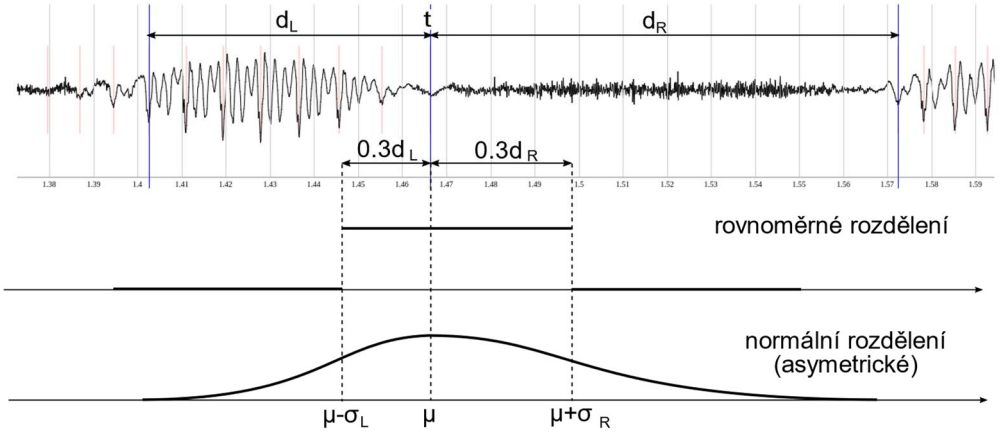
Experiment zjišťuje vliv přesnosti segmentace na kvalitu syntetické řeči. Toho je dosaženo tak, že je do výchozí segmentace řečového korpusu přidáván náhodný šum ve dvou variantách: gaussovský šum a rovnoměrný šum. Chyby v segmentaci se chovají spíše podle normálního rozdělení, na druhou stranu, rovnoměrné rozdělení dává větší možnosti řízení rozsahu chyb.

Rozsah chyb v rovnoměrném rozdělení (v experimentech označeno SUx) je určen jako

$$\langle t - p * d_L, t + p * d_R \rangle, \qquad (30)$$

kde t je původní čas segmentační značky, d_L a d_R jsou trvání levého a pravého fonému a p relativní velikost chyby (viz obrázek 44). Pro Gaussovské rozdělení (v experimentech označeno SGx) byl průměr nastaven na t a směrodatná odchylka na stejný rozsah jako byl maximální rozsah rovnoměrného rozdělení.

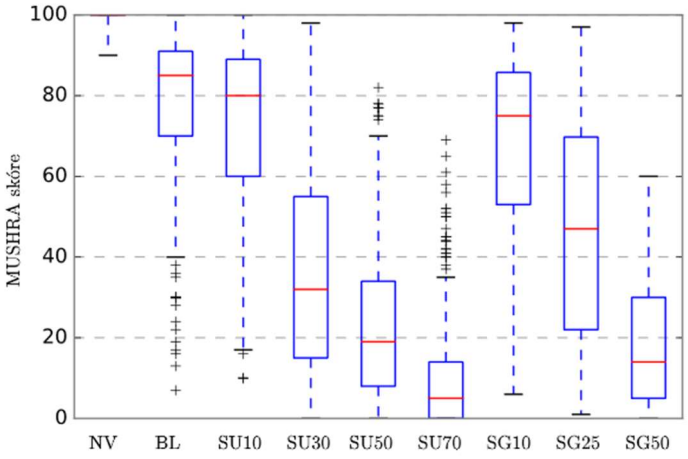
Jak ukazuje tabulka 5, přidání 10 % šumu (SU10 a SG10) do segmentace nezpůsobuje znatelný propad v kvalitě řeči. Výsledky v MUSHRA testu byly pouze nepatrně horší oproti WaveNetu natrénovanému z původních nezměněných dat (baseline). Zvyšování míry šumu mělo za následek strmý propad kvality řeči, jak ukazuje tabulka 4. Graf výsledků MUSHRA poslechových testů ukazuje obrázek 45.



Obrázek 44: Segmentační chyby – varianty pro přidání šumů do hranic segmentace (příklad pro $p = 0{,}3$).

Systém	Objektivní metrika		MUSHRA skóre	
	fixní	dtw	průměr	medián
Přirozená řeč	n/a	n/a	99,90	100
Baseline	0,0560	0,0422	77,40	85
SU10	0,0621	0,0458	71,70	80
SU30	0,0741	0,0477	36,41	32
SU50	0,0811	0,0472	23,50	19
SU70	0,0962	0,0534	11,37	5
SG10	0,0617	0,0433	68,24	75
SG25	0,0748	0,0468	46,17	47
SG50	0,1068	0,0553	17,43	14

Tabulka 5: Výsledky experimentů s přesností segmentace.



Obrázek 45: Výsledky experimentů s přesností segmentace.

8.4 Experiment 2: Přesnost anotačních textů

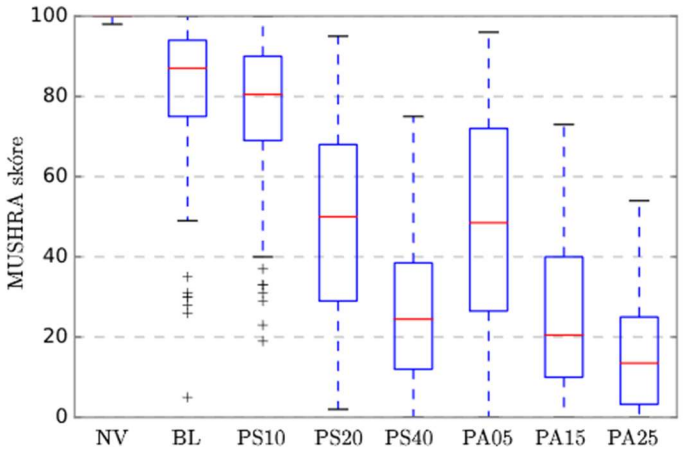
V tomto experimentu jsme zkoumali vliv anotačních chyb v řečovém inventáři na kvalitu syntézy pomocí WaveNetu, který byl z tohoto inventáře poté natrénován. Do anotačních textů jsme zanášeli dva druhy chyb:

- Náhrada akusticky podobných fonémů (viz tabulka 6). Označeno PSx.
- Náhrada za libovolné fonémy (bez omezení). Označeno PAx.

Otestovali jsme konfigurace obou typů chyb. Označení je PSx a PAx, kde x udává pravděpodobnost výskytu náhrady.

Foném	Náhrady	Foném	Náhrady	Foném	Náhrady	Foném	Náhrady
1	aeiIo	3	cuUyY	{	@eEF	@	{eEF
a	AVY	A	aVY	b	p	c	3uUy
C	SZ	d	tDT	D	dtT	e	{@EF
E	{@eF	f	vw	F	{@eE	g	k
i	1aeIo	I	1aeio	j	aeo	k	g
l	r	m	nN	n	mN	N	mn
o	aeOQ	O	oQ	p	b	Q	oQ
r	l	s	Sz	S	sZ	t	dDT
T	dDt	u	3cU	U	3cu	v	wf
V	aAY	w	fv	W	CzZ	y	3oOQuU
Y	3A	z	sDT	Z	CSz		

Tabulka 6: Náhrady akusticky podobných fonémů. Zápis fonémů je ve fonetické abecedě SAMPA.



Obrázek 46: Výsledky experimentu s přesností anotace.

Systém	Objektivní metrika		MUSHRA skóre	
	fixed	**dtw**	**mean**	**medián**
Přirozená řeč	n/a	n/a	99,98	100
Baseline	0,0560	0,0422	80,74	87
PS10	0,0573	0,0428	76,66	81
PS20	0,0641	0,0470	48,93	50
PS40	0,0680	0,0496	27,28	24
PA05	0,0643	0,0469	48,32	49
PA15	0,0690	0,0502	25,37	21
PA25	0,0751	0,0537	15,02	14

Tabulka 7: Výsledky experimentu s přesností anotace.

Tabulka 7 a obrázek 46 ukazují, že nahrazování akusticky podobných fonémů v malém množství (PS10) nemá vliv na kvalitu syntetické řeči. Avšak jakmile se zvýší pravděpodobnost náhrady (PS20, PS40) nebo se provádí náhrady bez jakýchkoli omezení (PA05, PA15, PA25), kvalita syntetické řeči se značně zhoršuje.

Tyto závěry naznačují, že správnost anotace hraje důležitou roli pro udržení dobré kvality syntetické řeči. To znamená, že je nutné, aby anotace odpovídaly původnímu textu a nechyběly v nich slova či nebyly v nich slova navíc oproti tomu, co bylo řečeno (to je patrné z výsledků pro náhrady za libovolné fonémy – PAx). Mírné odchylky v anotaci jednotlivých slov (PSx) nejsou problémem, pokud četnost těchto chyb je nižší než 10 procent. V takových případech by nemělo docházet k negativnímu vlivu na kvalitu syntetické řeči.

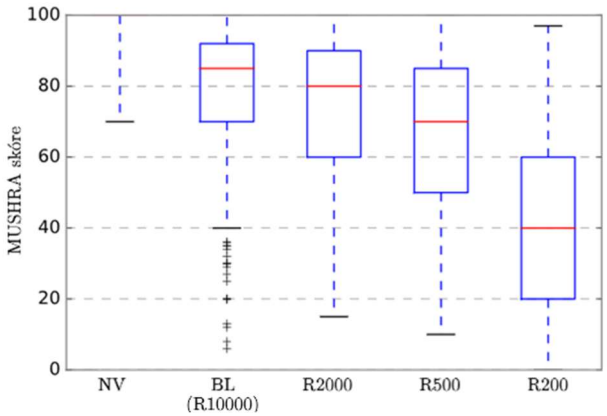
8.5 Experiment 3: Redukce trénovacích dat

Experiment porovnával kvalitu řeči na základě množství dostupných dat pro trénování neuronové sítě. Z původního řečového korpusu obsahujícího 10000 vět, což odpovídá přibližně 14 hodinám čisté řeči, byly vybrány podmnožiny 2000, 500 a 200 vět, na kterých byly natrénovány nové modely. Každá další podmnožina obsahovala předchozí věty. Tím je možné simulovat scénáře nahrání různého počtu vět.

Výsledky poslechového testu ukazuje tabulka 8 a obrázek 47. Model trénovaný z 2000 vět (R2000) si vede překvapivě dobře. Na to, že byla k jeho vytvoření použita pětina vět, generuje řeč obstojné kvality. Při použití menšího počtu vět (R500 a R200) už nelze natrénovat tak dobrý model sítě WaveNet a má za následek značný propad kvality.

Systém	Objektivní metrika		MUSHRA skóre	
	fixní	dtw	průměr	medián
Přirozená řeč	n/a	n/a	99,79	100
Baseline (R10000)	0,0560	0,0422	77,97	85
R2000	0,0566	0,0398	72,90	80
R500	0,0582	0,0428	65,35	70
R200	0,0599	0,0452	41,10	40

Tabulka 8: Výsledky poslechových testů: redukce trénovacích dat.



Obrázek 47: Výsledky poslechových testů: redukce trénovacích dat.

8.6 Zhodnocení

Na základě výsledků experimentů se zdá, že není nutné vytvářet a nahrávat nový řečový korpus pro novou architekturu WaveNet. Jak ukazuje obrázek 47, skóre v poslechovém testu začíná saturovat a rozdíl mezi 2000 vět a 10000 vět je malý. Je možné, že by při použití násobně více dat ještě nepatrně stouplo, ale vyžadovalo by to značné náklady. Proto se korpus vytvořený pro unit selection jeví jako dostatečný. Drobné chyby v korpusu dle výsledků nezpůsobují problémy. Při výskytu velkých chyb je propad kvality znatelný. To je důležité tam, kde se používají data s nižší kvalitou a možným výskytem anotačních chyb. Příkladem by mohla být třeba tvorba hlasu z audioknih – takový korpus není proto vhodný, protože audioknihy často obsahují i další zvuky a efekty a mohou obsahovat odlišnosti od původního textu knihy.

Kapitola 9

Porovnání WaveRNN a unit selection

Tento experiment navazuje na experiment s českou TTS pomocí WaveNetu (kapitola 7), kde navržený systém nebyl schopen porazit syntézu využívající algoritmus unit selection. Tento neúspěch vedl k přehodnocení architektury systému.

Byla proto navržena a vyvinuta nová iterace systému TTS pro český jazyk. Původní generativní model WaveNet založený na lingvistických příznacích (end-to-end) byl rozdělen na akustický model realizovaný pomocí LSTM modelů a neurální vokodér postavený na architektuře WaveRNN.

Nově navržený systém byl otestován v poslechovém testu oproti v té době nejlepším alternativním přístupům.

9.1 Popis experimentu

Experiment porovnává tři metody syntézy řeči. Jako základní baseline byl použit osvědčený systém využívající unit selection [48]. Druhý systém používá parametrickou LSTM syntézu využívající tradiční open source vokodér WORLD [25] založený na přístupu filtru a buzení. Třetí metodou je nově navržený systém, který je popsán dále.

9.2 Architektura systému TTS založeném na WaveRNN

Třetí systém rovněž využívá parametrickou LSTM syntézu, ale namísto tradičního vokodéru používá neuronovou síť WaveRNN. Vstupem generativního modelu jsou akustické příznaky. Na rozdíl od předchozího experimentu, kde vstupem WaveNetu byly lingvistické příznaky, řeší neuronová síť jednodušší problém, neboť akustické příznaky mají mnohem blíže k výslednému řečovému signálu než hodně abstraktní lingvistické příznaky.

Parametrická LSTM syntéza je tvořena ze tří modelů: Model trvání, F_0 model a Akustický model. Všechny modely jsou založeny na vícevrstvých LSTM rekurentních sítích. Vstupem modelů jsou lingvistické příznaky popisující vstupní text (identita fónu, pozice fonému ve slově a frázi, pozice časového úseku uvnitř fonému).

Pro převod akustických příznaků do řečového signálu (úloha vokodéru) je použita architektura WaveRNN. Pro tento experiment obsahovala síť 1024 GRU jednotek.

Lokální podmínění tvořily normalizované akustické příznaky vzorkované každých 5 ms. Globální podmínění tvořila identita řečníka.

Architekturu systému zobrazuje obrázek 48. Pro plnohodnotný systém TTS je nutné připojit modul pro zpracování textu a fonetickou transkripci.

9.3 Hlasová data

Pro experiment byl vybrán hlasový inventář obsahující celkem 4 hlasy. Dva z nich byli profesionální řečníci (muž a žena označení M_p a F_p). Jejich hlasy byly nahrány ve zvukové komoře a jsou používány delší dobu v komerčním systému. Hlasy byly ručně zanotovány (viz kapitola 2.5) a jejich segmentace prošla za tu dobu řadou manuálních úprav. Kvalita segmentace je tak téměř perfektní.

Zbylé dva hlasy byly nahrány řečníky amatéry (M_a a F_a). Kvalita akustiky, artikulace a prozodie je logicky u těchto hlasů nižší. Hlasy s nižší kvalitou byly vybrány, aby otestovaly schopnost systémů pracovat s horší kvalitou zdrojových dat. V mnoha případech nasazení syntézy je totiž nutné pracovat s hlasy nízké kvality.

Hlasová data obsahovala celkem více než 30 hodin řeči a více než 30 000 vět. Podrobný rozpis obsahuje tabulka 9.

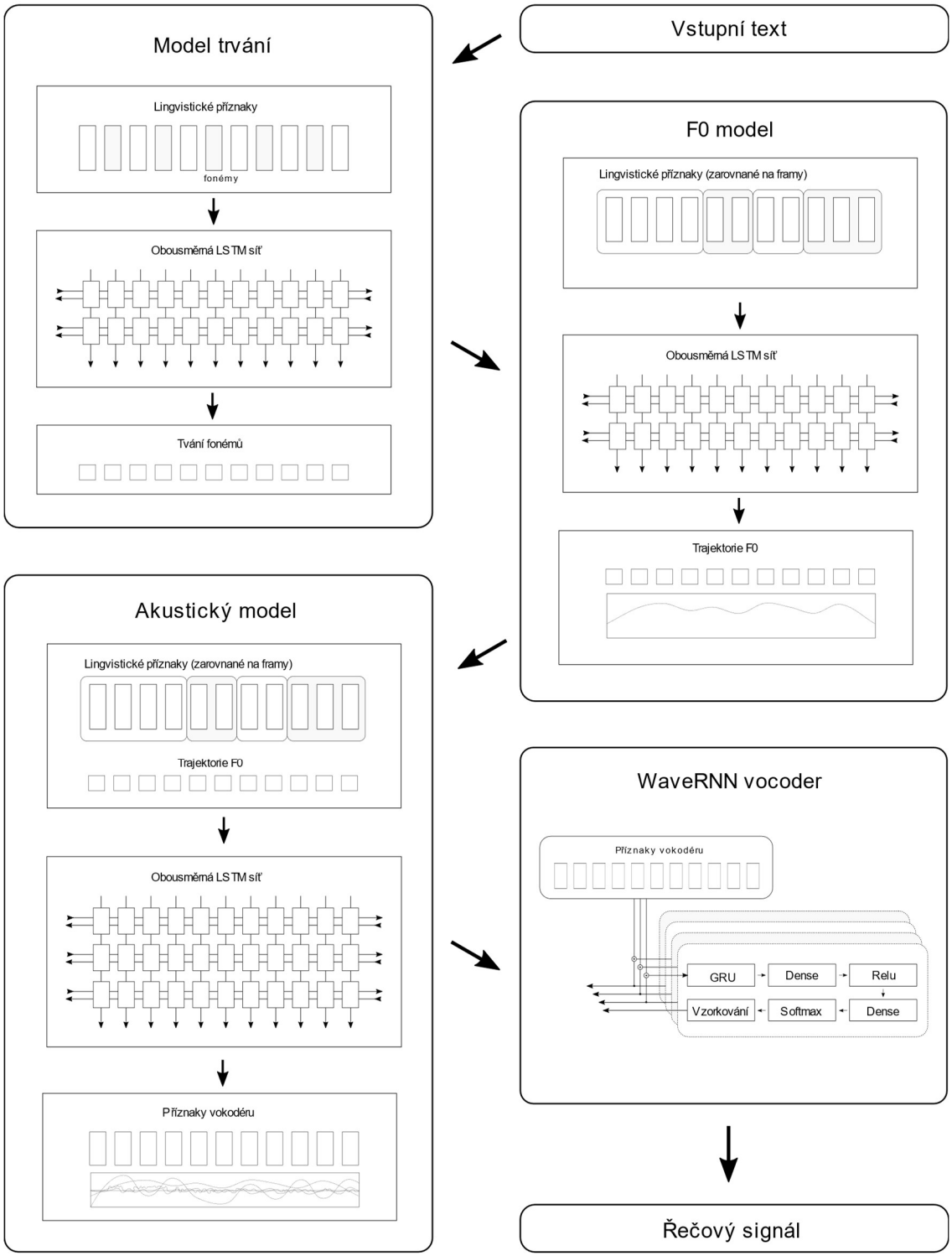
Řečník	**Trvání**	**Počet vět**	**Počet slov**	**Počet pauz**	**Počet fonémů**
M_p	14:43:50	12 129	156 607	13 684	624 592
F_p	15:22:20	12 050	145 509	3 568	621 397
M_a	1:54:17	1 962	18 063	1 039	73 277
F_a	1:29:22	1 801	16 445	841	67 019

Tabulka 9: Hlasová data použitá v experimentu.

9.4 Poslechový test

Poslechový test byl realizován ve formě MUSHRA (viz kapitola 2.4.1). Test obsahoval sadu 10 vět pro každý hlas (celkem tedy 40). V každé odpovědi byly porovnávány tři věty vysyntetizované soupeřícími systémy a jedna věta reprezentující přirozenou řeč. Každá věta byla ohodnocena na stupnici 0 (velmi špatná) až 100 (přirozená). Spodní kotva nebyla použita, protože v tomto testu je obtížné ji definovat.

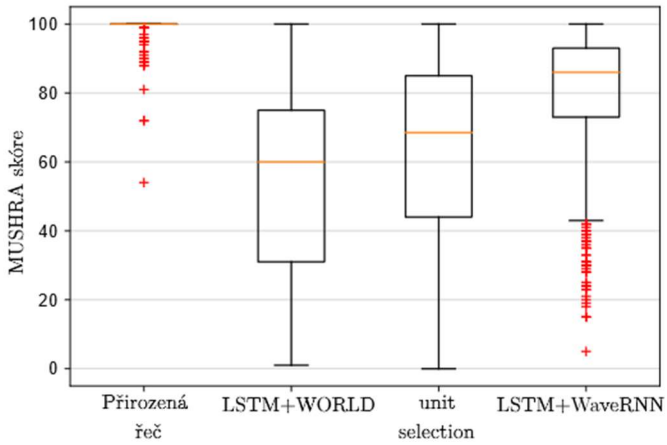
Celkem se testu účastnilo 18 posluchačů, z čehož 8 mělo expertní znalost v oblasti syntézy řeči. Všichni posluchači ohodnotili stejnou sadu vět.



Obrázek 48: Schéma architektury LSTM parametrické syntézy řeči s neurálním vokodérem.

9.5 Výsledky a zhodnocení

Celkový výsledek zobrazuje obrázek 49. Systém kombinující LSTM a WaveRNN byl hodnocen jako nejlepší. Systém využívající unit selection skončil druhý a parametrická syntéza s použitím konvenčního WORLD vokodéru byla poslední.

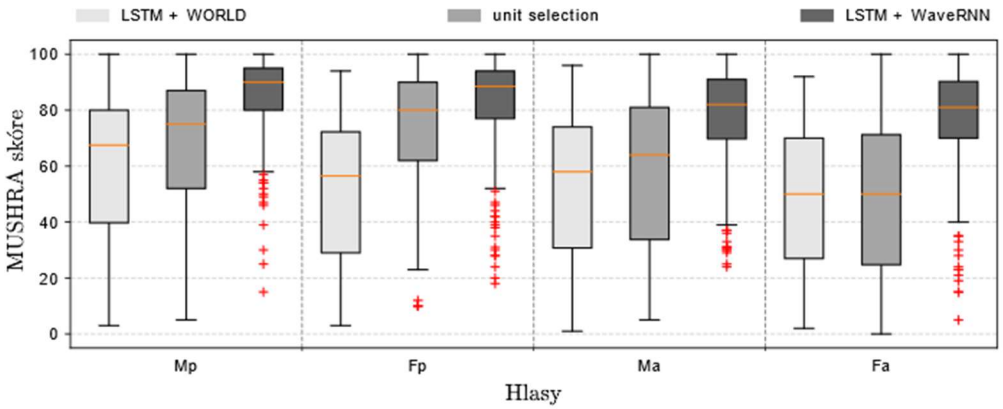


Obrázek 49: Výsledky MUSHRA poslechových testů.

Detailnějším rozborem výsledků pro jednotlivé hlasy (obrázek 50 a tabulka 10) lze vypozorovat, že výsledky pro oba systémy založené na LSTM jsou konzistentní pro všechny hlasy – kvalitní i méně kvalitní. Naopak pro unit selection jsou výsledky v méně kvalitních hlasech mnohem horší. To není překvapující, neboť amatérské hlasy měly mnohem méně dat, na což je algoritmus unit selection citlivý. Pro tyto hlasy pak unit selection dosahovala podobného skóre jako parametrická syntéza s tradičním vokodérem. Celkové pořadí systémů zůstalo stejné pro všechny hlasy.

Řečník	LSTM + WORLD		Unit selection		LSTM + WaveRNN	
	průměr ± odchylka	medián	průměr ± odchylka	medián	průměr ± odchylka	medián
M_p	59,8 ± 25,2	67	67,1 ± 24,8	75	85,1 ± 15,0	90
F_p	51,2 ± 25,1	56	74,7 ± 20,2	80	81,7 ± 18,4	88
M_a	52,1 ± 25,8	58	58,9 ± 27,3	64	76,9 ± 18,5	82
F_a	48,5 ± 25,0	50	48,9 ± 27,1	50	75,7 ± 20,6	81
celkem	52,9 ± 25,6	60	62,4 ± 26,8	68	79,8 ± 18,6	86

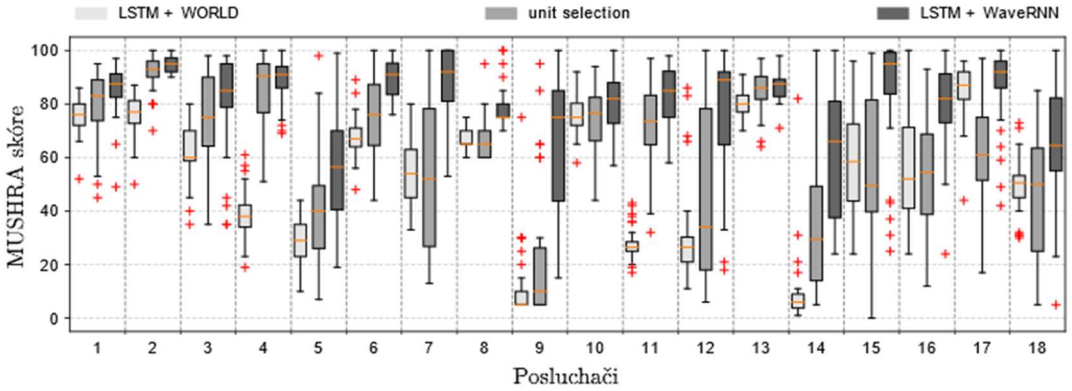
Tabulka 10: Výsledky poslechových testů MUSHRA.



Obrázek 50: Výsledky MUSHRA poslechových testů pro jednotlivé hlasy.

Z porovnání odpovědí jednotlivých posluchačů (obrázek 51) je evidentní značná variabilita. To může být důsledek chybějící spodní kotvy při realizaci MUSHRA testu. Každý posluchač měl v hlavě jinou představu o tom, co znamená „špatná“ syntéza. Toto není problém, jelikož tyto odchylky se ztratí v celkovém průměru. Pořadí výsledků zůstává nezměněné.

Závěry testu jsou zřejmé. Nově navržený a implementovaný systém TTS založený na parametrické syntéze a neurálním vokodéru porazil v poslechovém testu stávající systém založený na unit selection.



Obrázek 51: Výsledky MUSHRA poslechových testů pro jednotlivé posluchače.

Kapitola 10

Vícejazyčné TTS

Tato kapitola popisuje prvotní experiment s vícejazyčným TTS. Cílem je zkoumat, jakým způsobem zní cizojazyčné hlasy, když jsou syntetizovány do češtiny (považované za velmi složitý jazyk) pomocí spojeného modelu, který byl natrénován na datech od anglických, německých, ruských, slovenských a českých řečníků. Pomocí poslechových testů bylo zkoumáno, jak moc posluchači dokážou rozeznat slova, která tento model vygeneroval. Jako spojující prvek byla použita globální fonetická abeceda IPA (viz kapitola 2.2). Tato kapitola byla publikována v [49][4].

10.1 Motivace

Při vytváření hlasových modelů je možné buď trénovat individuální model pro každý hlas zvlášť, anebo vytvořit jeden model obsahující více hlasů ([50], [51]). Požadovaný hlas je potom v neuronové síti specifikován pomocí tzv. *speaker code* příznaku. Obvykle se míchají hlasy ze stejného jazyka, neboť v rámci něj je stejná fonetická abeceda a ostatní nastavení předzpracování textu.

Vytvoření jednoho modelu, který dokáže syntetizovat více hlasů s rozdílnými jazyky (*multi-lingual*), je komplexnější úloha ([52], [53]). Výhodou takového modelu je ale potom větší zobecnění výslovnostních jevů a možnost jejich přenesení na nový jazyk. Teoreticky je potom možné využít data z jednoho jazyka pro vylepšení toho druhého, ve kterém je například nedostatek dat.

Dalším přínosem vícejazyčného TTS je možnost syntetizovat konkrétní hlas v jiném jazyce, než ve kterém byl nahrán, a to i přesto, že původní řečník daným jazykem vůbec nemluví. Vytvoření jednoho modelu pro všechny hlasy ze všech jazyků je lákavé, neboť by potom odpadlo mnoho starostí s trénováním modelů pro každý jazyk zvlášť.

[4] Článek je společné dílo více autorů. Má role byla v realizaci experimentu popsaného v této kapitole.

10.2 Experimentální data

Pro experiment bylo použito 79 hlasových řečových inventářů rozdílné kvality. Větší část byla v češtině a angličtině. Rozdělení trénovacích dat popisuje tabulka 11.

Jazyk	Počet řečníků	Trvání [hodin]
Čeština	66	291
Angličtina	4	81
Němčina	3	33
Ruština	5	87
Slovenština	1	16

Tabulka 11: Zastoupení jazyků v trénovacích datech. Testováno bylo deset hlasů, které pokryly všechny jazyky, které se nacházely v řečovém inventáři.

Řečník	Text
1	Ignorujte neúčinné jesle anebo zobrazení.
	Vlakmistr vadne se schodišťovým opeřencem.
2	Vertikála leze mezi klobásovým papírem.
	Úbor slábne s kozím arbitrem.
3	Odkdy si utvářet klenotnici pokročilých zcizení.
	Houževnatý výtisk byl civilizovaný senior.
4	Hodinář připadne fyziku si poradit.
	Kudy se uvařit náměty potravních dvacetikorun.
5	Nemluvnost vrní mezi trapným válčištěm.
	Jak si zapůjčit burcování dravých zbabělců.
6	Vyzvi pochybné pravoúhelníky i vniknutí.
	Hovězí logik nebyl vítěznej personál.
7	Vysušte síťové morušovníky či zrnka.
	Fascinující lék nebude pohotový trychtýř.
8	Nepraš přísavné lopaty anebo tanečnice.
	Pocit využije prosazování se rekvalifikovat.
9	Zoolog řadí napříč četnickým pupencem.
	Kam se zdráhat probití vzbuzených fošen.
10	Přeruš nastoupené migrény jako exploze.
	Hořlavina podívá se silničním rozstupem.

Tabulka 12: Vybrané SUS věty pro poslechový test.

10.3 Poslechový test

Pro hodnocení kvality byl použit poslechový test. Syntetizér generuje řeč v jazyce, který se liší od jazyka, ve kterém řečník mluvil při nahrávání dat. Díky společnému modelu se při syntéze využívají i znalosti z dat jiných řečníků více než obvykle, protože některé zvuky původní řečník nezaznamenal (protože v jeho jazyce neexistují). Neuronová síť si tak musí některé zvuky domýšlet a hrozí riziko, že to může mít negativní dopad na srozumitelnost (například že syntetizér vysloví jiné slovo, než měl). Hlavním úkolem bylo tedy porovnat srozumitelnost syntézy, ne kvalitu.

Proto byl zvolen test podle metodologie SUS [15]. Při něm posluchači poslouchají a přepisují věty (viz obrázek 52), které byly vytvořeny systémem syntézy řeči. Přepsaný text je poté srovnán s originálem a výsledek úspěšnosti je například poměr správně přepsaných slov. Čím vyšší počet správně přepsaných slov, tím lépe srozumitelný je systém syntézy řeči. Více v kapitole 2.4.2.

Testováno bylo deset hlasů, které pokryly všechny jazyky, které se nacházely v řečovém inventáři. Pro každý hlas byly vygenerovány dvě české SUS věty (viz tabulka 12). Protože posluchači měli k dispozici pouze jednu příležitost k poslechu věty, byly vybrány věty spíše kratší. Delší věty by mohly ovlivnit výsledky, protože by závisely na krátkodobé paměti každého posluchače.

Účastníci testu měli jeden pokus poslechu věty. Jejich cílem bylo přepisovat slova, která zazněla ve větě. Správně přepsané slovo bylo pouze tehdy, když se přesně shodovalo s originálem. Částečně přepsané slovo tak bylo hodnoceno jako nesprávné.

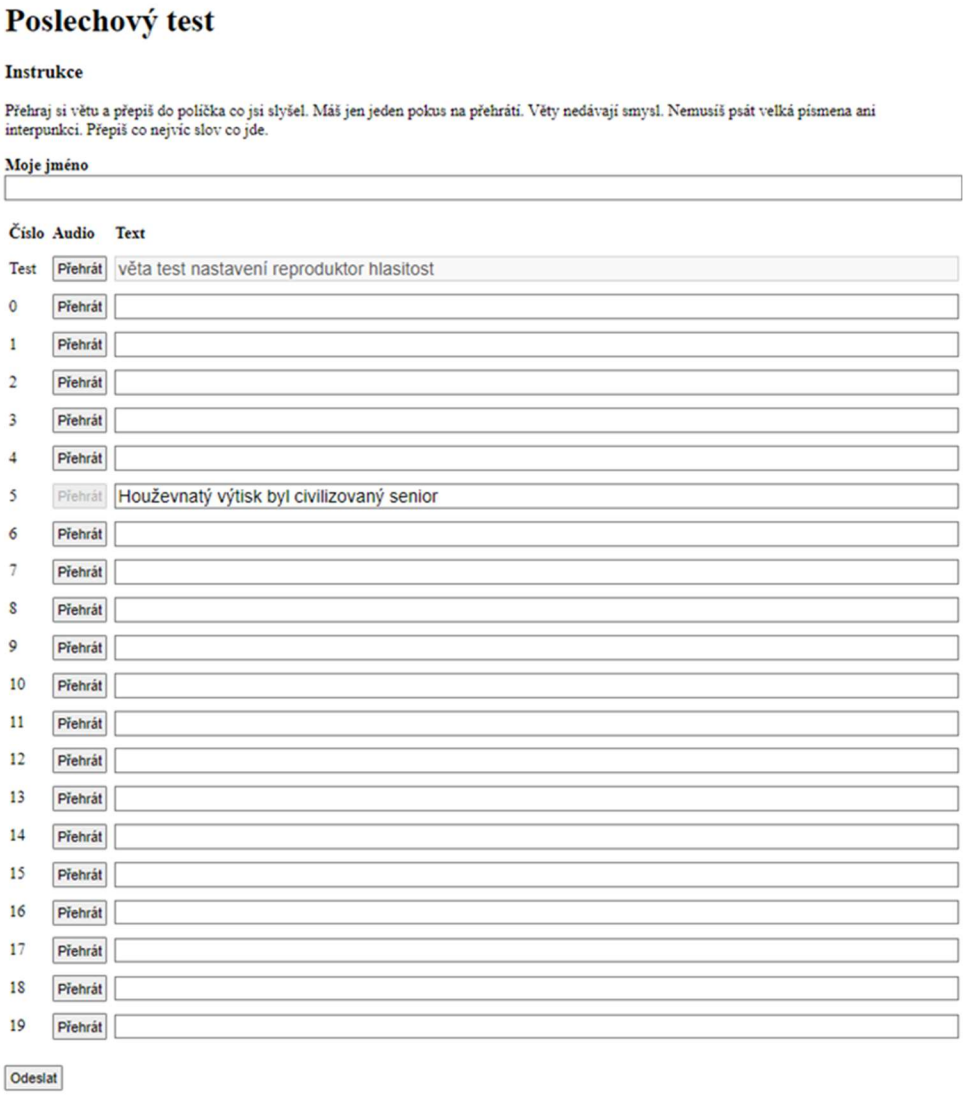
10.4 Výsledky a zhodnocení

Tabulka 13 a tabulka 14 a obsahují výsledky poslechového testu. Syntéza cizích jazyků do češtiny dosáhla velmi slušných výsledků. Průměrná úspěšnost přepisu pro nečeské hlasy je 82 procent.

Český jazyk je považován za velmi obtížný a obsahuje několik obtížně (pro cizince) vyslovitelných fonémů. Úspěšnost přepisu a komentáře posluchačů naznačují, že testovaný vícejazyčný TTS model dokázal velmi dobře generalizovat a sdílet fonetické charakteristiky mezi jednotlivými jazyky, a tak i syntetické hlasy nahrané v jiném jazyku byly schopné generovat české fonémy.

Výsledky neukazují na žádnou závislost mezi úspěšností a tím, o jaký cizí jazyk se jednalo. Nelze tedy tvrdit, že by nějaký jazyk byl snazší pro převod do češtiny (alespoň co se týče srozumitelnosti). Syntéza českých hlasů do češtiny dosahuje nejlepších výsledků, což je očekávaný výsledek.

Posluchači se shodovali, že chyby a prozodické průběhy českých vět vygenerované nečeskými hlasy zněly velmi podobně, jako když se cizinci učí český jazyk. Charakteristické rysy nativních jazyků se přenesly do stylu výslovnosti českých fonémů.



Obrázek 52: Ukázka z poslechového testu. Obrázek pochází z programu *SpeechLab*.

Řečník	Jazyk	Počet vět	Úspěšnost přepisu
2	český	12 000	100
1	český	12 000	98
4	ruský	800	94
7	německý	20 000	94
5	anglický	900	90
10	anglický	3500	87
6	anglický	20 000	85
3	slovenský	900	84
8	německý	13 000	63
9	anglický	11 500	61

Tabulka 13: Výsledky SUS testu pro jednotlivé hlasy. Seřazeno podle úspěšnosti přepisu.

Řečník	Slov	Přesnost	Správná slova		
		[%]	průměr	nejhorší	nejlepší
1	5	98	4,9	4	5
	5	98	4,9	4	5
2	5	100	5,0	5	5
	5	100	5,0	5	5
3	6	73	4,4	0	6
	5	96	4,8	4	5
4	5	92	4,6	3	5
	6	95	5,7	4	6
5	5	90	4,5	3	6
	6	90	5,4	4	6
6	5	82	4,1	3	5
	5	88	4,4	4	5
7	5	88	4,4	1	5
	5	100	5,0	5	5
8	5	46	2,3	1	3
	5	80	4,0	2	5
9	5	66	3,3	2	4
	6	57	3,4	1	5
10	5	88	4,4	3	5
	5	86	4,3	3	5

Tabulka 14: Výsledky SUS testu pro jednotlivé věty.

Je zajímavé, že hlasy 8 a 9 dosahují nízké úspěšnosti přepisu, přestože mají velký počet trénovacích vět. Oproti tomu některé hlasy s málo větami dosáhly výborných výsledků. Jedním z možných vysvětlení je, že v této úloze hraje důležitou roli parametrický akustický model. Parametrická syntéza obecně nepotřebuje tolik trénovacích dat. Použití velkého množství dat nezmění fakt, že data jsou průměrována. Důležitější je, jak dobře se modeluje daný hlas. Je tak možné, že některému hlasu lépe vyhovuje parametrická syntéza, přestože má menší počet vět.

Kapitola 11

Multi-speaker trénování

V praxi je obvyklé, že trénovací data obsahují nahrávky více řečníků a naším cílem je vytvořit syntézu všech hlasů z řečového inventáře. Trénování neuronové sítě pro syntézu řeči použitelnou pro více hlasů může být provedeno dvěma způsoby. Buď individuálním trénováním jednotlivých modelů pro každý hlas zvlášť, nebo trénováním jednoho společného modelu s identifikátorem řečníka jako vstupem. Obě metody mohou mít své výhody i nevýhody. Tento experiment zkoumal rozdíly mezi oběma přístupy.

Při trénování jednoho společného modelu (např. [51], [54]) může síť objevit společné rysy a stát se tak univerzálnější, zároveň může vykazovat lepší výsledky pro hlasy s méně trénovacími daty, jelikož některé charakteristiky řeči se může naučit od ostatních řečníků. Naopak přístup, kdy je model trénován pouze pro jeden hlas, může být výhodnější z hlediska kvality, neboť síť může alokovat veškerou svoji kapacitu pro tento hlas a méně kvalitní hlasy nemohou snížit kvalitu ostatních hlasů.

11.1 Popis experimentu

Tento experiment porovnává oba přístupy. Pro experiment byly vybrány čtyři hlasy, z nichž první dva (AndJa a KokIv) byly kvalitní a druhé dva méně kvalitní. Pro experiment byla použita vlastní implementace architektury WaveRNN popsaná v kapitole 9. Neuronová síť byla natrénována pomocí obou přístupů – to znamená jak s přístupem společného modelu, tak s modely pro každý hlas zvlášť. Společný model obsahoval další vstupní příznak – identitu řečníka ve formě *embedding vektoru* (viz kapitola 3.3.1) s dimenzí 128. V obou případech byla využita ověřená konfigurace se 768 GRU jednotkami, která byla použita i v [41].

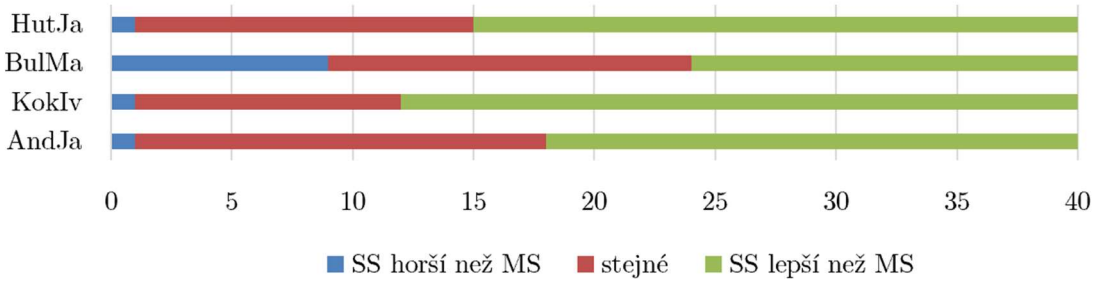
Po natrénování modelů byly vygenerovány testovací nahrávky, které byly porovnány v poslechovém testu CCR (viz kapitola 2.4.1). Každý hlas obsahoval 10 nahrávek vygenerovaných jak společným modelem, tak modelem pro daný hlas. Posluchači mohli pro každou nahrávku zvolit jednu z následujících odpovědí:

- SS (single-speaker model pro jeden hlas) zní lépe,
- MS (multi-speaker model) zní lépe,
- obě nahrávky znějí stejně.

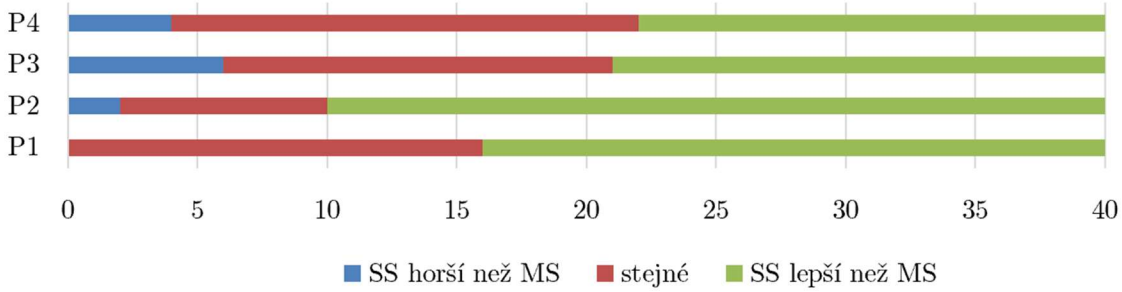
11.2 Výsledky

	Počty odpovědí			Procentuální poměr [%]		
	MS lepší	stejné	SS lepší	MS lepší	stejné	SS lepší
Jednotlivé hlasy						
AndJa	1	17	22	2,5	42,5	55
KokIv	1	11	28	2,5	27,5	70
BulMa	9	15	16	22,5	37,5	40
HutJa	1	14	25	2.5	35	62,5
Jednotliví posluchači						
P1	0	16	24	0	40	60
P2	2	8	30	5,0	20	75
P3	6	15	19	15	37,5	47,5
P4	4	18	18	10	45	45
Celkem	**12**	**57**	**91**	**7,5**	**35,62**	**56,88**

Tabulka 15: Výsledky poslechového testu porovnávající multi-speaker (MS) a single-speaker (SS) modely pro WaveRNN syntézu.



Obrázek 53: Výsledky poslechového testu. Odpovědi rozdělené pro jednotlivé hlasy.



Obrázek 54: Výsledky poslechového testu. Odpovědi rozdělené pro jednotlivé posluchače.

11.3 Shrnutí experimentu

Výsledky poslechového testu (tabulka 15, obrázek 53 a obrázek 54) jasně ukázaly, že přístup, kde každý hlas má vlastní WaveRNN model, je lepší. Z analýzy nahrávek se zdá, že společný model trpí poklesem kvality a zároveň v některých segmentech dochází k zamíchání barvy hlasu. Individuální modely těmito problémy netrpěly a zároveň méně kvalitní hlasy si zachovaly dobrou úroveň.

Hypotéza, že kvalitnější hlasy mohou povznést kvalitu méně kvalitního hlasu, se nepotvrdila. To je zajímavé, neboť v experimentech s trénováním LSTM parametrické syntézy to bylo přesně naopak, to znamená, že trénování společného modelu (viz například kapitola 10) zvýšilo kvalitu u malých či méně kvalitních hlasů. Vypovídá to o tom, že WaveRNN je citlivější na kvalitu a míchání nesourodých dat a vyžaduje jejich větší konzistenci.

Kapitola 12

Další metody syntézy řeči

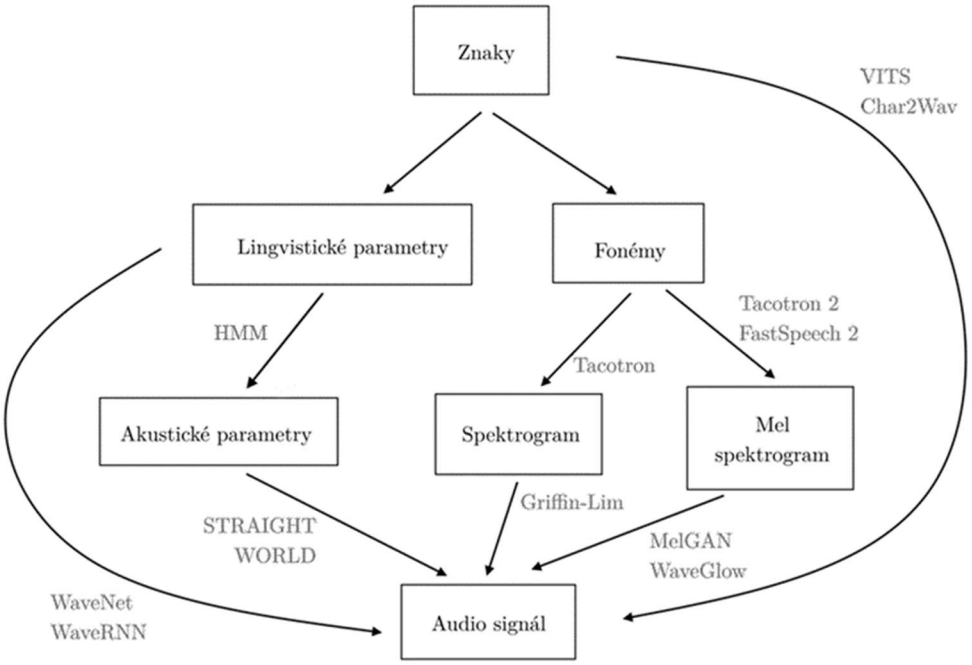
V oblasti syntézy řeči dochází v posledních letech k dynamickému vývoji. Téměř každý měsíc se objevují nové a nové architektury, které posouvají laťku kvality a přirozenosti stále výše. Tato kapitola byla zařazena netradičně na konec práce právě proto, aby ilustrovala, jak moc se v čase psaní práce změnilo prostředí a o jak velký skok se posunula oblast výzkumu syntézy řeči s příchodem nových architektur neuronových sítí.

Nově představené architektury se převážně dělí na dvě skupiny: akustický model a neurální vokodér. Akustický model je síť, jejímž vstupem je vstupní text, ať už ve formě grafému či fonému, a generuje akustickou reprezentaci. Druhá skupina modelů převádí tuto reprezentaci do výsledného řečového signálu. Zde existuje analogie s tradičními vokodéry, které tuto úlohu plnily dříve – proto se modelům z této kategorie říká neurální vokodéry.

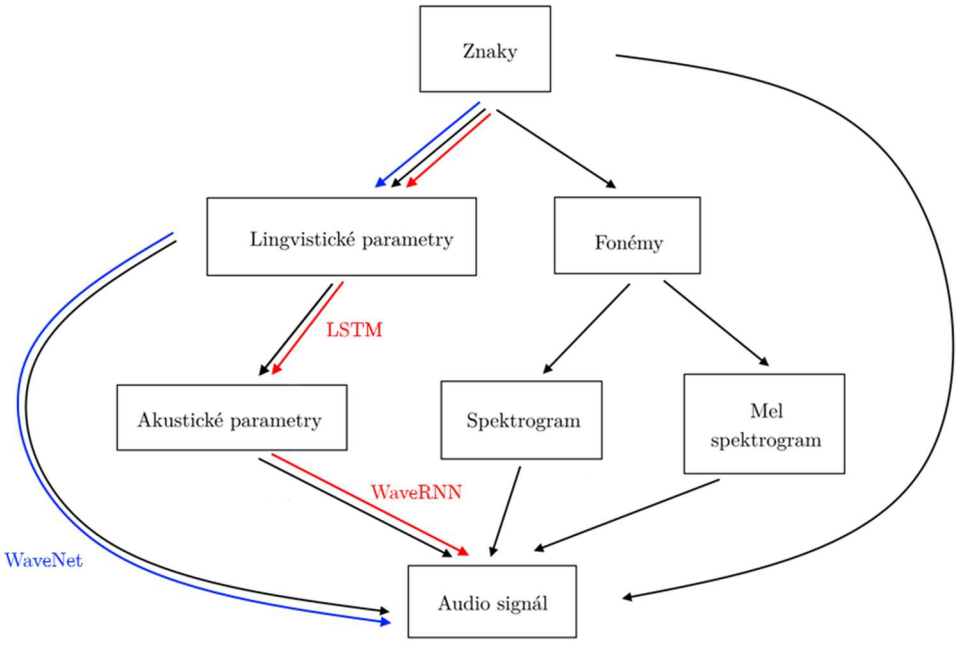
Obrázek 55 a obrázek 56 ukazuje schéma toku dat při generování syntézy řeči v neuronových sítích. Akustické modely obvykle generují výstup jako spektrogram, melovský spektrogram či dříve používané akustické parametry. Vokodéry jsou stavěné (trénované) vždy na jeden druh vstupu a ten se nedá zaměňovat.

12.1 End-to-end

Trendem poslední doby je, kromě vyšší přirozenosti a kvality, také schopnost sítě se natrénovat *end-to-end,* tedy vše najednou. Dříve byla úloha syntézy řeči složena z jednotlivých komponent (modelů), které se specializovaly na určitou část a byly trénovány zvlášť. Příkladem je architektura popsaná v kapitole 9. Predikci hlasivkové frekvence, model trvání, model akustických parametrů, vokodér – to vše byl vlastní model trénovaný zvlášť. Architektury představené později se snaží spojit vše do jediného modelu.

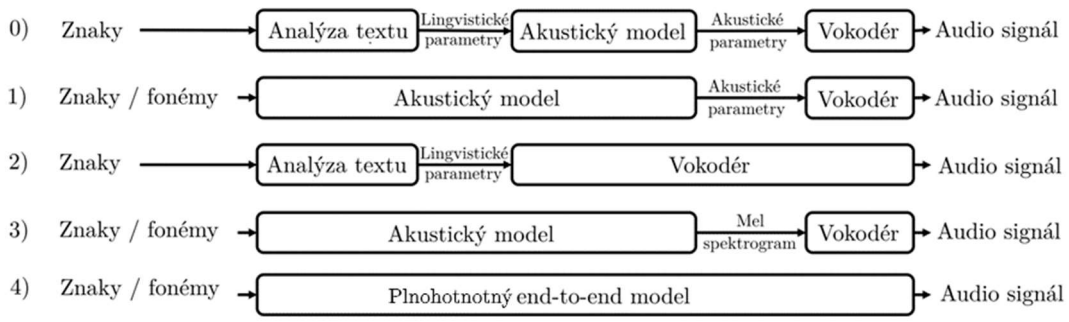


Obrázek 55: Jednotlivé mezikroky v syntéze řeči tak, jak jsou používány v posledních letech. U jednotlivých větví jsou šedě napsané příklady architektur, které danou úlohu řeší.



Obrázek 56: Mezikroky systému TTS navrženého v této práci. Modře je popsaný systém založený na architektuře WaveNet (kapitola 7 a 8). Červeně je znázorněn systém používající architekturu WaveRNN popsaný a používaný v kapitole 9, 10 a 11.

Termín *end-to-end* je velmi vágní, neboť každý si ho vykládá jinak. Dříve byla tímto termínem označována například architektura Tacotron (viz dále), která však generuje melovské spektrogramy. Z těch je nutné vygenerovat audio – například pomocí architektury WaveNet. Některé end-to-end sítě mají jako vstup text (grafémy), jiné fonémy. I zde je prostor pro spor. Úloha analýzy a normalizace textu by se také dala zařadit do požadavků pro úplný systém. Pro rozlišení významu se dokonce někdy používá „end-to-end“, „víc end-to-end“ a „plnohodnotný end-to-end“ [55]. Obrázek 57 naznačuje, jak se postupně spojovaly úlohy do větších celků ve snaze sestavit plnohodnotný end-to-end systém, který pokryje celou úlohu syntézy řeči.



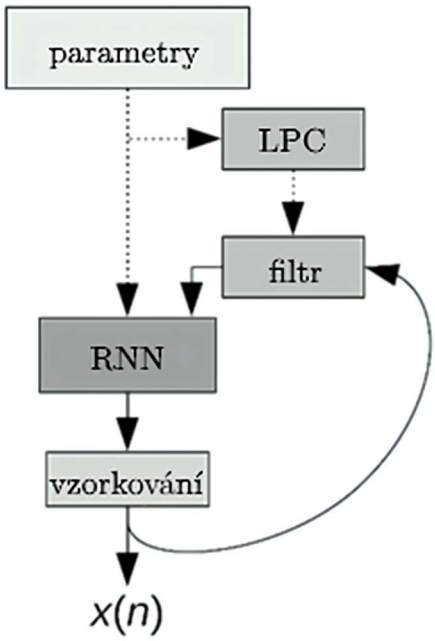
Obrázek 57: Vývoj architektur syntézy řeči směrem k end-to-end modelům.

Další odstavce popisují několik reprezentantů nově představených architektur. Snaha byla vybrat zástupce co nejrozdílnější, používající jiné přístupy (autoregresivní, GAN, VAE, …). Poslední část obsahuje reference na další architektury.

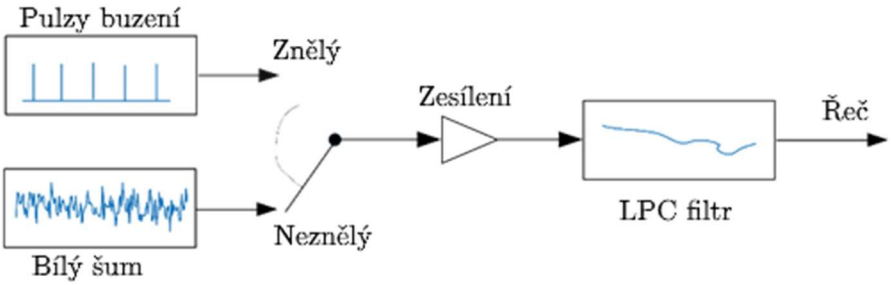
12.2 LPCNet

LPCNet[56] je varianta WaveRNN, která se zaměřuje na snížení výpočetních nároků při generování audio signálu. Oproti WaveRNN obsahuje navíc LPC predikční filtr, který předpovídá odhad následující hodnoty vzorku (obrázek 58), samotná neuronová síť pak predikuje rozdíl (reziduum). Díky tomu je RNN vrstva menší a implementace LPCNet je schopná pracovat v reálném čase, a to i na procesorech zařízení, které mají menší výkon, jako je například mobilní telefon či tablet.

LPCNet model generuje vzorky stejným způsobem jako WaveRNN. Rozdíl je ale takový, že síť může využít pomocný příznak – predikci následujícího vzorku pomocí LPC filtru, jehož koeficienty jsou průběžně odhadovány z předchozích vygenerovaných vzorků.



Obrázek 58: Architektura sítě LPCNet.

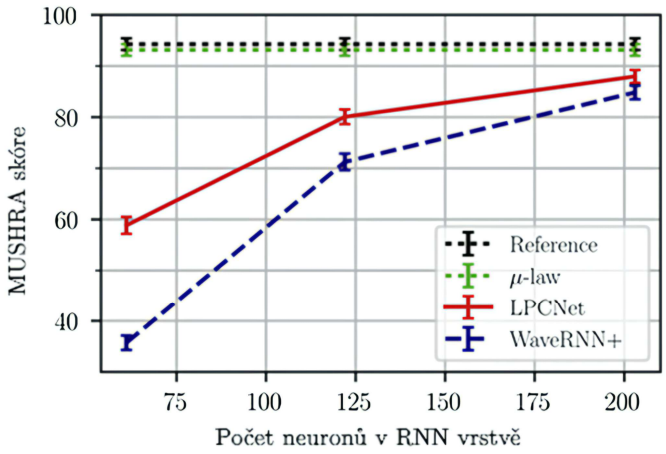


Obrázek 59: Princip fungování LPC filtru.

LPC filtr (obrázek 59) se používal v počátcích syntézy řeči. Jedná se o klasický přístup zdroj a filtr [30]. V tomto případě funguje filtr pro velmi rychlý a nepřesný odhad následujícího vzorku. To zjednodušuje práci WaveRNN modelu, který se vlastně učí pouze rozdílový model skutečnosti a LPC predikce. Díky tomu je možné použít mnohem menší počet neuronů v GRU rekurentní síti a získat tak mnohem rychlejší syntézu.

Obrázek 60 ukazuje rozdíl kvality LPCNetu a WaveRNN. Je zde jasně vidět, že největší přínos je při použití velmi malého modelu. Čím je větší počet neuronů, tím je menší rozdíl v kvalitě. Právě v této oblasti LPCNet exceluje. Při použití více jak 200 neuronů ztrácí LPC predikce smysl.

Hlavní cíl LPCNetu není ani tak v syntéze řeči, ale ve zpracování signálu. Autoři popisují možnost, jak použít model jako součást audio kodeku s velmi malým datovým tokem (několik kb/s). Model také najde využití v časové modifikaci řečového signálu, potlačení šumu a jako výplň audia při internetovém hovoru při ztrátě datového paketu.



Obrázek 60: Porovnání kvality se sítí WaveRNN. Adaptováno z [56].

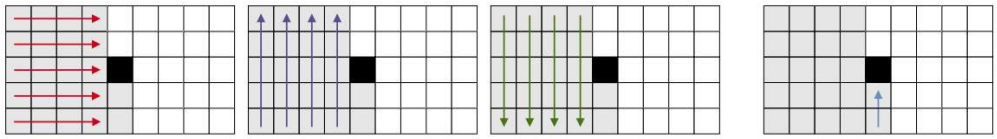
12.3 MelNet

Architektura MelNet [57] využívá autoregresivní přístup ke generování vysoce kvalitních melovských spektrogramů (na rozdíl od audio signálu v případě WaveRNN). Motivací pro použití spektrogramu je, že lépe modeluje dlouhodobé závislosti, neboť jsou informace reprezentovány kompaktněji ve 2D matici.

Princip fungování modelu blíže odpovídá tomu, jak člověk vnímá řeč – tj. jako frekvence v čase. V lidském uchu rovněž nejsou receptory, které vnímají zvuk jako signál v čase, ale naopak jako úrovně jednotlivých frekvencí v čase. Spektrogram tak má blíže k fungovaní lidského sluchu.

Architektura

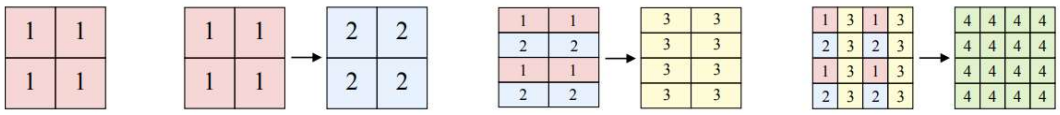
Model obsahuje GRU vrstvy, které autoregresivně generují jednotlivé položky spektrogramu od nižší frekvence k vyšší. Využívají k tomu stejného principu vzorkování z histogramu jako např. WaveNet. Tato část má název *frequency stack*. Vstupem každé vrstvy je jednak lokální podmínění, zároveň i výstup z dalších vrstev – *time delayed stack*. Ty tvoří tři GRU vrstvy, které zpracovávají předchozí hodnoty spektrogramu ve třech směrech: zleva doprava, zdola nahoru a shora dolů (viz obrázek 61).



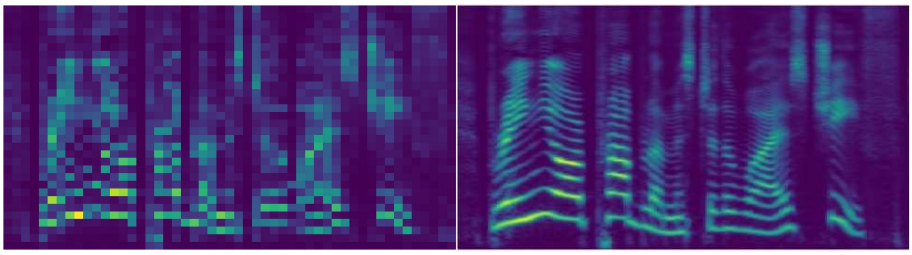
Obrázek 61: Směr GRU vrstev v modelu MelNet. První tři části jsou time delayed stack s různým směrem zpracování a poslední je frequency stack, který pracuje v rámci jednoho sloupce od nižší frekvence k vyšší.

Multi-scale modelling

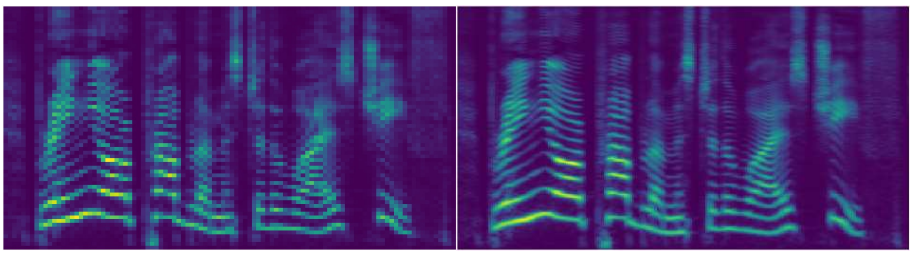
Další trik, na kterém staví MelNet, je tzv. *multi-scale modelling*. Ten spočívá v iterativním generování stejného spektrogramu s postupně vyšší přesností (obrázek 62, obrázek 63 a obrázek 64). V prvních iteracích je spektrogram vygenerován s menším rozlišením jak ve frekvenční, tak v časové ose. To umožňuje modelu soustředit se na dlouhodobé jevy a netrápit se s krátkodobými detaily. V dalších iteracích je k dispozici kostra spektrogramu, a to i do budoucnosti – předchozí výstup je totiž vstupem následující iterace.



Obrázek 62: Multi-scale modelling.



Obrázek 63: První a druhá iterace MelNetu. Je vytvořena kostra spektrogramu bez konkrétních detailů. Tím, že je sníženo rozlišení, je pro model jednodušší generovat dlouhodobé závislosti.



Obrázek 64: Třetí a čtvrtá iterace MelNetu. Model má k dispozici kostru spektrogramu, a tak se může soustředit na vyvářeení detailů spektrogramu.

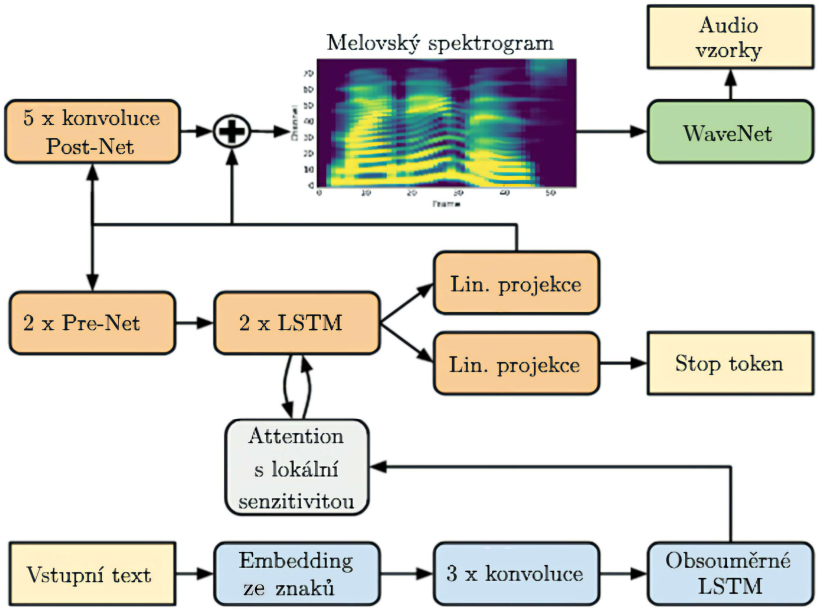
12.4 Tacotron

Architektura *Tacotron* je generativní model pro převod textu na řeč. První verze [58] pochází z roku 2017 a byla představena výzkumníky ze společnosti Google. Tacotron je velmi oblíbený a často se používá jako benchmark. Dnes již existuje mnoho variant, které různě upravují jeho architekturu.

V jádru Tacotronu (obrázek 65) se nachází *sequence-to-sequence* model pro generaci melovských spektrogramů. Tyto modely, známé také jako *seq2seq*, jsou používány tam, kde se délka vstupní sekvence liší od délky výstupní sekvence. U Tacotronu je délka vstupního textu kratší než časové úseky spektrogramu. Tyto modely obecně pracují s enkodérem a dekodérem. Enkodér prochází celý vstup a ukládá informace do stavového vektoru pro dekodér. Dekodér pak použije informace uložené ve stavovém vektoru a postupně vygeneruje výstupní sekvenci.

Tacotron obsahuje model architektury WaveNet, který se stará o finální převedení z melovského spektrogramu (viz kapitola 2.6) do audio signálu, tj. vykonává funkci neurálního vokodéru.

Model pro generování je složen z rekurentních vrstev LSTM a vrstev, které plní funkci tzv. *attention*. Tyto vrstvy pomáhají modelu určit, jakou váhu má každý vstupní grafém při generování jednotlivých snímků spektrogramu. Vrstvy attention tak slouží jako náhrada segmentace trénovacích dat. Díky tomu je velmi snadné trénovat Tacotron na velkých datech, neboť vše, co je pro trénování potřeba, jsou páry <text, audio>.



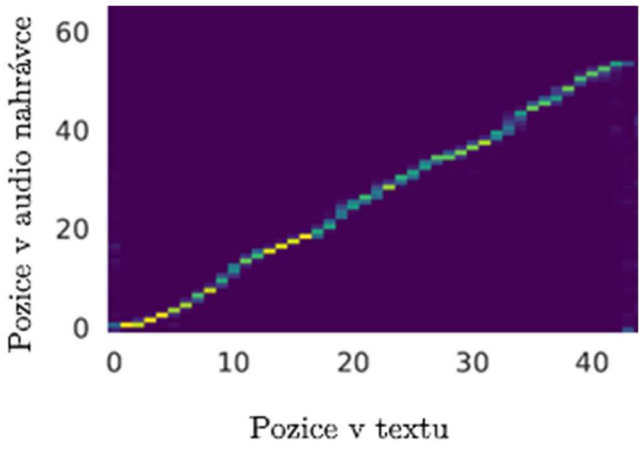
Obrázek 65: Architektura sítě Tacotron.

Vstupními daty mohou být buď grafémy, nebo fonémy. V případě grafémů je vstup většinou z normalizovaného textu. Vstupní data projdou nejprve konvolučními a rekurentními sítěmi k předzpracování informací uložených v sekvenci grafémů. Konvoluce a rekurence umožňuje vrstvám zobrazit i okolní grafémy, což znamená, že předzpracování může využívat kontextové závislosti.

Vstupem LSTM vrstev uvnitř *Tacotronu* je výstup attention vrstev, zároveň však mají přístup i k hodnotě výstupu celého hlavního bloku v předchozí iteraci (framu). Zpětnou smyčku zajišťují vrstvy *Pre-Net* doplněné o *dropout*, který je nutný, aby správně fungoval generativní proces. Jinak by se síť velmi rychle přetrénovala a naučila by se brát příliš mnoho informací z předchozího výstupu.

Prozodický projev Tacotronu je lepší než například u tradiční *LSTM* syntézy. Je to dáno tím, že řeč je generována najednou v jednom průchodu. V parametrické syntéze je totiž nejdřív provedena predikce trvání a až potom je řeč generována s pevně daným rozložením jednotlivých fonémů.

Síť Tacotron obsahuje *attention* vrstvy, které pro každý nově generovaný *frame* určují oblast zájmu ve vstupním textu. Síť se tak sama rozhoduje, kolik framů přidělí jednotlivým grafémům. Má tak plnou kontrolu nad procesem generování řeči. Výstup *attention* vrstev se postupně posouvá v textu zleva doprava (viz obrázek 66). Pro zastavení generování (konec věty) je uvnitř modelu zabudován jednoduchý binární klasifikátor, který generuje pravděpodobnost tzv. stop symbolu. Tím, že model *Tacotron* sám implicitně určuje, jak dlouho jsou grafémy generovány (jsou v oblasti zájmu), má mnohem přirozenější prozodii.



Obrázek 66: Ukázka výstupu attention vrstev.

12.5 MelGAN

MelGAN [59] je neuronová síť pro generování syntetické řeči postavená na architektuře GAN (*Generative adversarial network*) [60]. Tato architektura je často používaná v úlohách zpracování obrazu, avšak v oblasti syntézy řeči byla úspěšně aplikovaná až nedávno.

GAN je generativní model, který při trénování trénuje zároveň dva modely: generátor a diskriminátor. Role generátoru spočívá v generování realizace dat na základě vstupního podmínění a náhodného šumu. Rolí diskriminátoru je rozpoznat generovaná data od skutečných – správně klasifikovat, jestli na jeho vstupu je vzorek z trénovacích dat, anebo vzorek vygenerovaný generátorem.

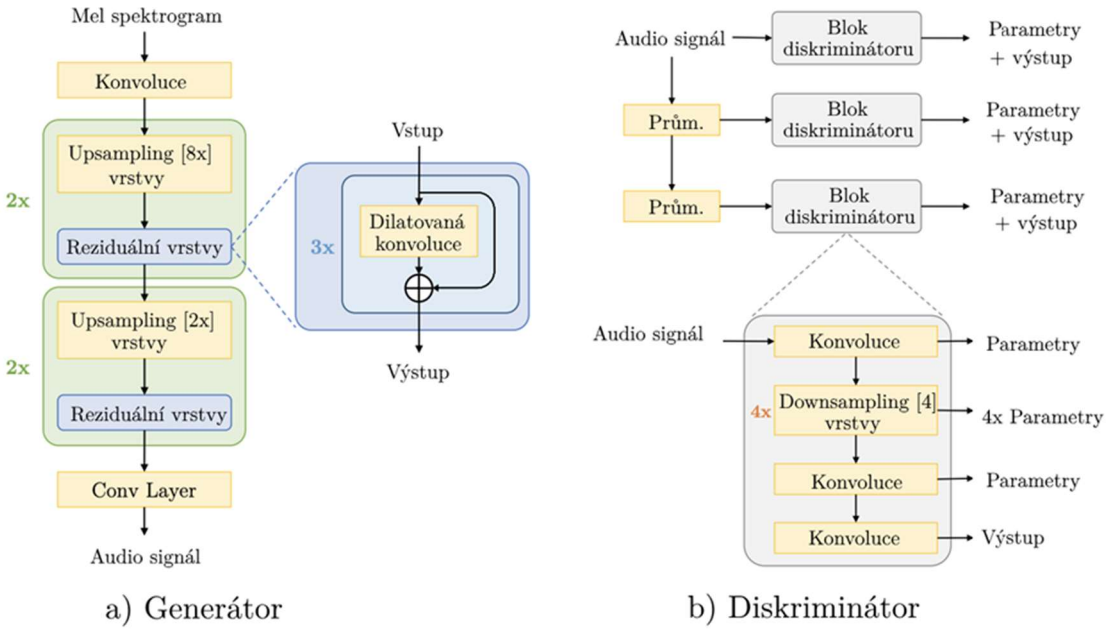
Při trénování generátoru se maximalizuje pravděpodobnost, že diskriminátor udělal chybu (tj. považoval generovanou syntetickou řeč za reálnou). To odpovídá úloze *minmax* z teorie her. Tento trik tak spočívá v tom, že se generátor neučí přímo z trénovacích dat, ale pouze se snaží „přehrát“ diskriminátor. Tím, jak se generátor zlepšuje, produkuje věrohodnější vzorky vygenerovaných dat a je tak pro diskriminátor stále těžší rozpoznat syntetizované vzorky od pravých. To nutí diskriminátor se stále zlepšovat a získávat další data z trénovacích dat, které se generátor posléze může z něho naučit.

MelGAN není autoregresivní síť, výstupní vzorky generuje najednou pomocí dopředných konvolučních vrstev. To s sebou nese řádově vyšší rychlost generování než v případě autoregresních architektur, které generují vzorky po jednom.

Velmi často se používá verze multi-band MelGAN [59], který vylepšuje originální MelGAN ve více oblastech. Nejdůležitější změnou je, že generátor generuje signál ve více frekvenčních pásmech (*bands*), které jsou následně sečteny pro vytvoření plného výstupu. To značně snížilo výpočetní složitost modelu a umožnilo rychlé a paralelní generování řeči i na počítači bez grafického akcelerátoru.

Obrázek 67 popisuje architekturu MelGAN. Síť se skládá z generátoru a diskriminátoru. Podobně jako například WaveNet, i MelGAN využívá reziduální spojení a dilatované konvoluční vrstvy. Architektura obsahuje převzorkovací vrstvy pro změnu frekvence signálu pro tzv. *multi-scale diskriminátor* (MSD). Ten tvoří 3 bloky konvolucí. První má jako vstup samotný audio signál, druhý a třetí má jako vstup 2krát, respektive 4krát zprůměrovaný a podvzorkovaný audio signál. Každý blok tak vidí signál v jiném měřítku.

Ve fázi generování (inference) se využívá pouze generátor. Vstupem sítě je melovský spektrogram a výstupem řečový audio signál. MelGAN se dá tedy klasifikovat jako neurální vokodér.

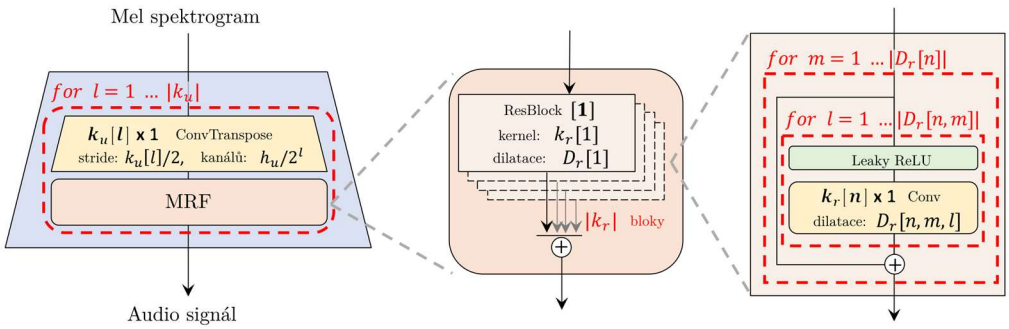


Obrázek 67: Architektura MelGAN. Struktura schématu převzata z [59].

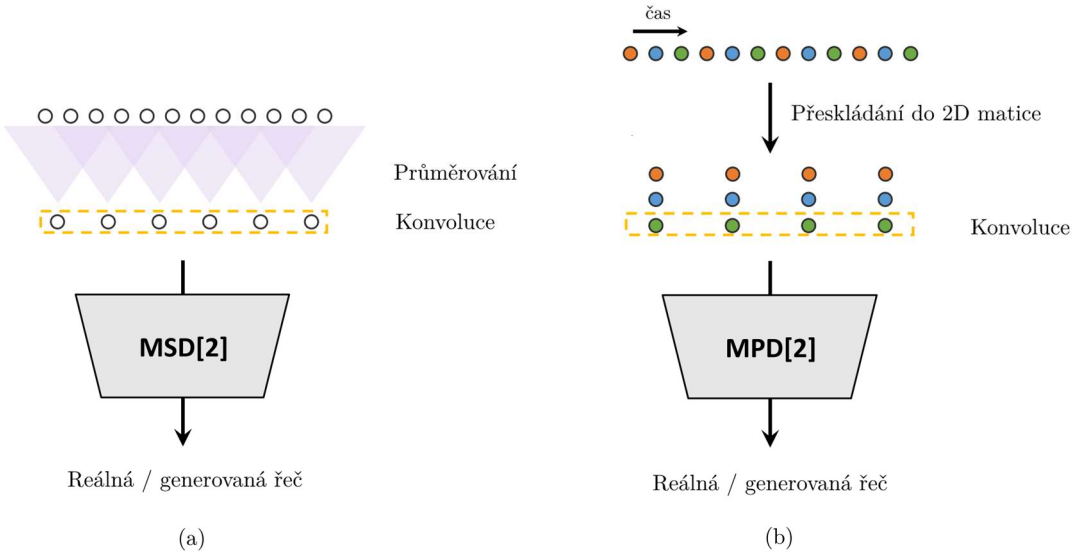
12.6 HiFi-GAN

Tato architektura byla představena v roce 2020 v [61]. Navazuje na úspěchy architektury MelGAN. Ta přinesla rychlé generování, ale nedokázala se v kvalitě vyrovnat autoregresivním modelům jako je WaveNet nebo WaveRNN. Cílem autorů to bylo změnit.

Tento model generuje řečový signál ze vstupních melovských spektrogramů. Jedná se tedy o neurální vokodér. Generátor (obrázek 68) tvoří poskládané transponované konvoluční vrstvy s různým nastavením velikosti kernelu a dilatace, které postupně zvyšují frekvenci melovského spektrogramu až do vzorkovací úrovně řečového signálu. I zde byl použit trik pro urychlení konvergence v podobě reziduálních spojů.



Obrázek 68: Generátor architektury HiFi-GAN. Struktura schématu převzata z [61].



Obrázek 69: Diskriminátor architektury HiFi-GAN. Při trénování je použito více diskriminátorů: multi-scale (a) a multi-period (b). Struktura schématu převzata z [61].

Jako první diskriminátor použili autoři multi-scale diskriminátor (MSD), který je použit i v architektuře MelGAN. Dále se autoři zaměřili na fakt, že řečový signál obsahuje harmonické složky s různou frekvencí, tedy i periodou. Proto přidali do architektury druhý diskriminátor, který pojmenovali *multi-period* diskriminátor (MPD). Ten obsahuje mnoho podmodulů, kde každý podmodul pracuje s jinou periodou. Perioda (p) v tomto případě znamená, že se z výstupního audio signálu vezme jeden vzorek a dalších (p–1) se vynechá, poté se proces opakuje. Autoři zvolili hodnoty period jako prvočísla (2, 3, 5, 7, 11), aby minimalizovali jejich překryv. Obrázek 69 popisuje diskriminátory této architektury.

V poslechovém testu dosáhla architektura HiFi-GAN lepší MOS skóre než architektura MelGAN, a dokonce i víc než WaveNet. Ten porazila i menší konfigurace této architektury, která má jen 1 milión parametrů. Tato konfigurace dokáže na procesoru počítače generovat řeč 13x rychleji, než je reálný čas. Při použití grafického akcelerátoru je rychlost generování tisíckrát rychlejší.

12.7 VITS

Architektura VITS (Variational Inference with adversarial learning for end-to-end Text-to-Speech) je jedna z mála „skutečných“ end-to-end architektur pro generování syntetické řeči vysoké kvality. V ostatních modelech bývá úloha rozdělená na dvě části – akustický model a vokodér (viz výše), ty se trénují odděleně. VITS model toto rozdělení nemá podobně jako FastSpeech 2s. Oproti němu však dosahuje vyšší kvality generované řeči.

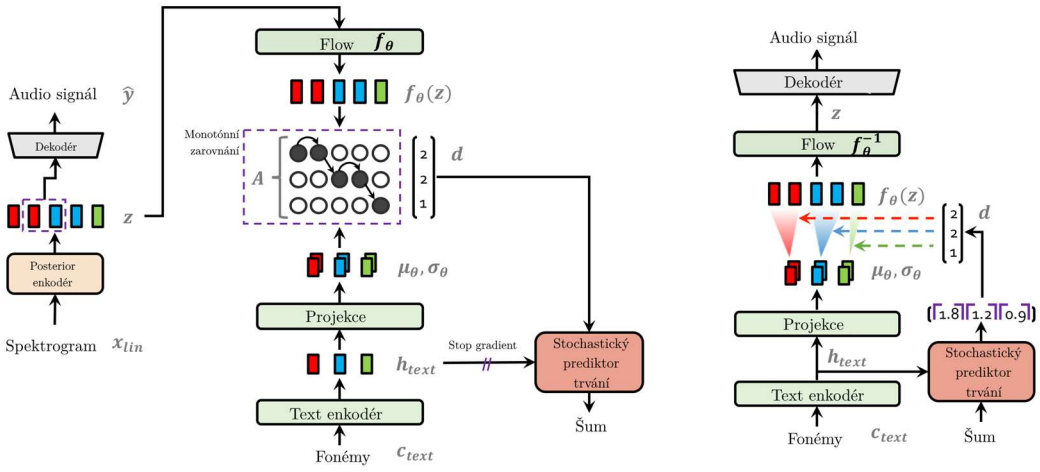
Architektura VITS je *conditional variational autoencoder* (VAE). Tento specifický typ neuronové sítě používá enkodér a dekodér pro projekci dat do vektoru nižší dimenze.

Rozdílem oproti autoenkodéru (AE) je, že VAE nejen rekonstruuje původní data, ale také predikuje pravděpodobnostní rozdělení vektoru. Z těchto pravděpodobností se následně vybírají generované hodnoty. Tyto hodnoty poté putují zpět přes dekodér a jsou použity k vytvoření rekonstruovaného výstupu. Modely VAE (VITS) většinou maximalizují kritérium *evidence lower bound* (ELBO) [62].

Obrázek 70 ukazuje architekturu neuronové sítě VITS. Fáze trénování je komplikovanější. V této fázi je vstupem sekvence fonémů a spektrogram trénovací nahrávky (ten lze vypočítat za běhu z audio signálu). Fáze inference je mnohem jednodušší. Vstupem je pouze sekvence fonémů a pomocný vektor náhodných dat, který je použit ve stochastickém prediktoru trvání.

Jak již bylo zmíněno, termín end-to-end je v úloze syntézy řeči vágní. Tento model má v původním článku [63] jako vstup fonémy. To znamená, že je přeci jen nutné provést převedení textu do fonémů (*g2p*). Výhodou je, že je možné snadno „vnutit“ vlastní trvání pro každý foném zvlášť a řeč tak zrychlovat či zpomalovat.

Pro zvýšení schopnosti generovat velmi komplexní vztahy uvnitř trénovacích dat použili autoři techniku *normalizing flows* [64]. V ní jsou při procesu trénování vstupní data „ničena“ sérií invertovatelných matematických operací, které ze signálu odebírají informaci a přibližují data šumu. V inferenci proces probíhá opačným směrem a ze vstupu (šumu) se generuje náhodná realizace dat. Vzhledem k tomu, že operace musí být invertovatelné, nemohou být nelineární (např. aktivační funkce) – schopnosti *normalizing flows* jsou proto omezené. Nicméně pro funkci VITS jsou zásadní. Autoři provedli experiment, kde síť natrénovali bez této části a propad v kvalitě byl značný (4,5 -> 2,98 v MOS).



a) Fáze trénování b) Fáze inference

Obrázek 70: Architektura VITS. Struktura schématu převzata z [63].

Lidská řeč je velmi variabilní, to znamená, že stejná věta může být vyslovena mnoha různými způsoby. Stejný vstup má proto mnoho realizací (*one-to-many*). Pro zvýšení expresivnosti tak autoři navrhli stochastický prediktor trvání. Vygenerované věty jsou proto přirozenější, protože mají větší variabilitu v trvání jednotlivých fonémů. I v této části modelu bylo použito *normalizing flows* pro zvětšení variability výstupu.

Dekodér je realizován stejně jako generátor v síti HiFi-GAN [61]. Je však zakomponován uvnitř modelu, tj. trénování těchto vrstev probíhá současně se zbytkem modelu). Dilatované konvoluce tvoří podstatnou část dekodéru, což umožňuje velmi rychlou inferenci.

VITS ke svému trénování proto nepotřebuje žádné zarovnání vstupního textu na výstupní audio signál. Síť se dokáže schopnost zarovnat data naučit sama pomocí algoritmu MAS (*monotonic alignment search*) [65]. Na základě vlastního zarovnání si síť v průběhu učení trénuje prediktor trvání. Pro zarovnání jsou použity spektrogramy. Autoři provedli pokus s melovským spektrogramem, ale zaznamenali pokles kvality (4,5 -> 4,31 MOS skóre).

Tato síť dokáže generovat vzorky audio signálu paralelně. Má proto velmi rychlé generování. Tomu pomáhá i fakt, že síť je jeden model. Autoři zmiňují rychlost generování až 67krát rychlejší, než je reálný čas v konfiguraci, která dosahuje nejvyšší kvality.

V poslechových testech v [63] dokázal VITS porazit jak Tacotron 2, tak i Glow-TTS [65], který byl předlohou akustického modelu pro VITS. Oba soupeřící modely používaly HiFi-GAN jako vokodér pro převedení vygenerovaných spektrogramů do audio signálu. MOS skóre v poslechových testech bylo dokonce na stejné úrovni jako byla přirozená řeč, což je vynikající výsledek.

12.8 Další architektury

Architektur pro syntézu řeči je v dnešní době velmi mnoho a téměř každý měsíc přibývají další. Zde je příklad dalších architektur, které dosud nebyly zmíněny. Popis těchto architektur lze nalézt ve výborném článku [55].

Vokodéry

- Parallel WaveNet [42]
- WaveGrad [66]
- WaveGlow [67]

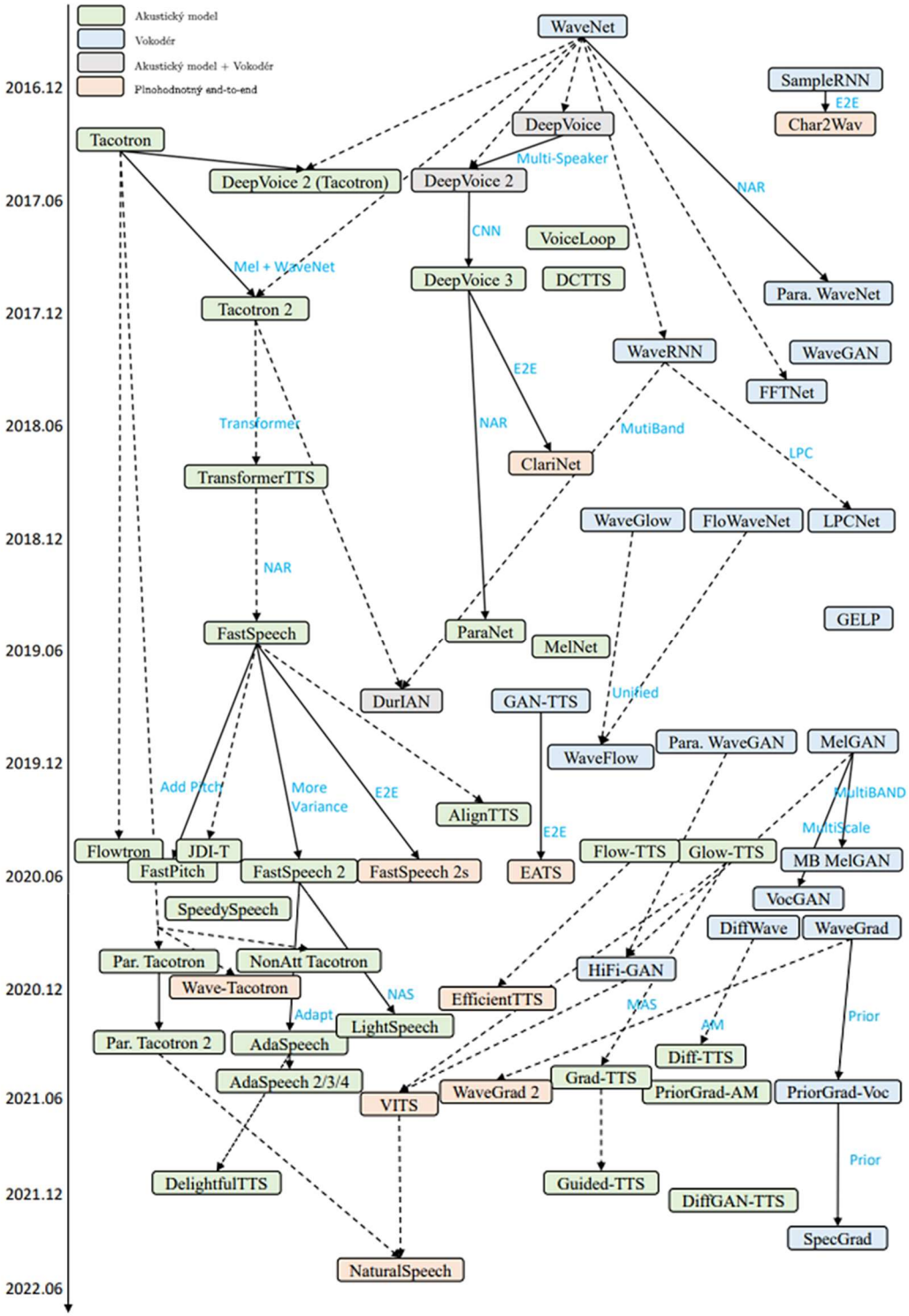
Akustické modely

- FastSpeech [68]
- FastSpeech 2 [69]
- Deep Voice [70]
- Deep Voice 2 [71]

End-to-end modely

- Clarinet [72]
- Char2Wav [73]
- NaturalSpeech [74]

Obrázek 71 ukazuje, jak se jednotlivé architektury a koncepty vyvíjely v čase. Zároveň je patrné, jak plodné je období posledních let na nové architektury. Na počátku stojí architektura WaveNet, která způsobila revoluci a masivní vývoj syntézy řeči pomocí neuronových sítí.



Obrázek 71: Evoluce modelů pro syntézu řeči pomocí neuronových sítí. První neuronová síť schopná generovat kvalitní řeč byla architektura WaveNet. Obrázek převzat z [55].

Kapitola 13

Závěr

Hlavním tématem disertační práce byly nové architektury pro syntézu řeči. Teoretická část práce obsahuje nejdříve shrnutí základních pojmů, tradičních metod a pak těch nových, založených na neuronových sítích s důrazem na síť WaveNet a WaveRNN. Jejich objev způsobil ohromný skok v kvalitě výstupu umělé syntézy řeči. Práce rovněž popisuje ostatní známé architektury jako je *LPCNet*, MelNet, *Tacotron, MelGAN, VITS* a další.

Praktická část obsahuje experimenty s generováním české řeči pomocí neuronových sítí. První experiment obnášel vlastní implementaci sítě WaveNet a její použití pro syntézu češtiny. Dále pak následovala série experimentů, které zkoumaly, jak moc, v jaké kvalitě a jak přesně anotovaná musí být trénovací data pro úspěšné natrénování syntézy.

Práce zároveň popisuje návrh nového systému TTS pro syntézu češtiny, který byl založen na síti WaveRNN a třech LSTM modelech (akustickém, modelu prozodie a trvání). Je zde rovněž poslechový test, ve kterém tento systém dosáhl lepších výsledků než v té době používaný systém na katedře kybernetiky FAV ZČU ARTIC [2] založený na tradiční metodě unit selection.

Navržený systém byl použit na experimenty s vícejazyčnou syntézou a multi-speaker trénováním, které jsou součástí práce. Experiment ukázal, že lze systém natrénovat na datech z více jazyků tak, že hlas řečníka může mluvit jiným jazykem, než ve kterém byl namluven. Zároveň je možné použít jeden velký model pro syntézu velkého množství hlasů.

Vývoj v oblasti umělé inteligence je velmi rychlý, a to i v oblasti syntézy řeči. Představované architektury sítí pro syntézu řeči zastarávají velmi rychle a málokterá si udrží pozornost po dobu více než několika let. Sítě WaveNet a WaveRNN jsou překvapivě výjimkou, neboť i dnes jsou stále používány v komerčním světě a citované v článcích. To potvrzuje, že rozhodnutí (které bylo učiněno při volbě tématu práce) soustředit se především na tyto dvě architektury pro nový model TTS bylo správné.

Je však pravděpodobné, že i tyto architektury budou nahrazeny. Není jasné, jestli to budou sítě typu *RNN*, *GAN* či *Transformer,* či nějaká úplně jiná, dosud neznámá. Může to být i znovuobjevení aktuálně známé architektury s přidanou modifikací, nebo jen natrénovanou na více datech či výkonnějších počítačích.

Reference

[1] J. Psutka, L. Müller, J. Matoušek a V. Radová, *Mluvíme s počítačem česky*. Praha: Academia, 2006.

[2] D. Tihelka, Z. Hanzlíček, M. Jůzová, J. Vít, J. Matoušek a M. Grůber, „Current State of Text-to-Speech System ARTIC: A Decade of Research on the Field of Speech Technologies", in *Text, Speech, and Dialogue*, 2018, s. 369–378.

[3] J. R. Novak, N. Minematsu a K. Hirose, „WFST-based Grapheme-to-Phoneme conversion: Open source tools for alignment, model-building and decoding", in *FSMNLP 2012 - Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, 2012.

[4] M. Jůzová, D. Tihelka a J. Vít, „Unified language-independent DNN-based G2P converter", in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019. doi: 10.21437/Interspeech.2019-2335.

[5] S. Yolchuyeva, G. Németh a B. Gyires-Tóth, „Transformer based Grapheme-to-phoneme conversion", in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019. doi: 10.21437/Interspeech.2019-1954.

[6] R. F. Kubichek, „Mel-Cepstral distance measure for objective speech quality assessment", in *IEEE Pac Rim Conf Commun Comput Signal Process*, 1993. doi: 10.1109/pacrim.1993.407206.

[7] J. G. Beerends, A. P. Hekstra, A. W. Rix a M. P. Hollier, „Perceptual evaluation of speech quality (PESQ): The new ITU standard for end-to-end speech quality assessment. Part II - Psychoacoustic model", *AES: Journal of the Audio Engineering Society*, roč. 50, č. 10, s. 765–778, říj. 2002.

[8] E. Salesky, J. Mader a S. Klinger, „Assessing Evaluation Metrics for Speech-to-Speech Translation", in *2021 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021 - Proceedings*, 2021. doi: 10.1109/ASRU51503.2021.9688073.

[9] R. Likert, „A technique for the measurement of attitudes", *Archives of Psychology*, roč. 140, 1932.

[10] ITU-T, „Mean Opinion Score (MOS) terminology", *SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS*, 2006.

[11] J. Liebetrau *et al.*, „Revision of rec. ITU-R BS.1534", in *137th Audio Engineering Society Convention 2014*, 2014.

[12] M. R. P. Thomas, J. Gudnason, P. A. Naylor, B. Geiser a P. Vary, „Voice source estimation for artificial bandwidth extension of telephone speech", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2010. doi: 10.1109/ICASSP.2010.5495149.

[13] S. Zieliński, P. Hardisty, C. Hummersone a F. Rumsey, „Potential biases in MUSHRA listening tests", in *Audio Engineering Society - 123rd Audio Engineering Society Convention 2007*.

[14] X. Huang, A. Acero a H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm & System Development*. 2001.

[15] C. Benoît, M. Grice a V. Hazan, „The SUS test 1: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences", *Speech Commun*, č. 4, 1996, doi: 10.1016/0167-6393(96)00026-X.

[16] H. Zen *et al.*, „Libritts: A corpus derived from LibriSpeech for text-to-speech", in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019. doi: 10.21437/Interspeech.2019-2441.

[17] F. Charpentier a M. Stella, „Diphone synthesis using an overlap-add technique for speech waveforms concatenation", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1986, s. 2015–2018.

[18] G. D. Forney, „The Viterbi Algorithm", *Proceedings of the IEEE*, roč. 61, č. 3, 1973, doi: 10.1109/PROC.1973.9030.

[19] J. Vít, „Detecting artifacts in synthetic speech", in *Tackling the complexity in speech*, O. Niebuhr a R. Skarnitzl, Ed. 2015.

[20] D. Tihelka, J. Kala a J. Matoušek, „Enhancements of viterbi search for fast unit selection synthesis", in *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, 2010. doi: 10.21437/interspeech.2010-78.

[21] H. Zen, K. Tokuda a A. W. Black, „Statistical Parametric Speech Synthesis", *Speech Commun*, roč. 51, č. 11, s. 1039–1064, 2009.

[22] H. Kawahara, J. Estill a O. Fujimura, „Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT", in *International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA)*, 2001, s. 59–64.

[23] Y. Agiomyrgiannakis, „Vocaine the vocoder and applications in speech synthesis", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, s. 4230–4234.

[24] D. Erro, I. Sainz, E. Navas a I. Hernaez, „Harmonics Plus Noise Model Based Vocoder for Statistical Parametric Speech Synthesis", *IEEE J Sel Top Signal Process*, roč. 8, č. 2, s. 184–194, 2014, doi: 10.1109/JSTSP.2013.2283471.

[25] M. Morise, F. Yokomori a K. Ozawa, „WORLD: A vocoder-based high-quality speech synthesis system for real-time applications", *IEICE Trans Inf Syst*, roč. E99D, č. 7, s. 1877–1884, 2016, doi: 10.1587/transinf.2015EDP7457.

[26] T. Toda a K. Tokuda, „A speech parameter generation algorithm considering global variance for HMM-based speech synthesis", *IEICE Trans Inf Syst*, roč. E90-D, č. 5, 2007, doi: 10.1093/ietisy/e90-d.5.816.

[27] S. Hochreiter a J. Schmidhuber, „Long Short-Term Memory", *Neural Comput*, roč. 9, č. 8, s. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[28] K. Cho *et al.*, „Learning phrase representations using RNN encoder-decoder for statistical machine translation", *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, s. 1724–1734, 2014, doi: 10.3115/v1/d14-1179.

[29] M. Pouget, T. Hueber, G. Bailly a T. Baumann, „HMM training strategy for incremental speech synthesis", in *Interspeech*, 2015, s. 1201–1205. doi: 10.21437/Interspeech.2015-304.

[30] G. Fant, *Acoustic Theory of Speech Production*. 1971. doi: 10.1515/9783110873429.

[31] V. Nair a G. E. Hinton, „Rectified linear units improve Restricted Boltzmann machines", in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010.

[32] J. S. Bridle, „Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters", *Adv Neural Inf Process Syst*, č. Ml, 1990.

[33] S. Pincus a B. H. Singer, „Randomness and degrees of irregularity", *Proc Natl Acad Sci U S A*, roč. 93, č. 5, 1996, doi: 10.1073/pnas.93.5.2083.

[34] P. Cui, X. Wang, J. Pei a W. Zhu, „A Survey on Network Embedding", *IEEE Transactions on Knowledge and Data Engineering*, roč. 31, č. 5. 2019. doi: 10.1109/TKDE.2018.2849727.

[35] J. Schmidhuber, „Deep Learning in neural networks: An overview", *Neural Networks*, roč. 61. 2015. doi: 10.1016/j.neunet.2014.09.003.

[36] A. van den Oord *et al.*, *WaveNet: A Generative Model for Raw Audio*. 2016. doi: 10.1109/ICASSP.2009.4960364.

[37] K. Tokuda a H. Zen, „Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015. doi: 10.1109/ICASSP.2015.7178765.

[38] D. Lyon, „The μ-law CODEC", *Journal of Object Technology*, roč. 7, č. 8, s. 21–31, 2008, doi: 10.5381/jot.2008.7.8.c2.

[39] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves a K. Kavukcuoglu, „Conditional image generation with PixelCNN decoders", in *Advances in Neural Information Processing Systems*, 2016.

[40] K. He, X. Zhang, S. Ren a J. Sun, „Deep residual learning for image recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.90.

[41] N. Kalchbrenner *et al.*, „Efficient Neural Audio Synthesis", in *Proceedings of Machine Learning Research*, 2018, roč. 80, s. 2410–2419.

[42] A. van den Oord *et al.*, „Parallel WaveNet: Fast High-Fidelity Speech Synthesis", in *Proceedings of the 35th International Conference on Machine Learning*, 2018, s. 3918–3926.

[43] J. Vít, „Webový nástroj pro opravy anotací řečového inventáře", in *Studentská vědecká konference*, 2016, s. 127–128.

[44] Z. Hanzlíček, J. Vít a D. Tihelka, „LSTM-Based Speech Segmentation for TTS Synthesis", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-27947-9_31.

[45] D. Tihelka, M. Jůzová a J. Vít, „Grappling with Web Technologies: The Problems of Remote Speech Recording", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, roč. 12335 LNAI. doi: 10.1007/978-3-030-60276-5_57.

[46] Z. Hanzlíček, J. Vít a D. Tihelka, „WaveNet-Based Speech Synthesis Applied to Czech - A Comparison with the Traditional Synthesis Methods", in *Text, Speech and Dialogue*, 2018, s. 445–452.

[47] J. Vít, Z. Hanzlíček a J. Matoušek, „On the Analysis of Training Data for Wavenet-Based Speech Synthesis", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018. doi: 10.1109/ICASSP.2018.8461960.

[48] J. Matoušek, D. Tihelka a J. Romportl, „Current state of Czech text-to-speech system ARTIC", in *Text, Speech and Dialogue*, 2006, s. 439–446.

[49] Z. Hanzlíček, J. Vít a M. Řezáčková, „Speakers Talking Foreign Languages in a Multilingual TTS System", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021. doi: 10.1007/978-3-030-83527-9_42.

[50] Y. Fan, Y. Qian, F. K. Soong a L. He, „Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis", in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, s. 4475–4479.

[51] H. T. Luong, X. Wang, J. Yamagishi a N. Nishizawa, „Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora", *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* s. 1303–1307, 2019. doi: 10.21437/Interspeech.2019-1311.

[52] B. Li a H. Zen, „Multi-Language Multi-Speaker Acoustic Modeling for LSTM-RNN Based Statistical Parametric Speech Synthesis", 2016, s. 2468–2472. doi: 10.21437/Interspeech.2016-172.

[53] Y. Zhang *et al.*, „Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning", *CoRR*, roč. abs/1907.0, 2019, [Online]. Dostupné z: http://arxiv.org/abs/1907.04448

[54] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda a T. Toda, „An investigation of multi-speaker training for WaveNet vocoder", in *Proceedings of ASRU 2017*, 2017, s. 712–718.

[55] X. Tan, T. Qin, F. Soong a T.-Y. Liu, „A Survey on Neural Speech Synthesis". arXiv, 2021. doi: 10.48550/ARXIV.2106.15561.

[56] J. M. Valin a J. Skoglund, „LPCNET: Improving Neural Speech Synthesis through Linear Prediction", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019, s. 5891–5895. doi: 10.1109/ICASSP.2019.8682804.

[57] S. Vasquez a M. Lewis, „MelNet: A Generative Model for Audio in the Frequency Domain". 2019. [Online]. Dostupné z: http://arxiv.org/abs/1906.01083

[58] Y. Wang *et al.*, „Tacotron: Towards end-To-end speech synthesis", in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017, s. 4006–4010. doi: 10.21437/Interspeech.2017-1452.

[59] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen a L. Xie, „Multi-Band Melgan: Faster Waveform Generation for High-Quality Text-To-Speech", in *2021 IEEE Spoken Language Technology Workshop, SLT 2021 - Proceedings*, 2021. doi: 10.1109/SLT48900.2021.9383551.

[60] I. J. Goodfellow *et al.*, „Generative Adversarial Networks". arXiv, 2014. doi: 10.48550/ARXIV.1406.2661.

[61] J. Kong, J. Kim a J. Bae, „HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis", in *Advances in Neural Information Processing Systems*, 2020.

[62] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola a L. K. Saul, „Introduction to variational methods for graphical models", *Mach Learn*, roč. 37, č. 2, 1999, doi: 10.1023/A:1007665907178.

[63] J. Kim, J. Kong a J. Son, „Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech". arXiv, 2021. doi: 10.48550/ARXIV.2106.06103.

[64] D. J. Rezende a S. Mohamed, „Variational inference with normalizing flows", in *32nd International Conference on Machine Learning, ICML 2015*.

[65] J. Kim, S. Kim, J. Kong a S. Yoon, „Glow-TTS: A generative flow for text-to-speech via monotonic alignment search", in *Advances in Neural Information Processing Systems*, 2020.

[66] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi a W. Chan, „WaveGrad: Estimating Gradients for Waveform Generation". arXiv, 2020. doi: 10.48550/ARXIV.2009.00713.

[67] R. Prenger, R. Valle a B. Catanzaro, „Waveglow: A Flow-based Generative Network for Speech Synthesis", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019, roč. 2019-May. doi: 10.1109/ICASSP.2019.8683143.

[68] A. Łańcucki, „FastPitch: Parallel text-to-speech with pitch prediction", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021, roč. 2021-June. doi: 10.1109/ICASSP39728.2021.9413889.

[69] Y. Ren *et al.*, „FastSpeech 2: Fast and High-Quality End-to-End Text to Speech". arXiv, 2020. doi: 10.48550/ARXIV.2006.04558.

[70] S. O. Arik *et al.*, „Deep Voice: Real-time Neural Text-to-Speech", *CoRR*, roč. abs/1702.0, č. 3, s. 1504–1508, 2014, [Online]. Dostupné z: https://arxiv.org/abs/1711.10433

[71] S. O. Arik *et al.*, „Deep voice 2: Multi-speaker neural text-to-speech", in *Advances in Neural Information Processing Systems*, 2017.

[72] W. Ping, K. Peng, a J. Chen, „Clarinet: Parallel wave generation in end-to-end text-to-speech", in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[73] J. Sotelo *et al.*, „Char2Wav: End-to-End Speech Synthesis", in *International Conference on Learning Representations (ICLR)*, 2017, s. 44–51. doi: 10.1227/01.NEU.0000297116.62323.15.

[74] X. Tan *et al.*, „NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality". arXiv, 2022. doi: 10.48550/ARXIV.2205.04421.

Seznam publikovaných prací

- Vít, J.: **Detekce scén jako podpora při automatickém dabingu v projektu ELJABR**. Bakalářská práce.
- Matoušek, J.; Vít, J.: **Improving automatic dubbing with subtitle timing optimisation using video cut detection**. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, s. 2385–2388.
- Leroy, J.; Rocca, F.; Mancas, M.; Ben, M.; Grisard, F.; Kliegr, T.; Kuchar, J.; Vít, J.; Pirner, I.; Zimmermann, P.: **KINterestTV – Towards Noninvasive Measure of User Interest While Watching TV**. In Innovative and Creative Developments in Multimodal Interaction Systems, sv. 425, Lisbon, Portugal: Springer Berlin Heidelberg, 2014, s. 179–199.
- Vít, J.: **Detecting artifacts in synthetic speech**. In Tackling the complexity in speech, editace O. Niebuhr; R. Skarnitzl, 2015.
- Vít, J.: **Syntéza řeči z audioknih**. In Studentská vědecká konference, Západočeská univerzita v Plzni, 2015, s. 123–124.
- Vít, J.: **Automatická detekce a vizualizace chyb konkatenační syntézy řeči**. Diplomová práce.
- Vít, J.: **Webový nástroj pro opravy anotací řečového inventáře**. In Studentská vědecká konference, Západočeská univerzita v Plzni, 2016, s. 127–128.
- Vít, J.: **Zvyšování přirozenosti počítačové syntézy řeči**. Odborná práce ke státní doktorské zkoušce.
- Vít, J.; Matoušek, J.: **Concatenation artifact detection trained from listeners evaluations**. In Text, Speech and Dialogue, Lecture Notes in Computer Science, sv. 8082, Pilsen, Czech Republic, 2013, s. 169–176.
- Vít, J.; Matoušek, J.: **Unit-selection speech synthesis adjustments for audiobook-based voices**. In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2016.
- Vít, J.: **WaveNet: nová metoda syntézy řeči.** In Studentská vědecká konference, Západočeská univerzita v Plzni, 2017.
- Gruber, M.; Matoušek, J.; Hanzlíček, Z.; Vít, J.; Tihelka, D.: **WebSubDub-Experimental System for Creating High-Quality Alternative Audio Track for TV Broadcasting.** In Proceedings of Interspeech, 2017

- Wan, V.; Agiomyrgiannakis, Y.; Silen, H.; Vít, J.: **Google's Next-Generation Real-Time Unit-Selection Synthesizer Using Sequence-to-Sequence LSTM-Based Autoencoders.** In Proceedings of Interspeech, 2017
- Vít, J.; Hanzlíček, Z.; Matoušek, J.: **On the analysis of training data for WaveNet-based speech synthesis,** In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, Canada, 2018
- Vít, J.: **On the analysis of training data for WaveNet-based speech synthesis.** Google PhD Speech Summit, 2018, London.
- Tihelka, D.; Hanzlíček, Z.; Jůzová, M.; Vít, J.; Matoušek, J.; Grůber, M.: **Current state of text-to-speech system ARTIC: A decade of research on the field of speech technologies.** In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2018.
- Hanzlíček, Z; Vít, J.; Tihelka, D.: **WaveNet-Based Speech Synthesis Applied to Czech.** In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2018.
- Chun, H.; Gonzalvo, J.; Chan, Ch.; Agiomyrgiannakis, I.; Wan, V.; Clark, R.; Vít, J.: **Text-to-speech synthesis using an autoencoder.** U.S. Patent US20180268806
- Jůzová, M.; Vít, J.: **Using Auto-Encoder BiLSTM Neural Network for Czech Grapheme-to-Phoneme Conversion.** In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2019.
- Jůzová, M.; Tihelka, D.; Vít, J.: **Unified Language-Independent DNN-Based G2P Converter.** In Proceedings of Interspeech, 2019
- Hanzlíček, Z.; Vít, J.: **LSTM-based Speech Segmentation for TTS Synthesis.** In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2019.
- Vít, J.; Hanzlíček, Z.: **Czech Speech Synthesis with Generative Neural Vocoder.** In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2019.
- Kenter, T.; Wan, V.; Chan, Ch.; Clark, R.; Vít, J: **CHiVE: Varying Prosody in Speech Synthesis with a Linguistically Driven Dynamic Hierarchical Conditional Variational Network**. In ICML 2019.
- Grůber M.; Vít J.; Matoušek J.: **Web-Based Speech Synthesis Editor**. In Proceedings of Interspeech, 2019.
- Hanzlíček Z.; Vít J.: **LSTM-Based Speech Segmentation Trained on Different Foreign Language.** In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2020.

- Tihelka D.; Jůzová M.; Vít J.: **Grappling with Web Technologies: the Problems of Remote Speech Recording**. In SPECOM 2020.
- Hanzlíček Z.; Vít J., Řezáčková M.: **Speakers Talking Foreign Languages in a Multi-lingual TTS System**. In Text, Speech and Dialogue, Lecture Notes in Computer Science, 2021.
- Matoušek J., Řezáčková M., Tihelka, D., Grůber M., Hanzlíček Z., Vít, J.: **Save Your Voice: Voice Banking and TTS for Anyone**. In Proceedings of Interspeech, 2021.