VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROBUSTNÍ ODŠUMOVÁNÍ A DEREVERBERACE AUDIA

ROBUST AUDIO DEREVERBERATION AND DENOISING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **SIMON KOŠINA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. IGOR SZŐKE, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2021/2022

Zadání bakalářské práce

Student: **Košina Simon**
Program: Informační technologie
Název: **Robustní odšumování a dereverberace audia**
Robust Audio Dereverberation and Denoising
Kategorie: Zpracování řeči a přirozeného jazyka

Zadání:

1. Seznamte se s metodami umělého zašumění (data augmentation) a odšumění signálu (data enhancement) pomocí machine learning. Dále se seznamte s dodanými daty (záznamy VHF komunikace).
2. Seznamte se s doporučenými nástroji pro odšumění audia a machine learning. Natrénujte vlastní síť pomocí dodaného nástroje.
3. Nastudujte, navrhněte a realizujte kroky (můžou být iterativní), jak z dodaných dat natrénovat co nejlepší odšumovač.
4. Průběžně vyhodnocujte úspěšnost.
5. Zhodnoťte dosažené výsledky a navrhněte směry dalšího vývoje.
6. Vytvořte A2 plakátek nebo 20-30 vteřinové video o Vašem projektu.

Literatura:

- SEGAN: Speech Enhancement Generative Adversarial Network; Santiago Pascual, Antonio Bonafonte, Joan Serra; INTERSPEECH 2017, https://arxiv.org/pdf/1703.09452.pdf
- Dále podle pokynů školitele

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a část bodu 3 ze zadání.

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

Vedoucí práce: **Szőke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 1. listopadu 2021

Abstrakt

Cieľom tejto práce je vytvorenie modelu pre odšumovanie a dereverberáciu audio nahrávok pochádzajúcich z leteckej VHF komunikácie. Práca popisuje teoretické základy strojového učenia a rôzne architektúry neurónových sieti, ktoré sa v prípade podobných problémov často používajú. Nasleduje popis použitých nástrojov, implementácie a dátových sád. Posledné kapitoly sa venujú vykonaným experimentom, dosiahnutým výsledkom a nadväzujúcej práci.

Abstract

The goal of this thesis was to create a speech enhancement and dereverberation model for audio recordings coming from aircraft VHF communication. First, the thesis covers some theoretical grounds of machine learning and types of neural networks commonly used in such scenarios. Following is a description of the used framework, datasets and the implementation itself. Last chapters are focused on the performed experiments and their evaluation. At the end we talk about the future work that can be done in order to further improve the achieved results.

Kľúčové slová

odstraňovanie šumu, strojové učenie, VHF komunikácia, GAN, generatívne neurónové siete, SpeechBrain

Keywords

speech enhancement, machine learning, VHF communication, GAN, generative adversarial network, SpeechBrain

Citácia

KOŠINA, Simon. *Robustní odšumování a dereverberace audia.* Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szőke, Ph.D.

Robustní odšumování a dereverberace audia

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Igora Szökeho, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Simon Košina
16. mája 2022

Poďakovanie

Ďakujem vedúcemu práce Ing. Igorovi Szöke, Ph. D. za konzultácie a cenné rady, ktoré mi v priebehu vypracovávania práce pomohli.

Obsah

Kapitola 1

Úvod

Obor odstraňovania šumu sa zaoberá vylepšovaním percepčných vlastností reči, ktorá bola poškodená aditívnym šumom [31]. Vo väčšine prípadoch je cieľom zlepšenie kvality a zrozumiteľnosti poškodených signálov. Takýto signál môže pochádzať z hlučného prostredia, alebo môže byť poškodený pri prenose cez nedokonalý komunikačný kanál.

Metódy odšumovania audio signálov majú v dnešnej dobe široké uplatnenie, keďže ľudia spolu čoraz častejšie komunikujú prostredníctvom telekomunikačných technológií. Metódy sa taktiež využívajú v rôznych sluchových pomôckach a pod. Kvalitnejší a zrozumiteľnejší audio signál môže znížiť úroveň vyčerpania poslucháčov, v prípade, že sú daným signálom vystavený po dlhšiu dobu [31].

Táto práca sa zaoberá odšumovaním a dereverberáciou audio signálov pochádzajúcich z leteckej VHF komunikácie. Degradácia týchto audio signálov nastáva hlavne kvôli vysokým hladinám rušivého hluku v kokpitoch lietadiel, poprípade vzniknutou ozvenou. Ďalším problémom je rušenie rádiových vĺn nevhodným počasím, alebo blokovanie komunikácie pohoriami. V takýchto prípadoch je oveľa dôležitejšie zlepšiť zrozumiteľnosť rečového signálu ako jeho kvalitu.

Kapitola 2 obsahuje úvod do problematiky a prehľad existujúcich riešení. Kapitola 3 opisuje základy strojového učenia a rôzne architektúry neurónových sietí využívané pri odšumovaní a dereverberácii audia. Technológie využité na implementáciu riešenia sú popísané v kapitole 4. Kapitola 4 taktiež obsahuje popis implementovaných modulov. Dátové sady využité v experimentoch sú popísané v kapitole 5, pričom samotné experimenty a ich zhodnotenie je popísané v kapitole 6. Celkové zhrnutie získaných výsledkov a možnosti ďalšieho zlepšenia výsledkov sa nachádzajú v poslednej kapitole 7.

Kapitola 2

Odšumovanie a dereverberácia audia

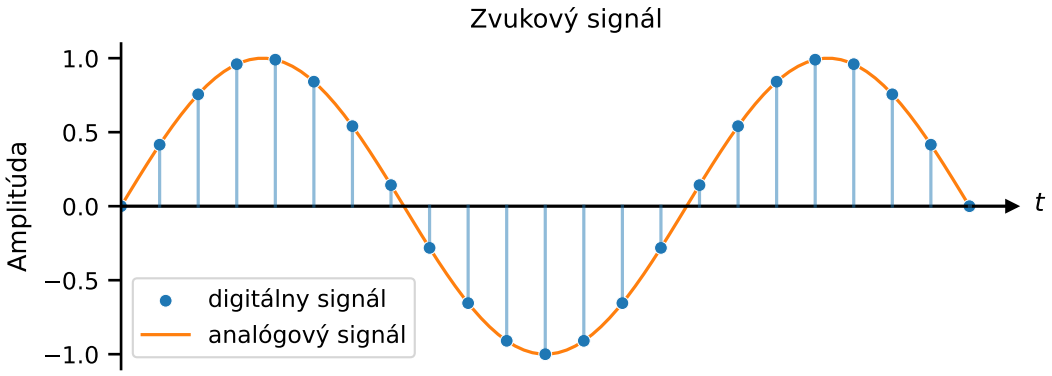
Pre zlepšenie kvality audio signálov je potrebné pochopiť spôsoby, akými môžu byť poškodené a kedy daná degradácia nastáva. Ďalej je potrebné definovať metódy na kvantifikovanie kvality a zrozumiteľnosti signálov, aby bolo možné porovnať výsledky jednotlivých algoritmov.

Táto kapitola popisuje proces tvorby zvukových signálov, vysvetľuje pojem šum a reverberácia. Obsahuje prehľad techník, ktoré sa využívajú na odstránenie spomenutých problémov a taktiež popisuje možnosti vyhodnotenia kvality a zrozumiteľnosti rečových audio signálov.

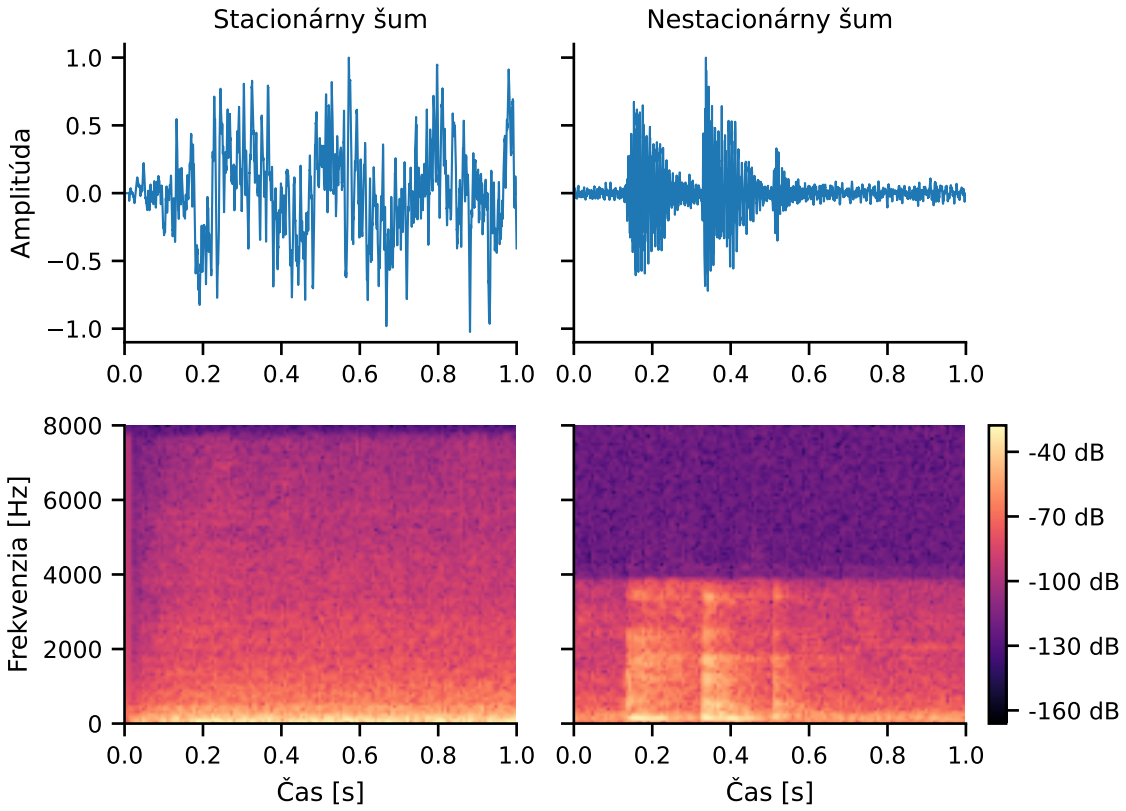
2.1 Zvukový signál

Zvukový signál reprezentuje zvuk, teda určité mechanické vlnenie, šíriace sa v prenosnom médiu, ktoré je schopné vyvolať v ľudskom uchu sluchový vnem. Prenosným médiom môže byť napríklad vzduch, či iné plynné, kvapalné alebo pevné látky. Audio signál je možné pomocou mikrofónu zaznamenať a typicky tak získať analógový zvukový signál, ktorý je spojitý a je vyjadrený pomocou elektrického napätia, či prúdu. Na to, aby bolo možné tento signál uložiť do pamäte počítača a následne s ním pracovať je potrebné ho previesť do digitálnej podoby.

Na tieto účely sa typicky využíva *pulzná kódová modulácia* (PCM, Pulse-code modulation), ktorej princíp spočíva v pravidelnom čítaní hodnoty signálu pomocou analógovo-digitálneho prevodníku a následnom prevode tejto hodnoty do binárnej podoby. Dĺžku časových intervalov medzi čítaniami určuje vzorkovacia frekvencia, udávajúca počet vzorkov za jednu sekundu. V prípade zvukových signálov sa väčšinou využívajú vzorkovacie frekvencie od 8 kHz do 44.1 kHz. Určenie konkrétnej binárnej hodnoty vzorku sa nazýva kvantovanie a je ovplyvnené rozlíšením, ktoré určuje presnosť jednotlivých vzorkov. Typicky sa používa 16 bitové rozlíšenie a teda je k dispozícii 65536 rôznych hodnôt.



Obr. 2.1: Porovnanie analógového signálu a jeho digitálnej reprezentácie.



Obr. 2.2: Porovnanie stacionárneho a nestacionárneho šumu v časovej a časovo-frekvenčnej doméne.

2.2 Šum

Šumom je zvyčajne myslená nechcená modifikácia, ktorej môže byť signál vystavený pri zaznamenávaní, ukladaní, prenose, spracovávaní, či prehrávaní. Na základe procesov, ktoré generujú tieto signáli rozlišujeme:

- **stacionárny šum** – generovaný náhodným procesom, ktorého štatistické vlastnosti sa časom nemenia. Potom, pre určitý stacionárny proces, $X(t)$ a $X(t + \Delta)$ majú rovnaké rozdelenie pravdepodobnosti [39]. Príkladom takéhoto procesu je biely šum, či ďalšie zafarbené šumy.
- **nestacionárny šum** – generovaný náhodným procesom, ktorého štatistické vlastnosti sú závislé na čase. Sem patrí mnoho šumov z bežného života ako napríklad rozprávanie ľudí v pozadí, rušivé zvuky z premávky a pod.

Problematika odstránenia nestacionárnych šumov je značne komplexnejšia, keďže vlastnosti nežiadúceho signálu sa neustále menia s časom.

Techniky odšumenia signálov
Techniky odšumenia signálov môžu byť kategorizované do dvoch hlavných skupín:

- **Konvenčné metódy** – vo všeobecnosti je cieľom týchto metód odhadnúť šum v degradovanom signále. Jednou z najstarších konvenčných metód je Wienerov filter [30]. Ďalším príkladom je spektrálna subtrakcia [3], ktorá odhaduje šum vo frekvenčnej doméne z úsekov signálu bez reči a následne využíva tento odhad pre odstránenie šumu v úsekoch signálu s rečou. Tiež sem môžu byť zaradené metódy založené na minimálnej strednej kvadratickej chybe (minimum mean-square error, MMSE) [9]. Veľkým nedostatkom týchto metód je ich predpoklad, že šum je stochastický proces a teda efektívnosť týchto metód je pomerne horšia v prostrediach s nestacionárnym šumom [1].
- **Metódy založené na hlbokých neurónových sieťach** – od pôvodného predstavenia hlbokých neurónových sietí (deep neural networks, DNN), popísaných v 3.2, v roku 2006 [20], bolo navrhnutých mnoho architektúr DNN a to napríklad autoenkódery [53], konvolučné neurónové siete (convolutional neural network, CNN) [27] a rekurentné neurónové siete (recurrent neural network, RNN) [18]. Na rozdiel od konvenčných metód, metódy založené na DNN umožňujú efektívne spracovanie nestacionárnych signálov [10]. Metódy v tejto kategórii je možné rozdeliť do dvoch skupín na základe domény, v ktorej pracujú [1].
  - Prvou skupinou sú metódy, ktoré pracujú v časovo-frekvenčnej doméne. Hlavným problémom bežnej Fourierovej transformácie signálu je predpoklad, že signály sú nekonečné a periodické. Skutočné signály, ako napríklad reč, však majú začiatok a následne sa amplitúda signálu postupne zmenšuje. Analýza v časovo-frekvenčnej doméne rieši tento problém skúmaním signálu jednak v časovej, ale aj vo frekvenčnej doméne. Príkladom takejto analýzy je krátkodobá Fourierova transformácia (short-time Fourier transform, STFT [25]. Metódy v tejto kategórii väčšinou využívajú priame mapovanie degradovaného signálu na signál čistý, alebo na základe degradovaného signálu vytvárajú masku. Pôvodne bola využívaná binárna masku, ktorá identifikuje každú jednotku v časovo-frekvenčnej

reprezentácii zašumeného signálu, buď ako reč, alebo ako šum [32]. Neskôr sa začala používať napríklad plynulá maska (soft mask), ktorej hodnoty predstavujú pravdepodobnosť, že daná jednotka časovo-frekvenčnej reprezentácie signálu je rečovo dominantná [32].

– Druhou skupinou sú metódy, ktoré pracujú so signálom priamo v časovej doméne. Príkladom takejto metódy je [12], kde bola využitá plne konvolučná sieť pre mapovanie zašumených signálov v časovej doméne na ich čisté verzie [1].

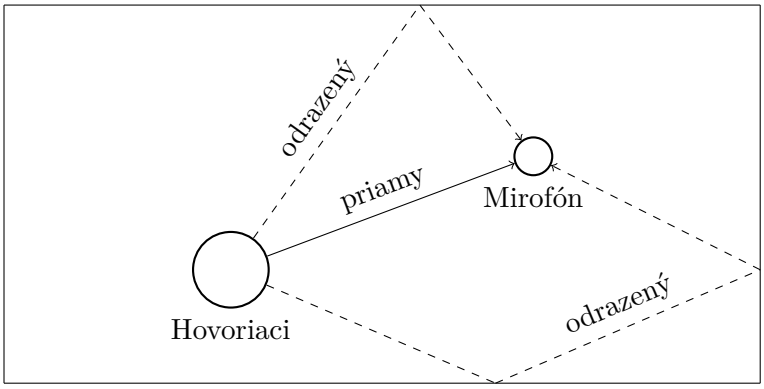
Taktiež existujú hybridné metódy, ktoré začleňujú konvenčné metódy do metód strojového učenia [1][48].

2.3 Reverberácia

Pri zaznamenávaní signálov v uzavretých priestoroch jedným, či viacerými mikrofónmi umiestnenými v určitej vzdialenosti od hovoriaceho, sa výsledný signál skladá z niekoľkých oneskorených a pozmenených kópii pôvodného signálu, keďže pôvodné zvukové vlny sa môžu niekoľkokrát odraziť od rôznych objektov v priestore ako sú steny, nábytok, či vzduch. Oneskorenie vzniká kvôli rôznym dĺžkam ciest pri šírení pôvodného a odrazeného zvuku [33].

Reverberácia mení vlastnosti rečového signálu a degraduje zrozumiteľnosť reči, čo je problematické v aplikáciach pracujúcich s týmito signálmi vrátane rozpoznávania reči, lokalizácie zdroja, verifikácie rečníka a významne zhoršuje efektívnosť algoritmov, ktoré ju neberú v úvahu. Škodlivé efekty sa zväčšujú spolu s rastúcou vzdialenosťou medzi rečníkom a mikrofónom [33].

Reverberáciu je tiež možné vnímať ako konvolúciu impulznej odozvy miestnosti a pôvodného signálu bez ozveny. Táto vlastnosť sa často využíva pri tvorbe umelo degradovaných nahrávok pre učenie spomínaných neurónových sietí.



Obr. 2.3: Znázornenie vzniku reverberácie v uzavretom priestore, obrázok inšpirovaný ilustráciou v [33].

Dereverberácia

Dereverberácia je proces, ktorého cieľom je odstrániť reverberáciu z audio signálu. Spravidla sa využívajú tri rôzne postupy [19]. V prvom postupe je využitý matematický model akustického systému a po vyladení parametrov modelu sa odhadne pôvodný signál. Pri druhom postupe je reverberácia považovaná za aditívny šum a využívajú sa odšumovacie

techniky určené pre dereverberáciu. Posledný postup sa snaží priamo odhadnúť čistý signál bez reverberácie z degradovaného signálu a to napríklad pomocou DNN.

2.4 Hodnotenie rečového signálu

Metódy vyhodnotenia kvality a zrozumiteľnosti rečových signálov sú rozdelené na objektívne a subjektívne. V prípade subjektívnych metód je väčšinou využitá skupina ľudí, ktorá hodnotí nahrávky podľa zadanej stupnice. Subjektívne metódy sú veľmi spoľahlivé, no ich vykonanie môže byť časovo náročné a v mnohých prípadoch je potreba mať zaškolených poslucháčov [31]. Objektívne metódy definujú matematické postupy pre výpočet hodnotenia kvality a zrozumiteľnosti. Vyhodnotenie objektívnych metód nie je také zdĺhavé, je však potrebné správne definovať hodnotiace funkcie a overiť ich koreláciu s metódami subjektívnymi [31].

Pojmy kvalita a zrozumiteľnosť rečového signálu nie sú totožné. Kvalita signálu je pomerne subjektívny pojem a každý poslucháč môže mať inú predstavu o tom, čo je „dobrá“ a „zlá“ kvalita audio signálu. Naopak zrozumiteľnosť je možné hodnotiť objektívnym spôsobom, kde skupine poslucháčov sú za určitých podmienok prehrané rôzne nahovorené audio signály a ich cieľom je identifikovať jednotlivé slová alebo fonémy v nich. Zrozumiteľnosť je následne možné vyhodnotiť podľa počtu správne identifikovaných slov a foném. Rôzne postupy pre subjektívne vyhodnotenie zrozumiteľnosti reči sú popísané v knihe [31].

MOS

Typickou metódou subjektívneho vyhodnotenia kvality je metóda *MOS* (mean opinion score) [22]. Jednotlivý poslucháči hodnotia kvalitu prehraných audio signálov, na základe predom definovanej päť bodovej diskrétnej stupnice, kde 5 hovorí, že signál bol výbornej kvality, zatiaľ čo 1 predstavuje neuspokojivú, či zlú kvalitu. Výsledné hodnotenie sa vypočíta ako priemer hodnotení všetkých poslucháčov.

Test sa skladá z dvoch fáz, kde v prvej sú poslucháčom prehrané nahrávky odpovedajúce jednotlivým hodnoteniam kvality. Cieľom je vyrovnať rozdiely medzi subjektívnymi predstavami poslucháčov o „dobrej“ a „zlej“ kvalite signálu. V druhej fáze prebieha samotné hodnotenie. Podrobné doporučenia pre podmienky vykonania testov, výber skupiny poslucháčov a ich zaškolenie sú dostupné v [23].

PESQ

PESQ (perceptual evaluation of speech quality measure) je objektívna metóda, slúžiaca na vyhodnotenie kvality signálu. Metóda je štandardizovaná v doporučení od ITU-T [21]. Cieľom tejto metódy je predikovať subjektívne hodnotenie kvality MOS. Napriek tomu, že hodnotenie PESQ úplne neodpovedá subjektívnemu hodnoteniu, jej korelácia s metódou MOS je pomerne vysoká [31].

Pre vyhodnotenie sú potrebné oba signály, t.j. pôvodný a upravený signál. V prvom rade je vyrovnaná hlasitosť týchto signálov a signály sú v zápätí časovo zarovnané. Nasleduje spracovanie signálov a výpočet predikcie subjektívneho MOS hodnotenia. Podrobný popis celého výpočtu je dostupný v [31].

Výsledne hodnotenia vypočítané pomocou metódy PESQ ležia v intervale $[-0.5, 4.5]$, pričom vyššia hodnota predstavuje lepšiu kvalitu signálu.

STOI

Príkladom objektívnej metódy pre kvalifikáciu zrozumiteľnosti reči je metóda *STOI* (short-time objective intelligibility), navrhnutá v [46]. Metóda pracuje s jednotlivými úsekmi časovo-frekvenčnej reprezentácie signálu.

STOI je možné využiť pre ohodnotenie čistých ale aj zašumených dát, čo pri metódach objektívnej kvalifikácie zrozumiteľnosti reči nie vždy platí. Výstupom tejto metódy sú hodnoty z intervalu $[0, 100]$ predstavujúce percento zrozumiteľnosti nahrávky, čo znamená, že väčšia hodnota je lepšia.

Metódy hodnotenia rečových signálov v kontexte tejto práce

Metódy hodnotenia rečových signálov je dôležité spomenúť jednak kvôli tomu, že existujú mnohé spôsoby zlepšovania kvality nahrávok, ktorých účinnosť je potrebné určitým spôsobom kvantifikovať aby bolo možné ich navzájom porovnať. Objektívne hodnotenia kvality a zrozumiteľnosti signálov, alebo ich modifikácie, sa tiež využívajú ako stratové funkcie, ktoré chceme optimalizovať, v prípade trénovania neurónových sietí spomínaných v kapitole 3.

Kapitola 3

Strojové učenie

Strojové učenie (machine learning, ML) je podmnožinou oboru umelej inteligencie, ktorá umožňuje vybudovanie matematického modelu pomocou učenia sa na príkladoch. Vybudovaný model má byť schopný vykonávať vlastné predikcie a rozhodnutia bez toho, aby na to bol explicitne naprogramovaný. Pre samotný proces trénovania modelu je potrebné mať dostatočné množstvo vzorových dát, ktoré sú algoritmy schopné analyzovať a identifikovať v nich vzory. Vzorové dáta, na ktorých sa model učí, sú nazývané ako *trénovacie dáta*.

Pre získanie nezaujatého vyhodnotenia finálneho modelu a jeho schopnosti vykonávať vlastné predikcie sú využité *testovacie dáta*, obsahujúce príklady, ktoré modelu neboli dostupné počas tréningu.

Vzhľadom na to, že trénovanie modelu trvá určitú dobu, je vhodné vytvoriť ďalšiu dátovú sadu a tou sú *validačné dáta*. Validačné dáta slúžia na získanie priebežného nezaujatého vyhodnotenia modelu počas jeho trénovania. Toto vyhodnotenie sa však môže stať skresleným v prípade, že vyhodnotenie modelu na validačných dátach je využité k jeho trénovaniu.

Pri práci s rečovými signálmi je možné využiť strojové učenie, napríklad na zlepšenie kvality a zrozumiteľnosti týchto signálov, automatické rozpoznanie reči, identifikáciu rečníka a pod.

3.1 Typy strojového učenia

Algoritmy strojového učenia sa z pravidla delia do 4 základných kategórií.

Učenie s učiteľom (Supervised learning)
Trénovacie dáta sú tvorené dvojicami vstupných objektov (vektor príznakov) a požadovaného výstupu. Požadovaným výstupom môže byť kategória, do ktorej je daný príklad zaradený alebo určitá spojitá hodnota. Učiaci algoritmus sa iteratívnym spôsobom snaží minimalizovať počet trénovacích príkladov zaradených do zlej kategórie, alebo chybu medzi vypočítanou a požadovanou hodnotou, až pokiaľ nedosiahne požadovanej presnosti. Metódy tohto typu sa väčšinou využívajú v prípade klasifikačných úloh alebo regresnej analýzy.

Učenie bez učiteľa (Unsupervised learning)
V tomto prípade nie sú trénovacie dáta nijak klasifikované, ani označené. Algoritmy musia sami nájsť vzory vyskytujúce sa v dátach. Príkladom takýchto algoritmov je zhlukovanie, ktoré má za cieľ rozdeliť trénovacie dáta do skupín s podobnými vlastnosťami. Počet sku-

pín je buď predom definovaný, alebo ho algoritmus určí sám na základe iných vstupných parametrov.

Kombinácia učenia s učiteľom a bez (Semi-supervised learning)
Tvorba označených dátových sád môže byť zdĺhavý a drahý proces, zatiaľ čo neoznačených dát je v dnešnej dobe k dispozícií pomerne veľa. Z tohto dôvodu vznikli metódy, ktoré sa snažia využiť výhody oboch prístupov a využívajú kombináciu veľkého množstva neoznačených dát spolu s určitým množstvom označených dát [8].

Spätnoväzbové učenie (Reinforcement learning)
Metódy tohto typu sa väčšinou zaoberajú interakciou agenta s prostredím pomocou akcií, ktoré vykonáva tak, aby maximalizoval svoju celkovú odmenu. V každom časovom okamžiku sú z prostredia agentovi prístupné určité observácie a na ich základe agent vykoná vybranú akciu v danom prostredí. Následne je agent za túto akciu odmenený, poprípade potrestaný. Chovanie agenta je dané pravidlami, ktoré mapujú observácie z prostredia na akcie. Cieľom spätnoväzbového učenia je nájsť vyhovujúcu sadu pravidiel.

3.2 Neurónová sieť

Súčasťou strojového učenia je tvorba modelu, ktorý je následne natrénovaný na dátach a využitý k vytváraniu vlastných predikcií. Jedným z typov využívaných modelov sú neurónové siete.

Umelé neurónové siete (artificial neural network, ANN) predstavujú prírodou inšpirovaný výpočetný systém, ktorý napodobňuje chovanie biologickej nervovej sústavy. Neurónové siete sú typicky organizované do vrstiev. Každá vrstva je tvorená určitým počtom prepojených jednotiek nazývaných ako *neuróny*.

Neuróny prijímajú vstupné signály, reprezentované reálnymi číslami, tie spracovávajú a vysielajú výstupný signál ďalším pripojeným neurónom. Výstupný signál sa spočíta ako suma vstupných signálov, na ktorú je následne aplikovaná aktivačná funkcia. K sume signálov môže byť taktiež pripočítaný bias pre posun aktivačnej funkcie. Jednotlivým spojeniam medzi neúronmi sú priradené váhy, ktoré predstavujú silu signálu. Výstup neuróny je teda definovaný ako

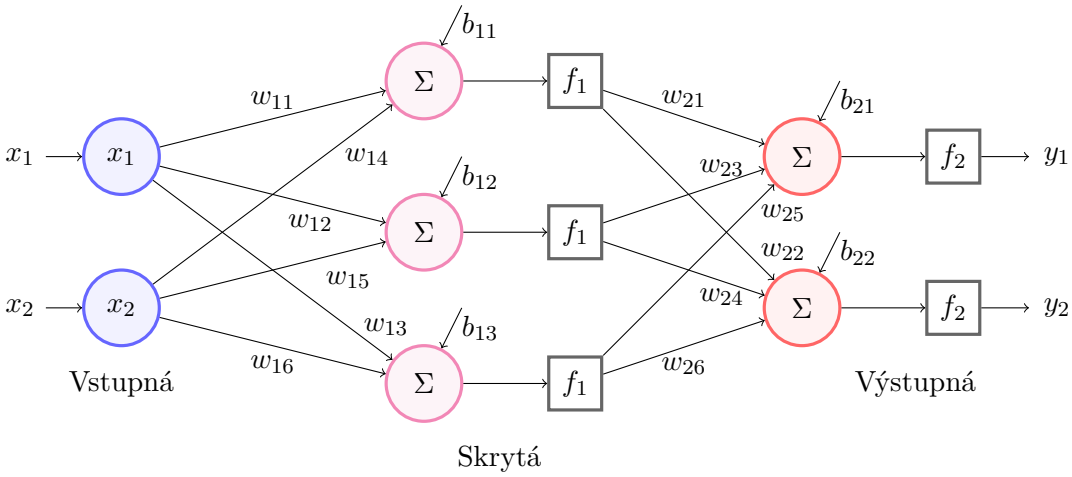
$$y = f(\sum_{\forall i} x_i w_i + b),$$

kde f je aktivačná funkcia, x_i je výstup i-tého neurónu z predchádzajúcej vrstvy pripojeného k aktuálnemu neurónu, w_i predstavuje váhu spojenia medzi i-tým a aktuálnym neurónom a b je bias pre aktuálny neurón.

Neurónové siete sú organizované do vrstiev, cez ktoré putuje daný signál. Prvá vrstva je nazývaná ako *vstupná*. Za ňou nasleduje niekoľko vrstiev, ktoré sa označujú ako *skryté*. Posledná vrstva je *výstupná*. Výstup danej vrstvy je možné kompaktne zapísať pomocou maticového násobenia ako

$$x_n = f(W_n x_{n-1} + b_n),$$

kde f je aktivačná funkcia, x_{n-1} je vektor výstupov predchádzajúcej vrstvy, W je matica váh spojení medzi predchádzajúcou a aktuálnou vrstvou a b_n je vektor bias-ov pre neuróny aktuálnej vrstvy.



Obr. 3.1: Jednotlivé vrstvy doprednej neurónovej siete spolu s aktivačnými funkciami.

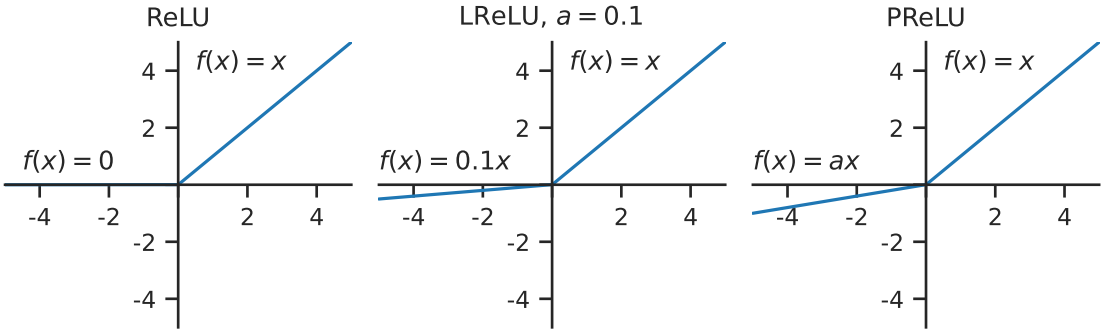
Aktivačné funkcie

Aktivačné funkcie v neurónových sieťach slúžia na transformáciu vstupných signálov na výstupný signál. Bez aktivačných funkcií, alebo v prípade využitia lineárnych aktivačných funkcií, sa model neurónovej siete správa podobne ako model lineárnej regresie. Využitie nelineárnych aktivačných funkcií umožňuje modelu aproximovať zložitejšie funkcie a objaviť v dátach komplexné štruktúry [44].

Niektoré aktivačné funkcie taktiež obmedzujú rozsah hodnôt pre výstupný signál neurónu, čo môže pozitívne ovplyvniť stabilitu trénovania modelu [29].

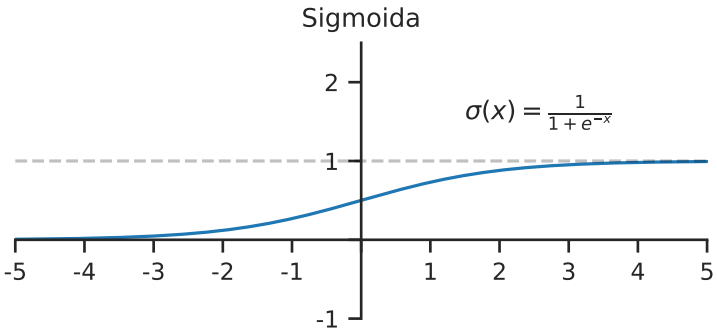
Ďalšou vlastnosťou, ktorá je skúmaná pri aktivačných funkciách je ich diferencovateľnosť, vzhľadom na to, že počas tréningu je typicky využívaný gradientný zostup. Táto vlastnosť však nie je podmienkou a v praxi sa bežne používajú aktivačné funkcie, ktoré nie sú derivovateľné v určitých bodoch. Jednou z takýchto aktivačných funkcií je napríklad pomerne populárna ReLU (Rectified Linear Unit), ktorá nie je derivovateľná pre $x = 0$.

- **Rectified Linear Unit (ReLU)** – ReLU predstavuje asi najčastejšie používanú nelineárnu aktivačnú funkciu. Existujú rôzne varianty ako napríklad LReLU (Leaky ReLU) alebo PRelu (Parametrized ReLU), ktoré pre $x < 0$ definujú $f(x) = ax$. LReLU pevne definuje a ako určitú malú hodnotu, napr. $a = 0.01$. V prípade PReLU sa hodnota a optimalizuje počas tréningu.



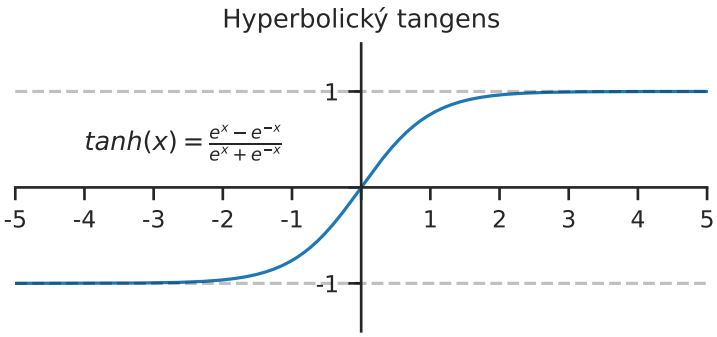
Obr. 3.2: Porovnanie rôznych variácií ReLU. LReLU definuje pevnú hodnotu a, zatiaľ čo PReLU optimalizuje a počas tréningu.

- **Sigmoida** – Sigmoida je príkladom logistickej funkcie. Oborom hodnôt tejto funkcie je interval 0 až 1. Využitie logistických aktivačných funkcií môže spôsobiť problémy zvané *miznúci* (vanishing) a *explodujúci gradient*. Bližší popis týchto problémov je dostupný v [2].



Obr. 3.3: Logistická aktivačná funkcia sigmoida.

- **Hyperbolický tangens (tanh)** – Hyperbolický tangens je funkcia podobná sigmoide. Na rozdiel od sigmoidy je však nepárna a jej oborom hodnôt je interval -1 až 1.



Obr. 3.4: Aktivačná funkcia hyperbolický tangens.

- **Softmax** – Softmax funkcia zovšeobecňuje logistickú funkciu pre niekoľko dimenzií. Väčšionu je použitá ako aktivačná funkcia poslednej vrstvy siete a normalizuje výstup siete na hodnoty, ktoré ležia v intervale $[0, 1]$, pričom ich celková suma je 1. Tieto hodnoty môžu byť interpretované ako pravdepodobnosti a využíva sa napr. pri klasifikačných problémoch s viacerými kategóriami.

Tréning

Tréning neurónových sieti spravidla prebieha iteratívnym spôsobom, kde cieľom algoritmu je optimalizovať stratovú funkciu zmenou váh v sieti. V každej iterácii, alebo kroku, vypočíta neurónová sieť výsledky pre dodanú množinu trénovacích dát. Po získaní výsledkov je vypočítaná chyba, ktorej sa sieť dopustila a pomocou trénovacieho algoritmu sú upravené parametre siete.

Typicky je na trénovanie využitý algoritmus zvaný *gradientný zostup* (gradient descent) a na výpočet gradientu sa používa algoritmus *spätnej propagácie* (backpropagation) [16]. V prípade trénovania neurónovej siete je potrebné zistiť gradient stratovej funkcie podľa parametrov modelu. Algoritmus spätnej propagácie rekurzívne aplikuje reťazové pravidlo (chain rule) pre výpočet jednotlivých parciálnych derivácií [16].

Gradient udáva smer najrýchlejšieho rastu funkcie a pre optimalizáciu váh je teda potrebný posun v opačnom smere. Nové hodnoty váh sú dané nasledovným vzťahom

$$w' = w - \eta \nabla L,$$

kde ∇L je gradient stratovej funkcie L, η je malé číslo nazývané *learning rate*, w sú aktuálne hodnoty váh a w' predstavuje hodnoty váh po vykonaní optimalizačného kroku [34]. Celý algoritmus sa opakuje pokiaľ hodnota stratovej funkcie L konverguje.

Počiatočné hodnoty váh sú inicializované náhodne alebo sa využijú určité heuristiky. Prechod cez všetky príklady v trénovacej sade dát sa nazýva *epoch*. Učenie siete sa typicky skladá z niekoľkých epochov. Pre vykonanie optimalizačného kroku sa však využíva podmnožina trénovacích dát o určitej veľkosti nazývaná *batch*. Veľkosť týchto podmnožín a learning rate predstavujú hyperparametre modelu, bližšie popísané v 3.2.

Stratové funkcie

Stratové funkcie slúžia na výpočet chyby, či vzdialenosti, medzi vypočítaným výstupom a

cieľovou hodnotou. Všeobecne sú stratové funkcie klasifikované do 2 kategórií podľa typu riešeného problému. Jedná sa o regresné alebo klasifikačné stratové funkcie.

V prípade odšumovania audio signálov sa najčastejšie využívajú regresné stratové funkcie *Mean Squared Error* (MSE) alebo *Mean Absolute Error* (MAE) [4], definované ako

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|,$$
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2,$$

kde y_i predstavuje cieľový signál, $\hat{y}_i$ predstavuje výstup siete a N je počet príkladov. Tieto funkcie je možné kombinovať s objektívnymi metrikami pre vyhodnotenie kvality a zrozumiteľnosti rečového signálu ako sú PESQ a STOI, bližšie popísané v sekcii 2.4.

Pre klasifikačné úlohy je možné využiť stratové funkcie ako sú *krížová entropia* (cross entropy) alebo *hinge loss* bližšie popísané v [5].

Hyperparametre

Hyperparametrami sú myslené parametre neurónovej siete, ktoré ovplyvňujú jej štruktúru alebo proces trénovania. Hyperparametre musia byť definované pred procesom trénovania siete. Pre získanie vyhovujúcich výsledkov počas tréningu je potreba nájsť správne hodnoty hyperparametrov. Tento proces sa nazýva optimalizácia hyperparametrov a vykonáva sa buď ručne alebo pomocou rôznych optimalizačných techník [40].

Medzi hyperparametre ovplyvňujúce štruktúru neurónovej siete patrí napríklad počet skrytých vrstiev a jednotiek v rámci nich. Ďalším hyperparametrom je *dropout rate*, ktorý môže znížiť pravdepodobnosť pretrénovania siete [45]. Taktiež sem zaraďujeme výber aktivačných funkcií a spôsob inicializácie váh siete.

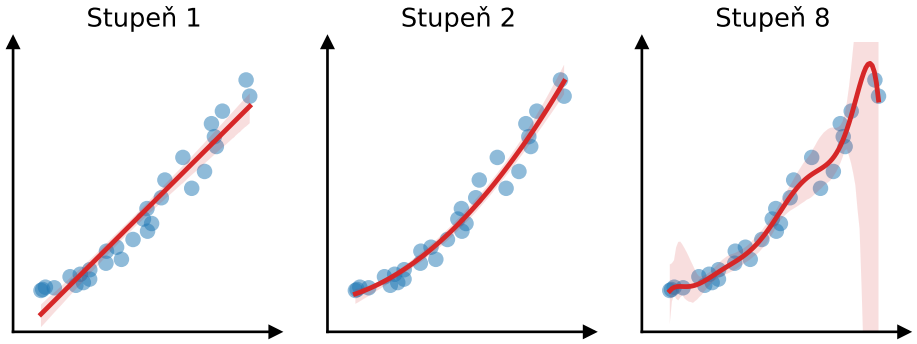
Druhou kategóriou sú hyperparametre ovplyvňujúce proces trénovania a to je napríklad už spomínaný learning rate. Learning rate určuje veľkosť zmien váh pri optimalizačnom kroku. V prípade príliš malej hodnoty je možné, že trénovací algoritmus pobeží dlho, naopak ak je zvolený learning rate príliš vysoký, môže sa stať, že algoritmus bude divergovať [6]. Určité optimalizačné algoritmy v priebehu tréningu hodnotu learning rate zmenšujú. Ďalším hyperparametrom je počet epochov, ktoré trénovací algoritmus vykoná. Po každej iterácii na trénovacích dátach sú využité validačné dáta na výpočet chyby modelu. Typicky tréning trvá tak dlho, pokiaľ hodnota chyby klesá. Taktiež je potrebné správne nastaviť hodnotu *batch size*. Väčšinou je dobré začať s menším číslom, ako napr. 32 alebo 64 a túto hodnotu postupne zvyšovať, ak je potreba [24]. Odporúča sa nastaviť túto hodnotu na mocninu 2, aby bolo možné využiť plný potenciál GPU počas tréningu. Pri menších hodnotách batch size je vhodné využiť menšie hodnoty parametru learning rate [24].

Potencionálne problémy

Pri trénovaní neurónových sietí sa môžu vyskytnúť dva hlavné problémy a to pretrénovanie a poddtrénovanie siete. Podtrénovanie nastáva v prípade, že model nie je dostatočne komplexný na to, aby mohol reprezentovať celú podstatu riešeného problému. Rovnaký problém sa môže vyskytnúť, ak počet epochov pri tréningu je príliš malý na to, aby model konvergoval.

Pretrénovanie naopak nastáva ak je daný model príliš komplexný pre daný problém, alebo tréning sa skladá z príliš mnoho epochov. Takýto model sa potom až moc presne

adaptuje na trénovacie dáta a nedokáže svoje predikcie zovšeobecniť na nové dáta, ktoré predtým nevidel. Je teda potrebné definovať testovaciu, či validačnú sadu dát, aby bolo možné nezávisle overiť daný model.



Obr. 3.5: Regresné modely s využitím rôznych stupňov kriviek. Zobrazený je aj 95% interval spoľahlivosti modelu. Lineárna funkcia úplne neodpovedá dátam, ktoré majú kvadratický trend (podtrénovanie). Kvadratická krivka vystihuje tieto dáta pomerne presne, zatiaľ čo krivka so stupňom 8 sa prispôsobilla konkrétnym trénovacím dátam (pretrénovanie).

3.3 Konvolučné neurónové siete

Konvolučné neurónove siete (convolutional neural network, CNN) sú špecializované neurónové siete pre spracovávanie dát so známou pravidelnou topológiou [16]. Takýmito dátami môžu byť napríklad 1-D audio signály alebo 2-D obrázky. Konvolučné neurónové siete sú neurónové siete, ktoré využívajú konvolúciu namiesto bežného maticového násobenia aspoň v jednej z ich vrstiev [16].

Popularita CNN významne vzrástla v roku 2012, kedy Alex Krizhevsky et al. vytvorili CNN architektúru AlexNet [27] a s pomerne veľkým náskokom vyhrali prestížnu súťaž *ImageNet Large Scale Visual Recognition Challenge* [43].

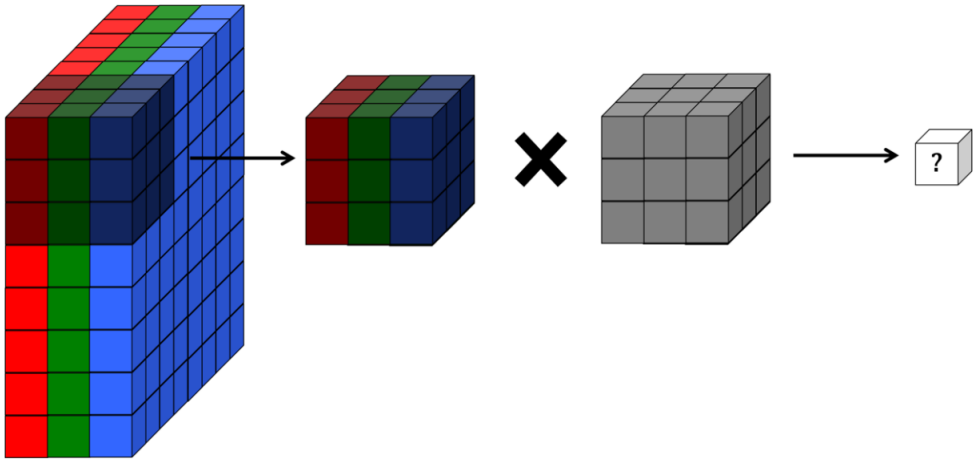
Podľa [16] sa jedna konvolučná vrstva skladá z troch hlavných fáz. V prvej z nich je na vstup paralelne aplikovaný konvolučný filter. Následne je na získané hodnoty potrebné aplikovať nelineárnu aktivačnú funkciu, vzhľadom na to, že konvolúcia sama o sebe je lineárna. V poslednej fáze je pomocou poolingovej vrstvy zmenená veľkosť výstupu.

V prípade bežných neurónových sieti sú v maticovom násobení zahrnuté váhy všetkých spojení danej vrstvy. Počet vstupných jednotiek siete a ich prepojení však rastie spolu s veľkosťou vstupu, čo môže byť problematické, ak na vstupe sú napríklad obrázky s miliónmi pixelov. Jednak je potreba uchovávať všetky hodnoty váh v pamäti, a taktiež môže byť ovplyvnený samotný výpočet výstupu. Výhodou CNN je fixná veľkosť konvolučného jadra, ktoré je väčšinou rádovo menšie ako samotný vstup. Konvolučné jadro a jeho parametre sú zdielané v rámci celej vrstvy siete, čo znamená, že v pamäti stačí uchovávať len niekoľko parametrov jadra.

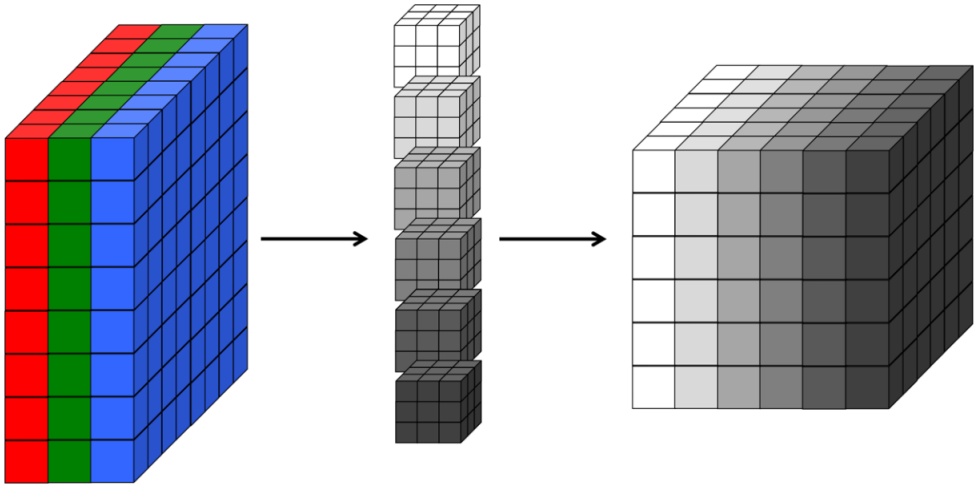
Každou aplikáciou konvolúcie je do výstupu zahrnuté aj okolie daných bodov. Hlbšie vrstvy CNN, tým pádom nepriamo interagujú s čoraz väčšou časťou pôvodného vstupu [16].

Vstupné dáta pre konvolučnú vrstvu sa môžu skladať až z niekoľkých kanálov. Počet kanálov definuje posledná dimenzia rozmeru dát. Typickým príkladom takýchto dát je RGB obrázok, kde je potreba spracovať 3 rôzne 2-D matice, jednu pre každú farebnú zložku.

Rozmery RGB obrázku so šírkou w a výškou h, je potom možné zapísať ako $h \times w \times 3$. HĺbkA konvolučného jadra býva rovnaká ako hĺbka (počet kanálov) vstupných dát, aby v rámci výpočtu boli využité všetky získané vlastnosti [6]. HĺbkA výstupu konvolučnej vrstvy potom odpovedá počtu filtrov v danej vrstve (v rámci jednej konvolučnej vrstvy môže byť definovaných až niekoľko konvolučných jadier, ktoré sa použijú pre výpočet).



Obr. 3.6: Aplikovanie objemového konvolučného filtru na všetky 3 kanály RGB obrázku, ilustrácie prevzatá z [6].

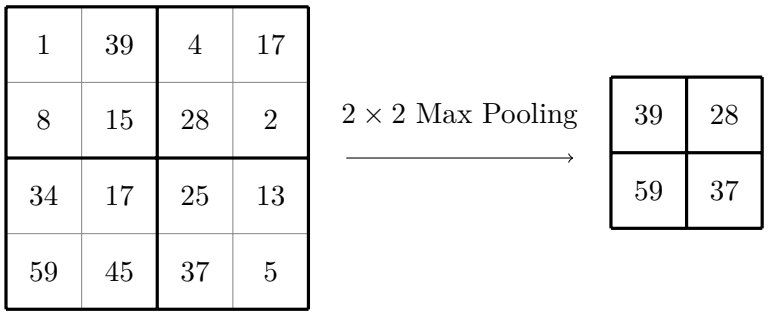


Obr. 3.7: Konvolučná vrstva s niekoľkými filtrami. Výstup obsahuje jeden kanál pre každý filter, ilustrácia prevzatá z [6].

Pooling

Poolingová vrstva redukuje rozmery vstupných dát kombinovaním niekoľkých vstupných bodov do jedného výsledného bodu. Daný bod potom predstavuje súhrnnú štatistiku svojho okolia. Typicky sa využíva funkcia *max pooling*, poprípade *average pooling*. Prehľad vybraných poolingových funkcií je dostupný v publikácii [15].

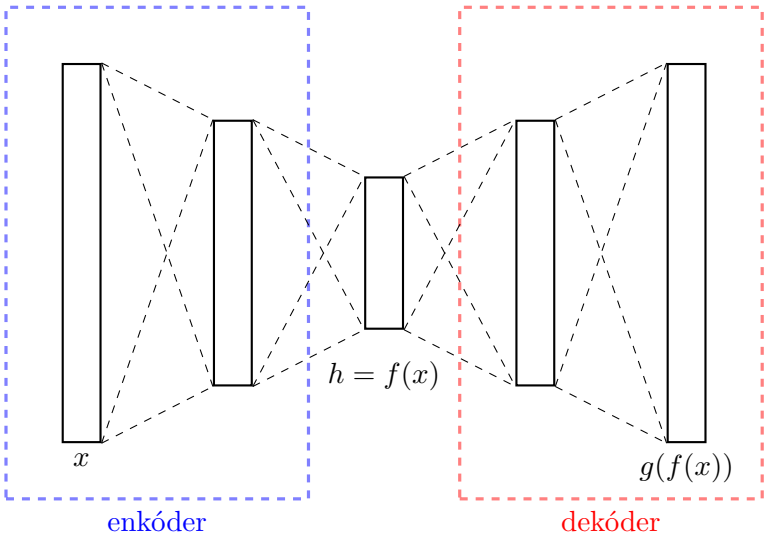
Použitie poolingových vrstiev pomáha zabezpečiť invariantnosť výstupu pri malých posunoch v rámci vstupných dát. Invariantnosť znamená, že v prípade malej zmeny vstupných dát, výstup poolingovej vrstvy zostane približne rovnaký [16].



Obr. 3.8: Max pooling funkcia pre 2-D dáta s veľkosťou okna 2×2 a rovnako veľkým krokom.

3.4 Autoenkódery

Ďalšou architektúrou neurónových sieti je *autoenkóder*. Vstup autoenkóderov je najprv komprimovaný do nízkorozmernej reprezentácie, označovanej ako *kód* alebo *skrýta premenná* (latent variable). Časť siete, ktorá zabezpečuje túto funkcionalitu sa nazýva *enkóder*. Druhá časť siete, nazývaná ako *dekóder*, mapuje nízkorozmernú reprezentáciu na výstup. Invertuje operácie, ktoré vykonal enkóder, aby zrekonštruoval pôvodný vstup.



Obr. 3.9: Znázornenie architektúry autoenkóderu. x predstavuje vstup, h je skrytá premenná alebo kód, $g(f(x))$ je rekonštruovaný vstup.

Bežný autoenkóder kopíruje svoj vstup na výstup, čo samo o sebe nie je veľmi užitočné. V takomto prípade bývajú rozmery skrytej premennej obmedzené na veľkosť menšiu, ako sú rozmery vstupných dát. Autoenkóder je teda nútený naučiť sa reprezentovať vstupné dáta a zachytiť ich charakteristické vlastnosti tak, aby bol schopný, čo najpresnejšie zrekonštruovať vstup pri dekódovaní [16].

Generatívne modely, ktoré odvádzajú a využívajú skryté premenné je taktiež možné považovať za určitý typ autoenkóderu [16].

Odšumovacie autoenkódery
Odšumovací autoenkóder je autoenkóder, ktorého vstupom sú poškodené dáta a je trénovaný, tak aby bol schopný predikovať pôvodné nepoškodené dáta. Typicky sa pri tréningu autoenkóderov minimalizuje funkcia

$$L(x, g(f(x))),$$

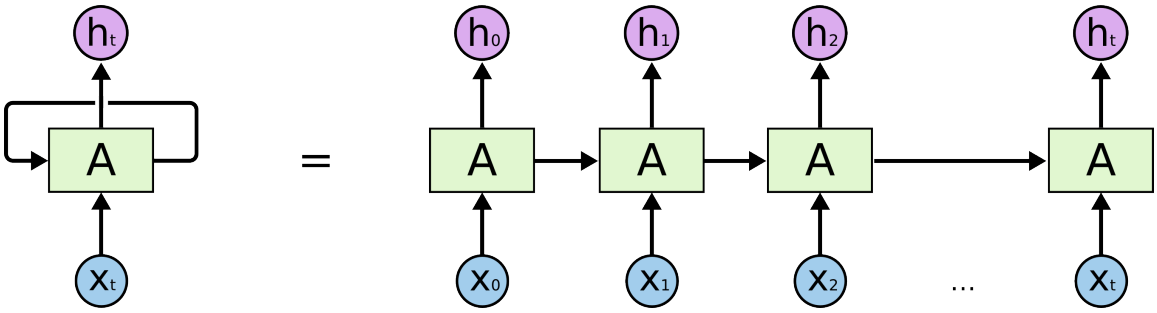
kde L je stratová funkcia penalizujúca rozdiely medzi x a $g(f(x))$, x sú vstupné dáta, $f(x)$ predstavuje výstup enkóderu a $g(x)$ je výstupom dekóderu [16]. Takýto autoenkóder sa naučí kopírovať svoj vstup na výstup. Vstup autoenkóderu x je možné degradovať napr. pridaním šumu. Degradovaný vstup je označený ako $\tilde{x}$ a minimalizuje sa teda funkcia

$$L(x, g(f(\tilde{x}))).$$

3.5 Rekurentné neurónové siete

Rekurentné neurónové siete (recurrent neural networks, RNN) sú trieda neurónových sietí slúžiacich na spracovávanie sekvenčných dát. Rozdiel medzi bežnými neurónovými sieťami a RNN spočíva v tom, akým spôsobom sa v nich šíri informácia pri spracovávaní vstupu. Tradičné neurónové siete sú acyklické, zatiaľ čo RNN obsahujú cykly, vďaka ktorým si sieť dokáže spätne predať určitú informáciu. Uchovávanie informácií v sieti umožňuje RNN pri spracovávaní aktuálneho vstupu vziať do úvahy aj predošlé vstupy.

Pri zafixovanom počte časových krokov, t, je možné si predstaviť RNN ako niekoľko inštancií bežných neurónových sietí, kde každá z nich predáva určitú informáciu svojmu následníkovi.



Obr. 3.10: RNN rozvinutá v čase. Ilustrácia prebraná z [35].

Počas tréningu RNN je využívaný algoritmus *BPTT* (backpropagation through time) ako variácia bežnej spätnej propagácie, spomínanej v sekcii 3.2. Algoritmus BPTT je založený na rozvinutí rekurentnej neurónovej siete v čase, podrobný popis algoritmu je dostupný v [16].

LSTM siete

RNN sú schopné naučiť sa využívať informácie z predchádzajúcich vstupov. V praxi však môže nastať problém, ak rozostup medzi potrebnou informáciou z minulých vstupov a vstupom, kedy danú informáciu treba využiť, je príliš veľký. Z tohto dôvodu boli navrhnuté siete *LSTM* (long short term memory) ako špeciálny typ rekurentných sietí, ktoré sú schopné naučiť sa uchovávať dlhodobý kontext. Kľúčovou vlastnosťou sietí LSTM je, že jednotlivé bunky siete majú stav. Do stavu bunky je možné pridávať, či z neho odstraňovať vybrané informácie. Tento proces je riadený pomocou niekoľkých „brán“, ktoré určujú, aké informácie treba odstrániť, a aké informácie budú do stavu pridané. Brány taktiež riadia výpočet výstupu danej bunky na základe jej stavu a vstupu. Hlbší popis jednotlivých brán a rôznych variácií LSTM je dostupný v článku [35].

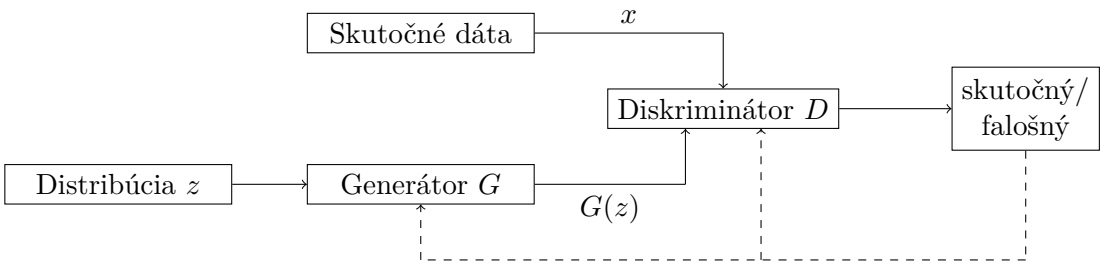
Ďalšou variantou sú obojsmerné LSTM siete, nazývané *BLSTM* (bi-directional long short term memory), tvorené dvoma nezávislými RNN sieťami. Jedna z daných sietí prijíma sekvenčný vstup v typickom poradí, pričom vstup druhej z týchto sietí je prevrátený. Táto vlastnosť dovoľuje sieti pri spracovávaní aktuálnej informácie využiť kontext z predošlých ale aj z následujúcich vstupov. Výstupy týchto sietí sú následne kombinované, typicky pomocou konkatenácie alebo sčítaním.

Vzhľadom na to, že audio signáli majú štruktúru sekvenčných dát, je možné na ich spracovanie využiť architektúry RNN, či LSTM. Využívanie kontextu z predošlých vstupov je veľkou výhodou pri odstraňovaní nestacionárnych šumov alebo reverberácie [55].

3.6 Generatívne neurónové siete

Generatívne neurónové siete (generative adverserial network, GAN) patria do triedy generatívnych modelov predstavených v roku 2014 [17].

Fungujú na princípe vzájomného súperenia dvoch modelov v hre s nulovým súčtom. Prvý model, nazývaný *generátor*, G, sa snaží na základe určitej skrytej distribúcie vytvoriť čo najreálnejšie vyzerajúce dáta. Druhý model, *diskriminátor*, D, dostane reálne dáta spolu s vygenerovanými a snaží sa ich odlíšiť.



Obr. 3.11: Štruktúra GAN. Ilustrácia inšpirovaná z [54].

Generátor nemá prístup k reálnym dátam a učí sa len na základe jeho interakcie s diskriminátorom – snaží sa ho oklamať, aby nad ním vyhral. Diskriminátor má prístup k reálnym aj vygenerovaným dátam. Tréning týchto modelov je možné zaradiť do metód strojového učenia bez učiteľa, keďže dáta nie je potrebné nijak anotovať a stačí vedieť, do akej kategórie patria. Diskriminátor je trénovaný, tak aby maximalizoval svoju presnosť rozlišovania medzi skutočnými dátami x alebo vygenerovanými dátami $G(z)$ [54]. Generátor je trénovaný tak,

aby minimalizoval $log(1 - D(G(z))$, kde $D(x)$ predstavuje pravdepodobnosť, že vstup bude zaradený skôr ku skutočným dátam ako k vygenerovaným [54].

Diskriminátor aj generátor sú trénovaný zároveň, pričom ich optimalizácia sa strieda. V prvom rade je zafixovaný generátor a maximalizuje sa presnosť diskriminátoru. Diskriminátor je následne zafixovaný a optimalizuje sa generátor. Ideálne je diskriminátor trénovaný až pokiaľ dosiahne optimum vzhľadom na aktuálny generátor, následne sa aktualizuje generátor. V praxi sa však diskriminátor trénuje iba počas niekoľkých iterácií a spolu s ním je aktualizovaný aj generátor [7]. Pri tréningu architektúr GAN sa môžu vyskytnúť určité problémy, ktoré sú bližšie popísané spolu s riešeniami v publikácii [7].

GAN v kontexte odšumovania audia

Pôvodne sa neurónové siete typu GAN využívali hlavne v oblasti počítačového videnia pre generovanie realistických obrazov. Neskôr sa však úspešne uplatnili aj v oblasti odšumovania audia. Jednou z prvých architektúr navrhnutých na tento účel je model SEGAN (speech enhancement generative adverserial network) [37]. Ďalej boli navrhnuté napr. modely MetricGAN [11], či MetricGAN+ [13], ktorý dosiahol vrcholové výsledky.

Dané modely využívajú pre generáciu odšumeného audio signálu variantu CGAN (conditional GAN), kde sa generátor učí mapovať poškodený vstup na čistý výstup, to znamená, že pre generovanie nevyužíva len distribúciu z, ale berie do úvahy aj určitú reprezentáciu zašumeného vstupu.

Kapitola 4

Implementácia

Kapitola popisuje nástroje použité na trénovanie neurónových sietí v tejto práci a taktiež obsahuje prehľad implementovaných modulov.

4.1 SpeechBrain

SpeechBrain [41] je open source sada nástrojov slúžiacich na spracovávanie reči, založená na rámcovom riešení PyTorch [38]. SpeechBrain ponúka mnoho technológií, medzi ktorými je aj podpora pre odšumovanie audia. Typický príkaz pre spustenie experimentu má formu

`python train.py hparams.yaml`,

kde `train.py` je skript pre tréning modelu a `hparams.yaml` je súbor obsahujúci konfiguráciu hyperparametrov.

Špecifikácia hyperparametrov
SpeechBrain oddeľuje definíciu hyperparametrov a ďalších metadát, od samotných trénovacích algoritmov. Výsledný kód je tým pádom prehľadnejší a kompaktnejší. Na definíciu hyperparametrov je využívaný formát HyperPyYAML, ktorý rozširuje funkcionalitu bežného YAML formátu.

Rozšírenie umožňuje v konfiguračnom súbore pomocou pridaných značiek špecifikovať objekty, moduly, triedy, či funkcie ako hodnoty parametrov. Taktiež je možné sa odkazovať na iné parametre v rámci súboru, zahrnúť ďalší konfiguračný súbor, vytvoriť kópiu objektu, alebo vykonať funkciu a jej výsledok využiť ako hodnotu parametru.

```
data_dir: ./data
train_clean: !apply:os.path.join
  - !ref <data_dir>
  - clean_trainset_28spk_wav
```

Obr. 4.1: Ukážka časti konfiguračného súboru. Definícia cesty k trénovacím dátam pomocou volania funkcie `os.path.join` so špecifikovanými argumentami. V rámci argumentov je využitý odkaz na iný parameter.

Skript pre trénovanie
Skript využíva funkcie a triedy definované v konfiguračnom súbore na natrénovanie modelu.

V rámci skriptu je definovaná postupnosť funkcií pre spracovanie dát (data processing pipeline), výpočet výstupného signálu a výpočet stratovej funkcie.

Využívané dáta je vhodné popísať pomocou anotačných súborov vo formáte JSON alebo CSV. V popise dát je potrebné uviesť aspoň jednoznačnú identifikáciu jednotlivých vzoriek, ďalej je možné definovať cestu k daným súborom, dĺžku nahrávok, identifikáciu hovoriaceho a pod. Typicky sa anotačné súbory vygenerujú pomocou skriptu pre prípravu dát. Vďaka dynamickému spracovaniu dát je v hlavnom skripte možné uložené dáta počas tréningu ľubovoľne transformovať.

V trénovacom skripte býva definovaná trieda, ktorá je potomkom poskytnutej triedy `Brain`. V nej je možné prepísať generické metódy na vlastné a presne určiť, čo sa bude vykonávať pri jednotlivých fázach tréningu a validácie. Trieda `Brain` a samotný proces tréningu je detailnejšie popísaný v sekcii 4.2.

SpeechBrain umožňuje jednoduché vytváranie záloh počas tréningu pomocou objektu triedy `Checkpointer`. Pri jeho vytváraní je potrebné špecifikovať adresár, do ktorého budú zálohy uložené. Taktiež je možné pomocou argumentov určiť, aké objekty budú počas tréningu ukladané, objektami môže byť napr. samotný model, počet vykonaných epochov a pod. Ukladanie záloh umožňuje obnoviť tréning modelu od určitého bodu v prípade, že skript bol nečakane ukončený. Zálohovanie prebieha po dokončení trénovacieho epochu a vyhodnotení modelu na validačnej sade dát, pričom dané vyhodnotenie je uložené spolu s modelom. Po príliš dlhom trénovaní je možné, že vyhodnotenie modelu sa postupne zhoršuje kvôli pretrénovaniu, takto je možné vybrať z uložených modelov ten, ktorého hodnotenie je najlepšie. Taktiež nie je potrebné uchovávať na disku všetky zálohy, no stačí, ak je uložených len niekoľko najlepšie ohodnotených modelov.

4.2 Proces tréningu modelu v SpeechBrain

V tejto práci je využitý model s architektúrou GAN, bližšie popísaný v sekcii 6.1. Pre tréning takéhoto modelu je potrebné prepísať pomerne veľa generických metód poskytovaných triedou `Brain`, keďže je nutné zohľadniť súbežný tréning modelu diskriminátora aj generátora.

Bežne tréning začína volaním metódy `fit()` na objekte triedy `Brain`. Tu sa v prvom rade zavolá metóda `make_dataloader()`, ktorá zabezpečí načítanie dát v správnom formáte a metóda `on_fit_start()`, ktorá pripraví model na tréning a inicializuje ďalšie potrebné súčasti. Potom sa začne samotný tréning zložený z niekoľkých epochov. V rámci jedného epochu je najprv zavolaná metóda `on_stage_start()`, kde je možné inicializovať premenné použité v danom epochu. Následne sa iteruje cez všetky batch-e trénovacej dátovej sady, pričom zakaždým je volaná metóda `fit_batch()`, ktorá vypočíta výstup modelu, vyhodnotí stratu pomocou metódy `compute_objectives()` a aktualizuje parametre modelu. Po spracovaní celej trénovacej sady je volaná metóda `on_stage_end()`, kde je možné spracovať štatistiky z tréningu, či uložiť model. Celý proces sa v zápätí opakuje pre validačnú dátovú sadu až na to, že tu typicky nedochádza k aktualizácií parametrov modelu. Pre rozlíšenie medzi jednotlivými etapami prijímajú spomínané metódy parameter `stage`, ktorý môže nadobúdať jednu z hodnôt `Stage.TRAIN`, `Stage.VALID` alebo `Stage.TEST`.

Obr. 4.2: Algoritmus trénovania modelu pomocou triedy Brain. Ilustrácia bola prebratá z dokumentácie SpeechBrain [41].

Modifikácia tréningu pre architektúry GAN

Pre trénovanie modelu MetricGAN+ [13] bolo potrebné upraviť metódu `on_stage_start` tak, aby v prvom rade prebehol tréning modelu diskriminátora a následne prebehol tréning modelu generátora. Trénovaný model je bližšie popísaný v sekcii 6.1, no hlavnou podstatou je, že diskriminátor je optimalizovaný tak, aby bol schopný ohodnotiť dáta podľa určitej cieľovej metriky (v tomto prípade skóre PESQ) a počas tréningu modelu generátora je cieľom optimalizovať danú cieľovú metriku.

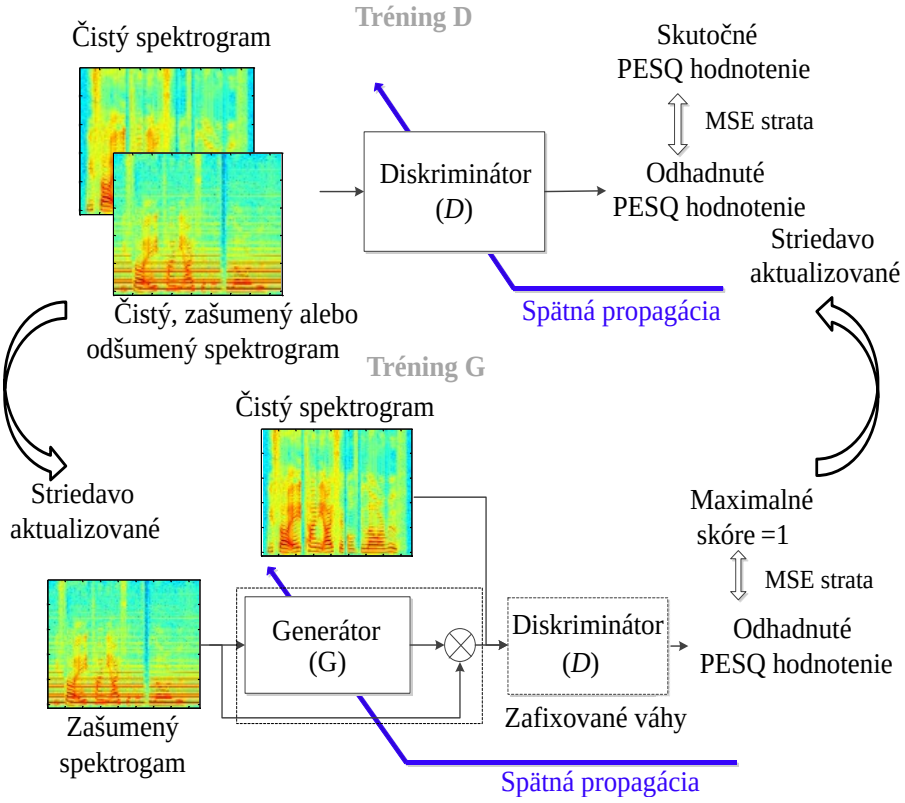
Tréning diskriminátora má na starosti pomocná metóda `train_discriminator`. V rámci epochu je diskriminátor trénovaný najprv na súčasných dátach. Následne je diskriminátor trénovaný na podmnožine historických dát z minulých epochov, aby sa predišlo katastrofickému zabúdaniu. A v poslednom rade je diskriminátor opäť trénovaný na súčasných dátach. Tréning modelu diskriminátora na súčasných dátach zahŕňa optimalizáciu predikcie cieľovej metriky na čistých audio nahrávkach, na zašumených audio nahrávkach a taktiež na nahrávkach vylepšených modelom generátora.

Po skončení tréningu diskriminátora je na rade tréning modelu generátora. Vstupom generátora sú zašumené audio nahrávky a jeho výstupom sú vylepšené verzie daných nahrávok. Pôvodné a odšumené nahrávky sú následne predané na vstup diskriminátora. Generátor je trénovaný tak, aby optimalizoval ohodnotenie vypočítané modelom diskriminátora.

Vzhľadom na to, že je potrebné počas jedného epochu rozlíšiť medzi jednotlivými fázami tréningu, bol v triede `Brain` zavedený stav, pomocou inštančnej premennej výčtového typu `EpochStage`, ktorý môže nadobudnúť jednu z hodnôt `GENERATOR` (tréning generátora), `DISC_CURRENT` (tréning diskriminátora na súčasných dátach) a `DISC_HISTORICAL` (tréning diskriminátora na historických dátach). Na základe tohto stavu sa vetvia metódy ako `fit_batch()`, či `compute_objectives()` tak, aby optimalizácia modelu a výpočet straty prebehol správne vzhľadom na aktuálny kontext.

Algoritmus 1 Tréning modelu s historickou dátovou sadou

```
for epoch = 1 to num_epochs do
    # Tréning D na súčasných nahrávkach
        tréning na čistých nahrávkach;
        tréning na odšumených nahrávkach;
        uloženie odšumených nahrávok a skutočného hodnotenia do hist. sady;
        tréning na zašumených nahrávkach;
    # Tréning D na historických nahrávkach
        hist_samples ← náhodný výber 20% nahrávok z hist. sady;
        tréning na nahrávkach z hist_samples;
    # Tréning D na súčasných nahrávkach
        opakuj kroky 1-4;
    # Tréning G
        tréning modelu generátora;
end for
```



Obr. 4.3: Striedavý tréning modelu generátora a diskriminátora. Obrázok je preložený z pôvodného článku [13].

4.3 Implementované moduly

Väčšina projektov, využívajúcich SpeechBrain [41], má pomerne typickú štruktúru trénovacích skriptov, ktorá je často označovaná ako trénovací *recept*. Väčšinou sa skladá z 2 skriptov pre prípravu dát a pre tréning modelu v jazyku Python. Ďalej obsahuje YAML súbor s popisom všetkých hyperparametrov a ešte jeden súbor v jazyku Python, v ktorom je definová trieda reprezentujúca samotný model, využívaný počas tréningu.

Vzhľadom na to, že v tejto práci bolo vykonaných niekoľko experimentov, kde boli využité rôzne dátové sady, pričom niektoré z nich bolo potrebné vytvoriť, boli z praktických dôvodov jednotlivé skripty rozdelené do viacerých súborov.

Medzi moduly slúžiace na prípravu dát patria:

- **`data_prepare.py`** – Skript slúži na prípravu dát a vytváranie anotačných súborov. Umožňuje kombinovať letecké dáta s nahrávkami z iných dátových sád ako napríklad VoiceBank, či LibriSpeech, popísané v následujúcej kapitole 5.
- **`librispeech_prepare.py`** – Skript pre stiahnutie dátovej sady LibriSpeech.
- **`voicebank_prepare.py`** – Skript pre stiahnutie dátovej sady VoiceBank a následnú zmenu vzorkovacej frekvencie nahrávok z 48 kHz na 16 kHz.
- **`noise_from_vhf.py`** – Modul pre získanie hlukov z nahrávok z leteckej komunikácie. Využíva dodané prepisy nahrávok na identifikáciu úsekov bez reči, pričom tieto úseky následne uloží do špecifikovaného adresára. Uložené nahrávky je však vhodné manuálne skontrolovať, keďže prepisy nie sú dokonalé a v daných úsekoch sa stále môže vyskytnúť reč.
- **`noise_csv.py`** – Skript pre vytvorenie súboru v `csv` formáte, ktorý popisuje nahrávky šumov využitých pri dynamickom vytváraní zašumených nahrávok triedou `AddNoise`[1] implementovanou v SpeechBrain.

Pre tréning modelu sú využité následujúce moduly:

- **`train.py`** – Recept pre tréning modelu bežným spôsobom.
- **`train_iterative.py`** – Recept pre tréning modelu iteratívnym spôsobom. Definuje funkcie, volané po každej iterácii, ktoré aktualizujú dátovú sadu, tak aby obsahovala ďalšie čisté nahrávky z leteckej komunikácie, pričom tieto nahrávky sú odšumené aktuálne trénovaným modelom.
- **`train.yaml`** – Súbor obsahujúci definície hyperparametrov pre tréning a ďalšie pomocné premenné popisujúce uloženie dát, adresáre pre výstup a pod.
- **`Brain.py`** – Definuje triedu `ModelBrain`, prepisujúcu niektoré z metód poskytnutých triedou `Brain`, implementovanou v SpeechBrain.
- **`models/MetricGAN.py`** – Definuje triedy, predstavujúce trénovaný model, `Generator` a `Discriminator` inšpirované článkom MetricGAN+ [13]. Architektúra týchto modelov je detailnejšie popísaná v sekcii 6.1.

[1] https://speechbrain.readthedocs.io/en/latest/API/speechbrain.processing.speech_augmentation.html

Taktiež bol vytvorený modul `helper.py`, ktorý implementuje rôzne pomocné funkcie využívané vo viacerých moduloch. Pre vyhodnotenie natrénovaných modelov bol vytvorený Jupyter Notebook[2] s názvom `evaluate.ipynb`, ktorý načíta najlepšie ohodnotený checkpoint z tréningu daného modelu, ten využije na odšumenie testovacích nahrávok a po ich manuálnom ohodnotení následne vytvorí tabuľku zobrazujúcu priemerné dosiahnuté hodnotenia a ich štatistickú významnosť.

[2] https://jupyter.org/

Kapitola 5

Dátová sada

Následujúca kapitola je zameraná na dátové sady využité v procese tréningu. Popísaný je proces vytvárania jednotlivých sád, pôvod nahrávok v nich, hluky využité na vytvorenie zašumenej sady a ďalšie dôležité vlastnosti. Taktiež je spomenutá testovacia sada nahrávok z leteckej komunikácie, ktorá slúži na vyhodnotenie natrénovaných modelov.

5.1 VoiceBank-DEMAND

Pre tréning referenčného modelu bola využitá dátová sada VoiceBank-DEMAND [50], varianta s 28 hovoriacimi, podobne ako pri trénovaní modelu MetricGAN+ [13]. Nahrávky v tejto sade pochádzajú z VoiceBank korpusu [52], z ktorého bolo vybratých 28 anglicky hovoriacich rečníkov s podobným prízvukom. Na každého rečníka pripadá približne 400 nahovorených viet. Nahrávky majú vzorkovaciu frekvenciu 48 kHz, no pre efektívnejší tréning je počas prípravy dát znížená ich vzorkovacia frekvencia na 16 kHz.

Na tvorbu zašumených dát bolo použitých 10 typov šumu [51]. Dva z nich sú umelo vytvorené a zvyšných osem sú reálne šumy pochádzajúce z DEMAND databázy [47]. Umelo vytvorené boli napríklad hlasy ďalších rečníkov v pozadí. Z databázy šumov boli vybraté rôzne zvuky z domácnosti (kuchyňa), kancelárie (konferenčná miestnosť), verejných priestorov (jedálne, reštaurácie, stanice), zvuky z premávky (auto, metro), či hluk z ulíc (rušná križovatka) [51]. Pre tvorbu trénovacích dát boli použité štyri rôzne pomery signálu a šumu (signal to noise ratio, SNR) a to 0 dB, 5 dB, 10 dB, 15 dB [51]. Výsledkom je dátová sada zašumených nahrávok so 40 odlišnými hlukovými podmienkami (štyri rôzne hodnoty SNR a desať typov šumu) a teda na jedného rečníka pripadá približne desať viet pre každú z týchto podmienok. Celková dĺžka nahrávok v trénovacej sade je približne 13 hodín.

Testovacia sada sa skladá z nahrávok dvoch ďalších rečníkov z VoiceBank korpusu a piatich odlišných typov šumu z DEMAND databázy. Jedná sa o anglicky hovoriaceho muža a ženu s podobným prízvukom ako v sade trénovacej. Hluky v testovacích nahrávkach pochádzajú z domácnosti (obývačka), kancelárie (kancelárske priestory), premávky (autobus) a dva hluky z ulíc (otvorená terasa kaviarne a námestie) [51]. V prípade testovacej sady boli využité štyri trochu vyššie hodnoty SNR a to 2.5 dB, 7.5 dB, 12.5 dB, 17.5 dB [51]. Kombináciou jednotlivých možností teda vzniká 20 rôznych hlukových podmienok (štyri úrovne SNR a päť typov hluku), čo znamená, že na jedného rečníka, v daných podmienkach, pripadá približne 20 viet.

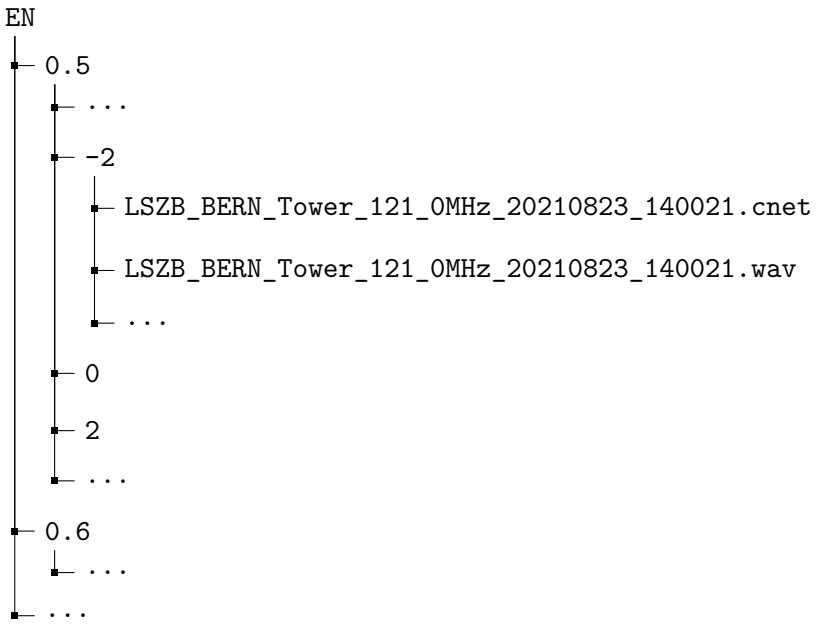
VoiceBank-DEMAND s reverberáciou

V niektorých experimentoch bola taktiež využitá dátová sada VoiceBank-DEMAND s reverberáciou [49]. Táto sada sa skladá z rovnakých čistých nahrávok ako dátová sada popísaná vyššie. Rozdiel spočíva v spôsobe, akým boli vytvorené zašumené dáta. Najprv prebehla konvolúcia čistej nahrávky s impulznou odozvou. Vybraný hluk bol taktiež konvoluovaný s rovnakou impulznou odozvou a následne boli tieto dva vytvorené signály spojené. Využité boli impulzné odozvy z výzvy ACE[1], databázy MIRD[2] a databázy MARDY[3].

5.2 Dáta z leteckej komunikácie

Dodané nahrávky zachytávajú komunikáciu pre riadenie leteckej prevádzky (ATC, Air traffic control). Tieto nahrávky sú typicky získané z rôznych rádiových kanálov, vysielajúcich vo VKV pásme (Veľmi Krátke Vlny, angl. VHF z Very High Frequency). Nahrávky majú väčšinou dĺžku niekoľkých sekúnd a vyskytuje sa v nich jeden, či dvaja rečníci. Ku každej nahrávke boli taktiež dodané automatické prepisy reči, vďaka ktorým z nich bolo možné extrahovať dlhšie úseky šumu na vytvorenie zašumených trénovacích dát.

Dáta boli dodané vo forme archívov, kde každý z nich obsahoval nahrávky zachytené za obdobie jedného kalendárneho mesiaca. Nahrávky spolu s prepismi boli rozdelené do zložiek, podľa ich odhadovaného SNR. Súbory s nahrávkami sú vo formáte `wav` a súbory s prepismi reči majú príponu `cnet`.



Obr. 5.1: Štruktúra archívu s dodanými nahrávkami.

Spolu s leteckými dátami bol dodaný aj estimátor SNR, ktorý bol využitý na roztriedenie nahrávok do jednotlivých skupín, implementovaný na základe článku [26].

[1] http://www.commsp.ee.ic.ac.uk/~sap/projects/ace-challenge/

[2] http://www.iks.rwth-aachen.de/en/research/tools-downloads/multichannel-impulse-response-database/

[3] http://www.iks.rwth-aachen.de/en/research/tools-downloads/multichannel-impulse-response-database/

Šum z leteckých dát

Spolu s nahrávkami komunikácie boli dodané aj rôzne šumy, ktoré sa v danom prostredí vyskytujú. Tieto nahrávky sa skladali napríklad zo statických šumov z lietadla, hlukov z letiska, motorov, či turbín lietadiel. Taktiež sem patrili šumy z rádiových kanálov. Dohromady tieto nahrávky mali dĺžku približne 10,5 hodín.

Ďalšie nahrávky šumu boli získané z dodaných dát z pasáži bez reči, pomocou skriptu `get_noise.py` popísaného v sekcii 4.3. Tu sa podarilo získať približne ďalších 2,5 hodín šumu.

Testovacia sada

Na to, aby bolo možné vyhodnotiť efektívnosť natrénovaných modelov bolo nutné definovať testovaciu sadu nahrávok. Pri daných nahrávkach je problém, že nie sú k dispozícii čisté dáta, aby bolo možné využiť objektívne metriky ako napríklad PESQ, kde je potrebná cieľová aj referenčná nahrávka.

Jednou z možností bolo umelo vytvoriť testovaciu sadu, no simulovať daný komunikačný kanál so všetkými nedokonalosťami je pomerne náročné. Z tohto dôvodu bola testovacia sada vytvorená len ako výber nahrávok z dodaných dát. Dodané nahrávky boli rozdelené do 5 skupín podľa ich odhadovanej hodnoty SNR a vznikli teda skupiny s hodnotami SNR -10 až -5, -5 až 0, 0 až 5, 5 až 10 a 10 až 15. Rozdelenie nahrávok podľa množstva šumu, ktoré obsahujú, jednoducho umožní pozorovať správanie natrénovaného modelu v rôznych prípadoch. Zo všetkých kandidátnych nahrávok pre danú skupinu bolo následne náhodne vybraných 25, ktoré budú tvoriť výslednú testovaciu sadu. Množstvo nahrávok bolo určené ako kompromis medzi dostatočne veľkým súborom, ktorý odpovedá skutočnosti a časovou náročnosťou vyhodnocovania, keďže nahrávky budú manuálne hodnotené.

V tomto prípade bude na hodnotenie nahrávok využitá metrika MOS, popísaná v sekcii 2.4. Priemerné hodnotenie jednotlivých skupín je uvedené v tabuľke 5.1. Nahrávky v testovacej sade budú odšumené danými natrénovanými modelmi a opäť ohodnotené. Následne bude možné porovnať nové hodnotenie nahrávok s hodnotením referenčným. Pre zistenie štatistickej významnosti týchto rozdielov bude využitý párový t-test[4] o 24 stupňoch voľnosti a hladinou významnosti $\alpha = 0.05$. Párový t-test porovnáva priemerné hodnoty dvoch populácií, kde vzorky z prvej populácie môžme jednoducho spárovať so vzorkami z populácie druhej. Typicky sa využíva v prípadoch, kde je daný objekt pozorovaný pred a po určitom zákroku. Z tohto dôvodu je test vhodný na určenie efektívnosti modelu, keďže je k dispozícii ohodnotenie jednotlivých vzorkov pred a po odšumení daným modelom. Nulová hypotéza H_0 hovorí, že stredné hodnoty dvoch porovnávaných populácií sú rovnaké. Alternatívna hypotéza H_1 teda tvrdí, že stredné hodnoty daných populácií sú rozdielne. V prípade, že pre danú skupinu nahrávok bude vypočítaná p-hodnota menšia alebo rovná α, zamietneme nulovú hypotézu H_0 na hladine významnosti α a príjmeme alternatívnu hypotézu H_1. Teda na zvolenej hladine významnosti platí, že skupina nahrávok má rozdielne priemerné hodnotenie pred a po odšumení. Následne je možné porovnať dané dve hodnoty a usúdiť, či sa priemerné hodnotenie zlepšilo alebo zhoršilo.

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html

Tabuľka 5.1: Priemerné hodnotenie MOS pre jednotlivé skupiny testovacích nahrávok.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS	1.16	1.04	1.40	1.52	2.04

5.3 LibriSpeech

V určitých experimentoch bola potrebná väčšia dátová sada ako pôvodná sada VoiceBank a to konkrétne v experimentoch popísaných v sekciách 6.4 a 6.5. Z tohto dôvodu bola v práci využitá taktiež dátová sada LibriSpeech [36]. Nahrávky v tejto sade majú podobný charakter ako nahrávky v spomínanej sade VoiceBank. Jedná sa o úseky audiokníh, pochádzajúcich z projektu LibriVox [5]. LibriVox poskytuje API, ktoré umožňuje získať údaje o rečníkoch a audioknihách, ktoré nahrali. Získané informácie boli následne porovnané s dátami poskytovaných z Internet Archive[6] a Project Gutenberg[7], vďaka čomu je možné získať prepisy spomínaných nahrávok. V procese vytvárania dátovej sady autori vyvinuli aplikáciu na zarovnanie prepisov a reči v nahrávkach. Daná aplikácia im zároveň umožnila získať informácie o pohlaví rečníkov a teda vo finálnej dátovej sade je rovnaký pomer mužských a ženských rečníkov.

Celá sada obsahuje približne 1000 hodín audio nahrávok a z praktických dôvodov bola rozdelená na niekoľko menšín podmnožín. Na rozdelenie do jednotlivých podmnožín bol využitý model na automatický prepis reči. Nahrávky boli zoradené na základe výsledkov, ktoré na nich daný model dosiahol a následne boli rozdelené na dve polovice. Nahrávky, na ktorých dosiahol model väčšiu úspešnosť boli označené ako „čisté“ (ang. clean) a druhá polovica dostala pomenovanie „ostatné“ (ang. other). Z „čistých“ nahrávok bolo náhodne vybratých 20 mužských a 20 ženských rečníkov na vytvorenie vývojovej sady a rovnaký proces bol využitý na vytvorenie testovacej sady. Zbytok „čistých“ nahrávok autori rozdelili do 2 skupín o dĺžke 100 a 360 hodín [36].

Pre nahrávky označené ako „ostatné“ bola taktiež vytvorená vývojová a testovacia sada. V tomto prípade však boli do spomínaných sád schválne vybraté náročnejšie nahrávky, t.j. nahrávky, kde úspešnosť modelu bola horšia [36].

Spolu teda bolo vytvorených 7 podmnožín tejto dátovej sady, pričom ich prehľad je uvedený v tabuľke 5.2. Pre tréning modelov v tejto práci bola využitá dátová sada označená ako `train-clean-500`, keďže nahrávky v nej sú menej zrozumiteľné , čo viac odpovedá dátam z leteckej komunikácie, na ktorých bude výsledný model aplikovaný.

[5] https://librivox.org/
[6] https://archive.org/
[7] https://www.gutenberg.org/

Tabuľka 5.2: Prehľad jednotlivých podmnožín dátovej sady [36].

názov sady	**celková dĺžka [h]**	**dĺžka na rečníka [m]**	**ženský rečníci**	**mužský rečníci**	**dohromady rečníkov**
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Kapitola 6

Experimenty

Táto kapitola obsahuje popis využitého modelu, vykonaných experimentov a ich vyhodnotenie.

6.1 Model

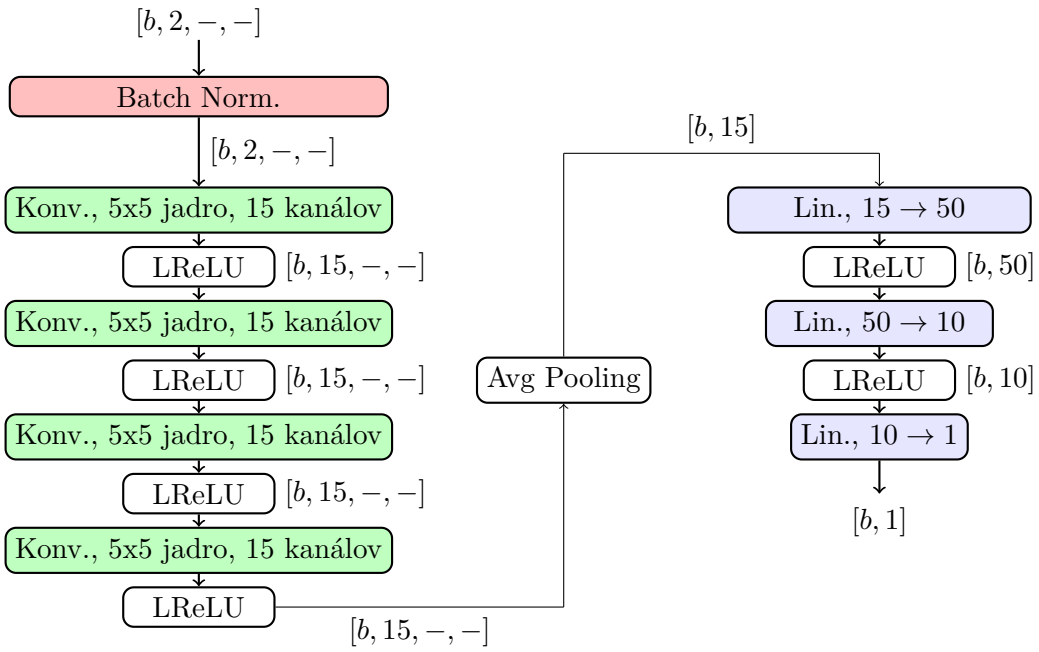
Trénovaný model bol inšpirovaný modelom MetricGAN+ [13]. Ako už jeho názov napovedá, tento model využíva architektúru GAN, popísanú v sekcii 3.6, čo znamená, že sa skladá z modelu diskriminátora a z modelu generátora.

V tomto prípade je ako diskriminátor využitá konvolučná neurónová sieť, ktorá sa skladá zo štyroch 2-D konvolučných vrstiev s pätnástimi kanálmi a konvolučným jadrom o veľkosti 5x5. Konvolúcia prebiehala s krokom nastaveným na hodnotu 1, bez výplne a teda veľkosť vstupných dimenzií je pomerne zachovaná. Na výstup každej konvolučnej vrstvy je aplikovaná aktivačná funkcia LReLU, popísaná v sekcii 3.2. Nasleduje pooling vrstva, ktorá pre každý z 15 kanálov vypočíta jeho priemernú hodnotu. Táto vrstva umožňuje modelu spracovávať vstup s variabilnou veľkosťou. Vypočítané priemerné hodnoty sú následne spracované troma plne prepojenými vrstvami s 50, 10 a 1 uzlom. Na výstupy prvých dvoch plne prepojených vrstiev je opäť aplikovaná aktivačná funkcia LReLU. Vstupom diskriminátora je odšumený spektrogram konkatenovaný spolu s referenčným, čistým spektrogramom. Výstupom poslednej vrstvy modelu je jeho aproximácia cieľovej metriky pre daný vstup, v tomto prípade sa jednalo o skóre PESQ.

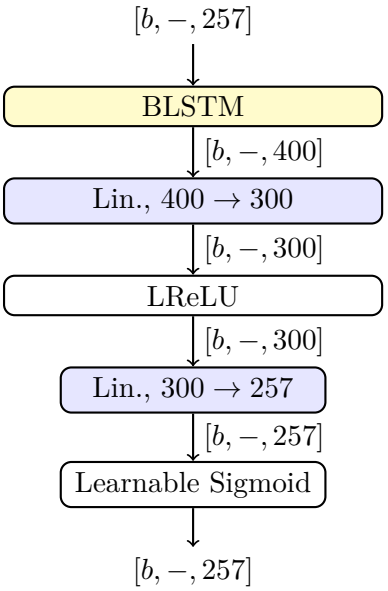
Generátor sa skladá z 2 obojsmerných LSTM vrstiev, popísaných v sekcii 3.5, pričom každá z nich obsahuje 200 uzlov. Za nimi sa nachádza jedna plne prepojená vrstva s 300 neurónmi, na výstup ktorej je aplikovaná aktivačná funkcia LReLU. Nasleduje ďalšia plne prepojená vrstva s 257 uzlami a aktivačná vrstva nazvaná *learnable sigmoid*. Tá funguje podobne ako bežná logistická aktivačná funkcia sigmoida, ktorá normalizuje vstup do intervalu $[0, 1]$. Funkcia však obsahuje parameter α, ktorý sa aktualizuje počas tréningu a teda má následovný predpis:

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

Výstupom generátora je maska, rovnako veľká ako vstupný spektrogram, ktorá by po vynásobení s pôvodným spektrogram mala odstrániť šum. Parameter α poslednej vrstvy zaručuje, že jednotlivé frekvencie spektrogramu majú svoju vlastnú normalizačnú funkciu, keďže šum aj reč sa môže správať odlišne pri nižších a vyšších frekvenciách.



Obr. 6.1: Model diskriminátora. Zobrazené sú jednotlivé vrstvy siete spolu s dimenziami vstupov/výstupov, pričom b označuje veľkosť batch-u a neuvedené dimenzie sa líšia v závislosti na veľkosti vstupného spektrogramu. Vstupom je odšumený spektrogram konkatenovaný spolu s referenčným spektrogramom, výstupom je aproximácia cieľovej metriky.

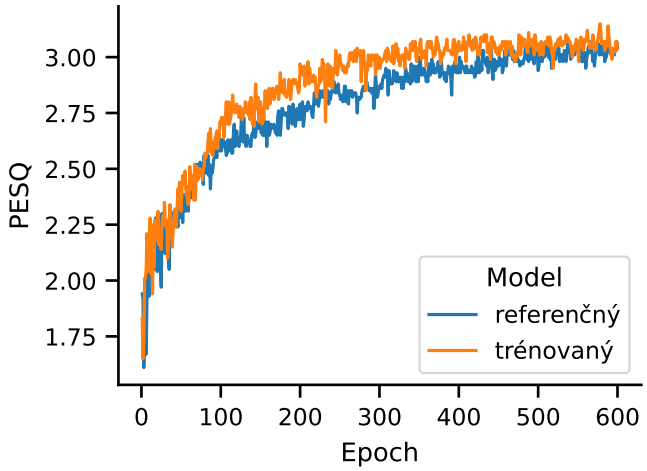


Obr. 6.2: Model generátora. Zobrazené sú jednotlivé vrstvy spolu s dimenziami vstupov/výstupov, kde b označuje veľkosť batch-u. Vstupom je spektrogram zašumeného audio signálu a výstupom je maska šumu pre daný spektrogram.

6.2 Referenčný model

V prvom rade bolo dôležité získať referenčné výsledky zlepšenia kvality signálu, aby bolo možné vyhodnotiť, aký efekt majú jednotlivé modifikácie procesu tréningu modelu. Pre získanie referenčných výsledkov bol využitý vyššie spomínaný model MetricGAN+. Aby bolo možné replikovať výsledky štúdie, autori sprístupnili svoj kód a výsledky tréningu verejnosti[1].

Tréning prebiehal na dátovej sade VoiceBank-DEMAND, popísanej v sekcii 5.1, pričom počas tréningu bolo PESQ skóre, popísané v sekcii 2.4, počítané priamo na testovacej sade a nie na validačnej. Tieto výsledky však neboli využité na aktualizovanie hodnôt trénovaného modelu.



Obr. 6.3: Porovnanie tréningu medzi trénovaným a referenčným modelom MetricGAN+ [13]. Vyššia hodnota PESQ skóre je lepšia.

Samotný tréning modelu prebiehal celkom podobne ako v pôvodnej štúdii. V rámci tejto práce sa podarilo dosiahnuť PESQ skóre 3.08 na testovacej sade VoiceBank-DEMAND. Pôvodný model MetricGAN+ dosiahol PESQ skóre 3.15, čo je približne o 2.27% viac. Tento rozdiel nie je až tak zásadný, keďže proces tréningu je do určitej miery ovplyvnení náhodou a cieľom bolo len získať určitý referenčný model, ktorý bude ďalej potrebné modifikovať pre odšumenie audia pochádzajúceho z VHF komunikácie.

Vyhodnotenie referenčného modelu na testovacích dátach

Natrénovaný model bol vyhodnotený na testovacej sade nahrávok z leteckej komunikácie, popísanej v sekcii 5.2. Z výsledkov v tabuľke 6.1 je vidno, že daný model nebol veľmi úspešný v zlepšení kvality zašumených nahrávok. V prípade nahrávok horšej kvality (odhadované SNR -10 až 0) bol šum potlačený a zrozumiteľnosť v nich bola o čosi zlepšená. Pre daný výber vzoriek však tieto zlepšenia neboli štatisticky významné. Na druhú stranu u nahrávok lepšej kvality (odhadované SNR 0 až 10) bola spolu so šumom potlačená aj samotná reč a výsledné ohodnotenie bolo horšie ako v prípade pôvodných nahrávok. V prípade nahrávok s odhadovaným SNR z intervalu 0 až 5 dosiahlo zhoršenie kvality štatistickej významosti.

[1] https://github.com/speechbrain/speechbrain/tree/develop/recipes/Voicebank/enhance/MetricGAN

Pre interval odhadovaného SNR 5 až 10 bolo toto zhoršenie na hrane štatistickej významnosti. Taktiež v prípade skupiny nahrávok s hodnotami SNR 10 až 15 boli určité nahrávky ohodnotené horšie ako nahrávky pôvodné.

Tabuľka 6.1: Vyhodnotenie referenčného modelu na testovacej sade nahrávok z leteckej komunikácie. Postup vyhodnotenia je popísaný v sekcii 5.2.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	1.16	1.04	**1.40**	1.52	2.04
MOS, odšumené	1.28	1.08	**1.16**	1.28	1.92
p-hodnota	0.08	0.33	**0.03**	0.06	0.27

Celkové priemerné hodnotenie pôvodných nahrávok je 1.43 a priemerné hodnotenie odšumených nahrávok je 1.34, čo znamená, že priemerná hodnotenie sa znížilo 0.94-krát.

6.3 Využitie dát z leteckej komunikácie

Ďalším krokom po získaní referenčných výsledkov bolo zakomponovať do tréningu dáta z leteckej komunikácie tak, aby trénovacia sada dát viac odpovedala realite. Z dodaných leteckých nahrávok boli vybrané tie, ktorých odhadované SNR boli najvyššie tak, aby nimi bolo možné nahradiť približne 25% trénovacích dát z pôvodnej dátovej sady. Tieto nahrávky boli následne považované za čisté dáta. Z čistých dodaných leteckých dát boli následne vytvorené zašumené nahrávky len pridaním bieleho šumu s hodnotami SNR 0 až 17.5. Zbytok trénovacej sady tvorili nahrávky z dátovej sady VoiceBank, popísanej v sekcii 5.1.

Tabuľka 6.2: Vyhodnotenie modelu trénovaného na dátovej sade, kde 25% nahrávok tvorili dáta z leteckej komunikácie, pričom na ich augmentáciu bol využitý biely šum. Postup vyhodnotenia je popísaný v sekcii 5.2.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	1.16	1.04	1.40	1.52	2.04
MOS, odšumené	1.20	1.12	1.36	1.64	2.04
p-hodnota	0.33	0.16	0.66	0.27	1.00

Z hodnotenia MOS v tabuľke 6.2 je vidno, že natrénovaný model sa správa o čosi lepšie ako pôvodný referenčný model. Zlepšenie kvality stále nedosiahlo štatistickej významnosti, no výsledky trendujú správnym smerom. Hlavným rozdielom oproti referenčnému modelu je, že v prípade nahrávok s vyšším SNR už nedochádza ku stlmeniu hovoriacich a k celkovému zhoršeniu kvality. Nové dosiahnuté priemerné hodnotenie MOS je 1.47 a teda sa podarilo dosiahnuť 1.03 násobné zlepšenie, čo však stále nie je dostatočné.

Pridanie reverberácie

Cieľom ďalšieho experimentu bolo zistiť, aký vplyv má pridanie reverberácie k zašumeným nahrávkam počas tréningu na výslednú kvalitu odšumených nahrávok. Namiesto pôvodnej dátovej sady VoiceBank-DEMAND, bola využitá sada VoiceBank s reverberáciou a šumom,

popísaná v sekcii 5.1. Podobne ako v predošlom experimente, 25% trénovacej dátovej sady tvorili čisté nahrávky z leteckej komunikácie, ku ktorým bol následne pridaný biely šum.

Výsledné odšumené nahrávky sa príliš nelíšili od nahrávok z predchádzajúceho experimentu bez reverberácie. Táto podobnosť sa odráža aj na ich priemernom hodnotení v tabuľke 6.3. Získané výsledky neboli veľmi prekvapujúce vzhľadom na to, že problémom väčšiny týchto nahrávok nie je ozvena, no skôr vysoká miera šumu, prerušovanie spojenia, či zachytávanie iných signálov, ktoré sú vysielané na podobných frekvenciách.

Tabuľka 6.3: Vyhodnotenie modelu trénovaného na nahrávkach s reverberáciou. Trénovacia sada sa skladala z 25% z nahrávok z leteckej komunikácie a zbytok tvorila dátová sada VoiceBank s reverberáciou a šumom, popísaná v sekcii 5.1. Postup vyhodnotenia je popísaný v sekcii 5.2.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	1.16	1.04	1.40	1.52	2.04
MOS, odšumené	1.20	1.12	1.36	1.60	2.08
p-hodnota	0.33	0.16	0.57	0.49	0.33

Priemerné hodnotenie MOS dosiahlo hodnoty 1.47, podobne ako v predchádzajúcom experimente a teda celkové zlepšenie bolo 1.03 násobné.

Využitie reálneho šumu

Následujúce experimenty nadväzujú na experiment zo sekcie 6.3, s tým rozdielom, že namiesto bieleho šumu je v degradovaných nahrávkach využitý reálny šum z leteckej VHF komunikácie, popísaný v sekcii 5.2.

V prvom experimente, 25% dátovej sady tvorili nahrávky z leteckej VHF komunikácie, zašumené skutočným šumom. Hodnoty SNR v zašumených nahrávkach boli v rozsahu -5 až 17,5. Zbytok trénovacej sady tvorili čisté a zašumené nahrávky z pôvodnej dátovej sady VoiceBank-DEMAND. Výsledky tohto experimentu sú ukázané v tabuľke 6.4. Zlepšenie kvality sa odrazilo na hodnotení MOS vo všetkých skupinách SNR, v prípade nahrávok s odhadovaným SNR -10 až -5, či -5 až 0 bolo dané zlepšenie kvality štatisticky významné s p-hodnotami 0.04 a 0.01 resp. Dosiahnuté priemerné hodnotenie MOS je 1.57, čo je 1.11-krát lepšie ako pôvodné hodnotenie 1.43.

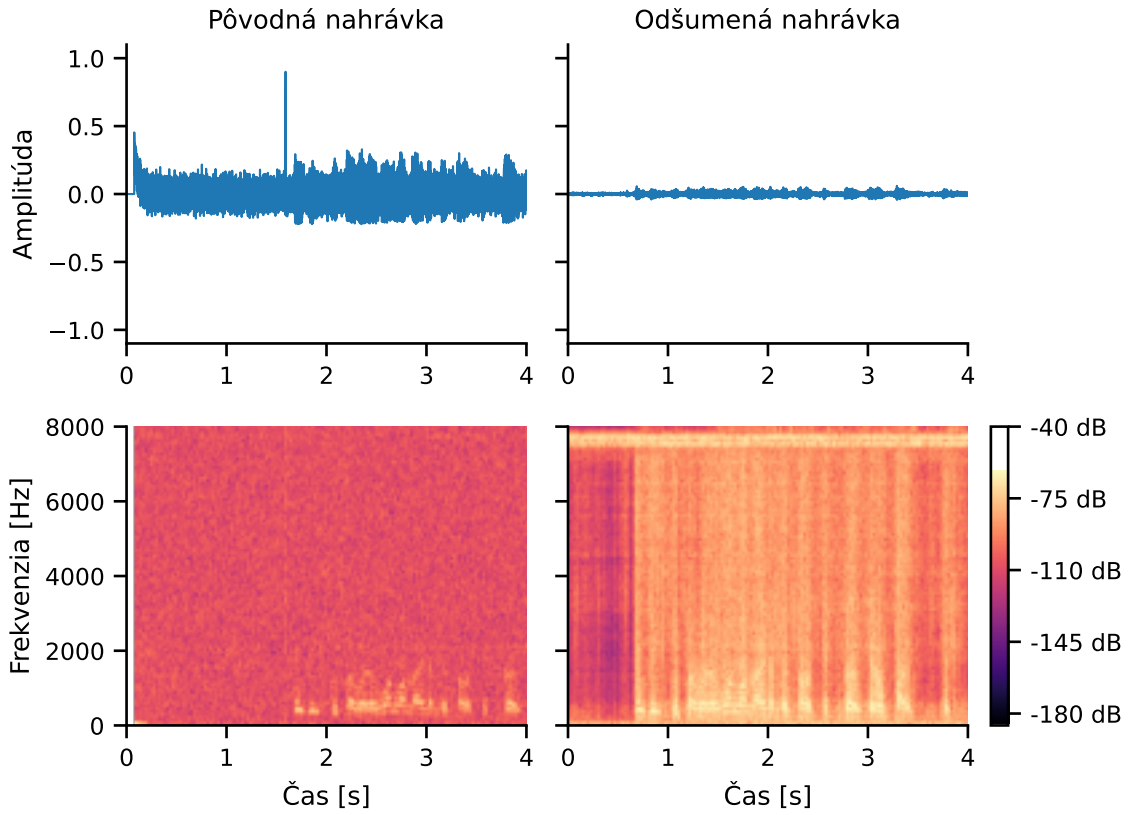
Tabuľka 6.4: Vyhodnotenie modelu trénovaného na dátovej sade, ktorá bola čiastočne zašumená reálnym hlukom z leteckej komunikácie. Postup vyhodnotenia je popísaný v sekcii 5.2.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	**1.16**	**1.04**	1.40	1.52	2.04
MOS, odšumené	**1.32**	**1.32**	1.48	1.72	2.08
p-hodnota	**0.04**	**0.01**	0.33	0.10	0.57

V druhom z týchto experimentov bol skutočný šum pridaný ku všetkým čistým nahrávkam. Čistá dátová sada bola rovnaká ako v predchádzajúcom experimente a teda 25% z nej tvorili čisté nahrávky z leteckej komunikácie a zbytok tvorila čistá dátová sada VoiceBank. Vyhodnotenie natrénovaného modelu je zobrazené v tabuľke 6.5. Oproti predchádzajúcim

experimentom model výrazne zlepšil kvalitu nahrávok. Pri oboch modeloch sa však často stalo, že hlasitosť rečníka, bola oproti pôvodnej nahrávke stlmená. Šum v pozadí bol však potlačený viac a teda celkový pocit z nahrávok bol oveľa príjemnejší. V prípade nahrávok horšej kvality, kde rečníka nebolo príliš, či vôbec počuť, bola väčšinou celá nahrávka potlačená a rečníkovi bolo možné rozumieť približne rovnako ako v pôvodnej nahrávke. Nahrávky s lepšou kvalitou zostali zachované.

V poslednom experimente sa podarilo dosiahnuť priemerného hodnotenie MOS 1.82, teda 1.27-krát lepšie hodnotenie oproti pôvodným nahrávkam. Uvedené výsledky boli pomerne očakávané, keďže model mal počas tréningu k dispozícií dáta, ktoré viac odpovedali prostrediu, v ktorom bude využívaný.



Obr. 6.4: Porovnanie nahrávky `LKTB_Approach_118_375MHz_20210729_070351.wav` pred a po odšumení modelom, trénovaným na dátovej sade celej zašumenej reálnym šumom z leteckej komunikácie, v časovej a časovo-frekvenčnej doméne.

Tabuľka 6.5: Vyhodnotenie modelu trénovaného na dátovej sade, ktorá bola celá zašumená reálnym hlukom z leteckej komunikácie. Postup vyhodnotenia je popísaný v sekcii 5.2.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	**1.16**	**1.04**	**1.40**	**1.52**	**2.04**
MOS, odšumené	**1.64**	**1.64**	**1.80**	**1.80**	**2.20**
p-hodnota	**0.00**	**0.00**	**0.00**	**0.05**	**0.04**

6.4 Väčšia dátová sada

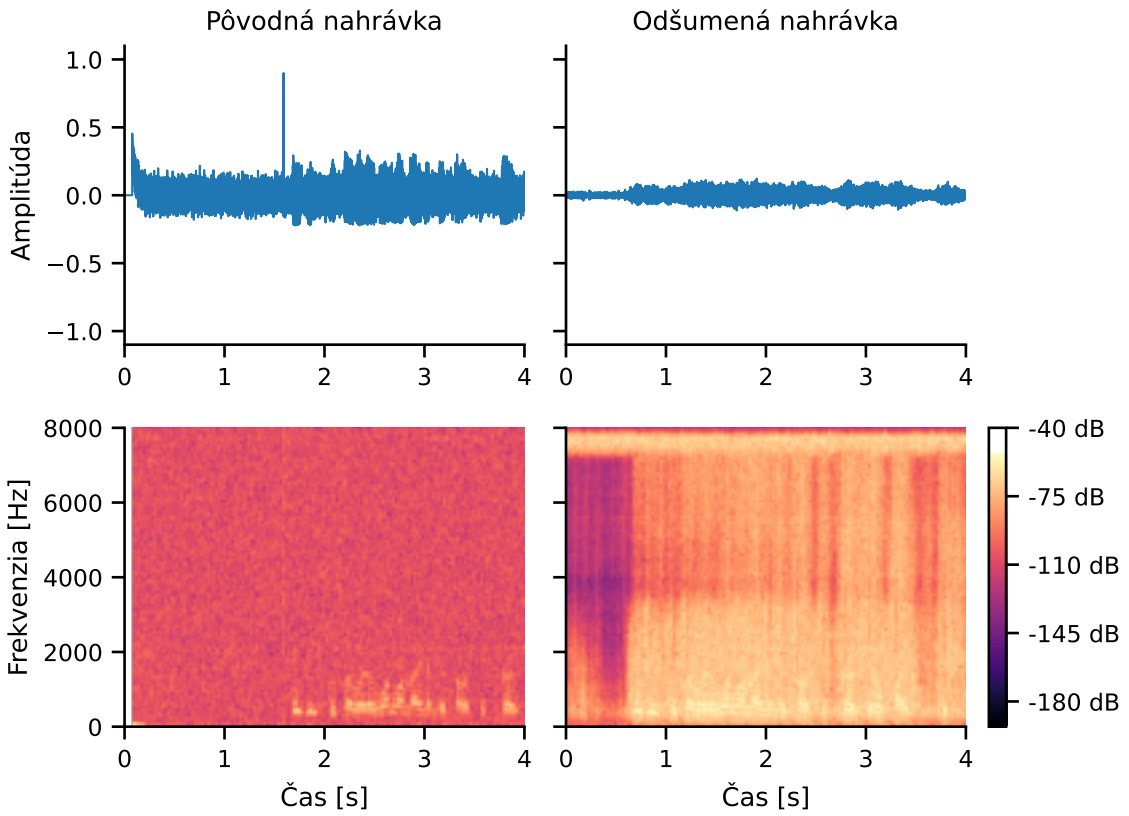
Z predchádzajúcich experimentov bol pravdepodobne najefektívnejší posledný model, teda model, trénovaný na dátovej sade tvorenej z 25% nahrávkami z leteckej komunikácie, pričom všetky čisté nahrávky boli znehodnotené skutočným šumom pochádzajúcim z daného prostredia. Z tohto dôvodu boli podobné parametre zvolené aj pre následujúci experiment. Namiesto 13 hodín nahrávok však bolo počas tréningu využitých až 24 hodín. Pomer čistých nahrávok bol zachovaný, štvrtinu tvorili nahrávky z leteckej komunikácie a zbytok tvorili nahrávky z dátovej sady LibriSpeech, popísanej v sekcii 5.3. Rovnako ako v predchádzajúcom experimente boli nahrávky zašumené šumom, špecifickým pre dané prostredie.

Vyhodnotenie tohto modelu je možné nájsť v tabuľke 6.6. Vo všetkých skupinách je vidno zlepšenie kvality. Oproti predchádzajúcemu modelu, však aktuálny model nepotlačoval hlasitosť rečníka, čo je možné vidieť na signáloch zobrazených v obrázkoch 6.5 a 6.4, keďže amplitúdy signálov neboli normalizované. V úsekoch nahrávok bez reči bol šum pomerne dobre potlačený. Pri niektorých nahrávkach však počas hovorenia bola hlasitosť šumu trochu vyššia, poprípade bolo v pozadí počuť rôzne artefakty. Nahrávky s dobrou kvalitou opäť zostali nezmenené. Celkové priemerné hodnotenie MOS odšumených nahrávok bolo 1.86, 1.30-krát lepšie ako hodnotenie pôvodných nahrávok.

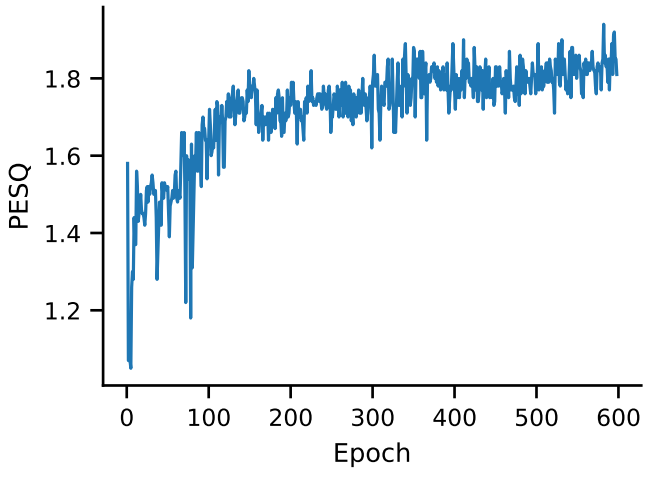
Tabuľka 6.6: Vyhodnotenie modelu trénovaného na zväčšenej dátovej sade. Postup vyhodnotenia je popísaný v sekcii 5.2.

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	**1.16**	**1.04**	**1.40**	**1.52**	**2.04**
MOS, odšumené	**1.60**	**1.60**	**1.88**	**1.96**	**2.24**
p-hodnota	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**

Vzhľadom na to, že počas tréningu diskriminátora je využívaná historická dátová sada, tvorená nahrávkami odšumenými generátorom v predchádzajúcich epochoch, bol tréning s ešte väčšími dátovými sadami pomerne problematický a zdĺhavý. Z tohto dôvodu neboli vykonané experimenty na väčších dátových sadách, poprípade s väčším modelom.



Obr. 6.5: Porovnanie nahrávky `LKTB_Approach_118_375MHz_20210729_070351.wav` pred a po odšumení modelom, trénovaným na väčšej dátovej sade, v časovej a časovo-frekvenčnej doméne.



Obr. 6.6: PESQ skóre modelu na validačnej dátovej sade počas tréningu.

6.5 Iteračný tréning

Jedným z problémov pri trénovaní na dátach pochádzajúcich z leteckej komunikácie je nedostatok čistých dát a hlukov z daného prostredia, z ktorých by bolo možné vytvoriť trénovaciu sadu. Z tohto dôvodu je v následujúcom experimente využitý iteračný tréning, pri ktorom je model trénovaný do určitého bodu a následne je využitý na vytvorenie nových čistých dát. Pri tvorení nových dát z leteckej komunikácie sú v prvom rade vybrané nahrávky s najlepším hodnotením SNR a tie sú následne odšumené pomocou doposiaľ natrénovaného modelu. Tréning modelu potom pokračuje na upravenej, zväčšenej dátovej sade a celý proces sa niekoľko krát opakuje.

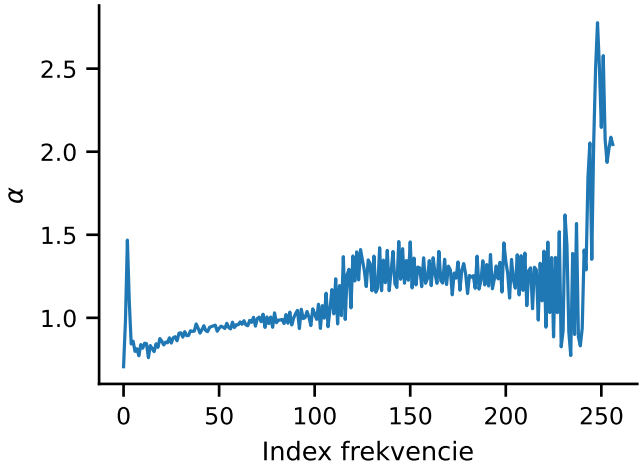
Z predchádzajúcich experimentov je vidno, že rast úspešnosti modelu na validačnej dátovej sade sa počas tréningu začína spomaľovať približne okolo 150 epochu. Úspešnosť modelu počas tréningu je uvedená napríklad na obr. 6.3, či 6.6. Z praktických dôvodov budú vykonané 4 iterácie tréningu po 150 epochoch, čo dohromady predstavuje 600 epochov. Prvá iterácia tréningu je pomerne podobná tréningu predchádzajúceho modelu, s tým rozdielom, že pomer trénovacích dát z leteckej komunikácie je o čosi menší. Počas prvej iterácie tvorilo trénovaciu dátovú sadu 22 hodín čistých nahrávok a 2 hodiny čistých nahrávok z leteckej komunikácie. Do druhej iterácie boli následne pridané ďalšie 4 hodiny odšumených dát z leteckej komunikácie. Pri tretej a poslednej, štvrtej iterácii bolo pridaných ďalších 8 a 16 hodín nahrávok z leteckej komunikácie resp.

Výsledky, ktoré model dosiahol na testovacej sade sú uvedené v tabuľke 6.7. Dosiahnuté hodnotenia MOS v jednotlivých skupinách SNR sú pomerne podobné ako hodnotenie predchádzajúce modelu s väčšou dátovou sadou. Model sa správal podobne ako v predchádzajúcom experimente, teda nepotlačoval hlasitosť rečníka a v úsekoch bez reči pomerne úspešne potlačil šum. Taktiež artefakty, ktoré sa vyskytovali pri predchádzajúcom modele neboli až tak výrazné a šum bol od reči o čosi lepšie oddelený, čo je možné pozorovať napríklad v odšumenej nahrávke na obr. 6.8 v porovnaní s obr. 6.5. Na druhú stranu bol tréning modelu iteratívnym spôsobom časovo náročnejší, kvôli rastúcej trénovacej dátovej sade. Celkové priemerné hodnotenie MOS je 1.88 a teda bolo dosiahnuté 1.31 násobné zlepšenie.

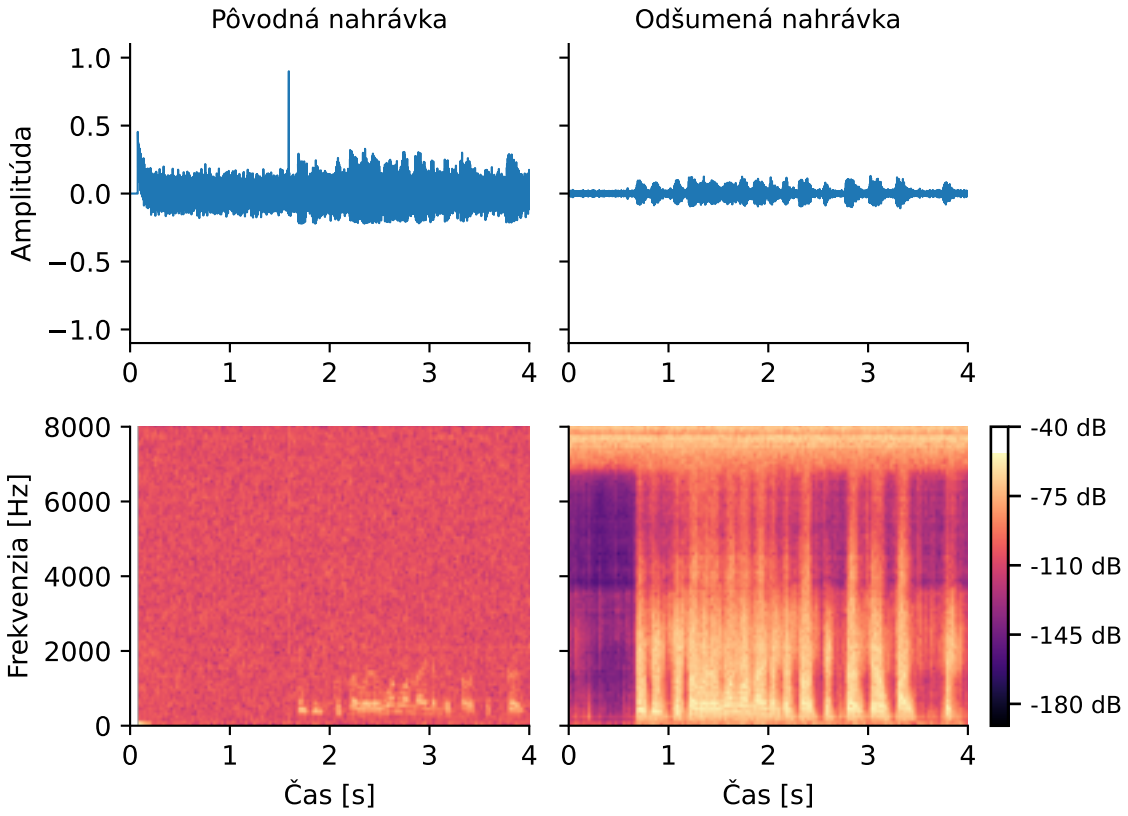
Tabuľka 6.7: Caption

SNR	-10 až -5	-5 až 0	0 až 5	5 až 10	10 až 15
MOS, pôvodné	**1.16**	**1.04**	**1.40**	**1.52**	**2.04**
MOS, odšumené	**1.66**	**1.54**	**1.92**	**2.02**	**2.26**
p-hodnota	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**

V spektrograme odšumenej nahrávky v obr. 6.8, je však stále vidno, že vyššie frekvencie nie sú potlačené tak, ako by mali. Tento jav sa prejavuje v nahrávkach slabými artefaktami v pozadí a môže byť spôsobený napríklad naučeným koeficientom α, poslednej vrstvy modelu generátora, keďže daný koeficient nadobúda pre dané frekvencie pomerne vysoké hodnoty. Naučené koeficienty sú zobrazené v obr. 6.7.



Obr. 6.7: Naučené koeficienty α pre jednotlivé frekvencie spektrogramu poslednej aktivačnej vrstvy modelu generátora, ktorá je bližšie popísaná v sekcii 6.1.



Obr. 6.8: Porovnanie nahrávky `LKTB_Approach_118_375MHz_20210729_070351.wav` pred a po odšumení modelom, trénovaným iteratívnym spôsobom. V časovej aj časovo-frekvenčnej doméne je vidieť lepšie oddelenie šumu od reči, ako v predchádzajúcich experimentoch (obr. 6.5 a 6.4).

Kapitola 7

Záver

7.1 Zhrnutie výsledkov

Cieľom tejto práce bolo vytvoriť model pre odšumenie a dereverberáciu nahrávok pochádzajúcich z leteckej komunikácie. Bolo vykonaných niekoľko experimentov na rôzne upravených dátových sadách a s rôznymi spôsobmi trénovania. Experimenty stavali na publikovanom článku [13], kde bol predstavený model MetricGAN+. Väčšina pokusov bola zameraná prevažne na odstránenie šumu, keďže ten bol hlavným rušivým faktorov pri počúvaní daných nahrávok. Toto sa podarilo s rôznou úspešnosťou, ako je možné vidieť v tabuľkách s výsledkami pri jednotlivých experimentoch. V rámci experimentov využívajúcich statickú dátovú sadu sa podarilo dosiahnuť 1.30 násobného zlepšenia hodnotenia MOS testovacích nahrávok a to v experimente so zväčšenou dátovou sadou, popísanom v sekcii 6.4.

Taktiež bol implementovaný iteratívny spôsob trénovania modelu, ktorého cieľom je získanie väčšieho množstva čistých dát pre vytvorenie trénovacej dátovej sady, v prípade, že čisté nahrávky nie sú k dispozícii. V danom experimente je trénovaný model využitý na odšumenie nahrávok, ktoré sú v následujúcej iterácii využité ako čisté trénovacie dáta. Týmto spôsobom sa podarilo dosiahnuť 1.31 násobného zlepšenia v hodnotení MOS a výsledné nahrávky boli o čosi príjemnejšie na posluch a šum v nich bol lepšie oddelený od reči. Iteratívny tréning modelu bol však časovo náročnejší.

7.2 Ďalší vývoj

Nižšie uvedené sú ďalšie možnosti trénovania modelov pre odšumenie audio nahrávok pochádzajúcich z leteckej komunikácie.

Historická dátová sada
Ďalej by bolo vhodné vyskúšať tréning na ešte väčšej dátovej sade, poprípade s väčším modelom. Tieto experimenty neboli vykonané, kvôli ich časovej náročnosti vzhľadom na rastúcu historickú dátovú sadu, ktorá je potrebná kvôli prevencii katastrofického zabúdania modelu. Tento problém by mohol byť vyriešený napríklad využitím inkrementálneho učenia, navrhnutého v článku [28].

Lepšia trénovacia sada
Pre vytvorenie realistickejších zašumených dát je potrebné získať viac šumu a hlukov z daného prostredia a to napríklad analýzou väčšieho množstva nahrávok a hľadaním úsekov bez reči.

Tréning bez čistých dát
Problematická je taktiež zlá dostupnosť čistých leteckých dát a šumu, z ktorých by bolo možné vytvoriť trénovaciu sadu čistých a zašumených nahrávok. Z tohto dôvodu by bolo vhodné zvoliť takú cieľovú metriku diskriminátora, kde pre výpočet nie je potrebná referenčná nahrávka. Týmto spôsobom by bolo možné optimalizovať daný model bez potreby čistých dát. Podobné experimenty boli vykonané napríklad v článku [14], kde autori využili ako cieľovú metriku diskriminátora hodnotenie získané modelom DNSMOS [42], vypočítané len na základe odšumenej nahrávky. V spomínanom článku model síce dosiahol o niečo horšie hodnotenie ako model trénovaný bežným spôsobom, t.j. s dostupnými čistými nahrávkami, no tento rozdiel by mohol byť vyrovnaný, napríklad využitím väčšej sady zašumených nahrávok, čo by v prípade záznamov z leteckej komunikácie nebol problém.

Vyhodnotenie modelov a cieľové metriky
Limitáciou daných experimentov je pravdepodobne spôsob hodnotenia testovacích nahrávok a v budúcnosti by bolo lepšie vykonať dané testy s väčšou skupinou poslucháčov. Poprípade sa pozrieť na koreláciu určitých objektívnych metrík so získaným hodnotením a následne dané metriky využiť na optimalizáciu generátora počas tréningu.

Literatúra

[1] Azarang, A. a Kehtarnavaz, N. A review of multi-objective deep learning speech denoising methods. *Speech Communication*. 2020, zv. 122, s. 1–10. DOI: https://doi.org/10.1016/j.specom.2020.04.002. ISSN 0167-6393. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0167639319304686.

[2] Bohra, Y. *Vanishing and exploding gradients in deep neural networks* [online], 18. júna 2021 [cit. 3.1.2022]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/06/the-challenge-of-vanishing-exploding-gradients-in-deep-neural-networks/.

[3] Boll, S. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*. IEEE. 1979, zv. 27, č. 2, s. 113–120.

[4] Braun, S. a Tashev, I. A consolidated view of loss functions for supervised deep learning-based speech enhancement. In: *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*. 2021, s. 72–76. DOI: 10.1109/TSP52935.2021.9522648.

[5] Brownlee, J. How to choose loss functions when training deep learning neural networks. [online]. Január 2019, revidované 25. 8. 2020, [cit. 31.12.2021]. Dostupné z: https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/.

[6] Buduma, N. a Locascio, N. *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly Media, 2017. ISBN 9781491925560. Dostupné z: https://books.google.sk/books?id=n0IlDwAAQBAJ.

[7] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. et al. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*. 2018, zv. 35, č. 1, s. 53–65. DOI: 10.1109/MSP.2017.2765202.

[8] Engelen, J. E. van a Hoos, H. H. A survey on semi-supervised learning. *Machine Learning*. Feb 2020, zv. 109, č. 2, s. 373–440. DOI: 10.1007/s10994-019-05855-6. ISSN 1573-0565. Dostupné z: https://doi.org/10.1007/s10994-019-05855-6.

[9] Ephraim, Y. a Malah, D. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE transactions on acoustics, speech, and signal processing*. IEEE. 1985, zv. 33, č. 2, s. 443–445.

[10] Erdogan, H., Hershey, J. R., Watanabe, S. a Le Roux, J. Deep Recurrent Networks for Separation and Recognition of Single-Channel Speech in Nonstationary

Background Audio. In: Watanabe, S., Delcroix, M., Metze, F. a Hershey, J. R., ed. *New Era for Robust Speech Recognition: Exploiting Deep Learning*. Cham: Springer International Publishing, 2017, s. 165–186. DOI: 10.1007/978-3-319-64680-0_7. ISBN 978-3-319-64680-0. Dostupné z: https://doi.org/10.1007/978-3-319-64680-0_7.

[11] Fu, S.-W., Liao, C.-F., Tsao, Y. a Lin, S.-D. MetricGAN: Generative Adversarial Networks based Black-box Metric Scores Optimization for Speech Enhancement. In: Chaudhuri, K. a Salakhutdinov, R., ed. *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 09–15 Jun 2019, sv. 97, s. 2031–2041. Proceedings of Machine Learning Research. Dostupné z: https://proceedings.mlr.press/v97/fu19b.html.

[12] Fu, S.-W., Tsao, Y., Lu, X. a Kawai, H. Raw waveform-based speech enhancement by fully convolutional networks. In: *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 2017, s. 006–012. DOI: 10.1109/APSIPA.2017.8281993.

[13] Fu, S.-W., Yu, C., Hsieh, T.-A., Plantinga, P., Ravanelli, M. et al. *MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement*. 2021.

[14] Fu, S.-W., Yu, C., Hung, K.-H., Ravanelli, M. a Tsao, Y. *MetricGAN-U: Unsupervised speech enhancement/ dereverberation based only on noisy/ reverberated speech*. arXiv, 2021. DOI: 10.48550/ARXIV.2110.05866. Dostupné z: https://arxiv.org/abs/2110.05866.

[15] Gholamalinezhad, H. a Khosravi, H. *Pooling Methods in Deep Neural Networks, a Review*, 16. septembra 2020. Dostupné z: https://arxiv.org/abs/2009.07485.

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

[17] Goodfellow, I. J., Pouget Abadie, J., Mirza, M., Xu, B., Warde Farley, D. et al. *Generative Adversarial Networks*. 2014.

[18] Graves, A., Mohamed, A.-r. a Hinton, G. Speech recognition with deep recurrent neural networks. In: Ieee. *2013 IEEE international conference on acoustics, speech and signal processing*. 2013, s. 6645–6649.

[19] Habets, E. *Fifty Years of Reverberation Reduction* [online]. AudioLabs, 2016 [cit. 29.12.2021]. Dostupné z: https://www.audiolabs-erlangen.de/content/05-fau/professor/00-habets/06-activities/Keynote_Habets_2016_AES_60.pdf.

[20] Hinton, G. E. a Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science*. 2006, zv. 313, č. 5786, s. 504–507. DOI: 10.1126/science.1127647. Dostupné z: https://www.science.org/doi/abs/10.1126/science.1127647.

[21] ITU T. *Recommendation P.862 (02/01): Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs* [online], 23. februára 2001 [cit. 2.1.2022]. Dostupné z: https://www.itu.int/rec/T-REC-P.862-200102-I/en.

[22] ITU T. *Recommendation P.800.2 (07/16): Mean opinion score interpretation and reporting* [online], 29. júla 2016 [cit. 1.1.2022]. Dostupné z: https://www.itu.int/rec/T-REC-P.800.2-201607-I/en.

[23] ITU T. *Recommendation BS.1284-2 (01/2019): General methods for the subjective assessment of sound quality* [online], 21. januára 2019 [cit. 1.1.2022]. Dostupné z: https://www.itu.int/rec/R-REC-BS.1284-2-201901-I/en.

[24] Kandel, I. a Castelli, M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*. 2020, zv. 6, č. 4, s. 312–315. DOI: https://doi.org/10.1016/j.icte.2020.04.010. ISSN 2405-9595. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2405959519303455.

[25] Kehtarnavaz, N. CHAPTER 7 - Frequency Domain Processing. In: Kehtarnavaz, N., ed. *Digital Signal Processing System Design (Second Edition)*. Second Edition. Burlington: Academic Press, 2008, s. 175–196. DOI: https://doi.org/10.1016/B978-0-12-374490-6.00007-6. ISBN 978-0-12-374490-6. Dostupné z: https://www.sciencedirect.com/science/article/pii/B9780123744906000076.

[26] Kim, C. a Stern, R. M. Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. 2008.

[27] Krizhevsky, A., Sutskever, I. a Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012, zv. 25, s. 1097–1105.

[28] Lee, C.-C., Lin, Y.-C., Lin, H.-T., Wang, H.-M. a Tsao, Y. *SERIL: Noise Adaptive Speech Enhancement using Regularization-based Incremental Learning*. arXiv, 2020. DOI: 10.48550/ARXIV.2005.11760. Dostupné z: https://arxiv.org/abs/2005.11760.

[29] Liew, S. S., Khalil Hani, M. a Bakhteri, R. Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing*. 2016, zv. 216, s. 718–734. DOI: https://doi.org/10.1016/j.neucom.2016.08.037. ISSN 0925-2312. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0925231216308797.

[30] Lim, J. S. a Oppenheim, A. V. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*. IEEE. 1979, zv. 67, č. 12, s. 1586–1604.

[31] Loizou, P. C. *Speech Enhancement: Theory and Practice*. 2. vyd. Boca Raton: CRC Press, 2013. ISBN 9781138075573.

[32] Narayanan, A. a Wang, D. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, s. 7092–7096. DOI: 10.1109/ICASSP.2013.6639038.

[33] Naylor, P. a Gaubitch, N. *Speech Dereverberation*. 1. vyd. Springer London, 2010. Signals and Communication Technology. ISBN 9781849960564. Dostupné z: https://books.google.sk/books?id=SYuTUqiB-q4C.

[34] Nielsen, M. A. *Neural networks and Deep learning*. Determination press San Francisco, CA, 2015.

[35] Olah, C. *Understanding LSTM networks*, 27. augusta 2015 [cit. 4.1.2022]. Dostupné z: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[36] Panayotov, V., Chen, G., Povey, D. a Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. 2015, s. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[37] Pascual, S., Bonafonte, A. a Serrà, J. *SEGAN: Speech Enhancement Generative Adversarial Network*. 2017.

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

[39] Pishro Nik, H. *Introduction to Probability, Statistics, and Random Processes*. 1. vyd. Kappa Research, LLC, 2014. ISBN 978-0990637202.

[40] Prabhu. *Understanding hyperparameters and its optimisation techniques* [online], 03. júla 2018 [cit. 31.12.2021]. Dostupné z: https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568.

[41] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S. et al. *SpeechBrain: A General-Purpose Speech Toolkit*. 2021. ArXiv:2106.04624.

[42] Reddy, C. K. A., Gopal, V. a Cutler, R. *DNSMOS: A Non-Intrusive Perceptual Objective Speech Quality metric to evaluate Noise Suppressors*. arXiv, 2020. DOI: 10.48550/ARXIV.2010.15258. Dostupné z: https://arxiv.org/abs/2010.15258.

[43] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*. 2015, zv. 115, č. 3, s. 211–252. DOI: 10.1007/s11263-015-0816-y.

[44] Sharma, S., Sharma, S. a Athaiya, A. Activation functions in neural networks. *Towards data science*. 2017, zv. 6, č. 12, s. 310–316. Dostupné z: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf.

[45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. a Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*. JMLR. org. 2014, zv. 15, č. 1, s. 1929–1958.

[46] Taal, C. H., Hendriks, R. C., Heusdens, R. a Jensen, J. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010, s. 4214–4217. DOI: 10.1109/ICASSP.2010.5495701.

[47] Thiemann, J., Ito, N. a Vincent, E. The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings. Montreal, Canada: [b.n.]. jún 2013. DOI: 10.5281/zenodo.1227120. The dataset itself is archived on Zenodo, with DOI 10.5281/zenodo.1227120. Dostupné z: https://hal.inria.fr/hal-00796707.

[48] Tu, Y.-H., Tashev, I., Zarar, S. a Lee, C.-H. A Hybrid Approach to Combining Conventional and Deep Learning Techniques for Single-Channel Speech Enhancement and Recognition. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, s. 2531–2535. DOI: 10.1109/ICASSP.2018.8461944.

[49] Valentini Botinhao, C. Noisy reverberant speech database for training speech enhancement algorithms and TTS models, 2016 [dataset]. [online]. University of Edinburgh: [b.n.]. 2017, [cit. 9.4.2022]. Dostupné z: https://doi.org/10.7488/ds/2139.

[50] Valentini Botinhao, C. Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. [online]. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR): [b.n.]. 2017, [cit. 13.2.2022]. Dostupné z: https://doi.org/10.7488/ds/2117.

[51] Valentini Botinhao, C., Wang, X., Takaki, S. a Yamagishi, J. Speech Enhancement for a Noise-Robust Text-to-Speech Synthesis System Using Deep Recurrent Neural Networks. 2016, s. 352–356. DOI: 10.21437/Interspeech.2016-159.

[52] Veaux, C., Yamagishi, J. a King, S. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. 2013, s. 1–4. DOI: 10.1109/ICSDA.2013.6709856.

[53] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A. et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*. 2010, zv. 11, č. 12.

[54] Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X. et al. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*. 2017, zv. 4, č. 4, s. 588–598. DOI: 10.1109/JAS.2017.7510583.

[55] Yuliani, A. R., Amri, M. F., Suryawati, E., Ramdan, A. a Pardede, H. F. Speech Enhancement Using Deep Learning Methods: A Review. *Jurnal Elektronika dan Telekomunikasi*. 2021, zv. 21, č. 1, s. 19–26.

Príloha A

Obsah priloženého pamäťového média

- `/`
  - `src/` ..... zdrojové kódy a dáta
    - `data/` ..... dátové sady, šumy
    - `figures/` ..... skripty pre tvorbu grafov
    - `models/` ..... popis modelu
    - `pretrained_models/` ..... model MetricGAN+ [13]
    - `results/` ..... výsledky experimentov
      - `iterative/4234/`
        - `vhf_examples/` ..... odšumené testovacie nahrávky
        - ...
      - ...
    - `snr_estimator/` ..... dodaný odhadovač SNR
    - `evaluate.ipynb` ..... notebook na vyhodnotenie modelu
    - `README.md` ..... inštalácia a spustenie
    - `requirements.txt` ..... potrebné Python knižnice
    - `*.py` ..... skripty pre tréning a tvorbu dát
  - `text_src/` ..... zdrojový tvar písomnej správy
  - `text.pdf` ..... písomná správa
  - `poster.pdf` ..... plagát

Príloha B

Plagát

Robustné odšumovanie a dereverberácia audia

Motivácia

Táto práca sa zaoberá odšumovaním a dereverberáciou audio signálov pochádzajúcich z leteckej VHF komunikácie pomocou strojového učenia. Degradácia týchto audio signálov nastáva hlavne kvôli vysokým hladinám rušivého hluku v kokpitoch lietadiel, poprípade vzniknutou ozvenou. Ďalším problémom je rušenie rádiových vĺn nevhodným počasím, alebo blokovanie komunikácie pohoriami.

Nahrávky z leteckej komunikácie

Model sa zamierava na odšumenie audio signálov pochádzajúcich z rôznych rádiových kanálov, vysielajúcich vo VKV pásme. Dané nahrávky zachytávajú komunikáciu pre riadenie leteckej prevádzky. Problémom využitia týchto dát pri tréningu je však neexistencia čistých nahrávok bez šumu, či šumu samotného, pomocou ktorého by bolo možné vytvoriť čistú a zašumenú trénovaciu sadu.

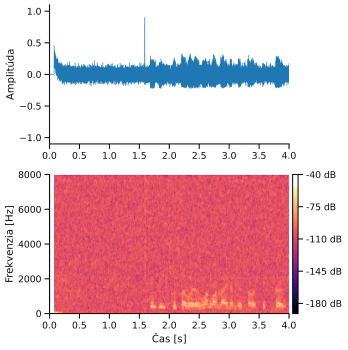
Riešenie

V rámci práce bol implementovaný iteratívny spôsob trénovania modelu, ktorého cieľom je získanie väčšieho množstva čistých dát pre vytvorenie trénovacej dátovej sady v prípade, že čisté nahrávky nie sú k dispozícii. Po skončení jednej iterácie zloženej z niekoľkých epochov je natrénovaný model využitý na odšumenie časti leteckých dát. Odšumené nahrávky sú následne pridané do trénovacej dátovej sady ako čisté nahrávky a tréning pokračuje ďalšou iteráciou. Spolu s nahrávkami boli dodané aj ich prepisy, čo umožnilo z týchto nahrávok vystrihnúť úseky bez reči a tie následne využiť pre zašumenie čistých nahrávok. Využitá bola neurónová sieť s architektúrou GAN, skladajúcou sa z modelov diskriminátora a generátora. Tieto dva modely sú trénované súbežne, pričom cieľom diskriminátora je naučiť sa odhadnúť skóre PESQ vstupných nahrávok a cieľom generátora je odšumiť vstupné nahrávky tak, aby optimalizoval hodnotenie diskriminátora.

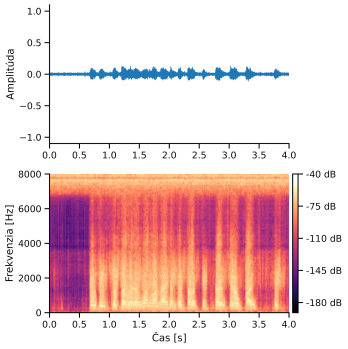
Výsledky

Vykonaných bolo niekoľko experimentov zameraných prevažne na tréning modelov pre odstránenie šumu, keďže ten bol hlavným rušivým faktorom pri počúvaní daných nahrávok. Toto sa podarilo s rôznou úspešnosťou. V rámci experimentov využívajúcich statickú dátovú sadu sa podarilo dosiahnuť 1.30 násobného zlepšenia hodnotenia MOS testovacích nahrávok. Modelu, trénovaným iteratívnym spôsobom, sa podarilo dosiahnuť 1.31 násobného zlepšenia v hodnotení MOS.

Pôvodná nahrávka



Odšumená nahrávka



Simon Košina | Vedúci práce: Ing. Igor Szöke, Ph.D.