UTRECHT UNIVERSITY

“A Learning Problem”: Deep Learning Target Word Classification for Infant Directed Speech

Bachelor Thesis Artificial Intelligence
7.5 ETCS

Yoram Frenkiel
6147526

Supervisor: Frans Adriaans
Second Reader: Thijs van Ommen

July 2021

UTRECHT UNIVERSITY

Abstract

Target word classification is an important part of speech recognition and artificial intelligence. Due to the rise of deep learning, target word classification models have shown impressive results. However, for some types of data, the accuracy of these models is still lacking. For one, research has found that using state-of-the-art target word classification software for Infant Directed Speech (IDS) - a special type of speech used when talking to infants - results in lower classification accuracy compared to Adult Directed Speech (ADS). In this thesis, we will answer the question: "Can deep learning models be used for successful classification of target words in Infant Directed Speech?" To answer this question two experiments have been conducted in which deep learning classification models (CNNs and RNNs) were trained and evaluated on IDS and ADS. The results of these models have been compared and analyzed. There was found to be no significant difference in classification accuracy between the two types of speaking. Furthermore, the CNN model classified IDS test samples with an accuracy of 85%. From this, it was concluded that deep learning models can be used for successful target word classification of IDS.

Keywords: Infant Directed Speech, Deep Learning, Target Word Classification, Convolutional Neural Network, Recurrent Neural Network

Acknowledgements

I would like to sincerely thank Frans Adriaans for his supervision, input, and feedback during my research. I would also like to thank him and all others who gave me the opportunity to work with the very interesting dataset.

Furthermore, my gratitude goes out to Mara for her unlimited support and feedback and to Cas for proofreading my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial intelligence is used in a variety of fields and for many applications. One such application is speech recognition. Speech recognition is used in many consumer products. Such as the smart assistants from Apple and Amazon (Siri and Alexa respectively). For these applications, a stream of continuous speech is converted to text to be interpreted by the system. Another part of speech recognition is (target) word classification. This task differs from general speech recognition (speech-to-text) in that it classifies single-word recordings. Target word classification can be used for classifying spoken commands - making a robot stop by saying 'stop' for example - or for the annotation of recordings.

Techniques for speech recognition and target word classification have been researched and developed for many years. In the nineties, we saw the use of Hidden Markov Models (HMM) for speech recognition. HMMs were popular because of their ease of use and strong statistical background [3]. Later on, hybrid models emerged. These models combined the original HMMs with Neural Networks (NN). These hybrid models resulted in higher accuracy speech recognition [4]. However, the biggest leap in accuracy and usability was seen with the use of deep learning for speech recognition. Both Abdel-Hamid et al. [5] and Graves, Mohamed & Hinton [6] computed highly accurate deep learning models for speech recognition using Convolutional Neural Network and Recurrent Neural Network approaches respectively. These approaches will be further explained in section 2.3. The same techniques have been used for target word classification. Solovyev et al. [7] successfully used 1- and 2-dimensional Convolutional Neural Networks on a simple speech commands dataset. They were able to correctly classify around 90% of test samples.

Contrary to the overall growth of accuracy in recent years, some use-cases of speech recognition and target word classification are still lacking. Most times this is due to some specific characteristic of the data. For example, there is evidence that speech recognition systems perform worse when used for a classification task on data with added noise[1] [8]. Furthermore, research on speech recognition has demonstrated that alterations should be made to current speech recognition models for children's speech [9]. Likewise, research indicated that speech recognition software under performed when the speakers were experiencing stress [10]. Moreover, Van der Klis et al. [11] saw a significant decrease in accuracy when using state-of-the-art speech recognition software for Infant Directed Speech compared to Adult Directed Speech. Infant Directed Speech is a special type of speech used when talking to infants and small children. The difference in recall was found to be ranging from 7% to 16% in favour of the Adult Directed Speech data, which shows a considerable usability problem for the software when used on Infant Directed Speech. The latter anomaly will be researched closer in this thesis.

Research on target word classification or speech recognition in general for Infant Directed Speech is interesting for two main reasons. For one, it will possibly tell something about the acoustic complexity of Infant Directed Speech. This is an interesting topic because Infant Directed Speech presumably improves language acquisition for children [12]. Hence, more insight into this type of speech could aid research on how humans learn languages because the classification models can illustrate the way language acquisition works. Secondly, this idea is emphasized with the use of deep learning. Deep learning techniques are based on the way our brain works, and in turn, can *learn* in a similar way humans do. Besides more insight into Infant Directed Speech and language acquisition, this research will also add to the growing number of papers on deep learning.

1.1 Research Question

In this thesis, we will explore the possibilities of using deep learning algorithms for target word classification for Infant Directed Speech. The main question we will try to answer in this study is: *Can deep learning models be used for successful classification of target words in Infant Directed Speech?* To be able to answer this question, two sub-questions will be used. Answering the first sub-question will aid the differentiation between Infant Directed Speech and Adult Directed Speech, this will give more insight into the acoustic complexity of Infant Directed Speech, and reads: *Is Infant Directed Speech harder or easier to classify than Adult Directed Speech?* The second

[1] The meaning of the term *noise* in data science is used for data that is in some way flawed. For audio data, this often means that the recordings contain actual noise, such as unintended speech, sounds or background noises.

sub-question is *Are the same patterns noticeable using different Neural Network models?* This question has two purposes. It will help to emphasize possible conclusions about Infant Directed Speech by making sure the results from one model are not just a result from the inner workings of that specific technique. Furthermore, this will give a better understanding of different deep learning classification techniques for audio data.

These questions will be answered by conducting a brief literature review on Infant Directed Speech, audio data, and deep learning models for target-word classification. Furthermore, we will conduct two experiments by modelling, training, and evaluating supervised target word classification models. For the first experiment, we will use a Convolutional Neural Network, for the second experiment a Recurrent Neural Network with Long Short Term Memory will be used. These models will be trained and evaluated on Infant Directed Speech and Adult Directed Speech data. In this way, we will be able to compare and analyze results from two different deep learning approaches on Adult Directed Speech and Infant Directed Speech.

Chapter 2

Theoretical Background

In this chapter, a theoretical basis will be laid for the different components used for experimentation. In section 2.1 the characteristics of both Infant Directed Speech and Adult Directed Speech will be discussed and a comparison between them will be made. Section 2.2 will explain how audio data is used for deep learning. This section will also explain the different audio features and pre-processing steps used. Lastly, section 2.3 will explain the theory behind deep learning and give insight into the different implementations we will be using.

2.1 Infant Directed Speech

Infant Directed Speech (IDS) is the speech used when talking to infants or young children. This speech has some defining characteristics which make it differ from Adult Directed Speech (ADS), this is the speech used when talking to adults. One or more of these characteristics could explain the lack of accuracy for target word classification tasks that was found in the research of Van der Klis et al. [11].

Firstly, IDS is characterized by having a higher frequency, which directly relates to pitch. Studies have given evidence that this characteristic can be found across different languages [13]. Furthermore, IDS has exaggerated pitch contours, which are the patterns of pitch change in speech [14]. This makes IDS more varied than ADS. The latter could be problematic for target word classification when utterances of the same word, from different speakers, vary more widely compared to ADS. On the other hand, this could help a model trained on IDS data to generalize better on unseen test data.

The second characteristic of IDS is the slower speaking rate. Speakers tend to use shorter phrases with longer pauses [13] [15]. This characteristic could have a positive effect on speech recognition because boundaries between words are better defined. However, this will not influence our classification task. As will be explained later in section 3.1, our data consist of recordings of individual words, thus only the slower speaking rate could influence the performance.

Lastly, a characteristic of IDS is the exaggeration of certain parts of a word. Ferland et al [15] found that speakers use more exaggerated intonations. In IDS, speakers also tend to exaggerate certain vowels [16] and phonemes. The latter is called prosodic exaggeration [17]. However, there is no proof that speakers hyper-articulate *full* words when talking to infants [18].

These characteristics are most prominent when talking to children aged 0 to 18 months. For this reason, researchers make a distinction between Infant Directed Speech and Child-Directed Speech. Our dataset contains recordings of target words addressed to infants of two different ages (18 months and 24 months). The aforementioned characteristics seem to have a positive effect on word recognition by infants [19] and seem to facilitate the learning and acquisition of a language [12]. As stated in the introduction deep learning models *learn* in a similar way as humans. This could entail that the positive effect on word recognition an infant encounters from IDS, also facilitates target word classification for deep learning models.

2.2 Audio Data

For this research, we will be using classification models on audio data. This type of data fits in with unstructured data such as images, video, and text. All of these need some form of pre-processing when used for deep learning. The information in this section is sourced from the book *Fundamentals of Music Processing* from Müller and Meinard [20]. To clarify the different pre-processing steps and their accompanying audio representations in this section, we will provide an example of two samples from our dataset (see section 3.1) for each step. These examples will consist of two recordings of the word 'appel' (Eng: 'apple') in IDS as well as ADS.

To get a better understanding of what these steps entail for audio data we will start with a basic definition of what sound is. Sounds are pressure waves created by a vibrating object. These waves make particles in the surrounding medium (air, water, etc.) oscillate [21]. When these vibrations reach a human ear, the brain is able to hear the sound. These ways can be represented by a waveform. Figure 2.1 shows the most simple form of a sound waveform, the sin wave.

A few characteristics of a sound waveform will be explained. First, the amplitude. Amplitude is given by the distance of a point on the curve to the baseline (an amplitude of zero). A longer distance, in either direction, means a higher amplitude. The next characteristic is the period or wavelength. The period is the time between two peaks in amplitude. The period of sound waves is important because it correlated directly with the frequency of sound; frequency is the inverse of the period. This means that a shorter period (short time between two peaks in amplitude) represents a higher frequency. Amplitude and frequency correspond to the loudness and pitch of sound respectively.

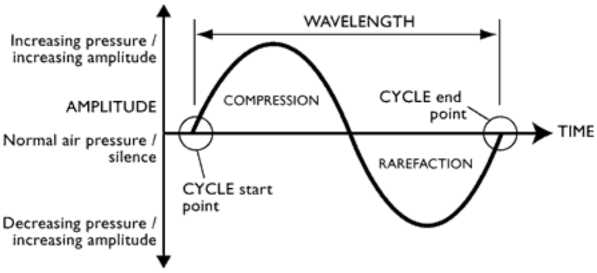

WAVELENGTH
Increasing pressure / increasing amplitude
AMPLITUDE
COMPRESSION
CYCLE end point
Normal air pressure / silence
TIME
RAREFACTION
CYCLE start point
Decreasing pressure / increasing amplitude

FIGURE 2.1: A representation of the simplest form of a sound wave, the sin wave.

2.2.1 Sampling and Quantization

To store and use the analogue waveforms on a computer we need to apply an Analog-Digital Conversion (ADC). This is the task of digitalizing an analogue waveform. For this step, we use sampling and quantization. When sampling, we measure the amplitude of a sound wave at a regular interval over time. The amount of samples and the size of interval is given by the sample rate. This states how many points (samples) are stored for one second of audio. For example, CD-ROMs use a sampling rate of 44100 (or 44.1 kHz). Intuitively, a higher sampling rate follows the analogue signal more closely and thus will result in a higher quality sound representation. However, this will increase the required amount of memory.

The second step, quantization, follows a similar pattern. When sampling the amplitude we quantize the value to the closest bit we use. This is done to reduce the number of bits that need to be stored[1]. The amount of bits used for quantization is given by the bit depth. In this case, CD-ROMs use a bit depth of 16bits/channel[2]. Again, a higher bit-depth results in a higher-quality representation. However, a higher bit depth uses

[1] Both scales (time and amplitude) from analogue waveforms are continuous. This means that the data can theoretically be measured at infinitely small intervals which would need an infinite amount of memory.

[2] 16-bit integers can store 2^{16} values.

more memory. The steps of sampling and quantization result in a digital sound wave as illustrated in figure 2.2.

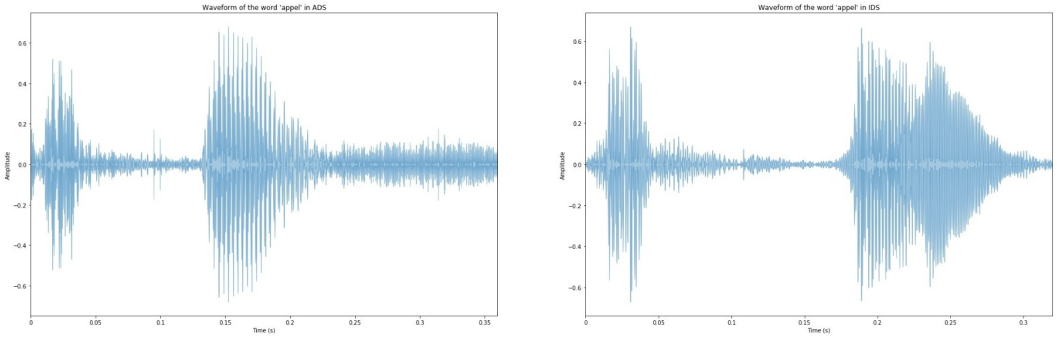


Figure 2.2: The digital sound waves of two recordings of the word 'appel'. The horizontal axis shows time in milliseconds and the vertical axis shows the amplitude. (left: ADS, right: IDS)

2.2.2 (Fast) Fourier Transforms and Short Time Fourier Transforms

The resulting sound waves are way more complex than a regular sin wave - consisting of a clear period and amplitude - as seen in figure 2.1. Still, we are able to extract telling information from this representation using a Fourier Transform (FT). This is a mathematical technique that decomposes a complex periodic soundwave, as shown in figure 2.2, into a sum of sine waves with different frequencies[3]. The result of an FT on a soundwave is called the power spectrum. The resulting power spectrums for our samples, as shown in figure 2.3, show peaks in amplitude at lower frequencies. Thus, the lower frequencies are represented heavily in these recordings which is normal for speech data.

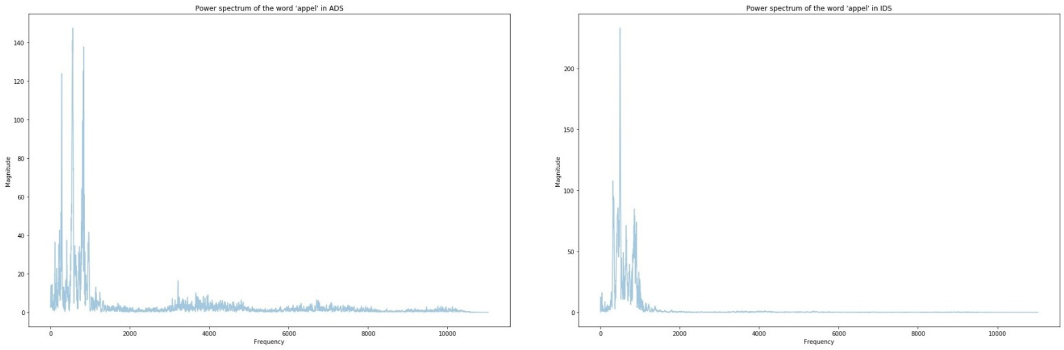


Figure 2.3: Power Spectrum of two recordings of the word 'appel'. The horizontal axis shows the frequency and the vertical axis shows the amplitude. (left: ADS, right: IDS)

[3] The most basic way to explain an FT: imagine the complex sound wave as a pasta dish. The FT can decompose this dish into individual ingredients (pasta, sauce, etc.).

The problem with power spectrums is the lack of information about time. When performing a Fourier Transform we moved from a time-domain (the x-axis shows time) to a frequency domain (the x-axis shows frequency). Thus, we lost our time information i.e. the information about how our sound changed over time[4]. To overcome this problem, we instead use a Short Time Fourier Transform (STFT). An STFT computes a Fast Fourier Transform[5] at multiple intervals along the x-axis. Thus, instead of decomposing the full soundwave, multiple smaller windows of the soundwave are transformed. The size of this window is given by the sample size and the number of samples the window moves across the soundwave is given by the hop-length. An STFT results in a spectrogram with 3-axis: time, frequency and amplitude. In most cases instead of the amplitude, the third axis shows decibels. This is the standard metric used for loudness and is computed by applying a log function over the amplitude. An example of such spectrogram can be found in figure 2.4.

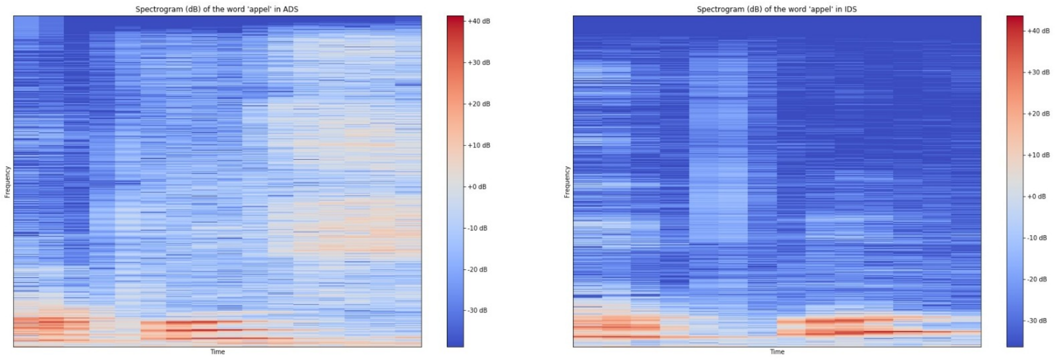


Figure 2.4: Spectrogram of two recordings of the word 'appel'. The horizontal axis shows time, the vertical axis shows frequency and the color represents the amplitude in decibels. (left: ADS, right: IDS)

2.2.3 Mel-Frequency Cepstrum Coefficients

A commonly used feature for speech recognition tasks is a Mel-Frequency Cepstrum Coefficients (MFCC). MFCCs are used to better capture the pitch of a sound i.e. it scales pitch such that it matches more closely to the way humans perceive pitch. In lower frequencies, a small change of pitch is far more noticeable than the same change of pitch in higher frequencies. We will be using MFCCs for the experimentation to get a better and more accurate representation of the differences in pitch between ADS and IDS recordings. To capture this the mel-scale is introduced (*mel* is short for melodic). This scale has been derived from experiments on human subjects. A frequency, measured in Hertz, can be converted to the mel-scale using formula 2.1.

[4] As stated in section 2.1 the change of pitch (over time) is one of the characteristics of IDS. Thus, it is very important to capture this in our data representation.

[5] A Fast Fourier Transform (FFT) differs slightly from a regular FT to decrease processing time.

MFCCs are computed using STFTs, of which the output is scaled according to the mel-scale. To better capture the change of the sound over time a $\Delta MFCC$ and even a $\Delta\Delta MFCC$ can be extracted. These Δ features use the difference of frequency between time t and $t-1$, instead of the absolute value. In figure 2.5 a basic MFCC is shown.

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

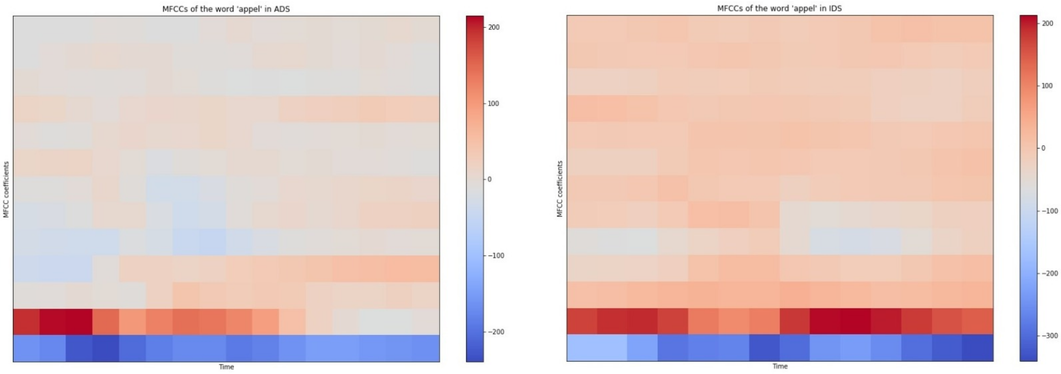


FIGURE 2.5: MFFCs of two recordings of the word 'appel'. The horizontal axis shows time, the vertical axis shows the MFCCs coefficients and the color represent amplitude. (left: ADS, right: IDS)

2.3 Deep Learning

As stated before we have seen a shift for speech recognition from HMMs to deep learning approaches. With the use of deep learning, we see rapid advancements[6]. Artificial Neural Networks (ANN) form the basis of deep learning. In this section, we will explain the concept of basic ANNs (section 2.3.1) and those of more complex ANNs which will be used for experimentation later on - Convolutional Neural Network in section 2.3.2 and Recurrent Neural Network in section 2.3.3.

2.3.1 Artificial Neural Networks

Artificial Neural Networks are connected graphs of nodes that are loosely based on the biological brain. The nodes activate after receiving an input of a certain strength in the same way neurons in our brain 'fire'. By reinforcing certain connections the ANN can learn patterns in the data, again in the same way connections in the brain strengthen when used often. However, most research on ANNs is focused on achieving

[6] These advancements have not only been for speech recognition applications. For example, in 2016 Google's AI finally beat a top player in the game of Go [22]. Something that was deemed impossible before.

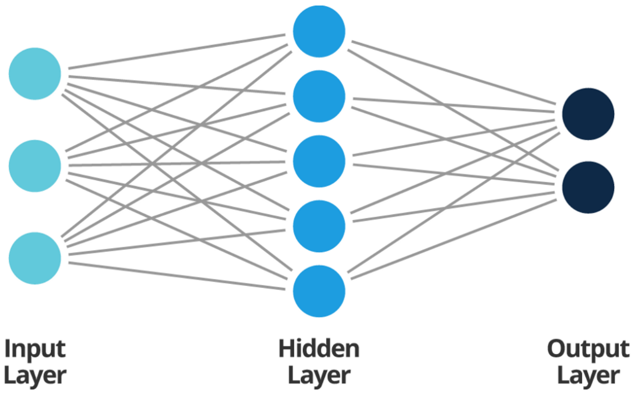


Figure 2.6: A simple Neural Network with an input layers consisting of two nodes, a hidden layer consisting of five nodes and a output layer consisting of two nodes.

highly accurate results rather than further mimicking the biological brain. ANNs have multiple use-cases but are most often used for classification tasks.

ANNs consist of an input layer, an output layer, and one or more hidden layers. The nodes in each layer are connected to the nodes in the following layer[7]. Furthermore, weights are associated with each connection between nodes and a bias term for each hidden layer is used. The amount of nodes in the input layer is based on the input data (for a 39x44 MFCC there are 1716 input nodes). The output layer has nodes equal to the number of classes (in our case we have 12 target words thus 12 output nodes). The number of nodes in the hidden layers can be chosen freely and may vary from hidden layer to hidden layer.

To use an ANN for classification we must train the network on labelled data[8]. First, the weights and biases of the model are initialized with random values. Then, a training sample is passed through the network. To learn how to classify this data best the weights and biases are updated according to the correctness of the output – if the classification of the network was correct the desired weights are strengthened, if the classification was incorrect the desired weights are weakened. In this way, the network adapts to the training samples, which results in a trained network that is capable of classifying unseen data.

2.3.1.1 Activation Function

The nodes in an ANN activate based on the activation function. The input of an activation function for a node is equal to the addition of the value of the connected input nodes times their corresponding weights plus the bias term. The output differs

[7] If all nodes in one layer are connected to all nodes in the following layer we call the network *dense*.
[8] Because labelled data is used ANNs are a form of supervised learning

for each activation function. One of the most basic examples of such function is the binary function – if the input of the node is larger or equal to 0 the output of that node is 1, the output is 0 otherwise. For most applications of deep learning, such as target word classification, more complex activation functions are used.

The one we will be using in most of our hidden layers is the Rectified Linear Function (ReLU), see figure 2.7. Note that this function outputs a real number for input values larger than 0. This function has been found to enable better training for deeper networks [23] and therefore is nowadays one of the most popular activation functions for deep learning [24]. Furthermore, we will be using the Softmax activation function, see figure 2.7. This function is used to normalize the output of the model into a probability distribution for each target word, where the sample will be classified with the target word with the highest probability value. This activation function will thus be used in the output layer of our models.

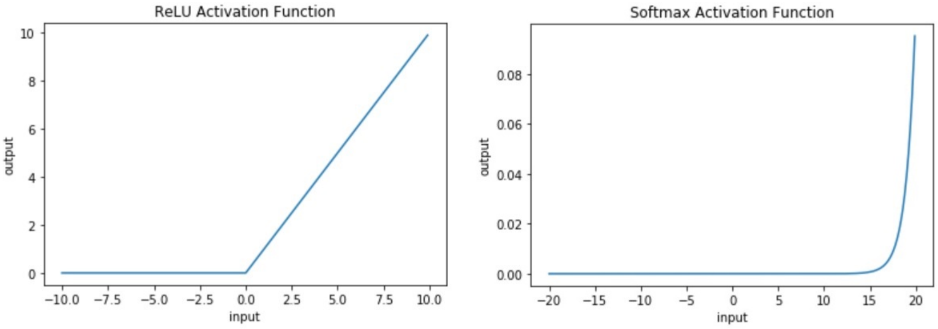


Figure 2.7: Left: plot of the Rectified Linear activation function. Right: plot of the Softmax activation function.

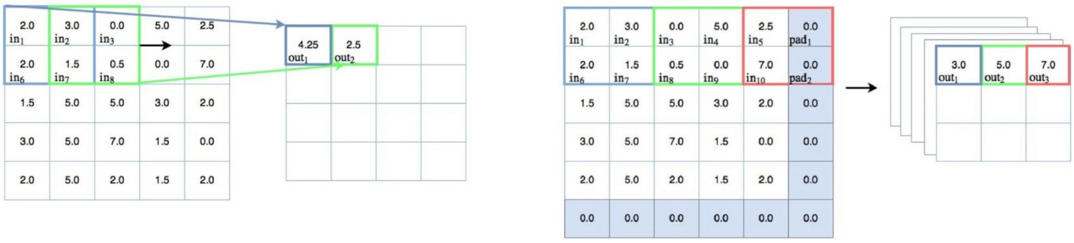
2.3.1.2 Overfitting

When training an ANN there is a risk for the network to overfit the training data [25]. This means the network is biased towards the training data and will not generalize well on unseen test data. This will result in a big difference between the training accuracy (the share of training samples that were classified correctly) and the test accuracy (the share of test samples that were classified correctly). There are multiple ways to overcome overfitting in ANNs, for this thesis we will be explaining just one: dropout.

Dropout is added to layers of a neural network to prevent overfitting and make the model generalize better to unseen test data. For every training run, dropout deactivates a certain percentage of the nodes and connections in a layer (often a value between 20 and 50 per cent is used). This will decrease the training accuracy short term but will in most cases result in a higher testing accuracy which is closer to the accuracy found when training.

2.3.2 Convolutional Neural Network

A neural network approach that has seen remarkable results in classification tasks is the Convolutional Neural Network (CNN). CNNs have been designed for processing and classifying grid-like data, such as images. For example, LeCun et al [26] first used CNN models for recognition of handwritten digits and letters. Images are a well-suited data type because they consist of a $m \times n$ grid of pixels, where each pixel can have one value (gray scale) or multiple values (red, green, and blue) representing its colour. For target word classification, MFCCs or spectrograms have the same properties as a gray scale image and thus are well suited as input data. CNNs try to extract features from the data by applying convolutions. A convolution applies a moving filter over the image. The input of the filter are nodes, representing pixels, in a certain neighbourhood, the nodes in this neighbourhood are weighted and added together which results in a single node output. An example of such convolution can be found in figure 2.8. A convolutional layer is able to detect certain key features from the data such as lines and edges. Convolutional layers also aid in reducing the number of parameters in the network - which reduces overfitting and cuts down on training times. This happens because not all input nodes are connected to the output node, thus the network is not fully connected.



2.0 in_1 | 3.0 in_2 | 0.0 in_3 | 5.0 | 2.5
2.0 in_6 | 1.5 in_7 | 0.5 in_8 | 0.0 | 7.0
1.5 | 5.0 | 5.0 | 3.0 | 2.0
3.0 | 5.0 | 7.0 | 1.5 | 0.0
2.0 | 5.0 | 2.0 | 1.5 | 2.0
4.25 out_1 | 2.5 out_2
2.0 in_1 | 3.0 in_2 | 0.0 in_3 | 5.0 in_4 | 2.5 in_5 | 0.0 pad_1
2.0 in_6 | 1.5 in_7 | 0.5 in_8 | 0.0 in_9 | 7.0 in_{10} | 0.0 pad_2
1.5 | 5.0 | 5.0 | 3.0 | 2.0 | 0.0
3.0 | 5.0 | 7.0 | 1.5 | 0.0 | 0.0
2.0 | 5.0 | 2.0 | 1.5 | 2.0 | 0.0
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0
3.0 out_1 | 5.0 out_2 | 7.0 out_3

Figure 2.8: Left: An example of a convolution: all inputs in a $2x2$ moving filter are weighted (the weight is equal to 0.5) and added together to form the output. Right: Example of max-pooling with a $2x2$ moving filter. Note that zero's are added to the edge of the input layer (image), this is called 0-padding. This is done to make sure there is an equal number of inputs (nodes) in each filter. Because max-pooling is used these zero's will not have any effect on the outcome of this step. [1]

Another important part of CNN models that both reduces the number of parameters and improves the feature detection capabilities of the model is pooling. In this step, a function is applied to a neighbourhood of nodes. Most commonly max-pooling is used, this will apply the `max()` function to the nodes. The output will thus be equal to the node in the neighbourhood with the highest value. Contrary to the convolutional step, there is no overlap between windows on which pooling is applied. This reduces the dimensions of the output layer and the number of parameters in the network. Pooling helps with detecting certain features regardless of orientation or scale. It can be said that pooling acts as a generalizer of the data. An example of this step can be found in figure 2.8 With the use of convolutions and pooling, we can detect objects – or in the case of speech recognition phonemes – within a high-resolution input image. In figure 2.9 we see a fully convolutional network.

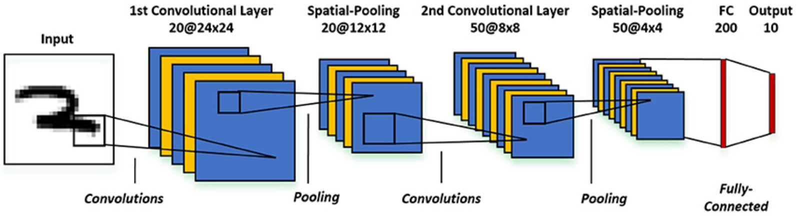


Figure 2.9: A representation of a full Convolutional Neural Network model used on a gray scale image. The CNN has two convolutional layers both followed with a pooling layer. Finally a fully connected output layer is used. [1]

2.3.3 Recurrent Neural Network and Long Short Term Memory

Recurrent Neural Networks are often used for their ability to correctly process sequence data. This characteristic is achieved with the use of a recurrent architecture. This means that the output of the previous sample is combined with the input of the current sample. Because the output of the previous sample still exists in the network, we can say the network has ‘memory’. However, this comes with a downside: important information from earlier samples is less present in the network than the information of the previous sample. Thus, fully recurrent networks have a strong short-term memory but lack a good long-term memory.

To overcome this problem Long Short-Term Memory (LSTM) units were introduced. These units replace the neurons found in an RNN to add long short-term memory to the network. These units add not only the outputs of the previous layer but the output from all previous samples. The internal state of such LSTM unit, consisting of an input gate, output gate, and forget gate, acts as memory. These gates enable an LSTM to store important data in the memory (‘remember’) and get rid of less important data (‘forget’). These capabilities have been found very useful for speech recognition and predictive speech [6]. For speech recognition, an LSTM is able to remember phonemes that have been seen before and forget, for example, background noise. For predictive speech, a recurrent neural network with LSTM will be able to remember important words earlier in a sentence to make a better prediction later on. It will be interesting to see if an RNN with LSTM layers is also capable of correctly classifying target words, as this is not the main use-case of these models.

Chapter 3

Method

In this chapter, we will describe the two experiments conducted for this thesis. The experiments will be on classifying target words from speech data. By conducting these experiments we hope to answer the question if it is possible to successfully classify target words in IDS. Besides, we will research if IDS is harder or easier to classify than ADS and if the same patterns are noticeable using two different deep learning approaches.

3.1 Data Description

For the experiments we will be using a subset[1] of M. Hans dataset [2]. This dataset was used for a study on the role of prosodic input in word learning and contains 916 audio files (`.wav`). These have been recorded at a 16-bit resolution and a sampling rate of 44.1 kHz. The data is split up into two sets of equal size (458 files): one set contains recordings of IDS and one set contains recordings of ADS.

Each audio file contains a recording of one of twelve target words in Dutch. These recordings are around 1 second in length. The full distribution, as shown in figure 3.1, shows that there are some outliers in the data. Also, the IDS recordings tend to be longer, which is in line with the second characteristic as discussed in section 2.1. We want to emphasize that we use short recordings of single target words and not a recording of continuous speech.

[1] The original dataset also contained recordings of Mandarin target words. Also, in the original (Dutch) dataset there were more audio files of IDS recordings than ADS recordings which could lead to an unfavourable advantage for the IDS results. Thus we have chosen to take a subset of the dataset such that there are an equal amount of samples in both sets (ADS and IDS).

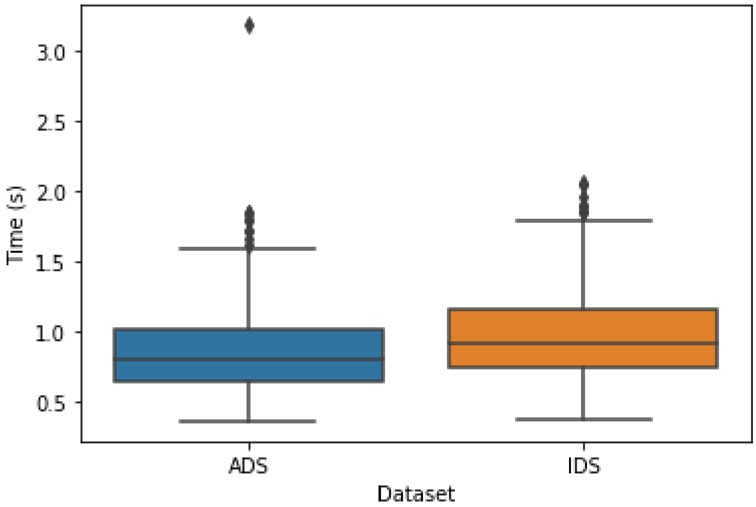


FIGURE 3.1: Distribution of audio file length visualized in a box-plot for both datasets. (left: ADS, right: IDS)

The recordings for the data are made in two stages; in the first stage, caretakers were recorded when reading a book, containing the target words, to an infant at the age of 18 months and when reading the same book to an adult. The same caretakers came back six months later to repeat this procedure. For both stages a different set of target words was recorded, both sets and their English translation can be found in table 3.1. Note, that the words 'appel' and 'opa' (Eng: ' apple' and 'grandpa') appear in both sets. Both datasets contain the same number of samples for each target word. The complete distribution of the 12 target words can be found in figure 3.2.

TABLE 3.1: The different set of Dutch target words recorded for each stage of the experiment conducted by M. Hans [2].

18 months		24 months	
Dutch	English	Dutch	English
appel	apple	appel	apple
bever	beaver	bamboe	bamboo
eland	moose	emoe	emu
kasteel	castle	jasmijn	jasmine
opa	grandpa	kapel	chapel
pompoen	pumpkin	opa	grandpa
walnoot	walnut	wezel	weasel

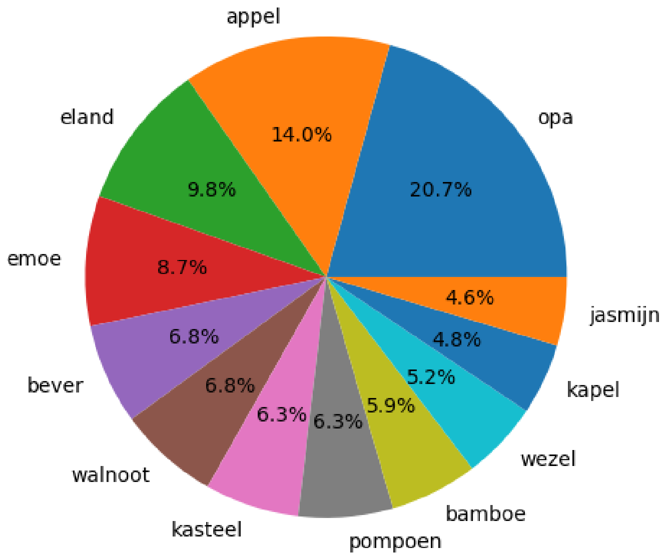


FIGURE 3.2: Distribution of the twelve target words in the dataset.

3.2 Data Preprocessing

For the data pre-processing we have used the Python library Librosa[2]. The audio files have been sampled with a sample rate of 22.05 kHz (= 22050 data points for 1 second of audio). This sample rate is half of that of the recordings, this is chosen to prevent overfitting and cut down on processing time.

Because the audio files are of differing lengths, the resulting (one-dimensional) vectors are as well. To be able to work with data of equal size throughout the pre-processing steps and experimentation we have chosen to consider 22050 samples. This means that for audio files that are longer than one second, the vector has been shortened and for files that are shorter than 1 second the vector has been extended. In the first case, we simply cut off the vector at index 22050. For extending vectors, we applied 0-padding. This is a technique where zeroes are added to the end of a vector to reach the desired length.

From these sampled vectors we extracted MFCCs (13 coefficients, window size of 2048 and a hop length of 512). From the resulting MFCCs we extracted both the ΔMFCC and the ΔΔMFCCs. For every instance, these three MFCCs have been concatenated. This results in a complex MFCC with a shape of $(44, 39)$. All MFCCs and their corresponding label (i.e. target word) have been stored in a `.json` file for further use.

[2] https://librosa.org/

3.3 Evaluation Methods and Metric

The output of our models is a one-dimensional array containing 12 real numbers, one for each target word. These values indicate for each class how likely it is that the sample belongs to that class. The predicted class is the target word corresponding to the highest value in this array.

The models used for experimentation will be evaluated on test accuracy and test error. Accuracy is equal to the division of the number of correctly classified samples by the total number of samples. For the error we use the *(sparse) categorical crossentropy* error function. This function computes the cross-entropy between the predicted class and the actual class and is standard for multi-class classification.

Furthermore, the precision, recall, and F-score are computed for each target word. These metrics are calculated with the use of a confusion matrix, which consists of four values: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). In a multi-class classification task, we calculate these values for every target word individually. For example, for the target word 'appel' these values are equal to:

- TP: the number of samples classified as 'appel' which are labelled 'appel'.
- FP: the number of samples classified as 'appel' which are not labelled 'appel'.
- TN: the number of samples not classified as 'appel' which are not labelled 'appel'.
- FN: the number of samples not classified as 'appel' which are labelled 'appel'.

With these values the precision, recall and F-score can be calculated. The meaning of and the formula for these values is as follows:

- Recall shows how complete the extraction was, TP / (TP + FN) [11]
- Precision shows the exactness of the extraction, TP / (TP + FP) [11]
- F-score is the harmonic mean between the precision and recall, 2TP / (2TP + FP + FN [11]

3.4 Experiment 1

The first experiment consists of training and evaluating CNN models on both ADS data and IDS data. Each dataset is split into a training, testing, and validation set using a 70, 20, 10 split. The splitting is done *stratified*, this results in samples of every target word being present in each set, and thus prevents any set from not having samples of a certain target word. Each model will be trained over 50 epochs with a batch size of 16.

The step of training and evaluating will be repeated 50 times, with different training, testing, and validation sets[3]. The average test accuracy and test error will be computed. Furthermore, the model that performed best (highest test accuracy and lowest test error) will be saved and analyzed.

By repeating the training step 50 times[4] we hope to eliminate the influence of randomness in our model. There are two main sources of randomness: 1. the use of dropout layers 2. the splitting of the data into train, test, and validation sets. Furthermore, by repeating the training and testing step multiple times, we will be able to spot patterns that may occur in the results more easily.

3.4.1 Model Description

We will train the model on a Convolutional Neural Network architecture[5] using the Python library Keras[6].

The CNN is a sequential model consisting of five layers. The first layer consists of a two-dimensional Convolutional input layer with Max-Pooling and Batch Normalization[7]. This Convolutional layer setup is repeated two more times followed by a Dense layer. These first four layers all use the ReLU activation function. Finally, a Dense output layer is added with a Softmax activation function. For this model, the Adam optimizer is used with a sparse categorical cross-entropy error function. The full model summary can be found in the appendix, figure A.1.

3.4.2 Results

Table 3.2 shows the mean test error and mean test accuracy over all 50 CNN models for both the ADS dataset and the IDS dataset. After performing a paired t-test we found no significant difference between the mean test error, t(98) = 1.924, p = 0.057. The mean test error was lower (i.e. preferred) for the models trained on ADS ($M =$

[3] Because the splitting of the data is random, there is a very small change the distribution is the same for two different runs.

[4] The value 50 is chosen because it provides meaningful results without making the run time of the experiment unnecessarily long.

[5] The model architecture is inspired by work of Dr. Valerio Velardo.

[6] `https://keras.io/`

[7] Batch Normalization is used to decrease the training time of the network.

0.741, SD = 0.104, 95% CI = [0.712, 0.770]) compared to the models trained on IDS data (M = 0.784, SD = 0.110, 95% CI = [0.754, 0.815]). The same paired t-test was performed for the mean test accuracy which also found no significant difference, t(98) = -1.375, p = 0.172. The mean test accuracy was higher (i.e. preferred) for the models trained on ADS (M = 0.796, SD = 0.039, 95% CI = [0.785, 0.806]) compared to the models trained on IDS data (M = 0.785, SD = 0.041, 95% CI = [0.774, 0.796]).

In table 3.3 the test error and test accuracy from the models that achieved the highest accuracy score for each type of data is shown. Similar to the mean scores, the scores from the model trained on ADS data are slightly better compared to the scores from the model trained on IDS data.

Table 3.4 shows the recall, precision and F-score for each target word over 50 models for both datasets[8]. The differences in precision, recall and F-score between the models trained on ADS data and the models trained on IDS data will be analyzed in the following section.

Table 3.2: Mean test error and test accuracy over 50 CNN models for both datasets.

Dataset	Test Error	Test Accuracy
ADS	0.741	0.796
IDS	0.784	0.785

Table 3.3: Test error and test accuracy of the best performing CNN model for both datasets.

Dataset	Test Error	Test Accuracy
ADS	0.500	0.880
IDS	0.592	0.859

3.4.3 Analysis

The results from the first experiment show a very slight difference in error (-0.043 for the ADS model) and accuracy (+0.011 for the ADS model) between the different datasets. These differences are found to be not significant. Thus, the CNN models were able to classify both types of data equally well. This could mean that the characteristics of IDS, mentioned in section 2.1, do not make this type of speech easier or harder to classify than ADS. It is possible that the characteristic of IDS influences the deep

[8] These scores are calculated using the total confusion matrix for each word. The total confusion matrix is computed by element-wise addition of all 50 confusion matrices (one for each model) for each word.

Table 3.4: Recall, precision and F-score over 50 CNN models for each target word. (left: ADS, right: IDS)

Word	Precision	Recall	F-Score
appel	0.824	0.758	0.794
bamboe	0.430	0.655	0.519
bever	0.727	0.779	0.752
eland	0.844	0.772	0.806
emoe	0.808	0.736	0.770
jasmijn	0.825	0.868	0.845
kapel	0.620	0.849	0.717
kasteel	0.930	0.918	0.924
opa	0.915	0.819	0.864
pompoen	0.727	0.838	0.779
walnoot	0.867	0.820	0.843
wezel	0.660	0.829	0.735

Word	Precision	Recall	F-Score
appel	0.812	0.824	0.818
bamboe	0.673	0.759	0.713
bever	0.690	0.714	0.702
eland	0.800	0.769	0.784
emoe	0.762	0.704	0.732
jasmijn	0.645	0.791	0.711
kapel	0.765	0.860	0.810
kasteel	0.883	0.872	0.877
opa	0.936	0.818	0.873
pompoen	0.630	0.756	0.687
walnoot	0.773	0.756	0.764
wezel	0.616	0.723	0.665

learning models, but that this influence is either negligible or the negative effect of one characteristic is compensated with the positive effect of another characteristic. However, the same can be true for ADS. Furthermore, the CNN models are trained and evaluated on both datasets. This could explain the fact that we are not seeing significant differences in accuracy between ADS and IDS, as found in the study of Van der Klis et al [11]. The model used in this study was, unlike our models, trained on 'regular' speech data, which largely or even only consist of ADS.

As stated before the IDS dataset consists of two subsets. Speech directed at 18-months-old infants and speech directed at 24-months-old infants. The latter is closer to ADS speech than IDS directed at 18-months-old infants [11], which could have led to better classification accuracy. However, the same pattern is noticeable for these subsets as for the complete dataset. There is no clear pattern that target words in either subset are classified better or worse. For example, both 'pompoen' (18 months) and 'wezel' (24 months) have low F-scores in IDS and both have an equally large difference in F-score compared to the model trained on ADS data. Furthermore, there does not seem to be evidence that certain word characteristics influence accuracy. For example, the word 'kasteel' performs very well across both datasets. However, the word 'kapel' - which consists of similar phonemes - performs worse in the ADS model compared to the IDS model.

Moreover, the result in precision, recall, and F-score show target words with a more than average amount of samples in the dataset ('opa' and 'appel') to perform very well. Yet, this does not necessarily mean that words with a less than average amount of samples under perform. Namely, the rarer words 'wezel' and 'bamboe' do show below average results but the words 'jasmijn' and 'kasteel' show a high F-score. The

ADS models resulted in a very low precision (0.430) for the word ‘bamboe’. This means many test samples were miss classified as this word.

3.5 Experiment 2

The second experiment will repeat the steps of the first experiment. However, instead of training a CNN model, we will be training an RNN-LSTM model. The data, set up, amount of epochs, and batch size are kept the same. The reason for conducting this second experiment is to see if the results show similar patterns as were found in experiment 1, which will in turn answer our second sub-question.

3.5.1 Model Description

We will train the model on a Recurrent Neural Network architecture[9] using the Python library Keras.

The RNN is a sequential model consisting of five layers. The first layer is an LSTM input layer. This is followed by two LSTM layers with dropout. Then a Dense layer with dropout is added. These first four layers all use the ReLU activation function. Finally, a Dense output layer is added with a Softmax activation function. For this model, the Adam optimizer is used with a sparse categorical cross-entropy error function. The full model summary can be found in the appendix, figure A.2.

3.5.2 Results

Table 3.5 shows the mean test error and mean test accuracy over all 50 RNN-LSTM models for both the ADS dataset and the IDS dataset. After performing a paired t-test we found no significant difference between the mean test error, t(98) = 0.637, p = 0.525. The mean test error was lower (i.e. preferred) for the models trained on ADS (M = 1.308, SD = 0.128, 95% CI = [1.270, 1.340]) compared to the models trained on IDS data (M = 1.324, SD = 0.123, 95% CI = [1.290, 1.360]). The same paired t-test was performed for the mean test accuracy which also showed no significant difference, t(98) = -0.678, p = 0.499. The mean test accuracy was higher (i.e. preferred) for the models trained on ADS (M = 0.612, SD = 0.055, 95% CI = [0.597, 0.627]) compared to the models trained on IDS data (M = 0.605, SD = 0.048, 95% CI = [0.592, 0.618]).

[9] The model architecture is inspired by work of Dr. Valerio Velardo.

In table 3.6 the test error and test accuracy from the models that performed best on the test dataset for each type of data is shown. In this case, the test error of the model trained on ADS data is slightly better compared to the test error of the model trained on IDS data but the test accuracy of this model is worse (lower) than the test accuracy of the model trained on IDS data.

Table 3.7 shows the recall, precision and F-score for each target word over 50 models for both datasets. The differences in precision, recall and F-score between the models trained on ADS data and the models trained on IDS data will be analyzed in the following section.

Table 3.5: Mean test error and test accuracy over 50 RNN-LSTM models for both datasets.

Dataset	Test Error	Test Accuracy
ADS	1.308	0.612
IDS	1.324	0.605

Table 3.6: Test error and test accuracy of the best performing RNN-LSTM model for both datasets.

Dataset	Test Error	Test Accuracy
ADS	1.032	0.717
IDS	1.081	0.739

Table 3.7: Recall, precision and F-score over 50 RNN models for each word. (left: ADS, right: IDS)

Word	Precision	Recall	F-Score
appel	0.783	0.597	0.677
bamboe	0.147	0.518	0.229
bever	0.377	0.543	0.445
eland	0.769	0.510	0.613
emoe	0.628	0.548	0.585
jasmijn	0.480	0.632	0.546
kapel	0.160	0.471	0.239
kasteel	0.833	0.731	0.779
opa	0.866	0.675	0.759
pompoen	0.533	0.705	0.607
walnoot	0.490	0.610	0.543
wezel	0.172	0.623	0.270

Word	Precision	Recall	F-Score
appel	0.797	0.621	0.698
bamboe	0.210	0.525	0.300
bever	0.537	0.540	0.538
eland	0.684	0.494	0.574
emoe	0.658	0.527	0.585
jasmijn	0.225	0.529	0.316
kapel	0.196	0.500	0.281
kasteel	0.773	0.609	0.681
opa	0.900	0.745	0.815
pompoen	0.400	0.545	0.461
walnoot	0.433	0.549	0.484
wezel	0.188	0.603	0.287

3.5.3 Analysis

First of all, it should be noted that the scores from the second experiment, using an RNN-LSTM architecture, are significantly worse. However, this should not be deemed problematic, because we are interested in the difference between the dataset and not in the difference between the two neural network architectures.

The results from the second experiment show similar small margins between the two datasets for error (-0.016 for the ADS model) and accuracy (+0.007 for the ADS model). The same is true when looking at the differences between IDS directed at 18-month-old infants and 24-month-old infants. There is no clear pattern that either performs better than the other. The results emphasize that the words of which more samples are present in the dataset ('opa' and 'appel') perform better compared to words that are rarer in the dataset.

The precision of the words 'bamboe', 'kapel', and 'wezel' are extremely low but have an average recall score. This shows that other words were often wrongly classified as one of these three. This, however, has happened in both datasets equally often and thus does not give much insight into the difference between Infant Directed Speech and Adult Directed Speech. These words all have a less than average amount of samples in the dataset, which emphasizes the positive influence of more data.

Chapter 4

Discussion

In this chapter, we will discuss the research conducted for this thesis. Additionally, we will evaluate our limitations for this research and propose possible alterations for further studies on this topic.

4.1 Implications

This research has shown that deep learning classification models are certainly capable of accurately classifying audio data. This is in line with earlier work and emphasizes that deep learning is the gold standard for complex classification tasks. The results showed little to no difference in classification accuracy between the datasets. Because of this, we can assume that there are no acoustic differences between ADS and IDS that are significant enough to make one harder to classify than the other. This however does not necessarily mean that there are no acoustic differences or differences in complexity between the two types of speaking at all. Further studies need to be conducted on this subject to draw decisive conclusions on the acoustic complexity of IDS compared to ADS.

The results did show interesting implications for deep learning classification. Firstly, the results showed classes with a more than average amount of samples to be classified better. However, this was not a rule set in stone as some classes that are less represented in the data still performed very well. From this, we can assume that deep learning can be used in situations where the amount of available training data is lacking. Nevertheless, using more data, if possible, is still the better option. Secondly, we have shown CNN models to outperform RNN models for this classifications task, which is to be expected because of the use of image-like MFFCs. These results emphasize that the choice of model for the task, data, and data representation at hand is crucial.

4.2 Limitations

For this thesis, the reader should consider some limitations. For one, the models have been trained on a relatively small dataset. This, although less problematic than expected, should still be considered. The amount of data can be seen as inadequate for making conclusions about certain subsets, such as the 18 and 24 months old infants split. Furthermore, the amount of data has shown to be insufficient for near-perfect classification (accuracy scores of 98%+).

Secondly, the data contained some noise. In the case of audio data, noise can consist of static background noises or unintentional sounds or speech. In our case, most noise comes from the infants that took part in the experiment. This means that the IDS data presumably contains more noise than the ADS dataset, this could have had a negative effect on the classification of the IDS set. Because of this, differences in classification accuracy could have arisen from this noise instead of the acoustic differences between the two types of speaking.

As explained in section 3.1, the audio files used were of differing lengths. This was solved by only considering a maximum of one second of audio, which leads to the loss of information for longer samples. Because the average length of samples in the IDS dataset was found to be longer than the average length of samples in the ADS set, this could have had a more negative effect on the classification of IDS samples compared to ADS samples.

Lastly, the evaluation of the models is primarily based on what went right. This means there has not been a thorough investigation into a wrong classification. These wrong classifications could have told us more about which words were often confused by the model and if this trend was more or less visible for either dataset. However, with the use of a cross-entropy error function and by computing the precision and recall for each word we were able to get some information about the wrongness of a classification.

4.3 Further Studies

Due to the aforementioned limitations, there are multiple ways in which the current study can be improved upon. These could be alterations upon the conducted research or research on new topics arisen from this research.

For one, this research could be conducted with more data. This would back up the results in a more solid matter. Also, this would lead to the possibility to investigate more specific characteristics of the data. This could entail the aforementioned difference between IDS directed at 18-month-old infants and IDS directed at 24-month-old infants. Furthermore, with a larger dataset, the data could contain a distinction between words that are easy to classify - words consisting of completely different phonemes - and words that are hard to classify - words consisting of similar phonemes. The results of such studies could aid the research on IDS and deep learning classification even further.

Another way this research could be improved upon is by comparing the classification results of different audio representations. In this study, MFCCs were used because of their advantage to better capture the pitch of a sound. However, using spectrograms or even raw audio waveforms as input for the models could give a better understanding of the way deep learning models work and might enlighten the places where ADS and IDS differ even better.

Lastly, research into unsupervised target word classification could be interesting. Unsupervised models train on unlabeled data by clustering (i.e. grouping) similar input samples together. In the case of target word classification, each cluster would represent a target word. This way of learning will possibly be an even better parallel to the way infants learn a language. Furthermore, because unsupervised classification models do not need labelled data they could be efficiently used for annotation purposes.

Chapter 5

Conclusion

In this chapter, we will answer the main research question. This will be done by first answering the two sub-questions.

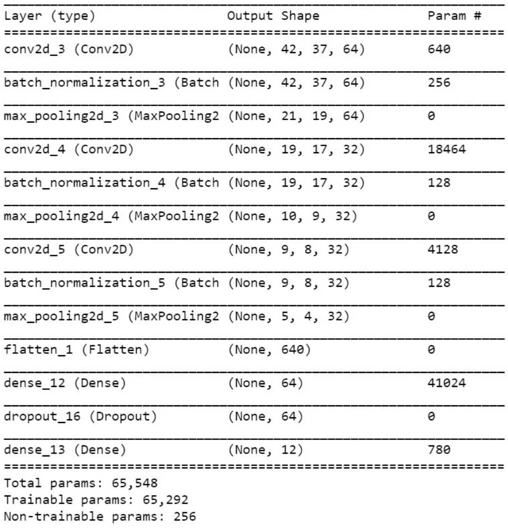
The first sub-question asked if Infant Directed Speech is harder or easier to classify than Adult Directed Speech. Both experiments showed a non-significant difference in classification accuracy and error between the two sets. From this, we can conclude that neither type of speech is harder to classify than the other. However, these results do not provide significant evidence about the difference in acoustic complexity between the Infant Directed Speech and Adult Directed Speech.

The second sub-question *"Are the same patterns noticeable using different Neural Network models?"* was answered by conducting two experiments, each with a different Neural Network approach. The results showed near-identical patterns between the two experiments. This accentuates the assumption that neither type of speech is harder to classify. But, these experiments did show Convolutional Neural Networks to outperform the Recurrent Neural Network. From this, we can conclude that for audio classification tasks using MFCCs Convolutional Neural Networks are best used.

With the use of the answers found for the sub-questions, we can answer the main research question of this thesis: *"Can deep learning models be used for successful classification of target words in Infant Directed Speech?"* This thesis showed that Infant Directed Speech can be classified with an average accuracy score of 79% and highs of 85% using a Convolution Neural Network model. These scores, although reasonably high, show there to be room for improvement. However, the fact that both types of speech are classified equally well in combination with the relatively small dataset and the results of earlier research on word classification gives enough reason to assume that Infant Directed Speech can be classified successfully.

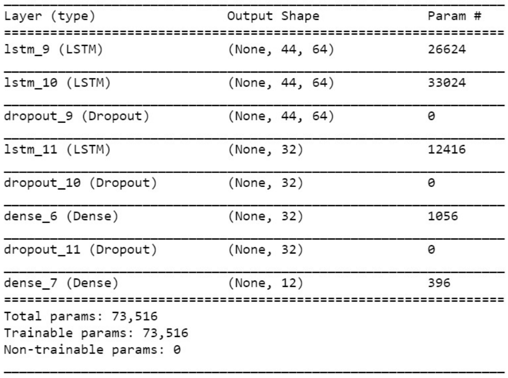
Appendix A

Appendix



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 42, 37, 64)        640
_________________________________________________________________
batch_normalization_3 (Batch (None, 42, 37, 64)        256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 21, 19, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 19, 17, 32)        18464
_________________________________________________________________
batch_normalization_4 (Batch (None, 19, 17, 32)        128
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 10, 9, 32)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 9, 8, 32)          4128
_________________________________________________________________
batch_normalization_5 (Batch (None, 9, 8, 32)          128
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 5, 4, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 640)               0
_________________________________________________________________
dense_12 (Dense)             (None, 64)                41024
_________________________________________________________________
dropout_16 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_13 (Dense)             (None, 12)                780
=================================================================
Total params: 65,548
Trainable params: 65,292
Non-trainable params: 256
```

Figure A.1: Full model summary for the Convolutional Neural Network.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_9 (LSTM)                (None, 44, 64)            26624
_________________________________________________________________
lstm_10 (LSTM)               (None, 44, 64)            33024
_________________________________________________________________
dropout_9 (Dropout)          (None, 44, 64)            0
_________________________________________________________________
lstm_11 (LSTM)               (None, 32)                12416
_________________________________________________________________
dropout_10 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_6 (Dense)              (None, 32)                1056
_________________________________________________________________
dropout_11 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_7 (Dense)              (None, 12)                396
=================================================================
Total params: 73,516
Trainable params: 73,516
Non-trainable params: 0
_________________________________________________________________
```

Figure A.2: Full model summary for the Recurrent Neural Network with Long Short Term Memory.

Bibliography

[1] URL `https://towardsai.net/`.

[2] Mengru Han. *The role of prosodic input in word learning: A cross-linguistic investigation of Dutch and Mandarin Chinese infant-directed speech.* PhD thesis, LOT, 2019.

[3] Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[4] Joseph Michael Tebelskis. *Speech recognition using neural networks.* Carnegie Mellon University, 1995.

[5] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[7] Roman A Solovyev, Maxim Vakhrushev, Alexander Radionov, Irina I Romanova, Aleksandr A Amerikanov, Vladimir Aliev, and Alexey A Shvets. Deep learning approaches for understanding simple speech commands. In *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*, pages 688–693. IEEE, 2020.

[8] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. "did you hear that?" adversarial examples against automatic speech recognition. *arXiv:1801.00554*, 2018.

[9] Matthew Black, Joseph Tepperman, Sungbok Lee, Patti Price, and Shrikanth S Narayanan. Automatic detection and classification of disfluent reading miscues in young children's speech for the purpose of assessment. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[10] Brian D Womack and John HL Hansen. Classification of speech under stress using target driven features. *Speech Communication*, 20(1-2):131–150, 1996.

[11] Anika van der Klis, Frans Adriaans, Mengru Han, and René Kager. Automatic recognition of target words in infant-directed speech. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*, pages 522–522, 2020.

[12] Erik D Thiessen, Emily A Hill, and Jenny R Saffran. Infant-directed speech facilitates word segmentation. *Infancy*, 7(1):53–71, 2005.

[13] Anne Fernald, Traute Taeschner, Judy Dunn, Mechthild Papousek, Bénédicte de Boysson-Bardies, and Ikuko Fukui. A cross-language study of prosodic modifications in mothers' and fathers' speech to preverbal infants. *Journal of child language*, 16(3):477–501, 1989.

[14] Laurel J Trainor and Renée N Desjardins. Pitch characteristics of infant-directed speech affect infants' ability to discriminate vowels. *Psychonomic bulletin & review*, 9(2):335–340, 2002.

[15] Anne Fernald and Thomas Simon. Expanded intonation contours in mothers' speech to newborns. *Developmental psychology*, 20(1):104, 1984.

[16] Patricia K Kuhl, Jean E Andruski, Inna A Chistovich, Ludmilla A Chistovich, Elena V Kozhevnikova, Viktoria L Ryskina, Elvira I Stolyarova, Ulla Sundberg, and Francisco Lacerda. Cross-language analysis of phonetic units in language addressed to infants. *Science*, 277(5326):684–686, 1997.

[17] Frans Adriaans and Daniel Swingley. Prosodic exaggeration within infant-directed speech: Consequences for vowel learnability. *The Journal of the Acoustical Society of America*, 141(5):3070–3078, 2017.

[18] Alejandrina Cristia and Amanda Seidl. The hyperarticulation hypothesis of infant-directed speech. *Journal of Child Language*, 41(4):913–934, 2014.

[19] Jae Yung Song, Katherine Demuth, and James Morgan. Effects of the acoustic properties of infant-directed speech on infant word recognition. *The Journal of the Acoustical Society of America*, 128(1):389–400, 2010.

[20] Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015.

[21] Philip McCord Morse, Acoustical Society of America, and American Institute of Physics. *Vibration and sound*, volume 2. McGraw-Hill New York, 1948.

[22] Elizabeth Gibney. Google masters go: deep-learning software excels at complex ancient board game. *Nature*, 529(7587):445–447, 2016.

[23] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[24] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv:1710.05941*, 2017.

[25] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[26] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.