TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Rozpoznávání emocí v audio nahrávkách s využitím hlubokých neuronových sítí

Bakalářská práce

Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie

Autor práce: **Tomáš Petříček**
Vedoucí práce: Ing. Lukáš Matějů, Ph.D.
Ústav informačních technologií a elektroniky

Liberec 2021

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání bakalářské práce

Rozpoznávání emocí v audio nahrávkách s využitím hlubokých neuronových sítí

Jméno a příjmení: **Tomáš Petříček**
Osobní číslo: M18000092
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávající katedra: Ústav informačních technologií a elektroniky
Akademický rok: **2020/2021**

Zásady pro vypracování:

1. Seznamte se s problematikou strojového učení s učitelem s využitím neuronových sítí.
2. Seznamte se s problematikou rozpoznávání emocí z audio nahrávek.
3. Shromážděte a předpřipravte datové sady pro rozpoznávání emocí. Vyberte vhodné emoce.
4. Vyzkoušejte různé přístupy pro rozpoznávání emocí využívající hluboké neuronové sítě.
5. Vyhodnoťte přístupy na připravené testovací datové sadě.

Rozsah grafických prací:	Dle potřeby dokumentace
Rozsah pracovní zprávy:	30-40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] BISHOP, Christopher M. Pattern recognition and machine learning. [New York]: Springer, c2006. Information science and statistics. ISBN 978-0-387-31073-2
[2] GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. Deep learning. Cambridge, MA: MIT press, [2016]. Adaptive computation and machine learning series. ISBN 978-0-262-03561-3
[3] KONAR, Amit a Aruna CHAKRABORTY. Emotion recognition: a pattern analysis approach. Hoboken, New Jersey: John Wiley & Sons, [2015]. ISBN 978-1-118-13066-7
[4] AKÇAY, Mehmet Berkehan a Kaya OĞUZ. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. Speech Communication. 2020, 116, 56-76

Vedoucí práce:	Ing. Lukáš Matějů, Ph.D. Ústav informačních technologií a elektroniky
Datum zadání práce:	19. října 2020
Předpokládaný termín odevzdání:	17. května 2021

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 19. října 2020

Prohlášení

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

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

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

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

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

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

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

13. května 2021 Tomáš Petříček

Rozpoznávání emocí v audio nahrávkách s využitím hlubokých neuronových sítí

Abstrakt

Bakalářská práce se věnuje rozpoznávání emocí v audio nahrávkách s využitím hlubokých neuronových sítí. V úvodní části je čtenář seznámen s motivací a využitím systému pro rozpoznávání emocí. Na začátku první kapitoly jsou definovány pojmy emoce a rozpoznávání emocí. Navazující podkapitoly jsou zaměřeny na vývoj systému pro rozpoznávání emocí. Další kapitola je věnována seznámení se základy neuronových sítí. Je uveden model používaný v experimentální části práce, vysvětleno trénování a vyhodnocení modelu. V následující části je popsán návrh a práce s balíčkem pro rozpoznávání emocí napsaném v jazyce Python. Poslední kapitola představuje experimenty provedené na datové sadě získané sjednocením datových sad RAVDESS, TESS, SAVEE a EMOVO. Pro klasifikaci byla použita neuronová síť typu MLP. Byla implementována a natrénována pomocí frameworku PyTorch. MFCC příznaky byly zvoleny pro učení modelu. Provedené experimenty byly zdokumentovány a rozebrány. V závěru práce jsou shrnuty výsledky a navrženy metody pro dosažení lepších výsledků. Při klasifikaci sedmi emocí bylo dosaženo přesnosti pro nahrávky 92,3 %.

Klíčová slova: rozpoznávání emocí z řeči, strojové učení, neuronové sítě, PyTorch, MFCC

Abstract

Bachelor's thesis deals with emotion recognition in audio recordings using deep neural networks. In the introductory part, the reader is familiarized with the motivation and application of speech emotion recognition system. At the beginning of the first chapter, the concepts of emotion and emotion recognition are defined. Subsequent sections are focused on the development of a speech emotion recognition system. Next chapter is devoted to acquaintance with the basics of neural networks. Model used in the experimental part of the work is presented, training and evaluation of the model are explained. The following section describes the design and workflow of emotion recognition package written in Python. The last chapter presents experiments performed on a dataset obtained by unifying the RAVDESS, TESS, SAVEE and EMOVO datasets. An MLP neural network was used for classification. It was implemented and trained using the PyTorch framework. MFCC features were selected for model learning. Performed experiments were documented and analyzed. At the end of the work, the results are summarized and methods for achieving better results are proposed. An accuracy of 92,3 % per recording has been achieved while classifying seven emotions.

Keywords: speech emotion recognition, machine learning, neural networks, PyTorch, MFCC

Poděkování

Nejprve bych chtěl poděkovat vedoucímu této práce panu doktoru Lukáši Matějů za důsledné vedení, pravidelné konzultace a užitečné rady.

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

Adam	Adaptive Momentum
BES	Berlin Emotional Speech database
CNN	konvoluční neuronová síť
DES	Danish Emotional Speech dataset
DFT	diskrétní Fourierova transformace
DNN	hluboká neuronová síť
FFN	dopředná neuronová síť
FFT	rychlá Fourierova transformace
FFmpeg	Fast Forward MPEG
GFCC	gammatone frekvenční kepstrální koeficienty
GMM	Gaussovské směsné modely
HMM	skryté Markovovy modely
HNR	poměr harmonie k hluku
HTK	Hidden Markov Model Toolkit
IDCT	inverzní diskrétní kosinová transformace
LFPC	logaritmické frekvenční silové koeficienty
LPCC	lineární predikční kepstrální koeficienty
MFCC	Melovské frekvenční kepstrální koeficienty
NLP	vícevrstvý perceptron
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song database
RNN	rekurentní neuronová síť
ReLU	usměrněná lineární aktivační funkce
SUSAS	Speech Under Simulated and Actual Stress database
SVM	metoda podpůrných vektorů
SpeechLab	Laboratoř počítačového zpracování řeči
TEO	příznaky založené na Teagerově energetickém operátoru
TIMIT	Texas Instruments a Massachusetts Institute of Technology
ZCR	metoda průchodu nulovou úrovní

1 Úvod

Řeč je naše nejpřirozenější forma komunikace a emoce nám pomáhají si lépe porozumět. Díky emocím můžeme svému okolí ukázat svůj vnitřní psychický stav. Při používání jiných forem komunikace je těžší vyjádřit své emoce, přesto lidé našli způsoby, jak je do komunikace zapojit. S rozvojem chatovácích platforem byly vyvinuty smajlíky, které reprezentují zjednodušený výraz lidského obličeje. Můžeme, tak ve zprávě poslat úplnější informaci a lépe si porozumět [1].

Nicméně i přesto, že jsme schopni vyjádřit své emoce, tak to neznamená, že se pochopíme. Emoce jsou subjektivní a každý z nás je může vnímat trochu jinak. Tato vlastnost emocí neusnadňuje ani vývoj systémů pro jejich rozpoznávání. Zatím nebyl nalezen způsob, jak emoce měřit. Proto lidé usilují o vyvinutí systémů schopných rozpoznávat emoce bez explicitně zadaných instrukcí.

Modely strojového učení jsou schopné najít skryté vzory v datech a naučit se je rozdělovat. Pro učení modelu pro rozpoznávání emocí lze použít data získaná z textu, změn výrazu tváře, hlasu, gest nebo držení těla [2]. Modely založené na rozpoznávání emocí z řeči mohou najít uplatnění na příklad při vývoji virtuálních asistentů.

V posledních letech došlo k rozvoji osobních virtuálních asistentů jako jsou Siri a Alexa, které jsou používány pro hlasové ovládání elektronických zařízení. Mohou odesílat textové zprávy, přijímat telefonní hovory, přehrávat hudbu nebo vyhledávat ve webovém prohlížeči. Systém schopný rozpoznávat emoce může zlepšit komunikaci s asistentem tak, aby se nám zdála přirozenější [3].

Dále může model najít uplatnění pro call centra. Data generovaná call centrem mohou posloužit k vývoji automatické obsluhy zákazníků nebo pro optimalizaci práce v call centru. Dispečerovi můžou být, podle emocionálního stavu zákazníka, nabídnuty scénáře, podle kterých je vhodné v dané situaci postupovat. Model může být využit také k vylepšení systémů doporučující videa nebo podcasty. Uplatnění může najít také při vývoji realističtějších her. Zdroje dat jsou různé. V dnešní době převažují uměle vytvořené datové sady [2].

Práce se zabývá aktuálním tématem a výsledky mohou vést k rozšíření systémů vyvíjených Laboratoří počítačového zpracování řeči (SpeechLab) [1]. Model by se mohl implementovat na příklad do monitorovacího systému pro přepis TV a radio streamů v různých jazycích.

[1] https://www.ite.tul.cz/speechlab/

2 Základy rozpoznávání emocí

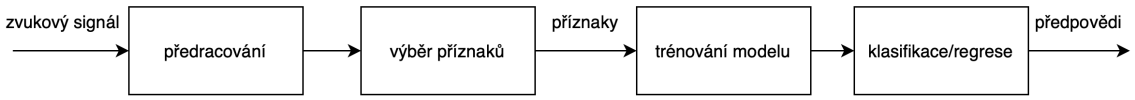
2.1 Emoce

V současnosti není zvolena jednotná definice emocí. Existuje jich mnoho. Emoce popisují náš vnitřní stav a jejich tvorba je ovlivněna mnoha faktory, jako je osobní zkušenost, fyzické, jednací a komunikační reakce. Pro úlohu rozpoznávání emocí je důležité vědět, jak lze emoce rozdělit [1].

Emoce můžeme dělit dvěma způsoby podle diskrétního modelu nebo prostorového modelu. Diskrétní model rozděluje emoce do kategorií. Mezi hlavní kategorie patří: smutek, radost, strach, hněv, znechucení a překvapení. Prostorový model dělí emoce do jednotlivých prostorů jako jsou mocenství, vzrušení nebo vliv. Vzrušení udává sílu emoce a má rozsah od znudění k nadšení. Výhodou diskrétního modelu je, že na rozdíl od prostorového modelu je intuitivnější [1].

2.2 Rozpoznávání emocí

Při vývoji typického systému pro rozpoznávání emocí z řeči je nejprve vybrána anotovaná datová sada. Data ze sady jsou nejdříve předzpracována. Dále jsou z nich vytaženy příznaky pro rozpoznávání a jsou z nich vybrány nejvhodnější pro rozpoznávání. Pomocí anotovaných příznaků lze natrénovat model strojového učení z kategorie učení s učitelem. Výsledný model je buď typu klasifikátor, pokud jsou emoce děleny diskrétně nebo regresor v případě, že jsou děleny prostorově. Výsledkem klasifikace nebo regrese jsou předpovědi [2]. Postup vývoje systému pro rozpoznávání emocí z řeči je znázorněn na obrázku 2.1.



Obrázek 2.1: Postup při rozpoznávání emocí

2.3 Současné poznání

Při rozpoznávání emocí z řeči není určeno, které emoce rozpoznávat a jestli pro rozpoznávání používat diskrétní nebo prostorový model. Nicméně při použití dis-

krétního modelu jsou vymezeny takzvané velké n emoce mezi než patří na příklad: hněv, strach, smutek nebo radost [2].

V současné době existuje mnoho datových sad. Datové sady se od sebe liší především vznikem, počtem emocí a jazykem. Podle způsobu vzniku jsou datové sady děleny na předstírané, vybuzené a přirozené. Mezi hojně používané předstírané datové sady patří Danish Emotional Speech dataset (DES) [4], Berlin Emotional Speech database (BES) [5], Speech Under Simulated and Actual Stress database (SUSAS) [6] nebo Ryerson Audio-Visual Database of Emotional Speech and Song database (RAVDESS) [7]. Mezi vybuzené datové sady patří na příklad eNTERFACE [8]. Většina datových sad má do jedné hodiny délky a nahrávky jsou namluveny kolem deseti mluvčích. Nicméně při rozpoznávání řeči je typicky potřeba datová sada s několika sty hodinami nahrávek a větší různorodost mluvčích [2].

Příznaky používané pro rozpoznávání emocí lze dělit na prozodické, spektrální, založené na kvalitě hlasu nebo příznaky založené na Teagerově energetickém operátoru (TEO). Mezi prozodické příznaky patří na příklad intonace nebo rytmus. K příznakům založeným na spektru lze přiřadit Melovské frekvenční kepstrální koeficienty (MFCC), lineární predikční kepstrální koeficienty (LPCC) nebo logaritmické frekvenční silové koeficienty (LFPC). Chvění, mihotání nebo poměr harmonie k hluku (HNR) jsou příznaky založené na kvalitě zvuku [1].

Pro rozpoznávání emocí jsou používány buď klasifikátory nebo regresory, podle toho jestli jsou emoce děleny na základě diskrétního nebo prostorového modelu. Z tradičních modelů strojového učení lze pro klasifikaci použít skryté Markovovy modely (HMM), Gaussovské směsné modely (GMM) nebo metodu podpůrných vektorů (SVM). Z řady klasifikátorů typu hluboká neuronová síť (DNN) lze použít vícevrstvý perceptron (NLP), konvoluční neuronové sítě (CNN) nebo rekurentní neuronové sítě (RNN) [1].

3 Datové sady

K rozpoznávání emocí se používají anotované datové sady. Mohou být děleny podle toho, jak byla data získána na předstírané, vybuzené a přirozené datové sady. Také se mohou lišit tím, kdo je namluvil. Mluvčí mohou být různého věku, pohlaví a mluvit různými jazyky. Datové sady se také odlišují tvrzeními, která byla vyslovena. Počet a kategorie rozpoznávaných emocí mohou být také různé [1]. V tabulce 3.1 je uveden přehled volně dostupných datových sad.

3.1 Předstírané datové sady

Data mohla být pořízena předstíráním, kdy herec při nahrávání předstírá, že emoci prožívá. Tento typ datové sady lze získat spoluprací s profesionálními herci nebo z video záznamů filmů a seriálů. Datovou sadu při spolupráci s herci lze relativně snadno sestavit, protože tvůrci mají poměrně velkou kontrolu nad celým procesem. Data získaná tímto způsobem, ale nemusejí odpovídat reálné situaci, a proto existují další dva způsoby získávání datových sad [2].

3.2 Vybuzené datové sady

Při získávání vybuzených datových sad je mluvčí umístěn do situace, která se velice podobá reálnému životu. Situace jsou většinou vybírány tak, aby odpovídaly potenciálnímu použití [2].

3.3 Přirozené datové sady

Poslední způsob získávání dat pro rozpoznávání je z přirozené řeči. Data mohou být získána na příklad z rozhovorů z radií, televizních show nebo záznamů z call center. Ačkoli by tato data měla být pro rozpoznávání nejvhodnější, tak je mnohem obtížnější z nich sestavit datovou sadu. Na data se mohou vztahovat právní nároky a je s nimi více práce při zpracování [1].

Tabulka 3.1: Přehled volně dostupných datových sad [1]

Název	**Jazyk**	**Počet tříd**	**Počet mluvčích**	**Pohlaví mluvčích**	**Typ**	**Velikost**
RAVDESS	angličtina	5,7	24	obě	předstíraná	2452 promluv
SAVEE	angličtina	7	14	můži	předstíraná	480 promluv
TESS	angličtina	6	2	ženy	předstíraná	2800 promluv
CHEAVD	mandariština	8	238	obě	předstíraná	2600 promluv
DES	dánština	5	4	obě	předstíraná	10 min řeči
EESDB	mandariština	7	16	obě	předstíraná	400 promluv
EMA	angličtina	4	3	obě	předstíraná	34 promluv
EMOVO	italština	7	6	obě	předstíraná	588 promluv
eNTERFACE'05	angličtina	6	42	obě	vybuzená	1166 promluv
Keio-ESD	japonština	47	71	muži	předstíraná	940 promluv
RECOLA Speech Database	francouzština	7	46	obě	přirozená	1308 promluv
SAMAINE Database	angličtina, řečtina, hebrejština	5	150	není známo	přirozená	959 konverzací
VAM	němčina	3	47	není známo	přirozená	947 promluv
TUM AVIC Database	angličtina	5	21	obě	přirozená	3901 promluv
AFEW Database	angličtina	7	330	není známo	přirozená	1426 promluv
TURES	turečtina	9	582	obě	předstíraná	5100 promluv
BAUM-1 Speech Database	turečtina	12	31	obě	předstíraná	1222 promluv

3.4 Vybrané datové sady

V následujících oddílech jsou podrobněji popsány datové sady použité v praktické části práce. Jsou uvedeny tři anglické datové sady a jedna italská. Všechny sady jsou předstírané.

3.4.1 RAVDESS

RAVDESS [7] označuje anglickou datovou sadu obsahující nahrávky řeči a písní. Spolu s nahrávkami zvuku byly pořízeny i video záznamy mluvčích. Nahrávky byly namluveny 24 herci, z nichž bylo dvanáct žen a dvanáct mužů, což činí datovou sadu pohlavně vyrovnanou. Mluvčí mluvili severoamerickou angličtinou. Nahrávky řeči zachycují osm emocí: klid, radost, smutek, hněv, strach, překvapení, znechucení a neutrální stav. Každý herec namluvil dvě tvrzení ve dvou úrovních emocionální intenzity, běžnou a silnou, pro všechny emoce. Namluvené výrazy byly „Kids are talking by the door" a „Dogs are sitting by the door". Počet zvukových nahrávek řeči je celkově 1440 a mají dobu trvání 1,5 hodiny. Každá nahrávka v datové sadě má přiřazenou anotaci, která udává druh nahrávky, druh emoce, emocionální intenzitu, tvrzení, číslo opakování a herce, který nahrávku namluvil. Délky nahrávek se pohybují kolem třech minut.

3.4.2 SAVEE

SAVEE [9] označuje anglickou datovou sadu pro rozpoznávání emocí. Obsahuje 480 promluv a má celkovou trvání půl hodiny. Rozlišuje sedm emocí, mezi které patří hněv, znechucení, strach, radost, smutek, překvapení a neutrální stav. Byla namluvena čtyřmi mužskými herci mluvícími britskou angličtinou ve věku mezi 27 až 31 lety. Pro tvorbu datové sady bylo vybráno 15 vět z datové sady Texas Instruments a Massachusetts Institute of Technology (TIMIT) [10]. Kromě řeči byla při nahrávání zaznamenána i mimika obličeje mluvčích, kteří na něm měli na obličeji namalovány modré značky.

3.4.3 TESS

TESS [11] označuje anglickou datovou sadu pro rozpoznávání emocí. Obsahuje 2800 promluv, které byly namluveny dvěma herečkami ve věku 26 a 64 let. Má celkovou délku trvání 1,6 hodiny. Rozlišuje sedm emocí mezi něž patří hněv, znechucení, strach, radost, překvapení, smutek a neutrální stav. Každá herečka namluvila 200 promluv pro všechny emoce. Promluva vždy začíná slovy "Say the word" a končí jedním z 200 vybraných slov [12].

3.4.4 EMOVO

EMOVO [13] je italská datová sada pro rozpoznávání emocí. Zvukové nahrávky byly vytvořeny šesti herci, z nichž tři byli muži a tři ženy. Každý herec vyslovil 14 vět pro

každou emoci. Datová sada rozlišuje sedm emocí, mezi něž patří znechucení, strach, hněv, radost, překvapení, smutek a neutrální stav. Skládá se z 588 promluv a má délku trvání půl hodiny [13].

4 Předzpracování dat pro rozpoznávání

Mnoho oblastí použití neuronových sítí vyžaduje důkladné předzpracování dat, jelikož původní podoba dat může být příliš složitá pro interpretaci neuronovou sítí [14]. Při předzpracování řeči mohou být použity techniky pro zvýšení kvality signálu, jako jsou odstranění ticha nebo šumu. Při získávání příznaků z řeči je často používáno rámcování a okénkování [2].

4.1 Rámcování

Rámcování bývá prvním krokem předzpracování dat pro rozpoznávání, kdy je zvukový signál rozdělen na menší časové úseky zvané rámce, které nabývají většinou rozsahu mezi 10–30 milisekundami. Často se jednotlivé rámce překrývají z 30–50 %, aby se zachoval vztah mezi jednotlivými rámci [1]. Důvodem rámcování je, že se emoce v průběhu řeči mohou měnit a rozdělení na menší časové úseky zajistí, že emoce zůstane v rámci jednoho úseku stejná [2].

4.2 Okénkování

Po rámcování většinou přichází okénkování, kdy je na jednotlivé rámce použita okénkovácí funkce. Snižuje amplitudu signálu na jeho okrajích, a tím snižuje úniky, ke kterým může dojít při použití rychlé Fourierovi transformace (FFT). Pro okénkování lze použít Hammingovu okénkovací funkci [1].

4.3 Odstranění ticha

Dále mohou být použity techniky pro detekci řeči. S jejich pomocí lze ze signálu odstranit tichá místa. Mohou být použity metody jako je metoda průchodu nulovou úrovní (ZCR), krátkodobá energie nebo autokorelační metody. ZCR udává míru přechodu signálu z kladných do záporných hodnot nebo naopak, a to v rámci jednoho úseku. Hodnota ukazatele je nízká v místech řeči a vysoká v místech ostatních. Při použití krátkodobé energie dosáhne vysokých hodnot v hlasové části a nízkých hodnotu v částech ostatních. Techniky pro odstranění tichých míst v řeči mohou snížit počet dat a zvýšit jejich přínos pro učení [1].

4.4 Normalizace

Může být použita také normalizace, která zmírňuje rozdíly v řeči mezi mluvčími a zároveň zachová přenášenou informaci. Normalizace může být použita na více úrovních, na úrovni jednotlivých rámců nebo na úrovni celé datové sady. Nejčastěji se pro normalizaci používá z-normalizace [1].

5 Výběr příznaků

Existuje mnoho příznaků extrahovaných z řeči, ale nejsou určeny příznaky, které by se hodily přímo pro rozpoznávání emocí. Ze signálu lze získat jak souhrnné, tak lokální příznaky pro rozpoznávání. Mezi souhrnné příznaky patří střední hodnota, směrodatná odchylka, minimální nebo maximální hodnota. Lokální příznaky lze získat z jednotlivých rámců a zastupují krátkodobé změny v signálu [1].

Příznaky pro rozpoznávání emocí v řeči můžeme dělit na prozodické, spektrální, založené na kvalitě hlasu nebo příznaky založené na TEO. Nejčastěji používané příznaky při rozpoznávání emocí v řeči jsou prozodické a spektrální příznaky [1].

5.1 Prozodické příznaky

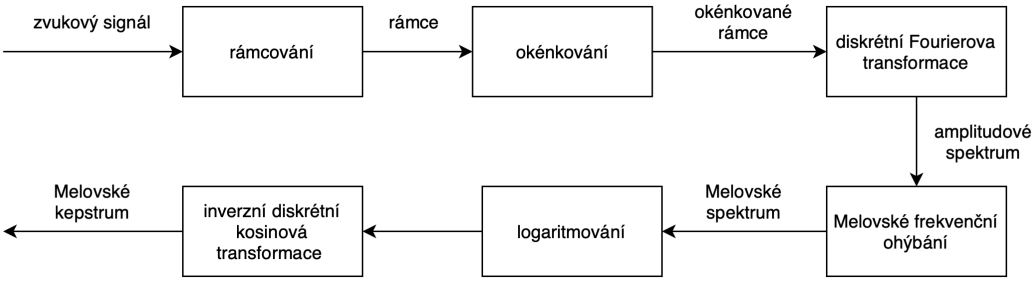
Mezi prozodické příznaky patří na příklad intonace nebo rytmus. Řadí se mezi souhrnné příznaky, protože je lze získat z delších hlasových úseků jako jsou hlásky, slova nebo věty. Mezi používané příznaky patří základní frekvence, energie nebo doba trvání. Energie udává míru změny amplitudy signálu v čase. Emoce jako hněv, štěstí nebo překvapení vykazují zvýšenou energii. Na straně druhé znechucení a smutek vykazují energii nízkou. Základní frekvence se postupně snižuje při projevení hněvu, a naopak stoupá při projevení radosti. Doba trvání potřebná k projevení hněvu je obecně kratší než při projevení smutku [1].

5.2 Příznaky založené na spektru

Příznaky založené na spektru jsou získávány pomocí Fourierovy transformace, kdy je signál převeden z oblasti časové do frekvenční. Mohou přinést hlubší porozumění signálu než příznaky prozodické. Příznaky jsou získávány z rámců, na které byla použita okénkovací funkce. MFCC jsou založeny na krátkodobém spektru signálu. Mezi další příznaky patří LPCC, LFPC nebo gammatone frekvenční kepstrální koeficienty (GFCC) [1].

5.2.1 Melovské frekvenční kepstrální koeficienty

Melovské frekvenční kepstrální koeficienty (MFCC) jsou příznaky používané pro rozpoznávání řeči. Technika převodu zvukového signálu na příznaky MFCC spočívá v provedení několika kroků zobrazených na obrázku 5.1.



Obrázek 5.1: Postup získání příznaků MFCC

Při rámcování je zvukový signál nejprve rozdělen na jednotlivé posuvné rámce. Dále je podroben okénkování, kdy je amplituda signálu snížena na konci a na začátku rámce. Může k tomu být použito Hammingovo nebo Hanningovo okénko. Poté je signál převeden pomocí diskrétní Fourierovi transformace (DFT) do frekvenční oblasti. Na výsledné amplitudové spektrum je použito Melovské frekvenční ohýbání. K převodu je použita Melovská banka filtrů složená z trojúhelníkových filtrů s měřítkem v Melovské škále, která zohledňuje vnímaní zvukových frekvencí člověkem. Lidé lépe rozlišují mezi nízkými frekvencemi než mezi vysokými. Dále je signál logaritmován a je provedena inverzní diskrétní kosinová transformace (IDCT). Z koeficientů Melovského kepstra, nazývaných také nulté koeficienty, lze získat koeficienty delta a akcelerační koeficienty (delta delta), které jsou získány použitím první a druhé derivace na nulté koeficienty [15].

5.3 Příznaky založené na kvalitě zvuku

Chvění, mihotání nebo HNR jsou příznaky založené na kvalitě zvuku. Chvění je měřeno na základě nestálosti frekvence a mihotání je založeno na nestálosti amplitudy. HNR udává poměr mezi hlukem a frekvenčním spektrem samohlásek [1].

5.4 Příznaky založené na Teagerově energetickém operátoru

Příznaky založené na Teagerově energetickém operátoru (TEO) jsou používány k detekci stresu. Využívají změny svalového napětí, které při stresové situaci nastává. V této kategorii předstihují příznaky jako jsou MFCC nebo výška tónu [1].

6 Vybrané základy neuronových sítí

6.1 Způsoby rozpoznávání emocí

Pro rozpoznávání emocí jsou používány klasifikátory nebo regresory, které spadají do oblasti strojového učení s učitelem [1]. Modely strojové učení s učitelem vyžadují označená data. Každý vzorek musí mít přiřazenou anotaci udávající třídu, do které patří. V případě klasifikace jsou to diskrétní třídy, které odpovídají jednotlivým emocím jako jsou na příklad hněv, radost nebo smutek. V případě regrese jsou to desetinné hodnoty, které označují stupeň mocenství, vzrušení nebo dominance většinou v rozsahu od -1 do $+1$ [2].

Často se při klasifikaci ke vzorku přidávají i vzorky z okolí pro zvýšení kontextu, který může zlepšit výsledky rozpoznávaní. Důvodem je, že emoce ovlivňují dlouhodobé charakteristiky řeči. Přesný počet rámců, který by se měl vzít není určen. Další způsob, jak klasifikovat emoce je podle statických příznaků jako jsou maximum, minimum nebo časová délka [2].

Při trénování se často používá křížová validace, kdy se mění trénovací a validační sada a je vytvořeno několik modelů, jejichž výsledky jsou zprůměrovány, tím je dosaženo smysluplnějších výsledků. U rozpoznávání emocí je používána především, protože jsou datové sady po většinou menšího rozsahu, a tak má rozdělení do jednotlivých sad větší vliv na výsledek. Jako metrika pro ohodnocení modelu se používá přesnost nebo vážená přesnost, která zohledňuje odlišný počet vzorků pro každou třídu. Přesnost udává pravděpodobnost, že daný vzorek patří do předpovězené třídy [2].

6.2 Vícevrstvý perceptron

Vícevrstvý perceptron (MLP) neboli dopředná neuronová síť (FFN) je složena z více lineárních vrstev. Mezi lineární vrstvy jsou umístěny vrstvy nelineární. Lineární vrstva se skládá z parametrů vah a biasů a nelineární vrstva je tvořena aktivační funkcí [16].

Aby mohla být neuronová síť označena za hlubokou neuronovou síť, musí mít dvě a více lineární vrstvy. První vrstva modelu se nazývá vstupní vrstvou a poslední vrstvou výstupní. Vrstvy mezi vstupní a výstupní vrstvou jsou označovány jako vrstvy skryté. Každá lineární vrstva má svoji šířku, která určuje počet neuronů ve vrstvě. Při klasifikaci do dvou tříd následuje za výstupní vrstvou funkce sigmoida a při klasifikaci do dvou a více tříd funkce softmax.

Jako aktivační funkce ve skrytých vrstvách jsou pro MLP používány nejčastěji funkce sigmoida a ReLU. Neuronovou síť typu MLP lze použít také pro regresi, kdy jsou jako aktivační funkce v nelineárních vrstvách umístěny identity [16].

6.3 Trénování

Při trénování neuronové sítě je provedena nejprve dopředná propagace následovaná zpětnou propagací. Po zpětném průchodu sítí jsou parametry sítě upraveny pomocí metody největšího spádu, která minimalizuje ztrátu modelu. Jeden proces učení nazýváme epochou.

Při dopředném průchodu lineární vrstvou jsou vstupní příznaky y_i přeměněny na výstupní příznaky x_j pomocí lineární kombinace, kterou lze vyjádřit:

$$x_j = b_j + \sum_{i}^{N} y_i w_{ij}, \tag{6.1}$$

kde proměnná i odpovídá indexu příznaku z předchozí vrstvy. N je celkový počet neuronů předchozí vrstvy. Proměnné w_{ij} a b_j reprezentují parametry lineární vrstvy [17]. Dále jsou příznaky aktivovány nelineární funkcí, kterou lze obecně vyjádřit:

$$z_j = h(x_j), \tag{6.2}$$

kde x_j jsou vstupní příznaky z předchozí vrstvy. Proměnná h je nelineární diferencovatelná aktivační funkce a z_j jsou výstupní příznaky [16].

Proces se opakuje v každé vrstvě sítě. Při dopředném průchodu je nutné si zapamatovat hodnoty vstupních příznaků jednotlivých vrstev sítě, které jsou používány při zpětné propagaci. Výsledkem dopředného průchodu jsou předpovězené hodnoty, z kterých lze pomocí kriteriální funkce získat ztrátu.

Při zpětné propagaci se ztráta modelu propaguje zpět sítí pomocí řetízkového pravidla. Po získání všech gradientů sítě se provede optimalizační algoritmus metody největšího spádu, který aktualizuje parametry sítě. Metodu nejvyššího spádu lze vyjádřit:

$$x_{t+1} = x_t - \alpha \frac{d}{dx} J(x_t), \tag{6.3}$$

kde proměnná x_{t+1} je hodnota aktualizovaného parametru sítě a x_t je hodnota původního parametru sítě. Hodnota gradientu pro parametr x je $\frac{d}{dx}$. Ztráta parametru x_t je označena J [16].

6.4 Rozdělení na datové sady

Po získání datové sady je většinou sada rozdělena na tři části: trénovací, validační a testovací. Trénovací sada je často největší a slouží k trénování modelu. Validační a testovací sady jsou menší a slouží k vyhodnocení modelu. Při rozdělování menších

datových sad je často použito 70 % pro trénování, 15 % pro validaci a 15 % pro testování. Při rozdělování větších datových sad je možné použít větší část 95 % pro trénování a zbytek pro validaci a testování. Validační datová sada se používá pro zvolení vhodného modelu a vhodných hyperparametrů. Testovací datová sada je použita v posledních částech vývoje modelu, kdy jsou na ní zjištěny konečné výsledky modelu [18].

6.5 Hyperparametry

Hyperparametry jsou parametry ovlivňující učení modelu. Na rozdíl od parametrů modelu se je nelze naučit z trénovacích dat [19]. Můžeme je dělit podle toho, jestli jsou používány při návrhu neuronové sítě nebo při samotném trénování. Při návrhu neuronové sítě lze na příklad upravovat počet a šířku skrytých vrstev. Mezi hyperparametry používané při trénování modelu lze zařadit míru učení, počet trénovacích epoch nebo velikost dávky [20].

6.6 Přeučování a nedoučování

Při trénování modelu je možné se setkat s nedoučováním a přeučováním. K nedoučování dochází, když model dosahuje nízkých výsledků na trénovací sadě. Nedoučování může být způsobeno na příklad přílišnou jednoduchostí modelu nebo zvolením nevhodných příznaků pro učení. Problém lze vyřešit vyzkoušením složitějšího modelu nebo zvolením lepších příznaků pro učení. Při přeučování dochází k opačnému jevu, kdy model velice dobře rozpoznává vzorky z trénovací sady, ale velice špatně vzorky ostatní. Mezi hlavní důvody pro vznik přeučování patří přílišná složitost modelu nebo nadměrné množství příznaků pro malou trénovací sadu. Přeučování na datové sadě můžeme poznat na příklad na validační sadě, kdy ztráta modelu stoupá. Mezi nejčastější techniky používané proti přeučování patří použití jednoduššího modelu, zmenšení počtu vstupních příznaků, přidání trénovacích dat nebo použití regularizace [18].

6.7 Regularizace

Regularizace označuje množinu technik, které zabraňují vytvoření příliš složitého modelu. Často jsou používány regularizace typu L1 a L2, které jsou umístěny v algoritmu pro upravovaní parametrů modelu. Je možné použít také techniky, které jsou součástí modelu samotného jako je dropout nebo vrstvy normalizace dávek [18].

6.7.1 Dropout

Princip dropout spočívá v náhodném vypínání neuronů ve vrstvě během trénování. Dropout bere jako parametr pravděpodobnost vypnutí jednoho neuronu. Čím vyš-

ší je pravděpodobnost vypnutí, tím stoupá účinek regularizace. Během používání a testovaní neuronové sítě k vypínání neuronů nedochází [18].

6.7.2 Normalizace dávky

Normalizace dávky spočívá v normalizaci výstupu z předešlé vrstvy modelu před vstupem do následující. Použití normalizace může vést k rychlejšímu a stabilnějšímu trénování [18]. Vztah pro výpočet normalizace dávky lze vyjádřit pomocí následujícího vzorce:

$$z_i = \frac{x_i - \mu_b}{\sqrt{\sigma_b^2}}, \tag{6.4}$$

kde z_i je normalizovaný vzorek. Hodnota μ_b označuje střední hodnotu dávky a σ_b^2 znázorňuje rozptyl dávky. Parametr x_i je jeden vzorek dávky [18].

6.8 Hodnocení úspěšnosti modelu

Model lze hodnotit podle průběhu ztráty modelu na datových sadách. Pokud ztráta klesá u všech sad, tak je předpoklad, že se model učí správně. Při klasifikaci lze použít více metrik a nástrojů pro vyhodnocení modelu. Lze počítat na příklad přesnost (accuracy), precisnost (precision), úplnost (recall) nebo matici záměn [18].

6.8.1 Přesnost

Přesnost udává počet všech správně klasifikovaných vzorků vůči počtu všech vzorků. Často se hodnota násobí stem, aby výsledek vyšel v procentech [18]. Přesnost lze znázornit následujícím vzorcem:

$$ACC = \frac{H}{N}, \tag{6.5}$$

kde ACC je přesnost. H je počet správně klasifikovaných vzorků a N je celkový počet vzorků [18].

6.8.2 Přesnost pro vzorek

Během epochy trénování jsou vzorky dávky podrobeny dopřednému průchodu sítí. Z výstupu neuronové sítě jsou získány předpovědi, které jsou porovnány s anotacemi a jsou zjištěny počty správně klasifikovaných vzorků. Na konci epochy je spočítána přesnost pro vzorek pomocí následujícího vzorce:

$$ACC_S = \frac{H_S}{N_S}, \tag{6.6}$$

kde ACC_S je přesnost pro vzorky. H_S je počet správně klasifikovaných vzorků a N_S je celkový počet vzorků.

6.8.3 Přesnost pro nahrávku

Při testování a validaci modelu se velikost dávky mění a odpovídá velikosti nahrávky. Během epochy jsou vzorky nahrávky podrobeny dopřednému průchodu sítí. Výstup neuronové sítě je průměrován a je získán výstup pro celou nahrávku, z kterého je zjištěna předpověď pro nahrávku. Předpověď je porovnána s anotací nahrávky. Pokud byla předpověď správná zvýší se počet správně klasifikovaných nahrávek o jedna. Na konci epochy je spočítána přesnost pro nahrávku pomocí vzorce:

$$ACC_R = \frac{H_R}{N_R}, \tag{6.7}$$

kde ACC_R je přesnost pro nahrávku. H_R je počet správně klasifikovaných nahrávek a N_R je celkový počet nahrávek.

6.8.4 Matice záměn

Matice záměn shrnuje výkon modelu. Na jedné z os matice jsou vyobrazeny správné třídy a na druhé třídy předpovězené. Jednotlivé prvky matice záměn udávají počet přiřazených vzorků. Pokud by byly všechny předpovědi správné, byly by hodnoty větší než nula jen na hlavní diagonále matice. Příklad matice záměn pro klasifikaci do dvou tříd, pozitivní a negativní, je znázorněn na obrázku 6.1.

Obrázek 6.1: Matice záměn pro dvě třídy

kde TP a TN jsou počty správně klasifikovaných pozitivních a negativní tříd. FP a FN jsou množství nesprávně přiřazených pozitivních a negativních tříd [18].

6.9 Vybrané koncepty

Následující oddíly jsou věnovány vybraným konceptům strojového učení, které jsou použity v praktické části práce. Je zmíněna kriteriální funkce křížová entropie, která je použita pro zjištění ztráty modelu. Popsána je také funkce softmax, která je použita k výpočtu pravděpodobností pro jednotlivé třídy z výstupu neuronové sítě.

6.9.1 Křížová entropie

Jako kriteriální funkci pro klasifikaci do více tříd lze použít křížovou entropii, která měří rozdíl mezi dvěma rozděleními pravděpodobnosti. Lze ji vyjádřit:

$$l = -\sum_{c}^{C} t_c \log p_c, \tag{6.8}$$

kde l je výsledná ztráta. Index c odpovídá klasifikovaným třídám a C je počet tříd. Proměnná p značí vstupní pravděpodobnosti pro jednotlivé třídy, které dosahují hodnot od nuly do jedné. t zastupuje skutečné třídy. Minimalizací křížové entropie minimalizujeme rozdíl mezi pravděpodobnostním rozdělením trénovacích dat a pravděpodobnostním rozdělením předpovídaných hodnot [21].

6.9.2 Usměrněná lineární aktivační funkce

Usměrněná lineární aktivační funkce (ReLU) je aktivační funkce, jejíž výstupem je buď nula, pokud je vstup záporný nebo se hodnota vstupu nezmění, pokud je kladný. Lze ji vyjádřit:

$$y = max(0, x), \tag{6.9}$$

kde y jsou výstupní hodnoty. Proměnná x zastupuje vstupní hodnoty. Neuronové sítě používající tuto funkci se většinou učí rychleji a dosáhnou lepších výsledků. Výhodou této funkce je, že nepodléhá přesycení, kdy jsou velká čísla na příklad u funkce sigmoid změněna na jedna a velmi malá na nula. Důsledkem přesycení je, že jsou funkce jako sigmoid nebo tanh velice citlivé na hodnoty okolo jejich středu a méně na odlehlejší hodnoty. Následně se to projeví při trénovaní modelu, kdy může dojít k problému mizejícího gradientu u hlubších neuronových sítí [22].

6.9.3 Softmax

Softmax je funkce, která mění vektor čísel na vektor pravděpodobností, kde jsou pravděpodobnosti pro jednotlivé prvky úměrné velikosti všech prvků vektoru. Funkci lze vyjádřit vzorcem:

$$p_i = \frac{e^{s_i}}{\sum_{c}^{C} e^{s_c}}, \tag{6.10}$$

kde p_i je hodnota výstupní pravděpodobnosti. Proměnná s odpovídá skóre výstupní vrstvy neuronové sítě. Index i odpovídá třídě výstupní pravděpodobnosti a index c třídám skóre. Hodnota C je rovna počtu klasifikovaných tříd. Součet všech výsledných pravděpodobností je roven jedné. Funkce se používá při klasifikaci do více tříd a stojí na konci klasifikátoru. Každá výstupní pravděpodobnost odpovídá jednotlivé třídě datové sady [23].

7 Vývoj balíčku pro rozpoznávání emocí

Byl vytvořen balíček s moduly pro rozpoznávání emocí. Hlavní motivací pro vznik byla snadná práce s daty, trénování a vyhodnocení úspěšnosti modelu. Balíček je napsán v jazyku `Python`[1] verze 3. Mezi vytvořené moduly patří: `convertors` pro převod dat, `data` pro práci s daty, `classifiers` na tvorbu klasifikátoru, `files` na práci se soubory, `datasets` k tvorbě datových sad, `prepare` pro přípravu datových sad a `train` sloužící k trénování modelu. Moduly jsou psány objektově. Byly použity knihovny a frameworky: `PyTorch`[2] 1.6 pro tvorbu a trénování modelu, `seaborn` 0.11 a `matplotlib` 3.3 pro vytvoření grafů, `logging` 0.4 pro logování při trénování modelu, `scikit-learn` 0.23 pro rozdělení dat na datové sady, `pandas` 1.1 a `numpy` 1.18 pro ukládání a práci s daty, `os` a `sys` pro práci s operačním systémem, `re` pro parsování anotací, `subprocess` pro spouštění příkazů z příkazové řádky, `PyHTK`[3] pro načítání souborů vytvořených pomocí sady nástrojů Hidden Markov Model Toolkit(`HTK`)[4].

Následující podkapitoly se věnují jednotlivým krokům, které byly provedeny při vývoji systému pro rozpoznávání emocí z řeči. U jednotlivých kroků je vždy popsána práce s vytvořenými moduly z balíčku pro rozpoznávání emocí.

7.1 Předzpracování dat

Nahrávky byly nejdříve předzpracovány. Byl sjednocen jejich formát a byly převedeny na příznaky MFCC. Formát byl sjednocen pomocí nástroje Fast Forward MPEG (FFmpeg)[5] přístupného z příkazové řádky. Nahrávky byly převedeny na vzorkovací frekvenci 16 kHz a byl zachován jeden zvukový kanál. Zvukové nahrávky byly převedeny pomocí příkazu na obrázku 7.1.

```
ffmpeg -i vstupni_soubor -ar vzorkovaci_frekvence -ac pocet_zvukovych_kanalu -y
vystupni_soubor
```

Obrázek 7.1: Příkaz FFmpeg

Dále byly z nahrávek extrahovány příznaky MFCC. K extrakci byla použita sada nástrojů HTK. Pro extrakci byl použit konfigurační soubor (viz obr. 7.2).

[1] https://www.python.org/
[2] https://pytorch.org
[3] https://github.com/danijel3/PyHTK
[4] https://htk.eng.cam.ac.uk
[5] https://ffmpeg.org

```
NONUMESCAPES=T
SOURCEFORMAT = WAVE
SOURCEKIND = WAVEFORM
NATURALREADORDER=T
NATURALWRITEORDER=T
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = F
SAVEWITHCRC = F
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
```

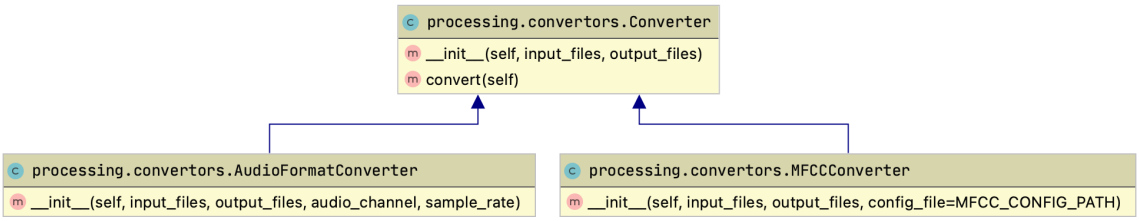
Obrázek 7.2: HTK konfigurační soubor

MFCC formát byl zvolen jako MFCC_0_D_A, jehož cílem bylo získat nulté, delta a akcelerační koeficienty. Dále lze vyčíst, že pro okénkování byla použita Hammingova okénkovací funkce a velikost okénka byla nastavena na 25 ms. Extrakce byla uskutečněna pomocí příkazu `HCOPY` na obrázku 7.3. Pro každý rámec nahrávky bylo získáno 39 příznaků.

```
HCopy -C configuracni_soubor vstupni_soubor vystupni_soubor
```

Obrázek 7.3: Příkaz HCopy

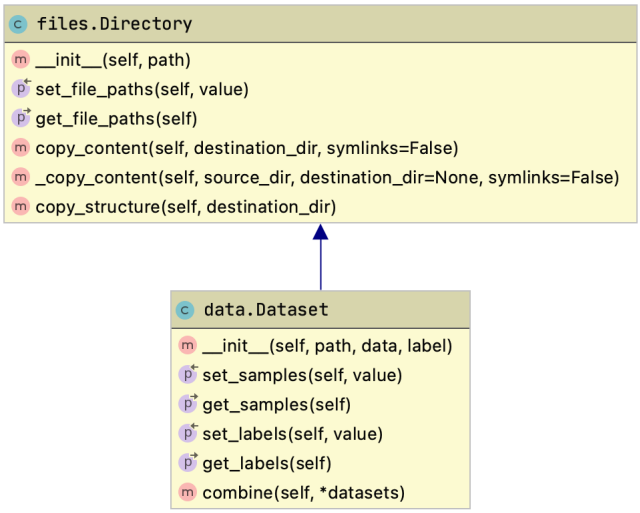
Třídy pro převod byly implementovány v modulu `convertors`. Byla vytvořena základní třída `Convertor`, z které konkrétní převaděče dědí. Pro sjednocení formátu byla vytvořena třída `AudioFormatConverter` a třída `MFCCConverter` pro převod na příznaky MFCC. Vztahy mezi převaděči jsou znázorněny na UML diagramu 7.4.



Obrázek 7.4: Třídy Converter

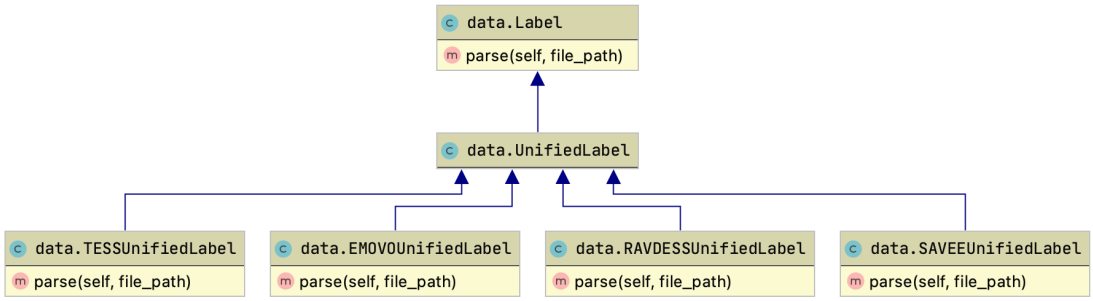
7.2 Příprava datových sad pro trénování

Pro načítání dat byla vytvořena třída `Dataset` v modulu `data`. Hlavními úkoly této třídy bylo načíst data a z anotací získat třídy emocí. Třída `Dataset` dědí od třídy `Directory` z modulu `files`, která umožňuje získat cesty k souborům ve složce. Vztah mezi třídami je znázorněn na diagramu 7.5.



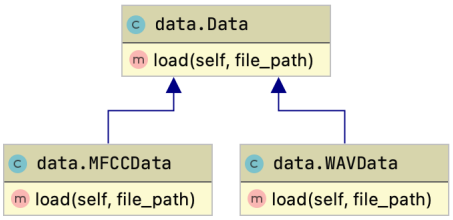
Obrázek 7.5: Třída Dataset

Třída `Dataset` má navíc atributy `samples` pro uložení načtených vzorků a `labels` pro uložení anotací pro vzorky. Při vytváření objektu třídy `Dataset` jsou předávány objekty třídy `Data` a `Label`. Objekt třídy `Data` načte na základě cesty k souboru vzorek a `Label` z anotace získá informace potřebné k vytvoření anotace pro vzorek. Jelikož jsou datové sady odlišně značeny, byly vytvořeny třídy dědící od třídy `UnifiedLabel` (viz obr. 7.6), které navíc anotace pro vzorky převedly na stejné značení.



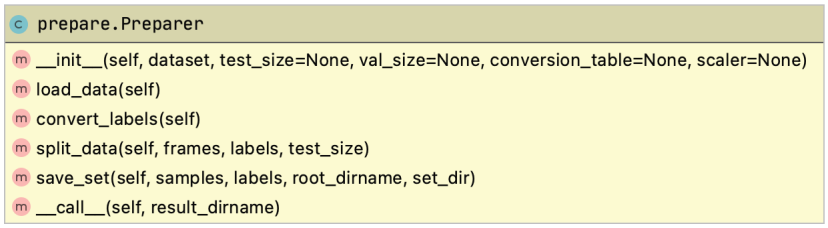
Obrázek 7.6: Třídy Label

Z třídy `Data` (viz obr. 7.7) dědí třída `MFCCData` umožňující načtení příznaků MFCC a `WAVData` zprostředkovávající načtení nahrávek ze souboru typu WAV. `MFCCData` používá třídu `HTKFile` z balíčku `PyHTK` pro načtení příznaků.



Obrázek 7.7: Třídy Data

Dále byla data rozdělena do trénovacích, validačních a testovacích sad. Pro rozdělení byla použita třída `Preparer` (viz obr. 7.8) z modulu `prepare`. Data byla nejdříve načtena pomocí objektů třídy `Dataset`. Umožňuje libovolný výběr datových sad a jejich sloučení. Data získaná z objektu `Dataset` mohla být rozdělena do trénovacího, testovacího a validačního setu. Rozdělení probíhalo rovnoměrně pomocí funkce `train_test_split` z modulu `sklearn.model_selection` a vzorky byly rozděleny po celých nahrávkách. Při rozdělování na sady byla zvolena možnost `stratify`, která rozdělila vzorky rovnoměrně do vzniklých sad podle anotací.



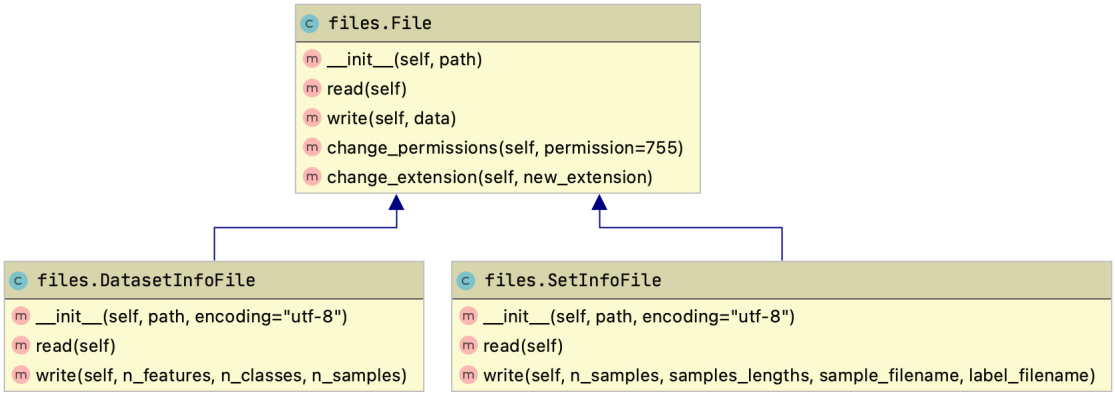
Obrázek 7.8: Třída Preparer

Jako poslední byly sady uloženy do složky. Každé sadě byla přiřazena vlastní složka, která obsahovala soubory labels.npy, samples.npy a info.txt. V souboru info.txt byly uloženy informace pro načítání datové sady. Struktura složky je znázorněna na obrázku 7.9.

```
int-7-re-95-05-05
├── info.txt
├── test
│   ├── info.txt
│   ├── labels.npy
│   └── samples.npy
├── train
│   ├── info.txt
│   ├── labels.npy
│   └── samples.npy
└── val
    ├── info.txt
    ├── labels.npy
    └── samples.npy
```

Obrázek 7.9: Souborová struktura uložené datové sady

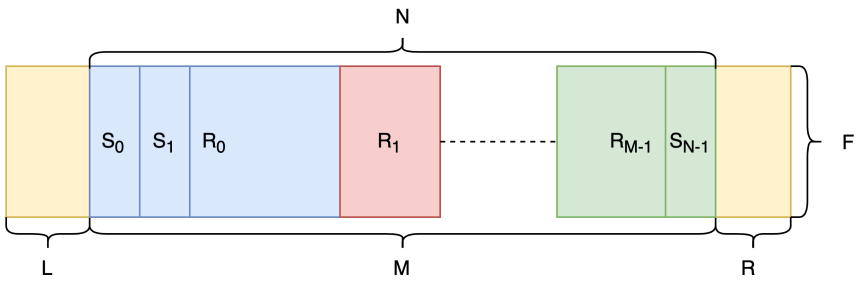
Pro ukládání a načítání informací ze souborů info.txt byly vytvořeny třídy `DatasetInfoFile` (viz obr. 7.10) a `SetInfoFile`. V souboru třídy `DatasetInfoFile` byly uloženy údaje o počtu příznaků, počtu tříd a počtu vzorků datové sady. Do `SetInfoFile` byly ukládány počty vzorků, délky vzorků a názvy souborů se vzorky a anotacemi. Data pro rozpoznávání byla uložena ve dvourozměrném poli `numpy`.



Obrázek 7.10: Třídy InfoFile

7.3 Načítání dat

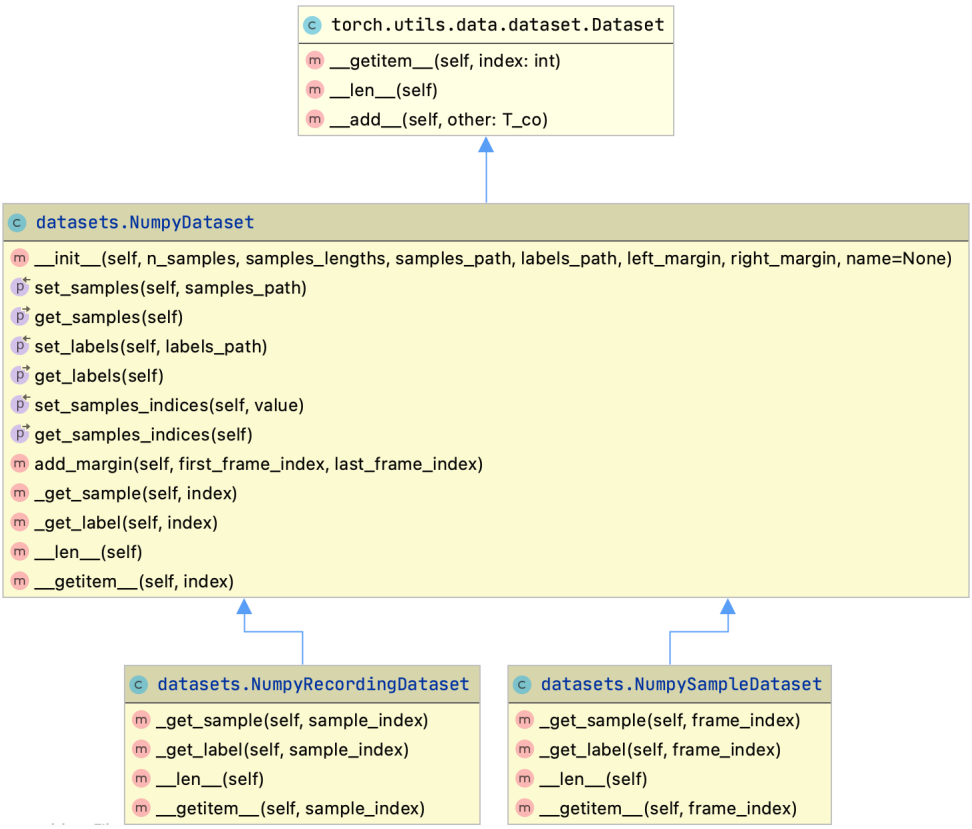
Pro načítání uložených datových sad byla vytvořena třída `NumpyDataset`, která dědí od třídy `Dataset` z balíčku `torch.utils.data`. Při vytvoření je objektu třídy přidělena cesta k datové sadě. Z textového souboru info.txt se přečtou informace potřebné k načtení dat. Mezi důležité parametry, které objekt `NumpyDataset` při svém vytvoření přebírá, patří velikosti pravého a levého okolí vzorku. Při načtení vzorků ze souboru samples.npy jsou ke vzorkům na začátku a na konci přiřazeny okraje, které mají velikosti pravého a levého okolí. Okraje jsou vytvořeny z kopií krajních vzorků a slouží pro zjednodušení výběrů vzorků. Vzorky jsou uloženy popořadě po jednotlivých nahrávkách. Objekt si ukládá indexy začátku a konce nahrávek. Načtené anotace z připravené datové sady odpovídají jednotlivým nahrávkám, proto jsou anotace roztaženy tak, aby odpovídaly délce vzorků nahrávky. Uložení vzorků v objektu `NumpyDataset` je znázorněno na diagramu 7.11.



Obrázek 7.11: Uspořádání dat v NumpyDatasetu

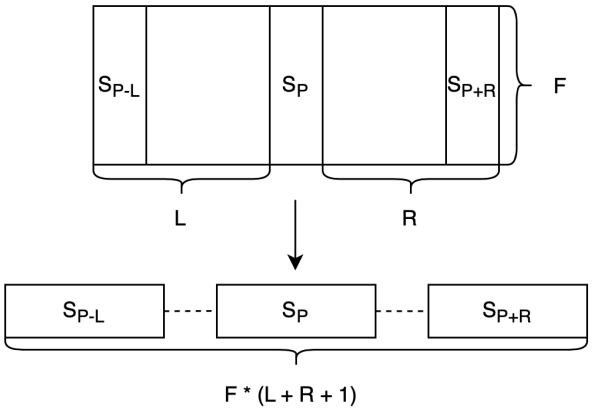
Písmenem S jsou označeny jednotlivé vzorky. R označuje jednotlivé nahrávky, které mohou mít různou délku. Písmeno N udává počet vzorků a M počet uložených nahrávek. L a R značí velikost pravého a levého okraje. Písmeno F odpovídá počtu příznaků pro jeden vzorek.

Z třídy `NumpyDataset` dědí třídy `NumpyFrameDataset` a `NumpySampleDataset` (viz obr. 7.12), které se liší v implementaci metod `__len__` a `__getitem__`. Metoda `__len__` při zavolání vrací délku datové sady, která u třídy `NumpyFrameDataset` odpovídá počtu vzorků a u třídy `NumpySampleDataset` odpovídá počtu nahrávek. Metoda `__getitem__` vrací jeden prvek z `Dataset`, u třídy NumpySampleDataset vrací metoda jeden vzorek a třídy `NumpyRecordingDataset` vrací jednu nahrávku. Třída `NumpyDataset` je tímto způsobem rozdělena. Důvodem pro rozdělení je, že při načítání dat pro validaci můžeme pomocí třídy `NumpyRecordingDataset` spočítat přesnost na celou nahrávku. Naopak při trénování je potřeba načítat vzorky po dávkách, které mají fixní délku na rozdíl od nahrávek, u kterých se délka mění. Vzorky se při trénování často míchají, a to je také důvod, proč je lepší mít třídu rozdělenou.



Obrázek 7.12: Třídy NumpyDataset

Ke vzorkům jsou při výběru přidávány vzorky z okolí pro větší zachycení kontextu nahrávky. Počet vzorků z okolí je načítán na základě velikosti pravého a levého okraje. Vybrané vzorky jsou před vrácením zploštěny. Proces výběru vzorku je znázorněn na obrázku 7.13.



Obrázek 7.13: Výběr vzorku

Vybraný vzorek má označení S_P a krajní vzorky mají označení S_{P-L} a S_{P+R}. Písmeno F označuje počet příznaků pro jeden vzorek. Značky L a P odpovídají délkám levého a pravého okraje. Výsledný zploštělý obrázek má délku odpovídající $F \cdot (L + R + 1)$.

7.4 Tvorba neuronové sítě

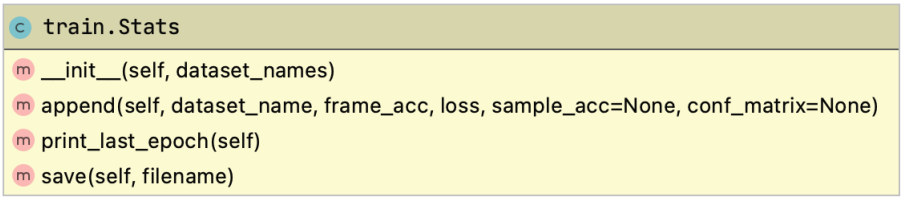
Pomocí frameworku `PyTorch` byl vytvořen model `FeedForwardNet` pomocí dědění z třídy `Module` z modulu `torch.nn`. V konstruktoru třídy `FeedForwardNet` byly poskládány jednotlivé vrsty neuronové sítě a byly uloženy do objektu `ModuleList` z modulu `torch.nn`. Dále byl pomocí metody `forward` implementován dopředný průchod.

7.4.1 Trénování

V modulu `train` byla vytvořena třída `Trainer` (viz obr. 7.14), jejíž úkolem je trénování modelu. Hlavní parametr předaný při vytváření instance třídy je `model`. Dále jsou předány trénovací, testovací a validační sady. Pro trénování modelu jsou důležité parametry `optimizer` k úpravě parametrů a kriteriální funkce pro získání ztráty. Parametr `device` umožňuje zvolit zařízení, na kterém bude trénování probíhat. Jako poslední je předán objekt typu `Stats` (viz obr. 7.15), do kterého se ukládají statistiky získané při trénování modelu.

```
train.Trainer
__init__(self, model, train_loader, val_dataset, test_datasets, optimizer, criterion, device, stats)
_train(self, train_loader)
_validate(self, dataset)
__call__(self, n_epochs=10)
```

Obrázek 7.14: Třída Trainer

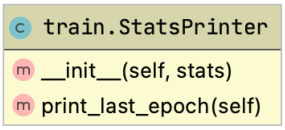


Obrázek 7.15: Třída Stats

Trénování je zahájeno zavoláním objektu třídy `Trainer`, kdy je předán počet trénovacích epoch. Každou epochu jsou do objektu třídy `Stats` ukládány statistiky trénování, mezi které patří přesnost na jeden vzorek, přesnost na jednu nahrávku, ztráta modelu a matice záměn. Zároveň jsou vybrané statistiky vypisovány v přehledném výpisu (viz obr. 7.16) do konzole. K výpisu je používán objekt třídy `StatsPrinter` (viz obr. 7.17).

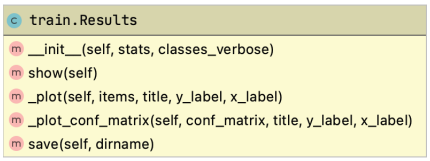
```
-------------------------------------------------------------------------------
  epoch      |      name      |      loss      |   acc_frames   |  acc_samples
-------------------------------------------------------------------------------
    1        | trénovací sada |     1.195      |     0.538      |
    1        | validační sada |     0.993      |     0.609      |     0.804
    1        | testovací sada |     1.009      |     0.611      |     0.798
-------------------------------------------------------------------------------
```

Obrázek 7.16: Přehledný výpis průběhu trénování



Obrázek 7.17: Třída StatsPrinter

Po ukončení trénování lze model uložit do souboru s koncovkou .pt. Statistiky trénování lze uložit do souboru CSV. Ze statistik lze pomocí třídy `Results` (viz obr. 7.18) získat grafy průběhu trénování. Pomocí funkce `plot` z knihovny `matplolib` jsou vykresleny grafy průběhu trénování pro přesnosti a ztrátu. Dále jsou vytvořeny pomocí funkce `heatmap` z knihovny `seaborn` grafy matice záměn. Grafy lze zobrazit a uložit jako obrázky typu PNG.



Obrázek 7.18: Třída Results

8 Experimentální vyhodnocení

Bylo provedeno několik experimentů pro vyladění hyperparametrů modelu s cílem zlepšení výsledků. Nejprve byl proveden výchozí experiment, který sloužil k porovnání s dalšími experimenty. Následně bylo provedeno několik pokusů se snahou vyladit hyperparametry modelu. Byla měněna především architektura sítě, počet a šířka vrstev. Upravovány byly také velikosti okolí vstupního vzorku. Trénování probíhalo během deseti epoch, jelikož bylo zjištěno u výchozího experimentu, že tento počet postačí pro získání představy o výkonu modelu. U všech modelů byl měněn pouze jeden parametr výchozího modelu a na konci byl na základě zjištěných výsledků vytvořen model finální.

8.1 Výběr datových sad

Hlavním kritériem při výběru byla dostupnost datové sady, proto byly vybrány sady bezplatné a dostupné pro vědecké účely. Byly vybrány především datové sady anglické: RAVDESS, TESS a SAVEE a jedna italská: EMOVO. Datové sady mají společných sedm emocí, mezi které patří emoce: hněv, strach, smutek, spokojenost, odpor, překvapení a neutrální stav. Jelikož měla datová sada RAVDESS osm emocí, byla emoce klidu přidána mezi neutrální stav. Informace o datových sadách byly shrnuty v tabulce 8.1.

Tabulka 8.1: Přehled vybraných datových sad

Název	**Jazyk**	**Počet tříd**	**Počet mluvčích**	**Pohlaví mluvčích**	**Počet promluv**	**Celková délka (hodiny)**
RAVDESS	angličtina	8	24	obě	1440	1.5
SAVEE	angličtina	7	4	muži	480	0.5
TESS	angličtina	7	2	ženy	2800	1.6
EMOVO	italština	7	6	obě	588	0.5

Celkem tedy byly získány čtyři datové sady o dvou jazycích se zastoupením sedmi emocí. Dohromady vzniklá datová sada má celkem přibližnou dobu trvání čtyři hodiny a obsahuje 5308 nahrávek.

Trénovací, validační a testovací sady byly vytvořeny sjednocením vybraných datových sad. Z každé vybrané datové sady byly rovnoměrně rozděleny vzorky podle tříd do sad v poměru 80 % nahrávek přišlo do trénovací sady, 10 % do sady testovací a 10 % do sady validační.

8.2 Výchozí model

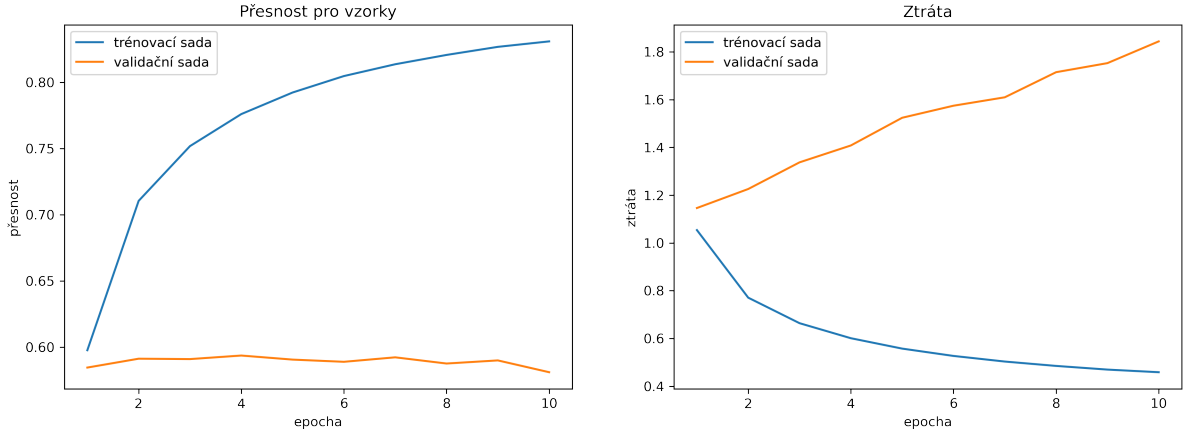
Jako první byl vytvořen výchozí model, který slouží k porovnání úspěšnosti mezi modely. Pro získání ztráty modelu byla jako kriteriální funkce zvolena křížová entropie. Byl zvolen algoritmus Adaptive Momentum (Adam) pro optimalizaci modelu a míra učení byla nastavena na hodnotu 0,001. Model se skládal ze tří skrytých lineárních vrstev o šířce 128 neuronů. Ve skryté vrstvě byla použita aktivační funkce ReLU. Model byl trénován po dobu deseti epoch s velikostí jedné dávky 256 vzorků. Velikost pravého a levého okolí byla rovna 25 vzorkům. Bylo klasifikováno sedm emocí: hněv, znechucení, strach, radost, překvapení, smutek a neutrální stav.

Z grafu pro přesnost pro vzorky (viz obr. 8.1) lze vyčíst, že byl velký rozdíl, přibližně 24 %, mezi dosaženou nejvyšší přesností na trénovací a validační sadě. Ztráta modelu postupně klesala na trénovací sadě, avšak lineárně stoupala sadě validační.

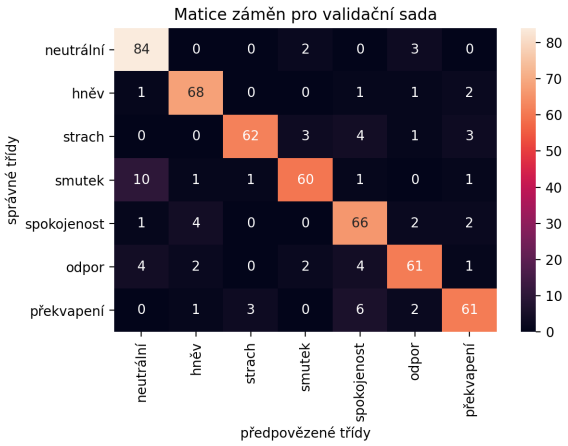
Model dosáhl přesnosti pro vzorky na validační sadě 58,8 % (viz tab. 8.2) a přesnosti pro nahrávku 88,5 %. Úspěch přesnosti pro nahrávky dokládá i rozložení předpovědí v matici záměn pro validační sadu (viz obr. 8.2), kde je většina předpovědí umístěna na hlavní diagonále. Zvyšující se ztráta na validační sadě naznačovala, že se model přeučoval na trénovací sadě. Přeučování nasvědčoval také konstantní průběh přesnosti pro vzorky na validační sadě.

Tabulka 8.2: Přehled nejlepších výsledků výchozího modelu

Trénovací sada	**Validační sada**	
Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
82,1 %	58,8 %	88,5 %



Obrázek 8.1: Přehled průběhu trénování výchozího modelu



Obrázek 8.2: Matice záměn pro validační sadu výchozího modelu

8.3 Změny počtu skrytých vrstev

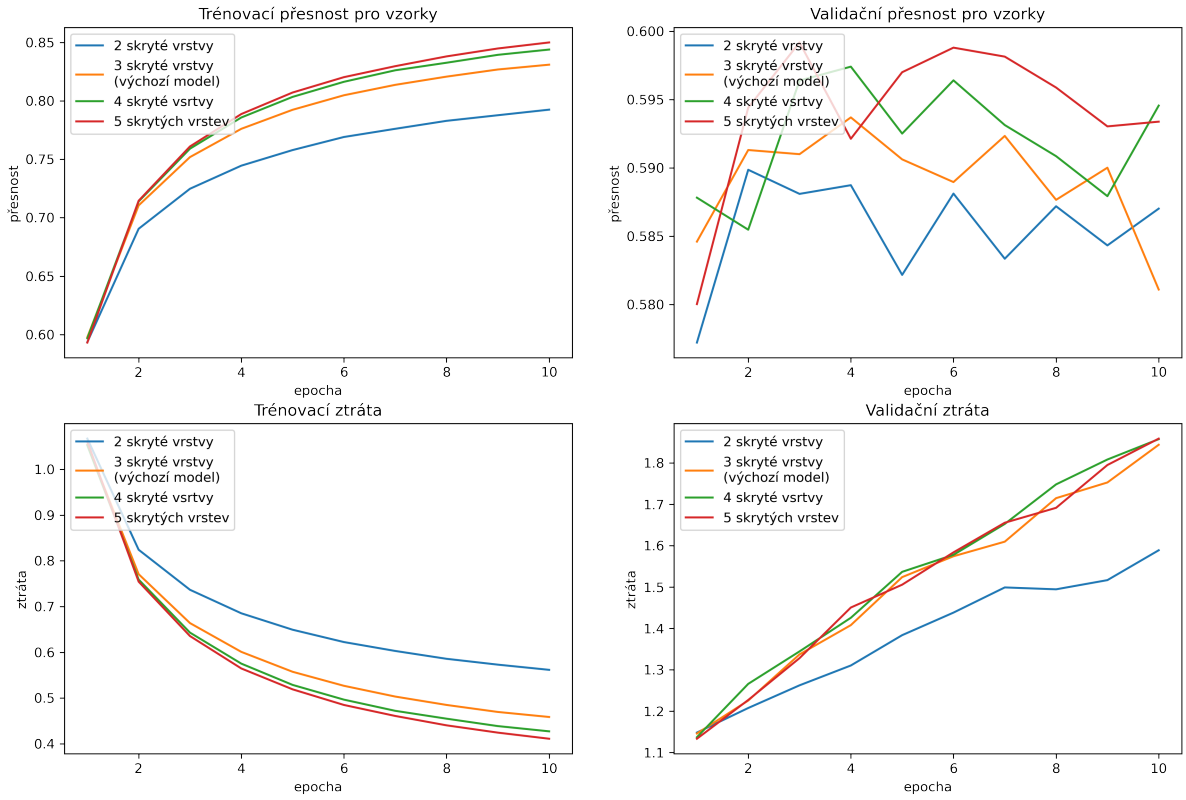
Byly provedeny tři experimenty, jejichž cílem bylo zjistit optimální počet skrytých vrstev modelu. Vyzkoušeny byly modely se dvěma, čtyřmi a pěti vrstvami. Tři vrstvy byly vynechány, jelikož je měl výchozí model.

Z grafů (viz obr. 8.3) pro přesnost pro vzorky lze usoudit, že s rostoucím počtem vrstev stoupala i přesnost, která byla podstatně nižší na validační sadě než na sadě trénovací. Přesnost u modelů se třemi a více vrstvami je velice podobná, na rozdíl od modelu se dvěma vrstvami, který se učil výrazně pomaleji. Ztráta na trénovací sadě po celou dobu trénování klesala a na validační sadě naopak lineárně rostla.

Nejlepších a zároveň srovnatelných výsledků dosáhly modely se čtyřmi a pěti skrytými vrstvami. Model se čtyřmi vrstvami dosáhl na validačních datech přesnosti pro vzorek 59,1 % a přesnosti pro nahrávku 89,3 % (viz tab. 8.3). Pro tvorbu finálního modulu byl zvolen počet čtyř skrytých vrstev, jelikož se menší model rychleji trénuje.

Tabulka 8.3: Přehled nejlepších výsledků pro různé počty skrytých vrstev

Počet skrytých vrstev	**Trénovací sada**	**Validační sada**	
	Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
2	78,3 %	58,7 %	89,8 %
3 (výchozí model)	82,1 %	58,8 %	88,5 %
4	83,3 %	59,1 %	89,3 %
5	84,5 %	59,3 %	89,3 %



Obrázek 8.3: Přehled průběhu trénování pro různé počty skrytých vrstev

8.4 Změny velikosti okolí vzorku

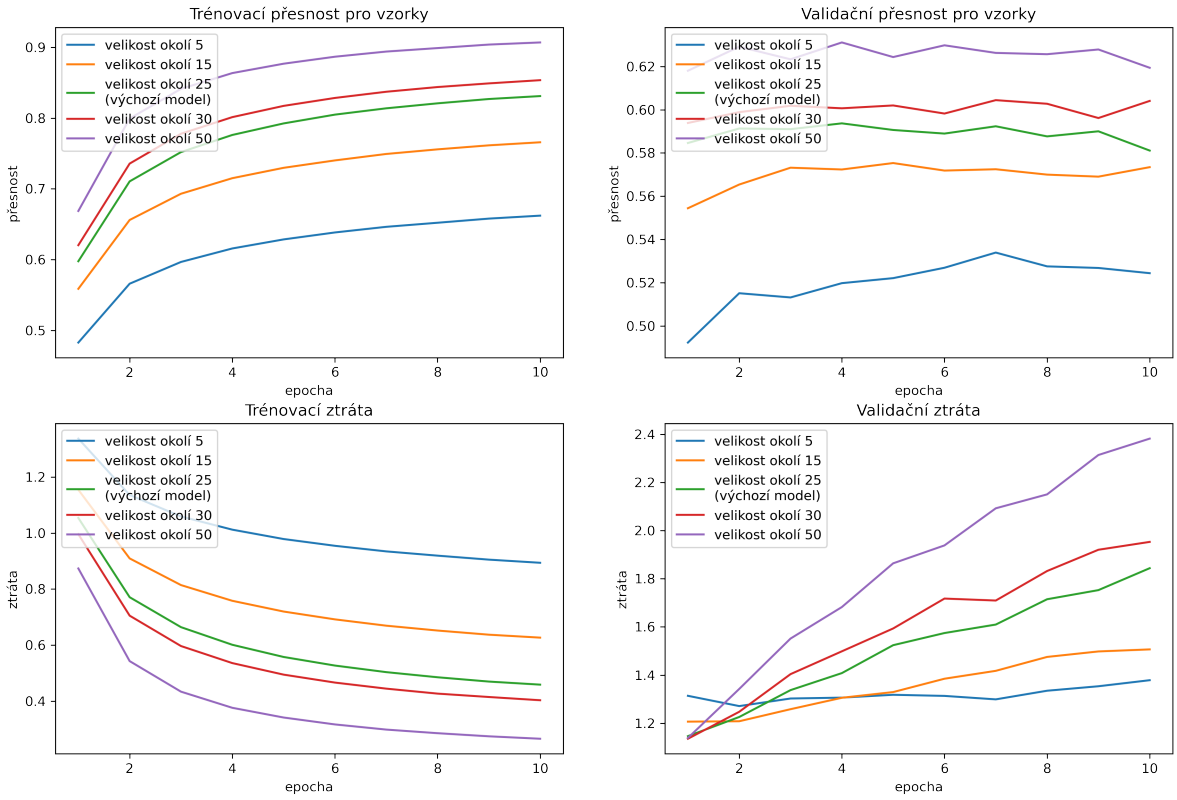
Velikost pravého a levého okolí vstupního vzorku bylo možné také upravovat. Celkem byly provedeny čtyři experimenty, při nichž velikost okolí pro pravou a levou stranu nabývala hodnot 5, 10, 30 a 50.

Z průběhu grafů (viz obr. 8.4) pro přesnost pro vzorky lze vyčíst, že přesnost rostla s větším počtem okolních vzorků, avšak byla výrazně nižší pro validační sadu. Ztráta na trénovací sadě klesala se zvyšujícím se okolím a na validační sadě naopak rostla.

Nejlepších výsledků dosáhl model s velikostí pravého a levého okolí 50 vzorků, které bylo zvoleno i pro finální model. Na validační sadě bylo docíleno přesnosti pro vzorky 61,9 % (viz tab. 8.4) a přesnosti pro nahrávku 86,8 %.

Tabulka 8.4: Přehled nejlepších výsledků pro různé velikosti vstupního okolí

Velikost vstupního okolí	**Trénovací sada**	**Validační sada**	
	Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
5	66,2 %	52,4 %	88,5 %
15	75,6 %	57 %	90 %
25 (výchozí model)	82,1 %	58,8 %	88,5 %
30	84,9 %	59,6 %	89,5 %
50	90,7 %	61,9 %	86,8 %



Obrázek 8.4: Přehled průběhu trénování pro různé velikosti okolí

8.5 Změny šířky skrytých vrstev

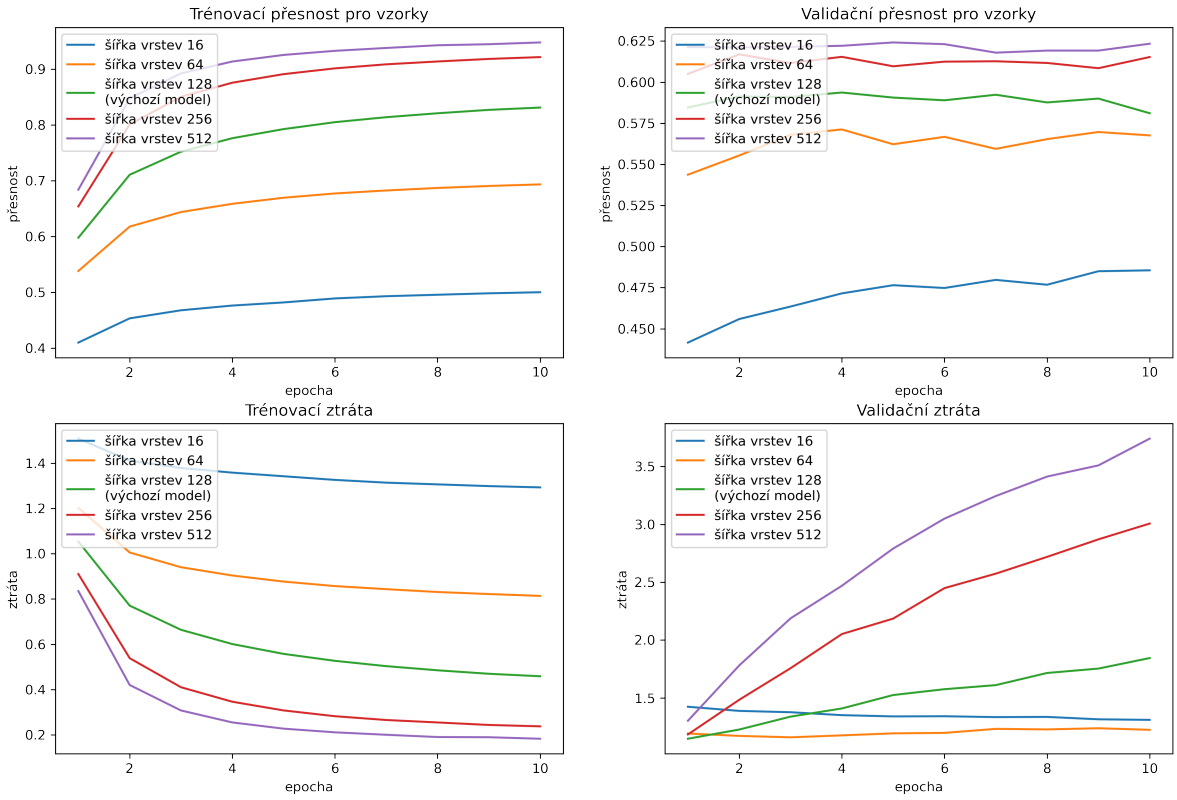
Šířka skrytých vrstev byla další hyperparameter, který bylo možné upravovat. Uskutečnily se čtyři experimenty, během kterých byla šířka skrytých vrstev 16, 64, 256 a 512 neuronů.

Z grafu pro přesnost pro vzorky (viz obr. 8.3) lze vyčíst, že přesnost stoupala se zvyšující se šířkou vrstev. Přesnost dosažená na validační sadě byla výrazně nižší než na sadě trénovací. U modelů se šířkou 256 a 512 neuronů nebyl téměř znatelný rozdíl mezi přesnostmi jako u ostatních modelů. Model se šířkou 16 neuronů se učil výrazně pomaleji než jiné modely. Ztráta modelu klesala se zvyšující se šířkou modelu na trénovací sadě. Na sadě validační nepatrně klesala u modelu s 16 neurony ve skryté vrstvě a byla téměř konstantní u modelu se šířkou 64 neuronů. U ostatních modelů ztráta rostla se zvyšujícím se okolím.

Nejlepších výsledků docílily modely se šířkou skrytých 256 a 512 neuronů. Model s šířkou 256 neuronů dosáhl na validační sadě přesnosti pro vzorky 61,5 % (viz tab. 8.5) a přesnosti pro nahrávky 89,6 %. Pro finální model byla vybrána hodnota 256 pro šířku skrytých vrstev, jelikož se menší model rychleji učí.

Tabulka 8.5: Přehled nejlepších výsledků pro různé šířky skrytých vrstev

Počet skrytých vrstev	**Trénovací sada**	**Validační sada**	
	Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
16	50 %	48,6 %	73,8 %
64	69 %	57 %	85,5 %
128 (výchozí model)	82,1 %	58,8 %	88,5 %
256	92,2 %	61,5 %	89,6
512	94,3 %	61,9 %	90,6



Obrázek 8.5: Přehled průběhu trénování pro různé šířky skrytých vrstev

8.6 Omezení počtu emocí

U některých oblastí použití je výhodnější rozpoznávat menší počet emocí nebo je to naopak žádoucí, a proto byly provedeny experimenty s omezováním počtu klasifikovaných emocí. Při experimentu se čtyřmi emocemi byly sjednoceny emoce hněvu a odporu na hněv, překvapení a spokojenost na spokojenost, smutek a strach na smutek. Se třemi emocemi byly rozlišovány pouze neutrální stav, negativní a pozitivní emoce. Mezi pozitivní emoce patřila spokojenost s překvapením a mezi negativní emoce hněv, odpor, smutek a strach.

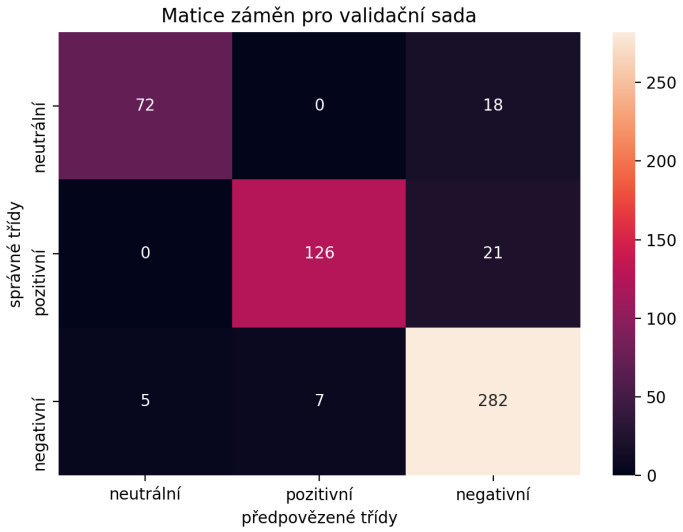
Z grafů (viz obr. 8.8 pro přesnost pro vzorek lze vyčíst, že při trénování přesnost rostla a při validaci byla téměř konstantní. Rozdíl mezi nejvyšší přesností na trénovacích a validačních sadách byl znatelný. U všech modelů se pohyboval kolem 20 %. S ubývajícím počtem emocí přesnost rostla. Ztráta na trénovacích datech po celou dobu klesala, avšak na validačních datech lineárně stoupala. S přibývajícím počtem emocí byla ztráta nižší.

Při klasifikaci do tří tříd bylo dosaženo přesnosti pro vzorek 71,3 % (viz tab. 8.6) a přesnosti pro nahrávku 90,4 %. Při klasifikaci do čtyř tříd bylo dosaženo stejné přesnosti pro nahrávku 90,4 %, ale nižší přesnosti pro vzorek 65,3 %. Na základě vzrůstající ztráty na validační sadě měly všechny modely tendenci se přeučovat na sadě trénovací.

Tabulka 8.6: Přehled nejlepších výsledků pro různý počet emocí

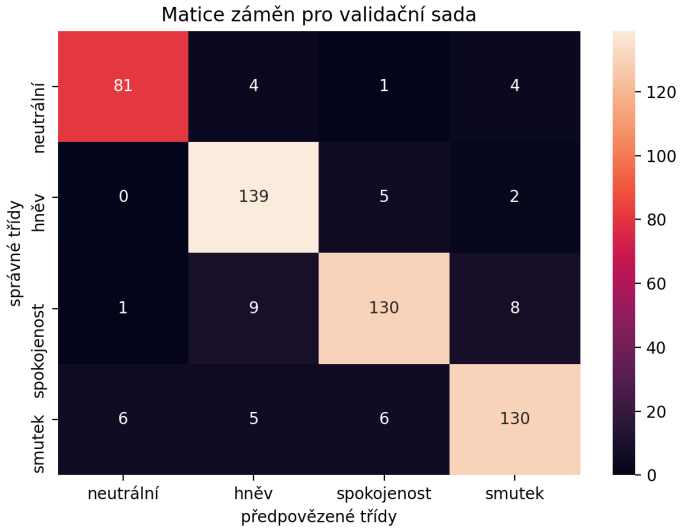
Počet emocí	**Trénovací sada**	**Validační sada**	
	Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
3	91,1 %	71,3 %	90,4 %
4	88 %	65,3 %	90,4 %
7 (výchozí model)	82,1 %	58,8 %	88,5 %

Počet nahrávek pro jednotlivé třídy je mnohem vyrovnanější při klasifikaci do čtyř tříd oproti klasifikaci do tří tříd. Při sloučení tříd pro klasifikaci do tří tříd byla největší část, 294 nahrávek, umístěna do negativní třídy (viz obr. 8.6), o něco méně, 147 nahrávek, do pozitivní třídy a nejméně, 90 nahrávek, do třídy neutrální.

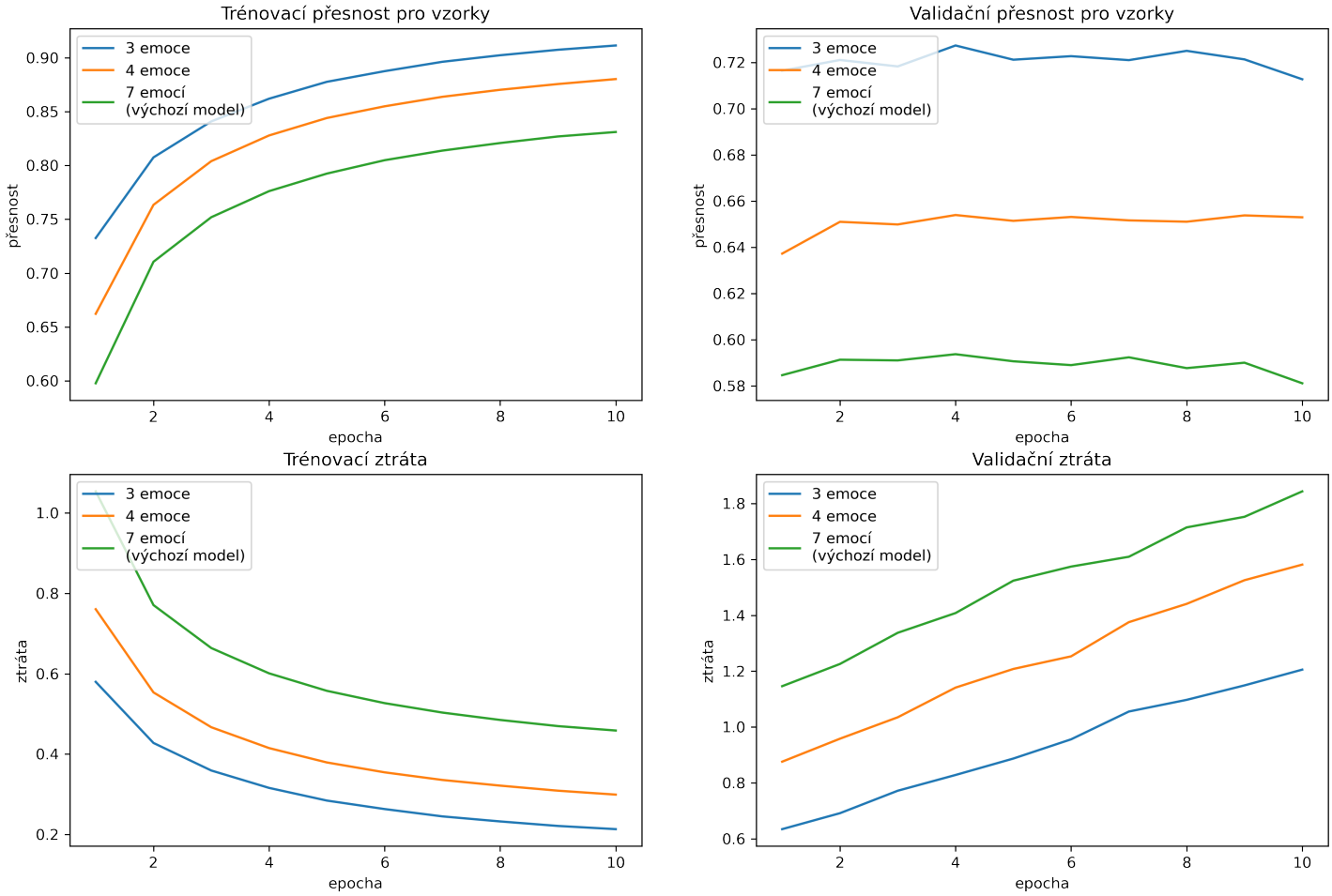


Obrázek 8.6: Matice záměn pro validační sadu při klasifikaci tří emocí

Při klasifikaci do čtyřech tříd byly nahrávky rozděleny téměř rovnoměrně do tříd hněvu, spokojenosti a smutku (viz obr. 8.7). Počet nahrávek se pohyboval v rozmezí 141–148 nahrávek. Méně, 90 nahrávek, bylo umístěno v neutrální třídě.



Obrázek 8.7: Matice záměn pro validační sadu při klasifikaci čtyř emocí



Obrázek 8.8: Přehled průběhu trénování pro různý počet emocí

8.7 Použití regularizace

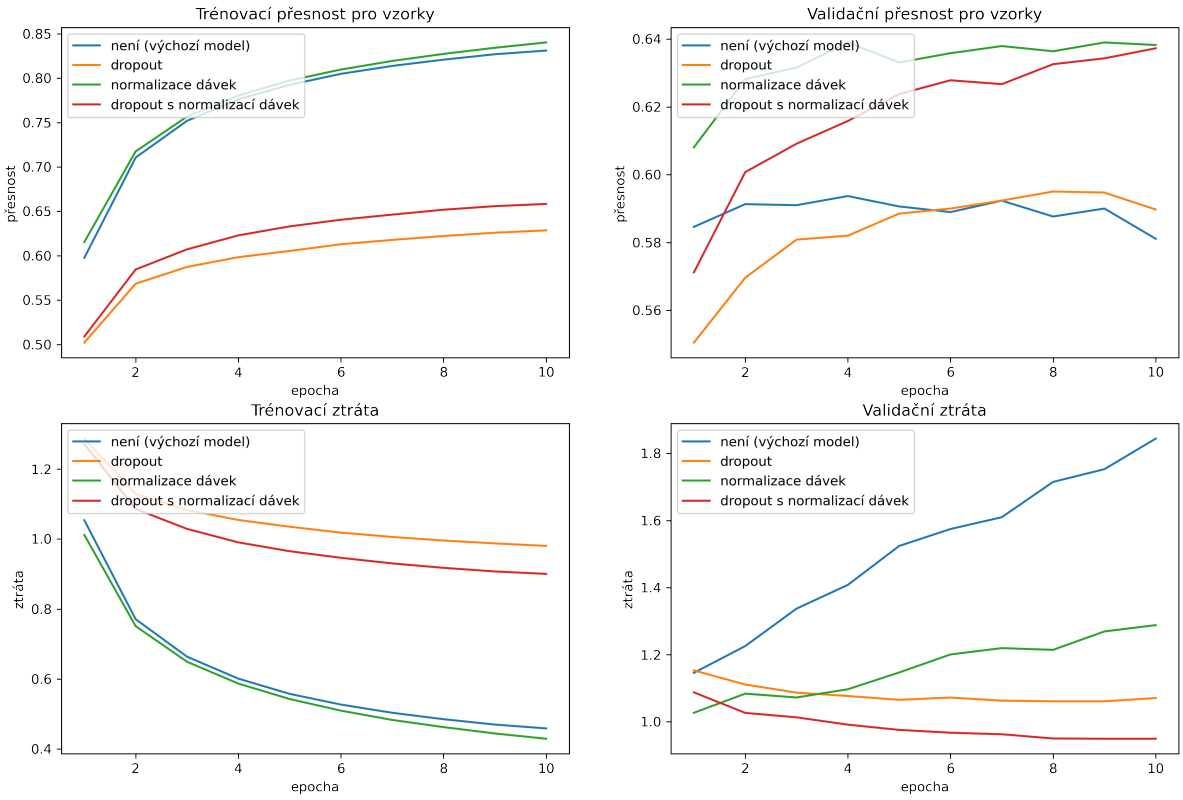
Jelikož u předchozích experimentů docházelo přeučování na validační sadě, bylo vyzkoušeno několik způsobů regularizace s cílem omezení přeučování. U prvního experimentu byl přidán za aktivační funkce ve skryté vrstvě dropout s pravděpodobností 25 %, že se výstup vynuluje. U dalšího modelu byly přidány za aktivační funkce normalizace dávek. U posledního modelu byly techniky regularizace propojeny a za aktivační funkci byla nejprve umístěna normalizace dávky následovaná dropout.

Z grafu (viz obr. 8.9) přesnosti pro vzorek trénovací sady lze vyčíst, že rozdíl mezi průběhem přesnosti výchozího modelu bez normalizace byl téměř stejný. Modely s dropout dosáhly podobných přesností, ale model s normalizací dosáhl přesnosti o 4 % vyšší. Z grafu pro přesnost pro vzorky validační sady lze vyčíst, že model bez regularizace a model s dropout dosáhly podobných nejlepších výsledků. Průběh přesnosti na validační sadě modelu bez regularizace byl téměř konstantní, na rozdíl od modelu s dropout, který měl tendenci růst. Modely s normalizací dávek dosáhly lepších a téměř totožných nejlepších přesností. Model s normalizací dávek a dropout se učil pomaleji než model bez dropout. Z grafu ztráty pro trénovací sadu lze usoudit, že ztráta pro modely s dropout klesala výrazně pomaleji než u modelů bez dropout. Z průběhu ztrát na validační sadě lze vyčíst, že u modelů s dropout ztráta klesala a u modelů bez dropout naopak rostla.

Nejlepší model byl model s normalizací dávek a dropout, který docílil přesnosti pro vzorky 63,4 % (viz tab. 8.7) a přesnosti pro nahrávku 86,6 %. Výchozí model bez regularizace a model s normalizací dávek dosáhly vyšší přesnosti pro nahrávky, ale měli vyšší tendenci se přeučovat na trénovací sadě.

Tabulka 8.7: Přehled nejlepších výsledků při použití regularizace

regularizace	**Trénovací sada**	**Validační sada**	
	Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
není (výchozí)	82,1 %	58,8 %	88,5 %
dropout	62,6 %	59,5 %	84,6 %
normalizace dávek	83,4 %	63,9 %	91,3 %
dropout s normalizací dávek	65,6 %	63,4 %	86,6 %



Obrázek 8.9: Přehled průběhu trénování při použití regularizace

8.8 Finální modely

Na základě předchozích experimentů byl vytvořen finální model, který měl čtyři skryté vrstvy o šířce 256 neuronů. Ve skrytých vrstvách byla použita regularizace v podobě normalizace dávek a dropout, které byly umístěny za aktivační funkci. Velikost pravého a levého okolí vstupního vzorku byla 50 vzorků. Jako aktivační funkce ve skrytých vrstvách byla použita funkce ReLU. Optimalizační algoritmus byl Adam s mírou učení 0.001. Velikost dávek při trénování byla 256 vzorků. Křížová entropie byla použita jako kriteriální funkce. Model byl natrénován pro klasifikaci do tří, čtyř a sedmi tříd a výsledky byly porovnány s výchozím modelem. Trénování probíhalo po dobu deseti epoch. Modely byly vyhodnoceny i na sadě testovací. Na obrázku 8.10 jsou informace o trénování finálního modelu z logovacího souboru.

Z grafů pro přesnost pro vzorek (viz obr. 8.11) lze usoudit, že s přibývajícím počtem emocí rostla i přesnost. Rozdíl mezi přesností trénovací a validační sady byl přibližně 10 %. Přesnost pro vzorky finálního modelu pro klasifikaci sedmi emocí a výchozího modelu byla téměř stejná na trénovací sadě, nicméně na sadě validační dosáhl finální model mnohem lepších výsledků. Rozdíl nejlepších přesností byl přibližně 13 %. Rozdíl mezi přesnostmi všech finálních modelů byl zhruba 4 % se snižujícím se počtem emocí. Z grafů ztráty modelů na validační sadě lze vyčíst, že ztráta u finálních modelů téměř vždy klesala. U modelu pro rozpoznávání tří emocí měla ztráta tendenci mírného růstu od čtvrté epochy trénování.

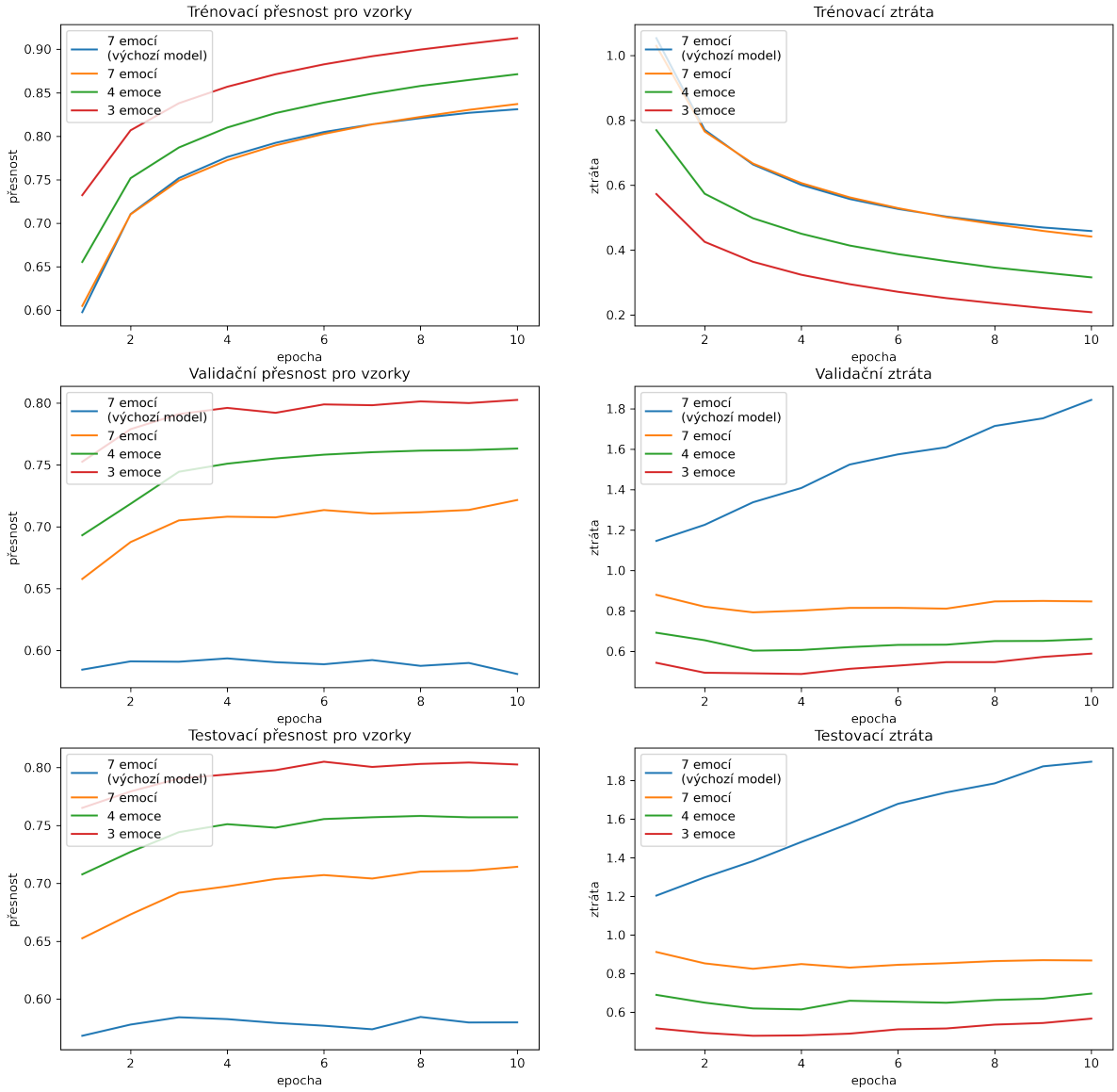
Model pro klasifikaci do sedmi tříd dosáhl na testovací sadě nejlepší přesnosti pro vzorky 71,4 % (viz tab. 8.8) a přesnosti pro nahrávku 92,3 %. Model pro klasifikaci do čtyř tříd dosáhl přesnosti pro vzorky 75,7 % a přesnosti pro nahrávky 93,6 %. Model pro klasifikaci do tří tříd docílil nejlepší přesnosti pro vzorek 80,1 % a přesnosti pro nahrávku 94,4 %. Trénování modelu trvalo přibližně 38 minut na čtyřjádrovém procesoru Intel Core i7.

Tabulka 8.8: Přehled nejlepších výsledků finálních modelů

Počet tříd	**Trénovací sada**	**Validační sada**		**Testovací sada**	
	Přesnost pro vzorek	**Přesnost pro vzorek**	**Přesnost pro nahrávku**	**Přesnost pro vzorek**	**Přesnost pro nahrávku**
3	89,2 %	79,8 %	92,7 %	80,1 %	94,4 %
4	87,1 %	76,3 %	92,8 %	75,7 %	93,6 %
7	83,7 %	72,2 %	92,7 %	71,4 %	92,3 %
7 (výchozí model)	82,1 %	58,8 %	88,5 %	58,5 %	87,9 %

```
Optimizer:
Adam (
Parameter Group 0
    amsgrad: False
    betas: (0.9, 0.999)
    eps: 1e-08
    lr: 0.001
    weight_decay: 0
)
Criterion:
CrossEntropyLoss()
Model:
FeedForwardNet(
  (layers): ModuleList(
    (0): Linear(in_features=3939, out_features=256, bias=True)
    (1): ReLU()
    (2): Linear(in_features=256, out_features=256, bias=True)
    (3): ReLU()
    (4): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Dropout(p=0.25, inplace=False)
    (6): Linear(in_features=256, out_features=256, bias=True)
    (7): ReLU()
    (8): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (9): Dropout(p=0.25, inplace=False)
    (10): Linear(in_features=256, out_features=256, bias=True)
    (11): ReLU()
    (12): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (13): Dropout(p=0.25, inplace=False)
    (14): Linear(in_features=256, out_features=256, bias=True)
    (15): ReLU()
    (16): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (17): Dropout(p=0.25, inplace=False)
    (18): Linear(in_features=256, out_features=7, bias=True)
  )
)
N epochs:  10
Batch size:  256
Left margin:  50
Right margin:  50
```

Obrázek 8.10: Výpis informací o trénování finálního modelu



Obrázek 8.11: Přehled průběhu trénování finálních modelů

9 Závěr

Tato práce se zabývala rozpoznáváním emocí z řeči. V teoretické části byly nejprve uvedeny emoce a hlavní dva modely pro jejich dělení. Byl popsán postup pro vytváření systému pro rozpoznávání emocí z řeči. V současném poznání byly uvedeny používané příznaky a datové sady.

Byly popsány tři hlavní typy datových sad a poskytnut přehled bezplatných datových sad. Podrobněji byly vylíčeny datové sady použité v praktické části práce. Dále byly popsány základní techniky pro zpracování řeči. Bylo uvedeno základní dělení příznaků pro rozpoznávání emocí z řeči a důkladněji popsán postup pro extrakci příznaků MFCC. Další kapitola byla věnována vybraným základům neuronových sítí, kde byly popsány koncepty spojené s učením a tvorbou modelu.

V jazyku Python byl vytvořen balíček pro rozpoznávání emocí. Skládal se z modulů pro převod dat, práci s daty, tvorbu klasifikátoru, práci se soubory, tvorbu datových sad, přípravu datových sad a trénování modelu. Balíček především umožňoval snadné vytvoření, uložení a načtení datových sad pro trénování. Zároveň zprostředkovával jednoduché trénování modelu a uložení výsledků trénování pro zpětnou analýzu.

Z volně dostupných datových sad byly vybrány tři anglické datové sady (RAVDESS, SAVEE, TESS) a jedna italská (EMOVO). Byl sjednocen formát nahrávek a byly získány příznaky MFCC. Pro učení modelu byla data z jednotlivých datových sad rovnoměrně rozdělena do trénovací, validační a testovací sady v poměru 80 %, 10 % a 10 %.

Jako klasifikátor byla zvolena neuronová síť typu MLP. Model byl vytvořen a natrénován pomocí frameworku PyTorch. Prvotně byl vytvořen výchozí model. Následně bylo provedeno a zdokumentováno několik experimentů s cílem vylepšit hyperparametry výchozího modelu. U většiny experimentů měl model tendenci se přeučovat na trénovacích datech, a proto se osvědčilo umístit do skryté vrstvy regularizační vrstvy, které zabraňovaly přeučování.

Na základě proběhlých experimentů byly vytvořeny tři finální modely pro různý počet emocí. Model pro klasifikaci tří emocí dosáhl na testovací sadě přesnosti pro vzorky 80,1 % a přesnosti pro nahrávky 94,4 %. Při klasifikaci do čtyř tříd dosáhl model na testovací sadě přesnosti pro vzorky 75,7 % a 93,6 % přesnosti pro nahrávky. Klasifikátor do sedmi tříd dosáhl na testovací sadě přesnosti pro vzorky 71,4 % a přesnosti pro nahrávky 92,3 %.

Pro zlepšení výsledků klasifikace by nejvíce pomohlo rozšíření datové sady pro trénování. Pomocí regularizačních vrstev lze do jisté míry přeučování modelu zabránit, ale míru přesnosti modelu omezuje nízký počet a různorodost dat. Dalším

způsobem, jak vylepšit výsledky klasifikace, je vyzkoušet jiné typy neuronových sítí. V úvahu na příklad přicházejí neuronové sítě typu CNN nebo RNN. V obou případech lze téměř bez změn přidat funkcionalitu do balíčku pro rozpoznávání emocí.

Použitá literatura

[1] AKÇAY, Mehmet Berkehan a Kaya OGUZ. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Commun.* 2020, roč. 116, s. 56–76. Dostupné z DOI: 10.1016/j.specom.2019.12.001.

[2] KONAR, Amit a Aruna CHAKRABORTY. *Emotion recognition: a pattern analysis approach*. John Wiley & Sons, Inc., 2015.

[3] VENKATARAMANAN, Kannan a Haresh Rengaraj RAJAMOHAN. Emotion Recognition from Speech. *CoRR*. 2019, roč. abs/1912.10458. Dostupné z arXiv: 1912.10458.

[4] ENGBERG, Inger S., Anya Varnich HANSEN, Ove ANDERSEN a Paul DALSGAARD. Design, recording and verification of a danish emotional speech database. In: KOKKINAKIS, George, FAKOTAKIS, Nikos a DERMATAS, Evangelos (ed.). *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*. ISCA, 1997. Dostupné také z: http://www.isca-speech.org/archive/eurospeech%5C_1997/e97%5C_1695.html.

[5] BURKHARDT, Felix, Astrid PAESCHKE, M. ROLFES, Walter SENDLMEIER a Benjamin WEISS. A database of German emotional speech. In: 2005, sv. 5, s. 1517–1520.

[6] HANSEN, John H. L. a Sahar E. BOU-GHAZALE. Getting started with SUSAS: a speech under simulated and actual stress database. In: KOKKINAKIS, George, FAKOTAKIS, Nikos a DERMATAS, Evangelos (ed.). *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*. ISCA, 1997. Dostupné také z: http://www.isca-speech.org/archive/eurospeech%5C_1997/e97%5C_1743.html.

[7] LIVINGSTONE, Steven R. a Frank A. RUSSO. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE*. 2018, roč. 13, č. 5, e0196391. Dostupné z DOI: 10.1371/journal.pone.0196391.

[8] MARTIN, Olivier, Irene KOTSIA, Benoît MACQ a Ioannis PITAS. The eNTERFACE'05 Audio-Visual Emotion Database. In: BARGA, Roger S. a ZHOU, Xiaofang (ed.). *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE 2006, 3-7 April 2006, Atlanta, GA, USA*. IEEE Computer Society, 2006, s. 8. Dostupné z DOI: 10.1109/ICDEW.2006.145.

[9] JACKSON, Philip a Sanaul HAQ. *Surrey Audio-Visual Expressed Emotion (SAVEE) Database*. [N.d.]. Dostupné také z: http://kahlan.eps.surrey.ac.uk/savee/Database.html.

[10] GAROFOLO, J., Lori LAMEL, W. FISHER, Jonathan FISCUS, D. PALLETT, N. DAHLGREN a V. ZUE. TIMIT Acoustic-phonetic Continuous Speech Corpus. *Linguistic Data Consortium*. 1992.

[11] PICHORA-FULLER, M. Kathleen a Kate DUPUIS. *Toronto emotional speech set (TESS)*. Scholars Portal Dataverse, 2020. DRAFT VERSION. Dostupné z DOI: 10.5683/SP2/E8H2MF.

[12] *Toronto emotional speech set (TESS)*. [N.d.]. Dostupné také z: https://tspace.library.utoronto.ca/handle/1807/24487.

[13] COSTANTINI, Giovanni, Iacopo IADEROLA, Andrea PAOLONI a Massimiliano TODISCO. EMOVO Corpus: an Italian Emotional Speech Database. In: CALZOLARI, Nicoletta, CHOUKRI, Khalid, DECLERCK, Thierry, LOFTSSON, Hrafn, MAEGAARD, Bente, MARIANI, Joseph, MORENO, Asunción, ODIJK, Jan a PIPERIDIS, Stelios (ed.). *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*. European Language Resources Association (ELRA), 2014, s. 3501–3504. Dostupné také z: http://www.lrec-conf.org/proceedings/lrec2014/summaries/591.html.

[14] GOODFELLOW, I., Y. BENGIO a A. COURVILLE. *Deep Learning*. MIT Press, 2016. Adaptive Computation and Machine Learning series. ISBN 9780262337373. Dostupné také z: https://books.google.cz/books?id=omivDQAAQBAJ.

[15] TIWARI, Vibha. MFCC and its applications in speaker recognition. *Int. J. Emerg. Technol.* 2010, roč. 1.

[16] BISHOP, Christopher M. *Pattern recognition and machine learning, 5th Edition*. Springer, 2007. Information science and statistics. ISBN 9780387310732. Dostupné také z: https://www.worldcat.org/oclc/71008143.

[17] MATĚJŮ, Lukáš, David GRIOL, Zoraida CALLEJAS, José Manuel MOLINA a Araceli SANCHIS. An empirical assessment of deep learning approaches to task-oriented dialog management. *Neurocomputing*. 2021, roč. 439, s. 327–339. ISSN 0925-2312. Dostupné z DOI: https://doi.org/10.1016/j.neucom.2020.01.126.

[18] BURKOV, A. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. ISBN 9781999579517. Dostupné také z: https://books.google.cz/books?id=0jbxwQEACAAJ.

[19] GÉRON, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated, 2019. ISBN 9781492032649. Dostupné také z: https://books.google.cz/books?id=OCS1twEACAAJ.

[20] DAGLI, Rishit. *The Art of Hyperparameter Tuning in Deep Neural Nets by Example*. Towards Data Science, 2021. Dostupné také z: https://towardsdatascience.com/the-art-of-hyperparameter-tuning-in-deep-neural-nets-by-example-685cb5429a38.

[21] BROWNLEE, Jason. *A Gentle Introduction to Cross-Entropy for Machine Learning*. 2020. Dostupné také z: https://machinelearningmastery.com/cross-entropy-for-machine-learning/.

[22] BROWNLEE, Jason. *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. 2020. Dostupné také z: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[23] BROWNLEE, Jason. *Softmax Activation Function with Python*. 2020. Dostupné také z: https://machinelearningmastery.com/softmax-activation-function-with-python/.

[24] FUMO, David. *Types of Machine Learning Algorithms You Should Know*. Towards Data Science, 2017. Dostupné také z: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861.

[25] LEONEL, Jorge. *Hyperparameters in Machine/Deep Learning*. Medium, 2019. Dostupné také z: https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981.

[26] HUI, Jonathan. *Speech Recognition-Feature Extraction MFCC & PLP*. Medium, 2019. Dostupné také z: https://jonathan-hui.medium.com/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9.