TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Hendrik Kivi, 176606 IAPM

VÕÕRKEELE AKTSENDI TUVASTAMINE JA LOKALISEERIMINE KÕNEST

Magistritöö

Juhendaja: Tanel Alumäe, PhD

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Hendrik Kivi

07.01.2020

Võõrkeele aktsendi tuvastamine ja lokaliseerimine kõnest

Annotatsioon

Magistritöö eesmärgiks on treenida närvivõrgumudel, mis suudab eesti keelsetest kõnenäidetest tuvastada võõrkeele aktsenti. Mudelist leitakse tunnused, mida närvivõrk õppis aktsendi klassifitseerimiseks. Töös lokaliseeritakse kõnenäidetest häälikud, mille juures tunnused esinevad ja proovitakse leida tunnustele keelelist seletust. Tunnuste saamiseks võetakse mudeli konvolutsioonilise kihi väljundiks olevad tunnuskaardid ning kombineeritakse need närvivõrgu kaaludega.

Töös näidatakse ära, et sellise meetodi abil on võimalik närvivõrku interpreteerida. Tuvastatakse üle 20 tunnuse, millele pakutakse välja keeleline interpretatsioon. Samuti tuuakse välja eri aktsendiklasside jaoks tunnused, mida närvivõrk pidas aktsendi tuvastamisel kõige olulistemaks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 6 peatükki, 11 joonist, 3 tabelit.

Identification and Localization of Foreign Accent in Speech

Abstract

The growing success of neural networks has resulted in huge development in building accent and speech identification systems. As a result of this, the systems have become more difficult to understand. In a way, neural networks are comparable to black boxes. Although using neural networks leads to excellent results, there is little known about the process of neural network in finding the result. Because of that, the interest in interpreting neural networks has grown considerably in recent years. Doing that would help to better understand the features that neural network is learning and therefore make them even more efficient. Hence, the current thesis is trying to interpret a neural network trained for accent identification, and to figure out the features that the neural network uses for identification.

The purpose of the master's thesis is to train a neural network model, which is able to identify foreign accent from estonian speech. Secondly, the model is used to extract features that neural network used to identify accent. Also, it is attempted to localize the phonemes of each feature and to identify the features that are most relevant for each accent group.

The data from Estonian Foreign Accent Corpus is used for training and testing the neural network model. The accent corpus contains estonian speech examples of speakers with different native languages. The trained models are used to extract the phonemes or combinations of phonemes that were used for identification of accent. To achieve this, the feature maps extracted from the last convolutional layer of the neural network are combined with the weights of neural network. The resulting regions of speech are aligned with the specific phonemes and the speech examples are analysed to determine, which linguistic features neural network uses for identifying the accent.

In the thesis, it is demonatrated that this method is able to successfully interpret neural network. More than 20 different features are identified and for each feature linguistic interpretation is proposed. Furthermore, for each accent the features, that the neural network considered most important for identifying the accent, were found.

This thesis is written in Estonian and is 37 pages long, including 6 chapters, 11 figures and 3 tables.

Lühendite ja mõistete sõnastik

GAP	*Global average pooling*. Väärtusi keskmistav ahenduskiht
MFCC	*Mel-frequency Cepstral Coefficients*. Mel-sageduse kepstri koefitsiendid
MSE	*Mean squared error*. Ruutkeskmine viga
ReLU	*Rectified linear unit*. Mittenegatiivne lineaarfunktsioon

Sisukord

Joonised

Tabelid

Peatükk 1

Sissejuhatus

Aktsendituvastus seisneb võõrkeele rääkija emakeele tuvastamises tema kõne põhjal. Automaatne aktsendituvastus on probleem, mida tänaseks on küllaltki põhjalikult uuritud ja selle jaoks on kasutatud mitmeid andmepõhiseid meetodeid nagu i-vektorid [8], rekurrentsed närvivõrgud ja konvolutsioonilised närvivõrgud [9]. Nende meetodite puuduseks on, et nad ei tagasta konkreetseid omadusi, mille järgi aktsenti klassifitseeritakse.

Käesolevas töös proovitakse seda tühimikku täita ja luuakse mudel aktsendituvastuseks, mis suudaks lisaks aktsendituvastamisele tagastada omadused, mille järgi mudel aktsenti tuvastab, ehk leida häälikukombinatsioonid, mille järgi aktsenti klassifitseeritakse.

1.1 Probleemipüstitus

Närvivõrkude esiletõus on toonud kaasa suure arengu nii aktsendi- ja kõnetuvastuse kui ka muude süsteemide arendamisel. Probleemiks on, et selle arengu tulemusena on need süsteemid muutunud vähem arusaadavamaks. Närvivõrgud on justkui mustad kastid, nende abil on võimalik saavutada väga häid tulemusi, aga keegi ei tea täpselt, kuidas nad selliste tulemusteni jõuavad. Seetõttu on viimasel ajal kasvanud huvi närvivõrkude interpreteerimise vastu, mis võimaldaks paremini mõista tunnuseid, mida närvivõrk õpib, ja seeläbi muuta neid veelgi efektiivsemaks. Selles töös proovitakse interpreteerida aktsendituvastuse jaoks treenitud närvivõrku ja aru saada, milliste tunnuste põhjal närvivõrk aktsenti tuvastab.

Magistritöö eesmärgiks on:

- luua süsteem, mis suudab tuvastada kõneleja aktsenti ja mida saab kasutada lokaliseerimiseks,
- leida tunnused, mida närvivõrk kasutab aktsendituvastuseks
- lokaliseerida häälikukombinatsioonid, mille vastu kindla aktsendiga kõnelejad eksivad.

Töö tulemuseks on programm, mis suudab helifailide põhjal automaatselt tuvastada kõneleja aktsenti. Klassifitseerimise täpsus ei ole niivõrd oluline, aga programm peab lokaliseerima kõnest konkreetsed kohad, kus aktsent esineb. Teiseks saadakse töö tulemusena nimekiri tunnustest, mida närvivõrk õppinud on. Saadud tunnused sisaldavad häälikut, millele tunnus vastab ja keelelist seletust. Lisaks leitakse olulisemad tunnused erinevate aktsendiklasside jaoks.

Püstitatud eesmärkide saavutamiseks luuakse närvivõrgu mudel, mis tuvastab kõne põhjal rääkija aktsendi. Mudeli loomisel katsetatakse erinevaid võrgu arhitektuure ja parameetreid, eesmärgiga konstrueerida mudel, mis suudaks võimalikult täpselt aktsenti klassifitseerida.

Mudeli treenimiseks ja testimiseks kasutatakse Eesti aktsendikorpuse andmeid. Aktsendikorpus sisaldab erinevate keeletaustadega inimeste eesti keelseid kõnenäiteid. Treenitud mudeleid kasutatakse, et leida häälikud või kõneregioonid, mille järgi mudel aktsendi tuvastab. Selleks eraldatakse mudeli konvolutsioonilise kihi väljundist saadud tunnuskaardid ja kasutatakse närvivõrgu kaale, et tuvastada piirkonnad, mida närvivõrk peab klassifitseerimise jaoks oluliseks. Leitud piirkonnad viiakse kokku konkreetsete häälikutega ja analüüsitakse kõnenäiteid, et tuvastada, milliseid keelelisi nähtuseid närvivõrk kasutab aktsendi tuvastamiseks.

1.2 Taust

Tänapäeval on küllaltki laialt levinud rakendused, mis kasutavad automaatset kõnetuvastust. Need rakendused on tihti disainitud ühe keele jaoks ja teise emakeelega kasutajate jaoks toimib kõnetuvastus märgatavalt halvemini. Seetõttu on kõnetuvastuse valdkonnas

pööratud palju tähelepanu automaatse aktsendituvastuse arendamisele, mis aitaks kõnetuvastuse süsteeme täpsemaks muuta. Samuti võimaldab aktsendituvastus luua kõneleja profiili, sealhulgas leida tema päritolu, mida kasutatakse näiteks suunatud turundamiseks.

Automaatset aksendituvastust on proovitud implementeerida erinevate meetoditega. Suures osas on proovitud taaskasutada tehnikaid, mida on juba kasutatud kõnetuvastuse jaoks. Ühe lähenemisena on kasutatud erinevaid statistilisi ja masinõppe meetodeid. Näiteks Deshpande et al. kasutavad oma töös [10], kus nad üritavad eristada Ameerika ja India inglise keelt, Format' sagedustunnustel põhinevat Gaussi segumudelit. Ghesquiere et al. [11] kasutasid samuti Format' sagedustunnuseid Flaami aktsentide tuvastamiseks. Tang ja Ghorbini [12] kasutasid Markovi peitmudelit ja tugivektormasinat ning võrdlesid nende sooritust aktsendi klassifitseerimisel.

Viimasel ajal on aktsendituvastuseks kasutatud rohkem i-vektori põhiseid meetodeid [13] [14] [15] [8]. I-vektorite puhul on tegu kõnetunnuste esitusega [16], mida saab kasutada klassifitseerimiseks ja identifitseerimiseks [8]. Najafian et al. klassifitseerisid kõnelejad 14 erinevasse Briti saartelt leitud aktsendiklassi. I-vektori põhise süsteemi abil saavutati täpsuseks 76,76%.

Automaatseks aktsendituvastuseks on kasutatud ka tehisnärvivõrke. Parimaid tulemusi on saavutatud sügavate närvivõrkudega [17] [18] ja rekurrentsete närvivõrkudega [19]. [20] [21] Jiao et al. [22] pakkusid välja süsteemi, mis kombineerib sügavaid närvivõrke ja rekurrentseid närvivõrke.

Eraldi probleem on lokaliseerimine, mida aktsendituvastuse kontekstis on küllaltki vähe uuritud. Üheks näiteks on Brown [6], kes uurib, kuidas erinevad häälikud mõjutavad tekstipõhiste süsteemide aktsendi klassifitseerimise edukust. Kui eelnevalt väljatoodud süsteemid olid tekstist sõltumatud, mis tähendab, et nad ei vaja klassifitseerimiseks kõne transkriptsiooni, siis Brown kasutab oma töös tekstipõhist aktsendituvastust. Ta leidis, et häälikud küll mõjutavad klassifitseerimise edukust, aga samas ei leidnud ta, et oleks konkreetseid häälikuid, mis järjepidevalt aitaks aktsenti klassifitseerida.

Närvivõrkude kasutamiset lokaliseerimiseks on katsetatud ka pildituvastuse valdkonnas. Zhou et al. [7] kasutasid piltide klassifitseerimiseks mõeldud konvolutsioonilisi närvivõrke ja genereerisid konvolutsioonilistest kihtidest saadud tunnuskaartide põhjal klasside aktivatsioonikaardid, mis võimaldas tuvastada olulised piirkonnad piltide peal. Käes-

olevas magistritöös kasutatakse sarnast meetodit oluliste piirkondade lokaliseerimiseks kõnest.

1.3 Ülevaade tööst

Töö teises peatükis antakse ülevaade töö teoreetilisest taustast, sealhulgas närvivõrkudest ja sellega seonduvatest detailidest, aktsendi olemusest ja tunnustest ning kõne spektraalsetest tunnustest. Kolmandas peatükis kirjeldatakse lähemalt kahte olemasolevat tehtud tööd, mis on käesoleva tööga seotud. Neljas peatükk sisaldab meetodi implementatsiooni kirjeldust. Seal kirjeldatakse lähemalt töös kasutatud andmeid, närvivõrgu mudeli arhitektuuri ja treenimist ning tunnuste leidmist. Viiendas peatükis tuuakse välja ning analüüsitakse töö tulemusi.

Peatükk 2

Teoreetiline taust

2.1 Tehisnärvivõrk

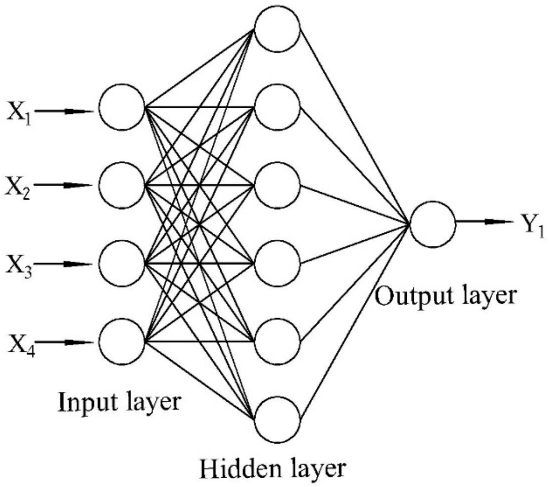
Tehisnärvivõrgud on arvutussüsteemid, mille tegemisel on inspiratsiooni võetud bioloogilisest närvivõrgust. Närvivõrk koosneb hulgast neuronitest, mis on omavahel ühendatud. Iga neuron saab sisse sisendväärtuse, mis korrutatakse neuroni kaaluga ja nii saadakse väljundväärtus. Selleks, et närvivõrk õpiks lastakse andmeid mitu korda närvivõrgust läbi. Ühte sellist tsüklit nimetatakse epohhiks. Iga epohhi käigus uuendatakse neuronite kaale, nii et need sobituks paremini andmetega. See võimaldabki närvivõrgul õppida.

Närvivõrk koosneb kihtidest, kus igas kihis on teatud hulk neuroneid. Esimene kiht on sisendkiht, viimane on väljundkiht ja nende vahel võib olla veel hulk kihte, mida kutsutakse peidetud kihtideks. Iga kiht saab sisendi eelmiselt kihilt ja annab väljundi edasi järgmisele kihile sisendiks.

Neuronite puhul on tegemist töötlemisüksustega, millel on kolm olulist komponenti: kaalud, vabaliikmed ja aktivatsioonifunktsioon. Neuroni sees toimuvat arvutust võib väljendada valemiga 2.1.

$$z = f(b + xw) = f(b + \sum_{i=1}^{n} x_i w_i), \tag{2.1}$$

Neuron saab sisse sisendväärtuse, ta korrutab selle kaaluga ja liidab vabaliikme väärtuse. Saadud tulemusele rakendatakse aktivatsioonifunktsiooni ning funktsiooni tulemus on



Joonis 2.1: Närvivõrgu struktuur [1]

neuroni väljundiks. Närvivõrgu treenimisel antakse treeningandmed närvivõrgu sisendkihile. Neuronid rakendavad eelnevalt mainitud protsessi andmetele ja tulemuseks saadavad andmed liiguvad mööda võrku edasi, kuni lõpuks jõutakse väljundkihini. Seejärel arvutab kahjufunktsioon vahe oodatud ja tegeliku väljundi vahel. Optimiseerija muudab kaalude ja vabaliikmete väärtusi, eesmärgiga minimiseerida seda vahet.

Selleks, et leida sobivad kaalude ja vabaliikmete väärtust tuleb närvivõrku treenida mitu epohhi. Üldiselt treenitakse nii kaua, kuni treeningkulu enam ei muutu. Tulemusena saadakse närvivõrk, mida saab kasutada uutest andmetest väljundi leidmiseks.

2.2 Aktivatsioonifunktsioon

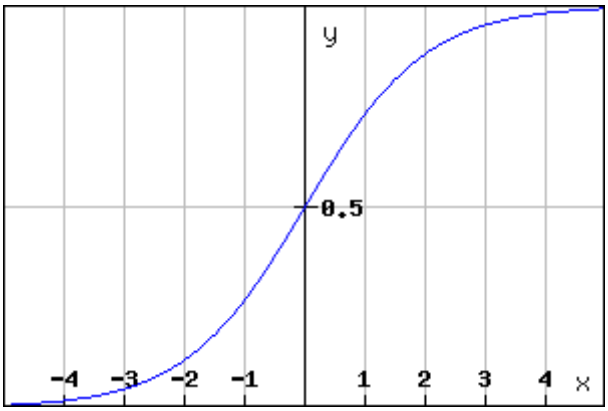
Närvivõrgu neuronites kasutatakse aktivatsioonifunktsiooni, mis arvutab sisendite põhjal neuroni väljundi. Aktivatsioonifunktsioon tagab, et väljund jääks soovitud vahemikku, näiteks 0 ja 1 või -1 ja 1 vahele. Üldjuhul kasutatakse aktivatsioonifunktsioonidena mittelineaarset funktsiooni, mis võimaldab mudelil õppida mittelineaarseid seoseid.

Üks aktivatsiooonifunktsioon, mida kasutatakse küllaltki palju, on sigmoidfunktsioon.

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

Nagu jooniselt 2.2 näha, jääb sigmoidfunktsiooni väljund 0 ja 1 vahele, mis on eriti sobiv,

kui eesmärgiks on ennustada millegi tõenäosust. Funktsioon on ka diferentseeruv ehk iga funktsiooni punkti vahel on kalle. See on oluline gradiendi leidmiseks tagasilevi käigus. Sigmoidfunktsiooni puuduseks on haihtuva gradiendi probleem. See tuleneb sellest, et sigmoidfunktsiooni tuletis on väike. Tagasilevi käigus korrutatakse tuletist mitu korda ja kui närvivõrgul on palju kihte, siis jääb gradient esimeste kihtideni jõudes väga väikeseks. See omakorda tähendab, et esimeste kihtide parameetrite uuendamine on ebaefektiivne ja mudel muutub ebatäpseks.

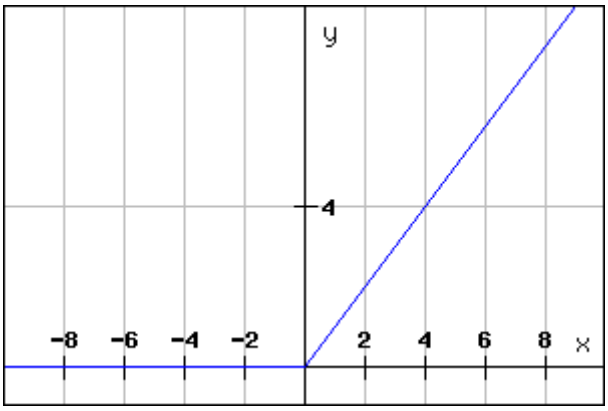


Joonis 2.2: Sigmoidfunktsioon

Haihtuva gradiendi probleemi aitab lahendada mittenegatiivne lineaarfunktsioon (ReLU, *Rectified linear unit*). ReLU väljastab sisendi väärtuse, kui neuroni sisend on positiivne, vastasel juhul on ReLU väljund 0.

$$f(x) = max(x, 0) \tag{2.3}$$

Kuigi ReLU on mittelineaarne, siis käitub ta suures osas samamoodi nagu lineaarne funktsioon, tänu millele ei teki haihtuva gradiendi probleemi. Teine ReLU positiivne omadus on, et ta on hõre. Sigmoidfunktsiooni puhul aktiveeruvad kõik neuronid, aga ReLU puhul on paljude neuronite väljund 0. See tähendab, et osa neuroneid võib väljundi arvutamisest välja jätta, mis muudab arvutuse efektiivsemaks. ReLU negatiivne külg on, et väljundite 0-ks muutumise tõttu, ei vasta need neuronid enam muutustele. Selle tagajärjel muutub suur osa närvivõrgust passiivseks. Selle lahendamiseks on loodud variatsioone RelU-st, nagu lekkiv ReLU, kus negatiivsete sisendite korral on väljundiks 0,01x.



Joonis 2.3: ReLU funktsioon

2.3 Kahjufunktsioon

Selleks, et iseloomustada kui hästi närvivõrgu mudel töötab etteantud andmete põhjal, kasutatakse kahjufunktsiooni. Mida rohkem mudeli ennustused erinevad soovitud tulemustest, seda suurem on kahjufunktsiooni väärtus. Optimeerimisalgoritmid kasutavad kahjufunktsiooni mudeli parandamiseks, proovides vähendada kahjufunktsiooni väärtust.

Kahjufunktsiooni valik sõltub sellest, millist probleemi lahendatakse. Üldjoontes jagatakse kahjufunktsioonid kahte kategooriasse, sõltuvalt sellest, kas tegemist on regressiooni või klassifitseerimise ülesandega. Regressiooni ülesannete puhul proovitakse ennustada pideva väärtusega numbrit. Klassifikatsiooni ülesannete puhul tahetakse leida sobivat väljundit lõpliku hulga kategooriliste väärtuste hulgast.

Üheks regressiooniprobleemi jaoks sobiva kahjufunktsiooni näiteks on ruutkeskmine viga (MSE), mis leiab ennustatuste ja tegelike väärtuste vahe ruudu ja arvutab nene keskmise kogu andmestiku kohta.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.4}$$

Klassifitseerimise ülesannete jaoks on üks sobivamaid kahjufunktsioone ristentroopiakahju.

$$L = -\frac{1}{N}\sum_{i=1}^{n}(Y log(\hat{Y}_i)) \tag{2.5}$$

Lühidalt öeldes võetakse ristentroopia valemi järgi õigele klassile vastavate ennustuste logaritm ja liidetakse need kokku. Mida suurem on õige klassi ennustuse tõenäosus, seda väiksem on funktsiooni kahju.

2.4 Tagasilevi ja gradientlaskumine

Info liikumist mööda närvivõrku edasi nimetatakse pärileviks. Tagasilevi on meetod, mis võimaldab edasilevi tulemusena saadud kuluväärtusest liikuda tagasi ja leida närvivõrgu jaoks sobivad kaalud. Teiste sõnadega arvutatakse kahjufunktsiooni gradient [23]. Gradient iseloomustab, kui palju muutub kahjufunktsiooni väärtus parameetri muutuse suhtes.

Tagasilevi peamine eelis on, et ta teeb vahekihtides asuvate neuronite kaalude leidmise oluliselt lihtsamaks. Kuna peidetud kihtide soovitav väljund ei ole teada, siis ei saa petsõlmede jaoks defineerida kahjufunktsiooni. See sõltub sõlmele eelneva ja järgneva kihi parameetritest.

Gradientlaskumine on optimeerimisalgoritm, mis muudab närvivõrgu parameetreid, nii et kahjufunktsioon oleks minimaalne. Parameetrite väärtuste muutmiseks kasutab algoritm tagasilevi kaudu arvutatud gradienti. Igal treeningiteratsioonil arvutatakse uued parameetrite väärtused järgneva valemi alusel:

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial E}{\partial \theta} \tag{2.6}$$

,

kus θ^t tähistab parameetrite väärtusi iteratsioonis t, $\frac{\partial E}{\partial \theta}$ on kahjufunktsiooni osatuletis parameetrite suhtes ja α on õpisamm.

Õpisamm on suurus, mis määrab, kui suurte sammudega algoritm liigub kahjufunktsiooni miinimumi suunas. Oluline on valida õpisamm, mis poleks liiga suur ega liiga väike. Kui õpisamm on liiga suur, siis on oht, et parameetreid muudetakse korraga liiga palju ja minnakse miinimumist mööda. Teisest küljest, kui õpisamm on liiga väike, siis võtab miinimumini jõudmine kaua aega ja närvivõrk õpib aeglaselt. Kui kahjufunktsioon enam ei muutu, siis öeldakse, et algoritm on konvergeerunud.

Gradientlaskumise algoritmist on erinevaid variante sõltuvalt sellest, kui palju andmeid kasutatakse ühes treeningtsüklis.

Tavaline gradientlaskumine leiab kahju iga andmestikus oleva treeningnäite kohta, aga uuendab mudeli parameetreid alles pärast kõigi treeningnäidete hindamist. Kuna mudelit uuendatakse harva, siis on algoritm arvutuslikult efektiivne ja gradient on stabiilne. Selle puuduseks on, et stabiilse gradiendi tõttu võib algoritm liiga vara konvergeeruda.

Stohhastiline gradientlaskumine uuendab mudelit iga treeningnäite järel. Selle meetodi eeliseks on mudeli pidev uuendamine, mis võib teatud probleemide korral viia kiirema õppimiseni. Teisest küljest on pidev uuendamine arvutuslikult ressursse nõudev. Samuti võivad mudeli parameetrid selle tõttu palju varieeruda ja algoritmil on raske leida miinimumi.

Kolmas variant on miniplokk-gradientlaskumine, mis kombineerib eelnevate meetodite põhimõtted. See jagab treeningandmestiku plokkideks ja uuendab mudelit iga ploki jaoks. Kuna see meetod pakub tasakaalu kahe eelneva meetodi vahel, olles ühelt poolt piisavalt efektiivne ja tesisest küljest piisavalt stabiilne, siis on see kõige enam kasutatav gradientlaskumise implementatsioon.

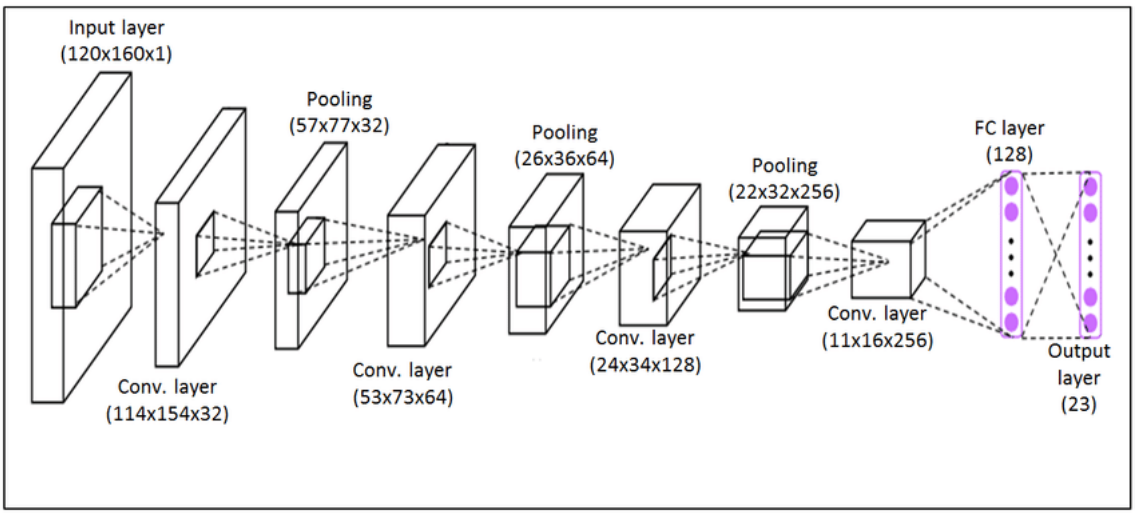
2.5 Konvolutsioonilised närvivõrgud

Konvolutsioonilised närvivõrgud on süvanärvivõrgud, mida kasutatakse peamiselt piltide analüüsimiseks. Konvolutsioonilise närvivõrgu eeliseks on, et ta suudab tuvastada klassidele iseloomulikke tunnuseid ilma inimese poolse juhendamiseta.

Konvolutsioonilise närvivõrgu arhitektuur näeb välja umbes nii nagu kujutatud joonisel 2.4. Sisendile rakendatakse mitmeid konvolutsioonilisi ja ahenduskihte, millele järgneb täissidusad kihid.

Konvolutsioonilise närvivõrgu kõige tähtsam element on konvolutsiooniline kiht. Konvolutsioon on matemaatiline operatsioon, mis seab kahele hulgale informatsioonile vastavusse kolmanda. Konvolutsiooniliste närvivõrkude puhul saadakse sisendandmetele konvolutsiooni filtrit rakendades tunnuste kaart.

Oletame, et meie sisendandmed suurusega 3 x 3 ja filter suurusega 2 x 2. Konvolutsioo-



Joonis 2.4: Konvolutsioonilise närvivõrgu arhitektuur [2]

ni käigus liigutatakse filtrit üle sisendi. Igas asukohas korrutatakse kummagi maatriksi vastavad elemendid omavahel ja liidetakse tulemused. Saadud arv on tulemuseks saadava tunnuskaarti üheks elemendiks. Tunnuskaarti suurus on sama, mis filtri suurus.

Sisendile rakendatakse mitu konvolutsiooni, millest iga üks kasutab erinevat filtrit. Seega saadakse iga konvolutsiooni tulemusena erinev tunnuskaart. Konvolutsioonikihi lõplik väljund saadakse kõikide tunnuskaartide kokku panemisel.

Üldjuhul sisaldab närvivõrk ka mittelineaarsust. Konvolutsiooni tulemustele rakendatakse mittelineaarset aktivatsioonifunktsiooni ja tunnuskaardid koosnevad aktsivatsioonifunktsiooni väljunditest.

Lisaks filtrite suurusele ja arvule on konvolutsioonil sellised parameetrid nagu samm ja ääris. Samm määratleb, mitme koha võrra filtrit igal sammul liigutatakse. Väikse sammu puhul on kattuvus väljade vahel suur. See-eest kui samm on suurem kui 1, siis jäetakse osa sisendi välju vahele ja tulemuseks saadav tunnuskaart on väiksem kui konvolutsiooni suurus.

Mõnikord on kasulik hoida väljund sama suurena kui sisend. Selleks saab muuta äärise parameetrit, mis täpsustab, mitu rida nulle lisatakse sisendi äärde. Nullide asemel võib lisada ka sisendi äärmiste väljade väärtusi.

Konvolutsioonilisele kihile järgneb tavaliselt ahenduskiht, mis vähendab sisendi dimensionaalsust, muutes iga tunnuskaarti suuruse väiksemaks. Selle tulemusena väheneb para-

meetrite hulk, mis vähendab treeningu aega ja aitab vältida ülesobitust. Kõige enamlevinud ahendusmeetod on *max pooling*, mis jagab sisendi osadeks ja väljastab iga osa jaoks maksimaalse väärtuse. Ahenduse idee seisneb selles, et tunnuse täpne asukoht ei ole nii oluline kui selle asukoht teiste tunnuste suhtes.

Pärast konvolutsioonilisi ja ahenduskihte järgneb täissidus kiht, mis toimib samamoodi nagu mitmekihiline pertseptron. Täissidusale kihile antakse sisendiks ühedimensiooniline vektor.

2.6 Regularisatsioon

Treenides närvivõrku, mille eesmärgiks on üldistada ehk toimida edukalt tundmatute andmete korral, on oht üle sobitada. Ülesobitamine tähendab, et närvivõrk õpib tundma tunnuseid, mis annavad hea tulemuse treeningandmete klassifitseerimiseks, aga seda ainult läbi juhuse. Teisisõnu need tunnused on omased ainult treeningandmetele ja ei kandu üle teistsugustele andmetele.

Ülesobitamise vältimiseks kasutatakse regularisatsiooni, mis normaliseerib mudelit ja väldib liiga keeruliste seoste õppimist. Üks populaarsemaid regularisatsiooni tehnikaid on dropout-meetod. Selle meetodi puhul jäetakse igas treeningfaasis juhuslikult osad neuronid välja, kaasaarvatud nendesse neuronitesse viivad ja neist väljuvad ühendused. Selle tulemusena jääb järele vähendatud närvivõrk. Igas treeningfaasis on iga neuroni väljajäämise tõenäosus 1 - p ja tõenäosus, et neuron hoitakse alles on p. Selles faasis kasutatakse treenimiseks ainult vähendatud närvivõrku ning seejärel sisestatakse väljajäetud neuronid võrku tagasi ilma nende kaalusid uuendamata.

2.7 Aktsent

Aktsent on keele normist erinev häädusviis. Inimese aktsent sõltub sellest, millises keskkonnas ta on üles kasvanud. Tavaliselt esineb aktsent võõrkeele rääkimisel. Vahel tekib aktsent ka juhul, kui inimene viibib pikemat aega keskkonnas, kus ei kõnelda tema emakeelt.

Eristatakse kahte tüüpi aktsenti: võõrkeele aktsent ja regionaalne aktsent. Võõrkeele aktsent tekib selle tõttu, et kõneleja emakeeles on häälduseeglid teistsugused kui kõneldavas keeles. Kui näiteks räägitavas keeles ei ole samu häälikuid, mis kõneleja emakeeles, siis asendab ta need häälikutega, mis on tema emakeeles olemas. Regionaalne aktsent esineb ühe keele lõikes ja sõltub sellest, kus inimene elab ja tema millisesse sotsiaalsesse gruppi ta kuulub. Inimesed, kes elavad lähestikku hakkavad ajapikku ühtemoodi hääldama ja nende hääldus erineb mujal elavate inimeste häälduseest [24].

Kui inimene sünnib, siis on ta võimeline õppima igale keelele omaseid hääldusi. Imikueas õpib laps tundma hääldusi, mis on omased tema emakeelele ja õpib eirama hääldusserisusi, mis ei ole tema keele jaoks olulised. Mida vanemaks inimene saab, seda raskem on võõraid hääldusi õppida.

Lisaks eristatakse fonoloogilist ja foneetilist aktsenti. Fonoloogilise aktsendi põhjuseks on erinevused kõneleja emakeele ja võõrkeele fonoloogilises süsteemis. Näiteks jaapanlastel on raske eristada inglise keele foneeme l ja r, kuna jaapani keeles kuuluvad need foneemid samasse fonoloogilisse kategooriasse, inglise keeles aga erinevatesse kategooriatesse. Kui tegemist on foneetilise aktsendiga, siis on inimene omandanud korrektsed fonoloogilised kategooriad, kuid ei suuda produtseerida vastavat foneetilist väljundit [25].

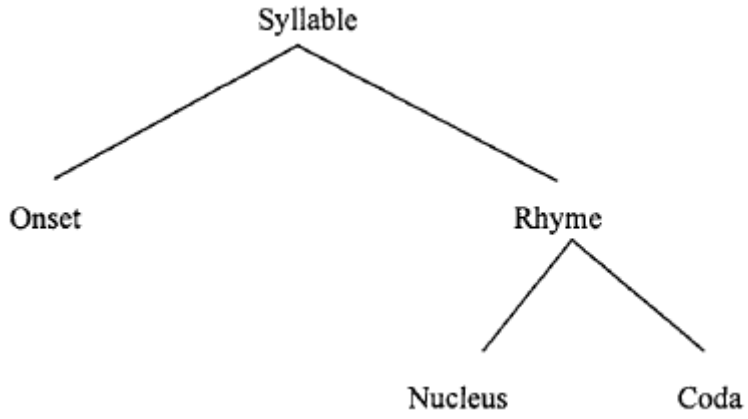
Seda, kui suure aktsendiga inimene võõrkeelt kõneleb, mõjutab suuresti, kui vanalt õppis inimene keelt. Kriitilise perioodi hüpotees väidab, et inimene peab teatud kindlal ajaperioodil keelega kokku puutuma, selleks et õppida keelt ilma aktsendita rääkima. Kui inimene puutub keelega pärast kriitilist perioodi kokku, siis ei ole võimalik õppida ilma aktsendita kõnelema [3].

Aktsendivaba kõne omandamise jaoks on oluline vallata keele fonoloogiat ja selle kolme komponenti: individuaalsed häälikud, häälikute kombinatsioonid, mis moodustavad silbid, ja prosoodia, mille alla kuuluvad rõhk, rütm, toon, intonatsioon. Kui kõneleja ei ole ühte neist komponentidest täielikult selgeks õppinud, siis esineb tema kõnes aktsent [3].

Häälikute all on mõeldud üksikuid helisid, näiteks inglise keele sõnas *two* on häälikud t ja u. Erinevates keeltes on häälikutel erinevad omadused. Inglise keele t hääldatakse ühtemoodi, pannes keel ülemiste hammaste taha, aga näiteks prantsuse ja itaalia keele kõnelejad ei pane t-d hääldades keelt vastu hambaid. Samuti on olulised allofoonilised reeglid, kuidas häälik muutub olenevalt kontekstist. Näiteks inglise keele sõnas *two* on t

aspireeritud, aga sõnas *stew* on t aspireerimata [3].

Häälikust ühe võrra kõrgem ühik on silp. Näiteks võime hääldada silpe, mis algavad kolme konsonandiga (e.g. *strong*), ja silpe, millel on neljane kooda, nagu sõnas *worlds*. Silbi mõistet on raske kõikide keelte jaoks üheselt määrata, sest see sõltub suuresti keelest. Hispaania keele sõnal *adios* on hispaanlaste jaoks kaks silpi, aga inglise keele kõneleja jaoks kolm silpi. Ühe levinuma vaatepunkti järgi koosneb silp kerinevatest alamühikutest nagu on näidatud joonisel 2.5. *Onset* ehk algus koosneb konsonandist või siirdehäälikust, näiteks t ja y sõnade *two* ja *you* ees. See-eest sõnas *off* ei ole niinimetatud algust. Ülejäänud silp on riim, mis koosneb omakorda tuumast ja koodast, mis on jälle kas konsonant või siirdehäälik. Sõna *top* koosneb algusest t, tuumast a ja koodast p. Teatud keeltes ei esine üldse silpe, kus ei ole algust, või ei esine silpe, kus on kooda. Võõrkeele õppijad kipuvad muutma sibistruktuuri, et sobitada seda oma emakeele silbistruktuuriga. Näiteks jaapani silbid peavad lõppema täishäälikuga [3].



Joonis 2.5: Silbi struktuur [3]

Mõiste prosoodia alla kuuluvad rõhk, pikkus, toon, rütm ja ajastus. Rõhk tähendab migi silbi tajutavat esiletõusu. Osades keeltes on rõhk ühtlane, osades keeltes on rõhk viimasel silbil, näiteks prantsuse keeles. Teistes keeltes, näiteks tšehhi ja ungari, on rõhk jällegi esimesel silbil. Rõhumustrite erinevused kanduvad üle võõrkeelde ja nii võib prantsuse emakeelega inimene inglise keelt kõneldes ekslikult panna rõhu viimasele silbile [3].

Pikkus kirjeldab hääliku kvantiteeti. Häälikud on kas lühikesed või pikad, üksikutes keeltes on ka kolmas kategooria [26]. Näiteks eesti keeles võivad nii täis- kui kaashäälikud olla kas lühikesed, pikad või ülipikad [3].

Tooni all mõeldakse hääle kõrgust ja intonatsiooni all hääle kõrguse erinevusi. Toon esineb silbi tasemel, intonatsioon võib esineda silbi tasemest kuni kogu lauseni. Tooni erinevused silbi tasemel võivad kaasa tuua kogu sõna tähenduse muutumise. Intonatsioon viitab hääle kõrguse muutusele kõnes, millega antakse edasi süntaktilisi või semantilisi erinevusi. Intonatsiooni amõjul saab kõnest välja lugeda näiteks, kas tegu on küsimuse või käsklusega või või erinevaid emotsioone nagu üllatus või ebakindlus. Kasutades keelele mitteomast tooni või intonatsiooni, võib tekkida erinevaid kommunikatsiooniprobleeme [3].

Rütm ja ajastus on ühed olulisemad tunnused, mis aitavad erinevaid keeli tuvastada, isegi siis, kui ei ole võimalik sõnu eristada. Rütm ja ajastus on rõhu ja pikkuse korduvad mustrid. Keeled jagunevad kolme erinevasse rütmi klassi: silbiajastuskeeled, rõhuajastuskeeled ja mooraajastuskeeled. Silbiajastuskeeltel paikneb rõhk mingil kindlal silbil ja silbid on enam-vähem sama pikkusega [26]. Silbiajastuskeeled on näiteks prantuses ja hispaania keel [3].

Rõhuajastuskeeltes, näiteks inglise keeles, on rütm seotud rõhuga. Rõhulised silbid on pikad ja rõhutud silbid on lühikesed ning silbid esinevad ühtlaste intervallide tagant [3].

Mooraajastuskeeltes toimub ajastamine silbipikkusühiku moora järgi. Moorad on võrdse pikkusega, seega silp pikkusega 2 moorat on kaks korda pikem kui silp pikkusega 1 moora. Mooraajastuskeele heaks näiteks on jaapani keel. Selle tõttu on inglise keelt õppivatel jaapanlastel tihti raskusi pikemate silpide õigesti hääldamisel, sest nad kipuvad neid venitama palju pikemaks kui vaja [3].

Üldist muljet, kas kõneleja räägib võõrkeelt, nimetatakse globaalseks võõraktsendiks. Võõrkeele aktsenti on seda kergem tuvastada mida pikem ja mitteformaalsem on kõne. Sedas seepärast, et ainult üksikuid sõnu öeldes on kõnelejal võimalik vältida erinevaid fonoloogilisi nähtusi (näiteks häälikud, rõhk, intonatsioon) [3].

2.8 Spektraalsed tunnused

Selleks, et kõneandmeid saaks aktsendituvastuseks kasutada on vaja kõnesignaalist kätte saada tunnused, mis iseloomustavad kõne sisu. Spektraalsete tunnuste saamiseks on eri-

nevaid meetoteid. Meetodid sisaldavad tavaliselt kõnesignaali jagamist teatud pikkustega kaadriteks ning spektraalanalüüsi teostamist.

Tunnuste eraldamise käigus muudetakse kõne lainekuju parameetriliseks esituseks, vähendades seejuures andmete hulka, nii et edasine analüüs oleks lihtsam, aga olulised kõne tunnused ei läheks kaotsi. Eraldamise jaoks on erinevaid lähenemisi ja nende tulemuseks on tavaliselt igale kõnesignaalile vastav multidimensionaalne tunnusvektor.

Enne, kui saab eraldada tunnused, tuleb läbi viia mõned eeltöötluse sammud. Esimene eeltöötluse samm on eel-rõhutus (*pre-emphasis*). Selleks lastakse signaal läbi lõpliku siirdega filtri (FIR-filter). Sellele järgneb kaadri blokeerimine (frame blocking), mis on meetod kõnesignaali kaadriteks jagamiseks. Seejärel kaadritega kõnesignaal akendatakse. Selle eesmärgiks on minimiseerida lahknevusi kaadri alguses ja lõpus. Üks kõige sagedamini kasutatav aknafunktsioone on Hamming-aken [4].

2.8.1 Mel-sageduse kepstri kordajad (MFCC)

Mel-sageduse kepstri kordajate (MFCC, *Mel frequency cepstral coefficients*) arvutamisel püütakse jäljendada inimese kõrva tööpõhimõtet. Kõnesignaal sisaldab erineva sagedusega toone, iga toon tegeliku sagedusega ja subjektiivse helikõrgusega arvutatakse Mel skaalal. Mel-sageduse skaalal on vahed alla 1000 Hz lineaarsed ja üle 1000 Hz on vahed logaritmilised. See tuleneb sellest, kuidas inimkõrv tajub erinevaid helisagedusi.

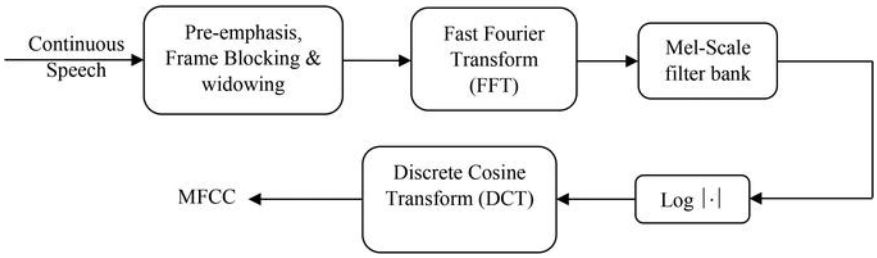
Esimene samm MFCC arvutamisel on kõnesignaali akendamine, et jagada signaal kaadriteks. Pärast seda rakendatakse Fourier' kiirteisendust, et leida iga kaadri võimsuse spekter. Järgnevalt viiakse läbi *filter-bank* töötlus võimsuse spektri peal kasutades Mel skaalat. Kõnesignaali peal rakendatakse diskreetset koosinusteisendust.

Mel-sageduse kepstri kordajad arvutatakse järgneva valemiga:

$$C_n = \sum_{n=1}^{k} (logS_k) cos[n(k - \frac{1}{2})\frac{\pi}{2}] \tag{2.7}$$

kus k on Mel-sageduse kepstri kordajate arv, Sk on filterbank väljund ja Cn on lõplikud Mel-sageduse kepstri kordajad.

Eelpool kirjeldatud MFCC meetodi sammud on toodud välja joonisel 2.6.



Joonis 2.6: MFCC meetodi protseduur [4]

Järgnevalt on kirjeldatud täpsemalt igat sammu MFCC arvutamisel.

Eeltöötluse käigus lastakse signaal läbi filtri, mis rõhutab kõrgemaid sagedusi. See protsess suurendab signaali energiat kõrgematel sagedustel. Filter rakendatakse signaalile x järgmise valemiga, kus kordaja a on tavaliselt 0,95 ja 0,99 vahel.

$$y(n) = x(n) - ax(t-1) \tag{2.8}$$

Teine samm on kõnesignaali kaadriteks jagamine. Võib eeldada, et signaali sagedus ei muutu väga lühikese ajaperioodi jooksul ja pole mõtet teha Fourier' teisendust terve signaali peal. Tehes Fourier' teisenduse lühikeste kaadrite peal, saadakse ligikaudne sageduse kontuur, liites kokku järjestikused kaadrid. Tüüpiline kaadri pikkus kõnetöötluses on 20 kuni 40 ms ja järjestikused kaadrid kattuvad umbes 50 % ulatuses.

Pärast signaali jagamist kaadriteks rakendatakse kaadritele aknafunktsiooni. Üks aknafunktsiooni näidetest on Hamming'i aken, mille kuju on järgnev:

$$w(n) = 0,54 - 0,46cos(\frac{2\pi n}{N-1}), \tag{2.9}$$

kus $0 \leq n \leq N-1$ ja N on akna pikkus. Aknafunktsiooni rakendamine signaalile on väljendatud valemis

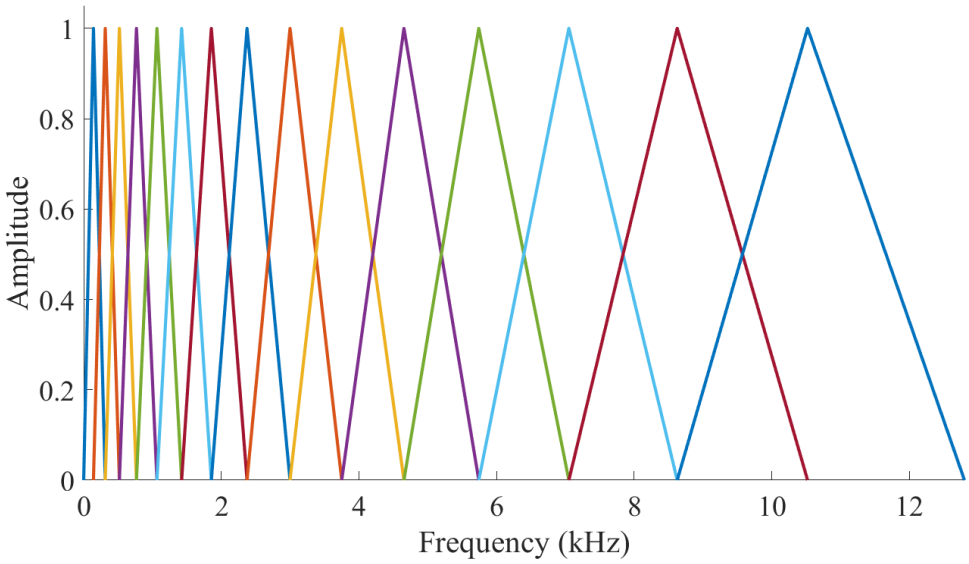
$$y(n) = x(n)w(n), \tag{2.10}$$

kus x(n) on sisendsignaal, y(n) on väljundsignaal ja w(n) on Hamming'i aken

Järgmise sammuna rakendatakse Fourier' kiirteisendust, et teisendada iga kaader aja domeenist sageduse domeeni. Sellele järgneb *filter-bank* töötlus. Selleks kasutatakse kolmnurkseid filtreid Mel-skaalal, mille abil eraldatakse sagedusribad. Iga filtri väljundsagedus on 1 filtri keskel ja väheneb lineaarselt 0 suunas. Mel-skaala eesmärk on imiteerida inimese mittelineaarset kuulmistaju. Sageduse teisendamine Hertz'idest Mel skaalale käib

valemiga 2.11:

$$m = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2.11}$$



Joonis 2.7: *Filter bank* Mel skaalal [5]

Diskreetse koosinusteisendusega teisendatakse Mel-spekter aja domeeni. Saadud tulemused on Mel-sageduse kepstri kordajad.

MFCC on üks enimkasutatud tunnuseid kõne- ja audiotuvastuse rakendustes. Seda peamiselt MFCC töökindluse ja suhteliselt madala keerukuse tõttu [5]. MFCC eelisteks on veel [5]:

- MFCC kvantifitseerib spektri terviklikku kuju, mis on väga oluline näiteks täishäälikute tuvastamisel. Samal ajal eemaldab see peenema spektraalse struktuuri, mis ei ole tihti nii oluline. Seega keskendub see signaali kõige olulisemale osale.
- MFCC arvutamine on lihtne ja sirgjooneline ning arvutuslikult küllaltki efektiivne.
- MFCC suutlikus on põhjalikult läbi testitud.

Teisest küljest on MFCC-l ka puuduseid. Osad nendest on [5]:

- Tajutav skaala ei ole väga hästi ära põhjendatud.
- MFCC ei ole müra suhtes robustne.
- Kolmnurksete filtrite valik ei ole samuti väga hästi ära põhjendatud. Samas pole välja pakutud alternatiivid polulaarsust leidnud.

Peatükk 3

Seotud tööd

Aktsendituvastuse kohta on tehtud mitmeid töid, aga töid, mis üritavad lokaliseerida tunnuseid, mida närvivõrk tuvastamiseks kasutab, on vähe. Selles peatükis kirjeldatakse lähemalt paari tööd, mis on tehtud lokaliseerimisest.

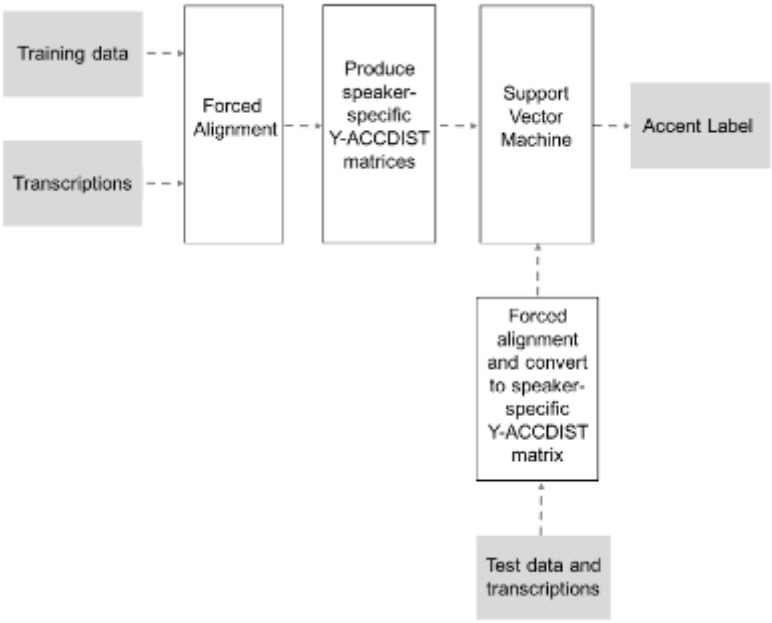
3.1 Segmentaalse sisu mõju automaatsele aktsendituvastusele

Georgina Brown uurib oma töös [6], kuidas tundmatute kõnenäidete häälikud mõjutavad automaatse aktsendituvastuse süsteemi edukust aktsentide klassifitseerimisel. Uurimiseks kasutas ta aktsendituvastussüsteemi Y-ACCDIST [27] [28] ja andmetena kasutas ta põhja-Inglise aktsentide korpust. Töö tulemusena leidis ta, millised häälikud mõjutavad rohkem ning millised häälikud mõjutavad vähem klassifitseerimise edukust.

Töös kasutatud andmed sisaldasid spontaanseid kõnenäiteid inimestelt, kes olid pärit kolmest eri linnast: Manchester, Newcastle ja York. Igas aktsendigrupis oli 15 inimest ja iga inimese kohta oli kasutada 10 minutit transkribeeritud kõnesalvestusi. Eksperimentide jaoks eraldati neist 30 sekundilised lõigud.

Y-ACCDIST süsteem kasutab klassifitseerimiseks tugivektor-masinat (TVM). TVM sisendiks on kõneleja-põhised maatriksid, mis leiti järgmiselt. Kõnenäidete ja trankriptsioonide joondamise tulemusena saadi iga hääliku jaoks MFCC vektor, millest leiti keskmine

MFCC väärtus. Seda kasutati, et koostada maatriks, mis esitab eukleidilist kaugust eri foneemide vahel. Süsteemi tööd kujutatakse diagrammil 3.1.



Joonis 3.1: Y-ACCDIST süsteemi töö käik [6]

Töös tehtud eksperimentides kasutati kõiki 900 kõnenäidet testimiseks. Iga näite klassifitseerimisel kasutati ülejäänud kõnelejate näiteid treenimiseks. Klassifitseerimise täpsus oli 53,3 %. Iga kõnenäite kohta leiti kõikide häälikute esinemissagedused ja märgiti ära, kas näide klassifitseeriti õigesti. Saadud tulemustele teostati logistilise regressiooni analüüsi, et hinnata kas õigesti klassifitseeritud kõnenäidete seas on teatud häälikuid oluliselt rohkem.

Analüüsi tulemusena leiti kolm foneemi: ɛ, u ja ə. Need tulemused saadi kasutades 30 sekundili kõnenäiteid. Autor katsetas ka erinevate kõnenäidete pikkusega ja sai teatud määral erinevad tulemused. Näiteks 20 sekundiliste näidetega tuli esile ainult ɜ häälik, aga 40 sekundiliste näidete puhul ɛ, ɜ ja ɛə. Autor leiab, et tulemused kinnitasid hüpoteesi, et foneemid mõjutavad aktsendituvastuse edukust, aga samas ei ole spetsiifilisi foneeme, mis tagaksid õige klassifitseerimise.

3.2 Sügavate tunnuste leidmine lokaliseerimiseks

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva ja Antonio Torralba tutvustavad oma töös [7] meetodit, mis võimaldab piltidelt lokaliseerida olulised objektid, mille järgi pilti klassifitseeritakse. Selleks kasutavad nad konvolutsioonilist närvivõrku, mis on treenitud piltide klassifitseerimiseks.

Kasutatav närvivõrk koosneb konvolutsioonilistest kihtidest, millele järgneb väärtusi keskmistav ahenduskiht (GAP, *global average pooling*), mis leiab igast konvolutsioonidest saadud tunnuskaardist keskmise väärtuse. Need väärtused on sisendiks täissidusale kihile, mis genereerib soovitud väljundi. Töös genereeritakse klasside aktivatsioonikaardid. Selleks võetakse konvolutsioonilise kihi väljundist saadud tunnuskaardid ja arvutatakse nende kaalutud summa. Kirjeldatud protsess on illustreeritud joonisel 3.2.

Joonis 3.2: Klassi aktivatsioonide leidmine [7]

Saadud klassi aktivatsioonikaarte saab visualiseerida pildi peal ja nii on võimalik näidata, milliseid piirkondi närvivõrk tuvastas mingi klassi kohta.

Töös hinnatakse väljapakutud meetodi sobivust nõrgalt juhendatud objekti lokaliseerimiseks. Selleks kasutati ILSVRC 2014 andmestikku ja testiti meetodit järgmiste konvolutsiooniliste närvivõrkude peal: AlexNet [29], VGGnet [30] ja GoogLeNet [31]. Närvivõrkudest võeti välja täissidus kiht ja asendati see GAP ning sellele järgneva täissidusa softmax-kihiga. Samuti asendati osa konvolutsioonilisi kihte, et saada sobiva suurusega väljundid.

Eksperimentide tulemusena leiti, et klassifitseerimise täpsus langes enamikel juhtudel 1 - 2% võrra, mis tulenes kihtide mudelist eemaldamisest. Samas näidati, et selle meetodi abil lokaliseerimine saavutas palju täpsemaid tulemusi kui võrdlusaluseks valitud BackProp [32] meetod. Täpsust hinnati viie kõige enam ennustatud klassi puhul. Kui GoogleNet peal oli BackProp meetodi eksimisprotsent 50,6 %, siis GAP meetodit kasutades oli eksimisprotsent 43,0%, mis oli kõikidest mudelitest kõige parem tulemus. Autorid leiavad, et tulemus on märkimisväärne arvestades, et närvivõrk ei olnud treenitud spetsiaalselt lokaliseerimiseks.

Peatükk 4

Implementatsioon

Siin peatükis kirjeldatakse täpsemalt, kuidas on võõrkeele aktsendi tuvastamiseks loodud programm implementeeritud.

4.1 Tehnilised valikud

Töö jaoks loodud programmide implementeerimiseks kasutati Pythoni programmeerimiskeelt. Peamine põhjus programeerimiskeele valikul oli Pythoni intuitiivne süntaks ja fakt, et Pythonil on masinõppe jaoks palju olemasolevaid teeke.

Peamine teek, mida töö jaoks kasutati, oli PyTorch. Pytorch on avatud lähtekoodiga masinõppe teek, mida kasutatakse sügavate närvivõrkude arendamiseks ja treenimiseks.

Olulisemad teegid, mida lisaks PyTorchile kasutati, olid NumPy ja Matplotlib. Numpyt kasutati teaduslike arvutuste tegemiseks, Matplotlib'i abil oli võimalik koostada jooniseid eksperimentide tulemustest.

4.2 Andmed

Selleks, et treenida närvivõrku tuvastama võõrkeele aktsenti, on vaja kõnenäiteid, mille abil närvivõrku treenida. Selles töös kasutati mudeli arendamiseks Eesti aktsendikorpuses olevaid helifaile.

4.2.1 Eesti aktsendikorpus

Eesti aktsendikorpus on Tallinna Tehnikaülikooli Küberneetika Instituudis loodav aktsendikorpus, mis sisaldab eesti keelt võõrkeelena kõnelevate inimeste häälduśnäiteid aktsendi akustilis-foneetiliseks uurimiseks. Korpuse peamiseks eesmärgiks on pakkuda uurimismaterjali eesti keele kui võõrkeele hääldusе eksperimentaalfoneetiliseks uurimiseks, kuid silmas on peetud ka võimalikke keeletehnoloogilisi rakendusi, nagu näiteks kõnetuvastuse treenimine aktsendiga kõne tuvastamiseks, kõneleja emakeele automaatne tuvastamine, eesti keele häälduśtreeningu programmi loomine jm. [33]

Kõnenäidete salvestamisel kasutatav tekstikorpus sisaldab 140 erinevat lauset või tekstilõiku. Kuna kõnelejate eesti keele oskus on erinev, sisaldab see peamiselt lihtlauseid, samas on kõik eesti vokaalid ja konsonandid esindatud. [33]

4.2.2 Andmete statistika

Töös on kokku kasutusel 28629 kõnenäidet, mis on jagatud treening- ja testandmestiku vahel. Treeningandmestikus on 23070 kõnenäidet ja testandmestikus 5559. Andmestikud sisaldavad näiteid 18 erineva emakeelega kõnelejatelt. Tabelis 4.1 on esitatud statistika töös kasutusel olevate andmete ja aktsentide kohta. Kuna teatud aktsentide jaoks on andmeid küllaltki vähe, siis klassifitseeritakse kõnenäiteid ainult 8-sse enam esindatud aktsenti ja ülejäänud näited klassifitseeritakse klassi "teised".

4.2.3 Andmete esitus

Selleks, et kasutada andmeid aktsendituvastuseks tuleb need kõigepealt teisendada sobivale kujule. Selles töös kasutatakse pudelikaela tunnuseid. Pudelikaela tunnused on tunnused, mis on treenitud mitmekihilise närvivõrgu poolt, kus ühe peidetud kihi neuronite arv on oluliselt väiksem kui teistel kihtidel. See tekitab kitsenduse, mis koondab klassifitseerimise jaoks asjakohase informatsiooni madala dimensionaalsusega kujule [34]. Pudelikaela närvivõrk peaks muutma andmed vähem tundlikumaks müra ja ülesobitamise suhtes [35].

Aktsent	Treeningandmeid	Testandmeid	Kokku
Vene	6672	278	6950
Soome	3334	834	4168
Eesti	2221	694	2915
Saksa	1946	417	2363
Prantsuse	1666	139	1805
Läti	1529	1390	2919
Leedu	1251	417	1251
Rootsi	973	0	1390
Jaapani	695	278	973
Itaalia	556	278	834
Inglise	417	278	695
Hispaania	417	278	695
Hindi	417	0	417
Slovaki	281	0	281
Poola	278	0	278
Aserbaidžaani	139	0	139
Taani	139	139	278
Holland	139	139	278

Tabel 4.1: Andmete statistika

Töös kasutati tunnuste leidmiseks eesti keelse kõne põhjal treenitud pudelikaela tunnuste eraldajat. Tegemist on aja viite põhise närvivõrguga, mille treenimiseks kasutati Kaldi kõnetuvastuse komplekti. Närvivõrgu sisendiks on Mel-sageduse kepstri koefitsiendid (MFCC), mis on eraldatud iga 10 ms tagant. Iga kaadri klassifitseerimiseks kasutatakse seda kaadrit ennnast ja tema ümber olevaid kaadreid. Mudel sisaldab viite konvolutsioonilist kihti suurusega 650, millele järgneb pudelikaela kiht suurusega 40, millele omakorda järgneb 650-dimensionaalne täissidus kiht. Klassifitseerimiseks kasutab mudel softmax-kihti. Pudelikaela tunnused eraldatakse 40-dimensionaalsest pudelikaela kihist.

4.3 Närvivõrgu mudel

Närvivõrgu mudeli disainimisel oli eesmärgiks leida mudel, mis identifitseeriks aktsente võimalikult suure täpsusega ja millest oleks võimalik lokaliseerida aktsendile omaseid tunnuseid.

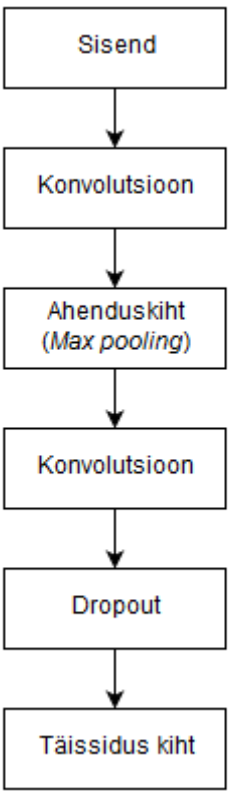
Närvivõrgu sisendiks on eelnevalt kirjeldatud pudelikaela tunnused koos neile vastava hääliku vektoriga. Sisendile rakendatakse kõigepealt ploki normaliseerimist. Sellele järgneb 1-dimensiooniline konvolutsioon. Konvolutsiooni väljundkanalite arv on 32, kerneli suurus on 3, sammu suurus on 1 ja äärise väärtus on 1, mis tähendab, et sisendi mõlemasse äärde lisatakse üks null. Konvolutsioonilisele kihile järgneb väärtuseid keskmistav ahenduskiht, mille akna suuruseks on 2 ja sammu suuruseks samuti 2. Sellele järgneb teine konvolutsiooniline kiht, mille väljundkanalite arv on 64. Konvolutsiooni parameetrid on samad, mis esimesel konvolutsioonil. Mõlemale konvolutsiooni väljundile rakendatakse mittenegatiivset lineaarfunktsiooni (ReLU). Saadud väljund on sisendiks täissidusale kihile.

Regulatsiooni meetodina rakendatakse dropout-meetodit. Elemendi väljajätmise tõenäosus igas treeningtsüklis on 0,2. Närvivõrgu mudeli struktuur on kujutatud joonisel 4.1.

4.4 Treenimine

Närvivõrgu treenimisel kasutatati miniplokktreeningu põhimõtet, mis tähendab, et treeningandmed jagati plokkideks ja mudeli parameetreid uuendatakse pärast kõigi plokis olevate andmete läbitöötamist. Selle eeliseks on suurem arvutuslik efektiivsus võrreldes stohhastilise gradientlaskumisega, kus uuendatakse mudelit pärast iga treeningnäite analüüsimist. Teisest küljest värskendatakse mudelit tihedamini kui ainult pärast kogu treeningandmestiku läbi käimist. See peaks vältima närvivõrgu liiga varast konvergeerumist. Närvivõrku treeniti plokkidega, mille suurus oli 32, mis andis hea tasakaalu kiire konvergeerumise ja täpsete veahinnangute vahel.

Treenimiseks on vajalik, et plokis olevad andmed oleksid maatriksi kujul, mille mõõtmeteks on ploki suurus korda kõnenäite tunnusvektorite arv. Probleemiks on, et kõnenäited



Joonis 4.1: Närvivõrgu mudeli struktuur

on erineva suurusega. Selle lahendamiseks lisatakse plokis lühemate näidete lõppu nullid, nii et kõik näited oleks sama pikad, kui ploki kõige pikem näide. Selleks, et vähendada nullide lisamist sorteeritakse andmed ja jagatakse need plokkideks nii, et ühes plokis oleks võimalikult sarnase suurusega andmed.

Närvivõrgu parameetrite uuendamiseks kasutatakse stohhastilise gradientlaskumise algoritmi. Pytorch'is on algoritm implementeeritud järgmise valemi järgi:

$$v = \rho * v + gp = p - lr * v \tag{4.1}$$

Valemis tähistab p parameetreid, g gradienti, v kiirust ja ρ inertsitegurit.

Optimeerimisalgoritmi parameetrid on valitud järgnevalt: inertsitegur on 0,5, algne õpisamm on 0,01. Õpisammu vähendatakse dünaamiliselt treeningu käigus. Kui kahju pole kahe epohhi jooksul paranenud, siis vähendatakse õpisammu poole võrra.

Treenimise käigus mudeli hindamiseks kasutatakse kahjufunktsioonina ristentroopiakahju. Selle funktsiooni puhul kahju kasvab, kui mudel ennustab suure tõenäoususega klassi,

mis erineb tegelikust klassist.

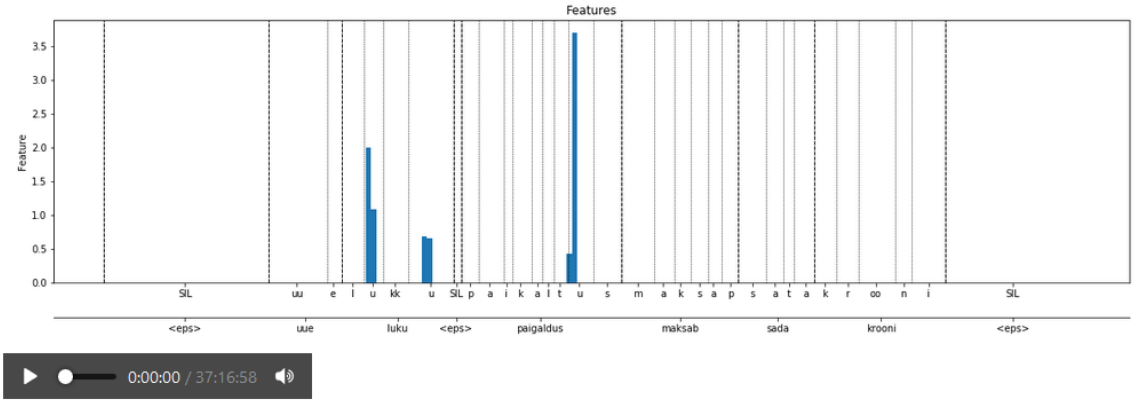
Mudelit treeniti 20 epohhi. Lõplikuks mudeli hindamiseks leiti protsentuaalselt, kui palju näidetest suutis mudel õigesti klassifitseerida. Treenitud närvivõrgu mudeli täpsus treeningandmete peal oli 81,1 %. Andes sisendiks testandmed oli täpsus mõistagi madalam - 51,6 % .

4.5 Tunnuste leidmine

Närvivõrgu teise konvolutsioonilise kihi väljundiks on tunnuskaardid. Tunnuskaardi puhul on tegu maatriksiga, kus on igale tunnusele vastavad vektorid. Tunnuste arv on võrdne konvolutsioonis kasutatud filtrite arvuga, seega leiti kokku 64 tunnust. Iga filter analüüsis erinevat tunnust ja selle tulemusena saadi igale tunnusele vastav vektor. Vektori pikkus vastab kõnefaili pikkusele, täpsemalt on vektori suurus faili kaadrite arv jagatud kahega. Seega on iga vektori element teatud skalaar, mis iseloomustab tunnuse esinemist kahe kaadri lõikes.

Töö käigus leiti tunnuskaardid kõikide testandmetes olevate kõnefailide kohta. Samuti leiti närvivõrgust aktsendi klassidele vastavad tunnuste kaalud. Selle abil oli võimalik tuvastada, millised on iga keele jaoks olulisemad tunnused, mille põhjal närvivõrk klassifitseerimise otsust teeb. Mida suurem on teatud tunnuse kaal aktsendi klassi jaoks, seda suuremat rolli omistab närvivõrk sellele tunnusele, otsustades, kas ta kuulub sellesse klassi.

Teades iga aktsendi olulisemaid tunnuseid ja olles leidnud kõnefailide jaoks tunnuskaardid, sai lokaliseerida tunnuste asukohad kõnenäidetes. Tunnuste analüüsimiseks leiti iga tunnuse jaoks kõnenäited, kus tunnuse väärtus oli kõige suurem. Samuti kirjutati programmijupp, mille abil on võimalik kujutada graafikut, mis näitab tunnuse väärtust igas kõnenäite kaadris, ja samal ajal kõnenäite audiot esitada. Graafiku põhjal on hea analüüsida, millise hääliku juures tunnus esineb. Joonis 4.2 on näide programmi väljundist.



Joonis 4.2: Näide graafikust, mis kujutab tunnuse väärtusi erinevates kõnenäite kaadrites

Peatükk 5

Tulemused

5.1 Leitud tunnused

Selleks, et tuvastada, milliseid tunnuseid närvivõrk leiab, kuulati läbi kõikide tunnuste jaoks vastavad kõnenäited ja analüüsiti neid. Siinkohal oli abiks foneetik Einar Meister, kelle abiga õnnestus tunnustele täpsem nimetus panna. Tabelis 5.1 on välja toodud tunnused, millele leidus keeleline seletus. Palju oli selliseid tunnuseid, millele ei andnud keelelist seletust leida. Näiteks olid paljud tunnused seotud salvestuste eripäradega.

Tabelis on välja toodud ainult need tunnused, mida õnnestus keeleliselt tõlgendada. Nii toimides jäi 64-st tunnusest alles 22. Tabelis on iga tunnuse kohta märgitud häälik või häälikud, mille juures tunnus esineb, samuti on välja toodud aktsendid, mille puhul tuleb vastav tunnus kõige tugevamini esile. Osa tunnuseid olid väga selgelt omased peamiselt ühele konkreetsele aktsendile, samas oli ka selliseid tunnuseid, kus ei saanud ühte või kahte konkreetset aktsenti välja tuua. Nende puhul on tabelisse kirjutatud kolm keelt, kus tunnuse väärtus oli kõige suurem, aga tunnus esines ka teistes aktsentides.

Kolmandaks on tabelis välja toodud tunnusele vastava keelelise nähtuse nimetus. Tuleb arvestada, et nähtuste tõlgendused on subjektiivsed ja ei ole võimalik kindlalt väita, et närvivõrk just selliseid keelelisi nähtuseid tunnustena näeb. Mõne tunnuse puhul oli suhteliselt selgesti eristatav aktsendile iseloomulik tunnus, aga mõne puhul on tegu pigem oletusega.

Tabelist 5.2 on näha, et kõige sagedamini esinev tunnus on palatalisatsioon. Palatalisat-

Häälik	Nähtuse nimetus	Aktsendid, kus esineb
li	palatalisatsioon	soome
i	eelmise hääliku palatalisatsioon	vene
k/p/t	aspiratsioon	saksa
le	rõhutu vokaali redutseerumine	soome, hispaania
a	rõhk vales kohas	saksa
t	rõhk vales kohas	prantsuse, läti
l/j	palatalisatsioon	läti, leedu
l	palatalisatsioon	soome
a	rõhutu vokaali redutseerumine	taani, soome, läti
s	hälve palataliseerimata s-st	itaalia, saksa, inglise
i	i spektri kvaliteet	itaalia, saksa, vene
s/t	palatalisatsioon	itaalia, jaapan, saksa
ta	klusiili sulufaasi kestus	soome
e	rõhutu vokaali redutseerumine	vene
i	rõhutu vokaali redutseerumine	läti, jaapan
l	palatalisatsioon	soome
i	palatalisatsioon	saksa, inglise, prantsuse
e	rõhutu vokaali redutseerumine	vene
s	s kvaliteet	läti, leedu
s	j spektri kvaliteet	läti, leedu
l	palatalisatsioon	soome
l	palatalisatsioon	leedu

Tabel 5.1: Tunnused

siooni ehk hääliku peenendamise all mõistetakse ülilühikest i-d, mis kostab palataliseeritava konsonandi järel. Eesti keeles palataliseeruvad häälikud nagu l,n,s ja t. Aktsendiga kõnelejad hääldavad neid tihti palataliseerimata. Närvivõrk tuvastas peamiselt palataliseerimata l-i soome aktsendi puhul, aga palatalisatsiooni vigu esineb ka näiteks läti, leedu ja vene aktsendi puhul. Samuti on kaks tunnust, kus palatalisatsiooni viga esineb s või t juures. Nendel puhkudel on kõnenäidetes hälve eesti keele palataliseerimata häälikute suhtes. Osade tunnuste asukoht pole palataliseerimata hääliku juures, vaid on sellele eel-

neva või järgneva hääliku koha peal. See tuleneb tõenäoliselt sellest, et hälve ühe hääliku hääldamisel mõjutab kõrvalasuvate häälikute hääldamist ja närvivõrk tuvastab hälvet teise hääliku koha peal.

Lisaks palatalisatsioonile oli palju tunnuseid, kus esines vokaali redutseerumine. Selle all mõeldakse muutust vokaali kvaliteedis, mis võib seisneda vokaali kestvuses või rõhus ja mille puhul häälda takse vokaali lühemalt või nõrgemini. Närvivõrk leidis vokaali kvaliteedi redutseerumisega seotud tunnuseid mitmete täishäälikute kohta ja seda esines erinevate aktsentide puhul.

Muudest tunnustest õnnestus tuvastada näiteks aspiratsioon. Eesti keeles häälda takse k, p ja t aspireerimata, aga osad aktsendiga kõnelejad hääldavad neid aspireeritult ehk lisavad h hääliku sinna juurde. Aspiratsiooni esines peamiselt saksa keele aktsendi korral.

Osasi tunnuseid tõlgendati rõhu vigadena, st kõnelejad panid hääldamisel rõhu valesse kohta. Üht tunnust tõlgendati klusiili sulufaasi kestuse veana. Tunnus esines peamiselt soome aktsendiga kõnelejate puhul häälikuühendi "ta"juures.

5.2 Tunnused aktsentide lõikes

Töö käigus leiti ka kõige olulisemad tunnused erinevate aktsentide lõikes. Tabelis 5.2 on esitatud viis olulisemat tunnust soome, vene, saksa, prantsuse, läti ja leedu aktsentide jaoks. Iga aktsendi jaoks leiti olulisemad tunnused selle järgi, milliste tunnuste kaal oli selle aktsendiklassi jaoks kõige suurem. Sarnaselt tabelile 5.1 on iga tunnuse kohta välja toodud sellele vastav häälik ja võimaluse korral ka keelelise nähtuse nimetus. Tabelis on ka tunnused, mille kohta ei õnnestunud kuulates mingit keelelist seletust anda, aga mille puhul tuli selgelt välja häälik, mille juures vastav tunnus esineb. Seetõttu on nähtuse nimetus jäetud mitmes kohas märkimata.

5.3 Tulemuste analüüs

Saadud tulemustest on näha, et töös kasutatud lähenemine oli väga edukas osade tunnuste leidmisel, aga samas oli lähenemisel ka puudusi.

Soome aktsent		Vene aktsent	
Häälik	Nähtuse nimetus	Häälik	Nähtuse nimetus
l	palatalisatsioon	at/ot	-
aj	j spektri kvaliteet	u	-
le	vokaali redutseerumine	i	-
i	palatalisatsioon	e	vokaali redutseerumine
a	-	s	-
Saksa aktsent		**Prantsuse aktsent**	
Häälik	Nähtuse nimetus	Häälik	Nähtuse nimetus
a	rõhk vales kohas	t	rõhk vales kohas
oo	-	e	-
k/t	aspiratsioon	u	-
u	-	a	vokaali redutseerumine
i	palatalisatsioon	i	vokaali redutseerumine
Läti aktsent		**Leedu aktsent**	
Häälik	Nähtuse nimetus	Häälik	Nähtuse nimetus
a	-	s	s kvaliteet
l	palatalisatsioon	l	palatalisatsioon
s	s kvaliteet	li	palatalisatsioon
l	palatalisatsioon	i	-
s	-		

Tabel 5.2: Tunnused aktsentide lõikes

Meetodi abil tuli väga hästi välja l palatalisatsioon. L-i hääldamine palataliseerimata on väga omane soome keele kõnelejatele ja seega oli ootuspärane, et see tunnus just soome keele puhul esile kerkis. Samuti tuli leitud tunnuste seast hästi välja k ja t aspiratsioon saksa keele puhul.

Kõik aktsendi tunnused siiski kasutatud lähenemisega nii edukalt välja ei tulnud. Näiteks õ ja ü hääliku hääldamine valmistab muu emakeelega inimeste jaoks tihti raskusi, aga töös ei tulnud välja, et närvivõrk seda tunnust tuvastaks. Selle põhjuseks võib olla, et treeningandmetes esineb vähe näiteid, kus need häälikud esinevad, ja selle tõttu ei ole

närvivõrk õppinud seda tuvastama.

Tulemustest tuleb ka välja, et paljude tunnuste kohta ei õnnestunud keelelist seletust leida. See tähendab, et oli tunnuseid, mida närvivõrk kasutas aktsendi identifitseerimiseks ja mis omasid klassifitseerimisel ka suurt kaalu, aga mille puhul ei ole tegemist keelelis-te tunnustega. Tõenäoliselt oli paljude selliste näidete puhul tegemist kõne kvaliteediga seotud nähtustega. Samuti võis närvivõrk leida keelelisi tunnuseid, mis on inimese jaoks raskesti tajutavad.

Tulemuste põhjal võib tunnuste lokaliseerimise lugeda õnnestunuks. Kõikide leitud tunnuste kohta leidus tunnusele vastav häälik. Enamikel juhtudel vastas tunnusele teatud spetsiifiline häälik, aga teatud juhtudel mitu häälikut, näiteks aspiratsioon helitute sulghäälikute k, p, t puhul. Rohkem oli siiski selliseid näiteid, kus ühele tunnusele vastas mingi kindel häälik ja ühe keelelise nähtuse kohta oli mitu tunnust.

5.4 Ettepanekud tulevaseks tööks

Käesolevas töös kasutatud meetodi abil õnnestus leida mitmeid tunnuseid, mida närvivõrk kasutab aktsendi tuvastamiseks. Võib väita, et selline lähenemine on närvivõrgu interpreteerimisel asjakohane, samas ei ole töös kasutatud meetod maksimaalselt välja arendatud ning sellel on palju ruumi täiendusteks. Järgnevalt on välja toodud mõned ideed, kuidas saaks antud tööd parandada ja edasi arendada.

Närvivõrgu süsteemide edukus sõltub suuresti andmetest, mida närvivõrk kasutab treenimiseks. Valminud töös kasutati kõnenäiteid umbes 200 erinevalt inimeselt, aga kõnenäidete arv erinevate keelte lõikes varieerus oluliselt. Kui vene emakeelega kõnelejaid oli 50 ja kõnenäiteid vene aktsendi jaoks 6950, siis näiteks leedu keele kohta oli kasutada ainult 9 erineva inimese andmed ja 1250 kõnenäidet. Selle tõttu on närvivõrgul raske teha üldistusi aktsentide kohta, mille jaoks on vähe andmeid ja tulemused nende aktsendigruppide kohta võivad olla ebatäpsed. Suurendades treenimiseks kasutatud kõnenäidete arvu, eriti aktsentide jaoks, mille kohta on hetkel vähe andmeid, oleks võimalik muuta närvivõrku täpsemaks ning leida rohkem tunnuseid.

Lisaks andmete hulga suurendamisele aitaks närvivõrgu täpsust parandada mudeli edasi

arendamine. Käesolevas töös ei keskendutud spetsiaalselt väga kõrge mudeli täpsuse saavutamisele, mis väljendub ka mudeli testimisel saadud täpsusest, milleks oli 51,6 %.

Antud töös kasutatud lähenemist saab kasutada ka muude andmete jaoks. Näiteks võib sama meetodit kasutada mõne teise keele kui eesti keele kohta ja uurida, millised tunnused tulevad välja selle keele erinevate aktsentide kohta. Lisaks võõrkeelte aktsentidele on võimalus ka tuvastada regionaalseid aktsente.

Peatükk 6

Kokkuvõte

Lõputöö eesmärk oli luua närvivõrkude abil süsteem võõrkeele aktsentide tuvastamiseks, mille abil oleks võimalik lokaliseerida tunnuseid, mida närvivõrk kasutab. Töös üritati interpreteerida närvivõrgu toimimist ja leida häälikud ning keelelised tunnused, mida närvivõrk õppis ning mille alusel ta klassifitseerimise otsuseid tegi.

Eesmärkide täitmiseks disainiti närvivõrgu mudel. Mudel koosnes kahest konvolutsioonilisest kihist, mille vahel oli väärtusi keskmistav ahenduskiht. Närvivõrgu treenimiseks kasutati Eesti aktsendikorpuses olevaid erineva keeletaustaga inimeste eesti keelseid kõnenäiteid. Kokku oli töös kasutusel üle 28000 kõnenäite, mis pärinesid 18 erineva emakeelega kõnelejalt.

Konvolutsiooniliste kihtide väljundist saadi kõnenäidete tunnuskaardid. Lisaks leiti närvivõrgust erinevatele aktsendiklassidele vastavad tunnuste kaalud. Ühendades tunnuskaardid ja tunnuste kaalud õnnestus lokaliseerida tunnuste asukohad kõnenäidetes. Seejärel analüüsiti erinevaid saadud tunnuseid, leiti igale tunnusele vastav häälik ja üritati leida tunnustele keeleline seletus.

Töö tulemusena saadi 22 tunnust, mille jaoks pakuti välja keeleline nähtus, mida närvivõrk võis tuvastada. Peamiste tunnustena kerkisid esile palatalisatsioon, vokaali redutseerimine ja rõhu hääldamine vales kohas. Kõige selgemini eristuvad tunnused olid seotud l-i hääldamisega palataliseerimata, mis esines sagedasti soome keele aktsendis. Samas oli palju tunnuseid, mida ei õnnestunud keeleliselt seletada. Töös leiti ka erinevate aktsentide lõikes kõige olulisemad tunnused ja toodi välja, millistel häälikutel on eri aktsentide

klassifitseerimisel kõige suurem roll.

Tööst saadud tulemustest võib järeldada, et kasutatud meetodi abil on võimalik närvivõrgu poolt leitud tunnuseid interpreteerida.

Kirjandus

[1] Haojie Chai, Xianming Chen, Yingchun Cai, and Jingyao Zhao. Artificial neural network modeling for predicting wood moisture content in high frequency vacuum drying process. *Forests*, 10(1):16, 2019.

[2] Marlon Oliveira, Houssem Chatbri, Suzanne Little, Noel E O'Connor, and Alistair Sutherland. A comparison between end-to-end approaches and feature extraction based approaches for sign language recognition. In *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2017.

[3] Roy C Major. *Foreign accent: The ontogeny and phylogeny of second language phonology*. Routledge, 2001.

[4] Sabur Ajibola Alim and Nahrul Khair Alang Rashid. Some commonly used speech feature extraction algorithms. *From Natural to Artificial Intelligence-Algorithms and Applications*, 2018.

[5] Cepstrum and mfcc. URL `https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC`. (28.12.2019).

[6] Georgina Brown. Segmental content effects on text-dependent automatic accent recognition. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 9–15, 2018.

[7] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[8] Maryam Najafian, Saeid Safavi, Philip Weber, and Martin Russell. Identification of

british english regional accents using fusion of i-vector and multi-accent phonotactic systems. In *ODYSSEY*, pages 132–139, 2016.

[9] Gil Keren, Jun Deng, Jouni Pohjalainen, and Björn W Schuller. Convolutional neural networks with data augmentation for classifying speakers' native language. In *INTERSPEECH*, pages 2393–2397, 2016.

[10] Shamalee Deshpande, Sharat Chikkerur, and Venu Govindaraju. Accent classification in speech. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, pages 139–143. IEEE, 2005.

[11] Pieter-Jan Ghesquiere and Dirk Van Compernolle. Flemish accent identification based on formant and duration features. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–749. IEEE, 2002.

[12] Hong Tang and Ali A Ghorbani. Accent classification using support vector machine and hidden markov model. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 629–631. Springer, 2003.

[13] Andrea DeMarco and Stephen J Cox. Iterative classification of regional british accents in i-vector space. In *Symposium on machine learning in speech and language processing*, 2012.

[14] Mohamad Hasan Bahari, Rahim Saeidi, David Van Leeuwen, et al. Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7344–7348. IEEE, 2013.

[15] Hamid Behravan, Ville Hautamäki, and Tomi Kinnunen. Factors affecting i-vector based foreign accent recognition: A case study in spoken finnish. *Speech Communication*, 66:118–129, 2015.

[16] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[17] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury,

et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[18] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal processing letters*, 21(1): 65–68, 2014.

[19] Heiga Zen and Haşim Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4470–4474. IEEE, 2015.

[20] Yishan Jiao, Ming Tu, Visar Berisha, and Julie Liss. Online speaking rate estimation using recurrent neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5245–5249. IEEE, 2016.

[21] Javier Gonzalez-Dominguez, Ignacio Lopez-Moreno, Haşim Sak, Joaquin Gonzalez-Rodriguez, and Pedro J Moreno. Automatic language identification using long short-term memory recurrent neural networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[22] Yishan Jiao, Ming Tu, Visar Berisha, and Julie M Liss. Accent identification by combining deep neural networks and recurrent neural networks trained on long and short term features. In *Interspeech*, pages 2388–2392, 2016.

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[24] Rosina Lippi-Green. *English with an accent: Language, ideology and discrimination in the United States*. Routledge, 2012.

[25] Lya Meister et al. Vene aktsent eesti keeles: akustilise analüüsi tulemusi. *Eesti Rakenduslingvistika Ühingu aastaraamat*, (2):131–152, 2006.

[26] Foneetika sõnastik. URL `https://term.eki.ee/termbase/view/9062800/`. (28.11.2019).

[27] Georgina Brown. *Y-ACCDIST: An automatic accent recognition system for forensic applications*. PhD thesis, University of York, 2014.

[28] Georgina Brown. Automatic recognition of geographically-proximate accents using content-controlled and content-mismatched speech data. In *ICPhS*, 2015.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] L Meister and E Meister. Aktsendikorpus ja võõrkeele aktsendi uurimine. *Keel ja Kirjandus*, 55(8-9):696–714, 2012.

[34] Dong Yu and Michael L Seltzer. Improved bottleneck features using pretrained deep neural networks. In *Twelfth annual conference of the international speech communication association*, 2011.

[35] Pavel Matejka, LeŽhang, Tim Ng, HarishŠri Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang. Neural network bottleneck features for language identification. *Proc. IEEE Odyssey*, pages 299–304, 2014.