UNIVERSITY OF LORRAINE

UNIVERSITY OF THE BASQUE COUNTRY

MASTER THESIS

Expressive speech synthesis based on transfer of prosodic information

Author:
Ajinkya KULKARNI

Supervisors:
Denis JOUVET
Vincent COLOTTE
Mathieu CONSTANT

Thesis submitted in fulfillment of the requirements for the degree of

Master of Science in Language and Communication Technologies

Based on work done during an internship with the

Multispeech Team, Loria, Inria
University of Lorraine

October 12, 2018

Declaration of Authorship

I, Ajinkya KULKARNI, declare that this thesis entitled "Expressive speech synthesis based on transfer of prosodic information" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment."

Carl Friedrich Gauss

UNIVERSITY OF LORRAINE

Abstract

Multispeech Team, Loria, Inria
University of Lorraine

Expressive speech synthesis based on transfer of prosodic information

by Ajinkya KULKARNI

Over the last decade, text-to-speech synthesis systems (TTS) were able to achieve good quality and intelligibility. Nowadays, TTS systems are widely used in smart home systems, dialog systems, and navigation systems. Currently, interactions with Text to speech systems in human machine interface is still constrained with more or less natural voice without consideration of the relationship between textual semantics and emotions in speech. Even though current state-of-the-art speech synthesis systems are able to achieve high intelligibility with naturalness in speech, machines are still constrained by neutral voice. Interacting with this generation of speech for a long duration, makes it monotonous and less interactive with the user. Furthermore, synthesized speech for long sentences results in a monotonous and dissociated user experience. It seems likely that expressive speech synthesis provides the ability to greatly improve the user experience with machines. Indeed current speech style is typically a "reading style", which results from the style of the speech data used to develop TTS systems. To create expressive speech corpus is a complex process in terms of accuracy of expressions, time involved in development and resources invested. In order to reuse the existing data on different users, we need expressivity adaptation techniques. With respect to this, we present an expressive speech synthesis system for a new speaker without requiring explicit recording of the expressive speech data. The expressive speech synthesis involves two basic tasks, identifying the expressive information from text and synthesizing speech for a specific emotion. The presented thesis work is focused only on synthesizing speech for a specific emotion.

Recently, deep learning architectures were used as a substitute for statistical parametric speech synthesis system. This thesis presents three main contributions, firstly, we build a baseline expressive speech synthesis system for the French language using deep learning framework. Using the deep learning framework, we modeled the duration and acoustic parameters of baseline expressive speech synthesis system, considering each emotion independently. Recent studies showed that features represented through hidden layers transition from general to task-specific along the network (Long et al., 2015). Therefore, we proposed domain adaptation technique, in which different layers were adapted with speech parameters from the different speakers, for transferring the expressivity to a new speaker for which only neutral voice data is available. Thirdly, we present variational autoencoders to model acoustic parameters conditioned on a text. This allows us to learn the latent representation of speech of different speaker independently, thus enabling the control of expressivity through interpolation of latent representation from a neutral and expressive speech in an unsupervised way.

Acknowledgements

First and foremost, I would like to thank my supervisors Denis Jouvet and Vincent Colette for sharing their knowledge, experience, and wisdom. I would like to express my gratitude to Denis Jouvet for his continual support and guidance throughout the thesis work, he enriched my research experience. I wish to thank my colleagues in Multispeech team at Loria, particularly Nicolas Turpault and Sara Dahmani for their participation in discussions. Furthermore, I wish to thank Chedy Raïssi for his valuable remarks on the thesis work. I would like to express my gratitude to my local coordinators Maxime Amblard and Miguel Couceiro for their support and guidance throughout the academic year.

Finally, I would like to thank my family and friends for their love and encouragement, without which this thesis would not have been possible.. . .

Contents

List of Figures

List of Tables

List of Abbreviations

TTS	**T**ext **T**o **S**peech
HMM	**H**idden **M**arkov **M**odel
VAE	**V**ariational **A**uto**e**ncoder
DNN	**D**eep **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
LSTM	**L**ong **S**hort **T**erm **M**emory
ELU	**E**xponential **L**inear **U**nit
Voder	**V**oice **O**perating **D**emonstrator
MOS	**M**ean **O**pinion **S**core
MUSHRA	**MU**ltiple **S**timuli **H**idden **R**eference **A**nchor
KL	**K**ullback **L**eibler

Dedicated to my family…

Chapter 1

Introduction

With recent advents in computational power, parametric speech synthesis systems were able to generate highly intelligible and natural voice. Every sentence spoke by a human inherently possesses expressiveness, which is conditioned on the pragmatics behind the content of the speech. Let's consider a scenario where we use state of the art speech synthesis system for vending machines, it might yield a reasonable user experience. However, the same speech synthesis system for reading audio books or interacting with smart home systems will make communication monotonous and unwieldy. Therefore, only sounding natural and intelligible is not sufficient.

The term expressivity in speech usually refers to the characteristics of speech which ranges from emotions, speaking style, the relationship of speech with gestures and facial expression. Throughout the thesis, we considered only emotional characteristics of expressivity in speech. The present state of the art speech synthesis system now uses deep neural architectures, which heavily depend on the speech corpus used to train the deep neural networks. Therefore, to build expressive speech synthesis for a new speaker, we have to create a speech corpus with various emotions. It is inconvenient to record an expressive speech corpus every time we want to build an expressive speech synthesis system for the new speaker's voice. Furthermore, creating an expressive speech corpus is laborious and expensive in terms of workload in speech acquisition, speech recording, labeling, alignment and evaluation of expressive speech corpus. On top of it, the speaker who is responsible for providing expressive speech has to enact the emotions while recording. This makes the speech acquisition process even more tedious. Besides that, many approaches have been proposed stating the use of audio-books, films, dialogues, etc. to create expressive speech synthesis (Oliva and Steiner, 2013), labeling the expressions is not a trivial task due to a large number of possible variations in single emotion. This creates a bottleneck in the development of expressive speech synthesis.

Recently, a significant amount of work has been done on voice conversion and on speaker adaptation with a small amount of available speaker speech data. We proposed deep learning architecture aiming to transfer the expressivity from a given expressive speech corpus to a different speaker's neutral speech synthesis system. This task requires the disentanglement of speakers style, global representation of speech and transfer of expressivity in the speech. In this thesis work, we investigated deep learning frameworks for disentangling and transferring the expressivity.

1.1 Thesis goals

The main objective of the thesis work is to develop expressive speech synthesis system for a given speaker voice with a transfer of expressivity from an existing expressive speech corpus of a different speaker, without the acquisition of expressive speech corpus for given speaker. The focus of the work is on investigating transfer

learning mechanism, which will accelerate the efforts towards exploiting existing expressive speech corpus. The major contributions of the thesis work are as given below,

- Deep neural network based baseline system for speech synthesis for expressions anger, disagree, fear, joy, sadness and surprise.
- Layer adaptation technique to transfer the expressivity through co-training with neutral and expressive speech
- Conditional variational autoencoders to learn the latent representation of phonetic content and speaker expressions
- Interpolating latent representation of speech to enable the control of expressivity in speech synthesis

1.2 Thesis overview

The structure of the thesis work is organized as follows. Chapter 2 begins with the existing state of the art systems for speech synthesis as well as for transfer of expressivity. Chapter 3 gives an overview of methods used for the preparation of training data for deep neural networks, covering preparation of context labels (linguistic input textual features), acoustic features. Furthermore, this chapter also introduces basic terminologies with respect to speech synthesis, which will be used throughout the thesis. In Chapter 4, we describe the baseline deep neural network model and layer adaptation technique for transfer of expressivity. Afterward, we will investigate the transfer of expressivity through representation learning with variational autoencoder neural framework in Chapter 5. In Chapter 6. we will discuss the experimentation conducted with developed baseline speech synthesis systems as well as approaches used for transferring the expressive information. Finally, in Chapter 7, the conclusion will be drawn regarding studies carried out in the present thesis work, the difficulties and the challenges of the task, as well as possible future work regarding the panorama of newly emerged synthesis techniques also based on neural networks.

Chapter 2

Speech synthesis review

In the history of mankind, speech and language co-evolved as a primary means of communication. The first attempt towards speech synthesis system dates back to 1791 when German-Danish scientist Christian Gottlieb Kratzenstein built human vocal tract model for the production of vowel sounds. In 1939 Homer Dudley introduced speech synthesizer Voder (Voice Operating Demonstrator) as the first electrical device for speech synthesis. Voder was inspired by Vocoder (Voice Coder) developed at Bell Laboratories in the 1930's (Lemmetty, 1999). The basic idea behind Vocoder was to process speech into time-varying acoustic parameters and synthesizer to reconstruct the original speech signal. Most of the current speech synthesis systems still use similar concepts i.e. they rely on vocoder for computing acoustic parameters and generation of the speech waveform.

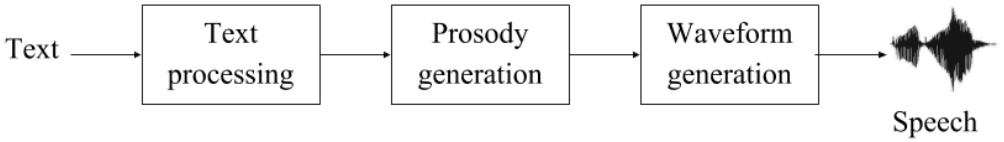

FIGURE 2.1: Framework for speech synthesis system

The general framework for speech synthesis system includes text analysis, prosody generation, and waveform generation as shown in Figure 2.1. The text analysis is an imperative step in speech synthesis which extracts linguistic, phonetic and prosodic information from the input text and creates a representation of text to predict prosody for generation of speech. The term prosody parameters are aspects of speech signal referring to the level above the individual phoneme such as intonation, rhythm, and intensity. Intonation is melody or pitch of speech and also called as the fundamental frequency. It is one of the important parameters which influences the segmental and suprasegmental level in speech such as tone and accent in the speech. Rhythm refers to the duration and frequency of syllable. Intensity is the loudness in a speech which is directly dependent on the amplitude levels in speech. These prosody parameters are used to generate a speech waveform with the vocoder.

In this chapter, we present the approaches adopted for speech synthesis over the last decade and the current state of the art of speech synthesis systems. Furthermore, we investigate the deep learning architectures used for transfer of expressivity in speech synthesis context.

2.1 Brief history of expressive speech synthesis

The work on expressive speech synthesis goes back to the beginning of the 1990s. The following section provides a concise summary of techniques used for expressive speech synthesis namely formant synthesis, unit selection, Hidden Markov Model (HMM) based expressive speech synthesis and neural network based expressive speech synthesis.

2.1.1 Formant synthesis

The term formant is attributed to distinctive (high energy) frequency components in speech, that corresponds to the acoustic resonance of the human vocal tract. Formant-based speech synthesis method is one of the oldest methods for efficient realization of the physical and spectral model as a source-filter model. In this approach, source models the glottal pulse which is then passed through filters to imitate the human vocal tract (Schröder, 2009). The synthetic speech is produced with a set of rules to define prosody parameters such as fundamental frequency, intensity, rhythm over time. Due to limitations of the rule-based system, formant synthesis is not able to generate natural voice. Even if formant synthesis produces a robotic voice, it can be used in creating explicit models for each emotion. Like DECTalk (Cahn, 1990) and HAMLET (Murray and L. Arnott, 1995) speech synthesis systems used emotion-specific modules to build expressive speech synthesis. The formant synthesis doesn't require pre-recorded speech for synthesis and doesn't rely on a large database of speech units. Thus, it has the potential to be used in the embedded systems where memory requirement is a crucial factor.

2.1.2 Unit selection

At the beginning of the 1990s, the first data-driven approach was used with a concatenation of diphones (a pair of phonemes) to produce a speech waveform for a given language (Schröder, 2009). This approach is enhanced furthermore with using various samples of diphone units recorded in natural speech. Unit selection method generates synthesized speech using selection and concatenation of units (diphones, triphones) from speech database recorded with variation in units with respect to prosody features and duration. In unit selection synthesis, for a given text diphone units are selected based on optimal path algorithms such as the Viterbi algorithm. In this method, an appropriate (optimal) list of units is selected using both concatenation scores (how well two adjacent units concatenate) and target scores (how well units matches target criteria), therefore this method is called the unit selection. As the unit selection method primarily depends on recorded speech, for expressive speech synthesis, diphone units must be recorded for each emotion (Fernandez and Ramabhadran, 2007). A given emotion is then generated by selecting units only from the corresponding subset of the recordings.

2.1.3 Statistical parametric speech synthesis

The research in unit selection speech synthesis was carried forward with the Hidden Markov Model (HMM) approach, which predicts acoustic and duration parameters from statistical models extracted from recorded speech. Application of Hidden Markov Model is not new in speech processing, as it was widely used not only in speech recognition systems but also to build the speech synthesis systems, which

made HMM most suitable model in speech processing domain (Rabiner, 1990). In this approach, the densities of context-dependent models are organized in a decision tree; at run-time, for a given "target" context to be realized, the tree yields the appropriate HMM state sequence corresponding to that context, describing mean and standard deviation of the acoustic features (King, 2010; Zen, Tokuda, and Black, 2009). Note that, whereas in speech recognition, the context is mainly restricted to the identity of previous and following phones, in speech synthesis the context includes much more linguistic, phonetic and prosodic information. This approach is called statistical as it represents the parameters using means and variances of probability density function that are estimated from training data. Furthermore, the approach is parametric as the speech signal is described by a set of parameters. Similar to unit selection based on expressive speech synthesis, the simple solution is to train the HMM model on a speech corpus with various emotions and speaking style. The further strength of HMM-based synthesis is that speaker-specific or style-specific voices can also be created by adaptation rather than training (Schröder, 2009).

2.1.4 Neural speech synthesis

In early 1990's, neural network based models were used for speech synthesis (Weijters and Thole, 1993), but due to lack of computational resources as well as large training data and efficient training algorithms, this approach was not able to catch attention at that time (Cawley and Noakes, 1993; Tuerk and Robinson, 1993). As stated earlier, HMM-based statistical parametric speech synthesis use decision tree clustered with context-dependent linguistic information to generate probability densities of speech parameters for a given text (Zen, Senior, and Schuster, 2013), (Wu, Watts, and King, 2016). In neural speech synthesis, HMM and their associated decision trees are replaced with a feedforward neural network, which are used to predict the speech parameters directly from the set of contextual labels, then the speech waveform is constructed from these predicted speech parameters. The neural network approach enhances the ability of the model to map the complex non-linear relationship between context-dependent decision tree to vocoder parameters as a multi-dimensional non-linear regression problem.

To produce synthetic speech with neural networks, the text is first converted into phonemes and subsequently, these phonemes are mapped to input contextual features (conveying phonetic, linguistic and prosodic information) denoted as X_t. The input contextual features are transformed into a set of answers about linguistic information such as current phoneme is a vowel or not, position information about current phoneme in a sentence, etc. Subsequently, neural network predicts the speech parameters, Y_t for a given input contextual features for each frame as shown in Figure 2.2. Finally, with the predicted speech parameters, the speech waveform is synthesized using a vocoder.

To better handle the dependencies between speech frames, recurrent neural network (RNN) based architectures such as long short-term memory (LSTM) or gated recurrent units (GRU), were employed as a sequence to sequence learning problem (Fan et al., 2014). (sequence of input features to sequence of vocoder parameters for each frame). For creating baseline speech synthesis for each emotion, we adapted deep neural network based framework, which we will discuss in more details in Chapter 4.

With recent advances in deep neural network based architectures, the various end to end speech synthesis systems have recently been proposed, namely Tacotron (Wang et al., 2017), Char2wav ("Char2wav: End-to-end Speech Synthesis"), Wavenet

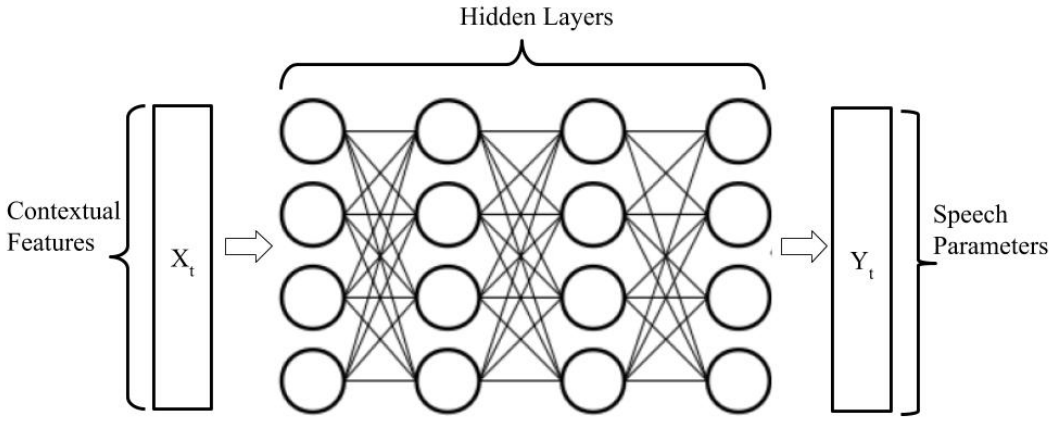

FIGURE 2.2: Deep neural network architecture for speech synthesis.

(Oord et al., 2016), voice loop (Taigman et al., 2017). The Tacotron speech synthesis system adopts a multi-stage encoder-decoder architecture with multiple RNNs and blocks called CBHG (convolutional block highway gated linear units), where each CBHG contain multiple convolutional layers, a highway network, and a bidirectional GRU. The speech waveform is generated by Griffin-Lim method from output synthesized spectrogram, where acoustic features are directly conditioned on input text instead of linguistic features, making Tacotron complete end to end system without explicit knowledge of a language. WaveNet architecture is based on stacks of dilated convolutions, which are termed "casual" for not looking into the future (Oord et al., 2016). The output audio is generated sample by sample, which, at typical sampling rates of thousands of hertz, is too slow for current TTS applications. The Char2Wav architecture uses RNNs for both the reader and the generator and an attention mechanism, the Graves positional attention mechanism is employed ("Char2wav: End-to-end Speech Synthesis")

2.2 Transfer of expressivity to speech synthesis system

For transfer of expressivity, an extension to Tacotron was recently proposed which learns a latent embedding of prosody, derived from a reference acoustic representation containing the desired prosody (Skerry-Ryan et al., 2018). This embedding extracts a representation of speech, which exhibits distinct features of speaker and expressiveness. Even after extracting the latent embedding of prosody from a reference signal, Tacotron faces issues with disentangling the textual information from prosodic information implicit in the reference signal. Tacotron systems require a considerable amount of speech data (100 hrs) with high-end graphics processing units to train the system, which limits the Tacotron to build on small speech corpus.

Several studies have shown variational autoencoder's success in disentangling the latent representation of natural images. Deep probabilistic generative models such as variational autoencoders (VAEs) have been proposed for representation learning of speech in a latent space for voice conversion systems (Blaauw and Bonada, 2016; Oord, Vinyals, and Kavukcuoglu, 2017; Hsu et al., 2016; Hsu et al., 2017). In such approaches, encoders are trained with speech parameters and speaker labels and decoder generates speech parameters conditioned on the speaker label (Hsu et al., 2016). Recently, (Akuzawa, Iwasawa, and Matsuo, 2018) combined the

voiceLoop, an autoregressive speech synthesis model with Variational autoencoder conditioned on a text. In this approach, VAE models the high-level characteristics of speaker and emotions in latent space to learn and control the speech expressions. In this thesis work, we adopted a similar method for the expressivity control with deep neural network based VAE model conditioned on linguistic features generated from text. Latent generative models can learn higher-level underlying factors from complex data in an unsupervised manner, we will discuss this in depth in Chapter 5.

Chapter 3

Data preparation

To develop speech synthesis in a parametric framework, we need to represent text and speech in form of features. In this chapter, we will discuss the speech corpus and speech data preprocess for the training of the text to speech synthesis models and for the synthesis of text to speech synthesis system. We will discuss the speech corpus and speech data preprocessed for training and synthesis of test examples. Basically, we preprocess the raw textual and speech data to extract the input contextual features and output speech parameters. First, we will describe the speech corpus. Then, we will present the front end for text analysis to produce features. In this thesis, we used the parametrization of speech approach for synthesis. For this, we will explain how can we represent the speech signal with vocoder parameters. The chapter is organized in the following manner, Section 3.1 describes the details about two French speech corpora used for creating baseline expressive speech synthesis as well as for transferring the expressivity. Afterward, section 3.2 introduces how raw text is mapped into context label features. Finally, Section 3.3 provides details regarding speech signal parameterization using a vocoding technique.

3.1 Speech corpus

To work on transfer of expressivity and baseline expressive speech synthesis, we worked with two speech corpora, namely Lorene neutral speech corpus and Christine expressive speech corpus in the French language both recorded with a female voice (each corpus correspond to the different female speaker). The term expressivity in speech usually refers to the characteristics of speech which ranges from emotions, speaking style, the relationship of speech with gestures and facial expression. Throughout the thesis, we considered only emotional characteristics of expressivity in speech. Lorene corpus is a neutral speech corpus with 1815 utterances approximately 3hr of recording. Christine expressive corpus consists of 6 emotions, namely joy, surprise, fear, anger, sadness, and disagree with 500 utterances for each expression which is approximate 1hr of recording and a neutral emotion with 2000 utterances which are approximate 3hr of recording. For recording expressive corpus, the actress has used a technique called an exercise in style, where she dissociates the semantics of syntax the sentences and acts same sentences in different emotions (Ouni et al., 2016). In this context, the emotions should be considered as acted ones as they are a bit exaggerated as in the case of a play at the theater.

In this thesis work, we used Christine corpus for building baseline expressive speech synthesis for all emotions. Furthermore, Christine corpus is used for extracting the expressivity to transfer it to the Lorene neutral speech model. All the speech signals were used at a sampling rate of 16 kHz. Each speech corpus is divided into train, test, and validation sets in the ratio of 80, 10, 10 respectively.

3.2 Text analysis

As stated earlier, in a parametric speech synthesis approach, the first step consists in converting raw text into a representation that will convey linguistic, phonetic, and prosodic information, resulting in a sequence of contextual labels (also called contextual features); each item of the sequence corresponds to a phone or a silence segment. Contextual labels for a given speech segment (whether a phone or silence segment) corresponds to a complex tree-structured representation which is flattened at phoneme, syllable, word level with prosodic and phonetic information. This information is further enhanced with context labels which process the information about adjacent phonemes, positions of adjacent and current phonemes, stress and accent information about these phonemes, part of speech for a word to which phoneme belongs, etc. Table 3.1 shows the list of contextual labels and their description.

TABLE 3.1: List of context labels.

Symbol	Description
p1	phoneme identity before the previous phoneme
p2	previous phoneme identity
p3	current phoneme identity
p4	next phoneme identity
p5	the phoneme after the next phoneme identity
p6	position of the current phoneme identity in the current syllable (forward)
p7	position of the current phoneme identity in the current syllable (backward)
a1	whether the previous syllable is stressed or not (0; not, 1: yes)
a2	whether the previous syllable is accented or not (0; not, 1: yes)
a3	number of phonemes in the previous syllable
b1	number of phonemes in the previous syllable
b2	whether the current syllable accented or not (0: not, 1: yes)
b3	the number of phonemes in the current syllable
b4	position of the current syllable in the current word (forward)
b5	position of the current syllable in the current word (backward)
b6	position of the current syllable in the current phrase(forward)
b7	position of the current syllable in the current phrase(backward)
b8	number of stressed syllables before the current syllable in the current phrase
b9	number of stressed syllables after the current syllable in the current phrase
b10	number of accented syllables before the current syllable in the current phrase
b11	number of accented syllables after the current syllable in the current phrase
b12	number of syllables from the previous stressed syllable to the current syllable
b13	number of syllables from the current syllable to the next stressed syllable
b14	number of syllables from the previous accented syllable to the current syllable
b15	number of syllables from the current syllable to the next accented syllable
b16	name of the vowel of the current syllable
c1	whether the next syllable stressed or not (0: not, 1:yes)
c2	whether the next syllable accented or not (0: not, 1:yes)
c3	the number of phonemes in the next syllable
d1	gpos (guess part-of-speech) of the previous word
d2	number of syllables in the previous word
e1	gpos (guess part-of-speech) of the current word
e2	number of syllables in the current word

Continued on next page

Table 3.1 – continued from previous page

Symbol	Description
e3	position of current word in the current phrase (forward)
e4	position of current word in the current phrase (backward)
e5	number of content words before the current word in the current phrase
e6	number of content words after the current word in the current phrase
e7	number of words from the previous content word to the current word
e8	number of words from the current word to the next content word
f1	gpos (guess part-of-speech) of the next word
f2	number of syllables in the previous word
h1	number of syllables in the current phrase
h2	number of words in the current phrase
h3	position of the current phrase in utterance (forward)
h4	position of the current phrase in utterance (backward)
h5	Phrase modality (question, exclamation, etc.)
i1	number of syllables in the next phrase
i2	number of words in the previous phrase
j1	number of syllables in this utterance
j2	number of words in this utterance
j3	number of phrases in this utterance

p1^p2- p3+p4=p5 @p6 p7/A:a1 a2 a3 /B:b1-b2-b3 @b4-b5 &b6-b7 #b8-b9 $b10-b11 !b12-b13 ;b14-b15 |b16 //C:c1+c2+c3/D:d1 d2 /E:e1+e2 @e3+e4 &e5+e6 #e7+e8 /F: f1 f2/G:g1 g2 /H:h1=h2 @h3=h4|h5 /I:i1 i2/J: j1+ j2- j3

This format is commonly used in HMM-based speech synthesis tool (HTS) and is also know as HTS format labels (Zen, 2006). This format is widely used in parametric speech synthesis systems. We used Soja tool developed by Multispeech team at LORIA lab, Nancy, as a front end for label generation. Furthermore, after generation of contextual labels, the corresponding input vector is created with the aid of the question file. The question file is a simple set of regular expression question which has binary values (1 when it regular expression matches the formatted label, 0 otherwise). These questions are prepared considering the characteristics of the language. Examples of questions (which are then formalized as regular expressions) are: is the current phoneme a vowel or not? is the current phoneme a nasal or not? is the current phoneme is a fricative or not? is the current phoneme stressed or not? etc. The question file for the French language is designed with 1356 questions which consider the linguistic, phonetic and prosodic details about phonemes such as phonetic category phoneme belong, positional information about phonemes, stress and accent information, guessed part of speech (GPOS) corresponding to the part of speech annotation of words in the text. The distribution of questions is as shown in Table 3.2, 76 questions concern the left-left phoneme (i.e. two phonemes preceding the current one), 76 questions concern the left phoneme (i.e. phoneme preceding the current one), and so on.

TABLE 3.2: Distribution of questions per entity in labels.

Entity	Number of questions
Left Left (LL) phoneme	76
Left (L) phoneme	76
Continued on next page	

Table 3.2 – continued from previous page

Symbol	Description
Current (C) phoneme	76
Right (R) phoneme	76
Right Right (RR) Phoneme	76
Syllables information	115
Phrasal information	110
Word, phrases, syllable counts	545
Stress information	98
Accent information	98
Positional information about syllable, word, phrases	167
Syllable information	110
GPOS L, C, R information	33

After generating contextual labels for the text, speech waveforms and contextual labels are aligned at the phoneme level with the Hidden Markov Model based forced alignment tool by HTK toolkit (Young and Young, 1994). This forced-alignment relies on Mel frequency cepstral coefficient (mfcc) acoustic features, and provides the beginning and ending time of each phone segment. The resulting timing information will later be used to train the duration model, and; the duration model will be used to predict the duration of each phoneme during the synthesis step. As a common practice in deep learning, we normalized the contextual feature vector created from the 1356 questions for smooth convergence of neural network parameters. The features were normalized using min-max normalization to range of [0.01,0.99], as shown in Equation 3.1,

$$x_{normalized} = (0.99 - 0.01)\frac{x - x_{min}}{x_{max} - x_{min}} + 0.01 \tag{3.1}$$

3.3 Acoustic parameters

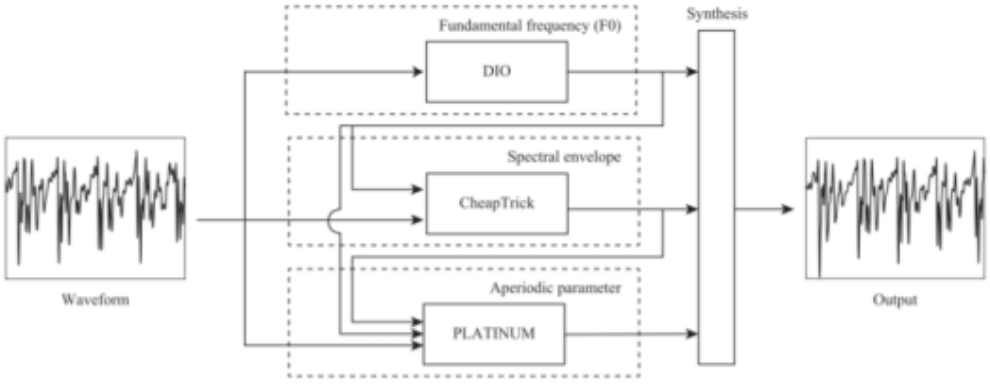

FIGURE 3.1: WORLD consists of three analysis algorithms for determining the F0, spectral envelope, and aperiodic parameters and a synthesis algorithm (Morise, Yolomori, and Ozawa, 2016).

For parameterization of the speech signal, we used WORLD vocoder which synthesizes high-quality speech as well as real-time processing. In 2009, Masanori Morise developed WORLD vocoder which is open source and freely available (Morise, Yolomori, and Ozawa, 2016). It consists of three speech analyses, namely spectral envelope, band aperiodicity, and fundamental frequency (F0) as depicted in Figure 3.1. WORLD vocoder estimate F0 contour with DIO F0 estimation algorithm (Morise, Kawahara, and Katayose, 2009). The spectral envelope is estimated using the CheapTrick algorithm which uses speech waveform and F0 information [(Morise, 2015)]. Afterward, aperiodicity in the waveform is estimated with the PLATINUM method, Aperiodic Parameter Extraction Algorithm [(Morise, 2012). We used WORLD vocoder throughout the thesis work to compute 187 acoustic parameters for every 5ms time frame, namely 180 spectral features, 60 Mel generalized cepstrum (mgc) with first and second derivatives , 3 frequency related parameters i.e. 1 log fundamental frequency (lf0) with first and second derivative and 3 excitation parameters i.e. 1 band-aperiodicity (bap) with first and second derivative and 1 value for voiced-unvoiced information (vuv). Based on the mean value (μ) and standard deviation values (σ), the acoustic features extracted from the WORLD vocoder were z-normalized as shown in Equation 3.2,

$$y_{normalized} = \frac{y - \mu}{\sigma} \tag{3.2}$$

Chapter 4

Baseline speech synthesis and layer adaptation

Deep learning is a subfield of machine learning that gained popularity in the last decade which exploits the large models with more computational resources. In this chapter, we discussed the development of baseline expressive speech synthesis with neural networks. As the focus of the thesis is on transferring the prosody information. We built several speech synthesis systems to model each expression explicitly. Furthermore, in this chapter, we will discuss the proposed transfer learning approach using layer adaptation method, which is similar to domain adaptation with pretrained models. This chapter is organized as follows. Section 4.1 introduces deep learning methods as a pretext to build prosody models with deep learning. Then, we discuss the two-stage baseline model for generating acoustic parameters conditioned on text in Section 4.2. Finally, Section 4.3 describes the proposed layer adaptation approach to transfer the expressive information from Christine expressive speech corpus to Lorene neutral speech model.

4.1 Introduction to deep learning

The first steps towards building simple neural networks began in the late 80s, which was inspired from the biological working of the neurons in the brain. Subsequently, neural networks have been applied to many fields and shown state of the art results. Since then, improvements in results were followed up with more computational power as well as with more training data used. We give here a brief introduction to neural networks.

4.1.1 Feedforward neural network

Feedforward neural networks are also called as multilayer perceptron. In which input information x flows in a forward direction across the layers of the network. The input x is transformed into desired output y through a series of cascaded transformation functions; the level of cascading defines the depth of the neural network. The basic unit in neural networks is called a perceptron or a neuron as shown in Figure 4.1.

$$h = \sum_{i=1}^{N} W_i \cdot x_i + b \tag{4.1}$$

$$y = f(h) \tag{4.2}$$

where perceptron takes input as x with N dimension and summation of weights W_N vectors with input x and bias b is added, then activation function $f(.)$ is applied to the term, which generates output y, here h is intermediate term given to activation function. The interpretation of a neuron can be visualized as a geometric hyperplane where weights act as slope and bias act as a shift from the origin.

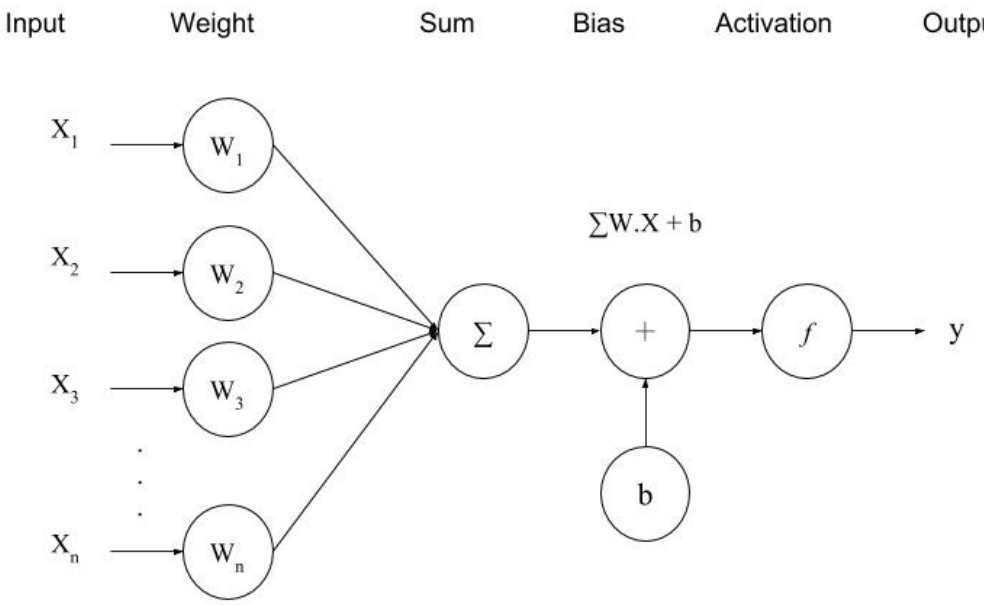

FIGURE 4.1: A single neuron perceptron network.

In practice, differentiable non-linear activation function such as sigmoidal or hyperbolic tangent function are opted to make network compatible with the backpropagation algorithm, which is used to learn the weight and bias parameters. In the backpropagation algorithm, for each example, the prediction error (cost function comparing the output predicted by the network to known output values) is estimated and backpropagated from the last layer to the first layer. Second weight and bias parameters are updated according to the gradient of error (partial derivatives of the error computed using backpropagation algorithm).

As a single neuron network is insufficient to understand the complexity of input data distribution, the model is extended by adding perceptron networks in parallel for each layer and additional layers of neurons are augmented, called hidden layers. The first layer process the N-dimensional input which is connected to hidden layers and hidden layer output is connected to the last layer which emits M outputs y, considering the dimension of output y is of size M, refer Figure 4.2. As we increase the number of hidden layers, the depth of the neural network also increases and thus called a deep neural network.

4.1.2 Recurrent neural network - Long short term memory

A recurrent neural network (RNN) is a generalization of feedforward neural networks for sequence learning, in which hidden networks and input-output networks are cyclically connected to each other, forming recurrently connected network (Graves, 2012). They are able to incorporate context information in a flexible way and are robust to localized distortions of the input data. These properties make them well

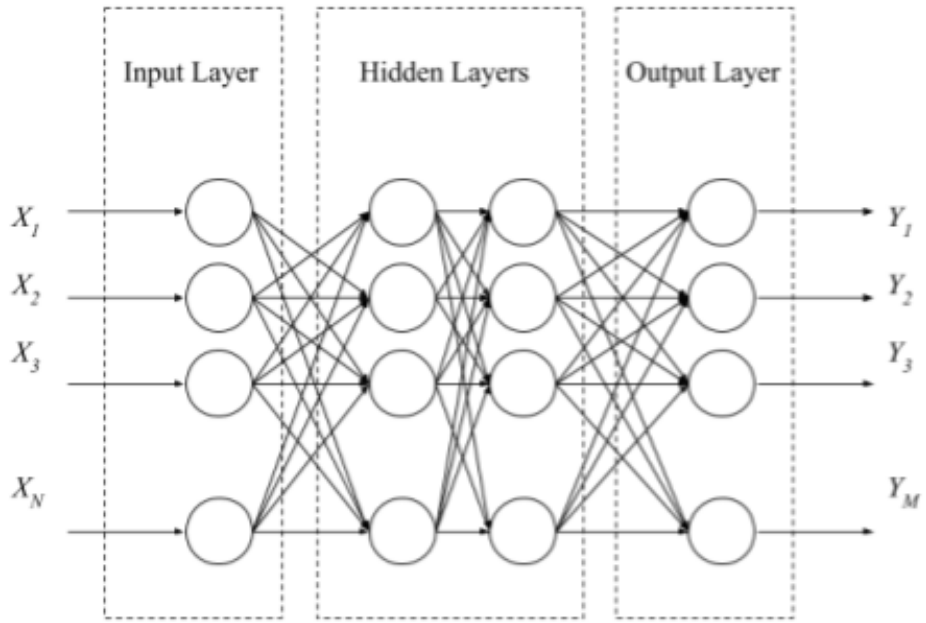

FIGURE 4.2: Multilayer perceptron neural network with input X with N-dimension size and output Y with M-dimension

suited to sequence labeling, where input sequences are transcribed with streams of labels. In backpropagation step, for long sequences, recurrent neural networks lead to vanishing gradient problem, which is overflow or underflow of error gradients. Thus, reduces the capability to handle long sequences for simple recurrent neural networks (Hochreiter and Schmidhuber, 1997).

For a recurrent neural networks with single recurrent hidden layer, the following Equations 4.3 , 4.4 are iterated over time i.e. from $t = 1$ to $t = T$ to calculate intermediate hidden representation as $h = (h_1, h_2 \ldots h_T)$,

$$h_t = f(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h) \tag{4.3}$$

$$y_t = W_{hy} \cdot h_t + b_y \tag{4.4}$$

where W_{xh} is the weight matrix between input vector, x and hidden representation in hidden layer h, W_{hh} is the weight matrix between two hidden layers and W_{hy} is the weight matrix between hidden layer and output layer. $f(.)$ is non linear activation function. b_h and b_y are bias vectors as regularizer term for hidden layer and output layer respectively.

In 1997, S Hochreiter, J Schmidhuber introduced long short-term memory (LSTM) architecture for Recurrent Neural Networks, which enhanced the power of recurrent neural network (Hochreiter and Schmidhuber, 1997). LSTM avoids the vanishing gradient problem by maintaining the internal state of recurrent cells, which replace the traditional neurons. LSTM models learn to forget the unnecessary information and keep the internal state of relevant information of features. A useful property of

the LSTM is that it learns to map an input sentence of variable length into a fixed-dimensional vector representation.

The basic idea behind the LSTM cell is to have a unit which will hold the information when its useful for a prediction of output and forgetting the same information when its no longer holds importance. In this architecture, LSTM cell includes forget gate, input gate and output gate with cell state and peephole connections to control the gating mechanism inside the cell. The hidden state representation of the LSTM cell can be described as shown in Figure 4.3

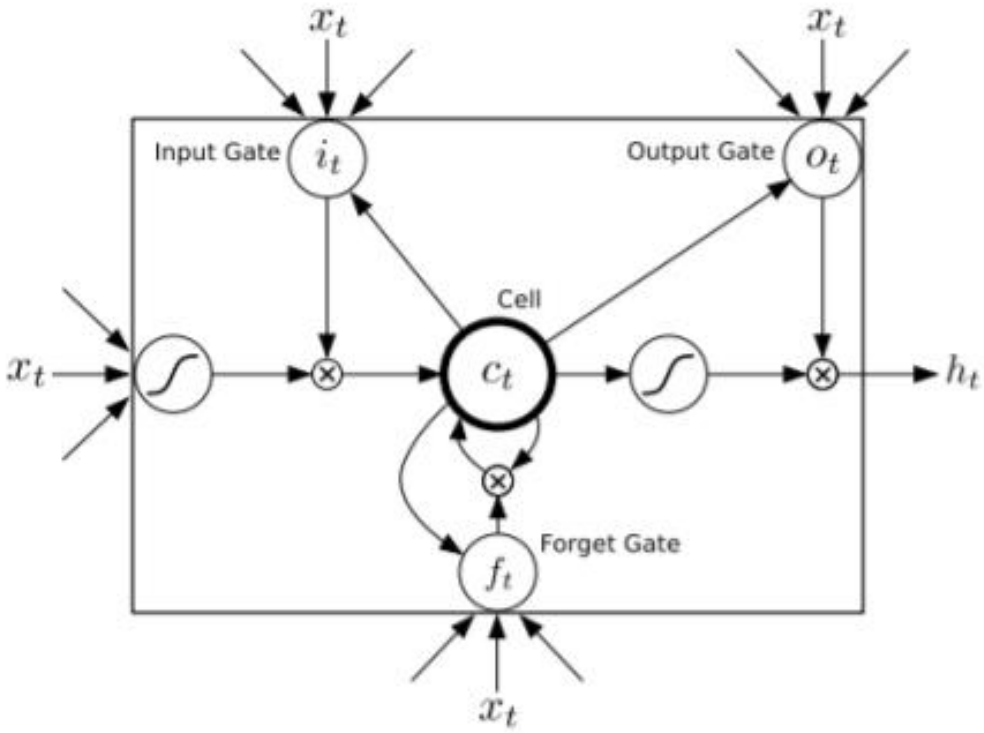

FIGURE 4.3: Long short term memory cell, (Graves, Mohamed, and Hinton, 2013).

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + W_{ci} \cdot c_{t-1} + b_i) \tag{4.5}$$

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + W_{cf} \cdot c_{t-1} + b_f) \tag{4.6}$$

$$o_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + W_{co} \cdot c_{t-1} + b_o) \tag{4.7}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c) \tag{4.8}$$

$$h_t = o_t \cdot tanh(c_t) \tag{4.9}$$

where i_t,f_t,o_t are activations of the input gate, forget gate and output gate, c_t is cell state or cell memory. σ is the sigmoid function acting as non linear function

4.1.3 Dropout

In a deep neural network increasing model, complexity leads to overfitting, which means that trained network is closely fit on training data. The overfitting is one of common the issue encountered while training the deep neural networks. Due to excessive learning of training data network parameters fails to generalize the model. In 2006 Srivastav proposed the idea of the dropout technique to resolve the overfitting problem (Srivastava et al., 2014). The key idea is to randomly drop units (along with their connections) from the neural network during training. For each training example, neuron units are dropped randomly with a probability p, building a thinned network which updates all the parameters except those associated with the dropped neurons as depicted in Figure 4.4.

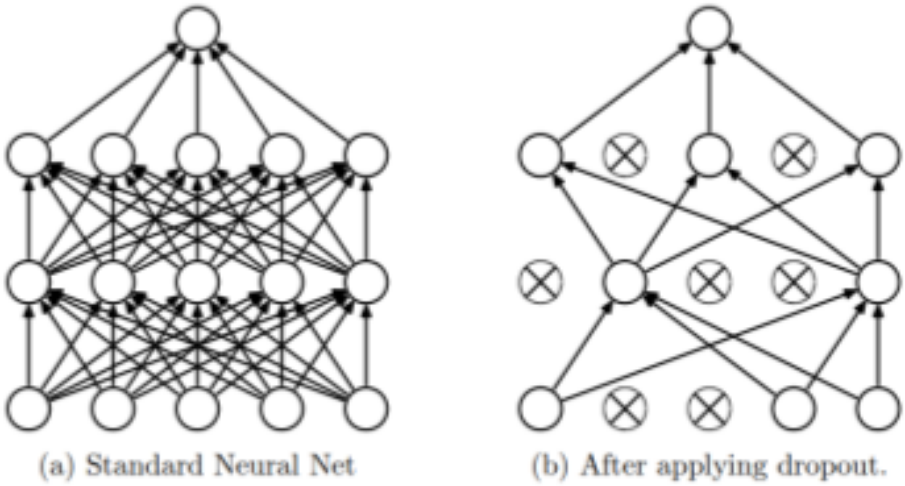

FIGURE 4.4: (a) Standard neural network, (b) example of thinned network after applying dropout, figure is based Figure 1 from (Srivastava et al., 2014).

In the testing phase, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This limits the overtuning of model parameters and avoids the co-adaptation of model parameters, which reduces the overfitting considerably. Dropout approach has shown improvement in the performance of neural networks in many supervised learning tasks including speech recognition, image recognition, and natural language processing. After each epoch, we infer the evolution of error over the network on the validation examples and the evolution of error decide to stop the training process or not.

4.2 Baseline speech synthesis

As explained in Chapter 2, In HMM-based statistical parametric speech synthesis, decision trees map linguistic context labels extracted from a front end to probability densities of acoustic parameters. These densities are then used to predict the sequence of speech parameters which are fed to the vocoder to generate synthesized speech waveform. In DNN based approach, the decision trees are replaced

by deep neural networks to enhance the model's ability to learn complexity in mapping (Zen, Senior, and Schuster, 2013). The number of context labels depends on the length of the sentence and is always less than the number of associated output acoustic frames. Therefore, we need an explicit duration model to predict the phoneme duration so that we can estimate an exact number of acoustic frames required for the speech waveform generation. Therefore, in the deep neural network, speech synthesis modeling is divided into duration model and the acoustic model.

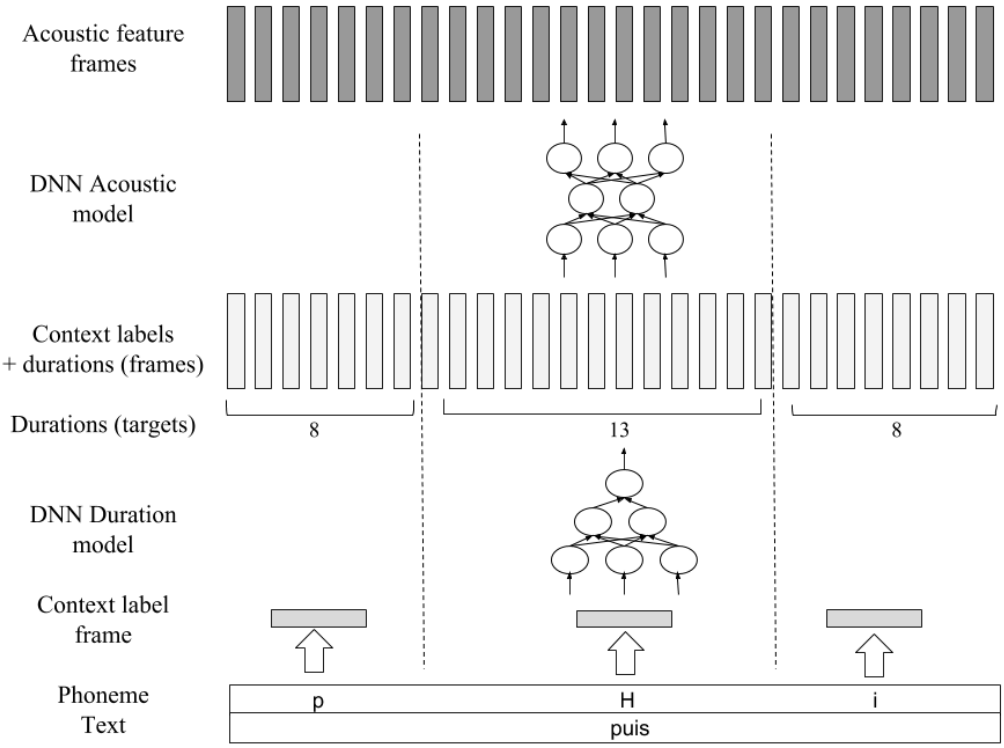

FIGURE 4.5: DNN based speech synthesis system with the duration model and the acoustic model.

In the training phase, the text is converted into a sequence of context labels by the front end. Each context label contains information about phoneme durations to be predicted and context label features as explained in Chapter 3. In the duration model, input contextual features for each phoneme is given to the neural network to predict duration output information. Thus, with the duration model, we estimate the number of acoustic frames required for each phoneme as a regression problem. Furthermore, contextual label information along with duration information is given to an acoustic model defined by deep neural network architecture to generate output acoustic parameters namely spectral parameters, aperiodicity, log fundamental frequency and voiced-unvoiced flag.

Figure 4.5 represents the processing steps in the synthesis phase. For a given input contextual label features, durations are predicted with forward propagation through the neural network. Then the output of the duration model along with contextual features are given as input to the acoustic model to predict the acoustic parameters. Afterward, denormalization is performed on generated acoustic parameters with mean and pre-computed variances from training data. Finally, the speech waveform is generated by applying vocoder to the predicted acoustic parameters.

In this thesis, we used feedforward neural networks and recurrent neural networks for the duration and the acoustic models. In a feedforward neural network, each input context label feature frame is mapped to an output acoustic frame, that is a frame to frame mapping. As recurrent neural networks are sequence learning algorithm, we give a sequence of context label features as input frame to be mapped to a sequence of output acoustic frames. For expressive speech synthesis, we modeled each emotion individually to create a dedicated speech synthesis system for each emotion.

4.3 Layer adaptation for transfer of prosody information

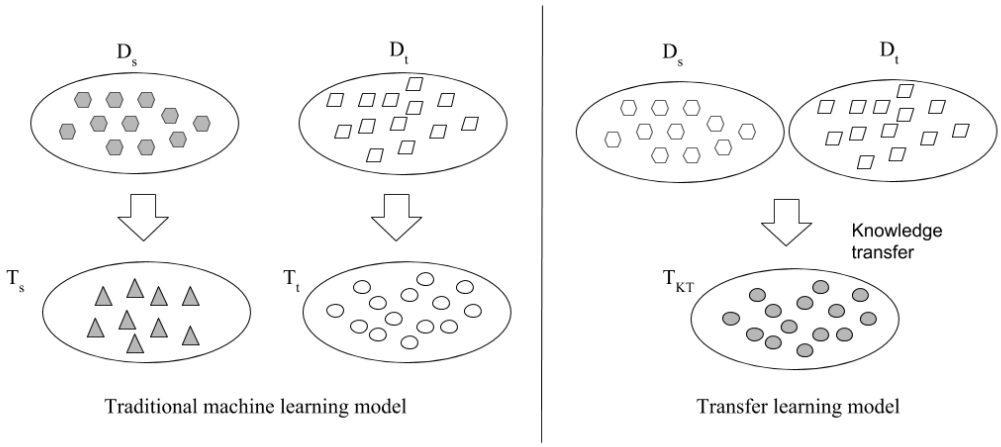

FIGURE 4.6: Learning process in traditional machine learning vs transfer learning setting, where D_S and D_t, denotes source and target domain, T_S and T_t denotes source task, target task and T_{KT} denotes task learned from knowledge transfer.

Transfer learning is a vital technique that generalizes models trained for one setting or task to other settings or tasks (Yosinski et al., 2014). For instance in the speech recognition system, an acoustic model for recognizing speech trained with speech samples from multiple speakers. The same model can be used to create a speech recognition system for a single speaker by adapting the multi-speaker model on a single user's speech samples. This approach is widely used in portable devices for speech recognition without building an explicit speech recognition system from scratch for every single user.

A general misbelief in deep learning is that without large training data, we can't build a system. As we know that data is an important part of training in deep learning, we can also transfer the learned knowledge representation from one task to other related tasks. The learning process is primarily driven by the training data available. With advancement in computational resources and deep learning, large sets of data are available in image, text, and speech, and there is a necessity to exploit the knowledge with transfer learning techniques.

The simple approach in transfer learning is domain adaptation in which existing model adapts the change of distributions from a source domain to target domain. For example, as represented in Figure 4.6 transfer learning, for source and target domain D_S and D_t and the tasks T_S and T_t, we want to extract the knowledge from

both source and target domains to generate target data distribution which has properties from both domains as depicted in Figure 4.6. In our case, we want to extract the speaker's (identity) characteristics from neutral speech parameters and prosody information from emotional speech parameters, which can be seen as training in a semi-supervised setting.

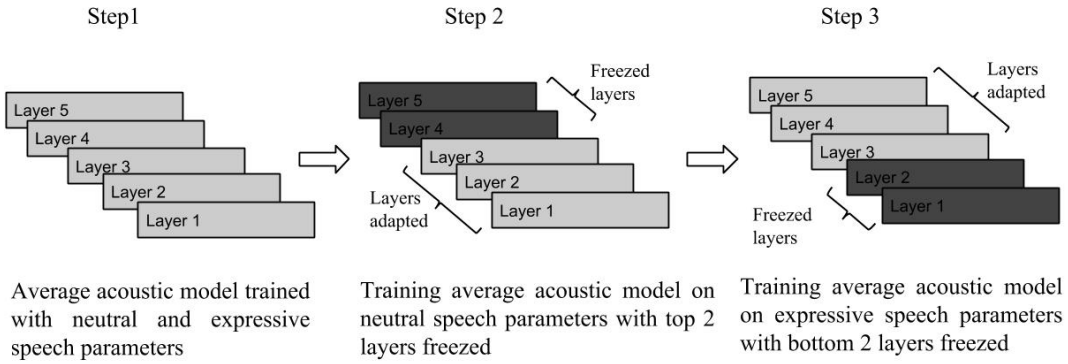

FIGURE 4.7: Layer adaptation technique for transfer of speaker's style characteristics and prosody information

In (Yosinski et al., 2014) reported that first layers in deep neural networks are able to capture global input features representation from the dataset distribution which is not task specific. As we move towards higher layers, hidden representation of layers tends to focus towards task-specific features. We proposed a layer adaptation approach in which we adapt the emotional and neutral acoustic parameters on layers selectively by freezing some of the layer parameters and fine-tuning the other layers. However, when updating the weights, no modifications are applied to the frozen layers. This is a popular technique used in transfer learning to freeze the base model and adapt the system to new data distribution.

Let's recall that our goal is to transfer expressivity information from one speaker to the neutral voice model of another speaker. The proposed following approach is one way to achieve the goal. First, we train an average acoustic model with emotional acoustic parameters from speaker A and neutral acoustic parameters from speaker B. Second, the bottom n layers of average model are frozen and the average acoustic model is adapted with expressive acoustic parameters from speaker A. Third, the top n layers frozen and adapted with neutral acoustic parameters of speaker B as shown in Figure 4.7. In the synthesis phase, we use duration model trained on emotional speech corpus along with the layer adapted acoustic model. In Chapter 6. We will discuss details of the experimentation carried out with variations in freezing the layer and fine-tuning of the network parameters.

Chapter 5

Variational autoencoders

Recent advancement in Bayesian deep learning provided means to amalgamate the deep learning models and probability theory, which enabled us to exploit the learned representation of the data distribution in a latent space. Variational autoencoders provide a powerful framework for learning compressed representations by encoding all of the information needed to reconstruct a data point in a latent code (Berthelot et al., 2018). If we interpolate the latent representation of two data points, the model generates the output which has semantic characteristics from two data distribution. In this chapter, we will discuss variational autoencoder as a generative model and its ability to learn the latent information from speech to control the expressivity in generated speech. The chapter is organized as follows. First, we introduce the autoencoder, variational autoencoder, and conditional variational autoencoder. Second, we discuss the proposed architecture for controlling the expressivity with conditional variational autoencoders.

5.1 Autoencoder

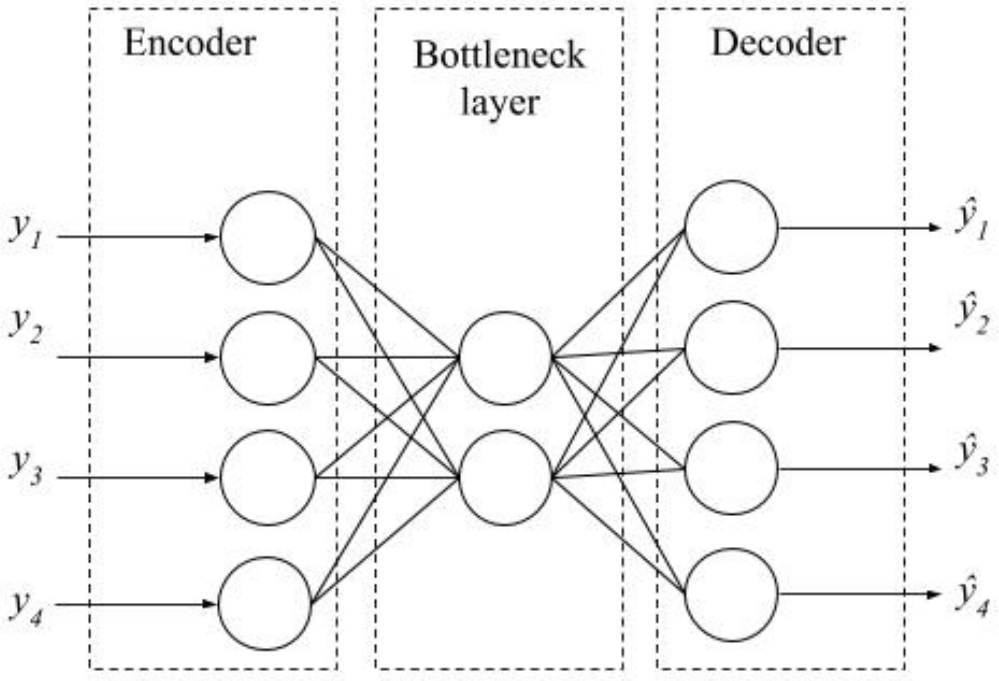

FIGURE 5.1: Autoencoder with bottleneck layer

Autoencoders are unsupervised learning approaches that use neural network architectures to compress and reconstruct the original input data. During the training, the cost function to be minimized is the error (difference) between the reconstructed signal and the original signal. An autoencoder consists of an encoder and a decoder network, where the encoder compresses the input information with up to a bottleneck hidden layer and the decoder reconstructs input signal from hidden representation generated by an encoder as shown in Figure 5.1. During the training of an autoencoder, only the reconstruction error is taken into consideration. Furthermore, we can use autoencoders to compress the meaningful attributes of the original input. In one sense, autoencoders are able to perform dimensionality reduction, which is a nonlinear generalization of principal component analysis. Furthermore, autoencoders are not simply to reconstruct the input but to create the bottleneck layer for hidden representation of input to extract the characteristics of input data.

5.2 Variational autoencoders

Variational autoencoders were introduced in 2013 by Kingma (Kingma and Welling, 2013) and Rezende (Rezende, Mohamed, and Wierstra, 2014) independently. Variational autoencoders have similar components as autoencoders, that is an encoder, decoder and loss function. However, for training loss function corresponds to the reconstruction error (as for the autoencoders) plus a regularization term defined with a Kullback-Leibler (KL) divergence. The encoder takes an input data x and represents it as latent representation z. Thus, the encoder learns to compress the input data into a lower dimension space. Let's denote $Q_\Theta(z|x)$ encoder distribution, which is a Gaussian probability density with parameters Θ. The decoder takes the latent representation z as input and output as reconstructed data sampled from the Gaussian distribution. The distribution of the decoder output is denoted by $P_\phi(x|z)$ with parameters ϕ.

$$Loss(\Theta, \phi) = E_{z\sim Q_\Theta(z|x)}[logP_\phi(x|z)] + KL(Q_\Theta(z|x)||P(z)) \tag{5.1}$$

As mentioned earlier the loss function of variational autoencoder have two terms, the first term is a reconstruction loss or expected log-likelihood, which represent the expectation over the reconstruction of input. This term signifies how well decoder learns to reconstruct the input data. The second term is a regularizer which is the KL divergence between the encoder's distribution $Q_\Theta(z|x)$ and true prior distribution $P(z)$ as mentioned in Equation 5.1. This measure indicates how close the learned distribution $Q_\Theta(z|x)$ is to the true prior distribution $P(z)$. In variational autoencoder, the true prior distribution is specified as a normal distribution with zero mean and unit variance. Thus, the KL divergence term will approximate the latent representation z to have a normal distribution to avoid the penalty. Without the regularizer penalty, the encoder will map the latent state for each data point in different parts of Euclidean space. As we want the latent space representation meaningful, we penalize with a KL divergence to keep the similar representation of data point close to each other. During the inference, we sample from latent space and the decoder network acts as a generative model with the ability to generate new data points similar to the examples seen during the training.

For the implementation of variational autoencoders, the encoder model output two parameters, that is the mean and the variance to describe the latent space distribution. During the training phase of variational autoencoders, we estimate the error gradient with respect to the final output and backpropagate the error and update

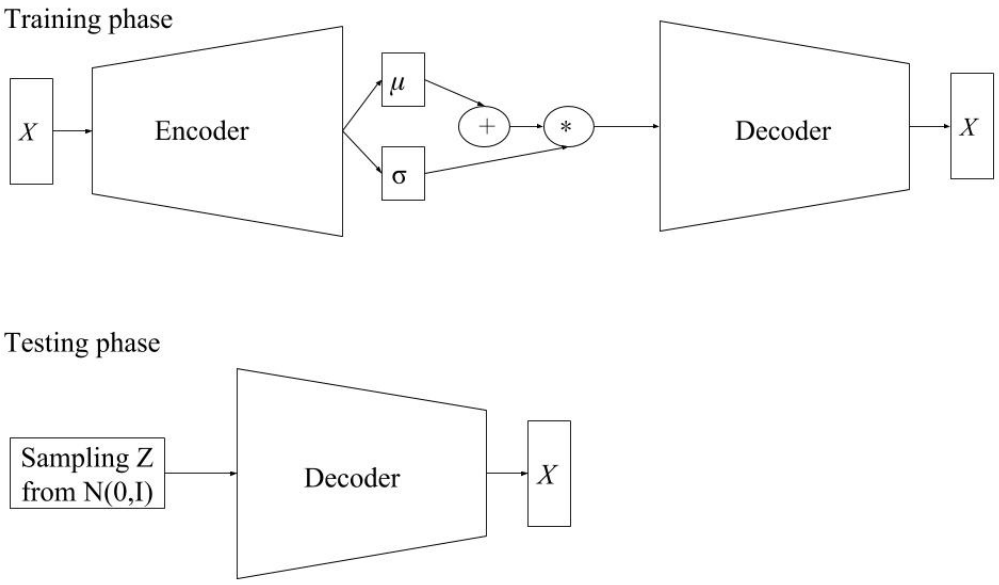

FIGURE 5.2: At training time the variational autoencoder is implemented with a reparameterization trick with input X. At testing-time the variational autoencoder acts as a generative model by sampling from normal distribution with zero mean and unit variance

the parameters using a stochastic gradient descent algorithm. As a stochastic gradient can be applied to stochastic inputs but not stochastic units within a network, to resolve this we use reparameterization trick (Kingma and Welling, 2013). In reparameterization, we randomly sample ϵ from the normal distribution and then shift sampled ϵ randomly with latent distribution's mean μ and then scaling it with the variance σ of latent distribution, as shown in Equation 5.2.

$$z = \mu + \epsilon \cdot \sigma \tag{5.2}$$

5.3 Conditional variational autoencoders

As mentioned earlier, the decoder of a variational autoencoder acts as a generative model, which takes as input a latent variable sampled from the normal distribution. However, with this model, we have no explicit control over the generation process. In traditional variational autoencoder, information about the type of input data is not taken into consideration. Equation 5.3 shows a modification of the loss function that condition the encoder and decoder distribution on a conditional variable c, which provides information about the type of input data that is mapped into the latent space. Therefore, the resulting distributions of the encoder and of the decoder conditioned on c is as given by $Q_\Theta(z|x,c)$ and $P_\phi(x|z,c)$ and true prior distribution as $P(z|c)$. These are conditional probability distributions, which show that for each possible condition c, there exists prior distribution $P(z|c)$. Conditional variational autoencoders allow us to handle input to output mapping as one to many learning problem without explicitly defining the output distribution (Rezende, Mohamed, and Wierstra, 2014).

$$Loss(\Theta, \phi) = E_{z \sim Q_\Theta(z|x)}[logP_\phi(x|z,c)] + KL(Q_\Theta(z|x,c)||P(z|c)) \quad (5.3)$$

5.4 Proposed model for transfer of prosody information

From the baseline speech synthesis we can recall that we model prosody generation using a duration model and an acoustic model. In architecture that we propose here, we assume that the prosody information is contained in the acoustic model, thus we proposed to use an acoustic model as conditional variational autoencoder with expressivity information to map the acoustic information in latent space for emotional speech and neutral speech.

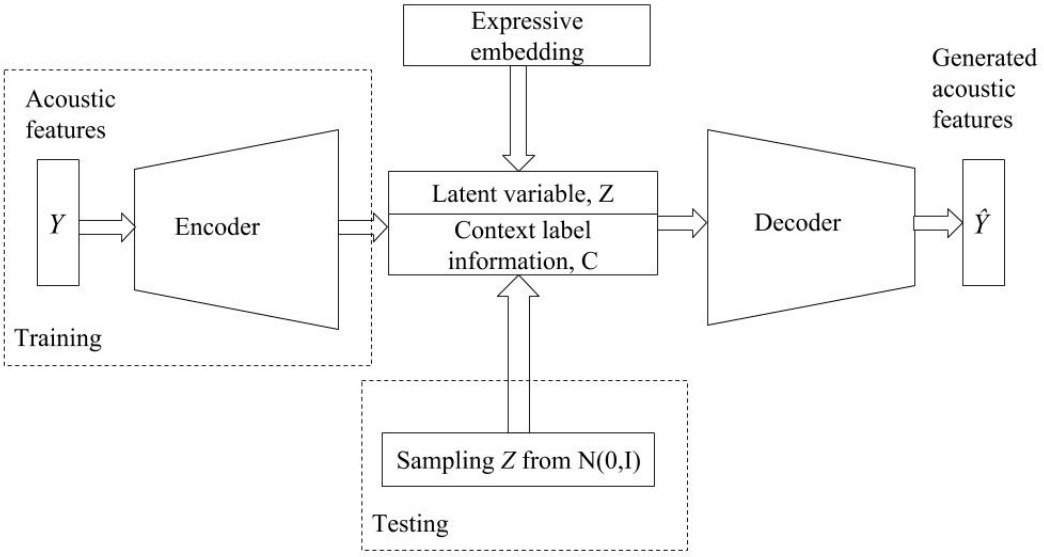

FIGURE 5.3: Proposed model based on conditional variational autoencoder with expressive embedding and conditioned on contextual label feature

The proposed model has an encoder, a decoder and the loss function is defined similarly to conditional variational autoencoder. The encoder takes as input acoustic parameters for each frame and output two vectors mean and variance to define the distribution of the latent space. As the proposed model is an autoencoder, the decoder generates acoustic parameters for each frame conditioned on contextual label features and expressivity information represented as one hot encoded vector. At inference time the decoder generates acoustic parameters y_n from latent variable z sampled from normal distribution concatenated with expressivity representation s and the contextual label features c. Thus, the modified loss function for proposed model is as given in Equation 5.4.

$$Loss(\Theta, \phi) = E_{z \sim Q_\Theta(z|y)}[logP_\phi(y|z,c,s)] + KL(Q_\Theta(z|y,c,s)||P(z|c,s)) \quad (5.4)$$

As mentioned earlier, a variational autoencoder can produce a semantically meaningful representation of acoustic parameters corresponding to contextual label features and expressivity identity. Thus, each attribute of the latent variable z sampled from the distribution shows the characteristics of acoustic parameters. This can be

used to control the prosody of output acoustic parameters. To control the expressivity we interpolated the latent variables along the lines as $\alpha \cdot Z_{neutral} + (1 - \alpha) \cdot Z_{anger}$, where α has some value in a range of [0,1] as illustrated in Figure 5.4. The latent variables are generated through encoder for expressive speech and neutral speech as $Z_{neutral} = f(y_{neutral})$ and $Z_{anger} = f(y_{anger})$, where $y_{neutral}$ and y_{anger} are acoustic parameters from neutral speech and emotion, anger speech, also $f(\cdot)$ refers to the encoder function. With this interpolation, the model should be able to generate the acoustic parameters which have averaged value defined by factor α. The proposed architecture is able to control the expressivity for the generation of the acoustic parameters in an unsupervised setting. We will discuss the experimentation conducted with the proposed model in Chapter 6.

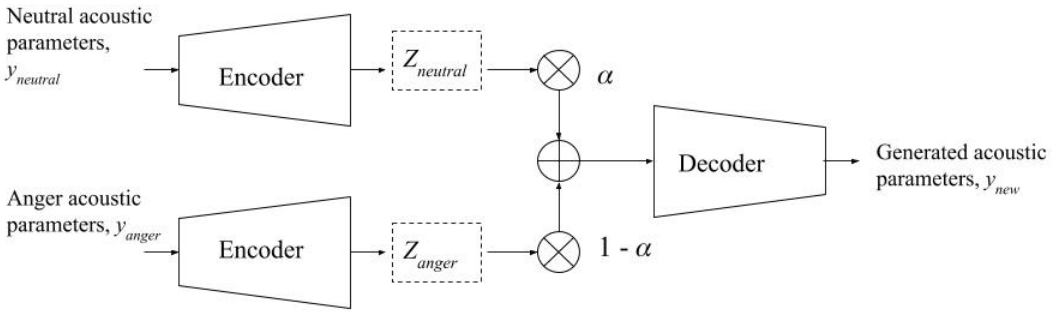

FIGURE 5.4: Interpolation of latent representation of neutral and anger acoustic parameters

Chapter 6

Experimentation and results

In this chapter, we will investigate the approaches proposed in Chapter 4 and Chapter 5. First, we discuss the baseline expressive speech synthesis models with experimentation, subjective and objective evaluation. Second, we discuss the various experiments conducted with layer adaptation approach and the subjective evaluation of transfer of expressive information. Third, we will discuss the experiments with a proposed architecture based on the variational autoencoders.

6.1 Baseline expressive speech synthesis system

As explained in Chapter 4, the development of our baseline expressive speech synthesis is based on deep neural network framework [1] [2] for implementing the duration model and the acoustic model. In this thesis work, we used Adam optimizer with learning rate 0.001 and default parameters that are η=0.001, β_1=0.9, β_2=0.999 for computing running averages of gradient and its square; $\epsilon = 1 \times 10^{-8}$ term added to the denominator to improve numerical stability. The speech signal was used at a sampling rate of 16 KHz and each speech corpus is divided into train, validation, and test set in the ratio of 80, 10, 10 respectively. In Chapter 3, we explained the data preparation process to create features for training from text and speech. Here, we use 1356 contextual label features as input for the duration model, which generates the duration parameters for each sound unit (phoneme or pause). The acoustic model is trained with the same 1365 input contextual label features plus 9 features for phone duration information, and for output 187 acoustic parameters namely Mel generalized cepstrum (mgc) coefficients, log of fundamental frequency (lf0), band-aperiodicity (bap), and voiced-unvoiced information (vuv) for each 5ms time frame. We trained each model till 25 iterations, mini-batch size of 64 and a dropout rate of 0.1. We used early stopping mechanism for 10 epochs, which will stop the training if the loss on validation set does not improve. From Christine corpus, we used all emotions that are neutral, joy, surprise, fear, anger, sadness, and disagree with 500 utterances from each. We considered also 1815 utterances from Lorene corpus for baseline neutral speech synthesis. Only neutral speech is available in the Lorene corpus, and our goal is to transfer expressive speech characteristics from Christine corpus to the Lorene neutral speech synthesis model. After experimentation with various parameters such as the number of layers, number of hidden units, learning rate and dropout, we built baseline speech synthesis systems using the following deep learning configurations,

- DNN : Feedforward neural network with 5 hidden layers, each having 512 units

[1] Merlin: The Neural Network (NN) based Speech Synthesis System (Wu, Watts, and King, 2016)
[2] PyTorch : Tensors and Dynamic neural networks in Python, URL

- LSTM : 5 RNN-LSTM hidden layers, each having 512 units
- Pretrained DNN : Feedforward with 5 hidden layers and 512 units per layer, then model is adapted for emotional style.

The performance of baseline models is measured with objective metrics and subjective listening tests. We conducted the error analysis for the objective evaluation using acoustic parameters generated by the baseline model and the acoustic parameters generated by the vocoder from speech corpus data. We used the following metrics :

- Mel Cepstrum Distortion (MCD) for Mel generated cepstrum, in decibel [dB] scale
- Root mean square error (RMSE) of Fundamental frequency prediction, in Hertz [Hz],
- Band aperiodicity distortion for band aperiodicity, in decibel [dB] scale
- Voiced-unvoiced prediction accuracy, in percentage (%)

The objective results for architectures mentioned before with the WORLD vocoder are presented in Table 6.1.

TABLE 6.1: Comparison of objective results using the WORLD vocoder. MCD: Mel-Cepstral Distortion in dB. BAP: distortion of band aperiodicities in dB. F0 RMSE is calculated on a linear scale in Hz, V/UV: voiced/unvoiced % error, DNN is feed forward neural network, LSTM is long short term memory, Lorene and Christine refers to the name of speech corpus used.

Model	Emotion	MCD	BAP	F0 RMSE	VUV
DNN	Lorene neutral	4.52	0.163	24.2	8.4
LSTM	Lorene neutral	6.84	0.327	25.3	22.7
DNN	Christine neutral	4.97	0.221	26.7	6.9
LSTM	Christine neutral	7.09	0.348	29.9	24.0
DNN	Christine anger	5.36	0.261	41.2	8.6
DNN	Christine joy	6.00	0.315	46.5	11.5
DNN	Christine fear	5.88	0.221	28.8	9.8
DNN	Christine sad	5.89	0.238	35.2	10.0
DNN	Christine disagree	5.84	0.243	38.5	10.3
Pretrained	Christine anger	6.76	0.313	42.8	14.8

We performed MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) listening test for subjective evaluation of anger baseline model (Schinkel-Bielefeld, 2017). MUSHRA test has the advantage over mean opinion score as fewer participants are required to obtain statistically significant results. In the MUSHRA test, there were 7 French listeners and each participant scored 15 sets that were randomly selected from the test set. Each set of stimuli consisted of 3 stimuli, two from baseline anger model and pretrained anger model as an anchor and one as hidden reference stimuli synthesized by vocoder from original acoustic parameters. The listeners

FIGURE 6.1: Graphical interface for MUSHRA scores, with 3 stimuli's to evaluate

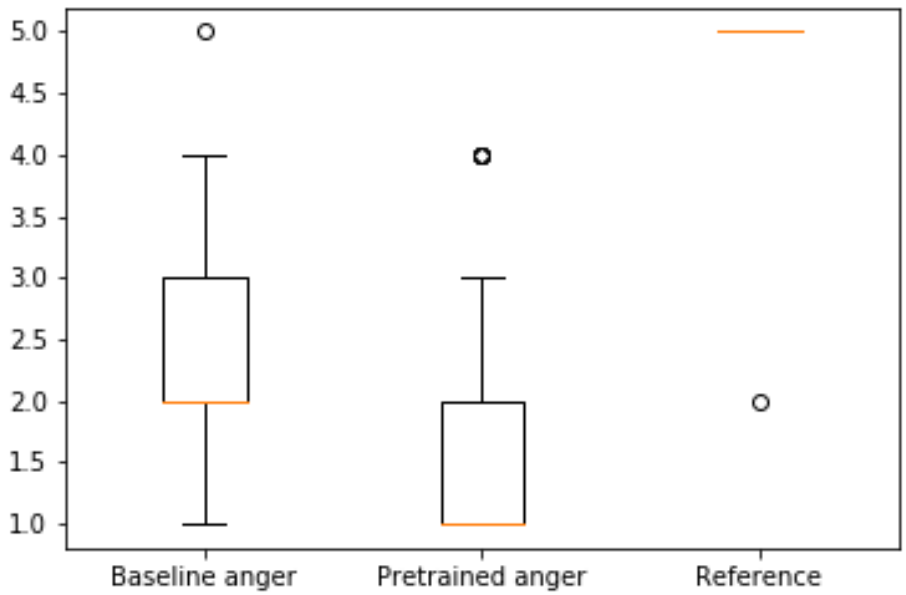

FIGURE 6.2: Box plots on MUSHRA scores for baseline anger, pretrained anger and reference

were instructed to score each stimuli from 1 to 5 and score one of them in each set as 5, which was expected to score the hidden reference stimuli as depicted in Figure 6.1. For MUSHRA listening test we used stimuli's from anger emotion baseline model from DNN model and pretrained model. The result of MUSHRA listening test are reported in Figure 6.2. It is observed that baseline anger model is slightly better than pretrained anger model. It is also found that listeners correctly identified the hidden reference in each set of stimuli. Mel cepstrum distortion score for baseline anger model is 5.362, while pretrained anger model's score 6.763. Thus, subjective results are coherent with objective results. Furthermore, Lorene and Christine baseline models with DNN outperformed the LSTM models. We observed that both speech corpora have long sentences, which may lead to a vanishing gradient problem during the training.

6.2 Layer adaptation for transfer of prosody information

First, we trained an average acoustic model with emotional and neutral acoustic speech data from different speakers; the model is a feedforward neural network, 5 hidden layers, 512 units. Second, the bottom n layers of the average acoustic model were frozen and the model (parameters associated to layers that are not frozen) is trained with emotional acoustic parameters. Third, top n layers were frozen and trained with the neutral acoustic feature from Lorene corpus. We conducted several experiments by using different values of n i.e. number of frozen layers and training iteratively first on the neutral acoustic parameters and then anger acoustic parameters vice-versa, for details refer Table 6.2.

TABLE 6.2: Experimentation with layer adapted models, where L_i refers to i^{th} layer, Lorene and Christine refers to name of corpus used

Name of Model	Layers frozen	Corpus	Layers frozen	Corpus
Model 1	[L3, L4, L5]	Lorene neutral	[L1, L2, L3]	Christine anger
Model 2	[L3, L4, L5]	Christine anger	[L1, L2, L3]	Lorene neutral
Model 3	[L4, L5]	Lorene neutral	[L1, L2]	Christine anger
Model 4	[L4, L5]	Christine anger	[L1, L2]	Lorene neutral

We then synthesized speech from acoustic parameters generated with models trained using layer adaptation and baseline models trained on neutral and anger emotions. We predicted the Mel cepstrum coefficient using layer adapted model, while fundamental frequency, aperiodicity, voiced-unvoiced flag were predicted from anger baseline model (DNN Christine anger). We presented the experimentation with this approach in Table 6.3.

TABLE 6.3: Experimentation with layer adapted models, where baseline anger refers to Christine anger baseline model trained using DNN, acoustic mgc model refers to mgc parameters used from the model and acoustic lf0 model refers to lf0, bap and vuv parameters used from the model

Name of Model	Duration model	mgc model	lf0 model
Model 5	Baseline anger	Model 1	Baseline anger
Model 6	Baseline anger	Model 2	Baseline anger
Model 7	Baseline anger	Model 3	Baseline anger
Model 8	Baseline anger	Model 4	Baseline anger

The main objective of the proposed approach is to build expressive speech synthesis system which will be able to synthesize speech with speaker's characteristics from Lorene corpus and expressive attribute from Christine corpus. As there are no reference speech samples to evaluate how much expressivity is transferred without modifying the speaker's characteristics, we propose to use a mean opinion score based on subjective evaluation. The mean opinion score is a subjective evaluation in which participants have to score stimuli from 1 to 5, where 1 is bad and 5 is excellent (Streijl, Winkler, and Hands, 2016). In this test, there were 7 French listeners and each participant scored 15 sets of stimuli randomly selected from the test set from Christine corpus. Each set consisted of 3 stimuli from model 4, model 8 and baseline anger. We conducted informal listening tests of all layer adapted models mentioned in Table 6.2, in which model 4 and model 8 exhibited the most expressive characteristics from the anger emotion and speaker's characteristics from Lorene corpus. The listeners instructed to give two scores from 1 to 5, first to evaluate anger emotion in stimuli compared to reference anger stimuli, and second to evaluate speaker's characteristics in stimuli compared reference speakers stimuli from Lorene speech samples as shown in Figure 6.5. To track the evaluation process baseline anger stimuli was used as we expected that participants will score baseline anger stimuli more to expressivity score and less to the speaker's characteristics score. We present the subjective evaluation results for expressivity and speakers characteristics in Figure 6.3 and Figure 6.4. From Figure 6.3, mean opinion scores for the expressivity shows that model 8 performed slightly better than model 4. Similarly, mean opinion scores for the speaker's characteristics model 8 performed better than model 4 as shown in Figure 6.4. The subjective results showed that model 8 outperformed the model 4. Also, results on baseline anger model shown the highest score for expressivity and lowest score for speaker's characteristics, which is consistent with the designed test.

6.3 Proposed model based on variational autoencoder

We proposed conditional variational autoencoder along with expressive embedding for representing acoustic parameters in latent space and control the expressivity through interpolation of latent representation as discussed in Chapter 5. The encoder takes as input 187 acoustic parameters per frame and the decoder generates as output 187 acoustic parameters per frame, with expressive embedding as one hot encoded vector of dimension 2. We used the prediction of the baseline anger duration model as conditioning to the variational autoencoder with 1365 contextual label features plus duration features. The latent space was 20-dimensional. The encoder and decoder were feedforward neural networks with hidden layers as [187,164,80,40,20]

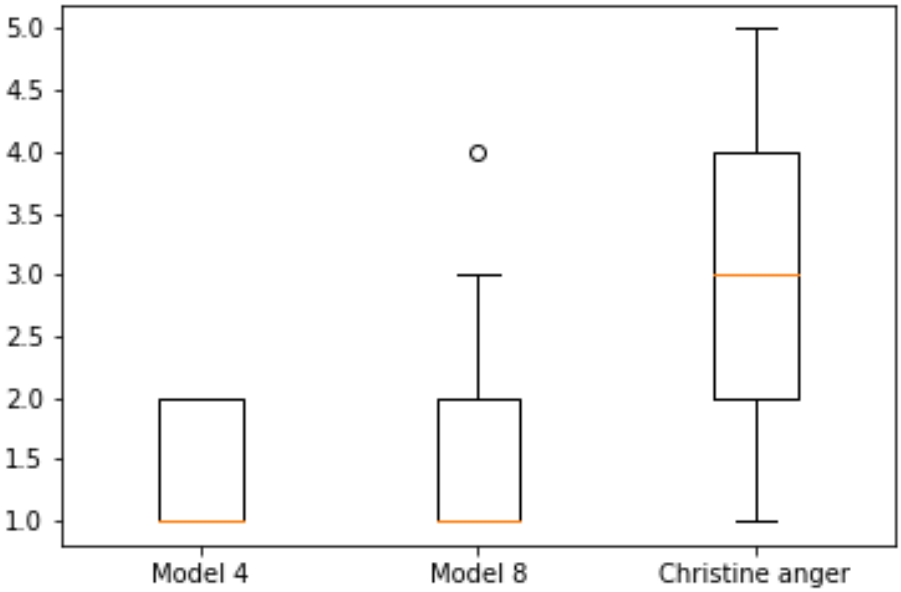

FIGURE 6.3: Box plots on mean opinion scores for expressivity on model 4, model 8 and baseline Christine anger

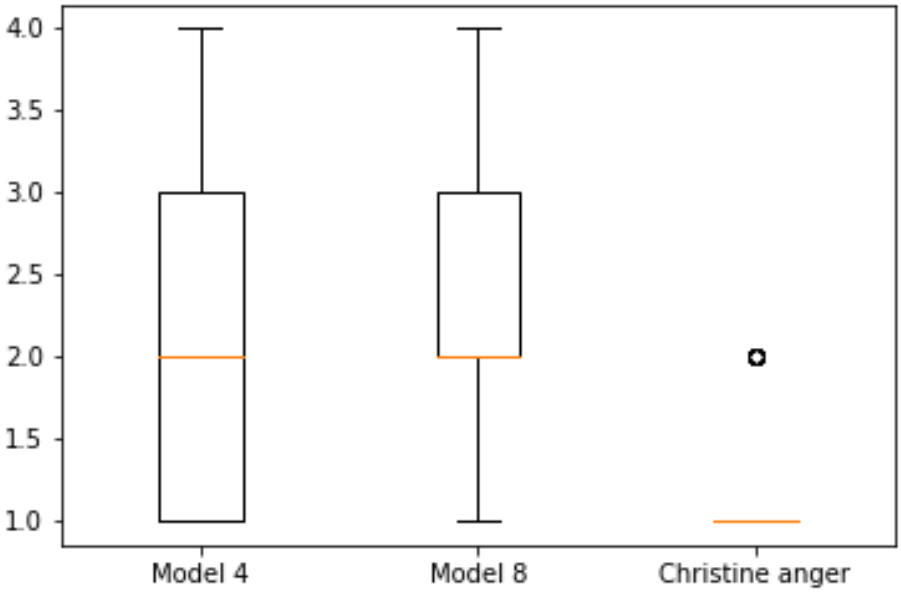

FIGURE 6.4: Box plots on mean opinion score for speaker's characteristics of model 4, model 8 and baseline Christine anger

and [1387,650,325,187] respectively. Because of conditioning (1365), latent variable (20) and expressive embedding (2), the size of input of the decoder was 1387. Exponential linear units (ELU) were applied to each layer for non-linearity in the neural network (Clevert, Unterthiner, and Hochreiter, 2015), which speeds up the learning

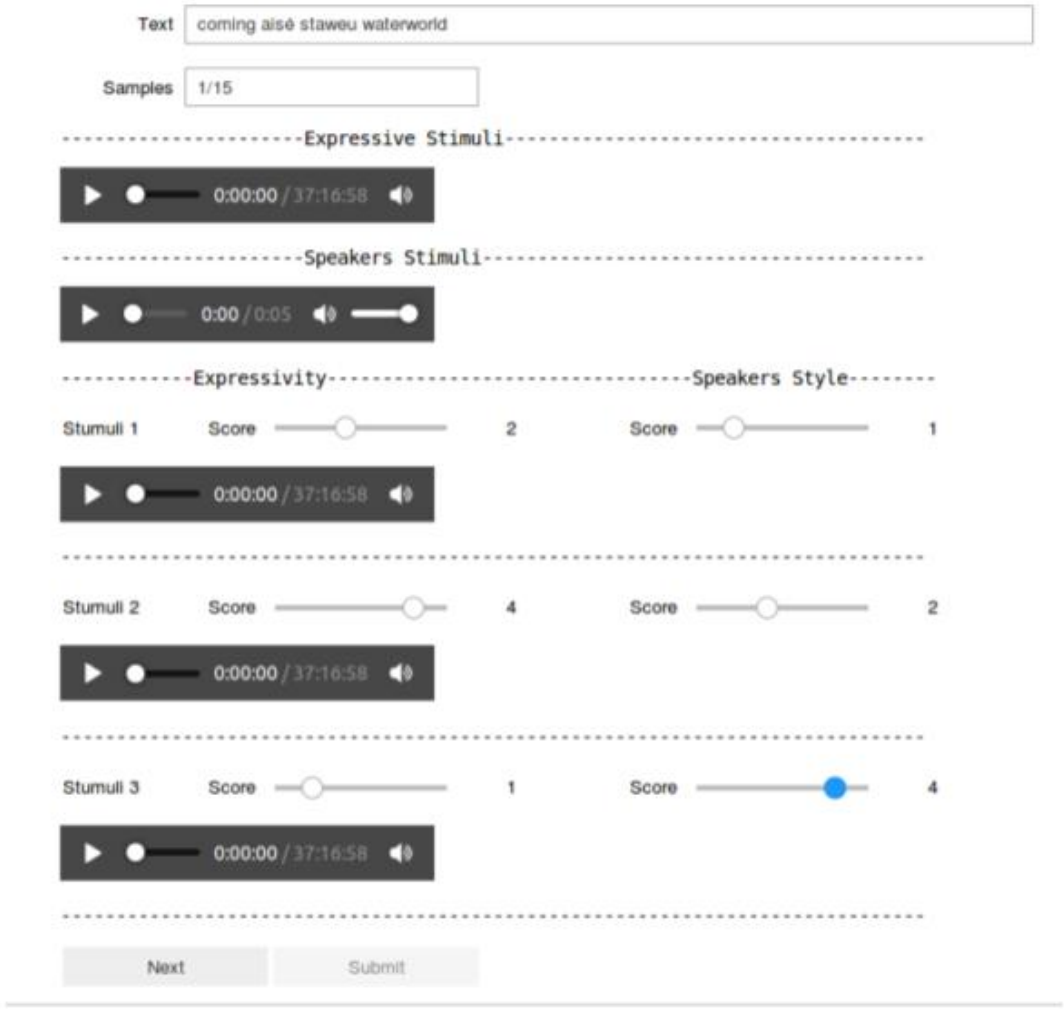

FIGURE 6.5: Graphical interface for MOS scores, with stimuli's to evaluate with the aid of reference expressive stimuli and reference speaker's stimuli

process in deep neural networks and leads to ease the vanishing gradient problem. For training the model, we used 500 utterances from Christine anger corpus and Lorene neutral corpus. We trained the mode for 50 iterations, mini-batch size for training was 64 and optimizer was stochastic gradient descent with learning rate 0.001. We used mean square error loss function to measure the reconstruction loss between input acoustic parameters and the acoustic parameters generated by the decoder. The prior distribution for the latent variables was a standard normal distribution with zero mean and unit variance. We conducted the following experiments with conditional variational autoencoder as given below,

- Conditional variational autoencoder with different latent space dimension, z = 2, 4, 8, 20, 40 and 60
- Acoustic parameters generated from baseline anger model (y_{anger}) and baseline neutral model ($y_{neutral}$) were given to encoder. Then, encoder generated latent representation Z_{anger} and $Z_{neutral}$, which are linearly interpolated as $\alpha \cdot Z_{neutral} + (1 - \alpha) \cdot Z_{anger}$ with interpolation factor, α = 0.2,0.5,0.8

- Conditional variational autoencoder with multiplying factor to KL loss term in Equation 5.3. The decaying of the β factor emphasizes more on learning of the prior distribution (Burgess et al., 2018). The modified loss function is as given below,

$$Loss(\Theta, \phi) = E_{z\ Q_\Theta(z|y)}[logP_\phi(y|z,c,s)] + \beta \cdot KL(Q_\Theta(z|y,c,s)||P(z|c,s)) \quad (6.1)$$

The synthesized speech samples using various latent space dimension shows that decreasing the dimension of latent space results in degradation of the quality of generated speech. We used the β factor to KL loss term, so that encoder's distribution represents the prior distribution. After applying the beta factor we did not notice any improvement in synthesized speech with interpolation. In the end, we interpolated the latent representation generated on anger and neutral acoustic parameters. We analyzed that speech synthesized through interpolation have small variations with insignificant expressivity content in the speech signal. Thus, after consideration of synthesized speech with interpolation and human resources required for the listening test, we decided not to perform the subjective evaluation.

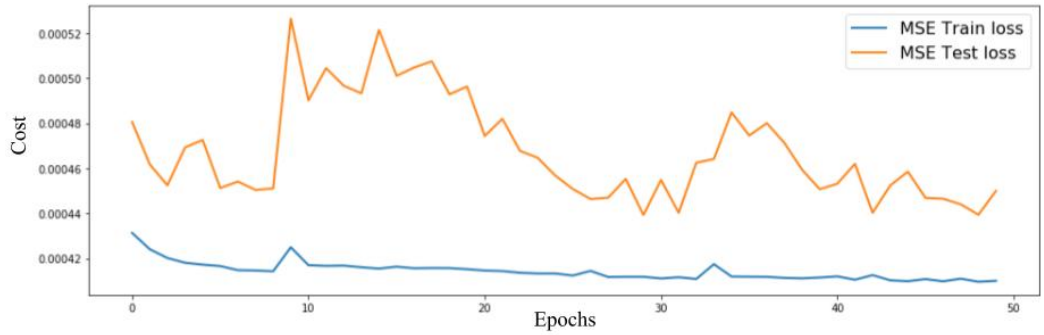

FIGURE 6.6: Mean square error (MSE) as reconstruction loss for the proposed model with the latent variable dimension 20 till 50 epochs

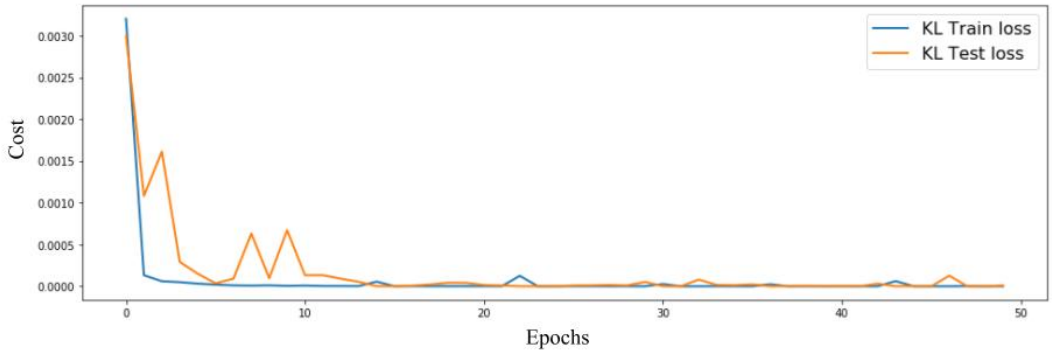

FIGURE 6.7: Kullback–Leibler (KL) divergence loss for proposed model with latent variable dimension 20 till 50 epochs

Both KL divergence loss and MSE loss suggests that model is converging to minima as shown in Figure 6.6 and Figure 6.7. However, the model quickly learns to minimize the KL divergence term leading latent variable z to map close to zero mean and unit variance in a normal distribution, consequently, encoding no useful information from acoustic parameters. To alleviate this issue, at the beginning of training we added a variable weight to the KL divergence term with value 0 and increased the weight gradually till it reaches 1 (Bowman et al., 2015). Even after applying this

trick, we did not notice significant improvements in controlling expressivity through interpolation of the latent variables. Further investigation and experiments are to be performed on this approach.

Chapter 7

Conclusion

In this thesis, we worked on expressive speech synthesis, and we have more particularly developed several baseline speech synthesis systems and investigated the transfer of expressivity information from expressive speech corpora to the neutral model of another speaker. We presented a deep neural network based speech synthesis, build for each emotion explicitly. Addition to this, we explored feed forward neural network and recurrent neural network long short-term memory architectures for implementation of the duration model and of the acoustic model. We observed that due to the long length of text in both speech corpus (Lorene and Christine), recurrent neural network based models couldn't perform well compared to feedforward neural networks. For training expressive speech synthesis, we had only 500 utterances for each emotion. Therefore, to exploit the existing speech corpus, we also trained expressive speech synthesis using pretraining, where baseline speech synthesis was initialized with a model trained on neutral speech corpus.

In Chapter 6, we presented objective results using Mel cepstrum distortion, root mean square error for fundamental frequency, band aperiodicity distortion and voiced-unvoiced prediction error. We conducted a subjective evaluation using the MUSHRA listening test to evaluate baseline anger model and pretrained anger model with the hidden reference signal. The results of subjective evaluation shown that participants were able to identify the hidden reference stimuli in all the sets. As the pretrained model is first trained on neutral speech corpus, synthesized speech from the pretrained model had a neutral component in synthesized anger speech. Furthermore, after listening to the expressive speech corpus, we observed that when a person is speaking with emotion, not all the words are emphasized with emotions. This creates a big challenge in the subjective evaluation of such emotions. As mentioned in Chapter 3, emotional speech corpus is recorded by professional actors acting different emotions; moreover semantics of recorded sentences were inconsistent with the content of the recorded speech. During the evaluation process, listeners faced difficulties to evaluate such stimuli.

In order to transfer the prosody information from the expressive speech corpus to the neutral model of another speaker, we proposed layer adaptation approach and conditional variational autoencoder model. In Chapter 5, we presented layer adaptation approach which is similar to domain adaptation, where layers are adapted to emotional and neutral speech corpus iteratively. We designed the experimental setup for this approach assuming that first layers in neural network extract the global feature representation and as we go to top layers, learned feature representation is more task specific. Furthermore, we conducted experiments using prediction of acoustic parameters from different models such as Mel cepstrum features from layer adapted model and fundamental frequency, band aperiodicity, voiced-unvoiced from baseline anger model. We conducted only a subjective evaluation of the layer adaptation approach due to unavailability of reference emotional speech

in Lorene corpus. We proposed a listening test based on mean opinion score, where we asked listeners to provide two different scores, one for measuring how much expressivity is in stimuli with reference expressive stimuli, and the other for measuring speaker's characteristics similarity compared with reference speaker's stimuli. The subjective results showed that the usage of acoustic parameters predicted from different models (baseline Christine anger and layer adapted model) enhance the performance of transferring the expressive characteristics than using single layer adapted model. This show that modeling acoustic parameters as separate channels improves the transferability. As speaker's characteristics and expressivity are both represented by the same acoustic parameters, it is not a trivial task to disentangle those aspects with acoustic parameters alone.

In the end, we proposed conditional variational autoencoder to represent acoustic parameters in latent space. The main objective of this work was to learning the representation of speaker's characteristics and expressivity attributes in latent space and control the expressivity through interpolation. The experimentation with different values for the latent space dimension showed that there is a loss of information in the decoding process which is dependent on the size of the latent variable. The synthesized speech samples through the interpolation process showed the insignificant modification in expressivity, while observed speaker's characteristic was influenced by neutral acoustic parameters and anger acoustic parameters from Lorene and Christine corpus respectively. As there is no process in place to identify each attribute of latent space representation, it is difficult to analyze if latent space representation learned to disentangle the speaker's characteristics from expressivity. During the training of conditional variational autoencoders, KL divergence loss between encoder's distribution and prior distribution was minimal, but it doesn't make sure that the encoder's mean and variance parameters learned the desired attributes of acoustic parameters. For the interpolation of latent representation, we assumed that the manifold of interpolation is to be flat, but we don't have access to the manifold of latent space distribution.

The future line of work envisaged removing the bottleneck modules in speech synthesis systems as well as an enhancement over proposed models for transfer of prosody information. First, creating an alternative representation of text instead of context labels, this can be developed using character embedding of the input text, which will able to understand the semantic and prosody information. For the generation of the speech waveform, usage of vocoder stage losses some information during parameterization. Recently published work with neural vocoder has shown significant improvements in generating speech waveform directly (Mehri et al., 2016), (Oord et al., 2016) . Furthermore, explicit modeling of duration model creates difficulty in transferring the prosodic features. For this approach what can be explored is to use an encoder-decoder architecture with attention mechanism, which will generate a speech waveform directly conditioned on character embedding. The possible enhancement in variational autoencoder model would be a use of adversarial network along with the decoder to discriminate expressive and neutral speech (Makhzani et al., 2015).

It is hard to identify which parameters of the neural network represent the attributes of speaker characteristics and of expressivity. Moreover, expressivity and speaker characteristics are bounded aspects of prosody parameters. The work done in this thesis is a preliminary work and it laid the groundwork for future implementations for transferring the prosody information.

Bibliography

Akuzawa, Kei, Yusuke Iwasawa, and Yutaka Matsuo (2018). "Expressive Speech Synthesis via Modeling Expressions with Variational Autoencoder". In: *CoRR* abs/1804.02135. arXiv: 1804.02135. URL: http://arxiv.org/abs/1804.02135.

Berthelot, David et al. (2018). "Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer". In: URL: https://arxiv.org/abs/1807.07543.

Blaauw, M. and J. Bonada (2016). "Modeling and Transforming Speech Using Variational Autoencoders". In: *Interspeech*. San Francisco, USA.

Bowman, Samuel R. et al. (2015). "Generating Sentences from a Continuous Space". In: *CoRR* abs/1511.06349. arXiv: 1511.06349. URL: http://arxiv.org/abs/1511.06349.

Burgess, Christopher P. et al. (2018). "Understanding disentangling in Beta-VAE". In: *arXiv:1804.03599*. URL: \url{https://arxiv.org/abs/1804.03599}.

Cahn, Janet (1990). "The Generation of Affect in Synthesized Speech". In: *Journal of the American Voice I/O Society* 8, pp. 1–19.

Cawley, G. C. and P. D. Noakes (1993). "LSP speech synthesis using backpropagation networks". In: pp. 291–294.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2015). "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: *CoRR* abs/1511.07289. arXiv: 1511.07289. URL: http://arxiv.org/abs/1511.07289.

Fan, Yuchen et al. (2014). "TTS synthesis with bidirectional LSTM based recurrent neural networks". In: *INTERSPEECH*.

Fernandez, Raul and Bhuvana Ramabhadran (2007). "Automatic exploration of corpus-specific properties for expressive text-to-speech: a case study in emphasis". In:

Graves, Alex (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey E. Hinton (2013). "Speech Recognition with Deep Recurrent Neural Networks". In: *CoRR* abs/1303.5778. arXiv: 1303.5778. URL: http://arxiv.org/abs/1303.5778.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hsu, Chin-Cheng et al. (2016). "Voice Conversion from Non-parallel Corpora Using Variational Auto-encoder". In:

Hsu, Chin-Cheng et al. (2017). "Voice Conversion from Unaligned Corpora using Variational Autoencoding Wasserstein Generative Adversarial Networks". In: *CoRR* abs/1704.00849. arXiv: 1704.00849. URL: http://arxiv.org/abs/1704.00849.

King, Simon (2010). "A tutorial on HMM speech synthesis". In:

Kingma, Diederik P. and Max Welling (2013). "Auto-Encoding Variational Bayes." In: *CoRR* abs/1312.6114. URL: http://dblp.uni-trier.de/db/journals/corr/corr1312.html#KingmaW13.

Lemmetty, Sami (1999). "History and Development of speech synthesis". In: pp. 4–10. URL: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/thesis.pdf.

Long, Mingsheng et al. (2015). "Learning Transferable Features with Deep Adaptation Networks". In: ICML'15, pp. 97–105. URL: http://dl.acm.org/citation.cfm?id=3045118.3045130.

Makhzani, Alireza et al. (2015). "Adversarial Autoencoders". In: *CoRR* abs/1511.05644. arXiv: 1511.05644. URL: http://arxiv.org/abs/1511.05644.

Mehri, Soroush et al. (2016). "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model". In: *CoRR* abs/1612.07837. arXiv: 1612.07837. URL: http://arxiv.org/abs/1612.07837.

Morise, Masanori (2012). "PLATINUM: A method to extract excitation signals for voice synthesis system". In: *Acoustical Science and Technology* 33.2, pp. 123–125. DOI: 10.1250/ast.33.123.

— (2015). "CheapTrick, a spectral envelope estimator for high-quality speech synthesis". In: vol. 67, pp. 1 –7. DOI: https://doi.org/10.1016/j.specom.2014.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0167639314000697.

Morise, Masanori, Hideki Kawahara, and Haruhiro Katayose (2009). "Fast and Reliable F0 Estimation Method Based on the Period Extraction of Vocal Fold Vibration of Singing Voice and Speech". In: *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. URL: http://www.aes.org/e-lib/browse.cfm?elib=15165.

Morise, Masanori, Fumiya Yolomori, and Kenji Ozawa (2016). "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications". In: *IEICE Transactions on Information and Systems* E99.D.7, pp. 1877–1884. DOI: 10.1587/transinf.2015EDP7457.

Murray, Iain and John L. Arnott (1995). "Implementation and testing of a system for producing emotion-by-rule in synthetic speech". In: 16, pp. 369–390.

Oliva, Marcela Charfuelan and Ingmar Steiner (2013). "Expressive speech synthesis in MARY TTS using audiobook data and EmotionML". In:

Oord, Aäron van den, Oriol Vinyals, and Koray Kavukcuoglu (2017). "Neural Discrete Representation Learning". In: *CoRR* abs/1711.00937. arXiv: 1711.00937. URL: http://arxiv.org/abs/1711.00937.

Oord, Aäron van den et al. (2016). "WaveNet: A Generative Model for Raw Audio". In: *Arxiv*. URL: https://arxiv.org/abs/1609.03499.

Ouni, Slim et al. (2016). "Acoustic and Visual Analysis of Expressive Speech: A Case Study of French Acted Speech". In: *Interspeech 2016*. Vol. 2016. ISCA. San Francisco, United States, pp. 580 –584. DOI: 10.21437/Interspeech.2016-730. URL: https://hal.inria.fr/hal-01398528.

Rabiner, Lawrence R. (1990). "Readings in Speech Recognition". In: ed. by Alex Waibel and Kai-Fu Lee, pp. 267–296. URL: http://dl.acm.org/citation.cfm?id=108235.108253.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1278–1286. URL: http://proceedings.mlr.press/v32/rezende14.html.

Schinkel-Bielefeld, Nadja (2017). "Audio Quality Evaluation in MUSHRA Tests–Influences between Loop Setting and a Listeners' Ratings". In: *Audio Engineering Society Convention 142*. URL: http://www.aes.org/e-lib/browse.cfm?elib=18655.

Schröder, Marc (2009). "Expressive Speech Synthesis: Past, Present, and Possible Futures". In: ed. by Jianhua Tao and Tieniu Tan, pp. 111–126. DOI: 10.1007/978-1-84800-306-4_7. URL: https://doi.org/10.1007/978-1-84800-306-4_7.

Skerry-Ryan, R. J. et al. (2018). "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron". In: *CoRR* abs/1803.09047. arXiv: 1803.09047. URL: http://arxiv.org/abs/1803.09047.

Sotelo, Jose et al. "Char2wav: End-to-end Speech Synthesis". In:

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Streijl, Robert C., Stefan Winkler, and David S. Hands (2016). "Mean Opinion Score (MOS) Revisited: Methods and Applications, Limitations and Alternatives". In: *Multimedia Syst.* 22.2, pp. 213–227. ISSN: 0942-4962. DOI: 10.1007/s00530-014-0446-1. URL: http://dx.doi.org/10.1007/s00530-014-0446-1.

Taigman, Yaniv et al. (2017). "Voice Synthesis for in-the-Wild Speakers via a Phonological Loop". In: *CoRR* abs/1707.06588. arXiv: 1707.06588. URL: http://arxiv.org/abs/1707.06588.

Tuerk, Christine and Tony Robinson (1993). "Speech synthesis using artificial neural networks trained on cepstral coefficients." In: URL: http://dblp.uni-trier.de/db/conf/interspeech/eurospeech1993.html#TuerkR93a.

Wang, Yuxuan et al. (2017). "Tacotron: A Fully End-to-End Text-To-Speech Synthesis Model". In: *CoRR* abs/1703.10135. arXiv: 1703.10135. URL: http://arxiv.org/abs/1703.10135.

Weijters, T. and J. Thole (1993). "Speech synthesis with artificial neural networks". In: pp. 1764–1769.

Wu, Zhizheng, Oliver Watts, and Simon King (2016). "Merlin: An Open Source Neural Network Speech Synthesis System". In: pp. 202–207.

Yosinski, Jason et al. (2014). "How transferable are features in deep neural networks?" In: *CoRR* abs/1411.1792. arXiv: 1411.1792. URL: http://arxiv.org/abs/1411.1792.

Young, S.J. and Sj Young (1994). "The HTK Hidden Markov Model Toolkit: Design and Philosophy". In: *Entropic Cambridge Research Laboratory, Ltd* 2, pp. 2–44.

Zen, H., A. Senior, and M. Schuster (2013). "Statistical parametric speech synthesis using deep neural networks". In: pp. 7962–7966. ISSN: 1520-6149. DOI: 10.1109/ICASSP.2013.6639215.

Zen, Heiga (2006). "An example of context-dependent label format for HMM-based speech synthesis in English". In: URL: https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/F0parametrisation/hts_lab_format.pdf.

Zen, Heiga, Keiichi Tokuda, and Alan W. Black (2009). "Review: Statistical Parametric Speech Synthesis". In: *Speech Commun.* 51.11, pp. 1039–1064. ISSN: 0167-6393. DOI: 10.1016/j.specom.2009.04.004. URL: http://dx.doi.org/10.1016/j.specom.2009.04.004.