Kristin Ottesen Steinskog

# Child Speech Recognition

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Torbjørn Karl Svendsen
June 2021

**Master's thesis**

**NTNU**
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Kristin Ottesen Steinskog

# Child Speech Recognition

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Torbjørn Karl Svendsen
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

# Preface

This report has been written for submission of my master thesis, spring 2021, at the program Electronics Systems Design and Innovation at the Norwegian University of Science and Technology (NTNU). I would like to express my great gratitude to my supervisor, Torbjørn Svendsen, for his guidance and support during my work with my thesis.

Kristin Ottesen Steinskog

Trondheim, 11.06.21

# Abstract

Child speech recognition is a challenging task, and most of the speech recognition systems today are based on speech from adults. Speech recognition technology can help speech and language development of young children. Hence, it is essential to improve speech recognition systems to apply better for children. The differences in the speech characteristics of child speech compared to adult speech are significant. Children have a shorter vocal tract length, which implies that they have higher formant frequencies than adults. These features affect the performance of the speech recognition systems. The purpose of this thesis is to improve and implement an automatic speech recognition (ASR) system for child speech by adapting a model trained on data from adult speech to child speech. It aims to investigate the method of transfer learning, where a model trained on adult speech is transferred to fit the acoustics of the speech signal of children.

The speech recognition system is implemented by Mozilla's Deep Speech architecture, and is trained and tested on child speech data from the CMU Kids corpus. Four transfer learning experiments are explored in addition to fine-tuning a pre-trained adult model. The results show a significant decrease in the word error rate (WER), where the best obtained results show a WER of 27.93% after fine-tuning of the model. This is a significant improvement down from a WER of 48.01%, which is the performance of the child speech data tested on the pre-trained adult model. The best achieved result of the transfer learning models has a WER of 36.68%, and indicates that it was difficult to get a low WER due to the lack of data. A WER of 27.93% is still quite high, but indicates that

the use of adult data can be effective for training an automatic child speech recognition system, when there are lack of child speech data.

# Sammendrag

Talegjenkjenning for barn er utfordrende ettersom dagens talegjenkjenningssystem er basert på tale fra voksne. Talegjenkjenning kan hjelpe utviklingen av tale og språk hos barn. Derfor er det viktig å forbedre talegjenkjenningssystemene, slik at de fungerer bedre for barn. Forskjellene i karakteristikken i tale hos barn sammenlignet med voksne er stor. Barn har kortere vokaltrakt, noe som gjør at de har høyere formantfrekvenser enn det voksne har. Dette påvirker ytelsen til talegjenkjenningssystemet. Hensikten med denne oppgaven er å forbedre og implementere et talegjenkjenningssystem for barnestemme. Dette er gjort ved å adaptere en modell som er trent på voksenstemme, til barnestemme. Det tar sikte på å undersøke metoden "transfer learning", hvor en modell som er trent på tale fra voksne er overført til å passe akustikken til talesignalet til barn.

Talegjenkjenningssystemet er implementert ved Mozilla's Deep Speech arkitektur, og er trent og testet på barnestemme fra korpuset CMU Kids. Fire transfer learning eksperiment er utforsket, i tillegg til finjustering av en ferdigtrent modell for voksenstemme. Resultatene viser betydelig nedgang i "word error rate (WER)", der det beste resultatet viser en WER på 27.93% etter finjustering av modellen. Dette er en betydelig forbedring, ned fra WER på 48.01%, som er ytelsen på den ferdigtrente voksenmodellen, testet med barnestemme. Det best oppnådde resultatet fra "transfer learning" modellene har en WER på 36.68%, og indikerer at det er vanskelig å få en lav WER på grunn av manglende data. En WER på 27.93% er fortsatt ganske høyt, men indikerer at det å bruke data fra voksenstemme kan være effektivt i implementeringen av en automa-

tisk talegjenkjenner for barn, der det er mangel på data fra barnestemme.

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **ASR** | Automatic Speech Recognition |
| **AM** | Acoustic Model |
| **CER** | Character Error Rate |
| **CTC** | Connectionist Temporal Classification |
| **DCT** | Discrete Cosine Transform |
| **DFT** | Discrete Fourier Transform |
| **DNN** | Deep Neural Network |
| **DS1** | Deep Speech 1 |
| **DS2** | Deep Speech 2 |
| **DSbM** | Deep Speech by Mozilla |
| **E2E** | End-to-end |
| **GMM** | Gaussian Mixture Model |
| **GPU** | Graphic processing unit |

**HMM** Hidden Markov Model

**LM** Language Model

**LSTM** Long Short-term Memory

**MFCC** Mel Frequency Cepstral Coefficient

**MLLR** Maximum Likelihood Linear Regression

**NTNU** Norwegian University of Science and Technology

**PLP** Perceptual Linear Prediction

**ReLU** Rectified Linear Unit

**RNN** Recurrent Neural Network

**SAT** Speaker Adaptive Training

**TL** Transfer Learning

**VTLN** Vocal Tract Length Normalization

**WER** Word Error Rate

# Chapter 1

# Introduction

## 1.1 Background

Today, automatic speech recognition (ASR) systems are popular for both adults and children, and their use increases every day. Technology is getting more accessible for children in both education and everyday life. The applications of speech recognition are typical as virtual assistants, learning languages and can be used as an aid for people with disabilities. The problem with traditional ASR systems today is that they are not designed for children. The ASR systems are based on speech from adults, which has significantly different acoustics than children. Hence, the performance of the systems is not as good for child speech compared to adult speech.

Children are the ones who can benefit most from speech recognition. It can help speech and language development for young children, and it can help improve their communication capability in society. If the ASR systems are not designed to work for child speech, children will lose the great opportunity of all these features. It is therefore important to make ASR systems more applicable to children.

The challenges with child speech recognition result from a large variation in the speech signal and spectra between children and adults. The main issues behind today's ASR systems lie in the spectral mismatch between children and adults. A child's vocal tract is shorter than adults, which makes the pitch and formant frequencies higher. Their language is also less developed compared to adults.

To make an ASR system for children, the best opportunity would have been to train on child speech, to get all the advantages from the features of the child speech signal. The issue with this is lack of child speech data compared to how much needed to train a solid ASR system. A large amount of data are needed to train a Deep Neural Network (DNN) ASR system. This does exist for speech from adults, but not children. To overcome this, transfer learning is a method used in situations with less data, where in this situation, the knowledge learnt from training on adult speech is transferred to train a child speech ASR system.

## 1.2 Objectives

The objective of this master thesis is to adapt an ASR model trained on adult speech to child speech, to get an improved ASR system for children. In other words, the word error rate (WER) needs to be reduced. The implementation will be done by Mozilla's Deep Speech, as this architecture is easy to adapt. Deep Speech also has a pre-trained model, which is trained on many hours of adult speech.

## 1.3 Structure of the Report

The rest of the report is structured as follows:

- Chapter 2 gives an introduction to the theory relevant for child speech characteristics and the automatic speech recognition system.
- Chapter 3 is a literature review of state-of-the-art automatic speech recognition systems and speech recognition for children.
- Chapter 4 presents the tools and the dataset.
- Chapter 5 is comprised of the Deep Speech model, the data preparation and the system implementation.
- In chapter 6, the results from the different experiments are presented.
- Chapter 7 discusses the results from chapter 6.
- Finally, in chapter 8, the conclusion and suggestions for future work are presented.

# Chapter 2

# Theory

The following chapter covers the most relevant background theory on child speech characteristics, automatic speech recognition systems and machine learning. Section 2.1 presents the characteristics of child speech and the differences from adult speech, both the variations in the acoustics and the language. In section 2.2, an introduction to how the automatic speech recognition systems work and how the system is built with an acoustic model and a language model is presented. The acoustic model and the language model are presented in respectively section 2.3 and 2.4. Section 2.5 covers different speaker adaptation techniques, while feature extraction is covered in section 2.6. The basic theory of deep learning is presented in section 2.7, and transfer learning is covered in section 2.4. Parts of this chapter is taken from my specialization project report *Child Speech Recognition* from fall 2020 [1]. These sections are 2.1 - 2.3, 2.5 and 2.6 and are modified in this thesis.

## 2.1 Child Speech Characteristics

The speech signal of a child has higher pitch and formant frequencies and differs to a great extent from an adult's speech signal. This makes the recognition of child speech difficult, and the characteristics of the child speech have to be taken into account when implementing an automatic speech recognition (ASR) system for children. This section will look at the characteristics of both the acoustics and the language of child speech compared to adults. Analysis *by Lee and Russell* [2], which uses PSR (Primary School Reading) corpus from [3] and the TIMIT corpus [4], have been used to examine the differences in formant frequencies for children and adults. To investigate the differences in the duration of phonemes, analyses from *Gerosa el al.*[5] are utilized.

### 2.1.1 Acoustic analysis

The speaker produces a speech signal of air pressure waves, which consists of variations of the pressure as a function of time [6]. Two essential features in speech signals are the fundamental frequency and formant frequencies. The fundamental frequency is the lowest frequency of a periodic waveform, while formant frequencies are a measure of the resonances on the vocal tract [5]. It is sufficient to detect the formants F1, F2 and F3 along with the fundamental frequency F0 to represent the vocal tract characteristics.

**Duration of phonemes**

There are spectral and temporal changes in child speech. It has been reported that the duration of segments is longer for younger children compared to older children and adults [5]. Figure 2.1 illustrates the duration of the phones from age 7-13 and for adults, for comparison from ChildIt training set and IBN training set. We observe the duration of phonemes is longer for younger children. The phone duration may be affected by mispronunciations and time alignment errors in addition to the reading ability of the person.

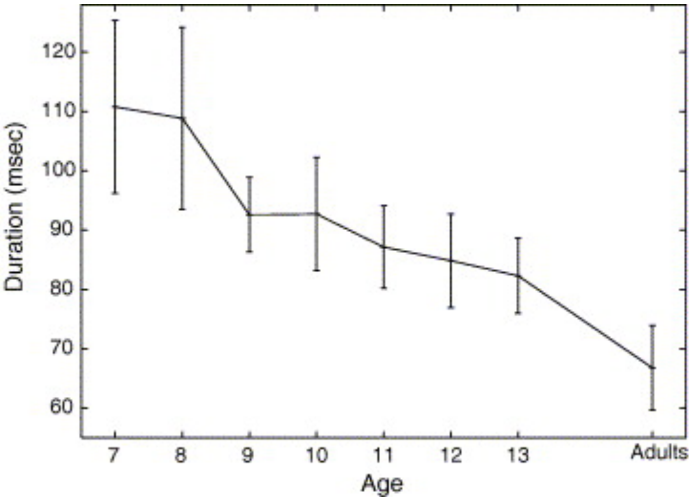

Figure 2.1: Figure is taken from [5]. Phone duration for children and adults in msec.

### Formant frequency

The formant frequencies vary with respect to the vocal tract length. The frequencies are higher when the vocal tract is shorter. A decrease of the formant frequencies occurs concurrently as the vocal tract has a progressive increase in length when the child grows. F1, F2 and F3 changes are more significant between the age of 5 to 7 and less marked between 9 to 11 years old. Under the age of 11, there is no significant difference in the vocal tract length for boys and girls of the same age. At the age of 15, the formant frequencies have become similar to adults, where females have a gradual decrease in formant frequencies up until this age. As males have a disproportional growth of the vocal tract during puberty, they have a substantial lowering in formant frequencies. The pitch of their speech signal reduces as a result of an enlarged glottis [5].

Table 2.1 shows the average vowel formant frequencies F1, F2 and F3 for children and adults. The average value of F1 for children is 942.1 Hz, which is 182.6 Hz higher than the average F1 value for adults. The average values of F2 and F3 are respectively 669.3Hz and 1008.5Hz higher for children than for adults [2].

Table 2.1: Average vowel formant frequency values for children's speech (PSR corpus) and adult speech (TIMIT) [2]

| **Vowel** | ***F1 Child (Hz)*** | ***F1 Adult (Hz)*** | ***F2 Child (Hz)*** | ***F2 Adult (Hz)*** | ***F3 Child (Hz)*** | ***F3 Adult (Hz)*** |
|---|---|---|---|---|---|---|
| IY | 1005 | 753 | 3299 | 2287 | 4478 | 3181 |
| IH | 883 | 740 | 2887 | 2057 | 4199 | 3082 |
| EH | 1152 | 760 | 2683 | 1917 | 4192 | 3025 |
| AE | 1334 | 812 | 2400 | 1952 | 3866 | 3058 |
| AH | 943 | 778 | 2305 | 1747 | 3958 | 3127 |
| AA | 1197 | 804 | 2251 | 1564 | 3943 | 2894 |
| AO | 674 | 727 | 1823 | 1448 | 4042 | 2908 |
| UH | 860 | 741 | 2224 | 1813 | 4108 | 3085 |
| UW | 704 | 683 | 2434 | 1678 | 3870 | 2975 |
| ER | 868 | 758 | 2448 | 1684 | 3959 | 2644 |
| AX | 1070 | 789 | 2476 | 1895 | 4085 | 3334 |
| EY | 924 | 743 | 3024 | 2132 | 4350 | 3097 |
| AW | 1033 | 808 | 2172 | 1653 | 3788 | 3010 |
| AY | 982 | 827 | 2508 | 1814 | 3971 | 3066 |
| OY | 636 | 713 | 2428 | 1590 | 3935 | 2930 |
| OW | 808 | 716 | 2188 | 1611 | 3836 | 3030 |

For F2 and F3, the sex differences are not apparent until the age of 15. To get a clearer understanding of the differences of the formant frequencies in respect of age and sex, we inspect figure 2.2. In this figure, the formant frequencies F1, F2 and F3 for males and females from the age of four until twenty are displayed. As stated above, the differences in the formant frequencies for each sex is not apparent for younger children. From the age of 14, it is observed that this difference increases between the sex as the formant frequencies of males decrease more than for females.

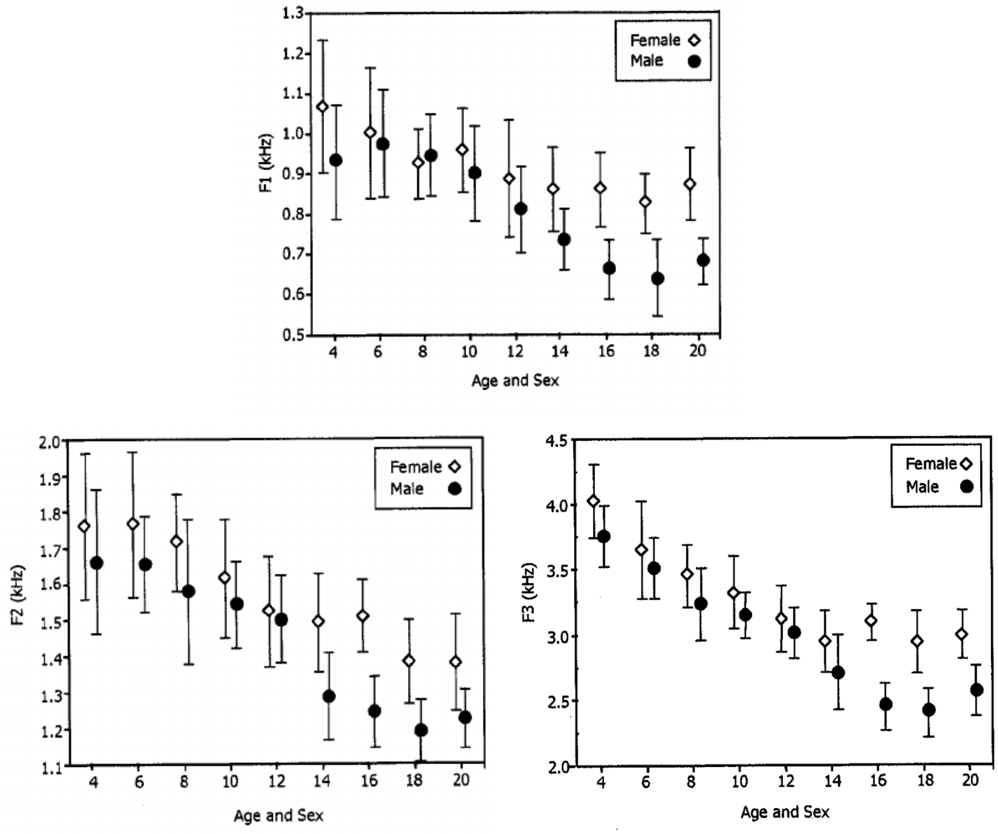

Figure 2.2: Figures taken from [7]. Differences in the formant frequencies for males and females.

### 2.1.2 Language analysis

Children have a less developed vocabulary than adults. This makes them produce more complicated sentences with words that are generally not connected. As their language is not fully developed, they also have more mistakes in the pronunciation of words compared to adults. Another characteristic is that children speak typically slower than adults. As children grow older, their vocabulary gets more extensive, and they have fewer mispronunciations. Around eight years old, all the speech sounds should be established, and children can produce complex and more complicated sentences. After this age, there will consequently be fewer complications regarding the language for a child speech targeted ASR system.

## 2.2 Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) is the process of deriving the transcription of an utterance, given the speech signal [8]. A general ASR system includes four main components and is illustrated in figure 2.3. These are feature extraction, the acoustic model, the language model and decoding.

The goal of automatic speech recognition is to predict the optimal word sequence **W**, given the spoken speech signal **X**, by maximizing the *a posteriori* probability (MAP) [9]:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\mathrm{argmax}} P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{X}), \tag{2.1}$$

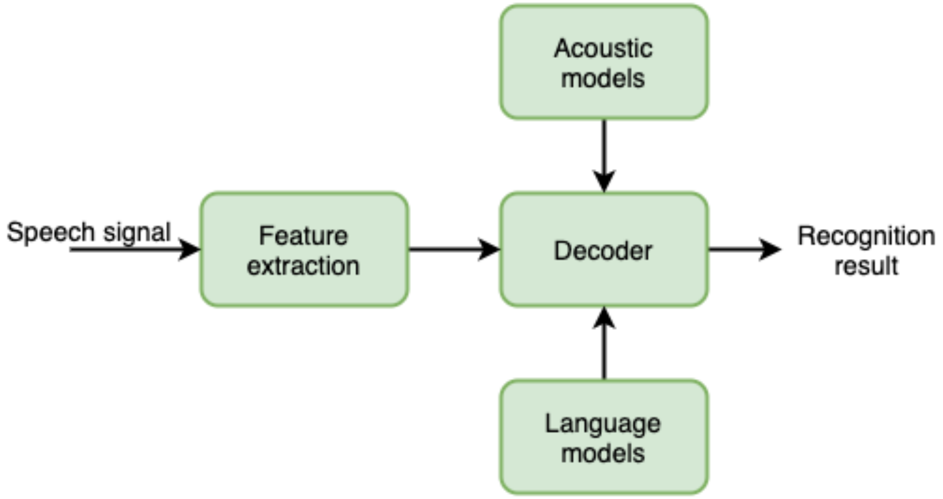

Figure 2.3: A general ASR system, with the four main components of a typical ASR system.

$\Lambda$ and $\Gamma$ are the parameters for the acoustic model and the language model, respectively. With Bayes' rule, we have,

$$P(\mathbf{W}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})} \tag{2.2}$$

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\text{argmax}} p_{\lambda}(\mathbf{X}|\mathbf{W})P_{\Gamma}(\mathbf{W}) \tag{2.3}$$

$p_{\Lambda}(\mathbf{X}|\mathbf{W})$ is the likelihood of the acoustic model, and $P_{\Gamma}(\mathbf{W})$ is the language model probability.

### 2.2.1 End-to-end ASR

End-to-end (E2E) models simplify ASR systems, as the acoustic models, pronunciation and language models are folded into one single neural network.

Recurrent neural networks (RNNs) are typically used in an E2E ASR system. These models are more suited for on-device applications as they are much smaller and less complex than conventional ASR systems [10].

## 2.3 Acoustic model (AM)

An acoustic model is used in speech recognition to model the speakers' pronunciation of the words in a sequence. The model utilizes speech samples and the corresponding transcriptions to learn.

### 2.3.1 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is the most common acoustic model. The HMM is a statistical model, which is represented as a stochastic model of discrete events and a variation of the Markov chain [11]. A Markov chain describes a sequence of possible states, where the probability of the following state only depends on the current state. The states in an HMM are hidden, and the state sequence produced by an HMM is not directly observable. This can only be approximated through another set of stochastic processes that produces the sequence of observations [1]. The HMM consists of the following elements [11]:

- Number of hidden states (N): Individual stated are represented as $S = (S_1, S_2, S_3, ..., S_N)$; the state at time $t$ is represented as $q_t$.
- State transition probability distribution: $\boldsymbol{P} = (p_{ij}$, to represent state transition from state $i$ to state $j$, where $p_{ij} = P(q_{t+1} = S_j | q_t = S_i)$, $1 \leq i$, $j \leq N$, $p_{ij} \geq 0$.

- Observation symbol probability distribution: ($\boldsymbol{B} = \{b_j(k)\}$) for state $j$, where $b_j(k) = P(x_t = o_k | q_t = S_j)$, $1 \le j \le N$, $1 \le k \le M$.
- Initial state distribution: ($\pi = \{\pi_i\}$), where $\pi_i = P(q_1 = S_1)$, $1 \le i \le N$.

A simple HMM model is illustrated in figure 2.4, where $s$ is the states, and $o$ is the observations.

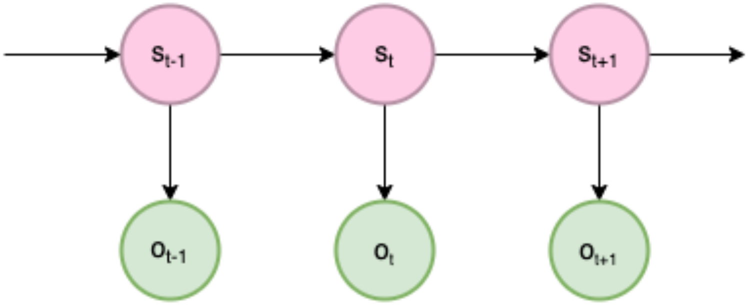

Figure 2.4: A simple HMM model.

### 2.3.2 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities [12]. In speech recognition systems, GMMs are often used to model the probability distribution of vocal tract related spectral features. Equation 2.4 gives the Gaussian mixture model, while the Gaussian density function is given by equation 2.5.

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} w_i g(\boldsymbol{x}|\mu_i, \sigma_i) \tag{2.4}$$

$$g(\boldsymbol{x}|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\sum_i|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \mu_i)^T \Sigma_i^{-1} (\boldsymbol{x} - \mu_i)\} \tag{2.5}$$

M is the number of Gaussian densities, while **x** is the values of the measurements. The mixture weights are given by $w_i$. $\mu_i$ is the mean vector, and the covariance matrix is defined by $\Sigma_i$. The mean vectors, covariance matrices and mixture weights are represented by $\lambda = \{w_i, \mu_i, \sigma_i\}$, where $i = 1, ..., M$ [12].

## 2.4 Language model (LM)

In ASR systems, a language model is used to improve the performance of the system. The language model learns to predict the probability of a given sequence of words that appear in a sentence.

### 2.4.1 N-gram language model

The n-gram language model is the simplest type of language models. An n-gram is a sequence of n words, where the probability of a word only depends on the previous $n$ words. If the sequence consists of only one word, it is called unigram, while for sequences consisting of two and three words, it is called bigrams and trigrams, respectively. If we have a sequence of words $P(w_1, w_2, ... w_n)$, the chain rule of probability can be applied to compute the probabilities of the sequence. Then we get,

$$\begin{aligned} P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1_2}...P(w_n|w_{1:n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_{1:k-1}) \end{aligned} \tag{2.6}$$

With a corpus of significant size, it can be a problem to estimate the probability of a word given all the history. Therefore, instead of computing the probability of a word given the entire history, the intuition of the n-gram model is to approximate the history by just the last few words [13]. With a bigram model, the probability of a word given the previous words is approximated by the conditional probability of only one preceding word. The trigram model approximates the probability by the two preceding words and goes further with the n-gram model.

## 2.5 Speaker adaptation

Speaker adaptation is techniques used to adapt an ASR system to a specific user. Different speaker adaptation techniques that are implemented in previous work are presented in this section.

### 2.5.1 Vocal Tract Length Normalization (VTLN)

Vocal tract length normalization (VTLN) is a common speaker adaptation method used in ASR. This takes the fact that the vocal tract length differs for each speaker and reduces the mismatch between the different speakers by normalizing the vocal tract length for each speaker. VTLN is normally implemented by frequency warping, where the warping factor is estimated to normalize the acoustic mismatch from the different vocal tract lengths. While performing feature extraction, the frequency axis is scaled according to the warping factor [14].

### 2.5.2 Maximum Likelihood Linear Regression (MLLR)

The maximum likelihood linear regression represents an adaptation technique based on a linear transform of the Gaussian mean or the variances in acoustic models [15]. This is used to reduce the mismatch between the adaptation data and the initial data [16]. The transformation matrix used to estimate the mean for the adaptation data is given by,

$$\hat{\mu}_j = W_j v_j, \tag{2.7}$$

$W_j$ is the $n \times (n+1)$ transformation matrix and $v_j$ is the extended mean vector $v_j = [1, \mu_{j1}, ..., \mu_{jn}]'$. The regression transformation is estimated to maximize the likelihood of the adaptation data [15].

### 2.5.3 Speaker Adaptive Training (SAT)

Speaker adaptive training is used to improve the speech recognition system and reduce the word error rate. The characteristics of each speaker are modeled by linear transformations of the mean parameters of the acoustic model [17]. The HMM parameters and speaker transformations are estimated by equation 2.8 to maximize the likelihood of the training data.

$$(\bar{\boldsymbol{\lambda}}_{\boldsymbol{c}}, \bar{\boldsymbol{g}}) = \operatorname*{argmax}_{\lambda_c, g} \prod_{r=1}^{R} L(O^{(r)}; \boldsymbol{G}^{(r)}(\boldsymbol{\lambda})) \tag{2.8}$$

$O^{(r)}$ is the training observation sequence and R is the speaker. $L(O^{(r)}; \boldsymbol{G}^{(r)}(\boldsymbol{\lambda})$ is the likelihood of the observations $O^{(r)}$ given the speaker dependent model $\boldsymbol{G}^{(r)}(\boldsymbol{\lambda})$ [18].

## 2.6 Feature extraction

In general, there is a lot of variation in a speech signal, especially between adult and child speech, as mentioned in chapter 2.1. To reduce this variability, feature extraction of the speech signal is performed. Mel frequency cepstral coefficient (MFCC) and perceptual linear prediction (PLP) are the two most common acoustic features used in speech recognition.

### 2.6.1 Mel Frequency Cepstral Coefficient (MFCC)

Preferred method of feature extracting is Mel Frequency Cepstral Coefficient (MFCC) extraction. This generates unique coefficients from the voice of each user. The technique to extract the MFCC's is illustrated in the block diagram in figure 2.5. The speech signal needs to be divided into smaller frames as it has to be examined over a short period of time to get stable acoustic characteristics. This is accomplished by windowing the signal, where each window are typically 20 ms. Hamming windows are commonly used for speech signals [1].

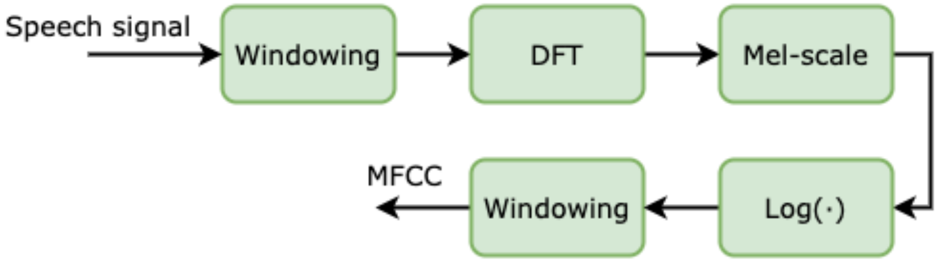

Figure 2.5: MFCC extraction

Each frame is then converted into the magnitude spectrum by applying discrete Fourier transform (2.9), where N is the number of points used to compute the DFT [19].

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp \frac{-j2\pi nk}{N}; \quad 0 < k < N-1 \tag{2.9}$$

Equation 2.10 gives the mel-frequency calculated by the frequency *f*. The signal is passed through mel-filter banks. The mel-frequency is scaled to match what the human ear can hear, and it is not linear to the physical frequency of the tone. Figure 2.6 shows the frequency in mel to the frequency in Hz.

$$f_{mel} = 2595 \cdot log(1 + \frac{f}{700Hz}) \tag{2.10}$$

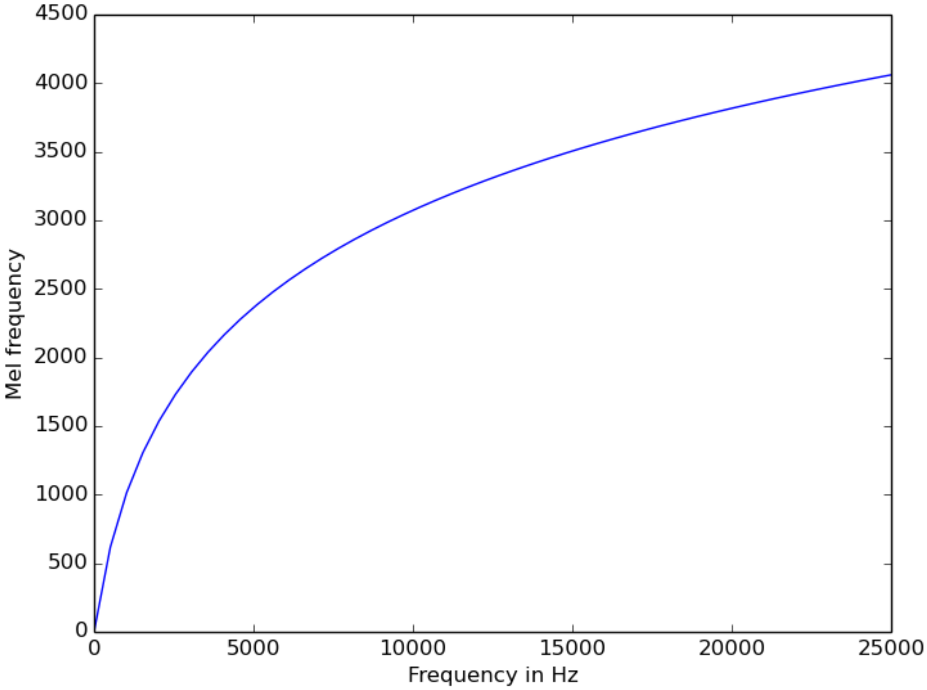

Figure 2.6: Mel-scale. Frequency in Hz vs. Mel frequency, from equation 2.10.

The mel-spectrum is then computed by,

$$s(m) = \sum_{k=0}^{N-1} \Big[ |X(k)|^2 H_m(k) \Big]; 0 \leq m \leq M-1. \tag{2.11}$$

M is the total number of filters and $H_m(k)$ are the filterbanks given by equation 2.12 where m is in the range 0 to M [19].

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{2(k-f(m-1))}{f(m)-f(m-1)}, & f(m-1) \leq k \leq f(m) \\ \frac{2(f(m+1)-k)}{f(m+1)-f(m)}, & f(m) \leq k \leq f(m+1) \\ 0, & k > f(m+1) \end{cases} \tag{2.12}$$

The filter banks are implemented in the frequency domain, where triangular filter banks are the most common. The mel-filter bank is illustrated in figure 2.7. The logarithm of the signal is applied after mel-scaling, and the Discrete Cosine Transform (DCT) is applied to produce a set of cepstral coefficients. MFCC is calculated as,

$$c(n) = \sum_{m=0}^{M-1} log_{10}(s(m)) cos\Big(\frac{\pi n(m-0.5)}{M}\Big); n = 0, 1, 2, ..., C-1, \tag{2.13}$$

where c(n) is the cepstral coefficients and C is the number of MFCCs.

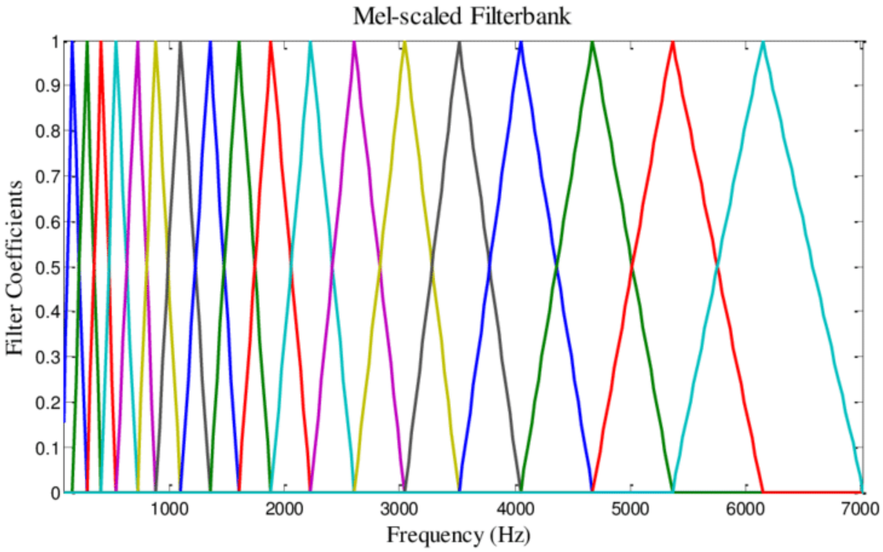

Figure 2.7: Mel-filter bank [20]

### 2.6.2 Perceptual Linear Prediction (PLP)

Perceptual linear prediction (PLP) is similar to MFCC, but are more robust when there is a mismatch between training and test data [21]. In PLP, the windowed speech signal is used to compute the power spectrum before a Bark filter-bank is applied to the power spectrum. To simulate the sensitivity of the hearing, the outputs of the filter-bank are weighted with an equal-loudness pre-emphasis. After transforming the equalized values with the power of 0.33, linear prediction (LP) is applied. Cepstral coefficients are then calculated [22].

## 2.7 Deep learning

Deep learning is a subset of machine learning, and models the human brain to process data. The algorithms in deep learning are called neural networks, consisting of a multi-layered structure based on the structure of the human brain with connected neuron nodes. Neural networks consist of both visible and hidden layers, where the first layer is the input layer and contains input neurons. These input neurons consist of the input data of the neural network. The output layer is the final layer which consists of output neurons, and gives the results of the calculations from the inputs of the output layer. The layers in between the input and output layers are the hidden layers, which do the computations on the input data. Figure 2.8 illustrates a neural network that consists of an input layer of three input neurons, two hidden layers, and an output layer with two output neurons. Every node in each layer are connected with the nodes in the adjacent layers [23], and each layers processes the information from the previous layer.

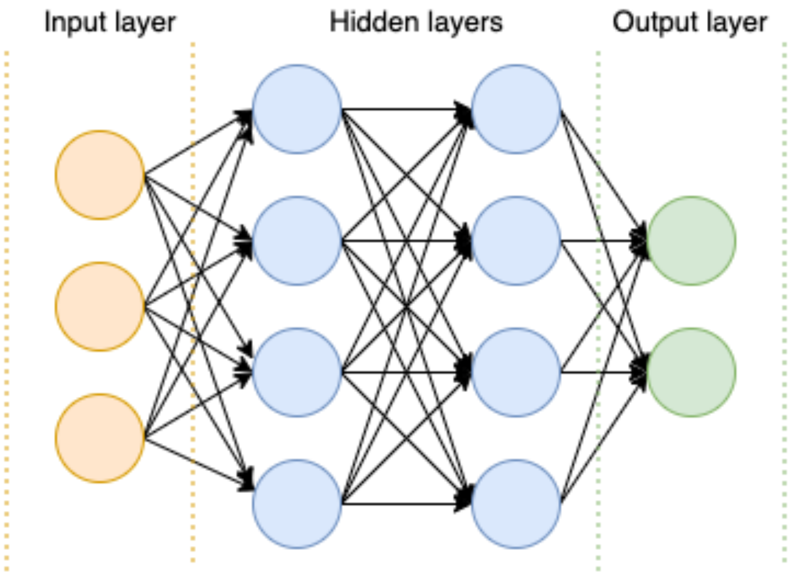

Figure 2.8: Neural network with input layer, two hidden layers and output layer.

### 2.7.1 Recurrent Neural Network (RNN)

Most machine learning models are not designed for sequential data. However, recurrent neural networks (RNNs) are designed to deal with this specific type of data of variable length. A recurrent neural network is an artificial neural network with internal loops [24], which makes decisions based on what it has learnt from the past. The RNN utilizes training data to learn, in addition to remembering what the network has learnt by the previous inputs while generating outputs [25]. Figure 2.9 illustrates a basic RNN structure.

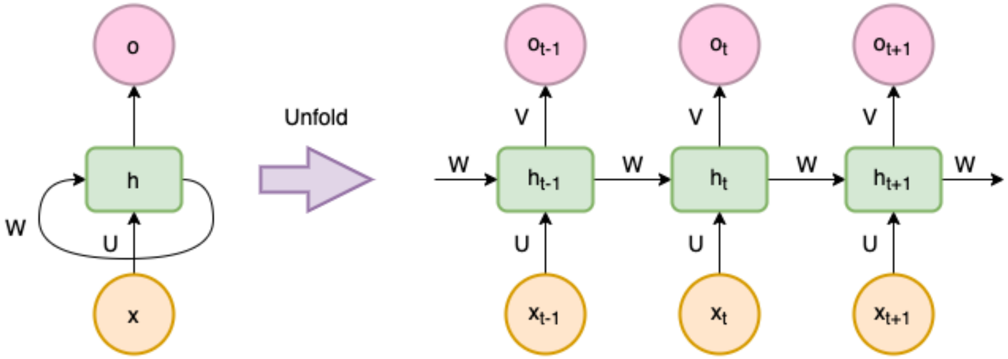

Figure 2.9: Basic RNN structure, adapted from [26].

$x$ is the inputs, $o$ the outputs, $h$ is the main block of the RNN, which contains the weights and activation function of the network, while $W$ is the communication from one step to another [26].

A problem with basic RNNs is the long-term dependencies. If the gap between the relevant information and the predicted word is large, RNN models have trouble connecting the information. An RNN that is capable of learning long-term dependencies is the long short-term memory (LSTM) networks.

## 2.8 Transfer learning

Transfer learning is a technique in machine learning where a model trained on a previous task is reused for a new task. The definition of transfer learning is:

*"Given a source domain $D_S$ and learning task $T_S$, a target domain $D_T$ and learning task $T_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $D_T$ using the knowledge in $D_S$ and $T_S$, where $D_S \neq D_T$, or $T_S \neq T_T$"* [27].

Generally, a lot of data are needed to train a neural network from scratch. In many cases, access to all the data required for training the neural network is not available. Transfer learning makes it possible to train a deep neural network with relatively small datasets, and it also makes the training process much shorter. The weights the network has learned from one task is transferred to another task. Transfer learning is widely used in computer vision, natural language processing and speech recognition tasks as they require a large amount of computational power [28]. The process of transfer learning is illustrated in figure 2.10.

The most common approach of transfer learning is using a pre-trained model. A pre-trained model has been trained on a large dataset to solve a problem similar to the problem that will be solved. It has to be decided which layers are going to be frozen and which to train due to the problem [29].

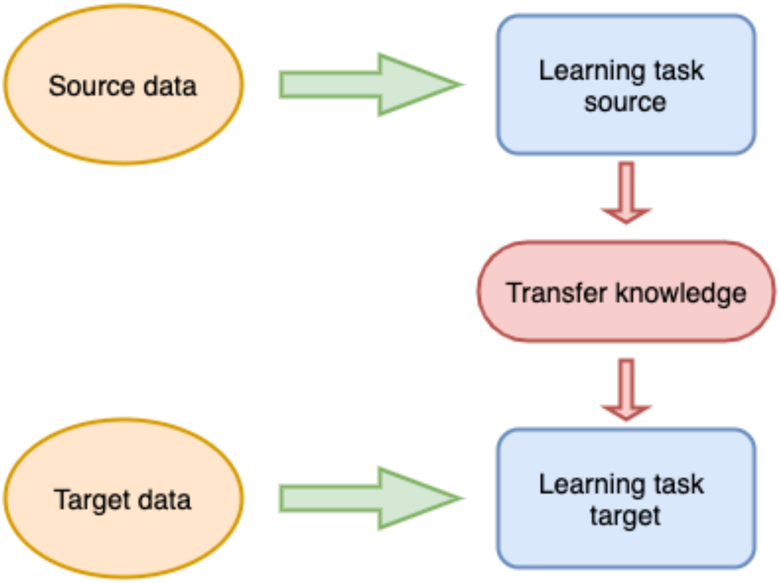

Figure 2.10: Transfer learning

# Chapter 3

# Related work

Previously, there have been many approaches to improving children's speech recognition systems. The state of the art ASR systems have for long been based on HMM models, which have also been the case for ASR systems for child speech. Later, HMMs have been used together with DNNs as hybrid models, where the parameters of the HMM have been calculated by the HMM. There have been tested several techniques to handle the acoustic variability in the child speech for these systems. Mel-Frequency Cepstral Coefficients (MFCC), spectrum-based filter bank features and Perceptual Linear Prediction (PLP) have been investigated by Shivakumar *et al.* [30]. All of these have shown an effect in decreasing the WER in the ASR for children, although MFCC features have dominated due to their robustness and compatibility with adult ASR systems [31]. Elenius and Bloomberg [32], Shivakumar *el al.* [30] and Gray *el al.*[33] have showed that adapting acoustic models with Maximum Likelihood Linear Regression (MLLR) and Maximum A-posteriori (MAP) was found to be effective. Speaker Adaptive Training (SAT) has shown an increase in the performance for children ASR.

Studies have reported that Vocal Tract Length Normalization (VTLN) effectively improves speech recognition performance on child speech with limited data. Liao *et al.* [34] did research on large vocabulary automatic speech recognition for children, where these results show no effect in spectral smoothing, VTLN and using pitch features, but argues with the fact that their system was trained on a large amount of data.

As the HMM-DNN systems are complicated to design, the use of DNN models has become more frequently used these days. The use of DNNs for ASR systems needs a lot of computing power, which has become possible in later days because of the opportunity to use graphic processing units (GPUs). In 20014, Baidu research presented an end-to-end state-of-the-art ASR system called Deep Speech 1 (DS1) [35]. They intended to make a well-optimised RNN training system. This aims to make an ASR system more simple, as traditional systems use heavy engineered processing stages. The RNN in DS1 consists of five hidden layers, with one recurrent layer. They present a deep learning-based end-to-end ASR system that outperforms existing state-of-the-art recognition systems in clear speech and noisy speech. Later, Baidu presented Deep Speech 2 (DS2), a new end-to-end deep learning approach that is used to recognise either English or Mandarin Chinese speech, two extensively different languages [36]. In DS2, they explore architectures with up to 11 hidden layers, with several recurrent layers and convolutional layers. The result of this is a system that achieves seven times speedup from DS1, and the system can even compete with the transcriptions of humans. The architecture of the DS1 system is further explained in chapter 5.1.

Most of the prior work on child speech recognition is based on GMM-HMMs, and less work has investigated DNNs for children's speech due to lack of large amounts of children's training data. However, there has been some research on applying transfer learning to the DNN models to adapt a model from adult speech to child speech. In Shivakumar and Gergiou's study on transfer learning from adult speech to children's speech [31], they compare the advantages of DNN acoustic models over the GMM-HMM systems, and the performance of the DNN acoustic model for adult and children. These studies validated the benefits of age-dependent transfer learning. Research by Tong *et al.*[37] shows that transfer learning can be an efficient way of improving an ASR system for children. They investigate acoustic adaptation and multi-task learning methods, where they take advantage of the adults' speech information and transfer them to children's ASR. Both approaches show to have an effect of improvement on child speech, where multi-task learning shows most effect. These results indicate that the performance of children's speech recognition systems can be benefited from available adult speech corpus.

# Chapter 4

# Tools and data

This chapter includes an overview of the tools and data used in this thesis. The corpus is presented, and a description of what the datasets consist of is given.

## 4.1 Deep Speech by Mozilla

Deep Speech is an open-source speech-to-text engine, using a model trained by machine learning techniques based on Baidu's Deep Speech 1 (DS1) research paper. The network is implemented by using Tensorflow[1]. The code and models are published under the Mozilla Public License 2.0.

## 4.2 CMU Kids Corpus

CMU Kids Corpus is a database consisting of speech from children in the age from six to eleven years old. The speakers are 24 boys and 52 girls, with a total of 5180 utterances. The speech is from children reading aloud from a four-page

[1] https://www.tensorflow.org/

colour reading supplement and is divided into good and poor readers. One wavefile corresponds to one sentence as there were presented one sentence at a time. The set of good readers consists of 44 speakers and 3333 utterances, while the set of poor readers has 32 speakers and 1847 utterances [38]. The sample rate of the speech signals is 16kHz.

Table 4.1: CMU Kids Corpus

| Data | Speakers | Utterances | Age |
|---|---|---|---|
| All | 76 | 5180 | 6-11 |
| Good readers | 44 | 3333 | 6-11 |
| Poor readers | 32 | 1847 | 6-11 |

## 4.3 Datasets

The dataset consists of training set, development set and test set. The training set is used to train the system, and the development set is used to validate the training, while the test set is used after training to test the system. The training set consists of 80% of the whole dataset, while the development and test set consist of 10% each. The implementation in this project does not use phonemes as conventional ASR systems do. Some of the CMU Kids corpus data is poor, where the sentences contain one or more divergences from the intended utterance. The transcriptions of these do then consist of phonemes and not only clean text. These data cannot be applied to this implementation without major changes in the transcriptions. Therefore, this part of the dataset has been removed. Another reason this data was removed, was to not train and test the model on data that contains error, as this would make the system's performance poor.

# Chapter 5

# Method

This chapter covers the implementation of the Deep Speech model from Mozilla in section 5.1, which is the basis for the ASR system. In section 5.2, the data preparation is covered, while the experimental setup is presented in section 5.3. This covers the different methods of improving the ASR system for child speech as transfer learning and fine-tuning. The different evaluation methods applied to this system is introduced in section 5.4.

## 5.1 Deep Speech model

The Deep Speech model is inspired by the architecture of Baidu's research, Deep Speech: Scaling up end-to-end speech recognition [39]. This architecture is more simple than the traditional architectures for speech recognition and does not need a phoneme directory. It is also more robust for noisy environments, as the model learns directly how to handle the noise while training. The model uses a recurrent neural network (RNN) with speech spectrograms as input and generates English text transcriptions as output. The model is illustrated in figure 5.1.

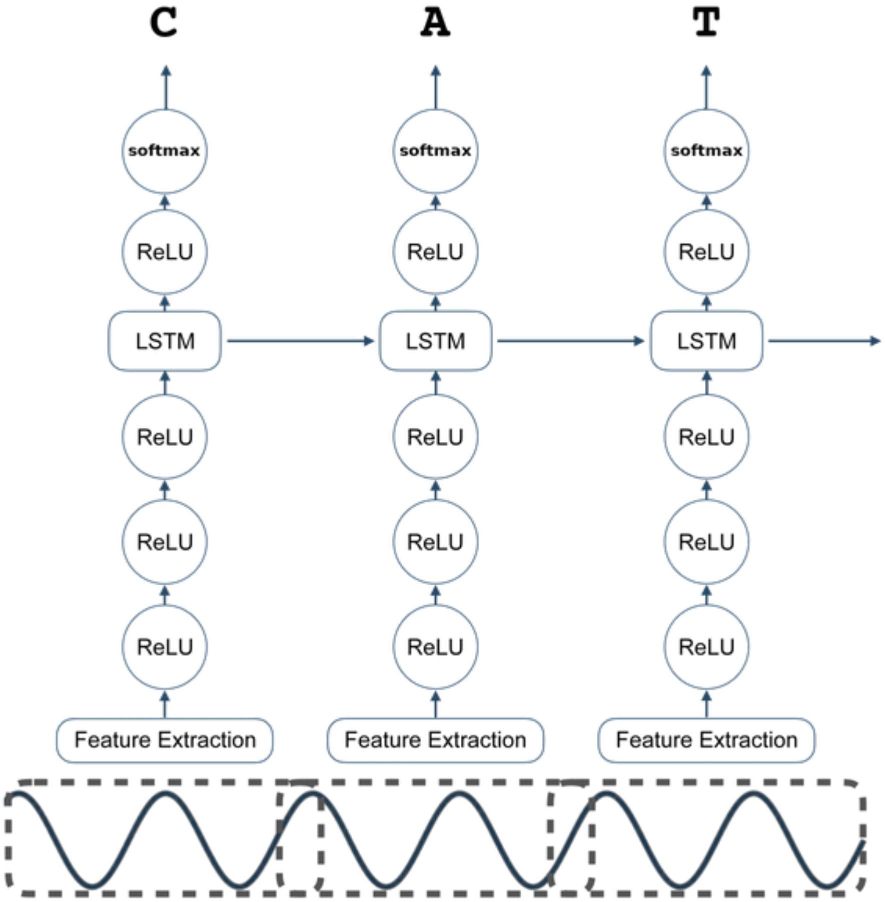

Figure 5.1: Illustration of the Deep Speech model[1]

The training set in Deep Speech is defined as follows,

$$T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots\}, \tag{5.1}$$

where $x^{(i)}$ is the speech utterance in time-series of length $t^{(i)}$, and $y^{(i)}$ is

[1] Taken from https://deepspeech.readthedocs.io/en/v0.9.3/DeepSpeech.html

the transcription of the speech signal $x^{(i)}$. The MFCCs are calculated to extract the features from the acoustic signal. As the sampling rate of the data used in this project is 16kHz, the number of MFCCs used are 26.

As illustrated in figure 5.1, the RNN model consists of five hidden layers, one input layer and one output layer. The nodes of each hidden layer are denoted $h^{(l)}$, at layer $l$, where the first three layers are not recurrent. The input layer is $h^{(0)}$, while $h^{(6)}$ is the output. The non-recurrent layers $l = [1,2,3]$ are calculated by

$$h_t^{(l)} = g(W^{(l)})h_t^{l-1} + b^{(l)}), \tag{5.2}$$

where the clipped rectified linear (ReLu) activation function $g(z)$ is given by $min\{max\{0,z\},20\}$. The weight matrix is $W^{(l)}$, while $b^{(l)}$ is the bias parameters. The fourth layer is the recurrent layer, which consists of hidden nodes with forward recurrence, calculated sequentially from $t = 1$ to $t = T^{(i)}$ for utterance i, by equation 5.3. The model in figure 5.1 illustrates that this layer is implemented with LSTM.

$$h_t^{(f)} = g(W^{(4)}h_t^{(3)} + W_r^{(f)}h_{t-1}^{(f)} + b^{(4)}) \tag{5.3}$$

The inputs of layer five is the forward nodes from the recurrent layer,

$$h^{(5)} = g(W^{(5)}h^{(f)} + b^{(5)}) \tag{5.4}$$

The output and the final layer predicts the probabilities for each characters $k$ in the alphabet at time $t$, and is calculated by the standard softmax function,

$$h_{t,k}^{(6)} = \hat{y}_{t,k} = \frac{exp(W_k^{(6)} h_t^{(5)} + b_k^{(6)})}{\sum_j exp(W_j^{(6)} h_t^{(5)} + b_j^{(6)})} \tag{5.5}$$

The CTC $\mathcal{L}(\hat{y}, y)$ loss is then computed to get a measure of the error in the prediction. CTC loss is defined in section 5.4. The gradient $\Delta\mathcal{L}(\hat{y}, y)$ is evaluated during training, where the Adam optimisation [40] is utilized.

After training, the model is re-scored with a language model. The Deep Speech pre-trained model includes an external scorer, which consists of a KenLM [41] language model and a sequence of all the words in the vocabulary.

## 5.2 Data preparation

Because of the way Deep Speech is implemented, and its easy way of training and testing with your own data (illustrated in figure 5.1), the data has to be divided into training, development and test sets before interfering with Deep Speech. The import script for the CMU Kids data has been implemented with inspiration from the *import_timit.py*[2] script, which imports the TIMIT dataset. The process used to implement this needed the data to be divided into separate folders with the training, development and test data.

Table 5.1: Example of running DeepSpeech[3]

```
python3 DeepSpeech.py
—train_files ../data/CV/en/clips/train.csv
—dev_files ../data/CV/en/clips/dev.csv
—test_files ../data/CV/en/clips/test.csv
```

[2] https://github.com/mozilla/DeepSpeech/tree/master/bin

[3] https://deepspeech.readthedocs.io/en/v0.9.3/TRAINING.html

As the CMU Kids corpus is not previously divided into train, development and test sets, this has to be proceeded. The CMU Kids corpus is divided into kids1 and kids2, consisting of one folder for each speaker. Each folder for every speaker contains label files, point files, signals and transcriptions. The label files contain a description of the phonetic segments, while the point files consist of comments on the phonetic divergence. As Deep Speech does not use phonemes, the only files needed for this implementation are the signals and the transcriptions. When splitting the files, it is important that the signal file and the affiliated transcription file is in the same set. An example of the data is illustrated in table 5.2 and 5.3 for respectively the signal files and the transcription files.

Table 5.2: Example of signal files split into train, dev and test sets

| | Path to signal file |
|---|---|
| Before split | cmu_kids/kids1/fabm/signal/fabm2aa1.sph<br>cmu_kids/kids2/fmtg/signal/fmtg1ap1.sph<br>cmu_kids/kids2/mjwm/signal/mjwm2aa1.sph |
| After split | cmu_kids/train/kids1/fabm/signal/fabm2aa1.sph<br>cmu_kids/dev/kids2/fmtg/signal/fmtg1ap1.sph<br>cmu_kids/test/kids2/mjwm/signal/mjwm2aa1.sph |

Table 5.3: Example of transcriptions files split into train, dev and test sets

| | Path to transcription file |
|---|---|
| Before split | cmu_kids/kids1/fabm/trans/fabm2aa1.trn<br>cmu_kids/kids2/fmtg/trans/fmtg1ap1.trn<br>cmu_kids/kids2/mjwm/trans/mjwm2aa1.trn |
| After split | cmu_kids/train/kids1/fabm/trans/fabm2aa1.trn<br>cmu_kids/dev/kids2/fmtg/trans/fmtg1ap1.trn<br>cmu_kids/test/kids2/mjwm/trans/mjwm2aa1.trn |

The importation of the data takes the speech signal and the transcription of the signal and makes CSV files for each train, development and test sets. The name of these files are respectively *cmu_kids_train.csv*, *cmu_kids_dev.csv* and *cmu_kids_test.csv*. These files comprise of *wav_filename*, which is the path to the signal file, *wav_filesize*, the size of the sample in bytes. The last part is *transcript*, which is the transcription of the sample. An example of the content is given in figure 5.4, which is a sample of five of the files in the CMU Kids test set.

Table 5.4: Sample of *cmu_kids_test.csv*

| wav_filename | wav_filesize | transcript |
|---|---|---|
| ../test/kids2/mglp/signal/mglp1bc1_rif.wav | 214440 | then not as much trash piles up |
| ../test/kids2/mglp/signal/mglp1bd1_rif.wav | 214440 | some people recycle food garbage |
| ../test/kids2/mglp/signal/mglp1bg1_rif.wav | 240040 | the soil is used to help gardens grow |
| ../test/kids2/mglp/signal/mglp1bh1_rif.wav | 217640 | some paper has an important sign on it |
| ../test/kids2/mglp/signal/mglp1bl1_rif.wav | 281640 | some workers recycle bikes |

## 5.3 Experiments

A transfer learning model has been implemented to create an ASR system for children. To check the reliability of the pre-trained model, this model has been tested with the *TIMIT test corpus*. Before any further training is done, the pre-trained model is tested with the *CMU Kids corpus*, in addition to training a model from scratch with the child speech data to compare the results. All the same data has been used for each experiment. Fine-tuning is performed on both the transfer learning model and the pre-trained model trained with the *CMU Kids corpus.*

After each epoch, the models were tested on the development set. This is to establish if the model improved through further training and to avoid overfitting. The weights from the best validation are saved. This allows the best possible results to be obtained during testing, and avoid using results from when a model starts overfitting. 2048 hidden units are used for all the experiments, as this is what the release model is trained with. For all the experiments, the aulus3 server at NTNU is used. This server has two NVIDIA GeForce RTX 2080 Ti GPUs, which is crucial for the training process to complete promptly.

### 5.3.1 Baseline

The CMU Kids data has been trained from scratch in Deep Speech. The training and test sets this model has been applied to are the same data for training and testing the transfer learning and fine-tuning models. This is done to get the most out of the comparability of the results. As this experiment does not train on a pre-trained model, the number of epochs is required to be higher than for the other experiments. The number of epochs used is 100. To optimize the efficiency of the training, automatic mixed precision is utilized, and the training batch size is set to 60.

### 5.3.2 Transfer learning

As there are not enough child speech data to train an accurate automatic speech recognition system for children, transfer learning (TL) has been used to improve the ASR system. The pre-trained model of Mozilla from release 0.9.3[4] was used for this implementation. This model is trained on American English with more than 5500 hours of data [39]. The test of this model shows a WER of 7.06% on the *LibriSpeech clean test corpus*. The model has best performance in low-noise environments.

The architecture of the TL needs to be identical to the pre-trained model. The layers from the pre-trained model for adult voices were not frozen during training, as the other layers then can adjust to fit the child speech model. Four TL implementations were investigated in this experiment. In the first TL implementation, the weights of only the output layer were re-initialised. This layer was then trained from scratch, while the remaining layers were retrained based on the previous weights. The second and third TL implementation re-initialised the two and three final layers, respectively. In the last TL implementation, the LSTM layer and layer five were re-initialised. As mentioned, all the weights from the remaining layers were re-used and trained further. The architecture of the models are illustrated in figure 5.2, 5.3, 5.4 and figure 5.5. These were all trained over 100 epochs, where the learning rate was reduced with a factor of 0.1 if no improvement in the validation loss after 10 epochs. This is done to prevent overfitting during training from the re-used weights.

[4] https://github.com/mozilla/DeepSpeech/releases/tag/v0.9.3

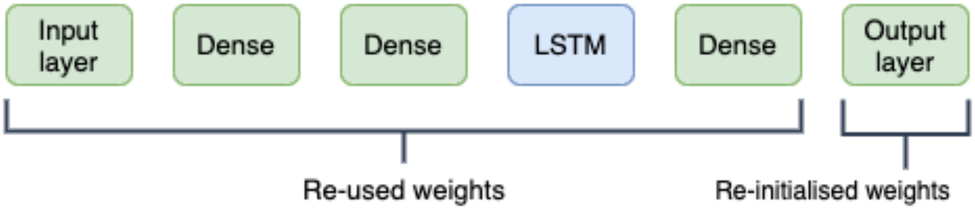

Figure 5.2: Architecture of TL model with output layer re-initialised

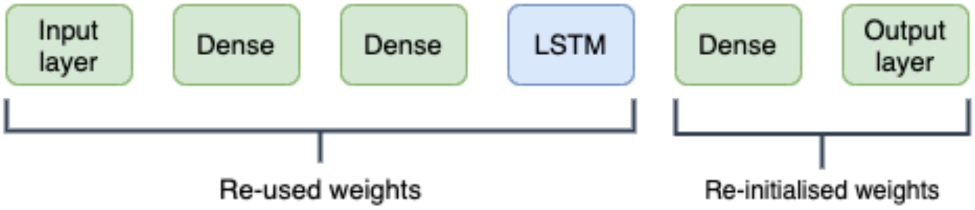

Figure 5.3: Architecture of TL model with the two last layers re-initialised

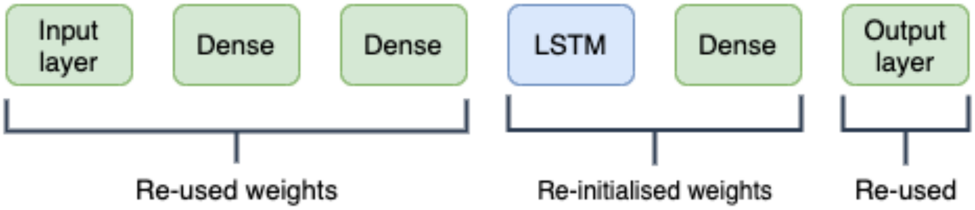

Figure 5.4: Architecture of TL model with the LSTM layer and layer five re-initialised

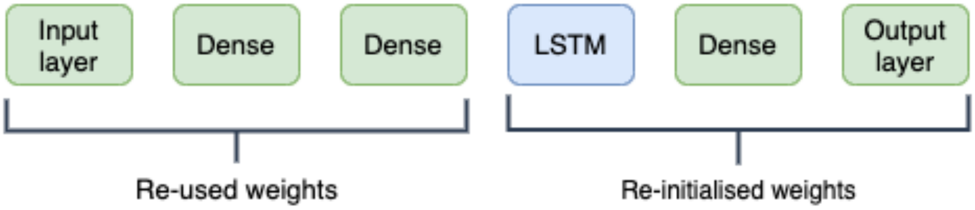

Figure 5.5: Architecture of TL model with the LSTM layer, layer five and output layer re-initialised

### 5.3.3 Fine-tuning

In addition to the transfer learning method used in the section above, the technique of acoustic adaptation was applied. By this, the parameters are fine-tuned to fit the new data. This was applied both directly on the pre-trained model and to the transfer learning models. Figure 5.6 illustrates the acoustic adaptation. There were done tests with different learning rates and dropout rates. If there were no improvement in the loss after ten epochs, the learning rate was reduced by the factor of 0.1. All the training with fine-tuning was done over 30 epochs.

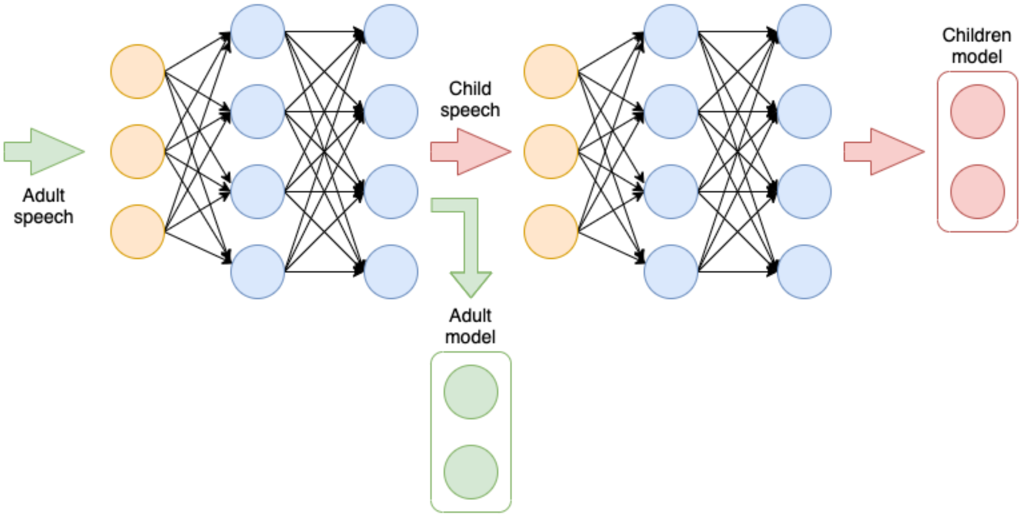

Figure 5.6: Model of acoustic adaptation. Illustration adapted from [37].

## 5.4 Evaluation

The methods of evaluating the ASR system in DeepSpeech are the three metrics word error rate (WER), character error rate (CER) and CTC loss. WER is the most important approach and will therefore be the evaluation metric focused on in this thesis.

### 5.4.1 Word Error Rate (WER)

An ASR system used on child speech usually has a high WER. WER is a measure of the effectiveness of an automatic speech recognition system [42] and is calculated by,

$$WER = \frac{S+I+D}{N}, \tag{5.6}$$

where S is the number of substitutions, I the number of insertions and D the deletions. N is the total number of words in the transcript. The WER is a measure of how well the language model performs, where it recognises words. If an ASR system has a high WER, the words are not being detected accurately, and the system is more unreliable than an ASR system with a lower WER.

### 5.4.2 Character Error Rate (CER)

CER is measured the same way as WER, with characters instead of words, as shown in equation 5.7, where N is then the number of characters.

$$CER = \frac{S+I+D}{N}. \tag{5.7}$$

The CER is a measure of how well the acoustic model performs, and recognises characters. The measure is often lower than the WER, as one small character error in the sentence will make the whole word be mistaken.

### 5.4.3 CTC Loss

The loss function used in Deep Speech is the Connectionist Temporal Classification (CTC). This method is well suited to speech recognition, as it is not required to know the alignment between the transcription and the audio. Without any language models, the loss function specifies the performance of the acoustic model. The CTC loss is computed to get a measure of the error in the prediction during training,

$$\mathcal{L} = -log p(y|x), \tag{5.8}$$

where x and y is defined as in section 5.1

# Chapter 6

# Results

In this chapter, all the results from the different experiments are reviewed. Section 6.1 covers the results from the system trained from scratch. Section 6.2 and 6.3 gives the results from testing the TIMIT dataset and the CMU Kids dataset on the pre-trained model without further training. In section 6.4, the transfer learning (TL) results are presented, while section 6.5 covers the results from the fine-tuning.

## 6.1 Baseline

Table 6.1 shows the results from the system training on the child speech data from scratch. As the training set consists of a very low amount of data, these results show a poor performance with a WER of 93.84%. This means that only 6.16% of all the words in the test set are recognised by this model.

Table 6.1: Results of model trained from scratch.

| **WER** | **CER** | **Loss** |
|---|---|---|
| 0.9384 | 0.4699 | 70.548 |

At the end of each test, five examples each of the best achieved WERs, the median WERs and the worst WERs are printed. The median WERs is the one that best illustrates how well the overall model works, while the best WERs and the worst WERs show the five best results and the five worst results. To get a better understanding of the results, three examples, one of each are illustrated in table 6.2, 6.3 and 6.4. These illustrate the WER and the CER in addition to the text that is predicted while testing and the source transcript, described with "*src*" and "*res*". As we observe from these results, the best WER is the only acceptable result, where it is possible to understand the recognised words. This still has a rather high WER of 33.33%, as it is a sentence that consists of only three words, where one is "wrong". However, there is only one character error, which makes the last word differ from the actual word. The results with median WER and the worst WER gives results that do not give much sense, as they cannot predict any of the words. The example of the worst WER even has a WER higher than 1, as there are added words compared to the transcription. This illustrates how poor the performance is when training on a really low amount of data, and how the inference results might end up if the WER is too high.

Table 6.2: Best WER trained from scratch

| **Best WER**: |
|---|
| WER: 0.3333, CER: 0.0500 |
| src: " people built houses"<br>res: " people built thouses" |

Table 6.3: Median WER trained from scratch

| **Median WER**: |
|---|
| WER: 1.0000, CER: 0.5000 |
| src: " rain forests have the most different kinds of butterflies"<br>res: " ring fore se on os tirtren ouunof botef les" |

Table 6.4: Worst WER trained from scratch

| **Worst WER**: |
|---|
| WER: 2.0000, CER: 1.1852 |
| src: " some noises were very loud"<br>res: " th some tcishe s shopstc ie fowe rost fa ts ors" |

## 6.2 Testing TIMIT on pre-trained adult model

To check the reliability of the pre-trained model, the adult data from the TIMIT test set is used for testing. As the accuracy of ASR systems often is lower with test set it is not trained on, it can not be directly compared to check the reliability, but it still gives us a prediction on how well it works. The results of the test on the TIMIT dataset are shown in table 6.5. The WER is 27.68%, which is relatively high to be used as an ASR system. This is expected to be lower than the WER of the test of child speech data on the pre-trained adult model.

Table 6.5: Pre-trained adult model tested with TIMIT test set

| **WER** | **CER** | **Loss** |
|---|---|---|
| 0.276756 | 0.104837 | 18.331354 |

## 6.3 Testing CMU Kids on pre-trained adult model

Table 6.6 shows the result of the metrics WER, CER and Loss after testing the CMU Kids data on the pre-trained model. As expected, these results are significantly better than the results from the system trained from scratch, with a WER of 48.01%. The WER is still quite high as a result of that the model is only trained on speech from adults. This is not suitable for use for inference as almost half of the words are not recognised.

Table 6.6: CMU Kids data tested on pre-trained adult model

| **WER** | **CER** | **Loss** |
|---|---|---|
| 0.480119 | 0.2292 | 35.7264 |

The test shows that this model recognises some of the audio as the best WER is 0, which means that all the words in this sentence are recognised. There is still a lot that does not recognise correctly, which makes the overall WER high, as the example of the median WER indicates. The example from the sentence with the worst WER recognises only one word correctly, which is eggs. These results are illustrated in table 6.7, 6.8 and 6.9.

Table 6.7: Example of best WER from test on pre-trained model

| **Best WER**: |
|---|
| WER: 0.0000, CER: 0.02273 |
| src: " a mother butterfly lays tiny eggs on a leaf"<br>res: "a mother butterfly lays tiny eggs on a leaf" |

Table 6.8: Example of median WER from test on pre-trained model

| **Median WER**: |
|---|
| WER: 0.4286, CER: 0.2500 |
| src: " then not as much trash piles up"<br>res: "then dat as much tresh powse up" |

Table 6.9: Example of worst WER from test on pre-trained model

| **Worst WER**: |
|---|
| WER: 1.6000, CER: 0.6667 |
| src: " one butterfly can lay eggs"<br>res: " " wone bu e fly tan lad two hundred eggs" |

## 6.4 Transfer learning

The results of the transfer learning experiments are presented in table 6.10. These show how the re-initialising of the different layers are improving the model adapted to the child speech. The results are presented with the description TL with the following layers that are re-initialised. If the description says TL layer 5 & 6, this means that layer 5 and the output layer 6 are re-initialised. As observed from the table, transfer learning of the two final layers 5 and 6 gives better results than transfer learning of just the output layer. When re-initialising the weights of the output layer, the WER decreases from 48.01% to 44.14%, while transfer learning of two layers makes the WER decrease to 38.27%. Transfer learning of the LSTM layer and layer 5 gives even better results. The WER is then reduced to 36.68%. When re-initialising all the three final layers, the results of WER of 37.97% show a slightly better performance than the TL of the two final layers, but a little worse than TL of the LSTM layer and layer 5. All the transfer learning experiments gives an improvement from the pre-trained adult model, but the WER is still high for all of them as all results show a WER higher than 36%.

Table 6.10: Transfer learning results

| Model | WER | CER | Loss |
|---|---|---|---|
| TL layer 6 | 0.441352 | 0.214710 | 30.968922 |
| TL layer 5 & 6 | 0.382704 | 0.161218 | 22.940805 |
| TL LSTM layer & layer 5 | 0.366799 | 0.156947 | 22.222095 |
| TL LSTM layer & layer 5 & 6 | 0.379722 | 0.159547 | 22.766584 |

## 6.5 Fine-tuning

Table 6.11 presents the results of the fine-tuning directly on the pre-trained model. There are done five experiments with different parameters, and the table describes the different fine-tuning parameters with the evaluation metrics for the associated model. The first row presents the results where the learning rate is set to 0.0001, while the remaining rows have the same learning rate of 0.0001 with an additional dropout rate set to respectively 0.2, 0.3, 0.35 and 0.4. As mentioned in chapter 5.3, the learning rate is not stationary as it is reduced if there are no improvements while training. When fine-tuning with a learning rate of 0.0001 and no dropout rate, the WER goes down from 48.01% to 31.41%. The results get slightly better when adding a dropout rate of 0.2 and 0.4 with the same learning rate. These results show that the best WER achieved is 27.93%, and is when fine-tuning the pre-trained model with a learning rate of 0.0001 and a dropout rate of 0.35. This result shows a significant reduction in the WER of 20.08% from the pre-trained adult model.

Table 6.11: Fine-tuning results from pre-trained model

| Model | WER | CER | Loss |
|---|---|---|---|
| Learning rate 0.0001 | 0.314115 | 0.138930 | 21.012878 |
| Learning rate 0.0001 + Dropout rate 0.2 | 0.307157 | 0.130015 | 19.787546 |
| Learning rate 0.0001 + Dropout rate 0.3 | 0.292242 | 0.124629 | 19.719782 |
| Learning rate 0.0001 + Dropout rate 0.35 | 0.279324 | 0.120542 | 19.41031 |
| Learning rate 0.0001 + Dropout rate 0.4 | 0.301193 | 0.128343 | 20.240250 |

The results from the fine-tuning of the transfer learning models are presented in table 6.12. When fine-tuning the TL layer 5 and 6 model, with a dropout rate of 0.3, the WER is reduced from 38.27% to 36.18%. Fine-tuning of TL layer 6 with a dropout rate of 0.4 gives a reduction in the WER from 44.14% to 42.74%. The two other TL models show only a slight reduction of the WER after fine-tuning. Overall, the best results are from fine-tuning directly from the pre-trained model and not from the TL models, where the best results was a WER of 27.93%, compared to a WER of 36.18% from the fine-tuned TL model.

Table 6.12: Fine-tuning results from TL models

| Model | WER | CER | Loss |
|---|---|---|---|
| TL layer 6: dropout rate 0.4 | 0.427435 | 0.198737 | 28.540545 |
| TL layer 5 and 6: dropout rate 0.3 | 0.361829 | 0.151734 | 21.868723 |
| TL layer LSTM and 5: dropout 0.4 | 0.363817 | 0.154903 | 22.270435 |
| TL layer LSTM & 5 & 6: dropout rate 0.4 | 0.378728 | 0.157689 | 22.735888 |

In table 6.13, 6.14 and 6.15, we have one example each of the results for respectively the best, median and worst WER of the model that achieves the best total WER. Compared to the other results in section 6.1 and 6.3, we observe that the predictions are much more accurate, which is reasonable due to the decrease in the total WER. The example of the median WER has only one character that differs from the source. The example of the recognition with the worst WER shows a WER that is still high, where one word is predicted correctly. One word is divided into several words, which makes the WER higher. Considering these results, it is apparent that the overall WER is still quite high as some predictions make the WER increase.

Table 6.13: Example of best WER from test on fine-tuned model

| **Best WER**: |
|---|
| WER: 0.0000, CER: 0.0000 |
| src: " some workers recycle bikes" |
| res: " some workers recycle bikes" |

Table 6.14: Median WER from test on fine-tuned model

| **Median WER**: |
|---|
| WER: 0.2000, CER: 0.031250 |
| src: " dolphins live in the ocean" |
| res: " dolphins live in tho ocean" |

Table 6.15: Example of worst WER from test on fine-tuned model

| **Worst WER**: |
|---|
| WER: 1.0000, CER: 0.416667 |
| src: " butterflies are insects" |
| res: " in i fliser insects" |

# Chapter 7

# Discussion

The implementation of transfer learning and fine-tuning gave a significant improvement in the performance of the ASR system for children. The baseline model trained from scratch gave a remarkable high WER and is worse than the performance of child speech on the pre-trained adult model. This is due to training on an insufficient amount of data for a DNN model that needs a lot of data to train.

## 7.1 Performance of adult and child speech data

The results from the tests of the pre-trained adult model with the adult speech data and the child speech data showed a difference in the performance of an ASR system on adult speech and child speech. The test showed a WER of respectively 27.68% and 48.01%. This is mainly due to variances in the speech signals for adults and children. Some of this difference may also come from the fact that there are two datasets with different structures.

## 7.2 Fine-tuning

Our best result was achieved from the fine-tuned model with a learning rate of 0.0001 and a dropout rate of 0.35. This resulted in a WER of 27.93%, which is a significant decrease from the performance achieved on the pre-trained adult model, where the WER was 48.01%. The other fine-tuned models also showed a significant improvement in the WER. These all had a WER in the range of 29.22% to 31.41%, where the result with the highest WER is the result from the fine-tuned model with a learning rate of 0.0001 without any dropout. The WER of the best fine-tuning result is comparable to the results of the test with the adult TIMIT data on the pre-trained model, which had a WER of 27.68%. The WER is still relatively high to be an ideal ASR system for child speech, since such a system would be expected to recognise most of the words.

## 7.3 Transfer learning

All the transfer learning models had a decrease in the WER from the pre-trained model, but the performance was slightly less than the fine-tuned models. The WER was down from 48.01% to 36.68% for the transfer learning model that achieved the best result. This is the model where the weights of the LSTM layer and layer 5 were re-initialised. The performance of the transfer learning experiment with the two and three final layers re-initialised had a WER of respectively 38.27% and 37.97%, which is a bit higher than the result achieved from the best transfer learning model. The results of the transfer learning model with only the final layer re-initialised showed a significantly higher WER than the other transfer learning implementations, as the WER only decreased from 48.01% to 44.14%.

Fine-tuning of the different transfer learning models gave a slight improvement of the WER, where the best achieved WER is from the fine-tuned model with the two final layers re-initialised, and then fine-tuned with a learning rate of 0.0001 and a dropout rate of 0.3. The WER of this decreased from 38.27% to 36.18%, and accomplished then a lower WER than the previous best transfer learning implementation. Fine-tuning of the transfer learning model where the final layer was re-initialised reduced the WER from 48.01% to 42.74%. The two remaining models had only a slight decrease in the WER and did not improve much during fine-tuning. This show that fine-tuning was most effective for the models with the weakest performance.

It would have been expected to get better results from the transfer learning models, as some of the layers are re-initialised to better adapt to the child speech acoustics. The reason why the results achieved for the transfer learning models are not as good as the fine-tuned models, may come from the fact that the amount of data is still too low. The corpus we have used is relatively small, and in addition, a large amount of the data has been removed since they could not be used for our system without any modification of the transcriptions. The re-initialised layers were insufficiently trained before the rest of the model overfitted. Therefore the models perform better when all layers benefit from adult speech and then fine-tuned to child speech.

Even though we would have expected a lower WER from the transfer learning models and the fine-tuned models, the WER showed a significant decrease, leading to training a child speech model to benefit from adult data. The best opportunity would have been to have a lot of child speech data, but when this is limited, our results showed at least an improvement from the pre-trained adult model. It is not possible to train an accurate ASR system with the DNN model in our situation with minimal data. This was illustrated in the results of the model trained from scratch, as it had a very high WER and could not be used as an ASR system.

# Chapter 8

# Conclusion and future Work

## 8.1 Conclusions

An end-to-end (E2E) automatic speech recognition (ASR) system for child speech has been implemented using the architecture of Deep Speech by Mozilla. This model was trained and tested on the CMU Kids corpus, with four different experiments by transfer learning (TL) and five fine-tuning experiments from a pre-trained adult model. The best TL model showed a WER of 36.68%, while the best fine-tuned model had a WER of 27.93%, which is the best result achieved. These are both improvements in the WER down from 48.01%. The TL models that benefited the most are when either the two final layers (layer 5 and layer 6) are re-initialised or the long short-term memory (LSTM) layer and layer five are re-initialised. Only changing the output layer did not give much improvement. The results indicated that it was hard to get the WER to be less than 36% by transfer learning in our situation due to lack of data. Even though the results did not improve as much as desired, the results showed that adult data could be useful for training a child speech ASR system. Transfer learning and fine-tuning are effective methods to adapt the model to the child's speech acoustics when dealing with an E2E ASR system.

## 8.2 Future work

There are many ways to improve ASR systems for child speech further. An increase in the amount of data would most likely increase the performance of the ASR system. A possible way is to train datasets of different sizes and compare the performance during transfer learning. Experiments with different values of the learning rate and dropout rate from the experiments in this thesis can be tested. In addition, the beta parameters of the Adam optimiser can be adjusted. Other transfer learning experiments can be implemented, where the input layers are re-initialised instead of the output layers. These are the layers affected most by the acoustic variability of the speech signal. The architecture of the model can be designed and changed from Deep Speech's implementation, as the use of this model needs the architecture to be consistent throughout all the work. Then it is possible to change other parameters of the model, for example number of hidden nodes. A model more similar to the model in Baidu's research Deep Speech 2 can be implemented. However, this has to be implemented with the amount of child speech data in mind, as Baidu's model is implemented to train on a large amount of data.

# Bibliography

[1] K. O. Steinskog, “Child speech recognition,” Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), 2020.

[2] Q. Li and M. Russell, “Why is automatic recognition of children’s speech difficult?,” Jan. 2001, pp. 2671–2674.

[3] M. Russell, R. W. Series, J. L. Wallace, C. Brown, and A. Skilling, “The star system: An interactive pronunciation tutor for young children,” *Computer Speech  Language*, vol. 14, no. 2, pp. 161–175, 2000, ISSN: 0885-2308. DOI: https://doi.org/10.1006/csla.2000.0139. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230800901396.

[4] e. a. Garofolo John S., “Timit acoustic-phonetic continuous speech corpus ldc93s1. web download. philadelphia: Linguistic data consortium, 1993.,” 1993. DOI: https://doi.org/10.35111/17gk-bn40.

[5] M. Gerosa, D. Giuliani, and F. Brugnara, “Acoustic variability and automatic recognition of children’s speech,” *Speech Communication*, vol. 49, no. 10, pp. 847–860, 2007, Intrinsic Speech Variations, ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2007.01.002. [Online].

Available: https://www.sciencedirect.com/science/article/pii/S0167639307000052.

[6] L. Docio-Fernandez and C. Garcia-Mateo, “Speech production,” in *Encyclopedia of Biometrics*, S. Z. Li and A. Jain, Eds. Boston, MA: Springer US, 2009, pp. 1290–1295, ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_199. [Online]. Available: https://doi.org/10.1007/978-0-387-73003-5_199.

[7] J. Huber, E. Stathopoulos, G. Curione, T. Ash, and K. Johnson, “Formants of children, women, and men: The effects of vocal intensity variation,” *The Journal of the Acoustical Society of America*, vol. 106, pp. 1532–42, Oct. 1999. DOI: 10.1121/1.427150.

[8] K. Samudravijaya, *Automatic speech recognition*. [Online]. Available: http://www.iitg.ac.in/samudravijaya/tutorials/asrTutorial.pdf.

[9] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, “Chapter 2 - fundamentals of speech recognition,” in *Robust Automatic Speech Recognition*, J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, Eds., Oxford: Academic Press, 2016, pp. 9–40, ISBN: 978-0-12-802398-3. DOI: https://doi.org/10.1016/B978-0-12-802398-3.00002-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128023983000027.

[10] B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, *Towards fast and accurate streaming end-to-end asr*, 2020. arXiv: 2004.11544 [eess.AS].

[11] M. Awad and R. Khanna, “Hidden markov model,” in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA: Apress, 2015, pp. 81–104, ISBN: 978-1-4302-

5990-9. DOI: 10.1007/978-1-4302-5990-9_5. [Online]. Available: https://doi.org/10.1007/978-1-4302-5990-9_5.

[12] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Boston, MA: Springer US, 2015, pp. 827–832, ISBN: 978-1-4899-7488-4. DOI: 10.1007/978-1-4899-7488-4_196. [Online]. Available: https://doi.org/10.1007/978-1-4899-7488-4_196.

[13] D. Jurafsky and J. H. Martin, "N-gram language models," in *Speech and language processing*, 3rd ed. Prentice Hall, Pearson Education International, 2014.

[14] M. B. Sung, B. K. Choi, Y. H. Choi, and H. S. Kim, "Vtln based approaches for speech recognition with very limited training speakers," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, 2014, pp. 285–288. DOI: 10.1109/ISMS.2014.55.

[15] J. Silovsky, P. Cerva, and J. Zdansky, "Mllr transforms based speaker recognition in broadcast streams," in *Cross-Modal Analysis of Speech, Gestures, Gaze and Facial Expressions*, A. Esposito and R. Vích, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 423–431, ISBN: 978-3-642-03320-9.

[16] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, A. Ragni, V. Valtchev, P. Woodland, and C. Zhang, *The HTK Book (version 3.5a)*. Dec. 2015.

[17] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *1997*

*IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 1997, 1043–1046 vol.2. DOI: 10.1109/ICASSP.1997.596119.

[18] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, vol. 2, 1996, 1137–1140 vol.2. DOI: 10.1109/ICSLP.1996.607807.

[19] A. K. V. K Sreenivasa Rao, "Mfcc features," in *Speech Processing in Mobile Environments*. Boston, MA: Springer US, Appendix A, ISBN: 978-3-319-03116-3. DOI: 10.1007/978-3-319-03116-3. [Online]. Available: https://doi.org/10.1007/978-3-319-03116-3.

[20] Y. mohd ali, "Analysis of accent-sensitive words in multi-resolution mel-frequency cepstral coefficients for classification of accents in malaysian english - scientific figure on researchgate.
available from: Https://www.researchgate.net/figure/mel-filter-banks-basis-functions-using-20-mel-filters-in-the-filter-bank_fig1_288632263 [accessed 8 des, 2020]."

[21] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech.," *The Journal of the Acoustical Society of America*, vol. 87 4, pp. 1738–52, 1990.

[22] F. Hoenig, G. Stemmer, C. Hacker, and F. Brugnara, "Revising perceptual linear prediction (plp)," Jan. 2005, pp. 2997–3000.

[23] M. A. Nielsen, *Neural networks and deep learning*. Determination Press, 2015.

[24] S. A. Marhon, C. J. F. Cameron, and S. C. Kremer, "Recurrent neural networks," in *Handbook on Neural Information Processing*, M. Bianchini, M. Maggini, and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 29–65, ISBN: 978-3-642-36657-4. DOI: 10.1007/978-3-642-36657-4_2. [Online]. Available: https://doi.org/10.1007/978-3-642-36657-4_2.

[25] M. Venkatachalam, *Recurrent neural networks*, Jun. 2019. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce.

[26] P. Borges, *Deep learning: Recurrent neural networks*, Oct. 2018. [Online]. Available: https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010.

[27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[28] N. Donges, *What is transfer learning? exploring the popular deep learning approach*. [Online]. Available: https://builtin.com/data-science/transfer-learning.

[29] P. Marcelino, *Transfer learning from pre-trained models*, Oct. 2018. [Online]. Available: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751.

[30] P. G. Shivakumar, A. Potamianos, S. Lee, and S. S. Narayanan, "Improving speech recognition for children using acoustic adaptation and pronunciation modeling," in *WOCCI*, 2014.

[31] P. Shivakumar and P. Georgiou, *Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations*, May 2018.

[32] D. Elenius and M. Blomberg, "Adaptation and normalization experiments in speech recognition for 4 to 8 year old children," Jan. 2005, pp. 2749–2752.

[33] S. Gray, D. Willett, J. Lu, J. Pinto, P. Maergner, and N. Bodenstab, "Child automatic speech recognition for us english: Child interaction with living-room-electronic-devices," in *WOCCI*, 2014.

[34] H. Liao, G. Pundak, O. Siohan, M. Carroll, N. Coccaro, Q.-M. Jiang, T. N. Sainath, A. Senior, F. Beaufays, and M. Bacchiani, "Large vocabulary automatic speech recognition for children," in *Interspeech*, 2015.

[35] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *CoRR*, vol. abs/1912.06670, 2019. arXiv: 1912.06670. [Online]. Available: http://arxiv.org/abs/1912.06670.

[36] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *Arvix*, Dec. 2015.

[37] R. Tong, L. Wang, and B. Ma, "Transfer learning for children's speech recognition," in *2017 International Conference on Asian Language Processing (IALP)*, 2017, pp. 36–39. DOI: 10.1109/IALP.2017.8300540.

[38] Eskenazi, J. M. Maxine, and D. Graff, "The cmu kids corpus ldc97s63. web download. philadelphia: Linguistic data consortium," 1997.

[39] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. arXiv: 1412.5567. [Online]. Available: http://arxiv.org/abs/1412.5567.

[40] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[41] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, ser. WMT '11, Edinburgh, Scotland: Association for Computational Linguistics, 2011, pp. 187–197, ISBN: 9781937284121.

[42] M. Gevirtz, *What is word error rate (wer)?* Mar. 2021. [Online]. Available: https://deepgram.com/blog/what-is-word-error-rate/.

NTNU

Norwegian University of
Science and Technology