Deep Neural Networks for Factored Speech Synthesis Modelling

O. Ricardo Cortez V.

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh

2015

Abstract

This dissertation addresses the problem of acoustic modelling of multi-speaker DNN-based Text-to-speech (TTS) systems. To solve this problem we built a novel multi-speaker DNN architecture for the acoustic realisation of TTS system trained with a multiple speaker corpora of 10 English native speakers. Moreover, we have proposed an alternative approach to train this DNN architecture based on averaging the learning of each speaker during the training stage.

The design of the DNN architecture proposed is based on a multi-speaker DNN-based model designed for Mandarin native speakers, which according to our knowledge, is the first multi-speaker DNN model proposed to solve the multi-task learning into DNN-based models. Their approach proves better results than a conventional speaker-dependent DNN-based model. However, when speakers with different accents are included to the training corpus. The performance of the DNN model is affected. In comparison to this model, our multi-speaker DNN-model is capable of modelling the acoustic realisation of multiple speakers with different accents. To succeed on this achievement we included a dedicated shared hidden layer to model abstract features related to the accents of the training corpus. This variation in the DNN architecture has resulted in a better performance than a speaker-dependent model with a Scottish accent. This performance was obtained when the DNN model is trained with six English native speakers of a multi-speaker corpora. Three of these speakers have a regional accent very similar to a Scottish accent speaker and the remaining three speakers have a regional accent very similar to an English accent. A significant improvement of our DNN model is its promising capability of modelling the acoustic features of multiple speakers with different accents.

The alternative training procedure we proposed (Averaged weights) has been tested with the DNN architecture we have proposed. The approach exploits the benefits of multi-task learning. We train simultaneously all of the speakers of the model and average the weights of the multi-speaker model after all speakers have performed its corresponding backpropagation. In comparison to the training procedure proposed by Yuchen et al. (2015) our approach provide two important benefits: 1) Flexible influence of shared learning and 2) A robust acoustic modelling by intelligent shared learning. Important limitations of our approach are a slow convergence and a limited number of speakers that can influence a better generalisation of the acoustic modelling.

Acknowledgements

This work is supported by the Consejo Nacional de Ciencia y Tecnologia (CONACyT) grant number 610400.

I would like to express my gratitude to my supervisor Dr. Korin Richmond for his continuous support during the development of the project. In addition, I would like to thank Dr. Zhizheng Wu for his kind support during the development of the experiments of this dissertation.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*O. Ricardo Cortez V.*)

Table of Contents

Chapter 1

Introduction

A Text-to-speech (TTS) system can be seen as a set of modules addressing the task of converting written text into speech. The common form of these systems includes a text-analysis system and a speech synthesis system (Taylor, 2009). The text-analysis system focuses on extracting the message from the written text. The speech synthesis system is in charge of encoding this message and produce a synthetic speech output.

Common implementations of TTS systems varies from medical applications, supporting people with disabilities such as the well-known case of Professor Stephen Hawking to commercial mainstream use like Apple's Siri. These range of applications have provided useful systems but that denote a limited quality in terms of human speech similarity and fluency. Thus, research community especialised in the speech synthesis field has made several efforts to build robust modules that lead to the improvement of speech quality.

Significant attention has been put to the speech synthesis system, as some of the fundamental approaches used in this module, as the case of parametric models, have not been capable of completely solve the main causes that affect the quality of synthesised speech.

Thus, the major goal of TTS systems is to produce synthetic and human sounding speech. The performance of these systems is evaluated according to intelligibility and naturalness of the speech. According to Taylor (2009), intelligibility is defined as the ability of the listener to understand the message generated by the TTS system. Naturalness refers to the fluency of the synthetic speech.

Conventional TTS systems aimed at creating human-like speech take advantage of parametric approaches to extract speech features from speech of a training corpus. One of the most important parametric approaches are Hidden Markov Models (HMMs), which focus on representing a speech model that also generates a speech output. These models extract several representations of speech features from a training corpus. A training corpus consists of a number of recorded speech samples produced by a given speaker. Based on the number of speakers TTS systems based on parametric speech synthesis can be categorised as speaker-dependent or multi-speaker systems. In general, the performance of multi-speaker systems is better than speaker-dependent systems. One of the main reasons of this behaviour is because multi-speaker systems are trained by using recording samples from diverse speakers. This brings the model significant understanding of the speech properties of the corpus. Based on this understanding, the model can extract representative parameters that help the model to produce a synthetic speech output which includes characteristics contained when humans produce speech. By incorporating important characteristics contained in human speech a parametric model can generate a synthesised speech of good quality.

One of the main disadvantages of HMMs models is that the quality of the speech is affected by factors such as the accuracy of acoustic models. This accuracy is influenced by auxiliary models such as decision tree-clustered context dependent model that support HMMs to encode the message of the input.

1.1 Motivation

An alternative approach recently proposed by Zen et al. (2013) is aimed at overcoming the main limitations of acoustic modelling of HMM-based models. This approach consists on using Deep Neural Networks (DNNs) for performing acoustic modelling of better quality. As in the case of HMMs, DNNs are able to build either speaker-dependent or multi-speaker models. However, most of the late research has centered on devising speaker-dependent DNN-based models. DNNs take advantage of using much larger data sets to represent general and more abstract representations of informative speech features. Thus, DNNs are lately a robust parametric approach that is improving the acoustic realisation for synthesising speech of higher quality and more similarity to human speech.

A natural motivation to build TTS systems of higher quality is to exploit the benefits of DNN-based models with the purpose of devising multi-speaker DNN models. The reason of this interest is based on the benefits that multi-speaker HMM-based models has in comparison to speaker-dependent HMM-based models. Therefore, we hypothesise that, as in the case of HMMs, we can exploit the benefits of having a phonetically and linguistically richer corpus and combining it with the DNN modelling benefits to extract better and more abstract representations of larger data sets. In consequence, developing a robust multi-speaker DNN model can lead to a better acoustic representation of the training corpus and produce speech of better quality and human sounding than the traditional speaker-dependent DNN-based models.

To the best of our knowledge, the unique research reference in multi-speaker DNN-based models is the one proposed by Yuchen et al. (2015). In this research work, they have proposed a novel multi-speaker DNN-based model for building TTS systems using a corpus of Mandarin native speakers. The configuration of the DNN architecture is divided in two parts, the first part of the DNN model focuses on modelling the linguistic and phonetic information contained in the input data; the second part of the model is in charge of improving the accuracy of the acoustic models. Moreover they use multi-task learning and speaker adaptation techniques to enhance the robustness of the DNN architecture. As a result, they have significantly improving the performance obtained in speaker-dependent DNN systems when training the multi-speaker DNN model with diverse Mandarin speakers that have the same regional accent. Contrarily, the performance of the multi-speaker DNN model proves a worse performance than speaker-dependent DNN-based models when including to the training corpora the corpus of a speaker with a different accent.

1.2 Contribution

Based on the previous work of multi-speaker DNN-based models (Yuchen et al., 2015), we propose to build a novel multi-speaker DNN architecture for the acoustic modelling of a multi-speaker corpus of English native speakers. To devise our multi-speaker DNN model we consider similar aspects of the architecture evaluated by Yuchen et al. (2015). For instance, we also divide our architecture in two parts and we maintain an output layer for each speaker of the training corpus. Nevertheless, an important variation of our architecture is the inclusion of a dedicated hidden layer for modelling features related to the accent of each speaker. The purpose of this variation is to overcome the issues found in the Mandaring multi-speaker model when including speakers with different regional accent. In this manner, our particular objective of our project is to evaluate the performance of our DNN model when using a training corpus of native English speakers with English accent and Scottish accent.

Finally, the structure of this dissertation includes background information, which consists of a description of the important topics of deep learning and speech synthesis for the understanding and analysis of the research developed throughout this dissertation. This description is included in chapter 2 . Chapter 3 discusses important aspects of the methodology implemented. We describe the hypotheses and considerations we take into account to devising our multi-speaker DNN-based model as well as we present the characteristics of auxiliary tools that we used for the implementation of our DNN architecture. Chapter 4 describes the set of experiments we designed and tested. Moreover, we present an extensive analysis of the results obtained indicating the main benefits and limitations of our approach. At the end of that chapter, we compare our DNN model with a conventional speaker-dependent DNN-based model. Finally, chapter 5 discusses the conclusions obtained based on the experiments performed as well as we suggest diverse research lines as future work to improve the multi-speaker DNN model or to devise more robust multi-speaker DNN-based models.

Chapter 2

Background

The content of this chapter is aimed at discussing the fundamental concepts involved in devising and implementing multi-speaker speech synthesis by exploiting the benefits of deep learning approaches. Deep learning and speech synthesis are the two main topics that form this chapter as their understanding is required for the analysis and improvement of traditional approaches used in Deep Neural Networks (DNN)-based models for speech synthesis systems. We focus on the advantages and disadvantages of each topic that were considered to propose and develop the methodology, experiments and analysis of the present dissertation.

2.1 Deep Learning

Deep learning is one of the most recent approaches of machine learning that has made substantial improvements in different research areas of Artificial Intelligence such as artificial vision and speech processing. For instance, the use of deep learning approaches like convolutional networks has improved performance and accuracy when recognising pictures from large-scale images (Simonyan and Zisserman, 2014) (Krizhevsky et al., 2012), as well as when implemented to perform speech detection tasks (Sukittanon et al., 2004). Similarly, Zen et al. (2013) have proposed the use of DNNs for building speech synthesis systems as they have shown to better model the mapping between the input text and the acoustic realisations, which helps the speech synthesis model produce artificial speech with more similarity to human speech in comparison to traditional techniques such as models based on Hidden Markov Models.

The purpose of deep learning methods is centered around modeling high-level patterns that can describe difficult characteristics to perform a target task. In order to extract these features deep learning methods consider a large set of non-linear transformations (Bengio, 2009). The usage of large amounts of data is fundamental to model complex descriptions in a compact manner. In addition to the lower level features that are obtained by mapping the input data to the output of the model, modeling patterns of higher complexity leads to build a more robust parametric model, which ideally gives better performance of the target task(Bengio, 2009).

A deep learning architecture is formed by hierarchical multiple-layered models. They include several hidden layers in which descriptive information is extracted and gradually more abstract representations are created (Bengio, 2009). The depth of the architecture is related to the number of layers that comprise the function learned (Bengio, 2009). We start by defining a high dimensional input which can contain numerical and discrete features representing factors of variation associated by -but not limited to-statistical relationships. Subsequently, intermediate-level and deep-level abstractions are identified and combined to extract a reasonable understanding of the desired task to be performed (Bengio, 2009).

2.1.1 Advantages and Limitations

There are different advantages that have dramatically increased the popularity of deep learning algorithms among the research community of several areas in Artificial Intelligence. For instance, they can increase the performance of systems on complex problems by effectively exploiting either labeled and unlabeled data (Deng and Yu, 2014). This advantage permits an implementation for supervised and unsupervised learning tasks. In the case of supervised learning tasks, unlike traditional methods of machine learning (e.g. Gaussian Mixture Models, Naive Bayes, etc.) whose performance entails feature engineering; deep learning promotes an end-to-end learning based on raw features (Deng and Yu, 2014) (Kotsiantis et al., 2007). For unsupervised learning tasks, we can take advantage of large amounts of existing unlabeled data to extract useful information and train a specific model in an unsupervised way (Bengio et al., 2013). This ability allows in some manner a lower human effort to correctly set the input data of model for a required artificial intelligence task.

Most of the time, using online learning approaches for training deep learning architectures entails diverse difficulties. For instance, optimising models based on DNNs requires often extensive computational time to optimise the function of the model, by which it is usually necessary to use GPU computations (Murphy, 2012). Similarly, setting the optimum training parameters (e.g learning rate) is sometimes difficult since a large or small learning step might not find the optimal function to represent the target task. Additionally, it is necessary to determine a correct criterion when training the model like setting the best error training value such that we avoid the possibility of overfitting the model (Murphy, 2012).

2.1.2 Deep Learning Architectures

Diverse research references differ when classifying deep learning algorithms; some relevant classifications can be found in the research papers developed by Bengio (2009) Murphy (2012) and Deng and Yu (2014). However, the most popular categorisation of these techniques is based on adopting the traditional machine learning approach. This approach consists on predicting an output function y given the value of an input variable or pattern x. Therefore, based on the classification suggested by Bengio (2009) and Murphy (2012), these algorithms can be classified by the probability distribution they try to represent. These two classes are the following:

1. Generative/unsupervised Models : They require a decision function derived from the generative model (Deng and Li, 2013). Then, they can generate meaningful information with the understanding learned from the networks. The most common approaches of this class are energy-based deep models; each configuration of the parameters of interest is associated to a scalar energy (Deng and Yu, 2014). Compared to discriminative models they easily handle uncertainty and can be interpreted in a simple manner. However, when they try to learn and infer complex systems, they are generally intractable (Deng and Yu, 2014). Part of the most representative generative models are :

    - Restricted Boltzmann Machine (RBMs)
    - Deep Belief Networks (DBNs)
    - Deep Boltzmann Machines (DBMs)

2. Deep Discriminative Models : According to (Murphy, 2012), their main goals

is to estimate a discriminative model of the form $p(y|x)$. Therefore they have proven high performance when used in pattern recognition tasks such as in speech recognition systems. Moreover, Deng and Yu (2014) suggest that deep discriminative models are more suitable for end-to-end learning of complex systems.

Deep Neural Networks (DNNs) are the most representative model of this class. DNNs can better represent abstract features of a complex task when the configuration of the architecture is set with the optimal number of hidden layers. The analysis made by (Deng and Jaitly, 2015) (see Table 2.1) demonstrates that performing tasks such as speech employing DNN-based models have diverse advantages in comparison to deep generative models. For instance, implementing simpler learning algorithms such as backpropagation permits an easier and more flexible implementation (Deng and Yu, 2014). In addition, we can exploit the use of massive real data and computing the best parameters of the DNN architecture by exploiting the benefits of GPU computations.

Part of these advantages have encouraged the construction of statistical parametric synthesis models based on DNNsZen et al. (2013), as well as further research in more complex tasks such as multi-speaker DNN-based text-to-speech models (Yuchen et al., 2015). For these reasons we discuss further details about DNNs, as they are the main basis of the work developed for this project.

Table 2.1: High-level comparisons between deep neural networks and deep generative models (Deng and Jaitly, 2015).

	Deep Neural Networks	Deep Generative Models
Domain Knowledge	Hard	Easy
Semi/ Unsupervised	Harder	Easier
Representation	Distributed	Local or Distributed
Inference/ Decode	Easy	Harder
Scalability/ Compute	Easier (Regular Computes / GPU)	Harder
Incorp. Uncertainty	Hard	Easy
Empirical Goal	Classification, Feature learning,etc.	Classification (via Bayes rule) Latent Variable Inference, etc.
Terminology	Neurons, Activations/gate functions, weights, etc.	Random variables, potential function, parameters,etc.
Learning Algorithm	Almost a single, unchallenged algorithm —Backpropagation	A major focus of open research, many algorithms
Evaluation	On a black box score - end performance	On almost every intermediate quantity
Implementation	Hard, but increasingly easier	Standardized methodsexist, but some tricks and insights needed
Experiments	Massive, real data	Modest, often simulated data
Parameterization	Dense matrices	Sparse (often); Conditional Probability Density Functions

2.1.3 Deep Neural Networks

A DNN can be seen as an artificial neural network that contains diverse hidden layers fully connected between the input and output layers (Deng and Yu, 2014). DNNs are in charge of modelling abstract non-linear relationships using non-linear functions such as a “tanh” or a logistic function. The depth of the architecture depends on the number of hidden layers k that form the model. Each hidden layer k computes an output vector $\mathbf{h}^k$ using the output $\mathbf{h}^{k-1}$ of the previous layer. $\mathbf{x} = h^0$ is the input of the architecture. We can define $\mathbf{h}^k$ as follows

$$\mathbf{h}^k = tanh(\mathbf{b}^k + \mathbf{W}^k \mathbf{h}^{k-1}) \tag{2.1}$$

where $\mathbf{b}^k$ is a vector of offsets and $\mathbf{W}^k$ is a matrix of weights. The function tanh is applied element-wise and can be replaced by other saturating non-linear functions such as a sigmoid function. The top layer $\mathbf{h}^l$ is combined with a supervised target output y into a loss function $L(\mathbf{h}^l, y)$ typically convex in

The output layer might use a non-linear function different from the one used in the hidden layers such as a linear regression function

$$\mathbf{h}_i^l = \frac{e^{\mathbf{b}_i^l + \mathbf{W}_i^l \mathbf{h}^{l-1}}}{\sum_j e^{\mathbf{b}_j^l + \mathbf{W}_j^l \mathbf{h}^{l-1}}} \tag{2.2}$$

where W_i^l is the ith row of W^l, $\mathbf{h}_i^l$ is positive and $\sum_i \mathbf{h}_i^l = 1$.

In this case it is often used the negative conditional log-likelihood $L(\mathbf{h}^l, y) = -logP(Y = y|x) = -log(\mathbf{h}_y^l)$ as a loss function, whose expected value over (x,y) pairs is to be minimised.

2.1.3.1 Training Methodology

Gradient-based training is one of the most popular approaches used for optimising DNNs models. The learning procedure implies defining a supervised cost function on the output layer considering a desired target. We need to use a gradient-based optimisation algorithm to adjust the weights and biases of the network so that we can determine a minimum cost on samples of the training set. Commonly, we start with a

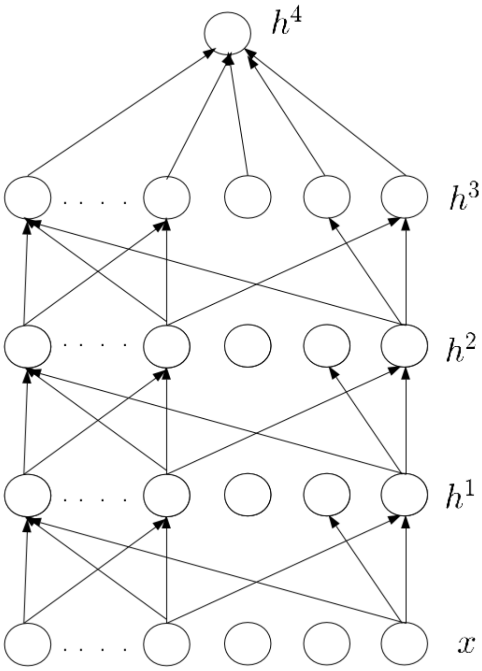

Figure 2.1: Typical Multilayer Neural Network. Each layer combines an affine operation and a non-linear function(Bengio, 2009).

random initialisation of the weights of the architecture and use backpropagation when applying Gradient Descent (Larochelle et al., 2009).

The most representative challenges when training a DNN architecture are: 1) Avoiding getting stuck in local minima and 2) Obtain a good generalisation as the architecture gets deeper (Erhan et al., 2009).

2.1.4 Backpropagation

Backpropagation is a recursive algorithm that uses the chain rule and stores some of the intermediate terms to find the best weights for each unit h^i to minimise the cost function $E(\mathbf{w})$. The manner DNNs are trained consists of backpropagating derivatives of a cost function, which measures the difference produced between the input $\mathbf{x}$ and the desired output y (Larochelle et al., 2009). Then, our target is to find the best weights for each unit h^i such that the cost function $E(\mathbf{w})$ is minimised.

Considering the softmax as the output function, we can define the cross-entropy error function $E(\mathbf{w})$ as

$$E(\mathbf{w}) = \sum_{i=1}^{N} log(1 + exp(-y^n \mathbf{w}^T x^n) \tag{2.3}$$

where N is the number

We can obtain the Gradient of $E(\mathbf{w})$ by deriving the cost function $E(\mathbf{w})$ with respect to the every weight ω_k from w, which is defined as follows:

$$g_k = \frac{\partial}{\partial(\omega_k)} E(\mathbf{w})) \tag{2.4}$$

According to Murphy (2012), the Gradient Descent can be written as follows:

$$\omega_{k+1} = \omega_k - \eta_k g_k \tag{2.5}$$

where η_k is the step size or learning rate.

Finally, considering equations 2.3, 2.4 and 2.5 we can use the pseudoalgorithm depicted in Figure 2.2, which correspond to the Stochastic Gradient Descent algorithm, to determine the optimal parameters (i.e. the best weights) that minimise the cost function $E(\mathbf{w})$

```
Initialize $\mathbf{W}$, $\eta$;
repeat
    Randomly permute data;
    for $i = 1 : N$ do
        $g = \nabla f(\mathbf{W}, \omega_i)$;
        $\omega_i \leftarrow proj_{\omega}(\omega - \eta g)$
        Update $\eta$

until converged;
```

Figure 2.2: Pseudoalgorithm of Stochastic Gradient Descent (Murphy, 2012)

2.2 Speech Synthesis

Text-to-speech (TTS) systems are focused on converting normalised text from a given language into synthetic speech produced by a computer. One of the major goals in speech synthesis research is to create robust models that lead to creating synthesised speech with similar sound to those produced by humans. Therefore, important factors to be considered in a speech synthesis systems are intelligibility, naturalness and speech quality (Taylor, 2009).

2.2.1 Statistical Parametric Speech Synthesis

Statistical parametric speech synthesis is one of the most popular approaches used in speech synthesis research today when building robust TTS systems. Statistical parametric models are robust models capable of extract, and learn, fundamental parameters of the training corpus (i.e. from normalised real voice recordings), which lead to a similar representation of human speech (Zen H. and Black, 2009). They bring powerful properties to map the original input voice into a voice with different properties and even to customise the emotion of the speech (Taylor, 2009). Finding the best parameterisation of the model is critical for the performance of these sort of TTS systems.

Some of the most popular statistical parametric approaches include:

- Hidden Markov Models (HMMs)
- Gaussian Mixture Models(GMMs)

2.2.2 Hidden Markov Models (HMMs) (General Overview)

Despite HMMs are a traditional approach to implement TTS systems, a detailed explanation of this technique is beyond the purpose of this work. Therefore, we only present a brief explanation of the important aspects that are considered in this method.

HMMs generate and model the most representative speech features of a training corpus. In order to create the synthetic speech the model uses sequences of frames known as observations (see Figure 2.3 for a further explanation):

$$O = (o_1, o_2, ..., o_T) \tag{2.6}$$

Each of the observations extracted from the corpus is usually represented by parameters such as mel-cepstral coefficientes (MFCCs) coefficients, Fundamental frequency

F_0, delta parameters, etc. These parameters are used to define an incorporate characteristics to the artificial speech so that the HMM model can produce a synthesised speech output of high quality.

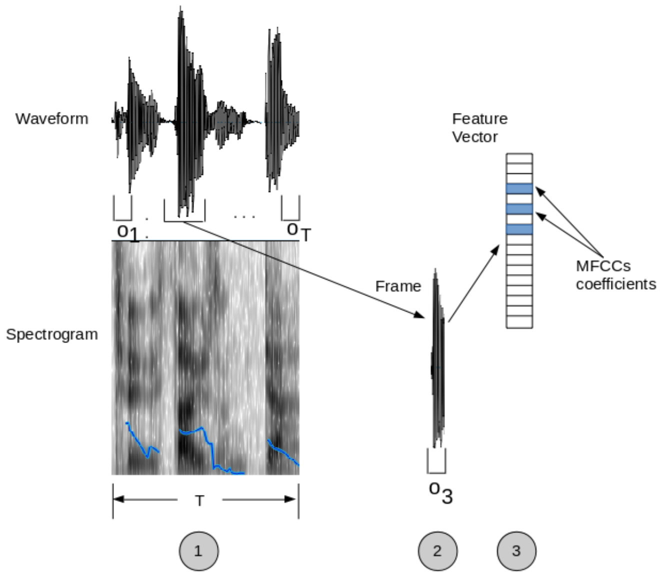

Figure 2.3: Common representation of a waveform considering its spectrogram, a extracted frame O_3 and its corresponding feature vector with MFCCs coefficients.

To have a general idea of what a frame is and how is this information analysed by parametric models such as HMM models this figure present a common representation of an observation O. In step 1 (see gray circle 1), the waveform presented contains O_T observations of a waveform of time T. Parallel to the waveform, it is depicted the spectrogram of the waveform shown, which illustrates natural variations of the vocal tract of a human when producing a given waveform. From the waveform of step 1, an observation O_3 is extracted. In step 3, it is represented the feature vector that describes information contained in step 2, this vector contains -among others- MFCCs coefficients.

One of the major disadvantage of HMM models is the quality of the artificial speech (Zen et al., 2013), which is commonly affected by three factors:

1. Vocoding
2. Accuracy of Acoustic Models
3. Over-smoothing

In HMM models, the accuracy of acoustic models is primarily affected by phonetic, linguistic and grammatical factors, which are modeled by the decision tree-clustered context dependent model. An alternative and powerful approach that has focused on enhancing the accuracy of acoustic modeling is to use DNN-based models, which (as detailed in section 2.2.3 are focused on devising robust architectures that model complex context dependencies with more accuracy.

2.2.3 Statistical Parametric Speech Synthesis using DNNs

The difference between conventional statistical parametric models and DNN-based parametric models resides on substituting the decision trees for a deep neural network (Zen et al., 2013). HMMs use decision tree models to represent linguistic factors of the input text. In order to determine a good linguistic representation they need to use local sets of parameters such as surrounding phonemes. By considering local sets of parameters, HMMs discard abstract but useful features of the speech, affecting the quality of the speech created as the model is not capable of provide a good representation of a natural and human sounding speech output. DNNs models overcome this limitation by maximising the influence of parameters contained in the training corpus; building a more representative model that synthesise speech with more similarity to human speech.

In order to synthesise a desired text, we consider the speech synthesis framework proposed by Zen et al. (2013) (see Figure 2.4). We map the raw text into a normalised input of features $\{x_n^t\}$: x_n^t denotes the n-th input feature at frame t. They include both binary and numeric values representing properties of the data such as the position of a phone, syllable and word in a phrase and sentence, identity, Part-of-speech of word, etc. The output features contain spectral parameters and dynamic features. Pairs of input and output features are used when implementing backpropagation during the training stage. Finally, the subsequent modules to the DNN architecture are in charge of building a synthesised waveform considering the speech parameters generated by a speech parameter generation algorithm (Zen et al., 2013).

2.2.4 Speaker Dependent TTS systems

According to the configuration depicted in Figure 2.4 we can use the speaker dependent approach to build a speaker dependent TTS system (Lee, 1990). This approach use speech recordings of one particular person to train the TTS system. Taking advantage

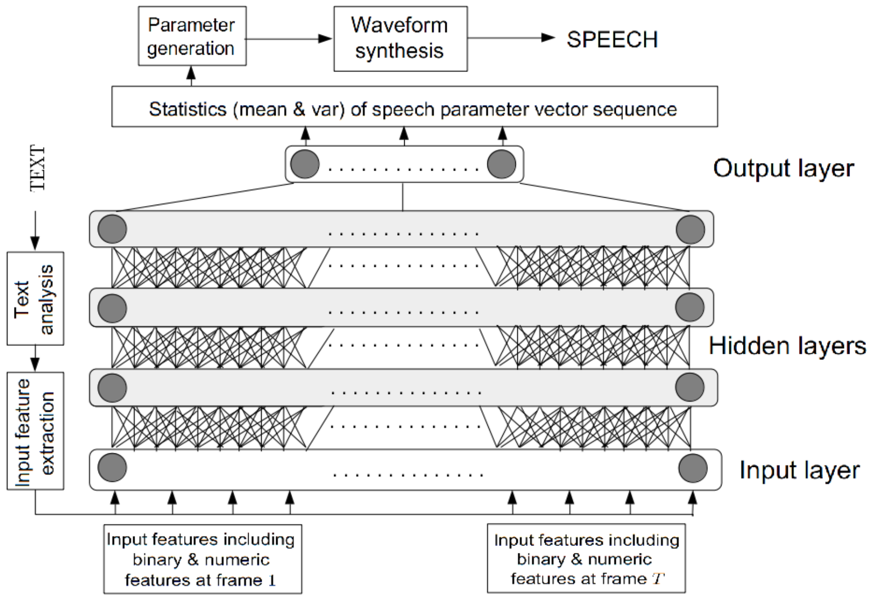
Parameter generation
Waveform synthesis
SPEECH
Statistics (mean & var) of speech parameter vector sequence
Output layer
TEXT
Text analysis
Hidden layers
Input feature extraction
Input layer
Input features including binary & numeric features at frame 1
Input features including binary & numeric features at frame T

Figure 2.4: Speech synthesis framework based on a DNN referenced in (Zen et al., 2013).

of the speech properties of a given speaker the model can represent natural speech variations that enhance the naturalness and quality of the artificial speech. However, in order to train a high-quality model, it is required to use rich speech data in terms of prosody and phonetic information.

Figure 2.5 illustrates the architecture of a speaker dependent model based on DNNs. The model requires an input feature vector including binary and numeric linguistic features, which are modeled by the hidden layers and an output layer that maps the linguistic parameters of the DNN architecture into acoustic features such as Fundamental frequency (F_0) and unvoiced/voiced flags.

2.2.5 Multi speaker TTS systems

Multi-speaker modeling extracts the main features from the speech of diverse speakers. Data from different speakers can lead to extracting richer features that determine the optimal acoustic space of the TTS model (Wan et al., 2012).

In this manner, an important benefit of multi-speaker models is their capability to adjust speech properties such as the sound of the voice and speaking style of the synthetic speech. Moreover, in comparison to the speaker dependent approach it is possible to produce artificial speech of better quality and more similar to the speech of humans

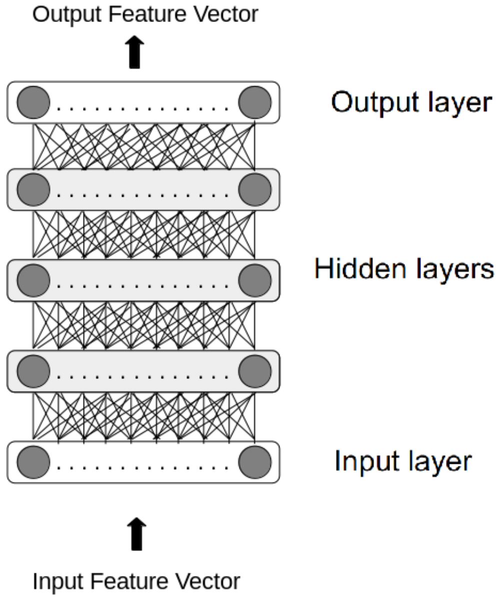

Figure 2.5: Speaker Dependent DNN architecture.

(Wan et al., 2012).

Figure 2.6 illustrates the multi-speaker DNN architecture proposed by Yuchen et al. (2015). The DNN architecture contains diverse hidden layers shared between the different speakers to model their linguistic parameters whilst the output layer for each speaker is independent from the rest of the speakers. Its major purpose is to map the linguistic features to the optimal acoustic model of the target speaker (Yuchen et al., 2015).

Important considerations of this approach suggest that the model should be primarily based on a set of combined speaker dependent models. For this reason it is necessary to train the DNN architecture for all the speakers simultaneously. One important drawback found by Yuchen et al. (2015) indicates that all the speakers must have similar characteristics (e.g. same accent) in order to avoid considerably affecting the performance of the multi-speaker system. Further research is required to lead to the solution of this problem.

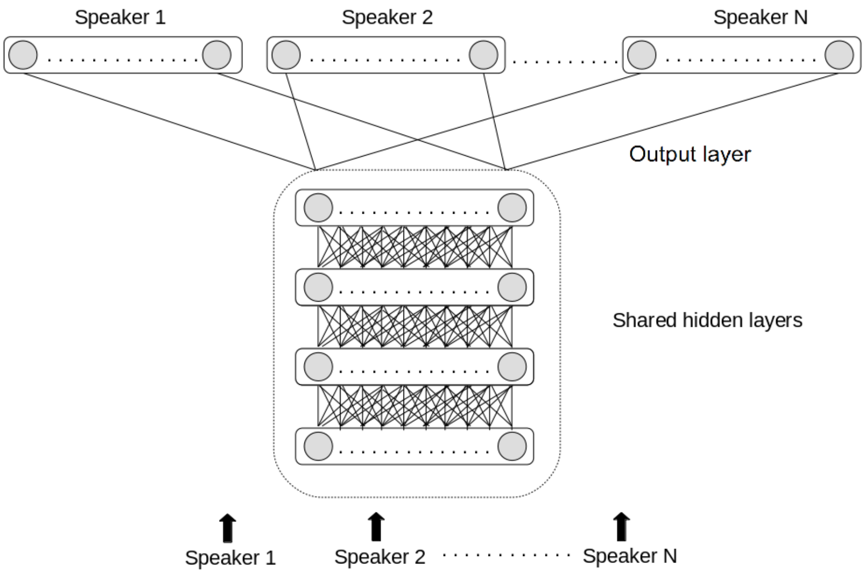

Figure 2.6: Multi-speaker DNN architecture evaluated by Yuchen et al. (2015). The architecture is formed by diverse hidden layers shared between the different speakers. Each speaker has its own output layer to model particular speech variations of the speaker.

Chapter 3

Methodology

This chapter describes the methodology proposed to build and test a robust multi-speaker DNN-based model. By devising this DNN architecture we are intended to improve the accuracy realisation of multi-speaker DNN models already implemented for Mandaring language. We detail diverse sources and tools that are auxiliary for the optimal implementation of our approach. For instance, the training corpus used and the deep learning framework involved when training the DNN architecture. Most importantly, we portray the approach proposed as an alternative architecture for building more robust multi-speaker DNN architectures. At the end of the chapter the experiments proposed for the support of the devised approach are explained and justified.

Based on the speech synthesis framework shown in Figure 2.4, the scope of our approach is entirely focused on devising an architecture based on DNN models using multi-speaker speech recordings. Therefore, the input of our model requires labeled data rather than purely raw text. The output of our model needs to be a n-dimensional feature vector rather than the common output of a TTS system: synthesised speech.

Considering the purpose of our approach, we are required to build a training corpus of diverse speakers categorised into two different accents. Our goal is to form a corpus of 10 native English speakers from Britain, which contains 5 speakers of diverse regional English accent as well as 5 speakers of diverse regional Scottish accent. The purpose of having two classes of accents will help us to evaluate whether our approach is able to model the main limitation found in the multi-speaker DNN-based TTS system proposed by (Yuchen et al., 2015), which is modeling multi-speaker data from speakers with diverse regional accents.

3.1 Training Corpus: The Voice Bank Corpus

The dataset selected is the Voice Bank Corpus which is constantly extended by the University of Edinburgh Veaux et al. (2013). This dataset is the largest corpus of British English recordings; it is formed by speech from more than 500 English native speakers covering diverse human speech variations such as regional accents, genre and social classes. The collection of each speaker includes roughly 60 minutes of recorded speech. Each collection contains 425 sentences: 400 sentences from the Scottish Herald newspaper and 25 sentences from both the Rainbow passage and the Accent Elicitation passage from the Speech Accent Archive. The set of sentences selected is intended to maximise the coverage of phonetic and prosodic information for the benefit of building a robust TTS system.

3.1.1 Recording Conditions

According to Veaux et al. (2013), most of the speech corpus was recorded in a semi-anechoic chamber at the University of Edinburgh using the same recording equipment for each speaker: a Sennheiser MKH 800 (omnidirectional diaphragm condenser with very wide bandwidth) and a DPA 4035 (headset-mounted condenser). The dataset was originally made at 96KHz sampling frequency and 24 bits per sample but later it was downsampled to 48KHz. The purpose of over-sampling each sample (i.e. mapping from 96KHz to 48 KHz) is to reduce noise during recordings that could affect the accuracy of the parameterisation of each speech sample. In addition, some recordings were made in the Scottish parliament. The purpose of them was to have a wider coverage of diverse regional accents from the Scottish accent. The recording conditions and equipment differs from the rest of the recordings since it was necessary to use an AKG-C414 microphone and a Focusrite Forte USB audio interface connected to a laptop.

3.1.2 Corpus Analysis

Two important difficulties were found when categorising the Voice Bank Corpus. First, information about the characteristics of each speaker was complicated to obtain by which it was necessary to determine the genre and accent of diverse speakers under our own analysis and criterion. In addition, some of the recordings of the diverse speakers contained very few samples to be trained by the DNN model. These speakers were automatically omitted from further analysis. On the other hand, the task of defining which accent corresponds to each speaker was more difficult than expected considering that I am non-native English speaker. Therefore, it might be possible that the accent we assign to some of the speakers might vary according to the judgment of other people. Finally, another aspects such as the social class or the age of the speakers were impossible to determine since we did not find available information that could help us to define such factors.

According to our evaluation, Table 3.1 shows the results generated from the 10 more relevant speakers after analysing the speech recordings from the corpus. These results include the ID of the speaker according to the Voice Bank Corpus, the gender of the speaker as well as the speakers we considered with the accent more similar to that speaker. Note that we only included the similar accent speakers relevant for the discussion of the present work. We only focus on the similarity of the accent however all the speakers can be related for different reasons than the condition we are interested in.

Table 3.1: Speech properties of 10 speakers from the Voice Bank Corpus. They include ID from the Voice Bank Corpus and its corresponding Gender and Accent according to our analysis.

SPEAKER	ID	GENDER	ACCENT	Similar (Relevant) Accent Speakers
1	p389	Female	English	p291, p337, p273, p226
2	p337	Female	English	p291, p389, p273, p226
3	p291	Female	English	p337, p389, p273, p226
4	p273	Male	English	p291, p337, p389, p226
5	p226	Male	English	p291, p337, p389, p273
6	p395	Male	Scottish	p46, p397, p327, p114
7	p397	Male	Scottish	p46, p395, p327, p114
8	p46	Male	Scottish	p397, p395, p327, p114
9	p327	Female	Scottish	p46, p397, p395, p114
10	p114	Female	Scottish	p46, p397, p395, p327

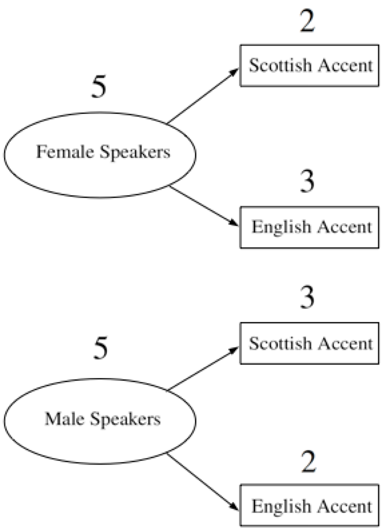

Figure 3.1: Classification of the dataset implemented. Speech from 5 Female speakers contains 2 Scottish Accent and 3 English accent data. Speech from 5 Male speakers includes 3 Scottish Accent and 2 English accent data.

3.1.3 Description of selected dataset

From the Voice Bank Corpus we have selected 10 representative speakers whose speech properties meet enough variations and similarities among the speakers to be learned by the DNN model. As illustrated in Figure 3.1 the dataset is divided in two classes: 1) Male speakers and 2) Female speakers. A very important aspect that was considered is the accent of the speakers. Then, from each class we have 2 speakers with accent *a* and 3 speakers with accent *b* so that we can analyse in a simple manner the influence of these aspects in both the training and performance of the DNN architecture.

3.1.3.1 Selection of Multi speakers data

In order to select the best speaker candidates for training the DNN model, it was necessary to analyse the Voice Bank Corpus and consider factors such as genre, accent, number of recordings, quality of recordings, etc. for the evaluation of a good speaker corpus. The 10 speakers selected meet the following criteria:

1. The speaker's collection has more than 300 recorded sentences. The purpose of a large number of labeled data will ideally meet the requirements of DNN architectures for obtaining a good performance because of understanding large amounts of data.

2. The recordings have very few noise from the environment. The intention of this condition is that the recordings have optimal conditions to construct accurate features from the recordings, which subsequently can lead to the ideal modeling of the training corpus.

3. The accent of the recordings was identified. Having certainty of the accent of the speaker allows to categorise each speaker when training the DNN model.

4. The recordings have labeled acoustic data available. Since the time for developing this project is limited, we are required to have labeled acoustic data available in order to focus on devising the DNN architecture rather than on implementing an approach that help us to construct such labeled data..

5. The recordings have some similarity in accent style to the recordings of at least other four speakers. Since our judgment when deciding the accent of each speaker is difficult, we also use this consideration to determine that the accent of the speaker belongs to only one of the two possible classes.

3.1.4 Labeled Data

After selecting the 10 speakers of our corpus, it was necessary to convert the waveform of each recording into an input feature vector x suitable for training the DNN architecture. Similarly, we needed to build an output feature vector in order to create pairs of input-output vectors used for training the DNN network with the backpropagation algorithm. These mappings were taken from the work developed by Wu (2015) in order to optimise the developing time of the project and focus primarily on devising and implementing the DNN architecture.

3.1.4.1 Input Feature Vector

The input feature vector is a 601 dimensional vector representing 5ms of the speech recording. This vector is formed by 592 binary linguistic features, which comprised quinphone (a set of 5 phones), part-of-speech (A tag that includes information about jts neighbours), phoneme (The basic unit of a language), syllable, word and phrase positions. The remaining 9 features include numerical features adding information of the frame position and phoneme in the architecture, state position in phoneme and state and phoneme duration. The numerical features were normalised to the range of 0.01 to 0.99 (Wu, 2015).

3.1.4.2 Output Feature Vector

The output feature vector is a 259 dimensional vector representing 5ms of speech data. This vector contains 60 features describing Mel Frequency Cepstral Coefficients (MFCCs), 25 features representing bandpass aperiodicity (BAPs) and one feature for the Fundamental frequency F_0. Moreover, this vector includes their corresponding delta and delta-delta features and a voiced/unvoiced (V/U) binary value. The output features were normalised by speaker-dependent mean and variance (Wu, 2015).

3.2 Software for training DNN architecture

In order to implement the DNN architecture we will use Python v.2.6 as the basis software for our project. There are four important reasons that support our decision:

- Python is the leading language of research work developed in the Speech Processing Field. Therefore, consulting available code on the web can help us to adapt it to our particular task easily.
- A lot of documentation of Python libraries and help forums are available on the web. It is expected that many problems arise when implementing diverse code using libraries such as Pylearn, Numpy, etc. For that reason, our expertise can be overwhelmed and further coding skills and software understanding can be required. Therefore, useful documentation is fundamental for helping to solve effectively unpredictable problems which can risk succeeding on the implementation of the DNN architecture.
- Python is the programming language we are more experienced with. Having enough familiarity and expertise with Python can lead to a faster and more efficient coding performance. Therefore, taking advantage of this knowledge can help us to inquire on devising a functional model and to deeply analyse and interprete the performance of the approach proposed.
- Python is a programming language already installed and available for its use

in the university facilities. Installing specific software might be a difficult task that sometimes require expertise and meet some hardware requirements. Factors such as the operating system and hardware can be decisive for correctly installing a desired program such as Python. Therefore, by having Python already installed and ready to use we can avoid this sort of problematic and reduce the chances to spend time in secondary tasks to be solved.

3.2.1 Deep Learning Framework

A deep learning framework is an auxiliary tool used for training diverse deep learning approaches. Some of the leading frameworks are Caffe (Jia et al., 2014), Theano (Bastien et al., 2012a) and Torch7 (Collobert et al., 2011). The elementary purpose of these frameworks is to bring a collection of reference models and deep learning algorithms easy to implement and modified for understanding and learning a desired task. These frameworks consist of libraries ready to install in the most common programming languages such as Python, C++ and MATLAB. Most of them have been implemented in diverse applications such as Object Classification or Learning Semantic Features.

Diverse aspects need to be considered when selecting the optimal framework. For instance, separation of representation and implementation is an essential factor exploited to prioritise computation memory for the network and easily switch the run mode between a CPU and a GPU implementation. Modularity deals with the manner in which we can extend new network layers and loss functions to the DNN architecture to be implemented. Similarly, speed and stability optimisations are vital for determining a more accurate answer when -for instance- a value x is really tiny; leading to an optimal training procedure of the deep learning architecture.

Table 3.2 illustrates the summarised comparison between the frameworks Caffe, Theano and Torch7 performed by Jia et al. (2014). This comparison comprises the benefits offered by each framework when training deep learning algorithms. The task evaluated consists on training modern Convolutional Neural Networks (CNNs) for image recognition tasks. As observed, Caffe is the best framework between the three options. The core language of Caffe is based on C++; although it supports library interfaces for feature extraction, training, etc. in alternative languages such as Python and MATLAB. Moreover, Caffe differs from the other frameworks in which it comprises pretrained models for an easier experimentation and getting familiarity with diverse deep learning techniques. On the other side, all frameworks share benefits for training several deep learning such as CNNs either in CPU or GPU mode. However, information about speed computation is not available to determine the performance of the three options.

Comparing Theano and Torch7 there are not enough references to determine which one surpass the performance of the other framework. The only noticeable difference is the core language each framework is based on. However, Bastien et al. (2012b) suggest that despite these two frameworks share diverse features such as fast execution speed, most of the times Theano has proven a better performance than Torch7.

Table 3.2: Comparison of leading deep learning frameworks training modern CNNs models (Jia et al., 2014).

Framework	Core language	Bindings	CPU	GPU	Open source	Training	Pretrained models	Development
Theano	Python		√	√	√	√	×	Distributed
Caffe	C++	Python, MATLAB	√	√	√	√	√	Distributed
Torch7	Lua		√	√	√	√	×	Distributed

3.2.2 Common problems when implementing Framework for DNN-based speech synthesis

After evaluating the most important benefits and limitations of Caffe, Theano and Torch7, we decided to use Caffe as our main deep learning framework. However, there were diverse limitations that forced us to switch to the Theano framework. Despite the architecture of Caffe is robust enough for its application in the Speech Synthesis field, the main focus of its applications found is based on Image recognition tasks. Therefore, the collection of useful documentation for building TTS systems using Caffe is limited. In addition, meeting all the prerequisites for installing and using Caffe was the most difficult thing to accomplish since we were not able to completely install all the libraries required for using this framework. For this reason, it was necessary to switch into a framework with more documentation and applications in the field of speech synthesis as well as to a framework already installed in the facilities of the university: Theano.

3.3 Design of alternative Multi-speaker DNN-based architecture

In order to design the model of our architecture we analysed the benefits and limitations of the multi-speaker DNN architecture proposed by Yuchen et al. (2015) due to this is the unique research work that we find as a reference of multi-speaker DNN modelling applied to TTS systems. Based on these analysis, we determined which factors of this architecture can be included in our model. A factor was included if it proves to have a positive impact on the performance of the multi-speaker DNN model. On the other side, when a factor affects the performance of the DNN model we consider using an alternative implementation that could solve the limitation of this architecture.

3.3.1 Analysis of Multi-speaker DNN architecture proposed by Yuchen et al. (2015)

The multi-speaker DNN model proposed and evaluated by Yuchen et al. (2015) is a multi-speaker architecture used for a TTS system of Mandarin language. They used phonetically and prosodically rich data of a training corpus in Mandarin language to train and evaluate the architecture proposed. The model they propose takes advantage of techniques of multi-task learning and speaker adaptation. This means that this architecture share the learning of each speaker during training stage in order to find the optimal multi-speaker model that represents the acoustic space formed by the speech features found in the multi-speaker corpus. To exploit these techniques, the architecture is decomposed in two parts. The first part models the linguistic transformation of the speech corpus. The second part is dedicated to the acoustic modelling of the speech features. The DNN architecture is set with 3 shared hidden layers of 512 nodes each among all the speakers. For the acoustic modelling each speaker has a dedicated output layer that is in charge of represent the very particular speech characteristics of the given speaker.

The results obtained by Yuchen et al. (2015) have proven promising results when the architecture is trained with up to 4 Mandarin native speakers. In the set of experiments comprising these number of speakers, the multi-speaker architecture demonstrates better performance than a speaker-dependent DNN architecture when modelling the acoustic models of each Chinese speaker. The metrics used for their evaluation comprises objective evaluation metrics such as RMSE in F_0 and normalised spectrum in log spectral distance (LSD) as well as subjective evaluations in which the output of 50 samples of both a speaker-dependent TTS system and a multi-speaker TTS system were assessed by 10 native Mandarin subjects. Both objective and subjective evaluatios demonstrate a better performance with the multi-speaker TTS system. Objective measures present over 1% up to 10% of improvement in comparison to the speaker-dependent TTS system. In subjective testings multi-speaker TTS system has a preference of 53% of the times whilst speaker-dependent model has a rate of 32% of preference.

3.3.1.1 Limitations of Multi-speaker DNN model

The model proposed by Yuchen et al. (2015) has proven promising results when training a DNN architecture with recordings of up to 4 non-native English speakers. However, the condition that all speakers need to meet is to have the same Mandarin regional accent. When a speaker with different regional accent is included in the training corpus, the performance of the DNN architecture is affected considerably. This means that the speaker-dependent model has better results in comparison to the objective metrics used for evaluation as well as in subjective evaluations since speaker-dependent model has a preference of 45% over the 34% of the multi-speaker model. Thus, the principal limitation of this system is that this architecture cannot model a better representation of a training data set when the speakers have different abstract properties such as the regional accent.

3.3.2 Considerations for designing robust multi-speaker DNN models for English language

Based on the results and limitations presented in the multi-speaker DNN architecture we take diverse considerations for designing a more robust multi-speaker DNN model than the architecture presented by Yuchen et al. (2015). A major difference is the training corpus used. The linguistic and prosodic properties of Mandarin language are significatively different to the characteristics of English language. The depth of the architecture proposed by Yuchen et al. (2015) contains an architecture of 3 shared hidden layers with a dedicated output layer for each speaker of the training corpus.

According to these considerations we propose to incorporate a better parametric model that improve the acoustic modelling of a multiple speaker corpus in English language. We determined that we can exploit two important approaches proposed by Yuchen et al. (2015): 1) We can exploit multi-task learning in order to help the model to represent the optimal acoustic space for a multi-speaker corpus and 2) We can based our model in the layer-structure model used in their DNN model by including a further decomposition of the hidden layers dedicated to better represent the linguistic transformation.

Building our proposed alternative multi-speaker DNN-based approach comprises the use of important topics already considered in both Deep Learning applications as well as in Speech Synthesis and Speech Recognition systems. Below we list the main approaches considered when devising our proposal as well as their influence in shaping a robust and more flexible architecture that optimally can enhance the quality of synthethic speech produced by multi-speaker TTS systems.

The main approaches considered during the design of the DNN model proposed in this work are the following:

1. Speaker Dependent TTS systems
2. Multi-task Learning

3.3.3 Multi-speaker TTS systems

Zen et al. (2013) and Yuchen et al. (2015) suggest that the conventional manner to build a robust multi-speaker DNN architecture should be formed by a set of multiple speaker dependent models (as shown in Figure 3.2) . Then as discussed in section 2.2.4 this approach -adapted to DNN models- focuses on constructing a specific DNN model for each of the speakers included in the training corpus.

A naive representation of a Multi-speaker dependent model is illustrated in Figure 3.2. It is formed by a DNN model of the form of a speaker-dependent DNN architecture. However, it differs from the speaker-dependent approach in the manner we train the model. Traditional speaker-dependent techniques uniquely consider the speech properties of one speaker to define the optimal acoustic model that represent the speech properties of that speaker. The naive approach considers the speech properties of more than one speaker to define the optimal acoustic space of all speakers. This means that the speech features of all speakers are used to build an averaged voiced formed of the combination of the acoustic space of each speaker of the training corpus.

The major disadvantage of this approach is that combining all the speech properties of the multiple speakers does not lead to an optimal representation of the acoustic model. The reason of this problem is because some features of each speaker are very specific, and therefore, independent to other speakers, which cause that the averaged voice contains an acoustic space disturbed (i.e. very different to the average representation of a human acoustic space).

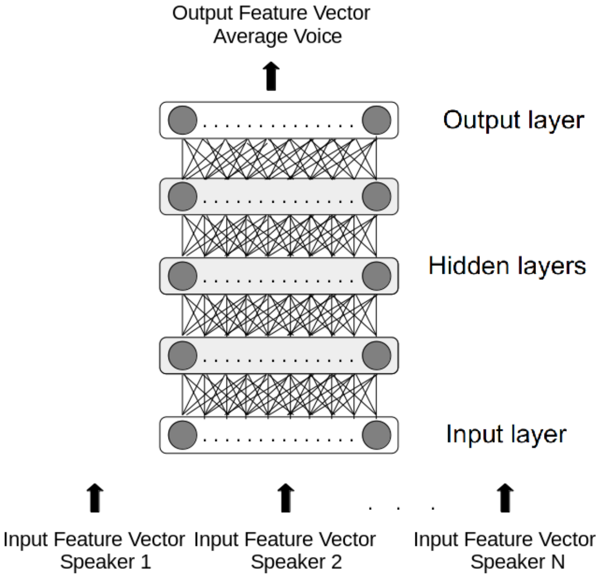

Figure 3.2: Naive representation of a multi-speaker DNN architecture formed by N speaker-dependent models.SD model indicates an specific and independent speaker-dependent included into a multi-speaker architecture.

3.3.3.1 Speaker Dependent approach

Constructing a particular model for a given speaker yields to the optimal modeling of natural variations among the speech recordings of the speaker (e.g. pitch, coarticulation, etc) (Taylor, 2009). A robust representation of these features can lead the model to mimic important aspects of the speaker‘s speech such as the intonation, regional accent, etc. Then, by incorporating real human speech properties, the synthetic speech produced by the TTS system can meet a fair performance in terms of the three important goals of TTS models: 1)Intelligibility 2)Naturalness and 3)Speech quality.

On the other side, one notable limitation of this approach is that the corpus required to train a speaker-dependent system needs big amounts of data in order to secure a good performance of the model (Taylor, 2009). In general, speaker-dependent corpora are limited and expensive to build. Therefore, implementing a training corpus with few templates can yield to a poor understanding and learning of the training corpus, which consequently can affect the overall performance of the TTS system.

The most important disadvantage of this approach is the strong dependence to the corpus of the given speaker. Therefore, by naively including several patterns (i.e. training data) of another speakers for training of this model might provoke a negative impact on the performance of the system rather than incorporating more robustness to the model. Therefore, in order to overcome some of the disadvantages of speaker-dependent systems such as the two problems aforementioned, it is necessary to adopt an alternative approach that allows the incorporation of data from different speakers to cover a wider range of acoustic features such as the implementation of multi-speaker DNN models.

3.3.3.2 Multi-task Learning

Multi-task learning is a popular machine learning approach that is primarily used to improve the performance of a 'universal' model that is integrated by diverse parallel models (Caruana, 1997). MTL takes advantage of the features extracted by a particular model and shared their understanding among the remaining parallel models to solve a universal task, which is related to these models. Therefore, by leveraging the domain specific information contained in each of these models, it is intended to help the model to have a better understanding of the desired task.

Similarly to the model proposed by Yuchen et al. (2015), we also include Multi-task learning as part of the training procedure for our DNN architecture. This technique has proven promising results when sharing diverse knowledge within the shared layers of the architecture presented by (Yuchen et al., 2015) (see Figure 2.6). However, our intention is to present different levels of transferred knowledge within the hidden layers of our architecture in order to generate a better representation of multi-speaker speech corpora.

3.3.4 Proposed Multi-speaker DNN Architecture

Figure 3.3 depicts the DNN architecture we propose for modeling multi-speaker corpora with diverse regional accents. The architecture is mainly based on the model presented by (Yuchen et al., 2015) (see figure 2.6).

Our approach is based on modelling multiple speaker data using multi-task learning and speaker adaptation. This means that all the speakers share diverse categories of hidden layers. However, as the properties between the set of speakers does not have an entire commonality among its speech features, we propose to have two subdivisions of shared layers in order to exploit the benefits of multi-task learning for a better generalisation of the most complex parameters of found in the speech templates.

On the one side, all the speakers of the training corpus shared *n* hidden layers, which are in charge of representing more general properties of the data such as the linguistic transformation of the speech corpora. On the other side, the second (and deeper) part of shared hidden layers is in charge of learning highly-complex transformations in a compact manner; this layer is divided into two categories:

1. A shared hidden layer for modeling accent *a* (e.g. English accent): This layer is shared between *k* speakers with regional accent *a*.
2. A shared hidden layer for modeling accent *b* (e.g. Scottish accent): This layer is shared between *l* speakers that have in common the regional accent *b*.

At the top of the model each speaker has its own output layer which is aimed at mapping the linguistic features learned by the model and adapt them to the speaker-specific acoustic space by using a linear regression function. In this manner, we consider that the output of the DNN architecture will be able to build an acoustic feature vector with accurate representations to generate synthesised speech of better quality than previous models proposed.

3.3.4.1 Training Methodology

The training procedure we use for our DNN model is based on the training methodology suggested by (Yuchen et al., 2015). We consider the backpropagation algorithm using SGD as the basis for training all the speakers corpus. However, as we take advantage of the multi-task learning approach it is required to combine and share the

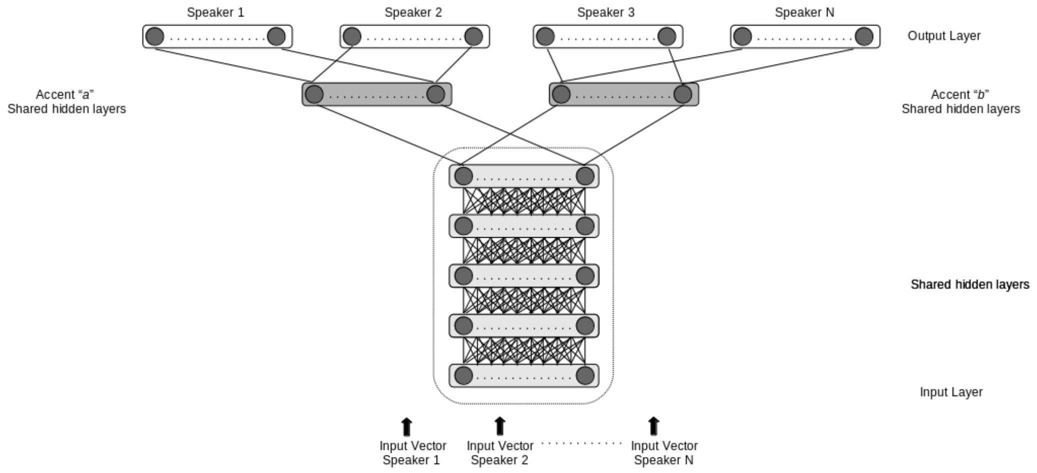

Figure 3.3: Multi-speaker DNN-based architecture proposed for modeling diverse regional accents.

knowledge acquired for each speaker at the beginning of a new epoch of the training iterative procedure. In order to secure a correct training of the model it is necessary that all the speakers are trained simultaneously as well as to shuffle the training data across all the speakers. According to the structure of the DNN model we backpropagate the error cost of each speaker through all the layers that are used for modelling the acoustic space of that speaker. This means that the weight updates of the accent *a* and accent *b* layers are independent each other and only affected by the error cost of the speakers that belong to that particular accent.

Figure 3.4 illustrates the methodology implemented for training the multi-speaker DNN approach proposed in section 3.3.4. The main steps comprises:

1. Set initial random values for each SD model. At the beginning of the training procedure we set random weights and an initial (best) cost error for initialising the training stage each SD model. As the speaker models are trained simultaneously, we set the same random weights for each SD model.

2. Load speaker feature vector. A mini-batch of each speaker data is stored temporarily to compute stochastic gradient descent (SGD) in a specific SD model.

3. Compute SGD. We compute backpropagation algorithm based on SGD. Note that we uniquely compute SGD considering the feature data of the speaker corresponding to that given SD model.

4. Comparison between best cost error and cost error of the updated model. After

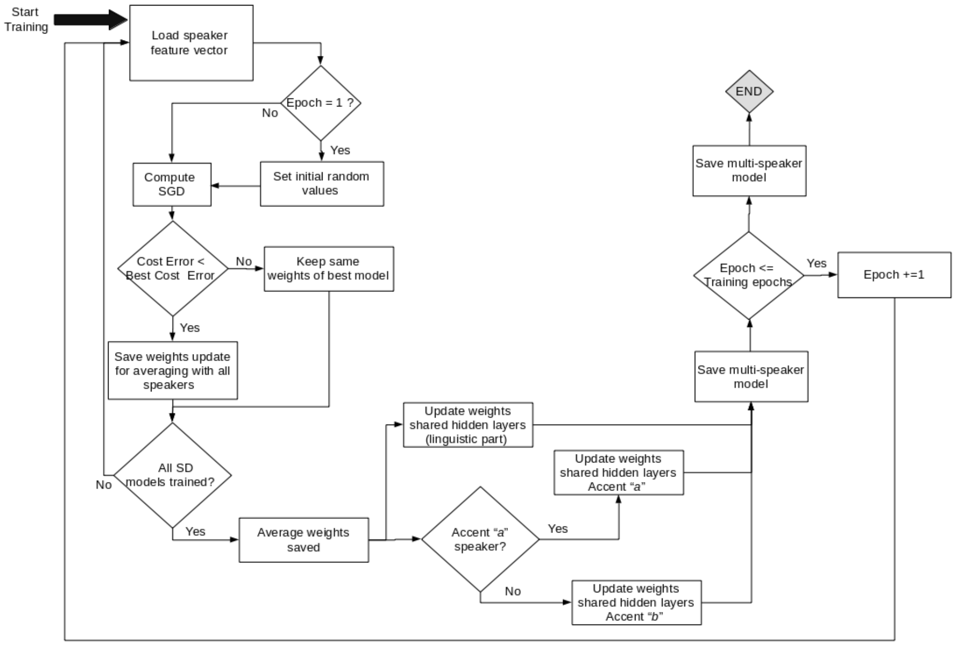
Start Training
Load speaker feature vector
Epoch = 1 ?
No
Yes
Set initial random values
Compute SGD
Cost Error < Best Cost Error
No
Keep same weights of best model
Yes
Save weights update for averaging with all speakers
All SD models trained?
No
Yes
Average weights saved
Update weights shared hidden layers (linguistic part)
Accent "a" speaker?
Yes
Update weights shared hidden layers Accent "a"
No
Update weights shared hidden layers Accent "b"
Save multi-speaker model
Epoch <= Training epochs
Yes
Epoch +=1
Save multi-speaker model
END

Figure 3.4: Diagram of training methodology implemented for multi-speaker DNN architecture.

computing SGD for a particular SD model, we evaluate its cost function with the best cost function error we obtained in previous epochs of the training stage. If the new model proves a better (i.e. a lower) cost error in comparison to the best cost function error obtained in the previous epochs, we save the updated parameters of the model. The saved parameters are stored in two categories:

- Parameters of the linguistic shared hidden layers.
- Parameters of the shared hidden layer for the accent of the corresponding SD model.

5. Check whether all SD models have been trained. We iterate over steps 2 and 3 for all the *n* speakers of the training corpus during each new epoch of the training stage.

6. Average weights saved. After computing SGD for the *n* speakers of the model, we average the weights of the shared hidden layers to update the shared hidden layers according to its own criteria. On the one side, we average the weights of the shared linguistic hidden layers considering the influence of all the speakers of the training corpus. On the other side, we average the weights of both the shared accent *a* hidden layer and the shared *b* hidden layer based on the speakers

that belong to that accent.

7. Update weights of shared hidden layers. Updating the parameters of the multi-speaker model is divided in two steps performed simultaneously. First, we update the parameters of the shared hidden layers that model the linguistic part of the speakers corpora. Similarly, we update the parameters of accent a and accent b according to the influence of the updated parameters of the speakers involved in each of these accent layers. Note that the parameters of the output layer of each model are not averaged, they are independently updated after performing step 4.

8. Save multi-speaker model. We set the new multi-speaker model as the 'universal' and best model by considering the updated parameters of all the shared hidden layers.

9. Evaluation of the number of epochs performed. Evaluate the number of iterations performed in the training procedure. If the number of epochs is smaller than the l number of epochs we repeat steps 2 to 9. Note that we assume that we have enough training data to perform a value of l epochs such that we can determine the best model we can build with the training corpus.

10. Set optimal multi-speaker DNN model. After l number of training epochs it is expected that the cost error has converged to a value very similar or closer to the optimal parameters for the best performance of the DNN model. Therefore, we define this model as the optimal DNN model and we finish the training methodology of the DNN architecture.

After concluding the training methodology it is expected that it leads to find the optimal DNN model that proves its ideal performance when creating artificial speech. However, it is necessary to be aware that diverse factors can affect or reduce the success of the training methodology. One of the most important factors is the quality of the training corpus; an accurate conversion of the speech waveform into its corresponding feature vector is crucial for the success of the training procedure. The reason of this condition is because this representation is the knowledge that the training procedure tries to represent, therefore, if the accuracy of this representation is not accurate, the acoustic modelling performed by the training approach does not represent the optimal acoustic model of the multiple speaker corpus.

Chapter 4

Evaluation

This chapter discusses the performance of the multi-speaker DNN-based model originally proposed in Chapter 3. The overall discussion of the architecture proposed is based on robustness, functionality and accuracy of acoustic models. The chapter is divided into three parts: 1) An introductory explanation of the objective evaluation metric we used to assess the DNN model. 2) A description of the set of experiments we performed and 3) An analysis and discussion of results.

From the set of experiments performed they are divided in three types:

1. Experiments of speaker-dependent DNN models. We built diverse configurations of DNN models. From these experiments we select the best configuration to use it as a referenced configuration for building a robust multi-speaker DNN model.

2. Experiments of naive multi-speaker DNN models. We trained diverse multi-speaker DNN models using the most simplistic way to construct a multi speaker model. This manner is by simply join a collection of multiple speakers corpus to construct an averaged and synthetic output voice.

3. Experiments of Multi-speaker dependent DNN models. We evaluate diverse multi-speaker models based on the proposed configuration we defined in chapter 3. These experiments illustrate two training approaches of the DNN model. The first one based on the training procedure evaluated by Yuchen et al. (2015) and the second one is an alternative approach that we proposed to improve the performance and overcome the most important limitations of the first training

methodology.

A TTS systems can be seen as a set of modules addressing particular tasks that in conjunction create a universal task, particularly, artificial speech with some level of naturalness and intelligibility. Commonly, testing the performance of these systems is a challenging task that needs to be aimed at determining their suitability and accuracy to the particular task we are interested in.

According to Taylor (2009), it is possible to evaluate both the overall performance of a TTS system by using **system tests** and the performance of a particular unit of the system by using **unit testings**. System tests usually measure intelligibility and naturalness - the two main goals of a TTS system. Unit testing comprises of objective and subjective evaluations that attempt to determine continuous or numeric values (e.g. root mean squared error in F_0) or they can assess the speech output by using listening tests evaluated by humans.

The manner we propose to evaluate the performance of our model is by considering Unit testing evaluations, as our main interest is to determine the accuracy of the acoustic models generated by the DNN architecture. This is motivated due to the fact that each component in the TTS system has a particular score and performance that is influenced and affected by factors that might not be directly related to the performance of unit of interest. For this reason, a poor performance in a unit different to the DNN architecture can alter the results and, therefore, the analysis about the performance of the DNN model. Thus, by uniquely testing the accuracy of the DNN model we can bring a more accurate and factual analysis of the behaviour and modelling performed by the DNN architecture under optimal conditions when training and testing.

4.1 Measure of Accuracy

We suggest to evaluate the performance of our system by using root mean squared error (RMSE) between the produced output of the DNN model and the reference or target output Damper (2001). The RMSE magnitude is given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y^n - f(x^n))^2} \tag{4.1}$$

where $f(\mathbf{x})$ is the output generated by the DNN model, y^n is the reference output and N is the dimension of the output feature vector.

4.2 Experiments proposed

In order to evaluate the functionality and performance of our proposal, we define three sets of experiments. The first set is focused at measuring the accuracy of acoustic models using speaker dependent DNN-based models. The second set comprises the performance of a naive multi-speaker DNN-based model. The third set is intended to evaluate the accuracy of the multi-speaker DNN-based model we propose in section 3.3.4.

4.2.1 Experiments of Speaker-dependent DNN model

The first set of experiments is aimed at defining two important conditions of the model. First, we determine the optimal DNN architecture of speaker-dependent DNN-based models considering the auxiliary tools we have already detailed in Chapter 3 such as the training corpus and the deep learning framework. Second, we establish the performance of a conventional speaker-dependent model based on DNNs under the same training and implementation conditions.

4.2.1.1 Optimal architecture of speaker-dependent DNN-based model

In order to determine the optimal DNN architecture for a speaker-dependent model we are required to define the ideal number of hidden layers and nodes of each hidden layer. The criteria we consider to determine the ideal architecture is based on the accuracy of the DNN model as well as on the time for training the architecture. We execute three types of DNN architectures using 4 different number of nodes or units. These configurations are based on diverse DNN models presented by the research community of the speech synthesis and machine learning fields (Yuchen et al., 2015)(Wu, 2015)(Bengio, 2009). Considering the set of speakers of our training corpus (see Table 3.1) we randomly select 2 speakers for training the 3 types of DNN architectures. One speaker with English accent and another speaker with Scottish accent. In total, we run 24 possible DNN models: each model is formed by 5, 6 or 7 hidden layers that are responsible of modelling the linguistic features of the input feature vector. In addition, each model type considers either 128, 256, 512 or 1024 units as part of the hidden layers. By building a broad set of architectures for each possible regional accent, we are able to determine the configuration that optimally perform the acoustic modelling task.

Experiments Conditions

As mentioned before we use the training corpus of 2 speakers with different accent and gender from Table 3.1. The corpus of the female speaker “p389” is used for training a speaker-dependent model with English accent. This training corpus contains 406 recording samples. Oppositely, the data set of the male speaker “p395” is adopted to train the speaker-dependent model with Scottish accent. This collection contains 389 recording samples.

The input feature vector is a 601 dimensional vector and the acoustic feature vector is 259 dimensions. A fixed learning rate of 0.0002 was set to the learning configuration.

Wu (2015) has evaluated the adaptability of DNN-based speech synthesis models to statistical parametric speech synthesis. He found that using speaker adaptation to the output layer of the DNN model improves the performance of DNN models when modelling acoustic representations of data. For this reason, as shown below, we have use most of the setting information shown in this research work in order to obtain a better representation of our DNN models when performing acoustic modelling.

We set 30 number of epochs when training the models . The batch size considered is set to 256 units. The hidden activation function is a “tanh” and the output activation function is a linear regression function. The weights of the model are randomly initialised and the training corpus is shuffled. From each speaker we consider 80% of its collection as training data and the remaining 20% as testing data. Moreover, a dedicated GPU server 4x GTX TITAN (2688 cores 6G ram) with 32 GB of memory is used to compute SGD when training each of the models presented in this set of experiments.

Results

Table 4.1 and Table 4.2 illustrate the results obtained for an English accent speaker-dependent model and for a Scottish accent speaker-dependent model. We considered

3 metrics for determining the performance of the model:

1. Number of epochs during training the model
2. Validation error of the model considering a test data set for validation
3. Overall Training Time that determines the minutes taken by the SGD to train the model.

In both cases (with English accent or Scottish accent corpus), the architecture with the best overall performance is a DNN model containing 6 hidden layers with 512 nodes each. Those models with 5 and 7 hidden layers usually show a worse performance in comparison to a 6 hidden layered model. This suggests that a shallow or very deep model can result in a poor representation of the training corpus. A shallow architecture (e.g. 5 hidden layered model) is unable to have a robust generalisation of the model as it only learns the most representative features of the data. A very deep architecture (e.g. 7 hidden layered model) memorises very particular and abstract features of the training data but it is also the case that the model learns noise contained in these training samples; causing that the model does not learn a generalised representation of the data and perform poorly when testing the model with unknown or testing samples. In terms of validation error the worst architecture in both accents is a 5 hidden layer with 128 nodes in each layer.

Training an English accent model is typically more effective and faster than training a Scottish accent model. The number of epochs in both architectures do not present a noticeable or direct relation to the number of hidden layers or the number of nodes of the model. However, the number of epochs considered in a Scottish accent model is usually bigger than those of the English accent models. For instance, a Scottish model of 6 hidden layers and 512 nodes takes 20.58 min to obtain a validation error of 250.69 units whereas an English accent model only takes 9.8 min to obtain a validation error of 235.161.

Based on the results and analysis presented in this section, the optimal speaker-dependent model for these accents is a DNN model of 6 hidden layers with 512 nodes each. Compared to the remaining models the overall training time of this configuration has the smallest value as well as the performance of the model is one of the most effective presented in Table 4.1 and Table 4.2. Therefore, we decided to use these configuration for devising our multi-speaker dependent model, as the efficiency and performance of a 6 hidden-layered model can help us to analyse the optimal performance of the model considering particular factors such as the size and data of our training corpus.

Table 4.1: Training performance results of 12 speaker-dependent models with English accent. Based on the overall performance, the best architecture is a 6 hidden layer model with 512 nodes each.

Regional Accent	English Accent											
Hidden Layers	5				6				7			
Nodes	128	256	512	1024	128	256	512	1024	128	256	512	1024
Epochs	30	23	16	16	27	27	16	16	24	29	16	16
Validation Error (RMSE)	238.637	236.856	235.551	234.56	238.303	236.446	235.161	233.9	238.377	236.156	234.557	233.366
Overall Training Time (min)	13.620	14.981	9.866	10.18	15.529	16.448	9.805	10.52	14.675	17.657	9.908	11.333

Table 4.2: Training performance results of 12 speaker-dependent models with Scottish accent. Based on the overall performance, the best architecture is a 6 hidden layer model with 512 nodes each.

Regional Accent	Scottish Accent											
Hidden Layers	5				6				7			
Nodes	128	256	512	1024	128	256	512	1024	128	256	512	1024
Epochs	25	26	25	26	24	27	25	26	27	26	25	28
Validation Error (RMSE)	254.269	252.704	251.633	250.413	254.158	252.188	250.692	249.314	253.77	251.725	249.791	248.233
Overall Training Time (min)	14.9345	21.216	19.967	22.56	18.5231	21.598	20.580	25.234	21.450	22.504	22.278	25.471

4.2.2 Naive Multi-speaker DNN model

The second set of experiments is focused on demonstrating the performance of a naive multi-speaker DNN model. This naive model has a similar configuration to the form of a speaker-dependent architecture. However, the naive method differs from the traditional speaker-dependent approach in which this DNN model is trained with a collection of diverse speakers recordings to build an average output feature vector.

We called this approach "naive" multi-speaker DNN model due to we use the most simplistic manner to train a multi-speaker DNN model. According to the speaker-dependent approach, it is expected that we train a speaker-dependent model by only using a training corpus of a specific speaker so that the model can extract the speech characteristics of that speaker and represent this features when producing the artificial speech. On the other hand, based on the naive multi-speaker approach, if we use the architecture of a speaker-dependent model and we train this model with a set of diverse speaker corpus, it can affect the performance of this naive multi-speaker model since the model might not might not be robust enough to create an artificial speech output with good quality.

The structure of this set of experiments comprises training two different sets of experiments. A set dedicated to an English accent model and another set for a Scottish accent model. Each set includes a training corpus of 2,3,4 and 5 speakers. The manner we combine the corpus is illustrated in Table 4.3. For instance, a multi-speaker model of three speakers with English accent includes the corpus of speakers “p389”, “p337” and “p291”. An Scottish multi-speaker model with the same number of speakers is trained with the collection of speakers “p395”,“p397” and “p46”.

Table 4.3: Set of speakers included in each of the naive multi-speaker models.

Regional Accent	English	Scottish
Number of speakers	Speakers included	Speakers Included
1	p389	p395
2	p389, p337	p395, p397
3	p389, p337, p291	p395, p397, p46
4	p389, p337, p291, p273	p395, p397, p46, 327
5	p389, p337, p291, p273, 226	p395, p397, p46, p327, p114

Experiments Conditions

In order to train the models we use the ten speakers of our training corpus (see Table 3.1). The collection of all speakers contains more than 385 recordings each. The DNN architectures are set with 6 hidden layers of 512 units, which are mainly in charge of modelling the linguistic features of the training corpus. We consider the same experiment conditions used in section 4.2.1.1: an input feature vector of 601 dimensions with its corresponding 259 dimensional acoustic feature vector; 30 number of epochs when training the models. We run the training procedure in a GPU mode using the same GPU server 4x GTX TITAN as well as a batch size of 256 and a buffer size of 5000 units.

Results

As shown in Table 4.4, the performance of naive multi-speaker models with English accent usually present a validation error slightly bigger than the speaker-dependent reference, which is included and referenced as Speakers = 1. The model with the biggest error is a 2-speaker model with 235.549 units and 19 training epochs. On the contrary, the model with the best performance is the 5-speaker model with 235.001 units and 20 training epochs. The overall training time is in all the cases bigger than the time spent to train a simple speaker-dependent model. The overall training time vary from 23.219 minutes up to 73.068 according to the number of speakers used for training the model.

On the other side, Table 4.5 illustrates the performance of the naive multi-speaker models trained with Scottish accent corpora. Unlike the speaker-dependent reference (see metrics of speakers = 1 in Table 4.5), the RMSE of all the models is smaller. This condition suggest that increasing the number of speakers can improve the performance of a naive multi-speaker model. However, the number of epochs used for training the model indicates in some cases such as in speaker = 2, that the training procedure takes less number of epochs to train the model. Therefore, it is very likely that despite having a more accurate output in terms of RMSE, the generalisation learned by the model does not represent properly the features of the training corpus.

Based on the results of Table 4.4 and Table 4.5 we have found an unexpected behaviour of the naive multi-speaker approach. The performance of the naive multi-speaker models either with English or Scottish accent demonstrate a better performance than the speaker-dependent model in terms of RMSE. These RMSE values controvert theory of traditional TTS models (e.g. HMMs) based on a speaker-dependent or a multi-speaker approach. Ideally, a speaker-dependent model can optimise the acoustic features of the training data and, therefore, bring the best representation of a single target speaker when producing synthetic speech. A multi-speaker model requires devising a more robust architecture to extract acoustic feature commonalities and build a system with a good performance when producing synthetic speech. In general, a naive multi-speaker model (e.g. based on HMMs) cannot perform better than a speaker-dependent or a multi-speaker model since the model cannot generate an accurate representation of the independent variations found in the acoustic space of each speaker. In consequence, as the generalisation of the training corpus is poor, the model should demonstrate a worse performance than the speaker-dependent or the multi-speaker models.

We hypothesise three potential factors that can be generating the results shown in Table 4.4 and Table 4.5: 1) The measure of accuracy 2) The size of the corpus and 3) The data of multiple speakers is normalised . We are using RMSE to measure the distance between the output reference and the output generated by the model. This measure is useful for having a good reference of the performance of the system. However, a notable limitation is that RMSE does not provide much representative information of the overall output signal. Therefore, despite having obtained better numeric results the output feature vector does not represent more accurately the properties of the speech. On the other side, the size of the corpus can influence the performance of the training procedure. As we are including data of more than one speaker, the model has more samples to build a more accurate representation of the acoustic space of the multi-speaker corpus. The last hypothesis which is related to the second hypothesis, can be explained by the manner we have built the training corpus. We have collected the data of 10 speakers, some of the speech properties of these speakers are common between these speakers. Therefore, the naive model can normalise these properties and obtain a better performance than the speaker-dependent model. Therefore, we can discriminate or support this hypothesis by trying to train the naive multi-speaker model with the corpus of speakers that do not present an important level of relation in the acoustic space and determine whether this hypothesis is the cause of the behaviour presented by the naive multi-speaker model.

In order to solve these potential limitations, we suggest using alternative testing methods (e.g. mel-cepstral distortion), which involves the interaction of the DNN model with other units of the TTS system such as parameter generation and waveform synthesis units to have a more accurate evaluation of this model either objectively or subjectively. However, we are not using these alternative testing methods due to building and correctly implementing these units would require spending important time considered to meet the major goal of this project.

Table 4.4: Performance results of 4 naive multi-speaker models with English accent. A speaker-dependent model (speaker = 1) is included as a reference for comparison with the multi-speaker models.

Regional Accent	English Accent				
Hidden Layers	6				
Nodes	512				
Speakers	1	2	3	4	5
Epochs	16	19	21	19	20
Validation Error (RMSE)	235.161	235.549	235.31	234.644	235.001
Overall Training Time (min)	9.805	23.219	43.339	53.176	73.068

Table 4.5: Performance results of 4 naive multi-speaker models with Scottish accent. A speaker-dependent model (speaker = 1) is included as a reference for comparison with the multi-speaker models.

Regional Accent	Scottish Accent				
Hidden Layers	6				
Nodes	512				
Speakers	1	2	3	4	5
Epochs	25	20	26	25	29
Validation Error (RMSE)	250.692	249.219	245.294	249.914	250.296
Overall Training Time (min)	20.580	28.449	41.377	76.027	86.359

4.2.3 Multi Speaker Dependent

The final and most important set of experiments is dedicated to evaluating the performance of our proposed Multi-speaker DNN model. To perform this evaluation we have divided the set of experiments in two parts. The first illustrates the performance of our architecture when it is trained with a very similar procedure to the approach proposed by Yuchen et al. (2015). This procedure considers sharing the knowledge of the trained model by summing the updates of the weights of the DNN model when it is trained by SGD. The second part of experiments proposes an alternative training procedure to train the DNN architecture. This procedure considers sharing the knowledge when training the model by averaging the updates of the weights of the DNN architecture when it is trained by SGD.

4.2.3.1 Training procedure based on methodology proposed by Yuchen et al. (2015)

The training procedure proposed by Yuchen et al. (2015) is analysed with the following set of experiments. These experiments are based on a training corpus with different number of speakers and listed below.

DNN model trained with:

1. 2 speakers collection. 1 speaker with English accent and 1 speaker with Scottish accent.

2. 4 speakers collection. 2 English accent speakers and 2 Scottish accent speakers.
3. 6 speakers collection. 3 speakers with English accent and 3 speakers with Scottish accent.
4. 10 speakers collection. 5 English accent speakers and 5 Scottish accent speakers.

The manner we combine the corpora collection is the same configuration that we used in the previous set of experiments (see section 4.2.2). As illustrated in Table 4.3 there are 5 possible multi-speaker models that we can build with these data sets, however, we only use a multi-speaker model of 1,2,3 and 5 speakers for each accent within their training procedure in order to illustrate the most representative results obtained.

Experiments Conditions

The collection of speech recordings differs from model to model, since the training samples contain between 380 and 415 recordings depending on the speaker. Each DNN architecture is set with 6 hidden layers of 512 units each. The hidden activation function used is a “tanh” function whereas a linear regression functions is considered as the output activation function. The learning rate varies according the hierarchy of the layers: the first five shared hidden layers have a learning rate of 0.00014 and the accent and output layers have a learning rate of 0.0001. The weights of the model are randomly initialised and the training corpus is shuffled (as suggested in (Zen et al., 2013)). The batch size is set to 256 and the buffer size is equal to 5000 units. As in the other sets of experiments the input feature vector is a 601 dimensional vector and the output acoustic vector is a 259 dimensional vector. The computation mode is set to GPU using the GPU server 4x GTX TITAN.

Results

Figure 4.1 illustrates the performance of a 2-speaker DNN model considering more than 15 epochs during the training stage. According to this figure, the optimal performance of the model is reached when training the model for 11 epochs as the RMSE for English accent speaker is close to 245 and over 262 for the Scottish accent speaker. English accent speaker shows a convex behaviour since after this epoch the performance starts decreasing continually. Scottish accent speaker demonstrates a slower convergence rate; suggesting that this accent may contain more abstract linguistic features that prohibit a faster learning by the DNN model.

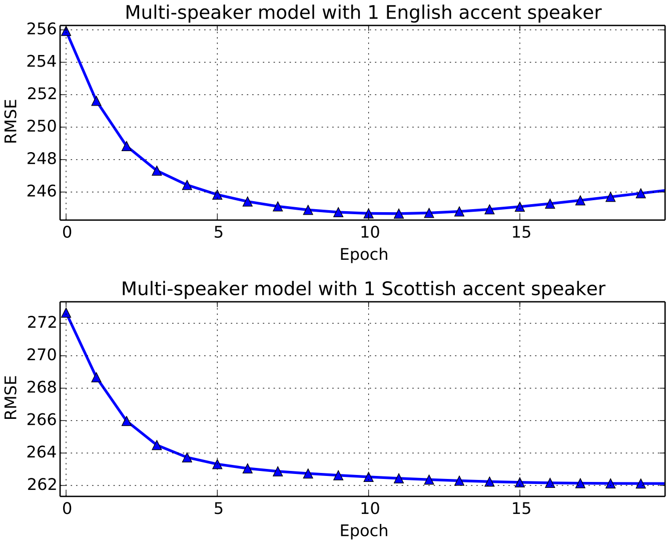

Figure 4.1: Multi-speaker DNN-based architecture training performance. Model trained with 1 English accent speaker and 1 Scottish accent speaker.

Figure 4.2 depicts the behaviour of the multi-speaker model when it is trained with four speakers of Scottish and English accent. Based on the overall performance of the model, the optimal parameters for the DNN architecture are obtained between the framework comprising from 11 to 14 epochs. Speaker 1 and 2 of English accent have a RMSE of over 244 and 248 units respectively. Scottish accent speakers have a bigger RMSE: speaker 1 has over 262 units and speaker 2 has 251 units. According to the performance of the English accent speakers, we can determine that a particular model (in this case speaker 2) can saturate the learning behaviour of another similar model (i.e. speaker 1) when sharing more knowledge than necessary for building the optimal DNN model. The reason is based on the performance of both models after epoch 13, where speaker 1 tends to have a similar behaviour than the behaviour of speaker 2, which is represented by a similar RMSE value of both models. On the contrary, it is also noticeable that despite English and Scottish speakers share important speech characteristics, the influence of speakers of one accent (e.g. English accent) do not clearly influence either positively or badly on the performance of speakers of another accent (e.g. Scottish accent).

Figure 4.3 supports the analysis we have described for the 4-speaker DNN model,
]

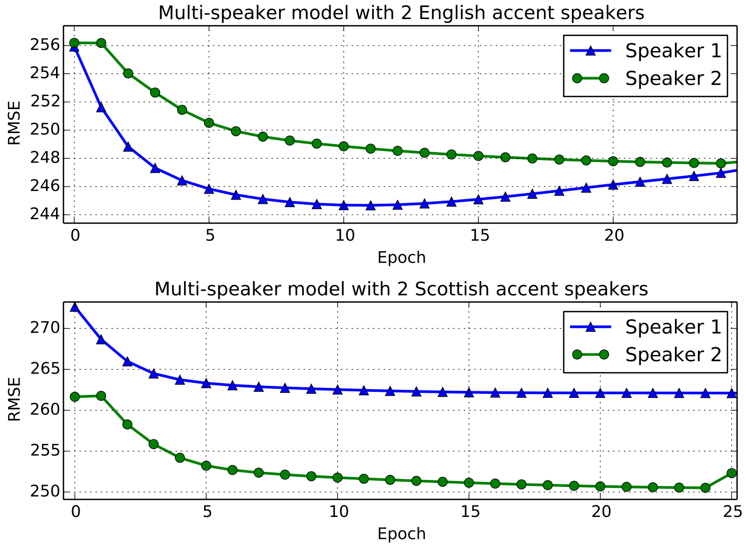

Figure 4.2: Multi-speaker DNN-based architecture training performance. Model trained with 2 English accent speakers and 2 Scottish accent speaker.

which is illustrated in 4.2. On the one side, when analysing the performance of the 3 speakers of any of the two possible accents, it is clear that these models cannot influence another model with slightly different properties even when they are sharing important factors such as accent and gender. Alternatively, when analysing the overall performance of the 6 speakers, the sharing of learning information is even more complicated, and therefore, barely noticeable. As observed in both Figures 4.3 and 4.2 the performance of speaker 1 and speaker 2 for each accent has the same performance, so the influence of speaker 3 in each type of accent is not helpful for improving the performance of the DNN model. Therefore, this behaviour contravene the major goal of the multi-speaker approach: building a more rich and robust acoustic representation using the speech properties of diverse speakers.

A very similar performance is observed by the model results shown in Figure 4.4. This model is composed of 5 English speakers and 5 Scottish speakers. For the two type of accents, speakers with similar speech properties (i.e. speakers with similar RMSE) are influenced by the learning shared of another similar speaker. For instance, speakers 1,2,4 and 5 for the case of English accent tend to point to a very similar RMSE, which is interpreted as an output feature vector of very similar acoustic features. In the same manner, speakers 2, 4 and 5 of Scottish accent behave in a similar manner to the behaviour of English accent speakers. Speaker 3 of English accent speakers is

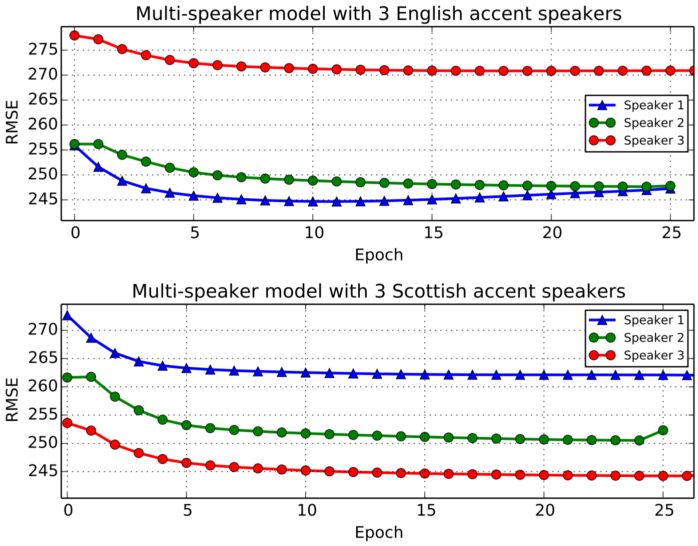

Figure 4.3: Multi-speaker DNN-based architecture training performance. Model trained with 3 English accent speakers and 3 Scottish accent speaker.

the speaker have a RMSE very different to the four remaining English accent speakers. This can indicate that the speech properties of this speaker are significantly different to the other speakers. For the case of Scottish accent, the speaker with biggest difference of speech properties is speaker 1. Based on the RMSE values of both accents, the speech features of English accent speakers are easier to model than the properties found within the collection of Scottish accent speakers.

4.2.3.2 Alternative training procedure: Averaged Weights

The alternative training procedure we propose is evaluated by the 4 experiment results described below.

Based on the performance of the training procedure performed by Yuchen et al. (2015) when training a multi-speaker DNN model, we suggest an alternative training procedure to exploit the multi-task learning benefits and build a more robust DNN model. The manner we propose to train the model is using the backpropagation algorithm with the SGD approach. As explained in section 3.3.4, the difference between our approach and the backpropagation algorithm used by Yuchen et al. (2015) for training the DNN model, is that we average and, subsequently, update the shared parameters of the DNN model every time we run a new training epoch, rather than simply updating

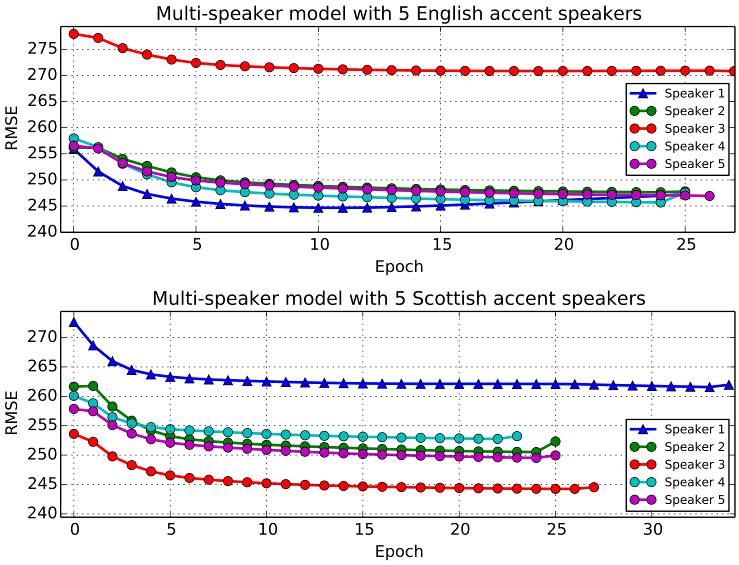

Figure 4.4: Multi-speaker DNN-based architecture training performance. Model trained with 5 English accent speakers and 5 Scottish accent speaker.

the shared parameters as it is done in the conventional method used by Yuchen et al. (2015). For each epoch, we simultaneously train all the speakers and use the same number of samples to train the corresponding layers of the DNN architecture to each speaker. With these samples we backpropagate the cost error using SGD and store the weights obtained of these speaker. After all speakers have performed its corresponding backpropagation, we average the weights of all of these updates and set as a factual update for the multi-speaker model this average computation.

As in the case of the experiments of section 4.2.3.1, each of this results is based on a training corpus with multiple number of speakers. Thus, the first experiment analyses the interaction between 2 speakers of different accent: 1 English accent speaker and 1 Scottish accent speaker. The second experiment evaluates the behaviour of a 4-speaker DNN model (two English accent speakers and two Scottish accent speakers). The third experiment is intended to determine the performance of the DNN model when using 3 speakers of English accent and another three speakers of Scottish accent in the training procedure. The last experiment includes a training corpora of 10 speakers, where five are English accent speakers and the remaining speakers are Scottish accent speakers.

Experiments Conditions

Similarly to the experiments conditions of section 4.2.3.1 we use the same DNN architecture (6 hidden layers of 512 units each) with the same hidden and output activation functions. We consider the same learning rate for each of the shared, accent and output layers as well as the same buffer and batch size, including the GPU computation mode and the dimensions of the input and output feature vectors.

Results

The performance of a 2-speaker DNN model is depicted in Figure 4.5. The behaviour of this model shows a smooth convergence by the time more number of epochs are considered when training the model. For the case of each speaker of the model, a good performance of the model can be obtained when it is trained for at least 15 epochs. In addition, a proportional performance is observed during this evaluation. For instance, the English speaker has a proportional similarity to the accuracy of the Scottish Speaker; suggesting that the knowledge shared is useful, but more importantly, incorporated to the particular requirements of the speaker. This is represented by the initial RMSE of the English speaker (over 256 units) and its smooth convergence to a value close to 243 units in epoch 15. The initial RMSE of the Scottish speaker is near to 272 units whilst its convergence at epoch 15 is over 260. That is, both speakers have an improvement in accuracy of almost 13 units. On the other side, the modelling of the English accent speaker is better generalised than the Scottish accent speaker as the accuracy of the English speaker is smaller than the RMSE of the Scottish speaker.

Figure 4.1 illustrates the performance of a 2-speaker DNN model considering more than 15 epochs during the training stage. According to this figure, the optimal performance of the model is reached when training the model for 11 epochs, as the RMSE for English accent speaker is close to 245 and over 262 for the Scottish accent speaker. English accent speaker shows a convex behaviour since after this epoch the performance starts decreasing continually. Scottish accent speaker demonstrates a slower convergence rate; suggesting that this accent may contain more abstract linguistic features that cannot be modelled easily by the DNN model.

Figure 4.6 and Figure 4.7 depict the performance of the second and third experiment developed. As observed in Figure 4.6, the behaviour of all speakers is improving

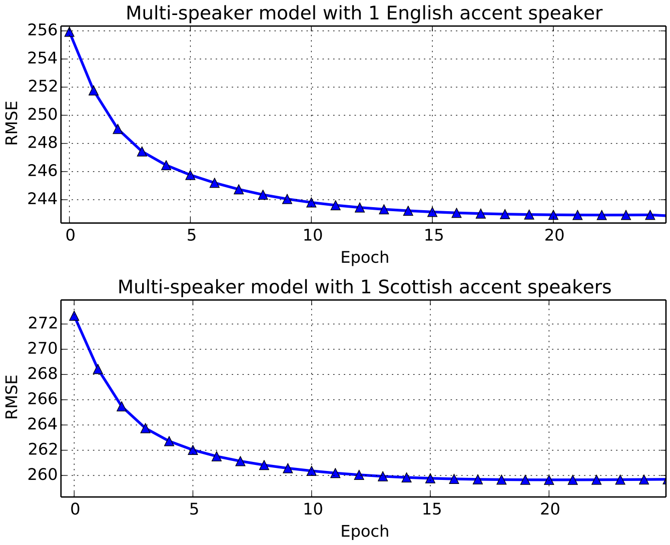

Figure 4.5: Multi-speaker DNN-based architecture training performance. Model trained with 1 English accent speaker and 1 Scottish accent speaker. Training procedure: Averaged weights.

by the time epochs are increasing. RMSE of English accent speakers at epoch 15 is between 244 and 246 whereas for the Scottish speakers the RMSE is comprised in the range of 244 to 260. A better performance is noticed at epoch 24, where all speakers have a smaller RMSE in comparison to the one obtained in epoch 15. As in the case of Figure 4.5, the relationship between the learning of a model and its influence to the rest of the speakers is still observed. For the case of the performance of the 6-speaker model (see Figure 4.7), the 6 speakers present a similar convergence behaviour throughout the training procedure of the model. However, this influence is stronger between more similar models such as the case of speaker 1 and speaker 2 of the English accent speakers. After epoch 20, all speakers present some convergence as each one of them have slightly variations in further epochs, which can be explained as the time in which the DNN model has reached its maximum sharing of learning. Thus, due to this model shows a convergence behaviour as the number of training epochs is increasing, it can suggest that by using this training procedure we can have a better representation of the acoustic space of the different speakers.

Figure 4.8 illustrates the performance of the biggest model built as it considers 10 speakers for training the muti-speaker DNN-based model. The behaviour of this model is different to the previous 3 experiments analysed for testing this approach. Each

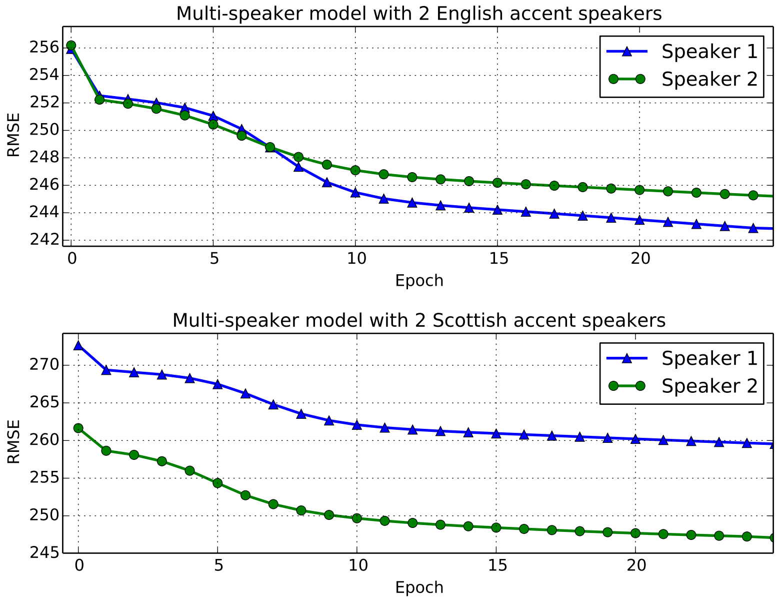

Figure 4.6: Multi-speaker DNN-based architecture training performance. Model trained with 2 English accent speaker and 2 Scottish accent speaker. Training procedure: Averaged weights.

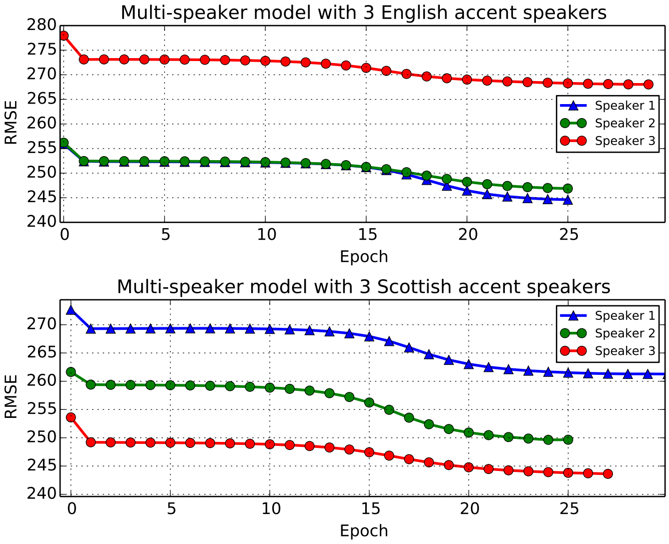

Figure 4.7: Multi-speaker DNN-based architecture training performance. Model trained with 3 English accent speaker and 3 Scottish accent speaker. Training procedure: Averaged weights.

speaker has small improvements in comparison to the random initialisation of the model. For instance, speaker 2 of the Scottish accent speakers maintains a uniform performance of RMSE with over 260 units throughout the training epochs, whereas the same speaker in the 6-speaker model (see Figure 4.7) has a more accurate performance with almost 250 units in RMSE. This overall performance may imply that our approach has some limitations when the number of speakers is big, since despite including a big sized training corpora with considerable rich speech samples, the model is not capable of representing such amount of shared learning.

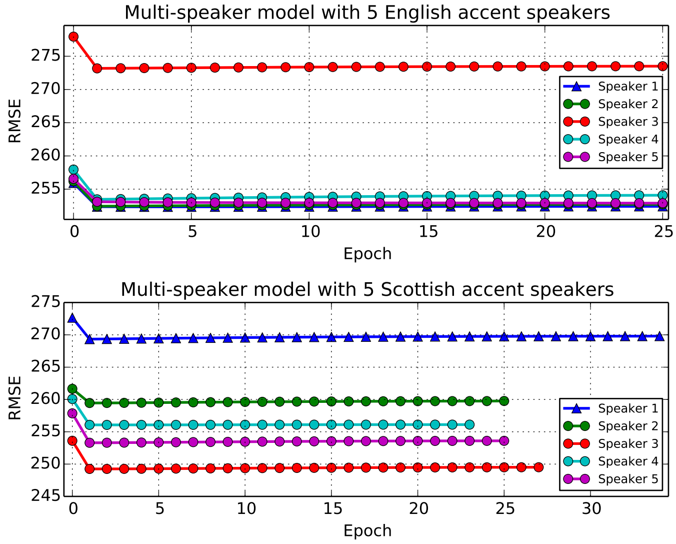

Figure 4.8: Multi-speaker DNN-based architecture training performance. Model trained with 5 English accent speaker and 5 Scottish accent speaker. Training procedure: Averaged weights.

4.3 Analysis of results

According to the analysis made to several speaker-dependent architectures, we have determined that the best architecture should include: 1) 6 shared hidden layers to modelling the input feature vector and 2) A specific output layer for each of the speakers the model has been trained with. We considered three aspects for determining the overall performance of the optimal architecture: 1) The number of epochs when training the model 2) RMSE of a testing set and 3) Overall training time. Diverse architectures proven better RMSE or overall training time, however, the combination of these aspects reduced the overall performance and, in consequence, its selection for implementing it in our approach. The reference values we have with the optimal DNN model are RMSE = 235.161 for an English accent model and RMSE = 250.692 for an Scottish accent model.

Table 4.6: Optimal performance of speaker-dependent model according to our experiments. speaker “p389” was used to trained English accent model and speaker “p395” was used to train Scottish accent model.

Optimal Speaker-dependent Performance		
Model Type	English accent	Scottish Accent
Epochs	16	25
Validation Error (RMSE)	235.161	250.692
Overall Training Time	9.805	20.580

Based on the naive multi-speaker model, we obtained unexpected results during experiments, since the RMSE in several models of English and Scottish accent is smaller than our referenced optimal speaker-dependent models. The purpose of testing the performance of the naive multi-speaker model was to determine a referenced value that explains a poor performance of a multi-speaker model. To explain the unexpected results obtained in this experimental set, we considered 2 potential factors that could affect these results: 1) The use of a simple measure of accuracy, which is the RMSE and 2) The size of the training corpus. We suggest using alternative testing methods such as mel-cepstral distortion to obtain a more accurate and informative measure of the performance of our DNN model.

Table 4.7: Comparative results of a 6-speaker DNN model trained by using training procedure proposed by Yuchen et al. (2015) and averaged weights approach.

	Traditional training approach		Averaged Weights Approach	
	Epochs	Validation Error (RMSE)	Epochs	Validation Error (RMSE)
English Accent				
Speaker 1	13	244.710	25	244.712
Speaker 2	13	248.532	25	246.978
Speaker 3	13	271.069	25	268.380
Scottish Accent				
Speaker 1	13	262.361	25	261.674
Speaker 2	13	251.483	25	249.639
Speaker 3	13	244.935	25	243.925

According to the results obtained in both training procedures (see section 4.2.3.1 and 4.2.3.2) we have obtained better results of acoustic modelling using the averaged weights approach when training the DNN model. Table 4.7, which compares the performance of each speaker of a 6-speaker DNN model using these approaches, shows that in most of the cases the performance of the model trained with the averaged weights approach improves over the accuracy of the output generated by the model trained with the traditional training approach used by Yuchen et al. (2015). For instance speaker 3 with English accent has an RMSE of 268.380 units using the alternative approach, whereas the same speaker modeled by the traditional approach has an accuracy of 271.069. However, is necessary to note that the averaged weights approach takes almost two times the number of epochs of the traditional approach to obtain a better RMSE performance.

Contrarily, based on the analysis made, we can determine some benefits and limitations of the performance of each of the approaches implemented. According to the performance of the traditional approach the most significant benefit of the technique implemented by Yuchen et al. (2015) is the efficiency when training the model. As already explained, this method can obtain similar RMSE considering half of the training epochs employed in the averaged weights approach. However, we identified three important limitations:

1. Similarity of training corpus of speakers. In order to share the learning of each speaker and influence significantly to the acoustic realisation of other speakers, it is necessary that the speakers have a strong similarity in the speech properties.

2. Limited influence of shared knowledge. Despite increasing the number of speakers when training the DNN model, the performance of the model is steady and poorly influenced by the shared knowledge by other speakers.

3. Saturation of shared learning. When two speakers “a” and “b” have very similar features, the features learned by speaker “a” can saturate the learning of speaker “b” and provoke that speaker “b” mimics or acquires very similar features to speaker “a”, which can generate that speaker “b” loses his particular speech characteristics.

On the other side, we have found two remarkable benefits of the averaged weights approach:

1. Flexible influence of shared learning. Differences between the speech features of

each speaker such as gender or accent does not restrict sharing learning gained by the diverse speakers and significantly impacts a better performance of the DNN model.

2. Robust acoustic modeling by intelligent shared learning. The learning obtained by each speaker influence a better generalisation of the acoustic modelling, but it restricts an over learning that could corrupt very particular properties of each speaker.

Two important limitations of the averaged weights approach are a slow convergence rate and the number of speakers that lead to a better performance of the multi-speaker modelling. The convergence rate is very slow compared to the performance of the traditional training methodology. We require almost twice the number of epochs used by the traditional approach. The number of speakers that can influence on a better generalisation of the acoustic modelling is limited. If we include several speakers to the multi-speaker model, when sharing learning features information, the model is not capable of acquiring useful information to improve the accuracy of the model.

Chapter 5

Conclusions

In this project we have investigated recent advancements in two research areas: speech synthesis and machine learning. We found that deep learning is a trend topic that has started to be implemented for solving some of the major problems of conventional TTS systems. An important goal for parametric speech synthesis research is to build TTS models that create artificial speech with the full range of naturalness and quality that characterise the voice of a healthy human when he is speaking. Therefore, we have proposed to build a robust multi-speaker model based on DNNs for TTS systems with the purpose of enhancing the quality of synthetic speech in English language.

We discover that the quality of synthesised speech is affected by a poor acoustic modelling. Therefore, we hypothesise that DNNs has powerful characteristics that can improve the acoustic realisation performed by traditional parametric models such as HMMs.

Some of the limitations that we encountered for building an acoustic model of high quality are the robustness of conventional parametric approaches such as HMMs, which provides an insufficient representation of phonetic and linguistic information when using decision trees models. Additionally, the size of the training corpus as well as meaningful data containing linguistic and phonetic features is essential for creating an accurate acoustic model.

We analysed the benefits and limitations of the multi-speaker DNN model proposed by Yuchen et al. (2015), and we included some of their major contributions for devising and building our multi-speaker DNN model. We considered dedicated layers in two levels; one for modelling abstract features such as regional accent and another particular layer at the output level for mapping very specific properties of the speech of each speaker. Moreover, we also exploited the benefits of multi-task learning to build a better acoustic coverage. However, the training method used by Yuchen et al. (2015) presented a very limited influence when sharing knowledge with speakers with different properties. Furthermore, this method presented problems of influencing saturation of shared learning to other speakers, which means that a speaker lost many of his particular characteristics and acquires the properties of the speaker that is sharing his own learning. For that reason, we have introduced the averaged weights procedure as an alternative manner to exploit the correlations and differences among the speech and linguistic features of diverse speakers when training a multi-speaker DNN-based model.

The performance of our proposed multi-speaker DNN model shows promising results to improve the quality of parametric speech synthesis models, either by training the model with the method used by Yuchen et al. (2015) as well as when trained with the averaged weights procedure we have proposed in this project. In comparison to our best reference, which is a speaker dependent model, the results obtained frequently proves a better performance when including up to 3 speakers per accent to train the multi-speaker model. For this DNN architecture, modelling English accent speakers corpus was easier and faster than modelling Scottish accent speakers.

The averaged weights procedure demonstrates two notable benefits in comparison to the training method of Yuchen et al. (2015). This approach provides flexible influence of shared learning, which indicates that the learning of a speaker can be shared and learned by another speaker independently to the particular properties of the speaker such as his regional accent. Most importantly, this technique leads to building a robust acoustic model since the shared learning is cleverly incorporated to generate a better generalisation of the acoustic modelling. Nevertheless, the main limitation of this approach is the number of speakers that can be included in the training procedure of our model since by the time five speakers per accent are included to train the model, the architecture cannot represent or learn this information to build a better generalisation of the acoustic space.

. When the number of speakers per accent is big (e.g. 5 speakers) the shared learning of each speaker cannot influence nor learn representative significantly to a better generalisation of the multi-speaker acoustic space.

There are two possible manners of evaluating the performance of a TTS system. The first manner is aimed at testing the overall performance of the TTS system. The second manner is focused at assessing the performance of a particular unit of the TTS system. We discovered that an extensive analysis of the units and tasks we are interested in evaluate is required. Thus, determining the optimal measure metrics gives very informative results when performing the experiments and, consequently, we can generate factual conclusions about the performance of our unit of interest.

5.1 Future Work

Due to the model we are proposing is one of the pioneering multi-speaker architectures designed for TTS systems of English language, there are several opportunities to develop further research advancements that could lead to building a multi-speaker model of higher-quality for enhancing the naturalness and similarity of synthetic speech to human speech.

One important extension for future work of multi-speaker DNN models is the evaluation of the model considering different conditions to those presented in this work. For instance, we performed experiments considering the same number of speakers for each accent. Then, an important question to be answered can be How the influence of diverse speakers (with similar properties and easily understood by our model) can improve or affect the overall performance of the multi-speaker model? Following this idea, another alternative proposal to enhance the robustness of our multi-speaker DNN model is How would our model behave if we included a penalisation term that determines the level of sharing learning for other speakers?

Similarly, as the training corpus considered might not represent an extensive training corpus to exploit the benefits of DNNs, we can evaluate the performance of our model considering a bigger training corpora in which the conditions of the data set are the optimal to maximise the performance of our DNN model.

Another possibility to be explored is the evaluation stage of our model. We have used the simplest technique to measure the accuracy of our architecture, however, as the target of speech synthesis research is aimed at providing synthetic speech with similar properties to those produced by a human, we can use the output feature vector of our DNN model to generate a speech waveform and perform objective and subjective testing methods such as mel-cepstral distortion or listening tests evaluated by diverse people.

Bibliography

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., and Bengio, Y. (2012a). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y. (2012b). Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828.

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Collobert, R., Kavukcuoglu, K., and Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376.

Damper, R. (2001). *Data-driven techniques in speech synthesis*. MIT Press.

Deng, L. and Jaitly, N. (2015). Deep discriminative and generative models for pattern recognition.

Deng, L. and Li, X. (2013). Machine learning paradigms for speech recognition: An overview. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(5):1060–1089.

Deng, L. and Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387.

Erhan, D., Manzagol, P., Bengio, Y., Bengio, S., and Vincent, P. (2009). The difficulty of training deep architectures and the effect of unsupervised pre-training. In *International Conference on artificial intelligence and statistics*, pages 153–160.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Kotsiantis, S., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009). Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, 10:1–40.

Lee, K. (1990). Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(4):599–609.

Murphy, K. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sukittanon, S., Surendran, A., Platt, J., and Burges, C. (2004). Convolutional networks for speech detection. In *Interspeech*. Citeseer.

Taylor, P. (2009). *Text-to-Speech Synthesis*. Cambridge University Press.

Veaux, C., Yamagishi, J., and King, S. (2013). The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In *Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), 2013 International Conference*, pages 1–4. IEEE.

Wan, V., Latorre, J., Chin, K., et al. (2012). Combining multiple high quality corpora for improving hmm-tts. In *INTERSPEECH*.

Wu, Z. (2015). A study of speaker adaptation for dnn-based speech synthesis. *Interspeech 2015*.

Yuchen, F., Yao, Q., Soong, F., and Lei, H. (2015). Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis. In *40th International Conference on Acoustics, Speech and Signal Processing, 2015*. ICASSP.

Zen, H., Senior, A., and Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7962–7966. IEEE.

Zen H., and Tokuda, K. and Black, A. (2009). Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064.