TÉCNICO LISBOA

Nasality detection in Parkinson's Disease

Matilde Maria Silva Peixoto

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Alberto Abad Gareta
Prof. Isabel Maria Martins Trancoso

Examination Committee

Chairperson: Prof. Pedro Filipe Zeferino Aidos Tomás
Supervisor: Prof. Alberto Abad Gareta
Member of the Committee: Dr. Rubén Solera Ureña

November 2022

Declaração/Declaration

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First of all, I would like to thank my supervisors, Prof. Alberto Abad and Prof. Isabel Trancoso, for not only providing me with a better understanding of speech processing, as I had very little prior knowledge, but also for their guidance and encouragement throughout the development of this thesis. Also, to Thomas Rolland, Carlos Carvalho and Catarina Botelho thank you for always taking the time to help me when I had doubts.

I would also like to leave a word of acknowledgement to the team from the Faculdade de Medicina da Universidade de Lisboa that shared with us the FralusoPark database; including Dr. Rita Cardoso, Dr. Helena Santos, Dr. Joana Carvalho, Prof. Isabel Guimarães, and Prof. Joaquim J. Ferreira.

I would like to express my gratitude to my parents for everything they have done for me over the years. Thank you for always believing in me and supporting me in everything I do. I would not be who I am without you.

To my sister, Catarina, thank you for all the moments we have shared growing up, for always encouraging me and for setting an example.

Last but not least, to all my friends, thank you for being there for me throughout my academic years and for providing me with good times and distractions from work.

Abstract

Neurodegenerative diseases, such as Parkinson's disease, are disabling conditions that constrain the daily lives of those who suffer from them. With the ageing of the general population they are expected to become more predominant. The advances made in speech analysis and machine learning, have revealed the potential to automatically detect these disorders through speech.

The objective of this Master Thesis is to assess the viability of automatically detecting hypernasality in Parkinson's disease for European Portuguese. Existing machine learning models for hypernasality detection for other languages are mostly trained on disordered speech databases. Given that European Portuguese is a low-resourced language, the approach used in this thesis was to use a healthy speech database to model the characteristics of nasalisation. We created a deep neural network classifier that divides the sounds into oral or nasal consonants and vowels. Our best results for this classifier (accuracy of 84.42%) were achieved with a time-delay neural network. Using the output of the classifier as acoustic correlates of nasality, we created a nasality score for the prediction of hypernasality in Parkinson's disease patients. The analysis made to the results shows that there are statistically significant differences between the control group and the Parkinson's group.

Keywords

Parkinson's disease, Speech, Nasalisation, Deep Learning.

Resumo

As doenças neurodegenerativas, como o Parkinson, são condições incapacitantes que restringem o quotidiano daqueles que delas sofrem. Devido ao envelhecimento da população em geral, é esperado que estas se tornem cada vez mais predominantes. Com os avanços em processamento da fala e aprendizagem automática, foi identificado o potencial para detectar automaticamente estes distúrbios através da fala.

O objetivo desta Dissertação de Mestrado é avaliar a viabilidade do detetar automaticamente hipernasalidade nos doentes de Parkinson para o Português europeu. A grande maioria dos modelos de aprendizagem automática que existem para detecção de hipernasalidade para outros idiomas através da fala são treinados com *corpus* de gravações de pacientes. Dado que o português europeu é uma língua com poucos recursos, a abordagem utilizada nesta tese consistiu em utilizar um *corpus* de indivíduos saudáveis para modelar as características da nasalização. Foi criada uma rede neuronal profunda que classifica os fonemas em consoantes e vogais orais ou nasais. Os melhores resultados obtidos para este classificador (exatidão de 84,42%) foram alcançados utilizando uma *time delay neural network* (TDNN). A saída desta rede foi utilizada para criar um valor de nasalação com o intuito de utilizar este valor para detetar hipernasalidade em pacientes de Parkinson. A análise feita aos resultados mostra que existem diferenças estatisticamente significativas entre o grupo do controlo e o grupo de Parkinson.

Palavras Chave

Parkinson, Fala, Nasalação, Aprendizagem Profunda

Contents

List of Figures

List of Tables

Acronyms

ALS	amyotrophic lateral sclerosis
ANOVA	Analysis of Variance
ASR	Automatic Speech Recognition
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DDK	Diadochokinetic
DL	Deep Learning
DNN	Deep Neural Network
EP	European Portuguese
FFT	Fast Fourier Transform
GeMAPS	Geneva Minimalistic Acoustic Parameter Set
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
HD	Huntington's Disease
KNN	k-Nearest Neighbours
LDA	Linear Discriminant Analysis
LLD	Low-Level Descriptors
MFCC	Mel-Frequency Cepstral Coefficients
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLD	Nonlinear Dynamics
NLP	Natural Language Processing
NN	Neural Networks

PD	Parkinson's Disease
PLDA	Probabilistic Linear Discriminant Analysis
PLP	Perceptual Linear Prediction
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
VLHR	voice low tone to high tone ratio

1

Introduction

Contents

1.1 Motivation

Dealing with any type of long-term health condition can create stress and burden in the lives of the patients and of those around them. Neurodegenerative diseases can greatly affect every level of the patient's life and can also result in a total ineptitude to perform everyday tasks. These patients may have: breathing difficulties, motor problems, cognitive problems or gradual memory loss (with the possibility of affecting long-term memory) [3].

According to the Institute for Health Metrics and Evaluation (IHME), the third most prevalent cause of disability and premature death in the EU are neurological disorders. A burden that, with the continuing ageing of the European population, is expected to increase. In 2017, the total number of people in the EU that had a neurological disorder was approximately 21 million [4].

Speech has been identified as a potential biomarker for diseases that affect the organs involved in its planning and production. These diseases include respiratory diseases such as the common cold or obstructive sleep apnea, mood disorders such as depression and bipolar disease and neurodegenerative diseases such as Parkinson's and Alzheimer's.

Even though neurodegenerative diseases do not have a cure yet, with prevention and early discovery, the symptoms can be treated and the progression of the disease delayed for a longer period of time. Furthermore, the monitoring of the different symptoms can help understand the evolution of the disease over time. This is where speech can make a difference. Speech is a non-invasive and cost-effective route to early diagnosis and monitoring of several diseases. Combined with experts evaluation, speech could help increase the quality of life of those who deal with these conditions.

Digital biomarkers, such as speech, have the potential to provide a complementary assessment approach to existing clinical assessments, as they allow non-invasive, objective, ecologically valid assessment that can be done easily and during day-to-day life. Additionally, a very important advantage in the current times, is that these types of assessment can be done remotely, which increases accessibility and lowers the inherent risks of going to healthcare centres. The use of these biomarkers eases the long-term follow-up with continuous and frequent check-ups, which in turn may result in more complete data that can help the physicians make better informed decisions [5].

One of the many voice qualities that can be perceived through speech is nasalisation. Nasal resonance can be defectively produced in speech by those who have velopharyngeal mechanism problems (hypernasal speech), resulting in a part of the air exiting through the nose in non-nasal sounds. As hypernasality can be a sign of problems with the peripheral nervous system, the central nervous system, or both, it is important for clinicians to identify it [6].

1.2 Objectives

The main goal of this thesis was to develop a system that analyses and monitors the level of nasalisation in a person's voice for European Portuguese (EP). Diseases such as Parkinson's may cause this feature of a person's voice to be more prominent than in healthy individuals.

To accomplish the main goal, we trained models to classify phonemes as nasal or non-nasal. A few distinct models were trained and evaluated, and the resulting solutions were compared in order to assess which one produces the best results.

From the outputs of the trained models, a nasalisation score was computed. The validity of this method as a means of detecting and monitoring hypernasality was assessed by comparing the results of the nasality score in the control and non-control groups to determine whether there is a statistically significant difference between the scores of the two groups.

1.3 Thesis Outline

This document is divided into 6 chapters. This first chapter is an introduction that outlines the motivation for this work and the objectives it proposes to accomplish. Chapter 2 presents an overview of concepts and theoretical background that are important to better understand this work. Chapter 3 explains the overall system developed in this thesis and introduces the corpora that were used. Chapter 4 provides an insight of the architectures built and chapter 5 presents and analyses the results obtained. Lastly, chapter 6 presents the conclusions taken from this work and gives suggestions for future work that might be developed.

2

Background

Contents

This chapter starts with a necessarily brief overview of the features that can be extracted from speech. An analysis of the several voice correlates that identify each disease under study will also be performed. The chapter concludes with an outline of the classification algorithms and the metrics used to evaluate these algorithms.

2.1 Speech Signal Processing

Signal processing corresponds to a technique that retrieves information from a signal in an efficient manner in order to make it more appropriate for a specific application [7]. In the particular case of speech, there is a diverse range of features that can be retrieved, such as prosodic features, vocal quality features and spectral features, all of which will be explained below.

2.1.1 Prosodic Features

Prosody refers to the speech characteristics that are not discrete phonetic segments (phonemes), but rather attributes of larger units of speech known as suprasegmentals. These correspond to the patterns of rhythm and melody of speech including pitch, loudness and duration of sounds and pauses. Therefore, this set of variables affects how a message is communicated. For example, the meaning of the phrase ”He did that” can change according to the intonation, it can be either a question, a statement or an exclamation.

Prosodic features can be objectively measured from speech signals by collecting variables such as the fundamental frequency (F0), duration of voiced and unvoiced sections, intensity and periodicity measures [8]. Impairments in these types of speech features include changes in pitch, loudness and timing, which may result in low intelligibility of speech [9] and can be correlated to several diseases.

2.1.2 Voice Quality Features

Voice quality features are, as the name indicates, related to the quality of the sounds produced. Some qualities that can suggest the existence of a speech disorder are excessive roughness or breathiness.

The most common voice quality measures used in relation with several disorders are [10]:

- Jitter - which is the local frequency variations in subsequent periods of a voiced speech signal. A high jitter causes a perception of roughness in the voice.
- Shimmer - that corresponds to the local amplitude variations in subsequent periods of a voiced speech signal.

- Harmonic-to-Noise Ratio (HNR) - which is the ratio between the harmonics of a sound and the other spectral peaks that do not correspond to a multiple of F0. Essentially, it represents the quantity of noise in the signal.

2.1.3 Spectral Features

Spectral features are features of the frequency-domain that are attained by converting a time signal to a frequency signal using the Fourier Transform. The Mel-Frequency Cepstral Coefficients (MFCC) are considered the *de facto standard* of these features.

The MFCC are the set of coefficients that comprise the Mel-frequency cepstrum, which, in turn, is a representation of the short-term power spectrum of a sound. MFCC carry information about the short-term magnitude spectrum of the speech signal and are an efficient model of the vocal tract transfer function.

In order to obtain the MFCC from a speech signal, there are a number of steps to be completed. First, the audio signal is broken into several overlapping frames. After this pre-processing each frame is converted to a magnitude spectrum by applying a Fast Fourier Transform (FFT) where the phase spectrum is not considered.

The magnitude coefficients are then converted to the Mel scale. This scale describes the relation between perceived frequency by the human auditory system to its physical frequency. The coefficients are passed through the Mel-filter bank, which is a set of band-pass filters. The conversion of frequencies in a linear scale to a Mel scale can be accomplished with equation 2.1, where f is the physical frequency.

$$Mel(f) = 2595log(1 + \frac{f}{700}) \tag{2.1}$$

The last two steps consist in passing the Mel coefficients through a logarithmic function, and then applying a Discrete Cosine Transform (DCT) to get the cepstral coefficients. The DCT function corresponds to equation 2.2:

$$c(n) = \sum_{m=0}^{M-1} log(s(m))cos\frac{\pi n(m-0.5)}{M}; \qquad n = 0, 1, 2, ..., C-1 \tag{2.2}$$

where M is the total number of filters, $c(n)$ are the cepstral coefficients, and C is the number of MFCC [11].

2.1.4 I-Vectors and X-Vectors

I-vectors and x-vectors have the capability to represent speech utterances in a compact way. They are both fixed-length vectors, regardless of the duration of the speech utterance. However, the extraction

algorithms that create the vectors of the two methods are quite different.

The general idea for extracting i-vectors [12] is that the session and channel dependent supervectors can be modelled as in equation 2.3 where the supervector m comes from the universal Background Model (UBM), T is a matrix where each column spans the subspace of the important speaker information and channel variability, and x is a gaussian distributed variable. The i-vector corresponds to the Maximum A Posteriori (MAP) point estimate of variable x, for each utterance [13] [14].

$$\theta = m + Tx \tag{2.3}$$

The x-vector is extracted from Deep Neural Network (DNN). In speech applications, the input of a DNN is a set of spectral features such as MFCC for a specific frame. The first few layers are frame-level layers, that are followed by a statistics pooling layer. The latter receives the output of the last frame-level layer and computes the mean and standard deviation. The statistics from the pooling layer are concatenated and sent to the next layers. All layers that succeed the statistics pooling, excluding the output layer, can be used to compute the x-vector [15] [16].

2.2 Tools for Feature Extraction

There are several publicly available toolkits for extracting the aforementioned features and many more. Here, a few of them are presented, including OpenSMILE and OpenXBOW.

2.2.1 OpenSMILE

OpenSMILE [17] is an open-source toolkit vastly used by the speech analysis community that allows a fast extraction of an extensive number of audio features. It has more than 6000 features including speech-related acoustic descriptors, such as loudness, MFCC, energy, voice quality features, formants, etc.

This toolkit can be used to extract typical acoustic feature sets, including the Geneva Minimalistic Acoustic Parameter Set (Geneva Minimalistic Acoustic Parameter Set (GeMAPS), that are usable out-of-the-box to guarantee comparable standards to related research. Nonetheless, it is also possible to interconnect different feature extractor components and create custom sets of features.

The GeMAPS is a set of features suggested by Eyben et al. (2015) [18] that was devised with the intent of serving as a baseline set of parameters for several areas of automatic voice analysis.

The choice of parameters was based on three factors: the potential that the acoustic parameter has to detect alterations in the voice during affective processes, the frequency of a parameter and its success rates in previous studies and its theoretical significance.

There are two versions of the acoustic parameter set, the minimalistic and the extended (eGeMAPS). The minimalistic parameter set consists of 62 parameters. These include 18 Low-Level Descriptors (LLD) consisting of parameters related with frequency (pitch, jitter, formants), energy (shimmer, HNR, loudness) and spectral (alpha ratio, HI, spectral slopes, formants relative energy and harmonic differences). To each LLD, means and coefficient variances are applied as functionals, resulting in 36 more features. The remaining consist in loudness functionals, voiced and unvoiced segments functionals and temporal features.

The extended parameter set has a total of 88 parameters. In addition to the LLD in the minimalistic set it has further spectral parameters (MFCC 1 to 4, spectral flux) and frequency related parameters (formant bandwidth). It also has correspondingly more functionals than the minimalistic set.

2.2.2 OpenXBOW

OpenXBOW [19] is an open-source toolkit that generates bag-of-words representations from different modalities' input, including acoustic low-level descriptors (LLD). The bag-of-words model is commonly used in Natural Language Processing (NLP). It is a simplifying representation of a text, that stores the words that are present in it and the number of times the words appear. In this representation the order of the words is not considered.

This principle, has been adopted in the audio classification field, where it is known as bag-of-audio-words (BoAW). BoAW represents audio features, that are extracted from the audio signal, such as MFCC, in 'audio words'. Each word corresponds to a combination of features. After the classifier has been trained, an histogram represents the frequency of the occurrences of each word. This method reduces the complexity of the features.

2.2.3 Kaldi

Kaldi [20] is an open-source toolkit for speech recognition and signal processing. It allows the extraction of standard features such as MFCC and Perceptual Linear Prediction (PLP), but also i-vectors and x-vectors. It supports acoustic modelling for conventional models such as Gaussian Mixture Models (GMM) and is easily extendable to other models. One of the main advantages of Kaldi is that it has complete recipes available for building speech recognition systems that can be used as a baseline for other experiments.

2.3 Nasalisation

Nasalisation is a voice quality that translates in the production of a sound where part of the air exits through the nose instead of the mouth. This phenomenon happens due to the lowering of the velum which opens an airway (velopharyngeal port) to the nasal cavity. In figure 2.1 it is possible to see how the position of the velum differs when producing a nasal or an oral sound.

The great majority of languages have nasal sounds. In Portuguese there are three nasal consonants which are /m/, /n/ and /η/. Regarding vowel nasalisation, there are three broad categories in which they can be divided [21]:

- Coarticulatory nasalisation - corresponds to the nasalisation of a vowel or part of a vowel when it is adjacent to a nasal consonant. This phenomenon is more prevalent when the nasal is positioned after the vowel, i.e., at the end of the syllable (e.g.: ”viagem”), than when it is before the vowel (e.g.: ”comer”) and is present to some extent in almost every language.
- Phonetic nasalisation - vowels that are distinctively nasalised that are not in the context of a nasal consonant. This only appears in 22% of the world’s languages.
- Functional nasalisation - nasality present due to damage in the functionality of the velopharyngeal port. This can occur because of anatomical defects or nervous system damages.

Portuguese is one of the languages that presents phonetic nasalisation. A study by Yuan and Liberman [22], that measures and compares the accuracy of nasality detection at five different vowel positions through time in three different languages: American English, Mandarin Chinese and Brazilian Portuguese, supports this premise. The results of this study show that throughout the first four positions of the vowel, i.e., the beginning and middle of the phoneme enunciation, the accuracies are significantly higher for Portuguese whilst in the last position, i.e., the end of the phoneme enunciation, they are similar for all languages. This is consistent with Portuguese having phonetic nasalisation opposed to only the coarticulatory nasalisation in English and Mandarin.

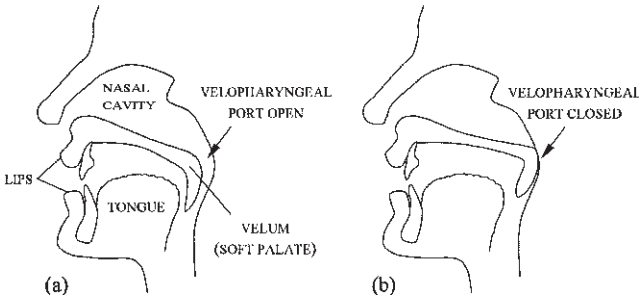


Figure 2.1: (a) nasal sound (b) oral sound (from [1]).

When there is a perceived excessiveness of nasal resonance in speech production, it is considered hypernasality. This can occur due to an inability to control the changes in airflow between the oral and nasal cavities. As velar movement requires dexterity, hypernasality is a typical sign of motor-speech disorders that include Parkinson's Disease (PD), Huntington's Disease (HD), amyotrophic lateral sclerosis (ALS), and cerebellar ataxia. It is also the distinctive perceptual characteristic of speech with a cleft palate [23].

The modelling of the vocal tract, which begins in the glottis and continues to the lips, can be an indicator of nasality. It is comparable to a linear, time-invariant filter and its shape and dimensions are essential for the study of articulatory processes in speech production. In hypernasal speech, the shape of the vocal tract is altered as a consequence of the coupling of the cavities (nasal and oral), the disparity between the two passageways of the nasal cavity, and the sinuses branching from the nasal cavity wall [24].

The detection of hypernasality is a complex task that requires the computation of the ratio of resonances across the mouth cavities. When there is a disproportionately high amount of nasal resonance it is considered hypernasal. This can be a difficult estimation to make since it is dependent on a number of different variables such as word choice and the size and shape of each individual's cavities. It results in a highly nonlinear and complex mapping between the perceived and the actual acoustic nasal resonance [25].

2.4 Acoustic Correlates for Parkinson's Disease

Neurodegenerative disorders affect primarily the brain cells. They are disabling conditions that progressively degenerate nerve cells that when dead cannot be regenerated [26]. There are several symptoms experienced when suffering from one of these diseases, being one of them difficulty in communication, either due to memory loss and forgetfulness or for motor reasons such as dysarthria. Dysarthria is frequently present in PD patients. It is defined as damage, weakness or paralysis in the muscles used to produce speech and can affect respiration, phonation, articulation and prosody [27].

In speech affecting disorders there are certain voice qualities that distinguish a person that suffers from a specific condition from a healthy individual. Here we will see what are these voice qualities for PD and which are the corresponding features that can be used to identify them in the speech utterances.

Even though patients suffering from PD can present impairments in all speech dimensions, several studies show that the speech dimension that is the most correlated to PD patients is phonation, followed by articulation [28] [29]. However, Rusz et al. [30], determined that for early untreated PD, the most predominantly affected speech dimension appears to be prosody.

To examine phonation, the most usual measures are extracted from sustained vowels. Due to ele-

vated laryngeal tension and low laryngeal stability, PD patients have shown to have higher mean and variation of the fundamental frequency, higher jitter and shimmer and a smaller NHR when compared to healthy subjects [31] [32] [33].

Articulation features are most commonly extracted from Diadochokinetic (DDK) tasks which consist in rapid syllable repetitions such as /pa/, /ta/, /ka/. Studies have shown that when examining articulation the stop consonants of PD patients were imprecise and similar to fricatives [30]. These articulatory deficits may result from small range of motion of lips and tongue and also from articulatory weakness [31].

Prosodic features are usually determined from continuous speech, either from reading tasks or spontaneous speech. Patients with PD tend to present a decreased intensity and F0 variability and decreased rhythm when compared to control groups [30] [31] [32].

For PD patients that develop dysarthria, which approximately 90% of patients do as the disease progresses [34], mild to moderate hypernasality is a symptom that has been shown to appear due to reduced control of the nasal cavity [35].

2.5 Machine Learning Classification Algorithms

Classification algorithms are used when the output intended is a discrete label. Since there are too many classification algorithms to cover separately, this section will only include a brief overview of some of the most common classification algorithms such as k-Nearest Neighbours (KNN), GMM, Support Vector Machine (SVM) and Probabilistic Linear Discriminant Analysis (PLDA), followed by an explanation of the deep learning models used in this work.

2.5.1 k-Nearest Neighbours

The main assumption behind the KNN algorithm is that similar things are close to each other. In order to predict the class of a new instance, KNN will find the k training instances closer to that instance and assign the most common label (mode), between those k instances, to the new instance.

2.5.2 Gaussian Mixture Models

GMM are probabilistic models that represent normally distributed classes. A Gaussian Mixture is a function that contains k Gaussians, where k is the number of classes in the dataset. The most commonly used technique to estimate the model's parameters is Expectation Maximisation (EM), which is an iterative algorithm that estimates the maximum likelihood of the parameters. It starts by assigning model parameters based on data and iterates until their estimates converge. When the algorithm finishes it

is possible to assign each data point to the most likely Gaussian, forming clusters that represent each class [36].

2.5.3 Support Vector Machines

SVM are a supervised machine learning method used for classification of both linear and non-linear data that are primarily used for binary classification. This classifier goal is to define the hyperplane that divides the data into classes and has the highest margin. In other words, it finds the plane that provides the maximum distance from the points of the classes to the classification frontier [37]. Given this definition, it is understandable that this is a sparse classifier. Only a small portion of the dataset is considered a support vector, because only the points closer to the hyperplane influence its position.

2.5.4 Probabilistic Linear Discriminant Analysis

PLDA relies on Linear Discriminant Analysis (LDA) which projects features in higher dimension space to lower-dimensional space. Since the goal is to maximise the separability between classes, LDA determines the subspace that presents a larger distance between data of different classes, relative to the spread within each class [38].

PLDA chooses the class for a new input based on the knowledge acquired with the training set (*a posteriori* probability). It bases the values of the covariance matrix, Σ, in the assumption that every class's input has the same matrix, and it models the probability of x belonging to each class as a Multivariate Normal Distribution.

2.5.5 Deep Learning

Deep Learning (DL) is a branch of Machine Learning (ML) that has this designation due to the greater depth of layers in its Neural Networks (NN) that allow a higher abstraction level of the data representation. [39].

Feedforward Neural Networks or Multi-Layer Perceptron (MLP), which are explained more in-depth in section 2.5.5.A, are the basis of deep learning models. They are inspired by neuroscience, since the perceptrons, the basic units of these models, are conceptually modelled from biological neurons [40].

A neural network is composed of several nodes (or perceptrons) that, in turn, are organised in layers of one or more nodes. They usually consist of an input layer, an output layer and a number of hidden layers. The perceptron is the simplest version of a network since it contains only one node.

Nodes receive a vector of features as input. This vector is then multiplied by a vector of weights and a bias value is added. The result of this computation is then passed through an activation function

returning one single value. This process can be described by equation 2.4 where y is the output, ϕ is the activation function, w is the weights, x the input vector and b the bias.

$$y = \phi(w^T x + b) \tag{2.4}$$

2.5.5.A Multi-Layer Perceptron

MLP are, as the name indicates, neural networks that have multiple perceptron units that are arranged in several layers. They have a minimum of three layers: one input layer, one hidden layer and one output layer and are fully-connected, which means that each node from one layer is linked to every node of the previous and next layers, as can be seen in figure 2.2.

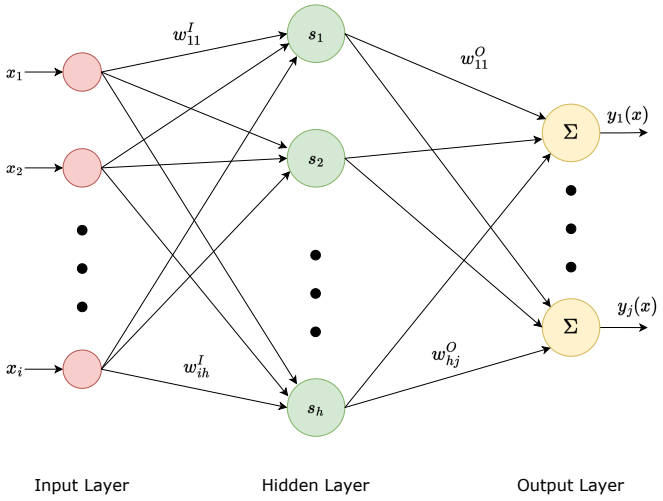


Figure 2.2: Feedforward Neural Network structure.

Excluding the input nodes, each unit is composed of input weights, an activation function and an output. The activation function is applied to the multiplication of the inputs with the corresponding weight in order to return a weighted output.

The weights are updated while the network is training. When the output of a multi-layer perceptron does not correspond to the expected values for a given input vector, an error signal is produced that corresponds to the difference between the desired and obtained output. The weights are then updated according to the magnitude of the error signal in order to reduce the overall error of the MLP [41].

This update can be formally described by equation 2.5 where $E(X, w)$ is the error function where X represents the input-target pairs dataset and w represents the neural network parameters (weights and bias). η is the learning rate that quantifies the rate at which the weights are updated during the training. Note that in the case where the output equals the target value the new weight equals the old weight.

$$w_i^{new} = w_i^{old} - \eta \frac{\partial E(X, w)}{\partial w} \tag{2.5}$$

Backpropagation

The backpropagation algorithm is used to evaluate the derivatives of the error function in order to minimise it. The algorithm works by propagating the errors in a backwards manner along the network, hence the name [42].

There are several functions that can be used as an error function, but the most commonly used in backpropagation is the mean squared error (MSE), which is described in equation 2.6, where $\hat{y}_i$ is the computed output of the network and y_i is the expected target.

$$E(X, w) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{2.6}$$

To calculate the derivative of $E(X, w)$ the algorithm starts by applying the chain rule of calculus to the error function partial derivative, resulting in the following equation:

$$\frac{\partial E}{\partial w_{ij}^k} = \frac{\partial E}{\partial a_j^k} \frac{\partial a_j^k}{\partial w_{ij}^k} \tag{2.7}$$

where a_j^k is the activation of node j in layer k before it is passed to the activation function to generate the output.

The first term of 2.7 is often referred as the error and can be denoted as δ_j^k. The second term can be calculated as:

$$\frac{\partial a_j^k}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} \left(\sum_{l=0}^{r_{k-1}} w_{lj}^k o_l^{k-1} \right) = o_l^{k-1} \tag{2.8}$$

since a is the weighted sum of the inputs for each unit. Thus, the derivative of the error function can be formulated as in equation 2.9 and corresponds to the product between the error term of node j of layer k and the output o of node i of the previous layer.

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_j^k o_i^{k-1} \tag{2.9}$$

Therefore, to evaluate the derivatives it is only needed to calculate the value of δ_j^k for each layer. For the output layer, the error value is straightforward and is given by equation 2.10 where $\phi(x)$ is the activation function.

$$\delta_j^k = \phi'(a_j) \frac{\partial E}{\partial \phi(a)} \tag{2.10}$$

For the hidden layers, the chain rule is used once again and the error term can be decomposed as follows:

$$\delta_j^k = \frac{\partial E}{\partial a_j^k} = \sum_{l=1}^{r_{k+1}} \frac{\partial E}{\partial a_l^{k+1}} \frac{\partial a_l^{k+1}}{\partial a_j^k} \tag{2.11}$$

The first term of the sum function corresponds to δ_l^{k+1} and knowing that a_l^{k+1} is defined as in equation 2.12, it is possible to formulate a final equation for the error term in hidden layers as seen in 2.13.

$$a_l^{k+1} = \sum_{j=1}^{r_k} w_{jl}^{k+1} \phi(a_j^k) \tag{2.12}$$

$$\delta_j^k = \phi'(a_j^k) \sum_{l=1}^{r_{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \tag{2.13}$$

Activation Functions

Applying an activation function on the weighted inputs is the final step before a node gives its output. There are several activation functions that can be used depending on the problem. For regression problems it is most common to use linear functions whereas classification problems tend to use non-linear functions, such as sigmoid, hyperbolic tangent (tanh) and Rectified Linear Unit (ReLU).

The sigmoid function is defined by equation 2.14 and has an s-shape looking curve. The function receives a value x and transforms it into a value between 0 and 1. Therefore it is mostly used for models that predict probabilities, since these only exist in the range [0, 1].

$$\phi = \frac{1}{1 + e^{-x}} \tag{2.14}$$

The tanh function has also an s-shape form and is described in equation 2.15, where σ is the sigmoid function. The hyperbolic tangent receives a value x and transforms it into a value between -1 and 1. The advantage when compared to sigmoid is that the negative inputs will be mapped strongly negative. This function is mostly used for classification between two classes.

$$\phi = tanh(x) = 2\sigma(2x) - 1 \tag{2.15}$$

The ReLU function is currently the most used and can be represented by equation 2.16. This function turns the input to 0 when it has a negative value and keeps its value when it is positive. ReLU is more efficient than other activation functions since the neurons are not all activated at a given moment [43]. When the output of a neuron is zero the neuron is deactivated, making it faster and a better option for networks with many layers.

$$\phi = max(0, x) \tag{2.16}$$

Optimisers

Optimisers are methods that define how the weights, θ, of a neural network should be changed in order to minimise the loss function, $J(\theta)$. In a pure optimisation problem the goal would be to minimise the objective function across the data-generating distribution p_{data}, whereas in machine learning there is only a finite set of samples, the training set $\hat{p}_{data}$. Thus, in a machine learning problem, in order to approximate it to an optimisation one, the objective would be to minimise the expected loss in the training set, so that it can generalise well to unknown data. This can be defined by equation 2.17 where m is the number of training samples, L is the loss function and $f(x;\theta)$ is the predicted output when the input is x.

$$\mathbb{E}_{x,y\ \hat{p}_{data}(x,y)}[L(f(x,\theta),y)] = \frac{1}{m}\sum_{i=1}^{m} L(f(x^{(i)};\theta),y^{(i)}) \tag{2.17}$$

The majority of the optimisation algorithms used for deep learning use only a part of the training dataset for each update of the parameters. This means that during a passage through the whole dataset, which is called an *epoch*, the parameters will be updated several times. These subsets are called minibatch and therefore the methods are known as minibatch methods or stochastic methods [40].

The most used optimisation algorithm is the Stochastic Gradient Descent (SGD) which computes the average of the first order derivative of a loss function (equation 2.18). The weights are then updated as in 2.5 with $\hat{g}$ equating to the derivative of the error function.

$$\hat{g} = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)};\theta),y^{(i)}) \tag{2.18}$$

SGD can take a long time to converge so a technique called momentum was introduced in order to accelerate the convergence to the relevant direction. It also reduces the oscillations and high variance of the parameters. The algorithm for momentum can be seen in equation 2.19 where γ is the momentum term. The update is done as in equation 2.20.

$$V(t) = \gamma V(t-1) + \eta \nabla J(\theta) \tag{2.19}$$

$$w_{new} = w_{old} - V(t) \tag{2.20}$$

There are several algorithms, namely Adagrad, AdaDelta and Adam, that where adapted from SGD to tackle its limitations such as the difficulty in tuning the learning rate. Although they tend to converge

faster than SGD their generalisation performance can be worse in some cases which explains why SGD is still widely used [44].

Adam [45] is usually the best option for fast and efficient neural network training. Its name means adaptive moment estimation, because it estimates both the first and second moments of the gradient to adapt the parameters. The moments of a variable correspond to the expected value of that variable to the power of n, with n being the order of the moment. The equations for the first and second moments of the gradient, g are given in 2.21 and 2.22, where m and v are moving averages and β are hyper-parameters that control the exponential decay rates of m and v.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2.21}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{2.22}$$

Since the moving averages are initialised as vectors of zeros they lead to estimations that are biased towards this value. To solve this a bias-corrected $\hat{m}$ and $\hat{v}$ are estimated as in equations 2.23 and 2.24.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2.23}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{2.24}$$

Finally the weights are updated according to the following equation:

$$w_{new} = w_{old} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2.25}$$

Weight Initialisation

The initialisation of the weights is an important consideration in the design of a neural network with great implications in the convergence of the model. For a long time, the weights were drawn randomly from a Gaussian or uniform distribution. In recent years, more sophisticated methods have been developed that relate the weight initialisation with the activation function in use [46].

The normalised Xavier weight initialisation [47] was initially derived for linear activation functions, but it has become the standard for non-linear functions such as sigmoid and tanh. This method calculates the weights according to a uniform probability distribution in the range shown in 2.26 where n_j and n_{j+1} are, respectively, the number of input and output units in the layer.

$$w \in \left[-\sqrt{\frac{6}{n_j + n_{j+1}}}, \sqrt{\frac{6}{n_j + n_{j+1}}} \right] \tag{2.26}$$

For the ReLU activation function another method was proposed, called the He weight initialisation [48]. This method calculates the weights according to a gaussian probability distribution with a mean value of 0 and a standard deviation of $\sqrt{2/n}$ as seen in 2.27. The performance of the Xavier initialisation has been shown to worsen with the increase in the number of layers of a neural network when using ReLU. In [46], the authors attribute this to the fact that the variance of the inputs is exponentially smaller in the deeper layers than in the shallower layers.

$$w \in G\left(0.0, \sqrt{\frac{2}{n}}\right) \tag{2.27}$$

2.5.5.B Time Delay Neural Network

The Time Delay Neural Network (TDNN) [49] is a multilayer feedforward neural network that first appeared in the late 1980s as an alternative to Hidden Markov Models (HMM) for a phoneme recognition task. This NN is still widely used for acoustic models by speech recognition software, such as Kaldi (2.2.3), with the aim of converting an acoustic speech signal into a sequence of phonemes.

The two main properties that make the TDNN a good fit for speech-related tasks are its ability to learn the temporal structure of acoustic events and their relationships, and the fact that it is shift-invariant, which means that the features learned by the network are independent of their precise location on the input data. Since the TDNN detects phonemes and their underlying acoustic properties regardless of temporal location, it outperforms static classification.

In the TDNN architecture each layer has a different temporal resolution, that increases as we go through the network's layers. The initial transforms of the TDNN are learnt on narrower contexts whilst the deeper layers process the hidden activations from a wider temporal context. This results in an ability to learn wider temporal relationships for the higher layers [50].

As shown in figure 2.3, each node of one layer is dependent on a range of adjacent nodes from the previous layer. The number of nodes is determined by a delay offset $[d1, d2]$ called the context. For each t, $d1$ and $d2$ represent the number of context frames before and after t, respectively. At each time step t, the TDNN performs a convolution operation, which consists on an element-wise multiplication of the kernel weights and input beneath it, followed by the sum of each element multiplication.

Specifically, in the first input layer of the figure, the context is set to $[-2, +2]$, which means that for each t the frames considered are $(t-2, t-1, t, t+1, t+2)$. Each five consecutive frames, serve as inputs to the activation function after being multiplied by a layer-wise shared weight. The outputs of the activation function are then normalised before being fed to the next layer [51].

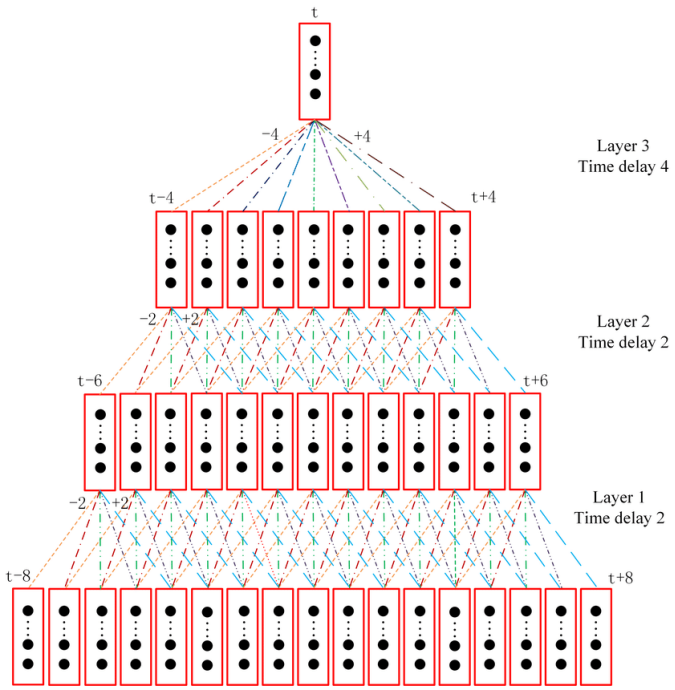


Figure 2.3: Time Delay Neural Network structure.

In the TDNN architecture, for each time step t, the input of every layer is fed forward together with its neighbours according to a previously defined context window. This results in an overlapping between frames in the deeper layers that can lead to a computationally inefficient network. Figure 2.4 demonstrates how sub-sampling may be used to enhance efficiency by passing just the sampled frames (boxes in blue) to the next layers. The parameter that defines the sub-sampling is called stride or dilation. In the case of figure 2.4 the stride is 3, meaning that for each time step t it only feds forward the frames $\{t-3, t, t+3\}$.

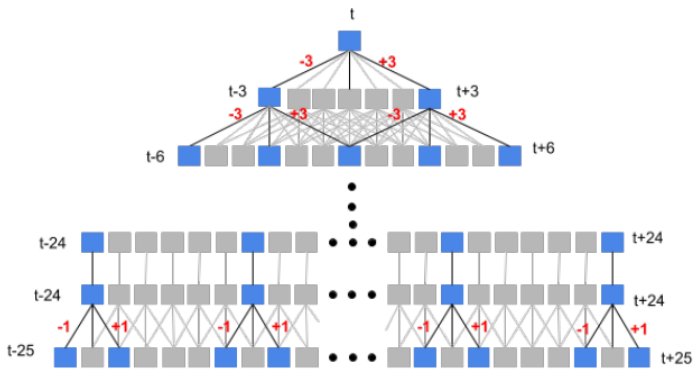


Figure 2.4: TDNN with sub-sampling (from [2]).

2.6 Metrics

In order to evaluate the performance of classification algorithms there are several different metrics that can be used. In this section we will cover a few of them, namely, classification accuracy, true positive rate (TPR) also called sensitivity or recall, true negative rate (TNR) or specificity, F1 score and unweighted average recall (UAR). These metrics are defined in equations 2.28 through 2.32.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.28}$$

$$TPR = \frac{TP}{TP + FN} \tag{2.29}$$

$$TNR = \frac{TN}{TN + FP} \tag{2.30}$$

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{2.31}$$

$$UAR = mean(TPR, TNR) \tag{2.32}$$

The expression that defines precision is presented in equation 2.33 and the definitions of TP, FN, FP and TN are in table 2.1. This table represents a confusion matrix which is also used to describe the performance of a classifier and gives an overview of all the correct and incorrect predictions.

$$Precision = \frac{TP}{TP + FP} \tag{2.33}$$

Table 2.1: Confusion Matrix.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

The metrics for multi-class classification are derived by generalising the previous equations for more than two classes. For instance, the average accuracy can be computed using the formula shown in 2.34 where l is the number of classes.

$$AverageAccuracy = \frac{\Sigma_{i=1}^{l} \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i}}{l} \tag{2.34}$$

The classification accuracy is a good estimator only for a balanced number of samples of each class.

If we have 95% data of class A and only 5% of class B the model can get an accuracy of 95% by predicting that every sample belongs to class A. For this reason, when handling unbalanced data it is best to not rely solely on the accuracy. The information provided by this metric should be complemented with the other mentioned metrics especially the confusion matrix.

3

Proposed Solution

Contents

This chapter starts with an overview of other work that has been done in the study of nasalisation. It also presents the general system that was used to tackle the problem under study. Finally, a characterisation of the datasets used in the development and testing of the system is conducted.

3.1 Related Work

Clinically, hypernasality is commonly assessed with perceptual evaluations done by speech-language pathologists. Alternatively, there are a few instrumental methods that can be used, such as magnetic resonance imaging (MRI), which can be used to observe the velum directly, and the nasometer [25].

The nasometer is the most used instrument in clinical settings and evaluates the extent of nasal escape in cleft palate. It consists on an headset with two microphones separated by a plate that measure the nasal and oral sound pressure levels of speech. These measurements are then used to obtain the acoustic energies and calculate the nasalance which corresponds to the ratio between the nasal and the total (sum of nasal and oral) acoustic energies [52].

These methods all rely on the availability of trained clinicians to manoeuvre the devices and to read and interpret the results. In the last decades, several studies that use speech signal analysis to detect nasality have emerged.

In Kataoka et al. [53], the authors study hypernasality in children by assessing different spectral features in the isolated vowel [i]. Using perceptual ratings as ground truth, they concluded that there is a high correlation between the amplitudes of 1/3-octave bands and the ratings. When compared to a control group, the hypernasal group presented increased amplitudes between the first and second formant (F1 and F2) and decreased amplitudes near F2.

Pruthi et al. [54] proposed a set of acoustic parameters (APs) for the detection of vowel nasalisation. There are nine APs that include A1-P0 and A1-P1, where A1 corresponds to the amplitude of the first formant and P0 and P1 correspond to the amplitude of an extra peak below and above F1, respectively. It also includes nPeaks40dB which counts the number of peaks within 40dB of the maximum dB amplitude in a frame and F1BW which is the bandwidth of F1. The authors achieved accuracies ranging between 69% and 96% for different datasets using a support vector machine as classifier.

A few studies ([55], [52]) used the voice low tone to high tone ratio (VLHR) as a rating for nasality. VLHR corresponds to the ratio between low-frequency power and high-frequency power of the sound power spectrum. This division between low and high is done by a specific cut-off frequency which can differ depending on the task. In Tsai et al. [52], the authors used this method in read speech and obtained significantly greater VLHR scores in the nasal sentences when compared to the non-nasal sentences.

There has also been a number of works relying on machine learning to determine hypernasality

in children with cleft lip and palate. In Orozco-Arroyave et al. [56], the authors present a method that detects if the speech is hypernasal or healthy by using Nonlinear Dynamics (NLD) features and entropy measurements as input features to an SVM. They tested their system with a German and a Spanish databases that contained recordings of healthy and hypernasal children and found that the combination of NLD features and entropy measurements provided the best results.

In Wang et al. [24], the authors describe a feature-independent algorithm that uses a Convolutional Neural Network (CNN) to detect hypernasality. The input of the CNN is a speech spectrogram and the dataset used is comprised of Chinese children and adults with cleft palate. The authors achieve accuracies of 93% and infer that /i/ is the most sensitive vowel to hypernasality.

Another study by the same authors [57], obtains similar accuracy results with a different architecture: Long Short-Term Memory (LSTM) based Deep Recurrent Neural Network (DRNN). In this study, the authors analyse hypernasality-sensitive vowels in Mandarin (/a/, /i/ and /u/) and conclude that vowels /i/ and /u/ are the most sensitive vowels to hypernasal speech, which aligns with the results from their other study.

3.2 System Overview

The main objective of the system implemented is to predict hypernasality in pathological speech for European Portuguese (EP). The method used to achieve this is based on the proposed method by Mathad et al. in [58], where the authors conducted a similar study for the English language with patients suffering from PD, HD, ALS and cerebellar ataxia. An overview of the system developed can be seen in figure 3.1.

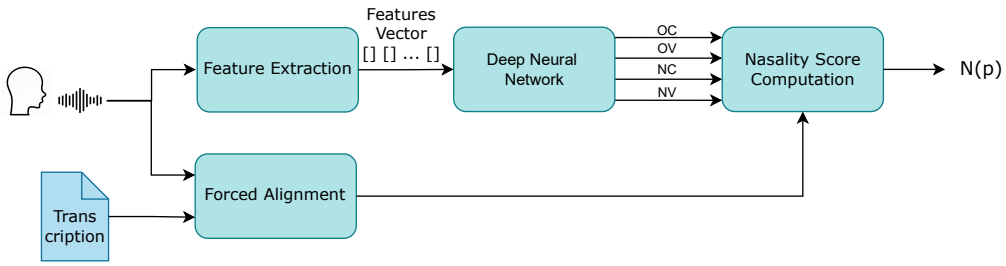


Figure 3.1: Overview of the proposed system.

One of the major challenges of using speech processing tools for clinical applications is the scarcity of data for the disorders. The key benefit of the approach that we will be employing over other works using machine learning is that the training data is composed of recordings from healthy individuals.

This aspect makes the solution disease-independent and also makes it easier to find datasets with a more reasonable size. Even though EP does not have the same amount of resources available as other languages, such as English or Mandarin, there are clearly more and larger datasets of healthy speech than there are of individuals with hypernasality.

The inputs of the system are an audio file of continuous speech and the text transcription of that file. The corpus used is characterised below in section 3.3. The final output consists of a nasality score which is a value comprised between 0 and 1. The system as a whole consists of four main blocks that will be individually analysed below.

Feature Extraction

The features extracted from the speech file are MFCC and their first and second order derivatives (Δ and $\Delta\Delta$MFCC). There are a total of 39 features for each frame: 13 MFCC, from coefficients 0^{th} to 12^{th}, and the same number of deltas and delta-deltas.

Considering that speech is a nonstationary random signal, that can only be assumed to be stationary in short-time intervals of 10 to 30 ms, Hamming windows are used to split the signal into frames of 20 ms each with overlap of 10 ms.

The reason for using MFCC as features is that they have the ability to model both the vocal chords and the vocal tract. When presented with disordered speech, FFT-based MFCC have the intrinsic capacity to represent either an irregular movement of the vocal chords or a lack of closure caused by compensatory movements in the vocal tract as a result of velopharyngeal incompetence [59].

The delta (differential) and delta-delta (acceleration) coefficients are used in order to better understand the speech signal. They provide an insight into the dynamics of the power spectrum, i.e., the trajectories of MFCC over time. The delta coefficients are computed according to the equation 3.1:

$$d_t = \frac{\Sigma_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2\Sigma_{\theta=1}^{\Theta} \theta^2} \tag{3.1}$$

where d_t is the delta coefficient from frame t computed in terms of the correspondent MFCC $c_{t-\theta}$ to $c_{t+\theta}$ and Θ is the size of the delta window. The acceleration coefficients are calculated in the same way, except they use the differential instead of the static coefficients [60].

Forced Alignment

Forced alignment is a special mode of automatic speech recognition. In Automatic Speech Recognition (ASR) the only input is the audio file and the system needs to predict the words that were said based on the acoustic features and the language model. When doing forced alignment ASR, there is one other

input which is a written transcript of the audio. In this mode, the system only needs to evaluate where each word or phoneme begins and ends.

This block was performed using TRIBUS [61] which is an ASR system for European Portuguese developed at INESC-ID. This system returns the alignment of the audio with the transcript at a phoneme-level. Forced alignment was performed on the test data so that when the system is calculating the nasality score, it knows which phoneme corresponds to each point in time.

Deep Neural Network

After the forced alignment is performed, the audio files go through a neural network. The NN was developed to identify phonemes as oral or nasal. For this purpose, it was trained to classify each frame as one of the five following classes: oral consonants (OC), oral vowels (OV), nasal consonants (NC), nasal vowels (NV) and silence.

The NN was trained on an healthy speech database called BD-PUBLICO that is described in more depth in 3.3.1. The idea explored here is that, when compared to healthy speech, pathological speech will present a higher rate of nasality, $P(NC)$ or $P(NV)$, in oral sounds.

Three different neural network architectures were developed in order to see which one could achieve the best performance. All of them will be described in chapter 4.

Nasality Score

The output of the NN is used to calculate the nasality score. Since the last layer of the NN is a softmax, its output can be considered as the probability of each class. Hence, for each input frame x, the output of the NN will be $P(OC|x)$, $P(OV|x)$, $P(NC|x)$, $P(NV|x)$, $P(sil|x)$, which corresponds to the probability of the input being an oral consonant, oral vowel, nasal consonant, nasal vowel and silence, respectively. The nasality score, $N(p)$, is computed based on these probabilities and the phonemes provided by the transcript. For each oral phoneme, p, a score is computed by averaging the corresponding nasal probability, i.e., $P(NC)$ for oral consonants and $P(NV)$ for oral vowels, of all the frames that belong to phoneme p. A formal equation of how $N(p)$ is calculated can be seen in 3.2 where x_i is the i^{th} frame in phoneme p_j and $|X^{p_j}|$ the total number of frames of the phoneme.

$$N(p_j) = \begin{cases} \sum_i P(NC|x_i)/|X^{p_j}|, if p_j \in OC \\ \sum_i P(NV|x_i)/|X^{p_j}|, if p_j \in OV \end{cases} . \tag{3.2}$$

For hypernasal speech, $P(NC|x)$ or $P(NV|x)$, when x is an oral phoneme might be, in some instances, higher than $P(OC|x)$ or $P(OV|x)$, respectively. This means that this phoneme would be classified as nasal instead of oral. By force aligning the transcript with the audio file and using that as a

reference for which phonemes are said at each point in time, we can use the nasality score of those hypernasal consonants and vowels.

Finally, in order to have one single score per audio file, the nasality scores of the upper quartile, i.e., the 25% highest nasality scores per phoneme are averaged.

3.3 Corpora

Two different corpora were used in the execution of this work: BD-PUBLICO and FraLusoPark. BD-PUBLICO was used to train the neural networks and test its performance, whereas FraLusoPark was used to evaluate the overall performance of the whole system for the task in hand. All corpora are characterised below.

3.3.1 BD-PUBLICO

BD-PUBLICO [62] is a healthy speech corpus which was owned by INESC-ID for automatic speech recognition purposes. This database consists of read speech from the Portuguese newspaper Público. It was created in 1997 and contains recordings from 120 different speakers, with ages ranging from 19 to 28 years old. The corpus is already divided into training, validation and test sets. The training set contains around 23 hours with 100 speakers (50 male and 50 female) and, the validation and test sets contain 2 hours and 10 speakers each (5 male and 5 female). All utterances are sampled at 16kHz. Table 3.1 presents the size of each subset.

Table 3.1: BD-PUBLICO speech corpus.

	# utters	**# speakers**	**# hours**
train set	8388	100	23
validation set	584	10	2
test set	592	10	2

Since this corpus was not created for the specific purpose of studying nasalisation, the read speech does not have particular emphasis on sentences with large amounts of nasal sounds. Thus, the dataset is naturally unbalanced with oral phonemes being the great majority. Table 3.2 shows the distribution of frames per class in the training set.

Table 3.2: Training set frames per class.

oral cons	**oral vowels**	**nasal cons**	**nasal vowels**
2972190	2512535	313440	607554

3.3.2 FraLusoPark Corpus

The FraLusoPark corpus [63] is a Parkinson's speech database that consists of audio recordings of Portuguese and French PD patients speaking in their respective languages. The EP subset is composed of 120 different speakers, 60 PD patients and 60 controls that are age-matched and gender-matched, with ages ranging from 35 to 85 years old. The PD patients are at various stages of the disease. They can be divided into three subsets: *early stage*, for patients that have had the disease for 0 to 3 years and no motor fluctuations; *medium stage*, for patients with 4 to 9 years of disease or between 0 and 3 years and experiencing motor fluctuations; and *advanced stage* for patients with more than 10 years of disease duration. Each of these subsets has a total of 20 individuals.

All participants were recorded doing several tasks, including steady vowel /a/ phonation, maximum phonation time, oral diadochokinesia and several continuous speech tasks. For the purpose of this work, the task that is relevant is a reading task of a short text called "The North Wind and the Sun" (European Portuguese adaptation), given its similarity to the reading task in BD-PUBLICO. The PD patients undertook this assessment in two distinct occasions: (1) at least 12 hours after withdrawal of all anti-Parkinsonian drugs and (2) following at least 1 hour after the administration of the usual medication.

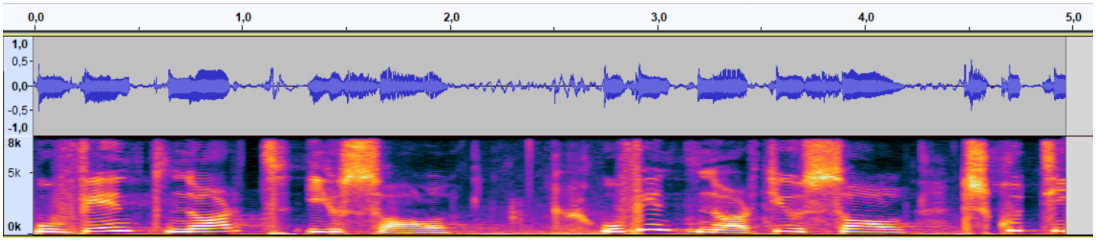
This dataset is not divided in training, validation and test sets and the total duration of the EP subset is around 4.5 hours. All utterances are sampled at 16kHz. The size of the set can be seen in table 3.3.

Table 3.3: FraLusoPark speech corpus.

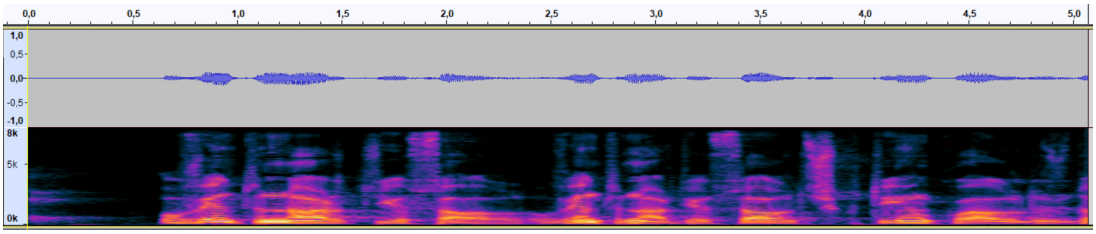
	# utters	**# speakers**	**# minutes**
control	110	60	80
G1	91	20	68
G2	85	20	65
G3	69	20	55
total PD	245	60	188
total	355	120	268

Figure 3.2 shows 5 seconds snippets of the spectrograms of two audio files that correspond to the reading task of the FraLusoPark corpus. The first one belongs to a healthy control (3.2a) and the second one belongs to a PD patient (3.2b).

The spectrogram represents the spectrum of frequencies through time. It is presented as a heat map, where the x-axis represents time, the y-axis frequency, and the colour of each point in the image reflects the amplitude of that frequency. There are a few differences between the two spectrograms that can be seen. The healthy control's voice spectrum exhibits a variation range of harmonics between (0, 8000)Hz. In the PD patient spectrogram this range varies between (0, 4000)Hz and the speech energy is primarily located close to the fundamental frequency as well as in the low and middle frequency regions.



(a) Healthy individual



(b) PD patient

Figure 3.2: Spectrograms of the beginning of the reading task ”The North Wind and the Sun”

4

Implementation

Contents

In this chapter the process conducted for training and testing the neural networks will be explained. Furthermore, the different architectures that were built for the neural network will be presented.

4.1 Models

As previously stated, three distinct neural network architectures were implemented: a feedforward neural network, a hierarchical neural network, and a time-delay neural network. Each of them will be discussed in detail in the sections that follow.

All of the models underwent a similar training and testing process, that can be seen in the schematic in figure 4.1. In order to obtain the labels, forced alignment was performed on the training dataset for the neural networks. The networks were trained using supervised learning, which means that the inputs need to be labelled in order for the model to learn the relationships between the inputs and the output data.

During the training phase, features were extracted from the data and afterwards were fed into the model along with the labels. The model was trained several times and the best performing models were saved. In the test phase, the features were inputted into the stored model, which, depending on the operation mode, can provide either the accuracy and loss of the model for that dataset (evaluation mode) or the probabilities of each class being the correct label (prediction mode).

The data used for the training phase were the training and validation sets of BD-Público (3.3.1), with the validation set being used to fine-tune the parameters presented below and to prevent the network from overfitting. For the test phase we used the test set of BD-Público, as well as the control subjects of the FraLusoPark corpus (3.3.2) to see how well the model adapts to recordings produced in different conditions from those of the training data.

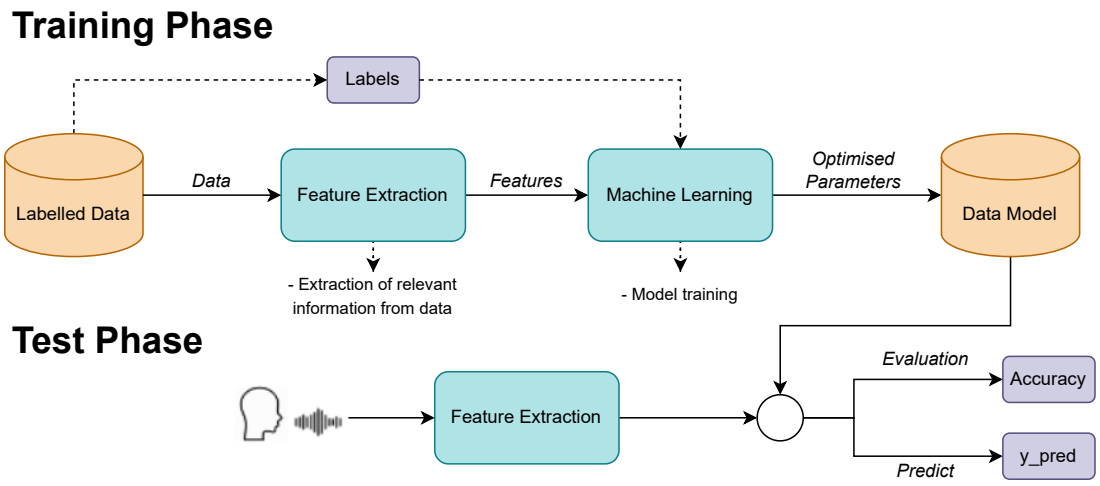


Figure 4.1: Training and testing procedures scheme.

All the models were trained using the Adam optimiser [45] and the He method [48] as a weight initialiser. The loss function chosen was the cross entropy loss function which measures the performance of both binary and multi-class models that have as output a probability between 0 and 1. For each of the models created we did an ablation study of the following parameters:

- **Learning rate** - The learning rate is a hyperparameter that defines the extent to which the model is modified in response to the predicted error every time the model weights are updated. The selection of the learning rate is challenging, as a value that is too low can lead to a long training process that gets stuck, while a number that is too high may result in the learning of a sub-optimal weight set or an unstable training process.

- **Batch size** - The batch size consists in the number of samples (in this case frames of speech) that are used to train a model before updating its internal parameters, such as biases and weights. It can be one of three options: batch gradient descent, where the batch size equals the size of the training set, in which case there is only one batch per epoch; stochastic gradient descent, where the batch size is equal to 1; and mini-batch gradient descent, where it takes on any value between the first two options. Very high batch sizes may result in poor generalisation, which means the neural network will perform poorly outside of the training set. Smaller batch sizes have been shown to converge faster to good solutions, however it is not always the global optima. The main idea is to test different batch sizes until a configuration that is appropriate for the particular neural network and dataset is discovered. The most common mini-batch sizes used are powers of 2 since these values take the most advantage of the CPU/GPUs processing.

- **Epochs** - An epoch is a hyperparameter that defines the amount of times that the learning algorithm loops through the entire dataset. Each sample in the training dataset has the chance to adjust the internal model parameters once per epoch. When training a model, it is important to find the right number of epochs because with the rise in this number, the model weights are adjusted more times, and the curve progresses from underfitting to optimal to overfitting.

- **Dropout** - Dropout is a technique used during the training of DNNs that have a large number of parameters with the goal of avoiding overfitting. The idea is to drop a portion of the units and their connections at each step of the training. Dropout prevents units from co-adapting excessively. Samples from an exponential range of several ”thinned” networks are used during training. By utilising a single unthinned network with a decreased weight at test time, it is simple to replicate the impact of averaging the predictions of all these thinned networks. This provides considerable gains over conventional regularisation techniques and reduces overfitting [64].

4.1.1 FeedForward Neural Network

The first model we trained was a feedforward neural network (2.5.5.A), whose topology is depicted in figure 4.2. It consists of 5 layers with dropout being applied between each layer. The hidden layers have a decreasing number of nodes, with the first having 1024 nodes, the second 512 nodes and the third 256 nodes. Before the output layer batch normalisation is applied. The different layers of this architecture are described below.

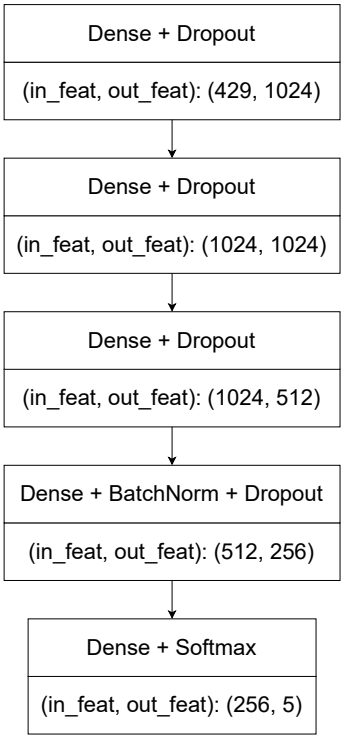


Figure 4.2: FeedForward Neural Network architecture.

- **Input Layer** - For each input frame there is a context applied at the entrance of the neural network. This means that we use the adjacent frames in time to enhance the number of features. In this particular case, the context is 11 frames, five preceding frames of the current input and five subsequent ($[t-5, t+5]$). Given that each frame has 39 features, the first layer has an input vector with a total of 39 x 11, or 429 features. Also, in order to avoid border effect, the first and last 5 frames are truncated. This results in an input shape of $(W-10)$ x 429 where W is the total number of frames.

- **Dense** - All the layers in this architecture are dense layers which consist of regular fully-connected layers that apply the computation shown in 2.4. The hidden layers all have a ReLU activation

function.

- **Batch Normalisation** - This method was introduced in [65] and has proved to increase the speed and stability of DNN training. Even though, the benefits of BatchNorm are widely accepted and recognised, the reasons for these improvements are still a cause for debate.

  The reasoning of the authors that introduced this method was that it reduced the *internal covariate shift*. This is defined as the changes that occur to the distribution of a layer's activations as a result of the training-induced modifications made to the network parameters in the previous layer. This change is thought to have a negative impact on the training process since it causes a constant shift of the training problem. More recent studies have related the success of BatchNorm to the smoothness of the optimisation landscape [66].

  BatchNorm resorts to the concept of normalising the input data to zero-mean and constant standard deviation, which has been known for decades to be beneficial to neural network training, and applies it to the intermediate layers [67].

  In order to minimise the cost of using batch normalisation, there are two simplifications that are made. The first is that each scalar feature is normalised independently and the second is that the normalisation is made per mini-batch and not with the whole training set. The normalisation step is presented in 4.1 where $E[x]$ represents the mean, $Var[x]$ the variance and ϵ a constant that is added for numerical stability.

$$\hat{x} = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \tag{4.1}$$

  There is no dispute among the machine learning community that BatchNorm improves training speed, allows greater learning rates, and enhances generalisation accuracy. By normalising activations throughout the network, it keeps the little changes to the layer parameters from escalating as the data is propagated through the DNN.

- **Output Layer** - This layer is also a dense layer, with the distinction that its activation function is softmax rather than ReLU. This provides us with the probabilities of each frame belonging to each class. The shape of the output is $(W - 10)$ x 5 where W is the total number of frames.

4.1.2 Hierarchical Neural Network

The second model trained was a hierarchical neural network. This architecture consists of a sequence of neural networks that form an acyclic graph. In this particular case, a binary tree.

Our model consists of three neural networks that have the same structure as the feedforward neural

network presented in 4.1.1. The main difference is that instead of having a single multi-class classifier, it divides the task into three smaller problems and uses one binary classifier for each of the sub-tasks.

As can be seen in figure 4.3, the first DNN divides each feature vector into consonants or vowels, while the remaining two split the consonants and vowels, respectively, into nasal or oral, resulting in the final labels: oral consonant, nasal consonant, oral vowel and nasal vowel.

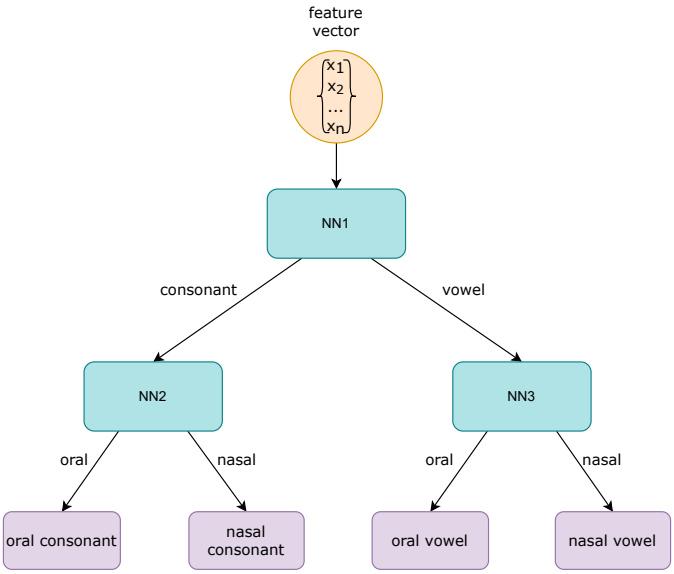


Figure 4.3: Hierarchical Neural Network scheme.

4.1.3 Time-Delay Neural Network

Figure 4.4 depicts the architecture of the final model trained, a time-delay neural network (2.5.5.B). All of the layers in this network are TDNN layers, which means that context is applied to each layer. We used 3 as the context size for all layers, with the exception of the output layer where this value corresponds to 1. Consequently, and given that each frame has 39 features, the first layer's input vector contains 39 x 3, or 117 features. As shown in the figure, the number of input features for each hidden layer is equal to three times the number of output features of the preceding layer. The first two hidden layers contain 1024 output nodes, whereas the latter two have 512 output nodes each. The output layer is also a TDNN layer with the difference that it uses softmax as the activation function. Given that the dilation is 1 for all layers, for each time step t the layers are passing frames $[t-1, t+1]$.

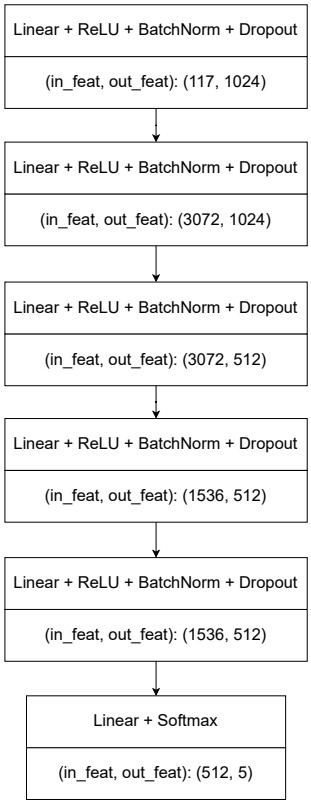


Figure 4.4: Time Delay Neural Network architecture.

A standard TDNN layer architecture can be seen in figure 4.5. It consists of a one dimension convolution, followed by a ReLU activation function and batch normalisation. In our model dropout is also applied after the activation function. The input layer and the hidden layers all follow this layout. As stated above, the output layer does have a different activation function and does not have batch normalisation or dropout.

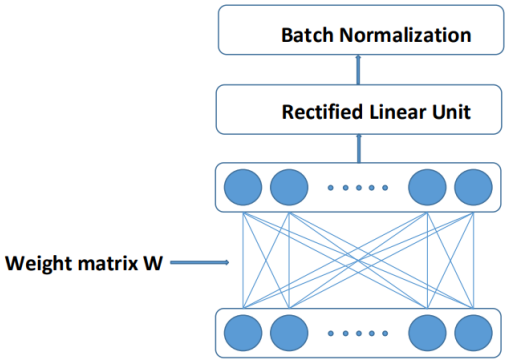


Figure 4.5: TDNN Layer (from [2]).

5

Results

Contents

This chapter begins with an analysis of the training results for each neural network. It also shows and compares the results of the nasalisation score for healthy controls and PD patients. Moreover, it highlights other noteworthy findings.

5.1 Comparison of Phoneme Models

5.1.1 FeedForward Neural Network

Since the feedforward neural network was the first to be trained, each hyperparameter was fine-tuned independently, i.e., only one parameter was adjusted for each training run. This allowed for a better understanding of each parameter's importance within the network. The most significant results obtained can be seen in the tables that follow.

Table 5.1: Batch size fine-tuning results.

	Train		Test	
Batch size	**Accuracy (%)**	**Loss**	**Accuracy (%)**	**Loss**
32	81.53	0.4828	82.04	0.4499
64	**81.52**	**0.4833**	**82.72**	**0.4424**
128	81.57	0.4817	82.52	0.4624

Table 5.2: Epochs fine-tuning results.

	Train		Test	
Epochs	**Accuracy (%)**	**Loss**	**Accuracy (%)**	**Loss**
20	**82.48**	**0.4517**	**83.15**	**0.4291**
30	82.76	0.4437	83.11	0.8544
50	83.09	0.4352	83.10	0.4312

Table 5.3: Learning Rate fine-tuning results.

	Train		Test	
Learning Rate	**Accuracy (%)**	**Loss**	**Accuracy (%)**	**Loss**
0.01	81.52	0.4833	82.72	0.4424
0.001	82.48	0.4517	83.15	0.4291
0.0001	**82.9**	**0.4395**	**83.2**	**0.4286**

Table 5.4: Dropout fine-tuning results.

	Train		Test	
Dropout	**Accuracy (%)**	**Loss**	**Accuracy (%)**	**Loss**
0.2	**82.48**	**0.4517**	**83.15**	**0.4291**
0.3	81.62	0.4764	82.94	0.4377
0.5	80.00	0.5239	82.32	0.4622

The results obtained for the accuracies and losses on the train and test sets show that the best combination of parameters is a batch of 64 utterances, a learning rate of 0.0001, a dropout of 0.2 (or 20%) and 20 epochs. With this combination, we achieve an accuracy of 83.2% and the confusion matrix presented in figure 5.1 for the BD-Publico test set. Even though for some of the parameters (batch size and number of epochs) the training accuracies were higher for the larger values, the test accuracies were lower which may indicate that the model was starting to overfit to the training dataset.

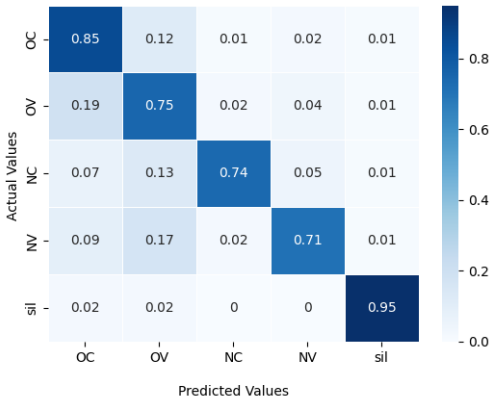


Figure 5.1: Feedforward NN confusion matrix: OC - oral consonant, OV - oral vowel, NC - nasal consonant, NV - nasal vowel.

5.1.2 Hierarchical Neural Network

For the hierarchical neural network, we used the same parameters that produced the best results for the feedforward neural network, and experimented with varying the number of hidden nodes per layer. Four distinct models were developed, and the number of nodes for each dense layer of each model is shown in figure 5.2.

Since the hierarchical neural network consists of three different NN, different combinations of the three models presented in figure 5.2 were tried. The results can be seen in table 5.5, where the first column indicates which models were used for each of the NN, i.e., how many nodes each dense layer had. The numbering of the NN in this table is the same as in figure 4.3: NN1 corresponds to the first NN

that divides phonemes into consonants and vowels, NN2 divides consonants into oral or nasal and NN3 divides vowels into oral or nasal.

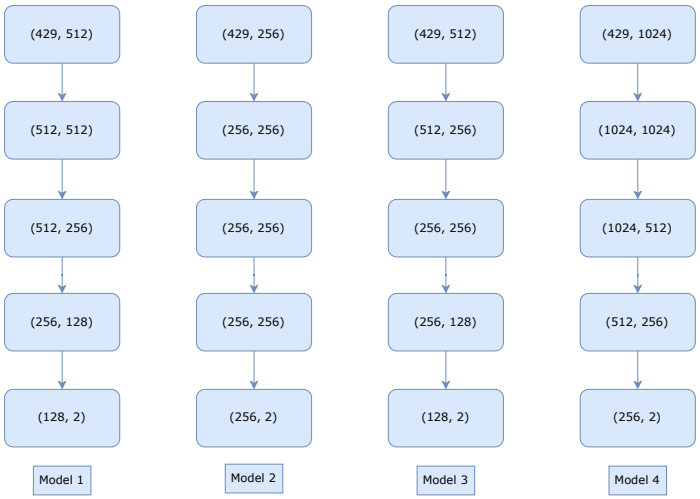


Figure 5.2: Number of input and output nodes per layer for each model.

Table 5.5: Hierarchical neural network results.

	Train			Test
Layers (NN1 -> NN2 -> NN3)	**Acc. NN1 (%)**	**Acc. NN2 (%)**	**Acc. NN3 (%)**	**Accuracy (%)**
1 -> 1 -> 1	82.24	97.51	92.29	77.04
2 -> 2 -> 2	80.73	95.76	91.08	75.78
1 -> 2 -> 2	82.84	98.22	93.76	77.99
3 -> 2 -> 2	82.67	97.95	93.03	77.72
4 -> 4 -> 4	**83.84**	**98.31**	**94.38**	**78.49**

The network that has the worst performance is NN1. Even though the results for the two networks that classify phonemes as oral or nasal are both high (above 90%), there is a 4-5% discrepancy between the consonant classifier (NN2) and the vowel classifier (NN3) in each iteration. This indicates that the network has a harder time differentiating nasal vowels from oral vowels than nasal consonants from oral consonants.

The architecture that achieved the best results is the one that uses the fourth model for all the networks. It obtained an accuracy of 78.49% in the test set and the confusion matrix presented in figure 5.3.

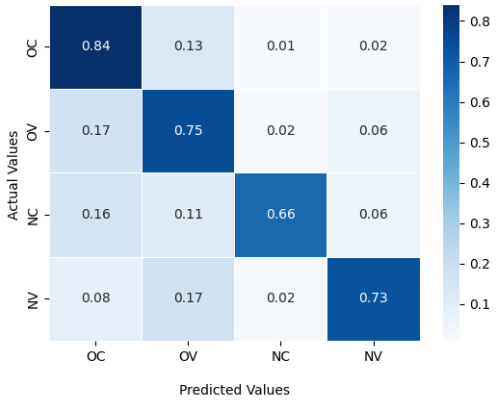


Figure 5.3: Hierarchical NN confusion matrix: OC - oral consonant, OV - oral vowel, NC - nasal consonant, NV - nasal vowel.

5.1.3 Time Delay Neural Network

For the TDNN, since it is a much more memory costly model, the batch size had to be small in order to guarantee that the model could run without surpassing the available memory. For this reason, the batch size used was 16. The remaining parameters were fine-tuned and some of the results can be found in table 5.6. These results show that the configurations that achieved the best results had learning rate and dropout values of 0.001 and 0.3, respectively.

Table 5.6: TDNN fine-tuning results.

Parameters				Train		Test	
Batch Size	**Epochs**	**Learning Rate**	**Dropout**	**Accuracy (%)**	**Loss**	**Accuracy (%)**	**Loss**
16	15	0.01	0.3	83.57	9.17e-3	84.00	2.31e-5
16	25	0.01	0.3	84.06	9.12e-3	84.05	2.30e-5
16	25	0.01	0.2	83.57	9.17e-3	84.00	2.31e-5
16	25	0.001	0.2	85.73	8.98e-3	84.15	2.30e-5
16	35	0.001	0.2	86.53	8.90e-3	84.16	2.30e-5
16	**35**	**0.001**	**0.3**	**85.61**	**8.99e-3**	**84.42**	**2.30e-5**
16	35	0.0001	0.3	85.06	9.03e-3	84.13	2.30e-5

Figure 5.4 shows the effect that the number of epochs has on the training and validation sets of BD-Publico. The training accuracy continues growing for the whole 100 epochs. However the validation accuracy stabilises after a few dozen epochs which indicates that training for a very high number of epochs would only get the model to fit better with the training data and would not bring any benefits to other data (it would not generalise better). This is also visible in table 5.6: with the increase in number of epochs the train accuracy goes up but the test accuracy does not have a significant change ($\pm 0.05\%$).

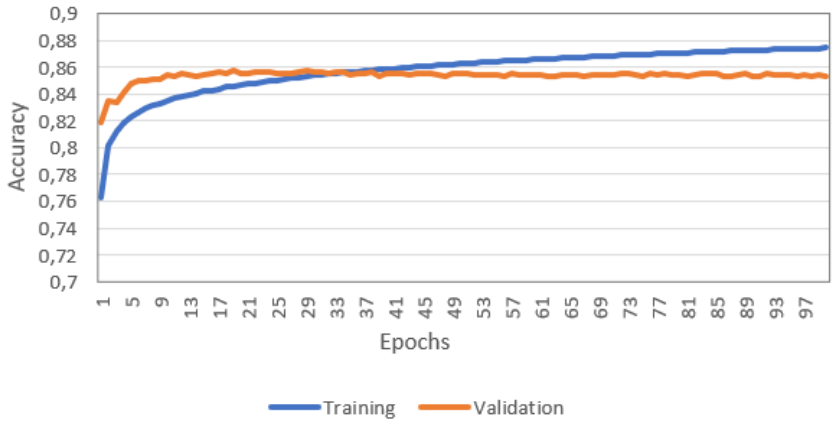


Figure 5.4: TDNN training and evaluation accuracies over 100 epochs.

Overall, the best iteration of the TDNN model had the following parameters: batch size of 16, 35 epochs, a learning rate of 0.001 and a dropout of 30%. This model achieved a test accuracy of 84.42% for the BD-Publico test set and a confusion matrix that can be seen in figure 5.5.

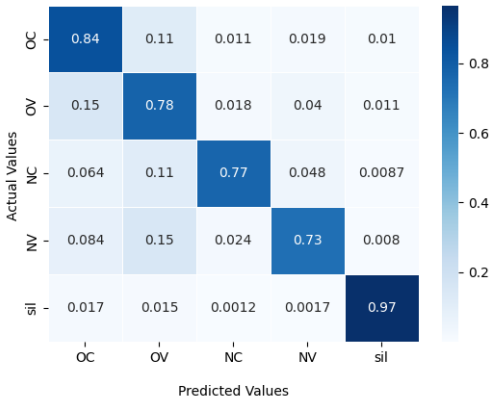


Figure 5.5: TDNN confusion matrix: OC - oral consonant, OV - oral vowel, NC - nasal consonant, NV - nasal vowel, sil - silence.

5.1.4 Networks Comparison

The hierarchical neural network only has four output classes, whereas the other two networks each have five (the same four plus the silence class). This is due to the fact that during the first round of training experiments, we eliminated the frames corresponding to silences. With the development of this work we determined that it would be beneficial to include the silences, since they are always present in the audio files.

For this reason, the results of the hierarchical neural network cannot be directly compared with the results of the other two NN. However, the silence class was not included in the feedforward neural network's first training iterations. These outcomes are not discussed in this report, although they were identical to those achieved by the hierarchical neural network. Due to the fact that the latter did not provide any advances in comparison to the former, along with limited time and resources, it was decided to concentrate on improving the feedforward neural network and developing the TDNN to determine which of these two achieved better results.

Table 5.7 displays the best outcomes achieved with these two neural network. Since the TDNN obtained the best results, this model was selected to perform the remaining tests that are described next.

Table 5.7: Neural Networks' best results.

Model	Accuracy (%)	Loss
FeedForward	83.2	0.4286
Time Delay	**84.42**	**2.30e-5**

5.2 Parkinson's Results

In order to understand how the neural network adapted to the FraLusoPark corpus recordings we ran an evaluation using only the controls of this corpus. The accuracy for this subset was 77.48%. This loss in accuracy, when compared to the BD-Publico dataset, may be explained by factors such as the difference in recording conditions.

The next analysis consisted in running the whole system as explained in 3.2. The results for each of the groups can be seen in figure 5.6. The figure shows a boxplot that represents the distribution in the nasality score per quartile for each group. The yellow lines represent the medians and the green triangle the mean values. Both of these ratios increase visibly from group to group ($Controls < G1 < G2 < G3$) with groups $G1$ and $G2$ having the lowest discrepancy.

In order to validate these results we conducted an Analysis of Variance (ANOVA) followed by a *post-hoc* analysis to determine which groups differ from each other. The Tamhane *post-hoc* test was chosen since it is insensitive to unequal variances between groups [68].

The ANOVA results showed that there are differences between the groups, $F(3, 349) = 13.58, p < 0.001$. The *post-hoc* analysis (table 5.8) shows that, if we consider $p < 0.05$, there are significant differences between the control group and all the Parkinson's groups and there are also significant differences between the group in early stages of Parkinson's and the group in advanced stages.

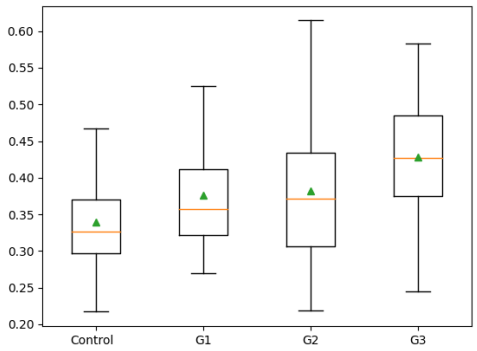


Figure 5.6: Boxplot of nasality score for Parkinson: G1 - early stage, G2 - medium stage, G3 - advanced stage.

It is also important to note that groups 2 and 3 have virtually no differences between them, with a p-value of 0.978 indicating this fact. This suggests that nasalisation is as prevalent in Parkinson's patients in medium stages as it is in advanced stages.

This is in accordance with the literature. In [69], the authors do an assessment of speech and voice quality in PD patients and mention two surveys, each done on 200 PD patients, that are based on perceptual ratings. These studies found that voice abnormalities are often noticed even in the early stages of Parkinson's disease and are present in almost all patients towards the late stages. In the later stages of PD, articulatory impairments, including hypernasality, become more apparent.

Table 5.8: Results of Tamhane *post-hoc* tests.

Group (I)	Group (J)	Significance	Cohen's *d*
Control	G1	.003	0.391
	G2	$<$ .001	0.723
	G3	$<$ .001	0.866
G1	G2	.298	0.332
	G3	.014	0.475
G2	G3	.978	0.143

In Mathad et al. [58], the authors had 14 speech-language pathologists rank hypernasality present in the audios on a scale from 1 to 7. These perceptual rankings were then compared with the results from their system. In our work, due to a lack of time and resources, we were not able to do perceptual rating tests with trained clinicians. However, an informal test was conducted with two (non-clinician) researchers from INESC-ID. This test revealed that when doing a perceptual evaluation of the audios of PD patients, there were other affected characteristics of the speech that were more prominent than nasalisation and, consequently, more discernible in the audio.

Nevertheless, the effect sizes presented in table 5.8, showed small (differences between control

and $G1$, $G1$ and $G2$) to large effect sizes (difference between control group and $G3$). The convention standardised by Cohen for the effect sizes measures stated that $d = 0.2, 0.5$ and 0.8 correspond to small, medium and large effects, respectively [70]. Also, aggregating the three groups with PD into one group and comparing it with the control group, resulted in an effect size in the moderate to large range, $t(351) = 5.51, p < .001, d = 0.63$.

6

Conclusions

Contents

This final chapter discusses the conclusions reached throughout the development of this work as well as some suggestions for future work.

6.1 Conclusions

The first goal proposed for this thesis was to create a deep-learning based system that could discriminate nasal sounds from non-nasal sounds in European Portuguese. For this end, three different architectures for neural networks were developed: a feedforward neural network, a hierarchical neural network and a time-delay neural network.

Creating deep learning models has some challenges including the need for large amounts of data to train the models and the need for good GPU with a considerable memory size. Despite these limitations, we were able to create models that achieved good results.

The TDNN was the neural network that achieved the highest level of accuracy in the BD-Publico test set, with a score of 84.42%. The success of this design in the field of speech processing may be attributed, in large part, to two different factors:

- shift-invariance denotes that the classifier does not need explicit segmentation before classification. When it comes to speech, this is a very favourable attribute to possess since spoken sounds are hardly ever of regular duration, and accurate segmentation is notoriously difficult to acquire.

- context - Every speech unit, at each layer of the network, has a context window that includes both the outputs from the previous layer and the time-delayed outputs of these units. This models the temporal pattern of the units. The other two NN only have context added to the input layer.

The second goal of this thesis was to assess if this model would be able to detect hypernasality in PD patients. For this end, we used the softmax probabilities that were the output of the model, as acoustic correlates of nasality to create a nasality score for the prediction of hypernasality. The results showed that there are significant differences between the control group and the three groups of Parkinson's patients. The effect sizes between the control group and the aggregation of the three PD groups was of moderate size. This indicates that, even though an excess of nasality is not the most prominent characteristic of PD speech, it is more prevalent in PD patients than it is in healthy individuals.

The results also demonstrated that the effect sizes between the control and the three stages of Parkinson's were increasingly higher with control and $G1$ having a small effect, control and $G2$ a moderate effect and, control and $G3$ a large effect. This indicates that this characteristic could, potentially, also be used to measure the progression of the disease.

6.2 Future Work

For future work we propose expanding the corpus size and establishing a more balanced data set by using texts for the reading tasks that place a special focus on nasal sounds, both consonants and vowels. It would also be interesting to develop end-to-end models. The proposed method in this thesis, uses a forced alignment, which introduces an extra layer of inaccuracy that can affect the performance of the model, due to the possibility of segmentation errors in the alignment. By using end-to-end models this problem is eliminated.

Given the loss in performance that was observed for the FraLusoPark dataset when compared to BD-Publico, it would be interesting to try to enhance the performance for acoustic settings that are different from the one's seen in training. One technique that might help achieve this goal would be data augmentation. Some other evaluations that should be done to test this system include:

- Testing with corpora for different disorders that have hypernasality as a common characteristic such as Huntington's disease, amyotrophic lateral sclerosis and cleft palate, in order to see if the system would generalise well for other conditions.
- Compare the results obtained with our system to perceptual evaluations done by clinicians.

Bibliography

[1] Cmkearne. (2019) Lecture 23: Physiology of resonation. [Online]. Available: https://quizlet.com/345171259/info

[2] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Interspeech*, 2018, pp. 3743–3747.

[3] P. Batista and A. Pereira, "Quality of life in patients with neurodegenerative diseases," *Dimensions*, vol. 1, p. 3, 2016.

[4] G. Deuschl, E. Beghi, F. Fazekas, T. Varga, K. A. Christoforidi, E. Sipido, C. L. Bassetti, T. Vos, and V. L. Feigin, "The burden of neurological diseases in Europe: an analysis for the global burden of disease study 2017," *The Lancet Public Health*, vol. 5, no. 10, pp. e551–e567, 2020.

[5] J. Robin, J. E. Harrison, L. D. Kaufman, F. Rudzicz, W. Simpson, and M. Yancheva, "Evaluation of speech-based digital biomarkers: Review and recommendations," *Digital Biomarkers*, vol. 4, no. 3, pp. 99–108, 2020.

[6] D. A. Cairns, J. H. Hansen, and J. E. Riski, "A noninvasive technique for detecting hypernasal speech using a nonlinear operator," *IEEE transactions on biomedical engineering*, vol. 43, no. 1, p. 35, 1996.

[7] U. Shrawankar and A. Mahajan, "Speech: a challenge to digital signal processing technology for human-to-computer interaction," *CoRR*, vol. abs/1305.1925, 2013.

[8] A. König *et al.*, "Automatic speech analysis for the assessment of patients with predementia and Alzheimer's disease," *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, vol. 1, no. 1, pp. 112–124, 2015.

[9] J. Rusz, R. Cmejla, H. Ruzickova, and E. Ruzicka, "Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated Parkinson's disease," *The journal of the Acoustical Society of America*, vol. 129, no. 1, pp. 350–367, 2011.

[10] R. Voleti, J. M. Liss, and V. Berisha, “A review of automated speech and language features for assessment of cognitive and thought disorders,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 2, pp. 282–298, 2019.

[11] “Appendix a: Mfcc features,” https://link.springer.com/content/pdf/bbm%3A978-3-319-49220-9%2F1.pdf.

[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[13] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems,” in *Twelfth annual conference of the international speech communication association*, 2011.

[14] N. S. Ibrahim and D. A. Ramli, “I-vector extraction for speaker recognition based on dimensionality reduction,” *Procedia Computer Science*, vol. 126, pp. 1534–1540, 2018.

[15] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, “Deep neural network embeddings for text-independent speaker verification.” in *Interspeech*, 2017, pp. 999–1003.

[16] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[17] F. Eyben and B. Schuller, “opensmile:) the munich open-source large-scale multimedia feature extractor,” *ACM SIGMultimedia Records*, vol. 6, no. 4, pp. 4–13, 2015.

[18] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, “The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing,” *IEEE transactions on affective computing*, vol. 7, no. 2, pp. 190–202, 2015.

[19] M. Schmitt and B. W. Schuller, “Openxbow - introducing the Passau open-source crossmodal bag-of-words toolkit,” *J. Mach. Learn. Res.*, vol. 18, pp. 96:1–96:5, 2017.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011.

[21] T. Pruthi, *Analysis, vocal-tract modeling and automatic detection of vowel nasalization*. University of Maryland, College Park, 2007.

[22] J. Yuan and M. Liberman, “Automatic measurement and comparison of vowel nasalization across languages.” in *ICPhS*, vol. 17, 2011, pp. 2244–2247.

[23] M. Saxon, A. Tripathi, Y. Jiao, J. Liss, and V. Berisha, “Robust estimation of hypernasality in dysarthria.” *CoRR*, vol. abs/1911.11360, 2019.

[24] X. Wang, M. Tang, S. Yang, H. Yin, H. Huang, and L. He, “Automatic hypernasality detection in cleft palate speech using cnn,” *Circuits, Systems, and Signal Processing*, vol. 38, no. 8, pp. 3521–3547, 2019.

[25] M. Saxon, J. Liss, and V. Berisha, “Objective measures of plosive nasalization in hypernasal speech,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6520–6524.

[26] “What is neurodegenerative disease?” https://www.neurodegenerationresearch.eu/what/.

[27] A. Pompili, A. Abad, P. Romano, I. P. Martins, R. Cardoso, H. Santos, J. Carvalho, I. Guimaraes, and J. J. Ferreira, “Automatic detection of Parkinson’s disease: an experimental analysis of common speech production tasks used for diagnosis,” in *International Conference on Text, Speech, and Dialogue*. Springer, 2017, pp. 411–419.

[28] C. L. Ludlow, N. P. Connor, and C. J. Bassich, “Speech timing in Parkinson’s and Huntington’s disease,” *Brain and language*, vol. 32, no. 2, pp. 195–214, 1987.

[29] P. Zwirner and G. J. Barnes, “Vocal tract steadiness: a measure of phonatory and upper airway motor control during phonation in dysarthria,” *Journal of Speech, Language, and Hearing Research*, vol. 35, no. 4, pp. 761–768, 1992.

[30] J. Rusz, R. Cmejla, H. Ruzickova, and E. Ruzicka, “Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated Parkinson’s disease,” *The journal of the Acoustical Society of America*, vol. 129, no. 1, pp. 350–367, 2011.

[31] A. M. Goberman and C. Coelho, “Acoustic analysis of parkinsonian speech i: Speech characteristics and l-dopa therapy,” *NeuroRehabilitation*, vol. 17, no. 3, pp. 237–246, 2002.

[32] R. J. Holmes, J. M. Oates, D. J. Phyland, and A. J. Hughes, “Voice characteristics in the progression of Parkinson’s disease,” *International Journal of Language & Communication Disorders*, vol. 35, no. 3, pp. 407–418, 2000.

[33] S. Skodda and U. Schlegel, “Speech rate and rhythm in Parkinson’s disease,” *Movement disorders: official journal of the Movement Disorder Society*, vol. 23, no. 7, pp. 985–992, 2008.

[34] K. Tjaden, “Speech and swallowing in Parkinson’s disease,” *Topics in geriatric rehabilitation*, vol. 24, no. 2, p. 115, 2008.

[35] J. C. Vasquez-Correa, T. Arias-Vergara, M. Schuster, J. R. Orozco-Arroyave, and E. Nöth, “Parallel representation learning for the classification of pathological speech: studies on Parkinson’s disease and cleft lip and palate,” *Speech Communication*, vol. 122, pp. 56–67, 2020.

[36] “Gaussian Mixture Model,” https://brilliant.org/wiki/gaussian-mixture-model/.

[37] R. Gandhi, “Support Vector Machine — Introduction to Machine Learning Algorithms,” https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, 2018.

[38] M. Fabien, “Linear discriminant analysis (lda), qda,” https://maelfabien.github.io/machinelearning/LDA/#concept.

[39] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[41] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[42] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[43] S. Sharma, S. Sharma, and A. Athaiya, “Activation functions in neural networks,” *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.

[44] Z. Zhang, “Improved adam optimizer for deep neural networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2015.

[46] S. K. Kumar, “On weight initialization in deep neural networks,” *CoRR*, vol. abs/1704.08863, 2017.

[47] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[48] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[49] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[50] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in *Sixteenth annual conference of the international speech communication association*, 2015.

[51] Z. Zhang, H. Huang, and K. Wang, “Using deep time delay neural network for slot filling in spoken language understanding,” *Symmetry*, vol. 12, no. 6, p. 993, 2020.

[52] Y.-J. Tsai, C.-P. Wang, and G.-S. Lee, “Voice low tone to high tone ratio, nasalance, and nasality ratings in connected speech of native mandarin speakers: a pilot study,” *The Cleft palate-craniofacial journal*, vol. 49, no. 4, pp. 437–446, 2012.

[53] R. Kataoka, D. W. Warren, D. J. Zajac, R. Mayo, and R. W. Lutz, “The relationship between spectral characteristics and perceived hypernasality in children,” *The Journal of the Acoustical Society of America*, vol. 109, no. 5, pp. 2181–2189, 2001.

[54] T. Pruthi and C. Y. Espy-Wilson, “Acoustic parameters for the automatic detection of vowel nasalization,” in *Eighth Annual Conference of the International Speech Communication Association*. Citeseer, 2007.

[55] G.-S. Lee, C.-P. Wang, and S. Fu, “Evaluation of hypernasality in vowels using voice low tone to high tone ratio,” *The Cleft Palate-Craniofacial Journal*, vol. 46, no. 1, pp. 47–52, 2009.

[56] J. R. Orozco-Arroyave, J. D. Arias-Londoño, J. F. Vargas-Bonilla, and E. Nöth, “Automatic detection of hypernasal speech signals using nonlinear and entropy measurements,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[57] X. Wang, S. Yang, M. Tang, H. Yin, H. Huang, and L. He, “Hypernasalitynet: Deep recurrent neural network for automatic hypernasality detection,” *International Journal of Medical Informatics*, vol. 129, pp. 1–12, 2019.

[58] V. C. Mathad, K. Chapman, J. Liss, N. Scherer, and V. Berisha, “Deep learning based prediction of hypernasality for clinical applications,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6554–6558.

[59] J. R. Orozco-Arroyave, J. F. Vargas-Bonilla, J. C. Vásquez-Correa, C. G. Castellanos-Domínguez, and E. Nöth, “Automatic detection of hypernasal speech of children with cleft lip and palate from spanish vowels and words using classical measures and nonlinear analysis,” *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 80, pp. 109–123, 2016.

[60] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for mfcc feature extraction," in *2010 4th International Conference on Signal Processing and Communication Systems*. IEEE, 2010, pp. 1–5.

[61] C. Carvalho and A. Abad, "Tribus: An end-to-end automatic speech recognition system for european portuguese," *IberSpeech*, 2021.

[62] J. P. Neto, C. A. Martins, H. Meinedo, and L. B. Almeida, "The design of a large vocabulary speech corpus for portuguese," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[63] S. Pinto *et al.*, "Dysarthria in individuals with Parkinson's disease: a protocol for a binational, cross-sectional, case-controlled study in french and european portuguese (fralusopark)," *BMJ open*, vol. 6, no. 11, p. e012885, 2016.

[64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[65] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[66] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" *Advances in neural information processing systems*, vol. 31, 2018.

[67] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," *Advances in neural information processing systems*, vol. 31, 2018.

[68] "One-way anova post hoc tests," https://www.ibm.com/docs/en/spss-statistics/saas?topic=anova-one-way-post-hoc-tests, accessed: 2022-10-18.

[69] S. Skodda, "Analysis of voice and speech performance in Parkinson's disease: a promising tool for the monitoring of disease progression and differential diagnosis," *Neurodegenerative Disease Management*, vol. 2, no. 5, pp. 535–545, 2012.

[70] P. D. Ellis, *The essential guide to effect sizes: Statistical power, meta-analysis, and the interpretation of research results*. Cambridge university press, 2010.