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

SEPARACE MLUVČÍCH V ČASOVÉ DOMÉNĚ

TIME DOMAIN AUDIO SEPARATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JIŘÍ PEŠKA**
AUTHOR

VEDOUCÍ PRÁCE **ing. KATEŘINA ŽMOLÍKOVÁ,**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Peška Jiří**
Program: Informační technologie
Název: **Separace mluvčích v časové doméně pomocí neuronové sítě**
Time-Domain Neural Network Based Speaker Separation
Kategorie: Zpracování řeči a přirozeného jazyka

Zadání:

1. Seznamte se s problémem separace mluvčích pomocí neuronových sítí.
2. Seznamte se s metodou TasNet pro jednokanálovou separaci signálu v časové doméně.
3. Implementujte danou metodu s využitím vhodného toolkitu (např. PyTorch, Keras).
4. Otestujte systém na vhodném datasetu. Zaměřte se na vyhodnocení vlivu velikosti sítě na přesnost.
5. Navrhněte a diskutujte možné zlepšení použité metody.

Literatura:

- Luo, Yi, and Nima Mesgarani. "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.8 (2019): 1256-1266.
- dle doporučení vedoucího

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

Vedoucí práce: **Žmolíková Kateřina, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 14. května 2020
Datum schválení: 5. listopadu 2019

Abstrakt

Práce se zabývá využitím konvolučních neuronových sítí pro automatickou separaci mluvčích v akustickém prostředí. Cílem je implementovat neuronovou síť podle architektury TasNet za použití frameworku pytorch, natrénovat síť s různými hodnotami hyperparametrů a porovnat kvalitu separací vzhledem k velikosti sítě.

Architektura oproti dosavadním metodám, které převáděly vstupní směs do časově–frekvenční reprezentace, používá konvoluční autoenkodér, který vstupní směs převádí do nezáporné reprezentace, která je optimalizovaná pro extrakci jednotlivých mluvčích. Samotné separace je docíleno aplikací masek, které jsou odhadnuty v separačním modulu. Modul tvoří opakující se posloupnost konvolučních bloků se zvyšující se dilatací, která napomáhá k modelování časových závislostí ve zpracovávané směsi.

K vyhodnocení přesnosti bylo použita metrika SDR (Sound to Distortion Ratio), která určuje poměr zastoupení šumu a zvuku v nahrávce. Natrénováním několika modelů s různými hodnotami hyperparametrů bylo možno zpozorovat závislost mezi velikostí sítě a hodnotou SDR. Zatímco menší síť dosahovala, po X epochách trénování, přesnosti XY, větší síť dosahovala až XX.

[[Jeste neco? 4. cast?]]

[[Doplnit SDR presnost do odstavce vyse.]]

Abstract

[[Prelozit do anglictiny CZ abstrakt]]

Klíčová slova

neuronové sítě, zpracování řeči, konvoluční neuronová síť, autoenkodér, separace mluvčích, strojové učení, tasnet, feed forward, hluboké učení

Keywords

artificial neural networks, speech processing, convolutional neural networks, autoencoder, speech separation, machine learning, tasnet, feed forward, deep learning

Citace

PEŠKA, Jiří. *Separace mluvčích v časové doméně*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce ing. Kateřina Žmolíková,

Separace mluvčích v časové doméně

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením ing. Kateřiny Žmolíkové. Další informace mi poskytli... Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jiří Peška
9. dubna 2020

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant apod.).

Obsah

Kapitola 1

Úvod

Zpracování řeči hraje v dnešní době důležitou roli v mnoha rozličných oborech. Mezi jeden z hlavních úkolů bezesporu patří separace zdrojů v nějakém zaznamenaném signálu, který může být složen ze signálů N mluvčích, ale i nechtěného hluku okolí. Vyřešení problému je předpoklad k dalším úkonům jako identifikace konkrétního mluvčího nebo třeba přepis nějaké konverzace na text. Se stále se zrychlujícím vývojem počítačů a s jejich zvyšujícím se výkonem se do popředí dostávají metody zpracování řeči založené na neuronových sítích, které v mnoha ohledech předčily ostatní algoritmy strojového učení.

Separace mluvčích v časové doméně dosahuje mimořádných výsledků v porovnání s dosavadními metodami založenými na převodu signálu z časové domény do frekvenční domény pomocí algoritmu STFT (Short–Time Fourier Transformation). Taková reprezentace signálu není optimální pro udržení časových závislostí, které jsou při zpracování řeči podstatné. V architektuře, která je navržena v referenční studii, je vstupní signál převeden do nezáporné reprezentace, která je optimální pro extrakci jednotlivých mluvčích. Silnou stránkou systému je hluboká architektura sítě, která lépe modeluje dlouhodobé závislosti v signálu. Zároveň se ale musí vypořádat s problémy, které hluboké neuronové sítě mohou přinášet.

Téma v oblasti strojového učení a neuronových sítí mě zaujalo, protože tento obor zažívá obrovský rozmach a pomalu se stává součástí téměř všech odvětví, a je tudíž velmi perspektivní pro další výzkum. Právě bakalářskou práci jsem vyhodnotil jako dobrou příležitost k tomu, se seznámit se s neuronovými sítěmi a možnost vyzkoušet si, jak se s nimi pracuje, jak se implementují modely za pomoci frameworku a jak náročná je aplikace na nějaký reálný problém.

[[spravit referenci na studii]] Mým úkolem v rámci práce je nastudovat si problematiku neuronových sítí a jejich základní principy, seznámit se problémem separace mluvčích pomocí neuronových sítí a následně implementovat síť podle architektury TasNet pro separaci mluvčích v časové doméně, která byla navržena a popsána ve studii **??**. Potom tuto neuronovou síť natrénovat s různými kombinacemi hodnot hyperparametrů, které ovlivňují velikost sítě její vlastnosti, a nakonec porovnat přesnost a kvalitu separace mezi jednotlivými, různě velkými sitěmi a mezi výsledky studie. Přesnost separace je vypočítána pomocí míry si–snr, udávající poměr mezi chtěným signálem a hlukem na pozadí. Sítě budou testovány a vyhodnocovány na testovací množině jednokanálových směsí dvou mluvčích.

V první části práce, kterou pokrývá kapitola 2, jsou popsány základní prvky neuronových sítí, struktura umělého neuronu, jeho vstupy a výstupy, váhy a role aktivační funkce. V návaznosti na to jsou popsány neuronové sítě feedforward, které se skládají z vrstev neuronů. Dále je vysvětlen proces učení, který se skládá z několika kroků – výpočet výstupu sítě,

výpočet chyby a úprava vah, kde hraje podstatnou roli metoda backpropagation. Nakonec je zmíněn typ sítí feedforward – konvoluční neuronové sítě, které fungují na základě konvoluční operace. Konvoluční sítě se používají nejčastěji pro zpracování obrazu kvůli vlastnostem, které umožňují extrahovat příznaky s různou úrovní složitosti od základních útvarů jako úsečka, barva a podobně až po komplexnější příznaky - třeba celý obličej. Tohoto lze využít i při zpracování zvuku, kde jsou tyto extrahované příznaky jednorozměrné.

Druhá část, kapitola 3, je věnována architektuře TasNet. Prvně je zmíněn konvoluční auto–enkodér a jeho princip. Dál je popsána podoba separačního modulu, jeho princip a konvoluční bloky, ze kterých je poskládán. Tento blok obsahuje konvoluční vrstvy, normalizace a aktivační funkce. Bloky jsou skládány za sebe se zvyšující se časovou dilatací, čemuž se říká TCN (Temporal Convolutional Network).

Kapitola 4 se zabývá implementací neuronové sítě a jejím trénováním. Je popsána a zdůvodněna volba frameworku, implementace sítě a struktura zdrojového kódu. Pro usnadnění často se upakujících úkonů jsem vytvořil pár pomocných scriptů, které jsou zde také popsány. Model prošel během implementace několika úpravami. Pro účely trénování a validace byly vstupní nahrávky rozdělovány na čtyřsekundové segmenty. Pro testování byly používány nahrávky celé. V této kapitole je popsán průběh trénování sítí, výsledky a použité nástroje.

V poslední části, která je pokryta v kapitole 5 jsou shrnuty experimenty s modelem a vyhodnocení výsledků, v rámci něhož je zkoumán vliv hyper-parametrů na učení sítě, na výsledky a přesnost separace v závislosti na zvolených parametrech nebo počtu konvolučních bloků. Výstup sítě v podobě separovaných mluvčích je porovnán také s výsledky referenční studie. Kvalita separace, neboli přesnost, je vypočítána pomocí si–snr metriky která udává poměr zastoupení chtěného signálu a hluku na pozadí. Je zde vyhodnoceno, jaký vliv má velikost sítě na přesnost separace a jsou zde shrnuty jednotlivé výsledky.

Kapitola 2

Neuronové sítě

V dnešní době zažívají neuronové sítě díky výkonosti počítačů velký rozmach. Jejich využití prostupuje skrze mnohé vědní obory a dokáží řešit celou řadu problémů, ve kterých dosahují výborných výsledků, které zdaleka předčily dosavadní postupy.

Neuronové sítě (*artificial neural networks*) jsou výpočetní model, který je inspirovaný strukturou lidského mozku, ve kterém je obrovské množství propojených a komunikujících neuronů. Ty se skládají ze vstupních dendridů, výstupních axonů a samotného těla neuronu. Na základě vnitřního potenciálu a vstupních hodnot je po přesažení prahové hodnoty neuron vybuzen a je vyslán signál na výstupní axon. Signál je nakonec předán dalším neuronům skrze jejich vstupní dendridy[3, p. 65–66].

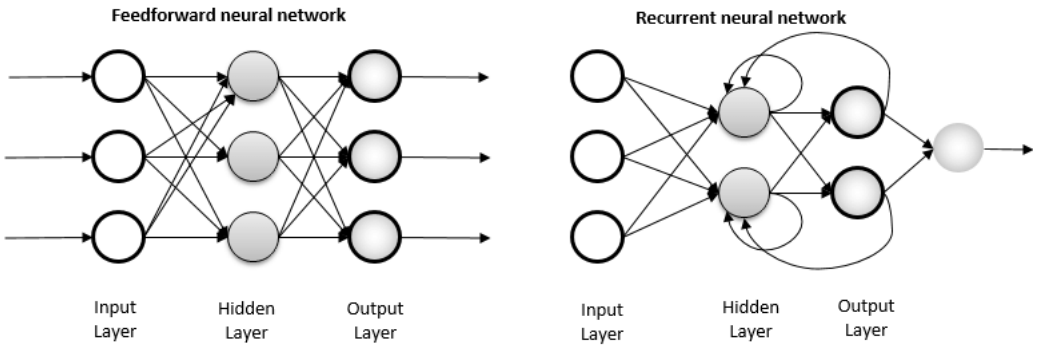
Účelem neuronové sítě je naučit se plnit zadanou úlohu. Rozdíl oproti běžným algoritmům je ale ten, že způsob, jakým síť má problém řešit, není explicitně naprogramován, ale je postupně naučen. Základní způsoby učení jsou s učitelem (*supervised*) a bez učitele (*unsupervised*).

Učení s učitelem spočívá v [[bla bla (dve tri vety)]]. **Učení bez učitele** spočívá v [[bla bla (dve tri vety)]].

Mezi problémy, které se dají řešit neuronovými sítěmi patří klasifikační a regresní problémy. Konkrétní příklad z oblasti klasifikace může být rozpoznávání objektů na obraze, psaného písma nebo detekce obličejů na videu, ale i mnohé aplikace ve zpracování řeči.

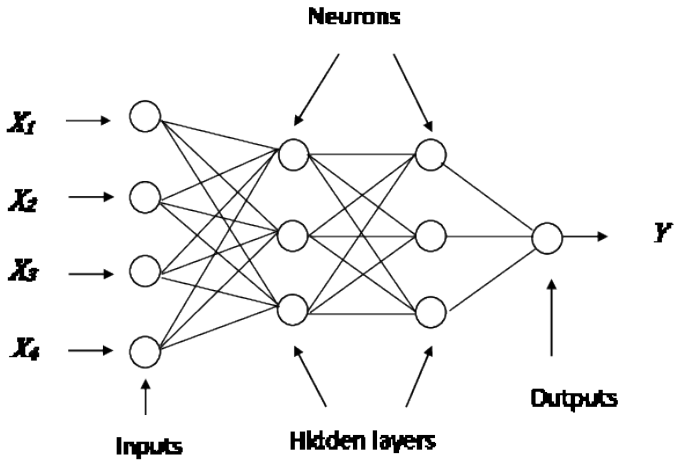
2.1 Organizace feedforward sítí

Feed forward neuronové sítě jsou typ umělých neuronových sítí, kde se nevyskytují cykly ve výpočetním grafu, tedy zpětná propojení vrstev, takže informace se pohybuje pouze jedním směrem, od vstupní vrstvy přes skryté vrstvy až do vrstvu výstupní. Sítě, které obsahují cykly, se nazývají rekurentní. Rozdíl znázorňuje obrázek 2.1.



Obrázek 2.1: Příklad grafu feed forward sítě a rekurentní neuronové sítě. Lze si všimnout orientace šipek u feed forward sítě, které směřují pouze jedním směrem, zatímco u rekurentní sítě šipky směřují i k předešlým uzlům grafu.

Struktura neuronové sítě je organizována do vrstev, které se skládají z neuronů. Feedforward síť je tvořena třemi typy vrstev (viz obrázek 2.2). Každá vrstva může obsahovat až $n, n \in \mathbb{N} \setminus \{0\}$ neuronů. Vstupní vrstva slouží k předání hodnot do sítě, ale nijak tyto hodnoty nemodifikuje. Nezměněné jsou zkopírovány do první skryté vrstvy. Následují skryté vrstvy, z nichž poslední je napojena na výstupní vrstvu. Ta má obvykle méně neuronů než předešlé vrstvy a hodnoty na výstupu mohou představovat třídy, do kterých má být klasifikován vstup. S počtem jednotlivých vrstev souvisí pojem hloubka sítě, která je rovna počtu všech vrstev neuronové sítě od vstupní až po výstupní vrstvu. Pojmem hluboká neuronová síť se označuje taková síť, která má dvě nebo více skrytých vrstev. Vrstvy se ještě dále rozdělují na plně propojené, pooling, s přeskočením či na konvoluční vrstvy.



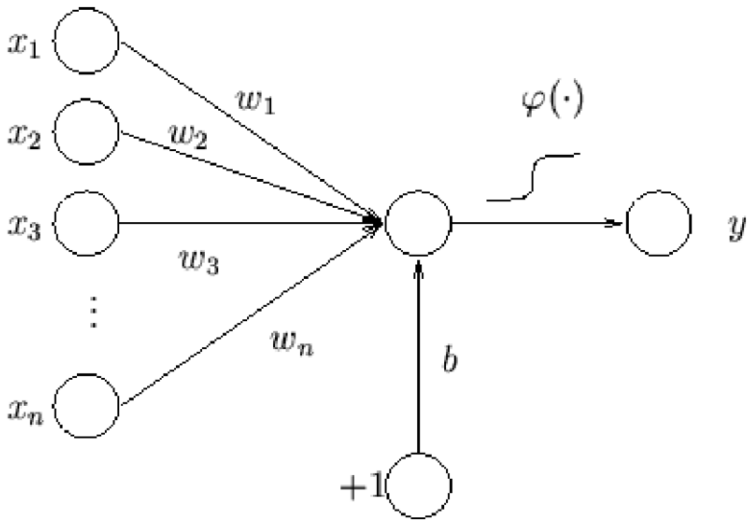
Obrázek 2.2: Schéma neuronové sítě, která má 2 skryté vrstvy

2.2 Umělý neuron

Základní stavební jednotka neuronových sítí je umělý neuron (*artificial neuron*) (viz obrázek 2.3). Tento model je založen na principu reálných neuronů, které se nacházejí v organizmech. Umělý neuron obsahuje libovolně mnoho vstupních propojení, přes které se mu předávají

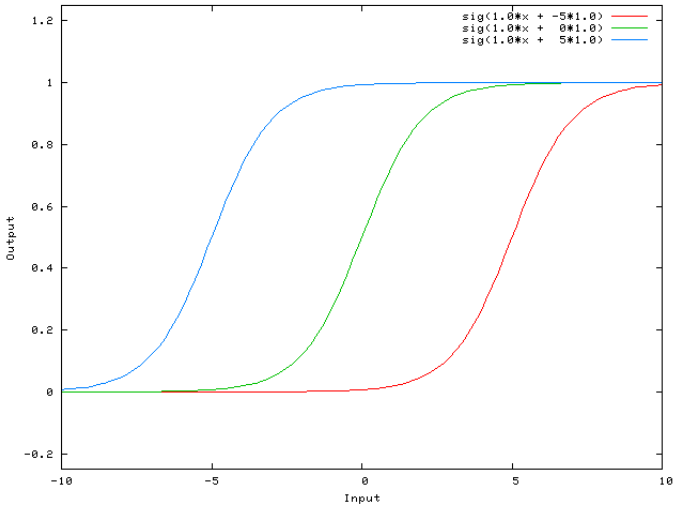
data v podobě vstupního vektoru $\overrightarrow{x} = [x_1, x_2, \ldots, x_n], x_n \in \mathbb{R}$. Sám neuron obsahuje hodnotu bias $b \in \mathbb{R}$ a vektor vah $\overrightarrow{w} = [w_1, w_2, \ldots, w_n], w_n \in \mathbb{R}$, jenž je upravován během trénování neuronu.

Výstupní hodnota závisí na vstupních datech, aktuálním vnitřním stavu (hodnoty vah a biase) a na zvolené aktivační funkci. Vstupní hodnoty jsou váhovány, což znamená, že každá vstupní hodnota je vynásobena s váhou na daném vstupním spojení. S použitím definovaných vektorů lze napsat, že vstupní vektor je vynásoben s vektorem vah.



Obrázek 2.3: Schéma umělého neuronu

Hodnota bias b, která je přičtena k sumě násobků vah a vstupních hodnot, je prahová hodnota modifikující dobu, kdy se aktivuje neuron a změní svůj výstup. Matematicky to znamená, že s grafem aktivační funkce horizontálně pohybuje doleva nebo doprava v závislosti na tom, je-li hodnota biasu pozitivní nebo negativní. Toto posunutí je znázorněno na obrázku 2.4. V závislosti na řešeném problému může být žádoucí, aby i hodnota bias byla modifikována během učení společně s ostatními váhami. V opačném případě je hodnota nastavena pevně na nějakou konstantní hodnotu, obvykle na hodnotu jedna.



Obrázek 2.4: Vliv hodnoty bias na aktivační funkci

Výstup neuronu se vypočítá jako

$$y = f\big((\sum_{k=1}^{n} w_k x_k) + b\big) \tag{2.1}$$

kde f je nějaká aktivační funkce, $x_k \in \mathbb{R}$ je vstupní hodnota, $w_k \in \mathbb{R}$ je váha, kterou se vstupní hodnota vynásobí a $b \in \mathbb{R}$ je hodnota bias, která je přičtena k celkové sumě předtím, než je výsledek předán aktivační funkci.

2.3 Aktivační funkce

Aktivační, neboli prahová funkce určuje výstupní hodnotu neuronu. Funkce se vybírá na základě problému, který se má neuronová síť naučit řešit. Správná volba prahové funkce vede k lepší konvergenci učení sítě. Naopak špatná volba může vést ke stále větší odchylce od správného řešení – může divergovat. Povaha problému může vyžadovat specifické vlastnosti aktivační funkce - lineární nebo nelineární. Pro nestandardní problémy je obvykle potřeba experimentálně zjistit, která funkce bude nejlépe vyhovovat danému problému.

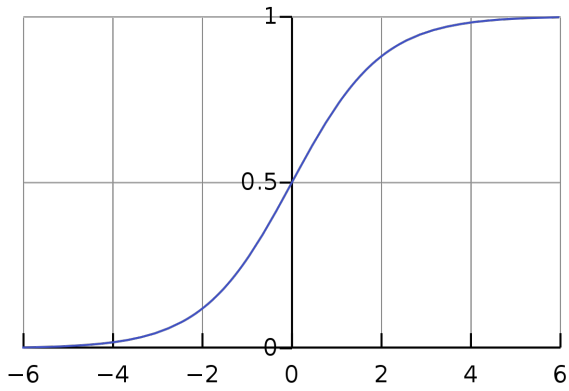
Pokud by veškeré aktivační funkce v modelu byly lineární, tak celkové mapování sítě by bylo omezeno pouze na lineární mapování vstupu na výstup. Reálné problémy ale lineární obvykle nejsou a v případě pokusu modelovat takovým modelem nelineární vztahy by vedlo k velice nepřesným výsledkům, který by byl zapříčiněn podučením (*underfitting*), což znamená, že model, který se učí zakódovat nějaký vzor v datasetu, je příliš jednoduchý. Proto je potřeba zavést do modelu i nelineární aktivační funkce, které tento problém řeší[3, p. 77–78].

Z pohledu učení je také důležité, aby aktivační funkce byla diferencovatelná. To umožňuje použití učících metod založených na výpočtu gradientu.

Sigmoid

$$f(x) = \frac{1}{1 + \exp(-z)} \tag{2.2}$$

[[popsat]]

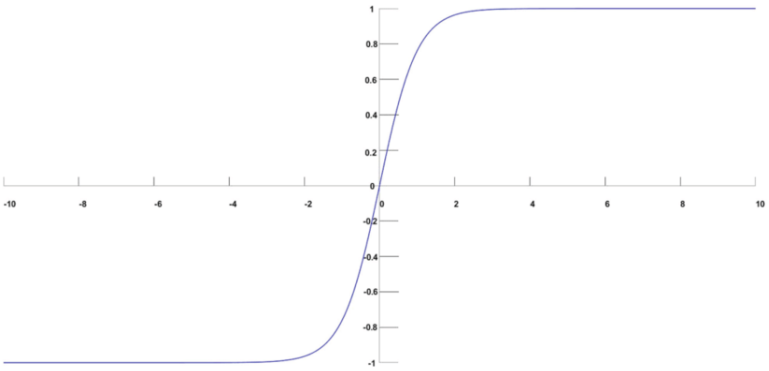


Obrázek 2.5: Graf aktivační funkce sigmoid

Softmax

$$f(x) = \frac{1}{1 + \exp(-z)} \tag{2.3}$$

[[popsat]]

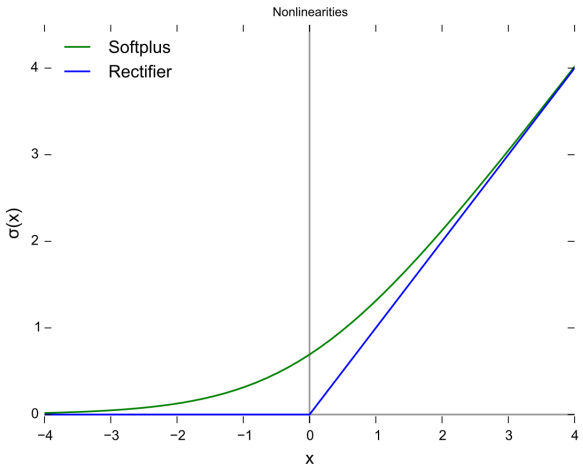


Obrázek 2.6: Graf aktivační funkce softmax

ReLU

Rectified Linear Unit je nejčastěji používaná aktivační funkce. Vyžaduje-li neuronová síť nějakou nelinearitu, je ReLU pro většinu případů ideální. Pro každou zápornou hodnotu x vrací 0 a pro kladnou hodnotu x vrací tutéž hodnotu x, jak udává rovnice

$$f(x) = max(0, x) \tag{2.4}$$

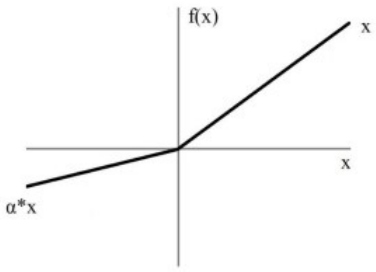


Obrázek 2.7: Graf aktivační funkce ReLU

PReLU

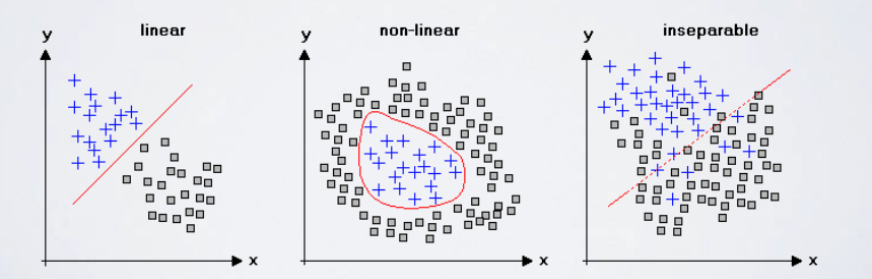
Parametrizovaná ReLU je nelineární aktivační funkce, která se používá v případě, že chceme produkovat na výstup malý nenulový gradient i v případě záporné vstupní hodnoty x. V tom případě je vstupní hodnota vynásobena parametrem α a to představuje výsledek. Parametr α se společně s ostatními váhami učí během učícího procesu.

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \tag{2.5}$$



Obrázek 2.8: Graf aktivační funkce PReLU

Síť složená z jediného neuronu by byla omezena pouze na klasifikaci lineárně separovatelných dat. To jsou taková data, která lze rozdělit, existuje-li alespoň jedna přímka v rovině taková, že na jedné straně přímky jsou všechna data klasifikována do první třídy, a na druhé straně data klasifikována do třídy druhé, jak lze vidět na obrázku 2.9.



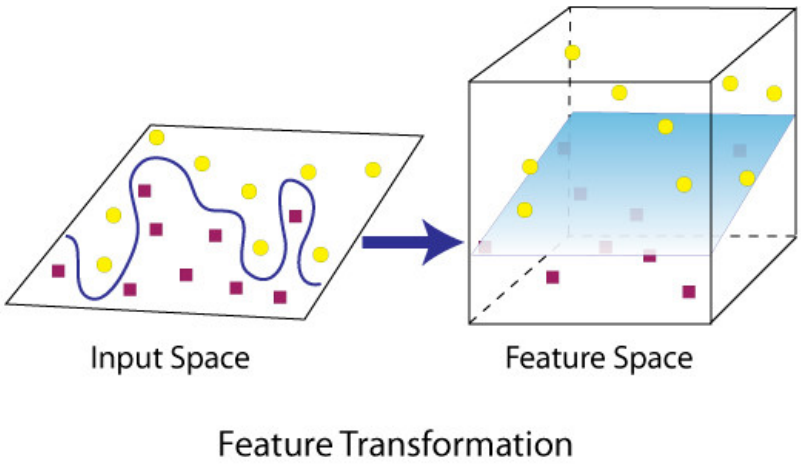
Obrázek 2.9: Lineárně separovatelná (vlevo) a neseparovatelná (vpravo) data

Mějme množinu dat, kterou tvoří dvě rozdílné třídy objektů $\Omega_1, \Omega_2 \subset \mathbb{R}^m$. Třídy jsou lineárně separovatelné, jestliže existuje $w \in \mathbb{R}^m$ a $\theta \in \mathbb{R}$ takové, že

$$\forall x \in \Omega_1 : x^T w > \theta \tag{2.6}$$

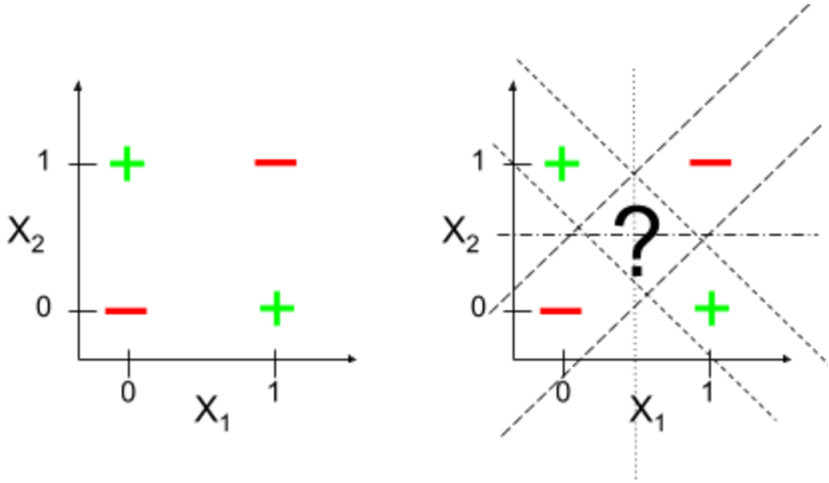
$$\forall x \in \Omega_2 : x^T w < \theta \tag{2.7}$$

Problém lze zobecnit tak, že řešením bude separace tříd pomocí hyper roviny v prostoru $\mathbb{R}^m$, jak zachycuje obrázek 2.10 [1].



Obrázek 2.10: Separace pomocí hyper roviny

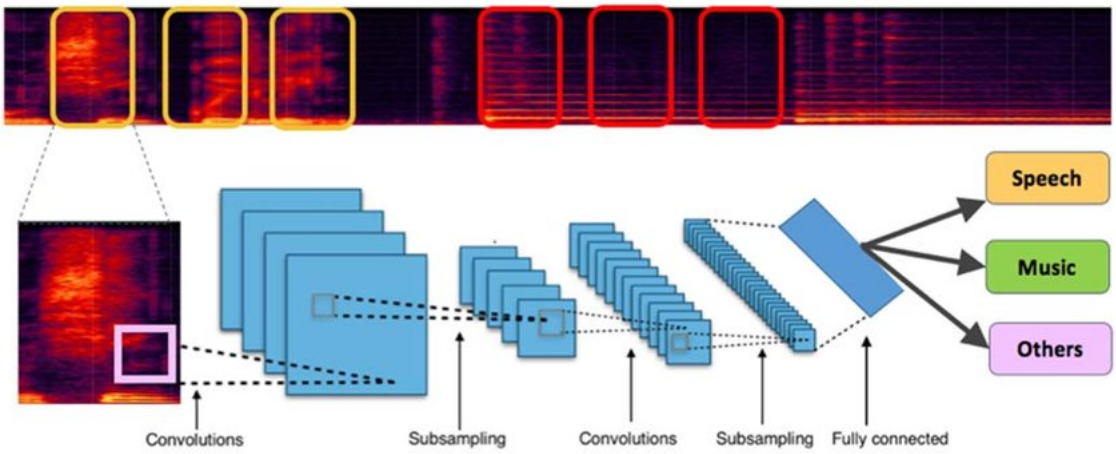
Neuron se nedokáže ale naučit klasifikovat třeba funkci XOR, protože ta není lineárně separovatelná. Byly by zapotřebí minimálně dvě přímky pro rozdělení dat, jak zobrazuje obrázek 2.11[**[citovat xor]**]. K překonání tohoto problému se neurony spojují do podoby neuronové sítě, která dokáže řešit libovolně komplexní problémy.



Obrázek 2.11: Graf funkce XOR, jejíž výstup není lineárně separovatelný

2.4 Konvoluce a konvoluční neuronové sítě

Konvoluční neuronové sítě jsou typ feedforward sítí.



Obrázek 2.12: Konvoluční neuronové sítě

2.4.1 Konvoluce

Diskrétní konvoluce je definována jako

kde $\star$ je konvoluční operátor, f je funkce signálu, funkce g je konvoluční jádro a f_i a g_i jsou hodnoty funkce na indexu i.

2.5 Hyperparametry

[[co to je, ze se to nastavuje pred trenovanim site, neni to meneno behem trenovani... learning rate, X, R, stride, padding, receptive field, ...]]

2.6 Učení neuronových sítí

[[proces uceni, cil uceni, objektivni funkce a loss, backpropagation a gradient descent, pripadne problemy trenovani hlubokych neuronovych siti a generalizace a underfitting]]

Cílem trénování sítě je aproximovat nějakou funkci f^*. Síti je předána vstupní hodnota x, pro kterou síť definuje mapování na výstupní hodnotu jako $y = f(x; \theta)$, kde θ je parametr, který se síť učí tak, aby dosáhla nejlepší aproximace funkce. [2, p. 163].

2.6.1 Objektivní funkce

[[cost funkce a popis, co to je, k cemu to je, proc to je]]

MSELoss

- vzorecek

Cross Entrophy

- vzorecek

2.6.2 Optimalizační algoritmy

[1 deep learning str 301] Mozna kaslat...

Adam

Adam je jeden z algoritmů s adaptivním učením. Jeho název byl odvozen z fráze "adaptive moments". [1 deep learning str 301] Mozna kaslat.

2.6.3 Backpropagation

- zpetne sireni chyby - adaptacni algoritmus, podil neuronu na chybe, - 3 opakujici se faze uceni:

1) feedforward - dopredu 2) zpetne sireni chyby - Backpropagation 3) uprava vah a biasu na zaklade chyby pomoci gradient descent - chain rule

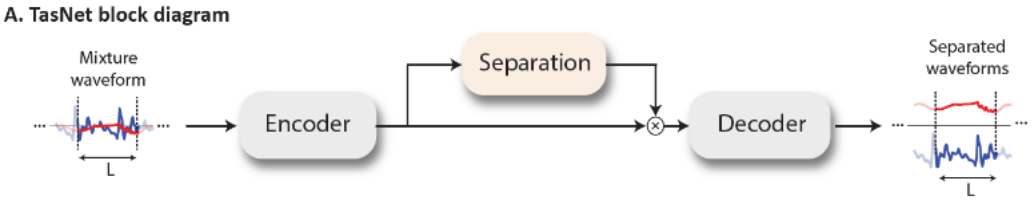
Gradient descent

2.6.4 Overfitting a generalizace

Kapitola 3

TasNet - Time–Domain Audio Separation Network

[[Architektura full – obrázek, bloky...]]



Obrázek 3.1: Zjednodušený model architektury

3.1 Konvoluční auto–enkodér

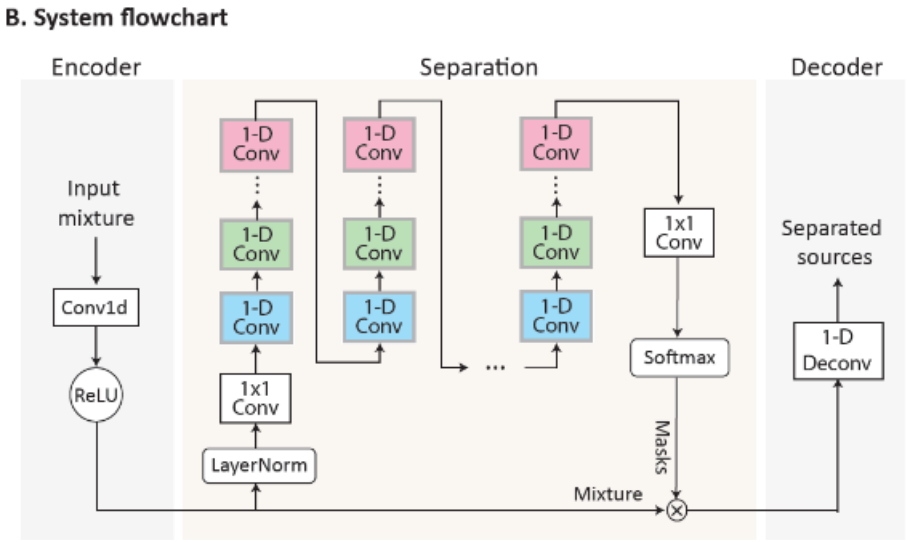
[[Konvoluční autoenkodér, vstup, výstup...]]

- schema bez separacniho modulu - non negative representation of audio

Obrázek 3.2: Schéma konvolučního autoenkodéru

3.2 Separační modul

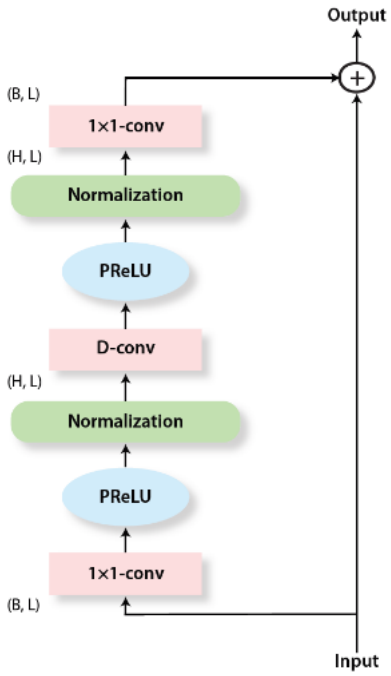
- odhad masek pro jednotlive mluvci - schema se separacnim modulem



Obrázek 3.3: Schéma architektury TasNet

3.2.1 Konvoluční bloky

- Z čeho se skládá – konvoluční vrstvy, normalizace - diagram konv bloku. - Mozna: Dilatace a time perception



Obrázek 3.4: Jeden konvoluční blok

Kapitola 4

Implementace a trénování sítě

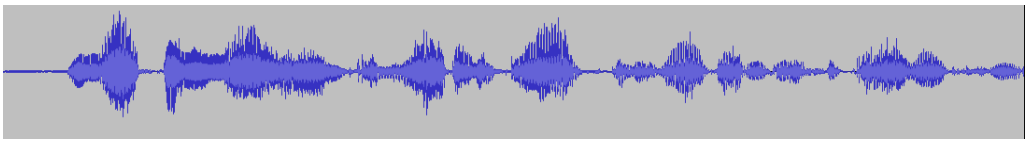
Pozn: colab, pytorch, stroj, bash, hyperparams, vykon a cas trenovani, seg–len, popis trid.

4.1 Implementace modelu

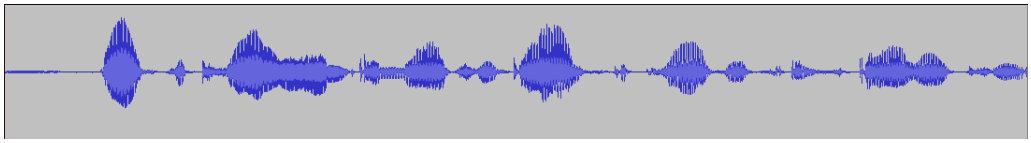
- pytorch, scripty, python3, bash, tridy, moduly, parametry a volby spusteni.

4.2 Dataset

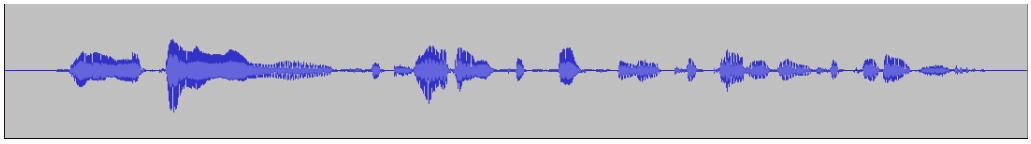
[[Ukazat zde vykreslenou vlnu nahravek mix, s1, s2]] [[popsat co je dataset a k cemu to slouzi]] Trénování a vyhodnocení modelu proběhlo na množině jednokanálových nahrávek směsí dvou mluvčích. Množina byla vygenerována náhodným výběrem různých mluvčích z Wall Street Journal (WSJ0) a vytvořením směsi. Celková délka trénovacích dat je přes 10 hodin a přes 6 hodin validačních dat. Nahrávky jsou převzorkovány na 8kHz a během trénování zarovnány na zero means a jednotkovou varianci[studie str 5 Dataset][49 - ze studie odkaz na script na generovani a popis na netu].



Obrázek 4.1: Ukázka nahrávky směsi dvou mluvčích



Obrázek 4.2: První mluvčí ze směsi



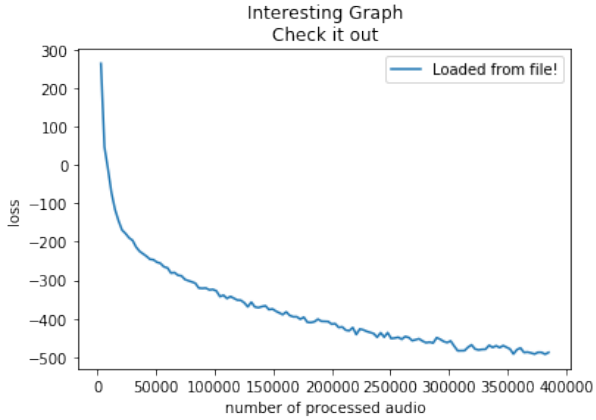
Obrázek 4.3: Druhý mluvčí ze směsi

Lze si všimnout, že sečtením signálů separovaných mluvčích na obrázku 4.2 a 4.3 dostaneme přesně signál směsi, což lze vyjádřit vztahem

$$x[t] = \sum_{i=1}^{C} s_i[t] \tag{4.1}$$

, kde $x[t] \in \mathbb{R}^{1\times T}$ je diskrétní signál směsi a $s_i[t] \in \mathbb{R}^{1\times T}$, kde $i = 1, \ldots, C$, je jeden z C zdrojů[cite studie str3 vlevo]. **[[doplnit info o zero means a jendotkove varianci]]**

4.3 Trénování



Obrázek 4.4: Příklad grafu loss hodnoty během učení

Obrázek 4.5: Hodnota loss při trénování modelů s různou velikostí hyperparametrůx

4.3.1 Význam validační množiny v trénování

Většina algoritmů strojového učení má nějakou sadu hyperparametrů, kterou je upravováno chování algoritmu. Hodnoty hyperparametrů obvykle bývají nastavovány ručně ještě před spuštěním procesu učení a hodnota se v průběhu nemění, protože hodnoty by bylo obtížné optimalizovat. Některá nastavení se nicméně mohou stát hyperparametrem a být upravována během trénování, ale není vhodné je měnit na základě výsledku učení na trénovací sadě, protože by mohlo dojít k přetrénováníoverfitting) v důsledku **[[CEHO??]]**. Pro tento případ potřebujeme validační sadu, která je odlišná od trénovací sady. Po každém zpracování trénovací sady následuje validační sada, po jejímž skončení jsou optimalizovány hyperparametry[][].

[[[kniha 117-118] kap 5.3 = Hyperparameters and validation set]] [[najit jeste nejakej zdroj s popisem a pripadne nejaky zajimavejsi info.]]

4.4 Vyhodnocovací metriky

- minimalizovat objektivni-hodnotici funkci sisnr.

4.4.1 Signal to noise ration

Source Distortion Ratio – SDR

Artifacts Ratio – SAR

Inference Ratio – SIR

Kapitola 5

Experimenty a vyhodnocení

- trenovani s ruznymi hyperparametry, uspesnost a tabulky s hyper parametry a dosazenymi vysledky a hodnotami sisnr, sdr atd. - model size comparison. - porovnani s vysledky ze studie - obrazky separovanych mluvcich - signalu. - spektra - grafy trenovani loss a vysledkuu. - pametova narocnost modelu

5.1 Možná rozšíření a navrhnutá vylepšení

- variabilnější dataset, mikrofony, šum a bordel prostředí - separace více mluvčích - hlučné prostředí - identifikace konkrétního řečníka - realtime separace

Kapitola 6

Závěr

- co jak dopadlo, vysledky a vyhodnoceni velikosti modelu a jaky byl nejlepsi,...

Cílem práce bylo implementovat síť podle architektury TasNet pro separaci mluvčích v časové doméně a porovnat vliv velikosti sítě na kvalitu separace. Síť byla implementována za pomoci frameworku pytorch a jazyku python a natrénována na datasetu obsahujícím jednokanálové směsi dvou mluvčích. Trénování proběhlo na [[**X**]] modelech, které se od sebe lišily počtem opakujících se konvolučních bloků, velikostí časové dilatace a délkou vstupních segmentů směsí. Pro účel vyhodnocení modelů byla použita metrika si-snr, která udává poměr chtěného signálu ku šumu na pozadí, tedy obecně kvalitu separace.

Experimenty ukázaly, že během testování nejlépe dopadla síť, která měla 8 konvolučních bloků po 4 opakováních, s délkou vstupního segmentu $L = 2$ sekundy. Tento model dosáhl po 100 epochách trénování hodnoty až [[**13,4**]] a tím se stal nejúspěšnějším modelem. Při fyzickém poslechu separovaných nahrávek bychom neslyšeli téměř žádný náznak druhého mluvčího. Oproti tomu, nejméně přesný model měl pouze 4 konvoluční bloky, 2 opakování a při délce segmentů $L = 4$ sekundy dosahoval hodnoty SDR pouze [[**9.2**]].

Zkoušel jsem separovat také nahrávky, které byly úplně mimo dataset, ale výsledek se nedá hodnotit jako úspěšný, jelikož hraje velkou roli prostředí, mikrofon, šum v pozadí a další vlivy, na které byla neuronová síť naučena. Tento problém by se dal překonat rozšířením trénovacího datasetu o větší škálu nahrávek mluvčích, které by byly pořízeny z různých zařízení v různě rušném prostředí.

[[**Doplnit ještě neco eh**]]

Mozna.

Literatura

[1] Bauckhage, C. a Cremers, O. Lecture Notes on Machine Learning: Linear Separability. Únor 2019.

[2] Ian Goodfellow, A. C. *DEEP LEARNING*. MIT Press, 2017. ISBN 9780262035613.

[3] Kelleher, J. D. *DEEP LEARNING / John D. Kelleher*. MIT Press, 2019. ISBN 9780262537551.