



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

AUTHOR

JIŘÍ PEŠKA

VEDOUcí PRÁCE

SUPERVISOR

ing. KATEŘINA ŽMOLÍKOVÁ,

BRNO 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

Keywords

neural networks, speech processing, convolutional neural networks, autoencoder, speech separation, machine learning, tasnet

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
5. dubna 2020

Poděkování

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

Obsah

1	Úvod	2
2	Neuronové sítě	4
2.1	Umělý neuron	4
2.1.1	Aktivační funkce	6
2.2	Feed forward networks	9
2.2.1	Objektivní funkce	9
2.2.2	Optimalizační algoritmy	9
2.2.3	Backpropagation	10
2.2.4	Overfitting a generalizace	10
2.3	Konvoluční neuronové sítě	11
2.3.1	Konvoluce	11
3	TasNet - Time-Domain Audio Separation Network	12
3.1	Konvoluční auto-ekodér	12
3.2	Separační modul	13
3.2.1	Konvoluční bloky	14
4	Implementace a trénování sítě	15
4.1	Implementace modelu	15
4.2	Dataset	15
4.3	Trénování	16
4.3.1	Význam validační množiny v trénování	16
4.4	Vyhodnocovací metriky	17
4.4.1	Signal to noise ration	17
5	Experimenty a vyhodnocení	18
5.1	Možná rozšíření a navrhnutá vylepšení	18
6	Závěr	19
	Literatura	20

Kapitola 1

Úvod

Zpracování řeči hraje v dnešní době důležitou roli v mnoha rozličných oborech. Mezi jedny 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 doposud používané algoritmy.

Separace mluvčích v časové doméně dosahuje mimořádných výsledků v porovnání s dosavadními metodami LSTM založenými na převodu signálu z časové domény do frekvenční domény. Taková reprezentace signálu není optimální pro udržení časových závislostí, které jsou při zpracování řeči podstatné. 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.

Téma v oblasti neuronových sítí jsem si vybral, jelikož ty zažívají obrovský rozmach a pomalu se stávají součástí téměř všech odvětví a tudíž je velmi perspektivní pro další výzkum. Právě bakalářskou práci jsem vyhodnotil jako dobrou příležitost k seznámení se s neuronovými sítěmi a vyzkoušení 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 a její vlastnosti, a nakonec porovnat přesnost a kvalitu separace mezi jednotlivými, různě velkými sítěmi 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 jednobáňových 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 je popsán proces učení neuronových sítí. Proces učení se skládá z několika souvisejících částí, které zahrnují výpočet výstupu neuronové sítě metodou feed forward, který transformuje vstupní vektor dat a počítá na základě něj výstupní vektor, který je předán zase další vrstvě a takto analogicky až do výstupní vrstvy. Dále je rozebrán výpočet chyby, která vzniká během procesu učení, metodou gradient descent a nakonec úprava vah neuronů metodou backpropagation (zpětná propagace chyby), která se počítá na základě

rozdílu mezi vstupními hodnotami a očekávanými výstupními hodnotami. **[[Nejsem si jistej, jestli tohle je dobře s tím backprop atd, mam dojem ze backprop pocita gradienty, a ze obj funkce pocita chybu.]]**

Po vysvětlení základních principů je navázáno konvolučními neuronovými sítěmi, které jsou založeny na konvoluční operaci. 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 jako část obličeje – ucho, nos, či úplně celý obličej. Tohoto lze využít i při zpracování zvuku, kde jsou tyto extrahované příznaky jednorozměrné.

Druhá část je věnována architektuře TasNet. V kapitole 3 je popsána podoba separačního modulu, jeho stavební bloky a princip. Postupně je zmíněn konvoluční auto-inkodér, u něž je vysvětlen jeho úkol v separačním modulu a následně konvoluční blok, který se sám sestává z konvolučních vrstev, normalizací a aktivčních funkcí. Tyto bloky jsou skládány za sebe se zvyšující se časovou dilatací a tvoří jádro separačního modulu.

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 opakujících 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 sítě, výsledky a použité stroje a nástroje.

V poslední části, která je pokryta v kapitole 5 jsou shrnuty experimenty s modelem a vyhodnocení výsledků, v jehož rámci 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 s referenční studií. 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 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 vyslán signál na výstupní axon. Signál je nakonec předán dalším neuronům skrze jejich vstupní dendridy[2, p. 65–66].

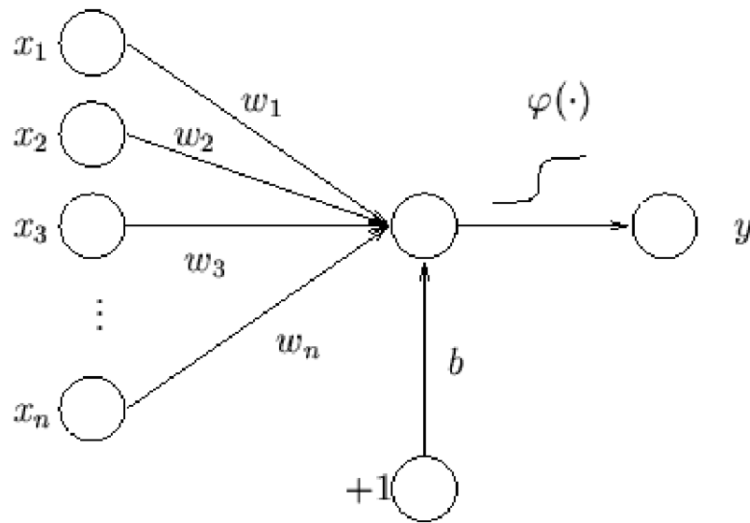
Neurony v umělé neuronové síti jsou organizovány do vrstev, kde se každá vrstva může skládat z $1 - N$ neuronů. První vrstva se nazývá vstupní, pak následují skryté vrstvy a nakonec výstupní vrstva.

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. Mezi problémy, které se dají řešit neuronovými sítěmi patří problémy v oblasti klasifikace, predikce a aproximace. 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. Na základně řešeného problému vzniklo mnoho typů neuronových sítí, z nichž popsány budou konvoluční neuronové sítě. Začneme představením základního stavebního kamene – neuronu.

2.1 Umělý neuron

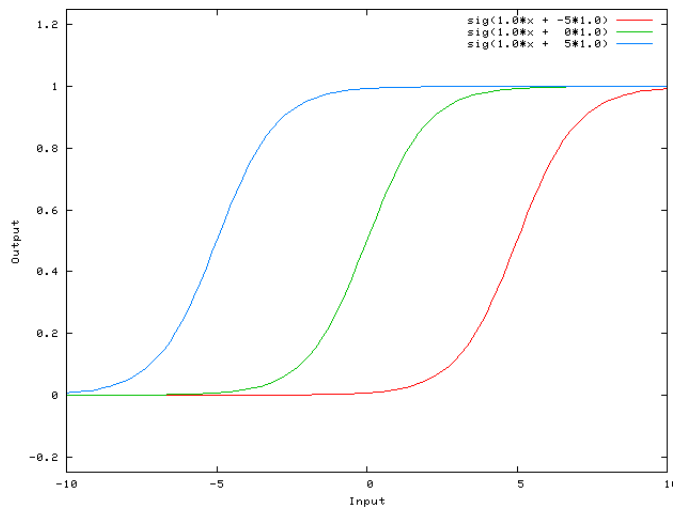
Základní stavební jednotka neuronových sítí je neuron, nebo přesněji perceptron (viz obrázek 2.1). Tento model je založen na principu reálných neuronů, které se nacházejí v organizmech. Perceptron obsahuje libovolně mnoho vstupních propojení, přes které se mu předávají data v podobě vstupního vektoru $\vec{x} = [x_1, x_2, \dots, x_n], x_n \in \mathbb{R}$. Sám neuron obsahuje hodnotu bias $b \in \mathbb{R}$ a vektor vah $\vec{w} = [w_1, w_2, \dots, w_n], w_n \in \mathbb{R}$, jehož modifikace představuje princip učení 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 vahou na daném vstupním spojení, neboli, s použitím definovaných vektorů, lze napsat, že vstupní vektor je vynásoben s vektorem vah.



Obrázek 2.1: Schéma umělého neuronu – perceptron

Hodnota bias b , která je přičtena k sumě násobků vah a vstupních hodnot, je prahová hodnota modifikující dobu, kdy se aktivuje perceptron 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 2.2. 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 jedna.



Obrázek 2.2: Vliv hodnoty bias na aktivační funkci

Výstup neuronu se tedy vypočítá jako

$$y = f\left(\left(\sum_{k=1}^n w_k x_k\right) + b\right) \quad (2.1)$$

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.

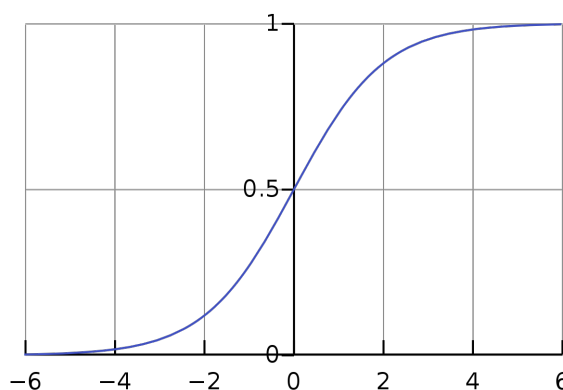
2.1.1 Aktivační funkce

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 – sigmoidní a podobně. Pro nestandardní problémy je obvykle potřeba experimentálně zjistit, která funkce bude nejlépe vyhovovat danému problému.

Sigmoid

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

[[popsat]]

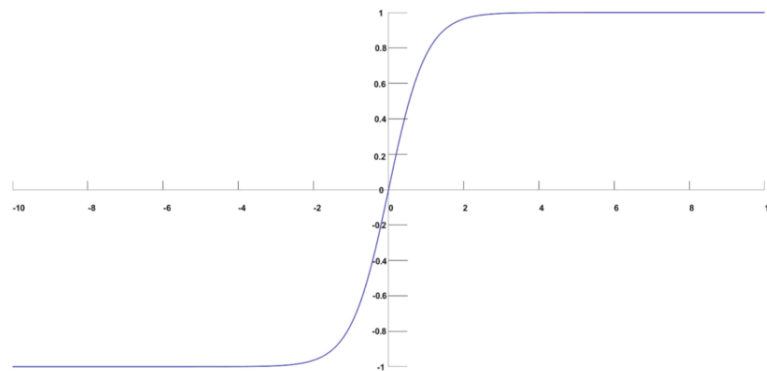


Obrázek 2.3: Graf aktivační funkce sigmoid

Softmax

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

[[popsat]]



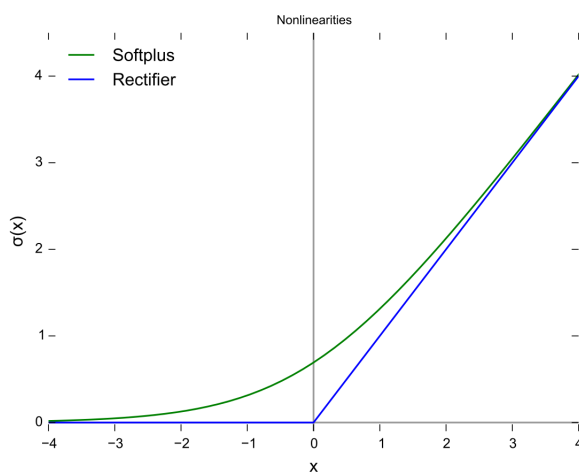
Obrázek 2.4: Graf aktivační funkce softmax

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[2, p. 77–78].

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

$$f(x) = \max(0, x) \quad (2.4)$$

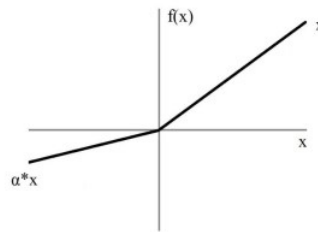


Obrázek 2.5: Graf aktivační funkce ReLU

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.

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

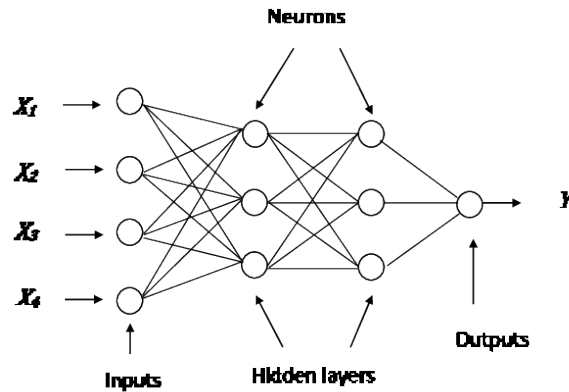


Obrázek 2.6: Graf aktivační funkce PReLU

Jeden neuron však dokáže řešit klasifikaci pouze do 2 skupin a tím je pro složitější problémy nepoužitelný, ale propojením neuronů do vícevrstvé hierarchie lze řešit téměř libovolně komplexní problém. Takové seskupení se označuje jako vícevrstvý perceptron (Multi Layer Perceptron, MLP), neboli neuronová síť.

Nejzákladnější neuronová síť je tvořena třemi typy vrstev (viz obrázek 2.7). Každá vrstva může obsahovat až $n, n \in \mathbb{N}/0$ 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.

Takto propojené neurony tvoří acyklický graf, který počítá a následně předává hodnoty směrem od vstupní vrstvy skrze skryté vrstvy až k vrstvě výstupní. Nenacházejí se zde žádná zpětná spojení, ve kterých by se výstup vracel zpět do sítě.[1]



Obrázek 2.7: Schéma neuronové sítě, která má 2 skryté vrstvy

Cílem takové neuronové sítě je aproximovat nějakou funkci f^* . Síti je předána vstupní hodnota x a výstupní hodnota $y^* = f^*(x)$ má být co nejblíží hodnotě $y = f(x)$.

2.2 Feed forward networks

[[nepřejmenovat to na cz?]] [[v rámci tohoto procesu učení, cíl učení a objektivní funkce, MLP,]]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

2.2.1 Objektivní funkce

= cost funkce -popis, co to je, k čemu to je, proč to je...

MSELoss

- vzoreček

Cross Entropy

- vzoreček

2.2.2 Optimalizační algoritmy

[1 deep learning str 301]

Adam

[[Kniha strana 301]]

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

2.2.3 Backpropagation

- zpětné šíření chyby - adaptivní algoritmus, podíl neuronu na chybě, - 3 opakující se fáze učení:

[[dodelat zde podkapitoly v lepším pořadí]]

1) feedforward - dopředu 2) zpětné šíření chyby - Backpropagation 3) úprava vah a biasů na základě chyby - chain rule

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

Gradient descent

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

2.2.4 Overfitting a generalizace

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

2.3 Konvoluční neuronové síť

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

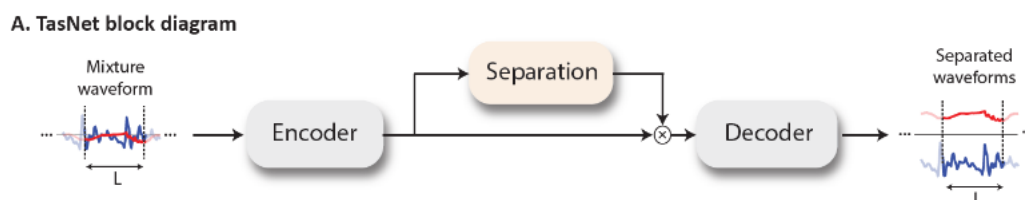
2.3.1 Konvoluce

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

Kapitola 3

TasNet - Time-Domain Audio Separation Network

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



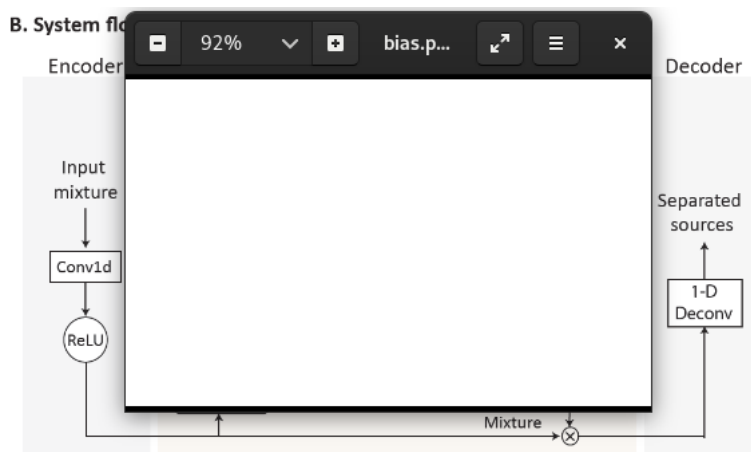
Obrázek 3.1: Zjednodušený model architektury

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

3.1 Konvoluční auto-enkodér

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

- schema bez separacního modulu - non negative representation of audio

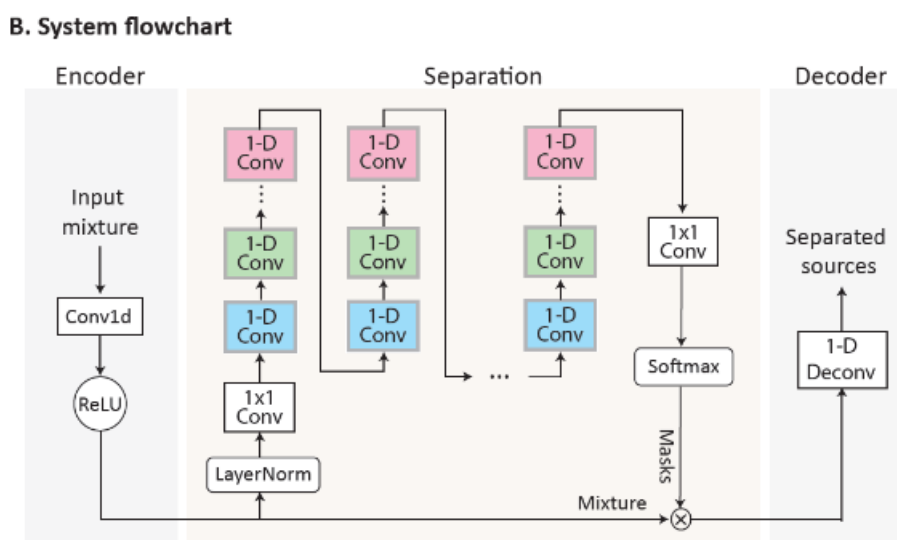


Obrázek 3.2: Schéma konvolučního autoenkodéru

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

3.2 Separační modul

- odhad masek pro jednotlivé mluvčí - schema se separacním modulem

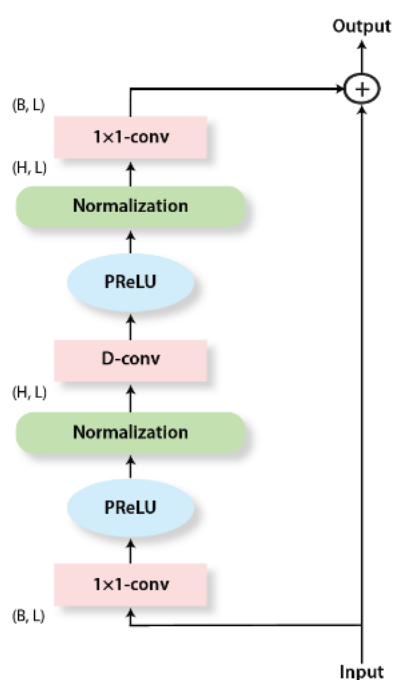


Obrázek 3.3: Schéma architektury TasNet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

3.2.1 Konvoluční bloky

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



Obrázek 3.4: Jeden konvoluční blok

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. 1

Kapitola 4

Implementace a trénování sítě

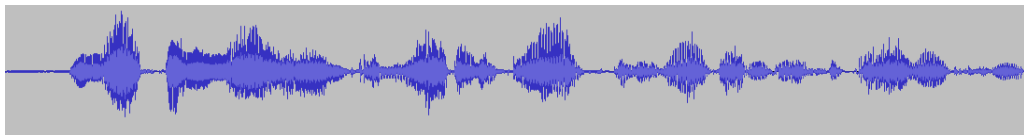
Pozn: colab, pytorch, stroj, bash, hyperparams, výkon a čas trénování, seg-len, popis tríd.

4.1 Implementace modelu

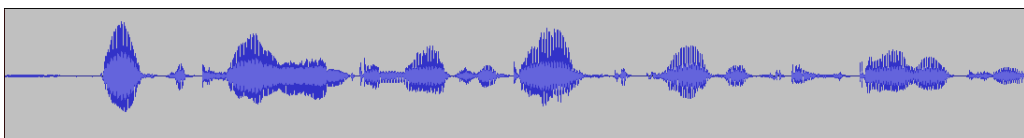
- pytorch, scripty, python3, bash, tridy, moduly, parametry a volby spuštění.

4.2 Dataset

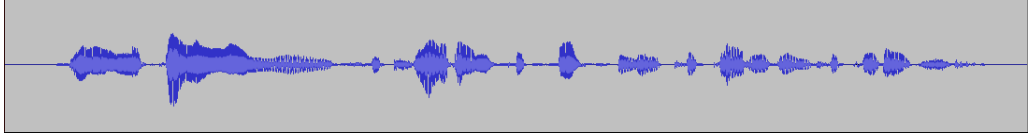
[[Ukazat zde vykreslenou vlnu nahravek mix, s1, s2]] **[[popsat co je dataset a k čemu to slouží]]** Trénování a vyhodnocení modelu proběhlo na množině jednokanálových nahrávek směsí dvou mluvčích. Množina byla vygenerována náhodným výběrem různých mluvčích z Wall Street Journal (WSJ0) a vytvořením směsi. 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 generování a popis na netu].



Obrázek 4.1: Ukázka nahrávky směsi dvou mluvčích



Obrázek 4.2: První mluvčí ze směsi



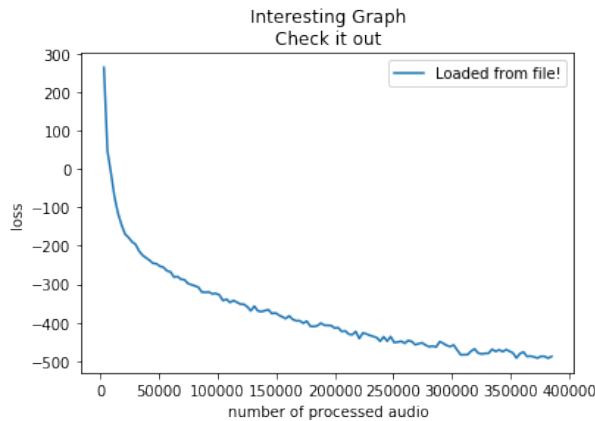
Obrázek 4.3: Druhý mluvčí ze směsi

Lze si všimnout, že sečtením signálů separovaných mluvčích na obrázku (ref obr1) a obrázku (ref obr2) dostaneme přesně signál směsi, což lze vyjádřit vztahem

$$x(t) = \sum_{i=1}^C s_i(t) \quad (4.1)$$

, kde $x(t) \in \mathbb{R}^{1 \times T}$ je diskretní signál směsi a $s_i(t) \in \mathbb{R}^{1 \times T}$, kde $i = 1, \dots, C$, je jeden z C zdrojů[ref studie str3 vlevo]. **[[doplňit info o zero means a jendotkove varianci]]**

4.3 Trénování



Obrázek 4.4: Příklad grafu loss hodnoty během učení

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 přetrénování(overfitting) 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 objektivní-hodnotící 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. - porovnaní s vysledky ze studie - obrazky separovanych mluvčích - 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,... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Literatura

- [1] IAN GOODFELLOW, A. C. *DEEP LEARNING*. MIT Press, 2017. ISBN 9780262035613.
- [2] KELLEHER, J. D. *DEEP LEARNING / John D. Kelleher*. MIT Press, 2019. ISBN 9780262537551.