Speaker De-Identification using a Factorized Hierarchical Variational Autoencoder

Master Thesis Mathias Bülow Kastbjerg

Aalborg University Mathematical Engineering

Aalborg University, June 7, 2018

Mathias Bülow Kastbjerg <mkastb13@student.aau.dk>

Copyright © Aalborg University 2018



Mathematical Engineering Aalborg University http://www.aau.dk

AALBORG UNIVERSITY

STUDENT REPORT

Title:

Speaker De-Identification using a Factorized Hierarchical Variational Autoencoder

Project Period: Spring Semester 2018

Project Group:

Participants: Mathias Bülow Kastbjerg

Supervisors: Zheng-Hua Tan

Copies: 1

Page Numbers: 34

Date of Completion: June 7, 2018

Abstract:

In recent years the concept of Speaker De-Identification (SDI) has emerged. SDI handles the task of changing the speaker identity of a speech signal from a source speaker to a target speaker. Specifically SDI focuses on masking the identity of the source speaker. In (Hsu, Zhang, and Glass 2017) a Factorized Hierarchical Variational Autoencoder (FHVAE) was introduced for speech analysis. The FHVAE aims to factorize the speech signal into a linguistic part and a non-linguistic part. This factorization motivates the use of the FHVAE for SDI. The focus of this project is to investigate the performance of the FHVAE model when used for SDI. The model is compared to a baseline system based on a GMM mapping and a Harmonic plus Stochastic Model. The performance of the models is evaluated on two criteria: 1) Intelligibility, measured by an Automatic Speech Recognition system computing the Word Error Rate (WER). 2) How well the systems mask the identity of the source speaker, measured a speaker recognition system computing the Equal Error Rate (EER). Furthermore it is investigated whether a simpler metric to measure the intelligibility can be developed. The FH-VAE model showed good results on intelligibility compared to the baseline, but was found inferior on the de-identification task. The search for a metric to replace the WER as a measure of ineligibility was unsuccessful.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Resumé

Voice Transformation (VT) omhandler metoder til behandling af talesignaler, hvor det sproglige indhold af talesignalerne er uændret. VT kan have forskellige formål, så som ændring af følelserne i talesignalet, forbedring af forståeligheden, eller ændring af stemmen som den lyder som kom den fra en anden person. Det sidste refereres ofte til som Voice Conversion (VC), hvor det ikke-sproglige indhold af signalet behandles. Et andet formål af VT, der har vundet indpas i de seneste år er, Speaker De-Identification (SDI). SDI omhandler ligesom VC ændring af identiteten taleren, men hvor VC har fokus på den nye stemme, så har SDI fokus på at maskere identiteten af den originale taler. I begge metoder er der dog fokus på at bevare forståeligheden af talesignalerne.

Dette projekt undersøger hvorvidt en Factorized Hierarchical Variational Autoencoder (FHVAE) model kan bruges til SDI. FHVAE modellen er baseret på Variational Bayes (VB). VB opstår når sandsynligheds modellens marginale likelihood eller posterior sandsynligheden ikke kan evalueres. FHVAE modellen opsplitter talesignalet i to dele; en indeholdende det sproglige indhold og en indeholdende det ikke sproglige indhold. Det er denne opsplitning der er motivationen bag brugen af FHVAE modelen til SDI. FHVAE modelen sammenlignes med en basis model baseret på en Gaussisk Mixture Model og en harmonisk og stokastisk model.

Modellerne vurderes på to parameter; forståeligheden af talesignalet, samt hvorvidt det lykkes modellerne at maskere identiteten af taleren. Forståeligheden måles af et automatisk tale genkendelses system (Automatic Speech Recognition -ASR), der måler antallet af korrekte ord i det behandlede talesignal. Maskeringen af talerens identitet er målt af et taler genkendelses system (speaker recognition), der estimere identiteten af taleren i et talesignal. Projektet omhandler også en søgen efter et simplere mål for forståeligheden af talesignalet.

FHVAE modellen viste sig bedre end basis modellen i forhold til forståelighed, men den var ringere end basis modellen i forhold maskeringen af talerens identitet. Hvor identitaten af taleren i basis modellen var tættere på den nye taler identitet, var FHVAE modellen tættere på kilde taleren. Det lykkedes ej heller at finde et mål der kunne erstatte ASR systemet.

Authors' notes

This project has been completed by a tenth semester student from the Master of Science in Mathematical Engineering at Aalborg University. The report is made readable for others with a similar foundation. The prerequisites include familiarity with the topics machine learning, deep learning and speech processing.

The project has been under supervision of Professor Zheng-Hua Tan, from the Department of Electronic Systems. It has been written over a period of four months from February to May 2018.

References will be used throughout, and can be found in the bibliography at the end. Specific page numbers or sections may be mentioned. The files are available for replication of results, and are found at AAU project library, projekter.aau.dk. The models have been implemented in Kaldi, Python or Matlab.

Contents

1	Introduction1.1Problem Statement1.2Structure of the Report	1 2 2				
2	Basic Models 2.1 Autoencoder	3 3 4 4 4				
3	Variational Bayes 3.1 Kullback-Leibler Divergence 3.2 Stochastic Gradient Variational Bayes Estimator 3.3 Variational Autoencoder 3.3.1 Factorized Hierarchical Variational Autoencoder 3.3.2 Discriminative Objective 3.3.3 Model Architecture	7 7 8 10 10 13 13				
4	Speaker De-Identification4.1Data sets	15 15 15 16 17 18				
5	Experiments and Results 5.1 Intelligibility 5.1.1 Replacement metrics to the WER 5.2 De-Identification	21 21 22 24				
6	Discussion	27				
7	Conclusion and Future Work 7.1 Future Work	29 29				
Bi	Bibliography 31					

x

A Scripts and Files

Contents

33

Chapter 1 Introduction

Voice Transformation (VT) covers various methods of modifying one or more aspects of speech signals, while maintaining the linguistic content of the signal, (Mohammadi and Kain 2017). VT has many applications, such as changing the emotions in the signal, improving intelligibility, or converting the speakers voice into that of another speaker, (Mohammadi and Kain 2017). The latter application is called Voice Conversion (VC), specifically VC is a subclass of VT that aims at changing the speaker identity, by modifying the non- or paralinguistic information in the signal, while preserving the linguistic information of the speech signal. The transformation of the speaker identity from the source speaker to the target speaker is done by estimating or training a conversion function to convert a given utterance. There exists several different systems of VC, some are Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Weighted Frequency Warping (WFW) and Neural Networks (NN) (Machado and Queiroz 2010).

In recent years another application of VT has emerged, namely Speaker De-Identification (SDI). SDI is closely related to VC. Where VC focuses primarily on the target speaker, the focus of SDI is on masking the identity of the source speaker, while preserving the intelligibility of the speech signal. Because of the close relation to VC, the same models used for VC can be used for SDI. In (Pobar and Ipsić 2014) they propose an online algorithm for SDI, based on a GMM mapping and a Harmonic Stochastic Model. In (Abou-Zleikha et al. 2015) a speaker selection scheme is introduced, that minimizes the confidence that the speaker of the converted utterance is identified as the source.

VC- and SDI-systems can be grouped according to certain factors. One factor is whether they use parallel or non-parallel training data. In parallel speech corpus's, all speakers speak the same utterances. Non-parallel speech corpus's, on the other hand, does not have that restriction, which introduces more variability in the data. Another factor is whether the systems utilize the linguistic content or not, these are referred to as text-dependent or text-independent systems. Text-dependent systems often require a parallel speech corpus. Non-parallel training data and text-independent systems increases the number of situations the systems need to handle, and therefore require stronger and more complicated models. In (Hsu et al. 2016) a Variational Autoencoder (VAE) is introduced trained on a non-parallel speech corpus. The VAE is based on Variational Bayes (VB), that arises when the marginal likelihood or the posterior of the probabilistic model is intractable, (Kingma and Welling 2013). VB introduces an extension to the stchastic gradient estimator, which enables it to handle the intractabilities. In (Hsu, Zhang, and Glass 2017) a Factorized Hierarchical Variational Autoencoder (FHVAE) is introduced for speech analysis. The FHVAE aims at factorize the speech signal into a linguistic part and a non-linguistic part, the two parts can then be processed independently. This factorization motivates the use of the FHVAE for SDI.

1.1 Problem Statement

It is the purpose of this project to investigate the performance of the FHVAE model when used for SDI. The model is compared to a baseline system based on a GMM mapping and a Harmonic plus Stochastic Model, using the UPC voice conversion toolkit, (Eslava 2008). The performance of the models is evaluated according to two criteria; intelligibility and how well the systems mask the identity of the source speaker. The intelligibility is measured by developing an Automatic Speech Recognition (ASR) system for computing the Word Error Rate (WER). The deidentification is measured by developing a speaker recognition system to compute the Equal Error Rate (EER) relative to the source and target speakers, respectively. Furthermore constructing an ASR system to compute the WER is a comprehensive task, it is therefore investigated if a simpler measure can be constructed to measure the intelligibility of the converted speech.

1.2 Structure of the Report

In chapter 2 some basic models and concepts of machine learning are introduced. Chapter 3 introduces the framework of variational Bayes. Chapter 4 introduces the data set used, the SDI process of the baseline model and the FHVAE model, the ASR system and the speaker recognition system. In chapter 5 the results are presented. The report is concluded with a discussion in chapter 6 and a conclusion and future work in chapter 7. Finally in appendix A the scripts used in this project are presented.

Chapter 2 Basic Models

This chapter presents a short introduction of some of the different neural network architectures used in this project.

2.1 Autoencoder

The autoencoder is a feed-forward neural network, that aims at replicating or copying its input to its output. The autoencoder has two parts; an encoder and a decoder. The encoder is a function, f, that maps the input, x, into a hidden representation, z. The decoder is a function, g, that maps the hidden representation into a reconstruction of the input, \hat{x} . In modern autoencoders the deterministic mappings of the encoder and decoder has been generalized to stochastic mappings $p_{enc}(z|x)$ and $p_{dec}(\hat{x}|z)$. The concept of an autoencoder is shown in figure 2.1. The aim of the autoencoder is to store only the most relevant information about

$$f(\mathbf{x}) \qquad g(\mathbf{z})$$

$$\mathbf{x} \longrightarrow \boxed{\text{Encoder}} \longrightarrow \mathbf{z} \longrightarrow \boxed{\text{Decoder}} \longrightarrow \hat{\mathbf{x}}$$

$$p_{enc}(\mathbf{z}|\mathbf{x}) \qquad p_{dec}(\hat{\mathbf{x}}|\mathbf{z})$$

Figure 2.1: The structure of the autoencoder.

the input in the hidden representation. To achieve this various restrictions can be imposed on the autoencoder. These restrictions prevents the autoencoder from achieving perfect reconstruction, that is where x = g(f(x)). If the dimension of z is stricktly less than the dimension of x, then the autoencoder is called undercomplete. In this way the autoencoder is forced to learn only the most important or discriminative features of the data. However if the functions f and g are both non-linear and they have too great capacity, the autoencoder can then still learn to do perfect reconstruction. Autoencoder are usually trained by gradient descent computed by back-propagation. The learning objective is to minimize a loss function L(x, g(f(x))), that penalizes g(f(x)) when it is dissimilar from x. Examples of loss functions are Mean Squarred Error (MSE) and binary crossentropy. If the loss function is the MSE and the decoder is linear, then the autoencoder will learn to span the same subspace as principal component analysis, (Goodfellow, Bengio, and Courville 2016, chapter 14).

2.2 Generative and Discriminative Models

Let x and y be random variables, where x is the observable variable and y is the target variable. Then different models can be distinguished as two different kinds of models; generative and discriminative. A generative model is a model based on the joint probability of the two variables, p(x, y), or the conditional probability of the observed variable given the target variable, p(x|y). A discriminative model is a model based on the conditional probability of the target variable, p(x|y). A discriminative model is a model based on the conditional probability of the target variable given the observed variable, p(y|x). The terminology arises since a generative model can 'generate' synthetic data, either of pairs of the observed and target variables, or of the observed variable given the target variable. The discriminative model, on the other hand, can discriminate target values given observations. Discriminative model can be used to recognize or infer target values given the observations.

Consider again the autoencoder in figure 2.1. Often the variable of interest is z, as it often is a compressed version of the most relevant information in the data. The autoencoder can then be used to make features from the data for other systems. The autoencoder can therefore be seen as being comprised of two models; the encoder is a discriminative model and the decoder is a generative model.

2.3 Recurrent Neural Networks

Recurrent neural networks (RNN) is a class of neural networks that has cyclic connections between nodes. The cyclic connections feed information from previous time-steps to the current time-step, this allows the network to capture temporal behavior in time-sequences. A simple RNN is seen in figure 2.2. There exist several different RNN architectures, one of which is Long Short Term Memory (LSTM).



Figure 2.2: A graphical model of a simple RNN, where *x* is the input to the network, *h* is the output of the network and *c* is the intermediate variable between the time-steps.

2.3.1 Long Short Term Memory

The LSTM unit was first proposed by (Hochreiter and Schmidhuber 1997), it has three layers; input layer, hidden layer and output layer. The hidden layer contains a memory cell, where the input and output are controlled through gates. Let x_t denote the input to the unit at time t and let h_t denote the output of the unit. The unit has two gates; the input gate i_t and the output gate o_t . Furthermore the unit

2.3. Recurrent Neural Networks

has a state vector c_t , that contains the current state of the unit. The unit is described as follows

$$i_t = \sigma_g(W_i x_t + U_i h_t + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_t + b_o)$$

$$c_t = i_t \circ \sigma_c(W_c x_t + U_c h_t + b_c)$$

$$h_t = o_t \circ \sigma_h(c_t).$$

where the *W*'s and *U*'s are weight matrices, the *b*'s are bias vectors, σ_g is the sigmoid function, σ_c and σ_h are the hyperbolic tangent and \circ denotes the Hadamard product. The unit is initialized with $c_0 = 0$ and $h_0 = 0$.

The model was improved in (Gers, Schmidhuber, and Cummins 1999), where it was found that the internal state vector could grow uncontrollably under certain conditions. In order to control the internal state vector, a forget gate, f_t , was introduced. Its purpose is to reset the internal state vector. The unit is then described as

$$i_t = \sigma_g(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_t + \mathbf{b}_i)$$

$$o_t = \sigma_g(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_t + \mathbf{b}_o)$$

$$f_t = \sigma_g(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_t + \mathbf{b}_f)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_t + \mathbf{b}_c)$$

$$h_t = o_t \circ \sigma_h(c_t).$$

The graphical structure of the unit is shown in figure 2.3.



Figure 2.3: The structure of the LSTM unit.

Chapter 3 Variational Bayes

Let $X = \{x^{(i)}\}_{i=1}^{N}$ be a data set consisting of N i.i.d. samples of a discrete or continuous random variable x. It is assumed that the data is generated by a random process involving an unobserved continuous random variable z. The unobserved variable z is drawn from a prior distribution $p_{\theta}(z)$, then x is drawn from a conditional distribution $p_{\theta}(x|z)$. It is assumed that $p_{\theta}(z)$ and $p_{\theta}(x|z)$ are differentiable almost everywhere w.r.t. θ and z. Most of this process is hidden; both the true parameters θ and the values of z are unknown.

Now consider cases where the integral of the marginal likelihood $p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz$ is intractable or where the true posterior $p_{\theta}(z|x) = \frac{p_{\theta}(z)p_{\theta}(x|z)}{p_{\theta}(x)}$ is intractable. In the first case it is not possible to evaluate or differentiate the marginal likelihood, preventing approximate marginal inference of the variable x. In the second case it is not possible to use the EM-algorithm to find the maximum likelihood (ML) or maximum a posteriori (MAP) estimate of the parameters θ . These intractabilities are common in moderately complicated likelihood functions $p_{\theta}(x|z)$, like neural networks with a nonlinear hidden layer (Kingma and Welling 2013).

3.1 Kullback-Leibler Divergence

Before proceeding, the Kullback-Leibler divergence is introduced:

Definition 3.1 (Kullback-Leibler Divergence)

Let *f* and *g* be two density functions. Then the Kullback-Leibler divergence from *f* to *g* is defined as

$$D(f||g) = \int f \log \frac{f}{g}.$$

Note that $D(f||g) = E_f \left[\log \frac{f}{g} \right]$, note also that D(f||g) is not necessarily the same as D(g||f). It can be proven that $D(f||g) \ge 0$, see for instance (Cover and Thomas 2006).

3.2 Stochastic Gradient Variational Bayes Estimator

To efficiently solve the problems of intractability, variational bayes is used. Let the recognition model, given by $q_{\phi}(z|x)$, be an estimate of the true posterior $p_{\theta}(z|x)$. The objective is then to jointly learn the parameters, ϕ , of the recognition model and the parameters, θ , of the generative model. Since the data are i.i.d. the marginal likelihood can be written as

$$\log p_{\theta}(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_{\theta}(\mathbf{x}^{(i)}).$$
(3.1)

Now log $p_{\theta}(\mathbf{x}^{(i)})$ can be rewritten as

$$\begin{split} \log p_{\theta}(\mathbf{x}^{(i)}) &= E_{q_{\phi}(z|\mathbf{x}^{(i)})} \left[\log p_{\theta}(\mathbf{x}^{(i)}) \right] \\ &= \int q_{\phi}(z|\mathbf{x}^{(i)}) \log \frac{p_{\theta}(\mathbf{x}^{(i)}, z)}{p_{\theta}(z|\mathbf{x}^{(i)})} dz \\ &= \int q_{\phi}(z|\mathbf{x}^{(i)}) \left[\log \frac{q_{\phi}(z|\mathbf{x}^{(i)})}{p_{\theta}(z|\mathbf{x}^{(i)})} - \log q_{\phi}(z|\mathbf{x}^{(i)}) + \log p_{\theta}(\mathbf{x}^{(i)}, z) \right] dz \\ &= \int q_{\phi}(z|\mathbf{x}^{(i)}) \log \frac{q_{\phi}(z|\mathbf{x}^{(i)})}{p_{\theta}(z|\mathbf{x}^{(i)})} dz \\ &+ \int q_{\phi}(z|\mathbf{x}^{(i)}) \left[-\log q_{\phi}(z|\mathbf{x}^{(i)}) + \log p_{\theta}(\mathbf{x}^{(i)}, z) \right] dz \\ &= D(q_{\phi}(z|\mathbf{x}^{(i)}) || p_{\theta}(z|\mathbf{x}^{(i)})) + \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}), \end{split}$$
(3.2)

where

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = E_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \left[-\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}) \right].$$
(3.3)

Since the KL-divergence is non-negative then $\log p_{\theta}(\mathbf{x}^{(i)}) \ge \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$, therefore $\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$ is called the variational lower bound on the marginal likelihood of data point *i*. Rewriting (3.3) gives

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = \int q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)}) \left[-\log q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}) \right] d\boldsymbol{z}$$

$$= \int q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)}) \left[-\log q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] d\boldsymbol{z}$$

$$= \int q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)}) \left[-\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] d\boldsymbol{z}$$

$$= -D(q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x}^{(i)}) | p_{\boldsymbol{\theta}}(\boldsymbol{z})) + E_{q_{\boldsymbol{\phi}}(\boldsymbol{z} | \boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right].$$
(3.4)

Using Monte Carlo Integration a gradient estimator of (3.3) is

$$\nabla_{\phi} E_{q_{\phi}(z)}[f(z)] = E_{q_{\phi}(z)}[f(z)\nabla_{\phi}\log q_{\phi}(z)] \simeq \frac{1}{L} \sum_{l=1}^{L} f(z^{(l)})\nabla_{\phi}\log q_{\phi}(z^{(l)}), \quad (3.5)$$

where $f(z) = -\log q_{\phi}(z|x^{(i)}) + \log p_{\theta}(x^{(i)}, z)$ and $z^{(l)} \sim q_{\phi}(z|x^{(i)})$. However this gradient estimator exhibits very high variance and is impractical to use, (Kingma and Welling 2013). In order to construct a better estimator the reparametirization trick is used.

Theorem 3.1 (Reparameterization Trick)

Let z be a continuous random variable, let $z \sim q_{\phi}(z|x)$ be some conditional distribution, and let f(z) be some differential function. If it is possible to express z as a deterministic variable by $z = g_{\phi}(\epsilon, x)$, where ϵ is an auxiliary variable with independent marginal $p(\epsilon)$ and g_{ϕ} is some vector-valued function parameterized by ϕ . Then

$$E_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[f(\boldsymbol{z})] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \boldsymbol{x})), \qquad (3.6)$$

where $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}).$

Proof Given the deterministic mapping $z = g_{\phi}(\epsilon, x)$, it is known that the probability contained in a differential area must be invariant under change of variable, that is

$$q_{\phi}(z|x)dz = p(\epsilon)d\epsilon.$$
(3.7)

It follows that

$$E_{q_{\phi}(z|\mathbf{x})}[f(z)] = \int q_{\phi}(z|\mathbf{x})f(z)dz$$

= $\int p(\epsilon)f(z)d\epsilon$
= $\int p(\epsilon)f(g_{\phi}(\epsilon, \mathbf{x}))d\epsilon$
= $E_{p(\epsilon)}[f(g_{\phi}(\epsilon, \mathbf{x}))].$ (3.8)

Using Monte Carlo Integration, a differential estimator of (3.8) can be formed;

$$E_{p(\boldsymbol{\epsilon})}[f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{x}))] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)},\boldsymbol{x})), \qquad (3.9)$$

where $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$.

Applying Theorem 3.1 on the variational lower bound in (3.3), with $f(z) = -\log q_{\phi}(z|\mathbf{x}^{(i)}) + \log p_{\theta}(\mathbf{x}^{(i)}, z)$, gives the Stochastic Gradient Variational Bayes (SGVB) estimator $\widetilde{\mathcal{L}}^{A}(\theta, \phi; \mathbf{x}^{(i)}) \simeq \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$, given by

$$\widetilde{\mathcal{L}}^{A}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) \simeq \frac{1}{L} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i,l)}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}^{(i,l)} | \boldsymbol{x}^{(i)})$$
(3.10)

where $z^{(i,l)} = g_{\phi}(\epsilon^{(l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$.

If the KL-divergence in (3.4) can be evaluated analytically, then applying Theorem 3.1 on the expectation in (3.4), with $f(z) = \log p_{\theta}(x^{(i)}|z)$, gives the SGVB estimator $\widetilde{\mathcal{L}}^{B}(\theta, \phi; x^{(i)}) \simeq \mathcal{L}(\theta, \phi; x^{(i)})$, given by

$$\widetilde{\mathcal{L}}^{B}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}^{(i)}) \simeq -D(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})|p_{\boldsymbol{\theta}}(\boldsymbol{z})) + \frac{1}{L}\sum_{l=1}^{L}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)})$$
(3.11)

where $\mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)})$ and $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$. The derivatives can be taken of $\widetilde{\mathcal{L}}^{A}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ and $\widetilde{\mathcal{L}}^{B}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and they can be used with stochastic optimization algorithms like Stochastic Gradient Descent or Adagrad, (Kingma and Welling 2013).

3.3 Variational Autoencoder

Variational Bayes can also be used to construct an autoencoder, referred to as a Variational Autoencoder (VAE). The recognition model $q_{\phi}(z|x)$ can be seen as a probabilistic encoder and the generative model $p_{\theta}(x|z)$ can be seen as a probabilistic decoder. Considering the estimator in (3.11), a sample of z is generated by $z^{(i,l)} = g_{\phi}(\epsilon^{(l)}, x^{(i)})$, which is then fed to the generative model. If $\log p_{\theta}(x|z) = 0$, then the reconstruction error is minimized, therefore the expectated value in (3.11) is the expected negative reconstruction error and the KL-divergence acts as a regularizer.

3.3.1 Factorized Hierarchical Variational Autoencoder

In this subsection the Factorized Hierarchical Variational Autoencoder (FHVAE) proposed in (Hsu, Zhang, and Glass 2017) is presented. Speech data is complicated to model, since it is affected by multiple factors like fundamental frequency (f_0), volume, speaker id and phonetical content. Speaker id affects f_0 , dialect and volume. Let a sequence refer to a sub-sequence of an utterance and let a segment refer to a variable of smaller temporal scale than a sequence. Then factors like f_0 , dialect and volume would tend to have a larger variance across sequences than within sequences, whereas phonetical content would tend to have a similar variance across and within sequences. This motivates the division of the factors or attributes into sequence-level attributes and segment-level attributes, that is attributes affected by long-term- and short-term statistical content, respectively. This motivates developing a FHVAE to model the speech data.

3.3. Variational Autoencoder

Let $\mathcal{D} = \{\mathbf{X}^{(i)}\}_{i=1}^{M}$ be a data set of M i.i.d. sequences $\mathbf{X}^{(i)}$ and let each sequence be $\mathbf{X}^{(i)} = \{\mathbf{x}^{(i,n)}\}_{n=1}^{N^{(i)}}$, where $\mathbf{x}^{(i,n)}$ is a segment and $N^{(i)}$ is the number of segments. Each sequence is assumed to be generated from three latent variables, $\mathbf{Z}_{1}^{(i)} = \{\mathbf{z}_{1}^{(i,n)}\}_{n=1}^{N^{(i)}}, \mathbf{Z}_{2}^{(i)} = \{\mathbf{z}_{2}^{(i,n)}\}_{n=1}^{N^{(i)}}$ and $\boldsymbol{\mu}_{2}^{(i)}$. The $Z_{1}^{(i)}$'s are latent segment variables, the $Z_{2}^{(i)}$'s are latent sequence variables and $\boldsymbol{\mu}_{2}^{(i)}$ is the mean vector of $Z_{2}^{(i)}$. The generative process is divided into three steps;

- 1. $\mu_2^{(i)}$ is drawn from a prior distribution $p_{\theta}(\mu_2)$.
- 2. $N^{(i)}$ i.i.d. latent segment variables $\{z_1^{(i,n)}\}_{n=1}^{N^{(i)}}$ and latent sequence variables $\{z_2^{(i,n)}\}_{n=1}^{N^{(i)}}$ are drawn from a sequence independent prior $p_{\theta}(z_1)$ and a sequence dependent prior $p_{\theta}(z_2|\mu_2)$.
- 3. $N^{(i)}$ i.i.d. segments $\{x^{(i,n)}\}_{n=1}^{N^{(i)}}$ are drawn from a conditional distribution $p_{\theta}(x|z_1, z_2)$.

It is assumed that Z_1 and Z_2 are independent. The generative model is parametrized by θ . The joint probability for each sequence can be factorized as follows

$$p_{\theta}(\mathbf{X}^{(i)}, \mathbf{Z}_{1}^{(i)}, \mathbf{Z}_{2}^{(i)}, \boldsymbol{\mu}_{2}^{(i)}) = p_{\theta}(\boldsymbol{\mu}_{2}^{(i)}) p_{\theta}(\mathbf{X}^{(i)}, \mathbf{Z}_{1}^{(i)}, \mathbf{Z}_{2}^{(i)} | \boldsymbol{\mu}_{2}^{(i)})$$

$$= p_{\theta}(\boldsymbol{\mu}_{2}^{(i)}) \prod_{n=1}^{N^{(i)}} p_{\theta}(\mathbf{x}^{(i,n)}, \mathbf{z}_{1}^{(i,n)}, \mathbf{z}_{2}^{(i,n)} | \boldsymbol{\mu}_{2}^{(i)})$$

$$= p_{\theta}(\boldsymbol{\mu}_{2}^{(i)}) \prod_{n=1}^{N^{(i)}} p_{\theta}(\mathbf{x}^{(i,n)}, \mathbf{z}_{1}^{(i,n)} | \mathbf{z}_{2}^{(i,n)}) p_{\theta}(\mathbf{z}_{2}^{(i,n)} | \boldsymbol{\mu}_{2}^{(i)})$$

$$= p_{\theta}(\boldsymbol{\mu}_{2}^{(i)}) \prod_{n=1}^{N^{(i)}} p_{\theta}(\mathbf{x}^{(i,n)} | \mathbf{z}_{1}^{(i,n)}, \mathbf{z}_{2}^{(i,n)}) p_{\theta}(\mathbf{z}_{1}^{(i,n)}) p_{\theta}(\mathbf{z}_{2}^{(i,n)} | \boldsymbol{\mu}_{2}^{(i)}),$$
(3.12)

where the density functions are assumed to be

$$p_{\theta}(\mathbf{x}|z_{1}, z_{2}) = \mathcal{N}(\mathbf{x}|f_{\mu_{x}}(z_{1}, z_{2}), \operatorname{diag}(f_{\sigma_{x}^{2}}(z_{1}, z_{2}))),$$

$$p_{\theta}(z_{1}) = \mathcal{N}(z_{1}|\mathbf{0}, \sigma_{z_{1}}^{2}\mathbf{I}),$$

$$p_{\theta}(z_{2}|\mu_{2}) = \mathcal{N}(z_{2}|\mu_{2}, \sigma_{z_{2}}^{2}\mathbf{I}),$$

$$p_{\theta}(\mu_{2}) = \mathcal{N}(\mu_{2}|\mathbf{0}, \sigma_{\mu_{2}}^{2}\mathbf{I}).$$

The functions $f_{\mu_x}(z_1, z_2)$ and $f_{\sigma_x^2}(z_1, z_2)$ are neural networks modeling the mean and diagonal co-variance, respectively. From the factorization of the generative model, it is seen that the z_2 's within a sequence are forced to be close to $\mu_2^{(i)}$ as well as each other, which encourages encoding of the sequence attributes. The z_1 's on the other hand have a global constraint, which encourages encoding of the segment attributes. In this setup the true posterior $p_{\theta}(\mathbf{Z}_1^{(i)}, \mathbf{Z}_2^{(i)}, \boldsymbol{\mu}_2^{(i)} | \mathbf{X}^{(i)})$ is intractable, an estimate $q_{\phi}(\mathbf{Z}_1^{(i)}, \mathbf{Z}_2^{(i)}, \boldsymbol{\mu}_2^{(i)} | \mathbf{X}^{(i)})$ is introduced as the inference model, which can be factorized as follows

$$q_{\phi}(\mathbf{Z}_{1}^{(i)}, \mathbf{Z}_{2}^{(i)}, \boldsymbol{\mu}_{2}^{(i)} | \mathbf{X}^{(i)}) = q_{\phi}(\boldsymbol{\mu}_{2}^{(i)}) q_{\phi}(\mathbf{Z}_{1}^{(i)}, \mathbf{Z}_{2}^{(i)} | \mathbf{X}^{(i)})$$

$$= q_{\phi}(\boldsymbol{\mu}_{2}^{(i)}) \prod_{n=1}^{N^{(i)}} q_{\phi}(\boldsymbol{z}_{1}^{(i,n)}, \boldsymbol{z}_{2}^{(i,n)} | \boldsymbol{x}^{(i,n)})$$

$$= q_{\phi}(\boldsymbol{\mu}_{2}^{(i)}) \prod_{n=1}^{N^{(i)}} q_{\phi}(\boldsymbol{z}_{1}^{(i,n)} | \boldsymbol{x}^{(i,n)}, \boldsymbol{z}_{2}^{(i,n)}) q_{\phi}(\boldsymbol{z}_{2}^{(i,n)} | \boldsymbol{x}^{(i,n)}), \quad (3.13)$$

where the posteriors are assumed to be

$$q_{\phi}(\mu_{2}^{(i)}) = \mathcal{N}(\mu_{2}^{(i)}|g_{\mu_{2}}(i), \sigma_{\mu_{2}}^{2}\mathbf{I}),$$

$$q_{\phi}(z_{2}|\mathbf{x}) = \mathcal{N}(z_{2}|g_{\mu_{z_{2}}}(\mathbf{x}), \operatorname{diag}(g_{\sigma_{z_{2}}^{2}}(\mathbf{x}))),$$

$$q_{\phi}(z_{1}|\mathbf{x}, z_{2}) = \mathcal{N}(z_{1}|g_{\mu_{z_{1}}}(\mathbf{x}, z_{2}), \operatorname{diag}(g_{\sigma_{z_{1}}^{2}}(\mathbf{x}, z_{2}))).$$

For the posterior $q_{\phi}(\mu_2^{(i)})$ the variance is fixed and the mean, $g_{\mu_2}(i)$, is not directly inferred from x, instead it is seen as a part of the model parameters with one for each utterance. The mean function $g_{\mu_2}(i)$ can be seen as a lookup table, with a mean for each sequence and is optimized during training. The functions $g_{\mu_{z_2}}(x)$, $g_{\sigma_{z_2}^2}(x)$, $g_{\mu_{z_1}}(x, z_2)$ and $g_{\sigma_{z_1}^2}(x, z_2)$ are neural networks for the means and diagonal co-variances of z_1 and z_2 , respectively. The inference model is parametrized by ϕ .

Using (3.3) the variational lower bound is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = E_{q_{\boldsymbol{\phi}}(\boldsymbol{Z}_{1}^{(i)}, \boldsymbol{Z}_{2}^{(i)}, \boldsymbol{\mu}_{2}^{(i)} | \boldsymbol{X}^{(i)})} \left[-\log q_{\boldsymbol{\phi}}(\boldsymbol{Z}_{1}^{(i)}, \boldsymbol{Z}_{2}^{(i)}, \boldsymbol{\mu}_{2}^{(i)} | \boldsymbol{X}^{(i)}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(i)}, \boldsymbol{Z}_{1}^{(i)}, \boldsymbol{Z}_{2}^{(i)}, \boldsymbol{\mu}_{2}^{(i)}) \right].$$
(3.14)

The variational lower bound in (3.14) can be rewritten to, (Hsu, Zhang, and Glass 2017, Appendix A)

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}^{(i)}) = \sum_{n=1}^{N^{(i)}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i,n)}), \qquad (3.15)$$

where

$$\begin{split} \mathcal{L}(\theta, \phi; \mathbf{x}^{(i,n)}) &= E_{q_{\phi}(z_{1}^{(i,n)}, z_{2}^{(i,n)} | \mathbf{x}^{(i,n)})} \left[\log p_{\theta}(\mathbf{x}^{(i,n)} | z_{1}^{(i,n)}, z_{2}^{(i,n)}) \right] \\ &\quad - E_{q_{\phi}(z_{2}^{(i,n)} | \mathbf{x}^{(i,n)})} \left[D(q_{\phi}(z_{1}^{(i,n)} | \mathbf{x}^{(i,n)}, z_{2}^{(i,n)}) || \log p_{\theta}(z_{1}^{(i,n)}) \right] \\ &\quad - D(q_{\phi}(z_{2}^{(i,n)} | \mathbf{x}^{(i,n)}) || \log p_{\theta}(z_{2}^{(i,n)} | \widetilde{\mu}_{2}) \\ &\quad + \frac{1}{N^{(i)}} \log p_{\theta}(\widetilde{\mu}_{2}) + \text{const,} \end{split}$$

and where $\tilde{\mu}_2 = g_{\mu_2}(i)$.

3.3. Variational Autoencoder

3.3.2 Discriminative Objective

In the generative model the prior for z_2 is conditioned on μ_2 to encourage the encoding of segment- and sequence-level attributes into different latent variables. However the prior probability for μ_2 is maximized when $\mu_2 = 0$ for all sequences, resulting in trivial mean vectors. Furthermore z_2 is inferred from the KL-divergence $D(q_{\phi}(z_2^{(i,n)}|\mathbf{x}^{(i,n)})||\log p_{\theta}(z_2^{(i,n)}|\tilde{\mu}_2))$, measured to the same conditional prior for all sequences. This would then result in z_1 and z_2 not being factorized into segment- and sequence-level attributes, respectively. Therefore the following discriminative objective is formulated

$$\log p(i|\mathbf{z}^{(i,n)}) = \log p(\mathbf{z}^{(i,n)}|i) - \log \left(\sum_{j=1}^{M} p(\mathbf{z}^{(i,n)}|j)\right)$$

:= log $p(\mathbf{z}^{(i,n)}|\widetilde{\mu}_{2}^{(i)}) - \log \left(\sum_{j=1}^{M} p(\mathbf{z}^{(i,n)}|\widetilde{\mu}_{2}^{(j)})\right)$

where p(i) is assumed uniform. Combining the discriminative objective with the variational lower bound gives

$$\mathcal{L}^{\text{dis}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i,n)}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i,n)}) + \alpha \log p(i|\boldsymbol{z}^{(i,n)}), \qquad (3.16)$$

where α is a weighting parameter. This is referred to as the discriminative variational lower bound.

3.3.3 Model Architecture

Let the sub-sequence, $x_{1:T}$, of X be a segment, that contains T time-steps, denoted x_t . To capture the temporal information in the segment, an RNN architecture is adopted. The FHVAE-model is build using LSTM and Multilayer Perceptron (MLP) networks. The recognition model is formulated as

$$(h_{z_{2},t}, c_{z_{2},t}) = \text{LSTM}(x_{t-1}, h_{z_{2},t-1}, c_{z_{2},t-1}; \phi_{\text{LSTM},z_{2}})$$

$$q_{\phi}(z_{2}|x_{1:T}) = \mathcal{N}(z_{2}|\text{MLP}(h_{z_{2},T}; \phi_{\text{MLP}_{\mu},z_{2}}), \text{diag}(\exp(\text{MLP}(h_{z_{2},T}; \phi_{\text{MLP}_{\sigma^{2}},z_{2}}))))$$

$$(h_{z_{1},t}, c_{z_{1},t}) = \text{LSTM}([x_{t-1}; z_{2}], h_{z_{1},t-1}, c_{z_{1},t-1}; \phi_{\text{LSTM},z_{1}})$$

$$q_{\phi}(z_{1}|x_{1:T}, z_{2}) = \mathcal{N}(z_{1}|\text{MLP}(h_{z_{1},T}; \phi_{\text{MLP}_{\mu},z_{1}}), \text{diag}(\exp(\text{MLP}(h_{z_{1},T}; \phi_{\text{MLP}_{x^{2}},z_{1}})))),$$

and is shown in figure 3.1. The generative model is formulated as

$$(\boldsymbol{h}_{x,t}, \boldsymbol{c}_{x,t}) = \text{LSTM}([\boldsymbol{z}_1; \boldsymbol{z}_1], \boldsymbol{h}_{x,t-1}, \boldsymbol{c}_{x,t-1}; \boldsymbol{\theta}_{\text{LSTM},x})$$
$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{z}_1, \boldsymbol{z}_2) = \mathcal{N}(\boldsymbol{x}_t | \text{MLP}(\boldsymbol{h}_{x,t}; \boldsymbol{\theta}_{\text{MLP}_{u,x}}), \text{diag}(\exp(\text{MLP}(\boldsymbol{h}_{x,t}; \boldsymbol{\theta}_{\text{MLP}_{u,2},x})))),$$

and is shown in figure 3.2. The optimization is done over a maximum of 500 epochs using the Adam optimizer, (Kingma and Ba 2014), with a patience of 50 epochs. The dimension of the hidden layers h_{z_1} , h_{z_2} and h_x is set to 256.





Figure 3.2: The structure of the generative model of the FHVAE model.

Chapter 4 Speaker De-Identification

This chapter presents the data sets used, the speaker de-identification method of the baseline and the FHVAE models and the methods for evaluation of the performance of the models.

4.1 Data sets

The database used in this project is the TIMIT speech corpus, (Garofolo 1993). The corpus consists of 630 speakers from 8 dialect regions in USA, each reading ten phonetically rich sentences. The sentences are divided into three groups; dialect (SA), compact (SX) and diverse (SI). The SA group consists of two sentences designed to reveal the dialect of the speakers. The SX group consists of 450 phonetically compact sentences, each speaker reads 5 sentences from this group and each sentence is read by 7 speakers. The SI group consists of 1890 phonetically diverse sentences, each speaker reads 3 sentences from this group and each sentence is only read by one speaker. In this study only the SI and SX sentences are used, unless otherwise noted. The corpus has 70% male and 30% female speakers.

The TIMIT speech corpus is divided into four subsets; training set, development set, core test set and test set. The training set consists of 462 speakers, which totals to 3696 utterances. The development set consists of 50 speakers (400 utterances). The core test set consists of 24 speakers (192 utterances). The test set consists of 94 speakers (752 utterances). There is no overlap between the sets. Unless otherwise noted each model is trained on the training set and the development set is used for validation during training. The models are then tested on the core test set. The test set is used to evaluate the performance of the baseline model and FHVAE model on the SDI task.

4.2 Baseline Model

The UPC voice conversion toolkit, (Eslava 2008), is used as the baseline model. The toolkit is based on a harmonic plus stochastic model, that decomposes the speech signal into a harmonic (deterministic) part and a stochastic part. The model assumes that the speech signal is locally composed of a sum of sinusoids with timevarying parameters. Furthermore the model assumes that the sinusoids are integer multiples of the local fundamental frequency, f_0 . The frequencies, amplitudes and phases of the harmonic components are extracted from the signal on a frame-by-frame level. By interpolating and regenerating the deterministic component, the stochastic part is found by subtracting the deterministic from the original speech signal. The stochastic part of the signal deals with the aperiodic components of the signal, that are not well represented by sinusoids. The stochastic part of the signal is analysed frame-by-frame using linear predictive coding.

The speaker de-identification is done, by training a Gaussian Mixture Model (GMM) on the harmonic components. The GMM is then used as a transformation function, to convert the harmonic components. The stochastic components are then estimated from the converted harmonic components. The pitch is adapted by linear transformations of the mean and variance of log f_0 . A GMM-model is trained for each source- and target speaker pair. For further details, see (Eslava 2008).

The UPC voice conversion toolkit supports both parallel and non-parallel data sets. Here a non-parallel implementation is used. In order to have sufficient data the SA utterances are also used. For each source speaker pair in the test set, 9 utterances from both the source and target speaker are used to train the GMM, the last utterance from each speaker is used for voice conversion. The utterances used for conversion are all from the SX group. There are a total of $94 \cdot 93 = 8742$ source-target pairs, with one converted utterance for each source-target pair there are 8742 utterances of the baseline model.

4.3 FHVAE Model

In the FHVAE Model the dimension of the latent variables z_1 and z_2 is set 32 and the weighting parameter of the discriminative objective is $\alpha = 10$. The input to the FHVAE Model is a 200 dimensional log-magnitude spectrogram of the speech signal, computed every 10ms. The SDI is done by modifying the latent sequence variable. The latent sequence variable is modified according to the following formula:

$$\hat{z}_{2,t} = z_{2,s} - \mu_{2,s} + \mu_{2,t}, \tag{4.1}$$

where $z_{2,s}$ is the latent sequence variable of the source speaker, $\mu_{2,s}$ and $\mu_{2,t}$ are the means of latent sequence variables of the source and target speaker, respectively, and $\hat{z}_{2,t}$ is the estimated latent sequence variable of the target speaker. $\hat{z}_{2,t}$ is then combined with the latent segment variable of the source speaker, $z_{1,s}$, to obtain the converted log-magnitude spectrogram. The phase is then estimated from the converted log-magnitude spectrogram, and the output speech signal is computed. For each source-target pair 8 utterances are converted, generating a total of $8742 \cdot 8 = 69936$ utterances.

Examples of the converted speech from both the baseline model and the FHVAE model are made available on dropbox¹. The performance of the models on the SDI task are evaluated using speech recognition and speaker verification, presented next.

4.4 Speech Recognition

Speech Recognition, or Automatic Speech Recognition (ASR), is a term covering the methods of recognizing and translating spoken language into text. There exist a variety of methods, e.g. Hidden Markov Models (HMM's), GMM's or Deep Neural Networks (DNN's). The accuracy of the systems are evaluated by the Word Error Rate (WER). Let N be the total number of words in the reference transcript and let Q be the number of incorrect words in the generated transcript, then the WER is defined as

$$WER = \frac{Q}{N}$$

An incorrect word can occur in three different ways; substitution, deletion and insertion. Substitutions are when a word in the reference transcript is substituted by another word in the generated transcript, for instance if 'hello world' becomes 'hollow world'. Deletions are when a word in the reference transcript is is not present the generated transcript, for instance if 'I am old' becomes 'I old'. Insertion are when there is an extra word in the generated transcript, that was not present in the reference transcript, for instance if 'I am old' becomes 'I am too old'. The WER can then be rewritten as

$$WER = \frac{Q}{N} = \frac{S+D+I}{S+D+C'}$$
(4.2)

where *C* is the number of correct words.

The ASR system used in this project is a DNN developed for the TIMIT Speech Corpus in the Kaldi toolkit, (Povey et al. 2011). The system is designed to be able to recognize 39 different phonemes. The system is comprised of three types of models; a monophone model, a triphone model and a neural network. The monophone and triphone models are both based on GMM's. The monophone model is an acoustic model that computes the acoustic parameters of each phoneme, wihtout any contextual information. The triphone model uses the monophone model to represents each phoneme in the context of the preceding and succeeding phoneme. Both the monophone model and the triphone model takes 12 dimensional Mel-Frequency Cepstral Coefficients (MFCC) of the data as input. The triphone model is then used to extract the features for the neural network. The neural network has two hidden layers, both with tanh as the activation function.

¹The examples are available at https://www.dropbox.com/sh/nr94jcs6z3mi0vd/ AADYWBOu8OdyIsstogi49o7Ga?dl=0.

4.5 Speaker Recognition

Speaker Recognition, is a term covering methods for identifying or verifying the identity of the speaker of a speech signal. Like ASR common methods include HMM's, GMM's or DNN's. The speaker recognition system used in this project is a supervised GMM initialized from the posteriors of the DNN used for ASR, proposed in (Snyder, Garcia-Romero, and Povey 2015). The mixture weights w_k , means μ_k and co-variances S_k are initialized as follows

$$\begin{split} z_{k}^{(i)} &= p(k|\boldsymbol{y}_{i}, \boldsymbol{\theta}), \\ w_{k} &= \sum_{i=1}^{N} z_{k}^{(i)}, \\ \boldsymbol{\mu}_{k} &= \frac{1}{w_{k}} \sum_{i=1}^{N} z_{k}^{(i)} \boldsymbol{x}_{i}, \\ \boldsymbol{S}_{k} &= \frac{1}{w_{k}} \sum_{i=1}^{N} z_{k}^{(i)} (\boldsymbol{x}_{i} - \boldsymbol{\mu}_{k}) (\boldsymbol{x}_{i} - \boldsymbol{\mu}_{k})^{T}, \end{split}$$

where $p(k|y_i, \theta)$ is the probability of the *k*'th triphone at frame *i* given DNN features y_i and DNN parameters θ and x_i are the corresponding speaker recognition features. The speaker recognition system takes 20 dimensional MFCC of the data as input.

The supervised GMM is then used to make an i-vector extractor. i-vectors were proposed by (Dehak et al. 2011), they assume a model where each utterance is represented by a vector, *M*, given by

$$M=m+Tw,$$

where *m* are the means of the supervised GMM concatenated into one vector, *T* is a rectangular matrix of low order and $w \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The vector *w* is referred to as the i-vector, see (Dehak et al. 2011) for the method to extract the i-vectors. The i-vectors are then used for scoring using Probabilistic Linear Discriminant Analysis (PLDA). PLDA was originally proposed by (Ioffe 2006). Let α be a latent variable that determines the speaker identity of *w*. The model assumes that the conditional probability of *w* given α is given by

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\phi}_{\boldsymbol{w}}),$$

where ϕ_w is a common covariance matrix for all speakers. The model also assumes a Gaussian prior for α

$$p(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\beta}, \boldsymbol{\phi}_{\boldsymbol{\alpha}}).$$

The parameters β , ϕ_w and ϕ_{α} can be learned using maximum likelihood, see (Ioffe 2006) for the algorithm.

4.5. Speaker Recognition

The dimensionality of the i-vectors is set to 600. The PLDA model is trained on the development set, the SI utterances of the test set are used to fine-tune the parameters for each speaker in the test set, this is also referred to as enrollment. To evaluate the accuracy of the speaker recognition system, each utterance is compared to each speaker using the PLDA model. The PLDA model computes a score of how similar the utterance is to each speaker. A threshold is then created to determine the speakers who could have spoken the utterance. If the wrong speaker is chosen, it is termed a false acceptance, and if the correct speaker is rejected, it is termed a false rejection. By shifting the threshold the number of false acceptance and false rejections will change. The Equal Error Rate (EER) is defined as the rate where the false acceptance rate and false rejection rate are equal. A good speaker recognition system would have a low EER.

Chapter 5 Experiments and Results

This chapter describes the experiments conducted and displays the results.

5.1 Intelligibility

The first experiment is designed to test the intelligibility of the converted speech signals, this is done using the ASR system from section 4.4. The average WER is computed for the core test set, the test set and the two converted test sets produced by the baseline model and the FHVAE model, respectively. In order to better compare the systems, a new measure is introduced called relative WER, WER_R , given by

$$WER_R = \frac{WER_M - WER_T}{WER_C},$$

where WER_M is the average WER of the given model, WER_T is the average WER of the test set and WER_C is the average WER of the core test set. If the WER of the model is high, then so is the relative WER. The results are shown in table 5.1. From

	WER	Correct	Substitutions	Deletions	Insertions	WER_R
Core Test:	22.8	79.9	13.6	6.5	2.7	-
Test:	22.0	80.9	13.0	6.0	3.0	-
Baseline Model:	38.1	66.5	24.8	8.7	4.6	0.706
FHVAE Model:	34.6	69.5	21.7	8.8	4.1	0.553

Table 5.1: This table shows the average of the WER as well as the average number of correct words, substitutions, deletions and insertions. This is shown for the core test set, the test set and the converted test sets produced by the baseline model and the FHVAE model. Furthermore the relative WER of the baseline model and the FHVAE model are shown.

the table it is seen that the FHVAE model achieves a lower WER than the baseline model.

5.1.1 Replacement metrics to the WER

Developing an ASR system to compute the WER is a comprehensive task, it would be prudent to see if it is possible to construct a simpler method. Let x and x' be the source utterance and the converted utterance of the FHVAE model, respectively, and let X and X' be the corresponding MFCCs. Let $S_0 = |WER_x - WER_{x'}|$, where WER_x and $WER_{x'}$ are the WER of the source utterance and converted utterance, respectively. The first metric proposed, denoted S_1 , is the MSE between the MFCCs of the source and the converted utterances. Let N be the number of frames and let P be the number of mel-frequencies. S_1 is then given by

$$S_1 = \frac{1}{NP} \sum_{n=1}^{N} \sum_{p=1}^{P} (\boldsymbol{X}[p,n] - \boldsymbol{X}'[p,n])^2.$$
(5.1)

It is the hope that S_1 will show some correlation with S_0 . In figure 5.1 S_0 is plotted against S_1 . The figure does not show any correlation between the metrics, this



Figure 5.1: The absolute difference in the WER is plotted against the MSE of the MFCCs.

might be due to the fact that the MFCC contain both linguistic and non-linguistic information about the signal. S_1 would not only capture the difference in WER but also other differences, that may have a larger variance. To improve on the metric an unsupervised GMM, with C = 40 components and a diagonal covariance matrix, is trained on the MFCCs of the training set. The number of components is chosen close to the number of phonemes in the hope that the GMM will cluster the data in a manner that resembles the phonemes. Let Y be the matrix where Y[c, n] is the probability that the frame X_n belongs to class c and let Y' be defined in a similar manner. The second metric proposed, denoted S_2 , is the MSE between Y and Y', given by

$$S_2 = \frac{1}{NC} \sum_{n=1}^{N} \sum_{c=1}^{C} (\boldsymbol{Y}[c,n] - \boldsymbol{Y}'[c,n])^2.$$
(5.2)

In figure 5.2 S_0 is plotted against S_2 . The figure does not show any correlation between the metrics. It might be that the frames of MFCCs are correctly labeled, even though that the difference Y[c, n] - Y'[c, n] might be large. The third and last



Figure 5.2: The absolute difference in the WER is plotted against the MSE of the GMM posterior probabilities.

metric focuses on the class labels instead of the probabilities. Let \tilde{Y} be the matrix where $\tilde{Y}[c, n] = 1$ if the frame X_n is classified as belonging to class c and $\tilde{Y}[c, n] = 0$ otherwise and let \tilde{Y}' be defined in a similar manner. The metric is computed as

$$S_3 = \frac{1}{NC} \sum_{n=1}^{N} \sum_{c=1}^{C} (\tilde{\mathbf{Y}}[c,n] - \tilde{\mathbf{Y}}'[c,n])^2.$$
(5.3)

In figure 5.3 S_0 is plotted against S_3 . The figure does not show any correlation between the metrics.



Figure 5.3: The absolute difference in the WER is plotted against the MSE of the predicted class labels.

5.2 De-Identification

To evaluate the performance of the systems on the SDI task, the speaker recognition system introduced in section 4.5 is used. The experiment is devided into two parts, labeled 'A' and 'B'. In part A each of the converted utterances is compared to all speakers except the target speaker, the PLDA model is then told that the source speaker is the true speaker. This is done to test the similarity of the converted speech signal to the source speaker. In part B each of the converted utterances is compared to all speakers except the target speaker. In part B each of the converted utterances is compared to all speakers except the source speaker. In this way the similarity of the converted speech signal to the target speaker is measured. If the EER is lower in part A than in part B, then the speaker identity of the converted utterances are closer to the source speaker. If on the other hand the EER is lower in part B than in part A, then the speaker identity of the converted utterances are closer to the speaker.

For the core test set and the test set, the average EER is computed for female, male and all speakers, respectively, the last is termed pooled. The two converted test sets produced by the baseline model and the FHVAE model, respectively, are evaluated on both parts A and B. In each case three average EER's are computed. The first two are intra-gender conversions, that is, either male or female source-target pairs. The last EER is for both inter- and intra-gender conversions, termed pooled. The results are shown in table 5.2. In order to better compare the two

		EER	
	Female	Male	Pooled
Core Test:	15	12.5	11.67
Test:	16.67	15.31	13.83
Baseline Part A:	42.76	42.46	43.3
Baseline Part B:	28.62	31.85	30.38
FHVAE Part A:	29.93	27.26	31.82
FHVAE Part B:	40.3	41.14	38.12

 Table 5.2: The table shows the average EER for core test set, test set, converted test sets by the baseline model and FHVAE model.

systems the metric relative EER, EER_R , is introduced, given by

$$EER_R = \frac{EER_B - EER_A}{EER_C},$$

where EER_A is the average EER of part A, EER_B is the average EER of part B and EER_C is the average EER of the core test set. If $EER_A > EER_B$ the metric is negative and if $EER_A < EER_B$ then the metric is positive. The results are shown in table 5.3. From the table it is seen that the baseline model achieves negative EER_R for in all categories, while the FHVAE model achieves positive EER_R for in all categories.

	EER_R		
	Female	Male	Pooled
Baseline:	-0.943	-0.849	-1.107
FHVAE:	0.691	1.110	0.540

Table 5.3: The table shows the relative EER for the baseline model and FHVAE model.

In order to get an overall score of the models on both intelligibility and speaker de-identification, the average of the WER_R and the pooled EER_R is computed. For the baseline model it is

$$\frac{0.706 - 1.107}{2} = -0.201,$$

and for the FHVAE model it is

$$\frac{0.553 + 0.540}{2} = 0.5465.$$

It is seen that the baseline model achieves a lower score than the FHVAE model.

Chapter 6 Discussion

In terms of intelligibility, both the proposed FHVAE model and the baseline model achieves a higher WER than the unmodified test set, but the FHVAE model achieves a lower relative WER. However the baseline model was far better than the FHVAE model in masking the identity of the source speaker. In fact the converted utterances of the FHVAE model was closer to the source speaker than the target speaker. In the overall performance, calculated by the average of the relative WER and relative EER, the baseline model was also superior to the FHVAE model. The poor performance of the FHVAE model, could suggest that the factorization in the FH-VAE model is not complete; there might still be remnants of the speakers identity in the segment variable z_1 . The phase of the converted utterances is estimated, which will have a negative impact on the quality of the signals. This would influence the WER negatively, it would also affect the EER, but it should affect parts A and B equally, it should therefore not have any influence on the relative EER.

There is one key difference between the baseline model and the FHVAE model. The baseline model trains a transformation function for each source-target speaker pair. The baseline model is therefore tuned for that specific source-target speaker pair. The FHVAE model, on the other hand, is trained on a large repository of different speakers. The FHVAE model is therefore equipped to handle unseen source-target speaker pairs. On one side the FHVAE model has the advantage that only one model is needed to handle the conversions, whereas the baseline model needs a new model for each source-target speaker pair. On the other side, the baseline model is fine-tuned to each source-target speaker pair, where the FHVAE is not.

The first metric proposed to replace the WER was the MSE of the MFCCs. It did not show any correlation with the WER. There is probably too much information in the MFCCs, which is then reduced to one value. This might increase the variance of the MSE. It could therefore be prudent to extract the relevant information from the MFCCs, before calculating the MSE. This was the notion behind the other two metrics, however they did not show any correlation with the WER.

Chapter 7 Conclusion and Future Work

Speaker de-identification was performed using a FHVAE model. The FHVAE model was found inferior to the baseline model, which was based on a GMM mapping and a harmonic plus stochastic model. The FHVAE model did show better results on intelligibility compared to the baseline, but the speaker identity of the converted speech signals was closer to the source speaker than the target.

The search for a metric to replace the WER as a measure of ineligibility proved unsuccessful. Three metrics were proposed. The first was the MSE of the difference in the MFCCs of the converted and source utterances. The second and third metric were based on an unsupervised GMM with 40 components trained on the MFCC of the utterances. The second metric was the MSE of the difference in the posterior probabilities, and the third metric was the MSE of the difference in the predicted labels. None of the metrics showed any correlation with the WER.

7.1 Future Work

It could be interesting to investigate whether different configurations of the FHVAE model could provide better results. The dimension of the latent variables z_1 and z_2 could be varied as well as the discriminative weighting parameter α . Another approach could be to train two FHVAE models, one trained on the log-magnitude spectrograms, and the other on the corresponding phase spectrograms. This system could potentially improve the performance of the FHVAE model, since it is not necessary to estimate the phase.

For the search of a metric to replace the WER, it could be investigated if initializing the GMM means with the means of the phonemes. This might help the GMM to capture the phones in the data. Another approach could be to use the monophone models, to extract or label features for the GMM. It could also be investigated whether other distance metrics than the MSE could provide better results.

Bibliography

- Abou-Zleikha, Mohamed et al. (2015). "A Discriminative Approach for Speaker Selection in Speaker De-Identification Systems". In: 23rd European Signal Processing Conference (EUSIPCO), 2015. United States: IEEE, pp. 2102–2106. ISBN: 978-0-9928626-3-3. DOI: 10.1109/EUSIPC0.2015.7362755.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer. ISBN: 978-0387-31073-2.
- Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory*. Second. Wiley. ISBN: 978-0-471-24195-9.
- Dehak, Najim et al. (2011). "Front-End Factor Analysis for Speaker Verification". In: IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESS-ING. Vol. 19. 4, pp. 788–798.
- Eslava, Daniel Erro (2008). "Intra-Lingual and Cross-Lingual Voice Conversion Using Harmonic Plus Stochastic Models". PhD thesis. Universitat Politècnica de Catalunya.
- Garofolo, John S. et al. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1*. Web Download. Philadelphia: Linguistic Data Consortium.
- Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins (1999). "Learning to Forget: Continual Prediction with LSTM". In: *Neural Computation* 12, pp. 2451– 2471.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-term Memory". In: 9, pp. 1735–80.
- Hsu, C.-C. et al. (2016). "Voice Conversion from Non-parallel Corpora Using Variational Auto-encoder". In: *ArXiv e-prints*. arXiv: 1610.04019 [stat.ML].
- Hsu, Wei-Ning, Yu Zhang, and James Glass (2017). "Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data". In: Advances in Neural Information Processing Systems 30. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 1878–1889. URL: http://papers.nips.cc/paper/6784unsupervised-learning-of-disentangled-and-interpretable-representationsfrom-sequential-data.pdf.
- Ioffe, Sergey (2006). "Probabilistic Linear Discriminant Analysis". In: *Computer Vision ECCV 2006*. Ed. by Ales Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 531–542. ISBN: 978-3-540-33839-0.

- Kingma, D. P. and M. Welling (2013). "Auto-Encoding Variational Bayes". In: *ArXiv*. URL: https://arxiv.org/abs/1312.6114v10.
- Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: CoRR abs/1412.6980. arXiv: 1412.6980. URL: http://arxiv.org/ abs/1412.6980.
- Machado, Anderson F. and Marcelo Queiroz (2010). "VOICE CONVERSION: A CRITICAL SURVEY". In: Sound and Music Computing Conference (SMC2010).
- Mohammadi, Seyed Hamidreza and Alexander Kain (2017). "An overview of voice conversion systems". In: Speech Communication 88, pp. 65 –82. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2017.01.008. URL: http://www. sciencedirect.com/science/article/pii/S0167639315300698.
- Pobar, M. and I. Ipsić (2014). "Online speaker de-identification using voice transformation". In: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1264–1267. DOI: 10.1109/MIPRO.2014.6859761.
- Povey, Daniel et al. (2011). "The kaldi speech recognition toolkit". In: *IEEE 2011 workshop*.
- Snyder, David, Daniel Garcia-Romero, and Daniel Povey (2015). "Time Delay Deep Neural Network-Based Universal Background Models for Speaker Recognition". In: *Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE.

Appendix A Scripts and Files

A list of the scripts and files that are used. They are found at AAU project library, projekter.aau.dk.

FHVAE model

- convert.sh: A modified version of FHVAE_Code/egs/timit/run_fhvae.sh. The script dumbs the latent variables and generates the converted utterances.
- prep_eval.py: A modified version of FHVAE_Code/src/scripts/run_nips17_fhvae_exp.py. The script is called by convert.sh to dumb the latent variables and generate the converted utterances.
- datasets_loaders_modified.py: A modified version of FHVAE_Code/src/datasets/datasets
 The script contains general datasets loaders adapted for the purpose of this project.
- modified_functions.py: This script contains modifed functions from the FHVAE code, adapted for the purpose of the project.

Baseline model

- run_VC.sh: This script produces the converted utterances of the baseline system.
- voice_conv.py: This script is called by run_VC.sh to produce the converted utterances of the baseline system.

ASR System

- run_speech_rec.sh: A modified version of kaldi/egs/timit/s5/run.sh. The script trains the ASR system and computes the WER of the converted speech samples.
- eval_data_prep.sh: A modified version of kaldi/egs/timit/s5/local/timit_data_prep.sh The script prepares the converted speech data for the ASR system.

- eval_data_prep.py: Is called by eval_data_prep.sh to generate various files from the converted data set.
- eval_format_data.sh: A modified version of kaldi/egs/timit/s5/local/timit_format_data.sh. The script takes data prepared in a corpus-dependent way in data/local/, and converts it into the "canonical" form, in various subdirectories of data/, e.g. data/lang, data/train, etc.
- WER_experiments.sh: This script conducts the different experiments on the metrics to replace the WER.
- WER_experiments.py: Is called by WER_experiments.sh to conduct the different experiments on the metrics to replace the WER.

Speaker Recognition System

- run_speak_rec.sh: A modified version of kaldi/egs/sre10/v2/run.sh. The script trains the speaker recognition system and computes the EER of the converted speech samples.
- make_test_enroll_data.py: This script splits a data set into a test set and an enrollment set. It also writes the trials file.
- make_trials_ab.py: This script writes the trials files for experiments A and B, note they are called 'b' and 'c' in the scripts.
- scoring_common_modified.sh: A modified version of kaldi/egs/sre10/v2/local/scoring_common.s The script generates a converted data set into female and male subsets.