NONLINEAR DISCRIMINANT ANALYSIS BASED FEATURE DIMENSIONALITY REDUCTION FOR AUTOMATIC SPEECH RECOGNITION

BY

HONGBING HU

BE, Chiba Institute of Technology, 2001
MS, Tohoku University, 2003

DISSERTATION

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate School of
Binghamton University
State University of New York
2010

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Accepted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate School of
Binghamton University
State University of New York
2010

August 4, 2010

Dr. Stephen A. Zahorian, Chair
Department of Electrical and Computer Engineering, Binghamton University

Dr. Mark Fowler, Member
Department of Electrical and Computer Engineering, Binghamton University

Dr. Xiaohua Li, Member
Department of Electrical and Computer Engineering, Binghamton University

Dr. Lijun Yin, Outside Examiner
Department of Computer Science, Binghamton University

Abstract

Automatic Speech Recognition (ASR) has advanced to the point where state of the art speech recognition algorithms perform reasonably well even for large vocabulary continuous speech recognition in practical environments. Among speech recognition problems, feature extraction, which compresses a speech signal into streams of acoustical feature vectors, has become even more important for ASR since acoustical modeling methods have been well established and language modeling largely depends on the nature of the targeted language. The focus of this dissertation is the determination of effective speech features, where both spectral and temporal variations in speech are captured in a low dimensional representation, for speech recognition tasks.

In this dissertation, a set of spectral-temporal features, namely Discrete Cosine Transform Coefficients (DCTCs) and Discrete Cosine Series Coefficients (DCSCs), is examined for the purpose of capturing both the spectral and temporal variations in speech. Experimental evaluations showed that temporal variations are also of great importance for speech recognition, especially using a long time context.

Additionally, in order to reduce the limitations of the acoustical modeling based on Hidden Markov Models (HMMs), a neural network is utilized as a feature transformer to maximize the discrimination and lessen the correlation of the DCTC/DCSC features. The transformed features lead to a large improvement in the phoneme speech recognition based on the TIMIT database, especially when a small number of states and Gaussian mixtures are used for HMMs.

The neural network feature transforms are viewed as two types of Nonlinear Discriminant Analysis (NLDA) methods for nonlinear dimensionality reduction of speech

features since high dimensional features considerably increase computation costs and greatly restrict performance improvement. The first method (NLDA1) uses the final outputs of the network to obtain dimensionality reduced features with the incorporation of the Principal Component Analysis (PCA) processing, while the second one (NLDA2) focuses on the middle layer outputs. The very high phone accuracy obtained with NLDA2 based on TIMIT database was 75.0% using a large number of network training iterations based on state-specific targets.

To

my wife Ning Wang,

and

my daughter Naomi.

Acknowledgements

First I would like to thank my adviser, Dr. Stephen A. Zahorian for invaluable patience and advice throughout the research work. His expertise in the area of speech processing allowed me to expand my knowledge throughout the entire course of my PhD study. His consistent support and countless guidance, which are not limited in academic activities, have given me the ability and confidence to carry out this research work as well as tasks in the future.

I would also like to thank Dr. Mark Fowler, Dr. Xiaohua (Edwards) Li, Dr. Lijun Yin, and Dr. Kenneth McLeod for serving on my dissertation committee, offering experienced suggestions and having constructive discussions.

I would like to thank all my colleagues in the Speech Communication Lab for establishing a positive working environment and sharing a wonderful time during my study, with particular thanks to Jiang Wu, Dr. Jui-Te Hwu, Jun Feng and Zhengqing Chen for their useful suggestions and assistance.

My special thanks must go to Paul Brazil for revising the English of this work and making this dissertation read better.

Finally, my deepest thanks go to my wife, Ning Wang, and my daughter, Naomi, for their understanding and support when I indulged myself in this work. I can never thank my family members in China enough for their encouragement and support, especially my parents and parents-in-law for flying from China to help us during this special period.

Table of Contents

List of Tables

List of Figures

Chapter I Introduction

1.1 Overview of Automatic Speech Recognition

Speech is one of the most important ways for humans to communicate with each other, distribute information and acquire knowledge. There is a long history of research using machines to extend humans' ability for processing speech ever since the invention of the telephone in the late 19th century [107]. Among speech processing techniques, Automatic Speech Recognition (ASR), an approach to convert spoken words into text with the aid of machines, has been the most challenging but attractive research topic [48,34,24,97]. In 1952, Bell Labs demonstrated the first automatic speech recognizer using analog circuits for the recognition of spoken digits over the telephone. In the initial recognition systems of the 1960s, filter banks were combined with dynamic programming to produce practical recognizers [70]. Most of the commercial ASR applications during this time period were based on custom special-purpose hardware and designed for the task of small vocabulary word recognition; for example, the vowel recognizer of Suzuki and Nakata [91], and the digit recognizer of NEC Laboratories were built this way [68].

In the 1970s, Linear Predictive Coding (LPC) was introduced as an efficient speech representation and soon emerged with a central role in speech recognition. ASR applications appearing in this period included Carnegie Mellon University's Harpy [60] and IBM's Voice-Activated Typewriter (VAT). The 1980s witnessed more development in speech recognition. MFCCs (Mel-Frequency Cepstral Coefficients) [17], which approximate the human auditory system's response, replaced LPC and became a dominant speech representation, remaining so even in the present time. The fundamental approach of acoustical modeling in speech recognition was shifted from a template-based match to

statistical modeling with the advent of Hidden Markov Models (HMMs) [2,75,74]. Since their introduction, HMMs have been intensively investigated and have become one of the most successful techniques for acoustic modeling. As computers grew in power during the 1990s, numerous commercial continuous speech recognition systems for general purpose use were developed, for example, BBN's BYBLOS [85] and SRI's DECIPHER [67]. As alternative approaches to HMMs, Neural Networks (NNs) and Support Vector Machines (SVMs) [11] were introduced, but their limitations, such as the difficulty in handling temporal variation, were also revealed [70].

Since the 1990s, ASR has been largely implemented in software and a number of toolkits including Cambridge University's HTK [106] and CMU's Sphinx [32] have been provided by institutes and organizations in assisting ASR research. Recent progress has occurred in the use of statistical learning algorithms, discriminative training and hybrid HMM/NN methods. Based on the research efforts covering more than a half century, ASR has advanced to the point where state of the art speech recognition applications perform reasonably well even for large vocabulary continuous speech recognition in practical environments. However, there still remains a gap between the performance of the best ASR and the ability of humans to process speech.

Speech recognition systems are characterized by many parameters including speaker dependence, speaking model (isolated words vs. continuous speech), speaking style (read speech vs. spontaneous speech) and vocabulary size [121]. A Speaker Dependent (SD) system targets the recognition of only the speech uttered by the speaker whose acoustical materials are used in the training of the system. In contrast, Speaker Independent (SI) systems are not limited to a specific speaker but are capable of being used on any speaker

without additional training procedures. The accuracy of this type of ASR systems is generally lower due to the difficulty of accommodating various speakers. Furthermore, Speaker Adaptation is a technique used for adapting a speaker-dependent system to the characteristics of new speakers so that the performance for a speaker-dependent system is better compared to that of a speaker-independent version.

In isolated word recognition systems, targets are words with clearly defined boundaries (e.g., spoken numbers), and the pronunciation of a target word tends not to affect other words, leading to a relatively simple recognition task. Continuous speech systems are more difficult because locating the start and end points of phonemes, which are considered as the smallest sound units, is required simultaneously in the process of the recognition. Another problem is coarticulation where each phoneme is affected by the production of surrounding ones [70].

Since read speech contains few irregularities of grammar and word pronunciation, the systems recognizing read speech are able to achieve high accuracies using statistical language modeling and ordinary pronunciation dictionaries. In contrast, spontaneous speech recognition systems are exposed to more difficult conditions, such as extemporaneously generated speech (e.g., pause words such as um's and ah's, phoneme reduction, out-of-vocabulary words) [121]. This difficult and challenging recognition problem has attracted a lot of attention recently and a great amount of research has been devoted to this topic.

The size of the vocabulary of a speech recognition system affects the complexity, processing requirements, and the accuracy of the system. Some applications only require a few words (e.g. numbers only), others require very large dictionaries (e.g. dictation

machines). In general, ASR systems fall under the categories of small vocabulary (tens of words), medium vocabulary (hundreds of words), large vocabulary (thousands of words) and very-large vocabulary (tens of thousands of words) [64]. The vocabulary size of recent Large Vocabulary Continuous Speech Recognition (LVCSR) systems has increased to more than 65000 words [107].

The significant improvement in ASR resulting from the research efforts over the past several decades have enabled ASR to be used in many fields such as:

1. Dictation: converting spoken words into written texts. Applications include inputting words into a computer (e.g., the speech recognition function in Windows), and automatic generation of TV transcription;
2. Command and Control: Operating machines by uttering predefined commands. Applications include voice dialing for cell phone, automotive speech recognition (e.g., Ford Sync), hand-free computing, home appliance control, and aircraft cockpit control;
3. Computer-Assisted Language Learning (CALL): Using ASR to evaluate the utterances of a language learner. Examples include foreign language learning and articulation training for hearing impaired people;
4. Human-Machine Dialogue: Enabling machines to understand the words spoken by humans and conduct requested tasks. A large number of applications in this field have been actively used in robotics, and;
5. Content-based Spoken Audio Search: Recognizing spoken words in multimedia contents. Applications include online video search with voice and Music Information Retrieval System (MIRS).

1.2 Challenges for ASR

Humans are generally able to understand what is being said under many conditions including unfamiliar accents, background noise, improper grammar, and even unknown words. However, ASR technology is still not comparable to human recognition under these more real-world conditions. ASR is a difficult problem, largely because of the many sources of variability associated with speech.

First, it is well-known that the speech signal not only conveys the linguistic information, but also contains speaker specific properties such as gender, age, and accent [4]. Even for the same speaker, the pronunciation varies due to the speaker's physical situation, emotional state, and speaking rate. For example, compared to consonants, the duration of vowels is significantly more reduced from slow to fast speech [4]. These variations are collectively referred to as speaker variability.

The second challenge for ASR is phonetic variability because of the coarticulation phenomenon. The acoustic realizations of phonemes, the smallest sound units of which words are composed, are highly dependent on the context in which they appear [121]. This phonetic variability causes great difficulties for ASR systems to recognize phonemes using simple rules. This variability is more often encountered in continuous speech for which the acoustic signal for an identical phoneme at different positions varies greatly.

Acoustic variability that ASR must handle results from the speaker environment including background noise, room reverberation, microphones and transmission channels. For example, the performance of an ASR system usually degrades a lot when it is moved from a quiet environment to an environment with background noise. For telephone based

ASR applications, special consideration is required due to the limited bandwidth of the telephone speech, and channel noise.

Moreover, ASR is much more difficult for spontaneous speech than for read speech, because of the greater variability contained in spontaneous speech, in which the reduction of pronunciation of certain phonemes often happens. In addition, other irregularities including false starts, hesitations, and filled pauses, need to be dealt with [4].

Finally, differences in linguistic background such as dialect and language result in cross-speaker variability, adding further difficulty to language modeling for speech recognition [121].

1.3 Scope of Dissertation

This dissertation explores the use of spectral-temporal features, the combination of acoustical models, and the reduction of feature spaces as an attempt to achieve a compact highly accurate speech recognition system. Despite their widespread use in ASR, Mel-Frequency Cepstral Coefficients (MFCCs) primarily capture spectral information utilizing a human perceptual Mel scale [17]. However, the temporal variability of the speech signal is also considered to be very important for humans to distinguish sound. A set of spectral-temporal features, namely Discrete Cosine Transform Coefficients (DCTCs) and Discrete Cosine Series Coefficients (DCSCs), which models temporal variations with cosine series expansion, is examined in terms of effects on the robustness of ASR systems.

Moreover, HMMs have been one of the most successful recognition models for ASR systems due to their good time alignment capability and well-developed mathematical frameworks. In contrast, Neural Networks (NNs) have poor time alignment capabilities

but have powerful discrimination abilities, making them attractive as a component of speech recognition systems. Neural networks are not based on any assumptions about the statistics of input data, and have strong discriminative power with a supervised training approach. However, neural networks rarely succeed in large vocabulary speech recognition tasks due to the lack of ability to model long-term temporal variations. In this dissertation, a combination of HMMs and neural networks is investigated. The hybrid approach attempts to take advantage of both HMMs and neural networks in the interest of improving flexibility and recognition performance for ASR.

In contrast to many linear dimensionality reduction techniques including Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA), neural network based nonlinear reduction approaches are able to form a dimensionally-reduced representation for complex data such as speech features, while preserving both the variability and discriminability of the original data. As another objective of this dissertation, a neural network based nonlinear discriminant analysis (NLDA) is proposed with the goal of creating a compact set of highly discriminative features for accurate speech recognition.

The specific objectives of this work are:

1. To explore the spectral and temporal properties of the DCTC-DCSC features in pursuit of an "optimal" feature set for HMM based acoustical modeling;
2. To investigate a hybrid NN/HMM recognition model in which a neural network is employed as a feature transformer to obtain highly discriminative and uncorrelated features to be better modeled by HMMs;

3. To propose Nonlinear Discriminant Analysis (NLDA) transformations based on a neural network for the purpose of reducing feature dimensionality as well as improving discrimination among speech features, and;
4. To demonstrate the effectiveness of the proposed methods through comprehensive evaluation using the TIMIT database in terms of phone accuracy.

Recently, significant improvements in phonetic speech recognition have been reported by a number of research studies using different features and various recognition models. Table 1 lists proposed approaches which have reported very high accuracies using the TIMIT database [28,122], followed by the summaries of these methods. The ultimate goal of this work is to achieve a more accurate recognition algorithm with the approaches listed above and prove the effectiveness of proposed methods through comprehensive evaluation.

Table 1: TIMIT results reported in literature

Study	Feature	Recognizer	Accuracy (%)
Somervuo (2003) [89]	MFCC	NN/HMM	68.5
Ketabdar and Bourlard (2008) [53]	PLP	NN/HMM	71.5
Pinto and Hermansky (2008) [73]	LPC	HMM/MLP	74.6
Sha and Saul (2007) [87]	MFCC	HMM	70.0
Schwarz et al. (2006) [86]	MFCC	Tandem NN	78.5
Zahorian et al. (2009) [112]	DCTC/DCSC	HMM	73.9

In the work of Somervuo [89] (Somervuo2003Experiments), a multilayer perceptron network (neural network) was introduced to perform feature transformations for HMM based phone recognition. The use of final layer output as transformed features in this

work is similar to NLDA1 presented in this dissertation, but the Softmax-activation function was used in the output layer during the training and several dimensionality reduced features were combined as the input to the neural network.

Ketabdar and Bourlard [53] integrated two neural networks into a hierarchical structure to obtain phone posteriors as features for HMM based recognition. The first network transforms cepstral features to phone evidences in terms of posterior probability, and the second network processes a temporal context of the phone evidences. The resulting enhanced posteriors are then used for phone and work recognition based on HMMs.

In the work of Pinto and Hermansky [73], the log-likelihood of the features obtained from a Gaussian mixture model was combined with the posterior probability of phonemes from a discriminative neural network for recognition of phonemes. A phoneme recognition accuracy of 74% on the TIMIT database was obtained with the multi-stream combination.

Motivated by Support Vector Machines (SVMs), the parameters of GMMs were trained discriminatively to maximize the margin of correct classification, as measured in terms of Mahanalobis distance in the work of Sha and Saul [87]. This improved training procedure was experimentally shown to be superior to the HMMs trained by maximum likelihood estimation.

In the work of Schwarz et al. [86], a tandem structure of NN/HMMs with separate classifications of input patterns in frequency bands, referred to as TempoRAl Patterns (TRAPs), was employed. Multiple front-end networks were trained to classify input patterns to phoneme posteriors from different bands, and another back-end neural network was used to merge the posteriors from all bands. Furthermore, the original features

were concatenated with the posteriors from the front-end network for a more comprehensive feature set. A third network was also tested to further merge the outputs of the back-end network and the original features.

1.4 Outline of Dissertation

This remainder of the dissertation is organized as follows.

Chapter II contains a brief review of signal processing for automatic speech recognition, including feature extraction and acoustical modeling. HMMS and neural networks, the most popular acoustical models used in speech recognition, are discussed after the introduction of LPC and MFCC features. Another essential topic in speech recognition, language modeling, is also briefly described. In addition, several state-of-the-art ASR systems are presented for an aid in understanding the status of current ASR technology.

Chapter III describes the HMM based statistical modeling of the DCTC-DCSC features. The tradeoff of spectral-temporal DCTC-DCSCs in both the frequency and time domains is discussed. Fundamental algorithms of HMMs including the Viterbi decoding and Baum-Welch training are also presented, followed by a brief introduction of an HMM toolkit HTK. Finally, a comparison of DCTC-DCSCs with MFCCs based on HMMs in various conditions is described.

Chapter IV presents hybrid NN/HMM models for speech recognition. After a brief review of recent hybrid NN/HMM models, a tandem NN/HMM model, in which a neural network is trained with the aim of improving the discrimination and lessening correlation of speech features, is addressed. Key components of the method including network training, training targets and node nonlinearities are investigated. An experimental evaluation using the TIMIT database is also described.

Chapter V introduces a neural network based nonlinear discriminant analysis for speech recognition. The advantage of neural networks in dimensionality reduction is the focus of this chapter and a neural network is employed to obtain a compact set of high discriminant features for recognition based on HMMs. The use of state level training targets and the resulting improvement in network training then follow. The chapter ends with an evaluation of the method and a comparison with other linear discriminant methods such as PCA and LDA. Finally, a literature comparison with recently reported results using the same TIMIT database is provided.

The conclusions and suggestions for future work are summarized in Chapter VI.

Chapter II Background

2.1 Overview of Speech Recognition System

The objective of a speech recognition system is to produce a word sequence given a speech waveform. The general structure of an ASR system is illustrated in Figure 1.

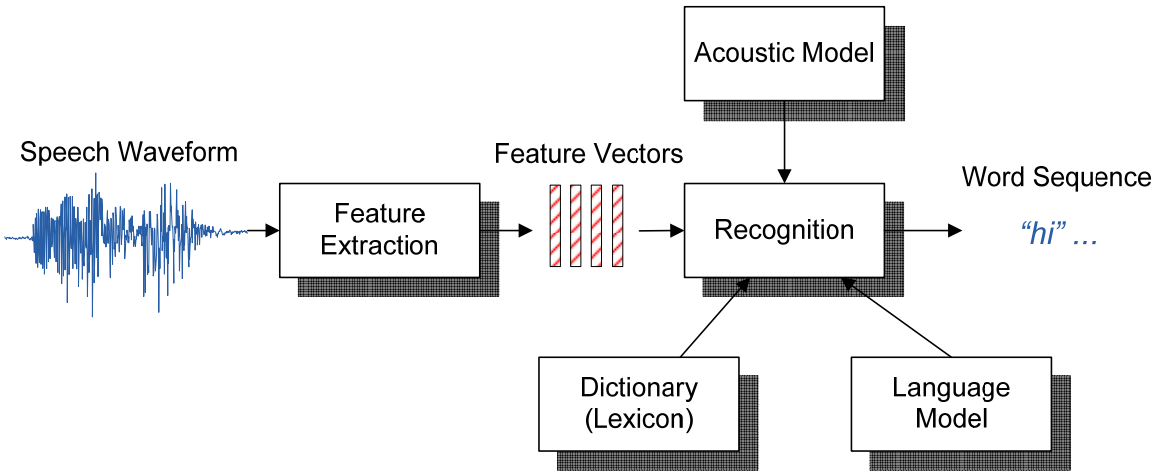

Figure 1: General structure of a speech recognition system

The first stage of speech recognition is to compress a speech signal into streams of acoustical feature vectors, referred to as speech feature vectors. The extracted vectors are assumed to have sufficient information and to be compact enough for efficient recognition [107]. This processing is known as feature extraction or front-end processing, and the details will be described in Section 2.2. Given the feature vectors, three main sources of information are required to recognize the most likely word sequence: the acoustic model, the language model and the dictionary [76]. The acoustic model maps speech feature vectors to sub-word units such as phonemes. Hidden Markov Models (HMMs) and neural networks are the most successful approaches for this role as described in Section 2.3. The dictionary, or the lexicon, provides the pronunciations of actual words presented in the language model in terms of the sub-word units used to construct the

acoustic model. The language model represents the local syntactic information of the uttered sentences and contains information of the possibility of each word or combination of words [107], for example, unigram, bigram as introduced in Section 2.4.

The most likely word sequence can be hypothesized using statistical approaches with the acoustic and language information. Within a statistical framework, the general decision criterion to find the most likely word sequence H for a sequence of feature vectors or observations O, can be determined as the result of computing:

$$\hat{H} = \arg\max_{H} P(H \mid O) \quad . \tag{II.1}$$

Considering that the most likely word sequence is independent of the prior probability of the observation $P(O)$, the posteriori probability $P(H|O)$ can be substituted using Bayes theorem so that the decision rule becomes:

$$\hat{H} = \arg\max_{H} \left\{ \frac{P(O \mid H)P(H)}{P(O)} \right\} = \arg\max_{H} \{P(O \mid H)P(H)\} \tag{II.2}$$

where $P(H)$ is the prior probability of a particular word sequence determined by a language model. The conditional probability $P(O|H)$, which is calculated using the acoustic model and the dictionary, represents the likelihood of the observation O with respect to the word sequence H. The evidence factor, $P(O)$, is merely used as a scale factor so that the sum of the posterior probabilities $P(H|O)$ equates to one [19].

2.2 Feature Extraction

2.2.1 Speech Signal Representations

A speech signal is assumed to be the convolution of an excitation source, and an acoustic filter, the vocal tract. As air is expelled from the lungs, tensed vocal cords are caused by the air flow to vibrate and generate quasi-periodic pulses as the excitation. These pulses are then filtered when passing through the vocal tract, producing voiced sounds. The vocal tract consists of articulators, such as jaw, tongue, lips and velum, which are varied to create different sounds. When vocal cords are relaxed, the airflow either passes through a constriction in the vocal tract or builds up pressure behind a closure point and the pressure is suddenly released, causing unvoiced sounds. Unvoiced sounds are controlled by the positions of constriction or closure [21]. Speech is simply a sequence of these voiced and unvoiced sounds, which vary slowly (5-100ms) due to the gradual movement of the articulators [108]. A cross-sectional view of the human vocal system used for speech production is given in Figure 2.

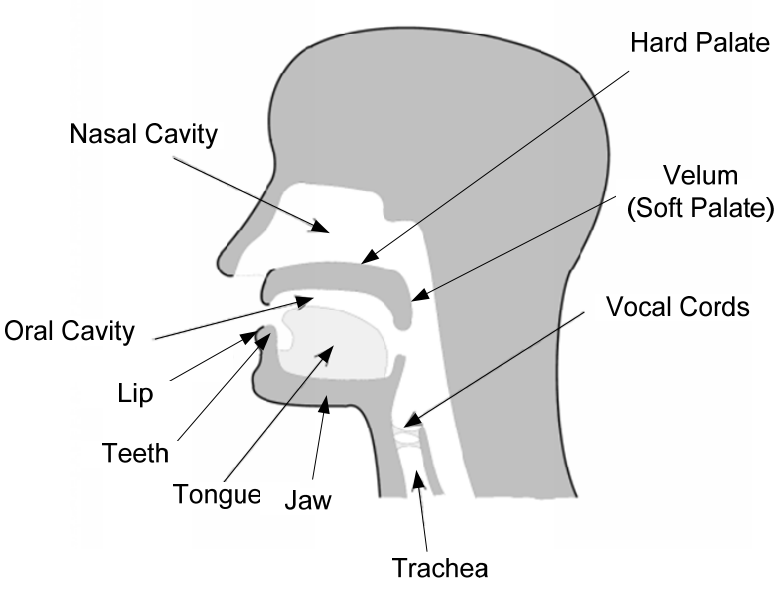

Figure 2: The human vocal system

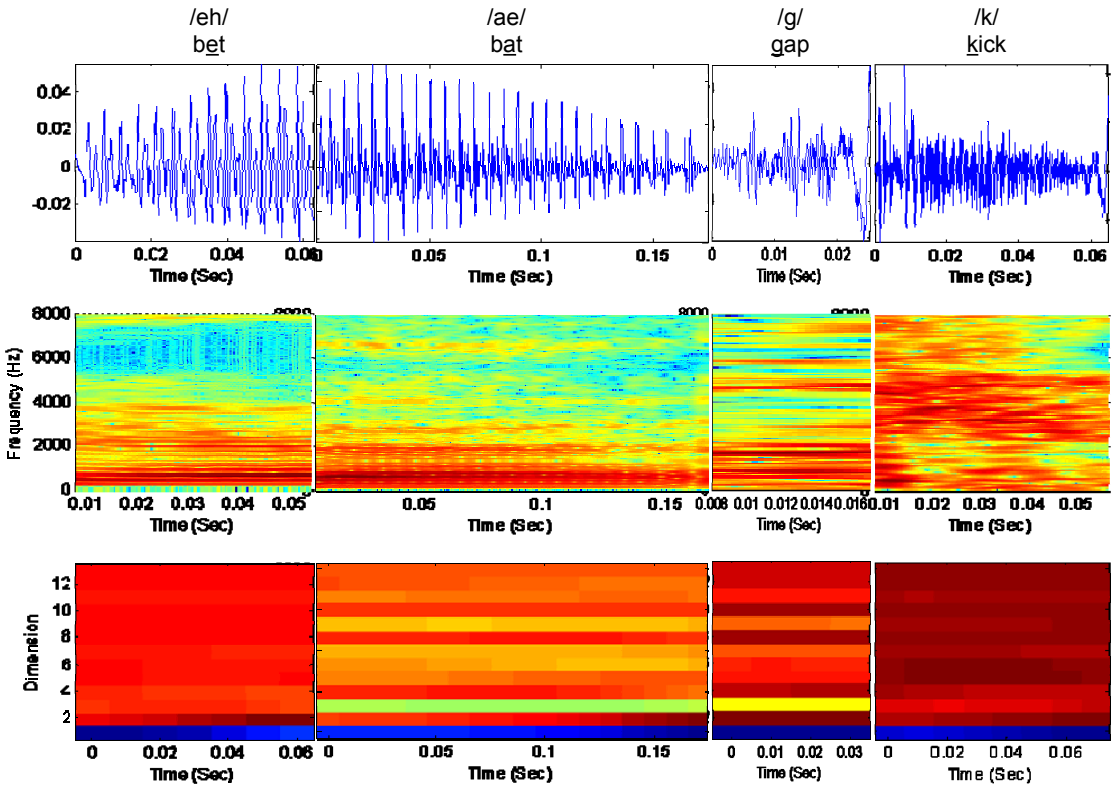

Figure 3: Illustrations of speech waveform (top panel), spectrogram (middle panel) and feature vectors (bottom panel).

Figure 3 shows the speech waveforms, spectrograms of four articulations: /eh/, /ae/, /g/, and /k/ as well as the corresponding DCTC feature vectors as will be described in Section 3.2.2. Different sounds exhibit diverse speech lengths and energy distributions on spectrogram, resulting in distinct characteristics of feature vectors.

For speech recognition, the vocal tract imparts more useful information than the excitation source because sounds are distinguished from one another primarily by the resonances of the vocal tract [37]. Therefore, the first step of speech recognition is to separate and preserve the vocal tract information while ignoring the effects of speaker differences caused by the excitation, as well as channel distortion and background noise, collectively referred to as feature extraction.

Signal modeling is the principal technique for the separation of different types of information in speech. There are two primary approaches for speech signal modeling: articulation-based signal representation and perceptually-motivated signal representation [37]. The former approach attempts to model speech signal properties that reflect the shape of the vocal tract, rather than the excitation source. For example, Linear Prediction (LP) analysis, which is presented in Section 2.2.2, utilizes an all-pole filter to model the vocal tract with the aim of obtaining a smooth envelope of a speech signal. In contrast, the perceptually-motivated signal representation emulates the human auditory process. Since the human ear resolves frequencies differently, nonlinear perceptual scales such as Mel-scale [71] and Bark scale [88], contribute to form a more effective speech representation. Filter bank analysis is one of the most popular methods along with this approach [72]. As will be introduced in Section 2.2.3, Mel-frequency cepstral analysis, which has a basis on the Mel-scale, is one of the most popular methods in perceptually-motivated representation.

2.2.2 Linear Prediction Analysis

In linear prediction (LP) analysis, the vocal tract is modeled by an all-pole filter with transfer function:

$$H(z) = \frac{G}{\sum_{i=0}^{p} a_i z^{-1}} \quad , \tag{II.3}$$

where G is a gain, P is the number of poles or the order of the LP analysis, and $\{a_i\}$ are defined as the linear prediction coefficients with a_0=1.

This all-pole filter models the gradual movement of the vocal tract, and the coefficients result in a smooth spectral representation of a speech signal. In general, higher order leads to a finer representation and lower prediction error. However, when the order becomes too large, the model fits individual harmonics and the separation of the vocal tract and excitation is not satisfied [37].

Using the Z transformation, Equation (II.3) can be rewritten to be a linear filter operation with a window of signal $\{x[n], x[n-1], ...\}$ and the prediction coefficients $\{a_i\}$. As shown in Equation (II.4), the rewritten form exhibits that a signal sample can be predicted as the weighted linear combination of its previous samples [72]:

$$\hat{x}[n] = -\sum_{k=1}^{p} a_k x[n-k] + e[n]. \tag{II.4}$$

The term P represents the number of previous samples to be considered, and $e[n]$ represents the prediction error between the estimated value and actual value.

The coefficients are generally estimated by minimizing the mean squared prediction error criterion over the analysis frame. This error for the frame n as a set of p linear equations is thus computed as:

$$E_n = \sum_{m=0}^{N-1} e_n^2[m] = \sum_{m=0}^{N-1} (x_n[m] - \hat{x}_m)^2 = \sum_{m=0}^{N-1} (x_n[m] - \sum_{k=1}^{p} (a_k x_n[m-k]))^2 \tag{II.5}$$

where N is the length of analysis frame. Minimizing the mean squared error results in the Yule-Walker equations:

$$\sum_{k=1}^{p} a_k \varphi_m[i,k] = \varphi_m[i,0] \tag{II.6}$$

where $\varphi_m[i,k]$ is the covariance function for $x[n]$.

There are three approaches to compute prediction coefficients in Equation (II.6): covariance methods based on the covariance matrix, autocorrelation methods, and lattice methods [72,103]. In autocorrelation methods, the autocorrelation matrix has the Toeplitz property because of the symmetric windowing of the speech samples and the resulting short time nature of the speech frames. Therefore, the parameters can be effectively estimated using Levinson-Durbin recursion and the stability of the filter is assured. For the above reasons, the autocorrelation methods have been used almost exclusively in speech recognition [37].

2.2.3 Mel-Frequency Cepstral Analysis

As one of the most popular perceptually-motivated coefficients, Mel-Frequency Cepstral Coefficients (MFCCs) approximate the nonlinear response of the human auditory system [17]. Experiments in human perception have shown that the position of maximum displacement along the basilar membrane for stimuli such as pure tones is proportional to the logarithm of the frequency of the tone [72]. Therefore, instead of using the linearly spaced frequency bands obtained with the Fourier transform, MFCCs are derived from the cosine transform of the logarithmic Mel-scale bands [76].

The Mel scale was developed by Stevens and Volkman in 1940 as a result of a human auditory perception study. The scale was based on experiments where listeners were asked to divide frequency range into perceptually equal intervals with the reference

frequency of 1000 Hz as 1000 Mels [71]. The approximation of the Mel frequency is computed as:

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700}). \tag{II.7}$$

Feature extraction based on MFCCs utilizes a bank of triangular filters of which center frequencies and bandwidth are warped using the Mel-frequency scale. The following equation shows *M* triangular filters in a Mel filter bank:

$$H_m[k] = \begin{cases} 0, & k < f[m-1], \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1]}, & f[m-1] < k < f[m], \\ \dfrac{2(f[m+1] - 1)}{(f[m+1] - f[m-1])(f[m+1] - f[m]}, & f[m] \le k \le f[m+1], \\ 0, & k > f]m+1] \end{cases}. \tag{II.8}$$

The function *f*[] is determined by the lowest f_{lowest} and highest $f_{highest}$ frequencies of the filter bank, as well as the sampling frequency f_{sample} and the number of bins in the linear frequency domain *N* according to :

$$f[m] = \frac{N}{f_{sampling}} Mel^{-1}(f_{lowest} + m\frac{f_{lowest} - f_{highest}}{M+1}) . \tag{II.9}$$

The inverse of the Mel frequency can be calculated from *Mel(f)* in Equation (II.7) as:

$$Mel^{-1}(f) = 700(10^{\frac{f}{2595}} - 1). \tag{II.10}$$

A 20 channel filter bank over a frequency range [0, 4 kHz] created with VOICEBOX [10] is shown in Figure 4.

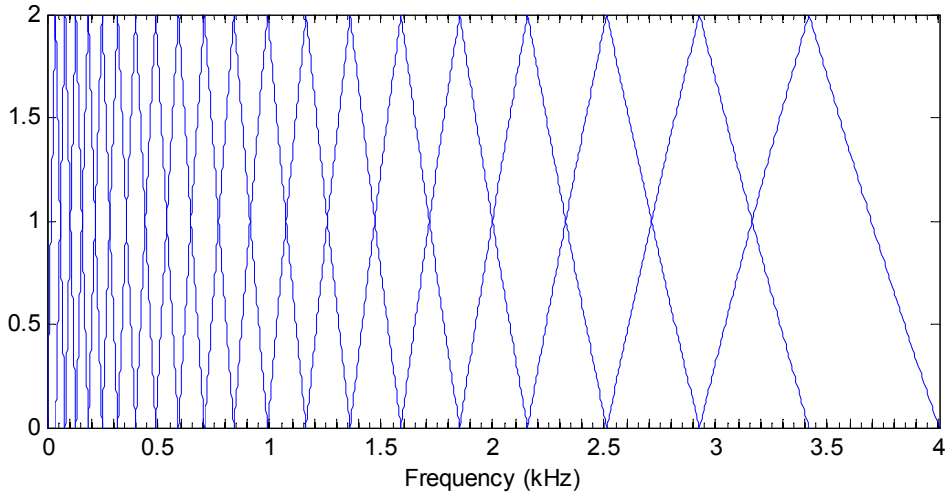

Figure 4: A 20 channel Mel-scale filter bank

To implement the filter bank, the window of speech is transformed using a Fourier transform and the spectral magnitude is computed. Each magnitude component is multiplied by the corresponding triangular filter and the results are accumulated for the coefficient of a band. Thus, each band holds a weighted sum representing the spectral magnitude in that filter bank channel.

Finally, the Mel-weighted cepstral coefficients c_i are calculated from the log filter-bank amplitudes, denoted m_j, using the discrete Cosine transform:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos\left(\frac{\pi i}{N}(j-0.5)\right) \qquad \text{(II.11)}$$

where N is the number of filter bank channels.

Among MFCCs, The first cepstral coefficient is a measurement of the overall amplitude of the log spectrum, and the second one delivers a measure of the balance between the two halves of the spectrum. Higher order coefficients provide information on the finer features of the spectrum. Compared to linear frequency cepstral analysis, MFCCs possess

a significant advantage with better suppression of insignificant spectral variation in the higher frequency bands. In addition, MFCCs are able to preserve sufficient information for speech recognition with a small number of required coefficients. For example, six coefficients succeed in capturing most of the relevant information [17].

2.3 Acoustic Modeling

2.3.1 Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) were introduced into ASR with the aim of handling both temporal and spectral variability of the speech using stochastic approaches. A hidden Markov model (HMM) is characterized by a finite-state Markov model and a set of output distributions. The transition parameters in the Markov chain model temporal variation of speech, while the parameters in the output distributions represent spectral variability.

Figure 5 shows an example of a 3-state left-to-right HMM, where $a_{i,j}$ represents a state transition probability for state i to state j, and $b_i(x)$ is the observation probability or output probability of the feature vector x at state i.

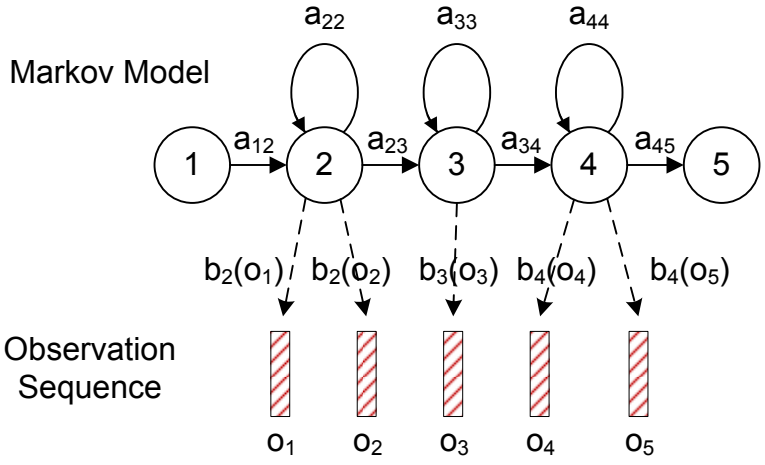

Figure 5: A 3-state left-to-right HMM model

To compute the probability that a sequence of speech feature vectors O $\{o_1, o_2 \ldots\}$ is observed from an HMM, both the transition probabilities between states and the observation probabilities feature vectors are considered. Given the state sequence X $\{x_1, x_2 \ldots\}$, an overall probability is calculated as the product of the transition probabilities and the output probabilities:

$$P(O, X \mid M) = a_{12} b_2(o_1) a_{22} b_2(o_2) a_{23} b_3(o_3) \ldots . \tag{II.12}$$

However, in practice, only the observation sequence O is a known quantity and the state sequence X is hidden, so this model is called a Hidden Markov Model [106]. With the unknown sequence X, the required likelihood of occurrence is computed by summing over all possible state sequences $X = x(1), x(2), x(3), \ldots, x(T)$ given as:

$$P(O \mid M) = \sum_{X} a_{x(o)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)} \tag{II.13}$$

where $x(0)$ is constrained to be the entry state of the model and $x(t+1)$ the exit state.

As an alternative to Equation II.13, the likelihood of occurrence can be approximated by merely considering the most likely state sequence as:

$$P(O \mid M) = \max\left\{ a_{x(o)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)} \right\}. \tag{II.14}$$

The success of HMMs arises from the existence of efficient parameter training and recognition algorithms. For example, the Baum-Welch algorithm is efficient for estimating transition and output parameters, and the Viterbi algorithm is powerful for finding the maximum overall probability and decoding the state sequence. Detailed descriptions of these algorithms are presented in Section 3.3.

When HMMs started to appear in speech recognition, most uses were based on discrete HMMs, which rely on the assumption of discrete symbols in the input. For these models, a quantization using a clustering technique [94], is required to convert continuous speech vectors into a finite size codebook. However, this process undoubtedly sacrifices recognition accuracy. Alternatively, Continuous Density HMMs (CDHMMs) [29] model the output distribution using continuous probability density functions, usually referred to as likelihoods. As will be described in Section 3.3, Gaussian or Mixtures of Gaussian components are the most popular and effective choices for representing the state distribution in CDHMMs. For continuous large vocabulary recognition tasks, CDHMMs exhibit better recognition accuracy.

In ASR systems, an HMM is typically used to model a speech unit which could be a word, a phoneme or a syllable. When the vocabulary required for a recognition task is small, the ideal usage of an HMM is to represent words as a basic recognition unit. However, in large vocabulary speech recognition, HMMs are mostly used to model subword units such as phonemes, because most languages have a limited number of phonemes and the models can be trained with a reasonable level of training data. A monophone HMM is a context-independent unit in the sense that a phoneme doesn't distinguish its adjacent ones. However, due to coarticulation effects, the pronunciation of a phoneme is strongly influenced by its neighboring phonemes, especially in spoken speech. Thus, context-dependent units such as biphones or triphones are widely used. A biphone HMM models a phoneme with its left or right context, and a triphone HMM represents a phoneme with its left and right context [56].

2.3.2 Neural Networks

Neural networks emerged as an attractive acoustic modeling approach in ASR in the late 1980s. Since then, neural networks have been used in many aspects of speech recognition such as phoneme classification [96], isolated word recognition [104], and speaker adaptation [109].

Figure 6 shows a simple feed-forward three-layer neural network, which is also called a Multilayer Perceptron (MLP). This network consists of an input layer, a hidden layer, and an output layer, with multiple nodes (or units) contained in each layer.

The function of each node is based on the properties contained in biological neurons, and hence a node is also referred to as a "neuron."

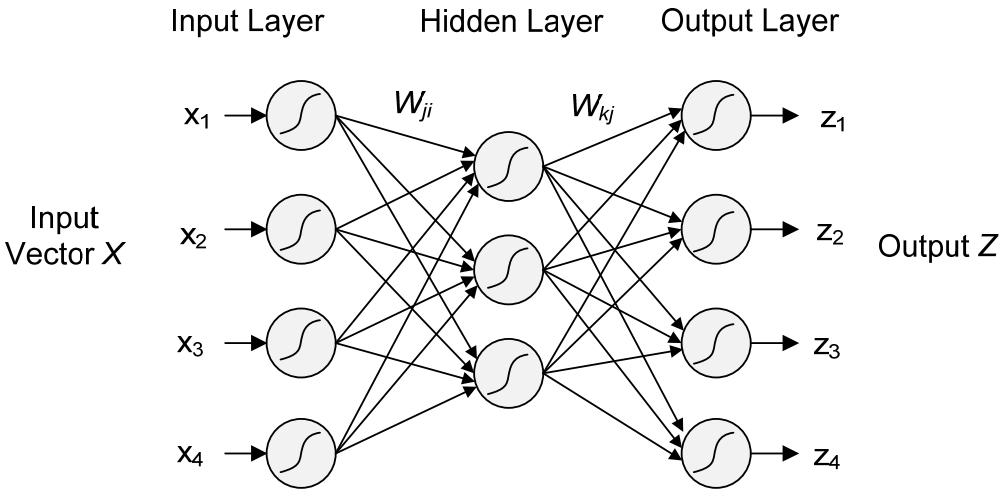

Figure 6: A feed-forward neural network with 3 hidden layers

Each connection between two nodes is associated with a weight. A node computes the weighted sum of the inputs from such connections to form its scalar net activation. For example, the net activation of a hidden node is the inner product of the connected input nodes with their weights. Given an input vector X and the weight vector W, the activation net of a hidden node is computed as:

$$net_j = \sum_{i=0}^{d} x_i w_{ji} = W_j^t X \tag{II.15}$$

where the subscript i indexes nodes in the input layer, and j in the hidden layer. The variable of w_{ji} denotes the weight from the ith input node to the jth hidden node, and d is the number of input noides.

The output of this hidden node is the nonlinear function of its activation, f(net). The f(net) is sometimes called the activation function or "nonlinearity" of a node [19].

Similiary, the output nodes compute the nonlinear function of their net activations based on the hidden node signals such as:

$$g_k(x) = z_k = f\left(\sum_{j=0}^{n} w_{kj} f(net_j)\right) = f\left(\sum_{j=0}^{n} w_{kj} f\left(\sum_{i=0}^{d} w_{ji} x_i\right)\right) \tag{II.16}$$

where subscript k indexes each unit in the output layer and n denotes the number of hidden nodes.

In general, the activation function is desired to be nonlinear, continuously differentiable, and saturate. Other properites required for the function are continuity and smoothness so that it can be defined throughout the range of its argument [19].

A major paradigm for the weight estimation is superivsed learning in which the neural network is designed to minimize the cost function using training data. As will be introduced in Section 4.2.2, Backpropagation is one of the simplest but most genenral methods for supervised training of multilayer neural networks.

2.4 Language Modeling

The language model is another key component in speech recognition. For small vocabulary speech recognition tasks, a word network based on an automaton performs

effectively for accurate recognition, but large vocabulary continuous speech recognition systems often require statistical language models, such as N-gram language models.

The assumption of an N-gram model is that the probability of a word appearing $P(W_k)$ can be calculated using a history of *N-1* words as:

$$P(W_k \mid W_{k-1},\cdots,W_1) \approx P(W_k \mid W_{k-1},\cdots,W_{k-N+1}), \tag{II.17}$$

where *N-1* is the pre-determined size of word history. *N* is normally small, for example three, which is referred to as a trigram model used to capture word triplets.

Other commonly used models include bigram and unigram. Bigram models use statistics of word pairs and a unigram is simply the probability of a word being used independent of its context. The estimation of these probabilities is performed through the analysis of a significant amount of text to capture both syntactic and semantic redundancies.

2.5 State of the art ASR systems

Although HMMs and neural networks have been successfully used in a variety of ASR tasks within straightforward and well-defined mathematical frameworks, the state-of-the-art ASR systems involve considerable refinements. These refinements include feature projection, discriminative training, speaker adaptation, noise compensation, prosody consideration and recognition model combination [27,62].

2.5.1 Feature projection

In feature projection, dynamic first and second differential parameters, the so-called delta and delta-delta parameters were added to the static feature parameters in order to overcome the limitations of the conditional independence associated with HMM. Mean-

while, data-driven approaches can be used to reduce the correlation of the features as well as reducing the dimensionality. The standard approach is to use linear transformations, such as unsupervised Principal Component Analysis (PCA) [69], and supervised Linear Discriminant Analysis (LDA) [49,43].The LDA can be extended to take into account covariance in each class as used in Heteroscedastic Discriminant Analysis (HDA) [84,55], and Maximum Likelihood Linear Transform (MLLT) [31]. It is also advantageous to combine various feature sets because a variety of information from different sources can be collected in one comprehensive speech presentation. For example, the auditory and articulatory motivated features can be effectively combined both directly using LDA and indirectly using a Discriminative Model Combination (DMC) [120].

2.5.2 Discriminative HMM Estimation

In an HMM-based recognizer, the estimation of the HMM model parameters are typically based on maximizing the likelihood that the models generate the training sequence. However, the maximum discriminability between models is not guaranteed with the Maximum Likelihood (ML) training criterion [27]. Alternatively, discriminative parameter estimation optimizes the correctness of a model by formulating an objective function based on both correct and incorrect answers [95].

Maximum Mutual Information (MMI) Estimation has been studied in the context of small vocabulary speech tasks and substantial gains in performance have been reported [12]. Another similar discriminative training method, Minimum Classification Error (MCE) is a measure of error based on a smooth function of the difference between the correct word sequence and all other competing sequences [47].

2.5.3 Speaker Adaptation

In the case of speech recognition, there will always be new speakers who are poorly represented by the training data. The solution to this problem is speaker adaptation, which allows a small amount of data from a target speaker to be used to transform an acoustic model set so that the updated version can more closely match that speaker [27].

There are two popular schemes in adaptation. Feature-based approaches depend on acoustic features and attempt to remove speaker specific information. For example, Cepstral Mean Normalization (CMN) removes the average value of the feature-vector from each observation, and Cepstral Moment Normalization (CMtN) scales each individual feature coefficient to have a unit spectral moment and empirically this has been found to reduce sensitivity to additive noise [93].

In contrast, linear transformation based approaches attempt to adapt acoustic model parameters with the presence of adaptation data. In Maximum Likelihood Linear Regression (MLLR), a set of linear transforms are used to map an existing model set into a new adapted model set so that the likelihood of the adaptation data is maximized [27]. MLLR is generally very robust and well suited to unsupervised incremental adaptation [25]. In contrast, the Maximum A Posteriori (MAP) adaptation uses standard statistical approaches to obtain robust parameter estimates [30]. However, the adaption of MAP is a time consuming process since each Gaussian component in the model is updated separately.

2.5.4 Noise Compensation

For a speech recognition application to be used in practical environments, a noise compensation algorithm is necessary to cancel the effect of ambient noises. The ideal solution is to design a noise robust feature representation. For example, the Perceptual

Linear Predictive (PLP) is proven to be effective in a noisy environment [33]. In the so-called RASTA-PLP coefficients, the features are processed with a special bandpass filtering, which result in a noise robust representation [35]. Noise robustness can also be improved by training on speech recorded under a variety of conditions as so called multi-style training.

Features can be compensated or enhanced to remove the effects of noise, such as spectral subtraction [6]. Meanwhile, model compensation approaches take the similar strategy to MLLR for speaker adaption, which transform clean models to match the noisy environment. Model-based compensation entails combining clean speech models with a model of the noise. For example, Parallel Model Combination (PMC) maps the Gaussian means and variances in the cepstral domain using the noisy speech in the time domain [26].

Feature compensation provides a simple and efficient way in eliminating noises. However, model compensation has the potential for strong robustness as the model is adapted to encode the characteristics of the noise added speech [27].

2.5.5 Prosody Consideration

Numerous studies show the importance of prosody for human speech recognition. Pitch (fundamental frequency or F_0) is especially important for ASR in tonal languages, such as Mandarin speech. The integration of fundamental frequency with other acoustic features in the recognition process is expected to significantly increase the performance of ASR systems [100,111,110].

The consideration of pitch information differs from training recognizers with objective accent data, to various ways of modeling pitch information [118,99,13]. It was found

the training on non-native speech data achieves the most obvious gains in performance on accented data. The simplest use of adaptation is to apply speaker adaptation techniques such as MLLR and MAP on speaker-independent models to fit the characteristics of a foreign accent [99]. More sophisticated approaches focus on recognition models and use various HMMs to represent pitch and other features in a separate or integrated manner. Although some promising results have been published, the recognition accuracy on accented speech still requires further improvement.

2.5.6 Recognition Model Combination

Since any particular models or algorithms have different characteristics, it is of great interest to combine different models. Neural networks and HMMs, two of the most successful recognition models, are good candidates for forming a model combination since they are based on distinct assumptions and different mathematical frameworks. Over the past two decades, there has been an enormous research effort devoted to combining HMMs and Neural Networks with a single, hybrid architecture, called hybrid NN/HMM speech recognition [94]. The combination of neural networks and HMMs is the target domain of this work, and will be more thoroughly discussed in Chapter IV and Chapter V.

Chapter III Statistical Modeling of Spectral and Temporal Features

3.1 Introduction

Accurate ASR systems require both highly discriminative speech features and an effective recognition model. Because the human ear resolves frequencies differently, it is well known that the incorporation of nonlinear perceptual scales in feature extraction contributes to a more effective speech representation [119]. In addition to the static spectral information, the temporal trajectory information, which captures the time variation over short time intervals, has been demonstrated to be a very useful speech feature in recent research [65,35,50,15].

For the recognition side, statistical speech modeling such as HMMs emerged as one of the most successful recognition approaches for ASR. HMMs assume that the speech signal can be well characterized as a statistical process, and provide a flexible framework for making use of temporal-spectral features. For example, the output probability in an HMM can be used to model the spectral variability of the speech, and the transition probability to model the temporal variability.

In this chapter, in order to capture both the spectral and temporal variations in speech, a modified Discrete Cosine Transform (DCT) analysis of the log magnitude spectrum combined with a Discrete Cosine Series (DCS) expansion of DCT coefficients over time is introduced in Section 3.2. The modified DCT analysis extracts spectral information using a perceptual nonlinear scale, while the DCS expansion is designed to capture temporal variation in the spectrum. Therefore, these DCT/DCS coefficients lead

to a feature set that can be computed to emphasize frequency resolution or time resolution or a combination of the two factors [112].

Section 3.3 provides a detailed description of Continuous Density HMMs (CDHMMs) which incorporate Gaussian Mixture Models (GMMs) for the determination of the output probability of continuous speech features. Several key techniques of CDHMMs are also explained, such as the Baum-Welch training algorithm and Viterbi decoding algorithm.

An evaluation of the DCT/DCS coefficients using continuous density HMMs are described in Section 3.4 to show the effectiveness of using temporal-spectral features for speech recognition, In the experiment, several variations of the DCT/DCS coefficients were extracted from the TIMIT database and evaluated using various HMM configurations in terms of phonetic recognition accuracy.

3.2 DCTC-DCSC Features

3.2.1 Spectral and Temporal Analysis

As described in Section 2.2, the dominant technique used for extracting speech features relies on FFT-based spectral analysis in which the FFT of the speech signal is performed and filter bank analysis is applied to compute Mel-frequency Cepstral Coefficients (MFCCs) [17]. As an alternative to MFCCs, Zahorian et al. [115,52] presented a technique based on the encoding of global spectral shape for feature extraction. A modified Discrete Cosine Transform (DCT), referred to as Discrete Cosine Transform Coefficients (DCTCs), is directly applied to the log magnitude spectrum for spectral features. This modification using a frequency warping function is designed to simulate the nonlinearity of the human ear in speech perception.

Meanwhile, recent research has shown that the temporal trajectory information is a useful source of speech information for the purpose of improving the performance of speech recognition systems [35,50]. This information, commonly referred to as a dynamic feature, is used to capture the changes of each feature component from frame to frame, such as the delta coefficients for MFCC. These dynamic features are concatenated to the static features, forming a spectral-temporal feature vector to accommodate speech variations in both frequency and time domains. In the works of Zahorian et al. [115,112], a modified Discrete Cosine Series (DCS) expansion is proposed to represent a DCTC trajectory over a block of frames. The resulting parameters are called Discrete Cosine Series Coefficients (DCSCs). An important difference between DCSCs and MFCCs with delta terms is that the first DCSC is the smoothed version of the corresponding DCTCs, resulting in a more noise robust signal representation [51].

3.2.2 Discrete Cosine Transform Coefficients (DCTCs)

Zahorian and Nossair first theoretically showed the derivation of DCTC and DSCS parameters [113]. In this and the following sections, an improved version as described in the work of Zahorian et al. [112] is provided along with a description of the warping functions used in this dissertation.

First, let $X(f)$ be the magnitude spectrum represented with linear amplitude and frequency scales and $X'(f')$ be the magnitude spectrum as represented with perceptual amplitude and frequency scales. The relationships between linear and perceptual frequencies, and linear and perceptual amplitudes, are defined by:

$$f' = g(f), \quad X'(f) = a(X(f)) \ , \tag{III.1}$$

where *g(f)* is a nonlinear frequency warping and *a(X)* a nonlinear amplitude scaling. The differential of the perceptual frequency with respect to the linear one is:

$$df' = \frac{dg}{df} df \ . \tag{III.2}$$

For convenience of notation, f and f' are normalized to the range [0, 1]. The DCTC features for encoding the perceptual spectrum are computed using a cosine transform:

$$DCTC(i) = \int_0^1 X'(f') \cos(\pi i f') df' \tag{III.3}$$

where *DCTC(i)* is the *i*th feature as computed from a single spectral frame. Substituting Equations III.1 and III.2, equation III.3 can be rewritten as:

$$DCTC(i) = \int_0^1 a(X(g(f))) \cos(\pi i g(f)) \frac{dg}{df} df \ . \tag{III.4}$$

Modified basis vectors can be defined as:

$$\Phi_i(f) = \cos[\pi i g(f)] \frac{dg}{df} . \tag{III.5}$$

Therefore, the calculation of DCTCs using these basis vectors and the nonlinear functions becomes:

$$DCTC(i) = \int_0^1 a(X(g(f))) \Phi_i(f) df \ . \tag{III.6}$$

Consequently, using the modified basis vectors, all integrations in Equation III.6 are with respect to linear frequency. Any differentiable warping function can be precisely implemented, with no need for the triangular filter bank typically used to implement warping [113]. The frequency warping function is implemented both through the mod-

ified cosine basis vectors and through the interpolation of the spectrum, in contrast to the implementation based on the basis vectors only as reported in the work of Zahorian and Nossair [113]. This modification more closely matches the mathematical derivation of the DCTCs. This modification also results in slightly higher accuracy than that obtained using the warping applied to basis vectors only, but with less pronounced warping.

The crucial elements of this approach are the selection of the nonlinear amplitude scaling $a(X)$ and the nonlinear frequency scaling $g(f)$, so that the cosine transform is with respect to a perceptual scale. In practice, the scaling $a(X)$ is typically a log, and the scaling $g(f)$ is a Mel-like function.

In this dissertation, DCTCs were computed with Equation III.4 using a logarithmic amplitude scale. The frequency warping function used the Mel function given by:

$$f' = g(f) = 2.0959 \times \log_{10}(1 + \frac{f}{\alpha}) \tag{III.7}$$

where α is a warping factor in the range of [0, 1].

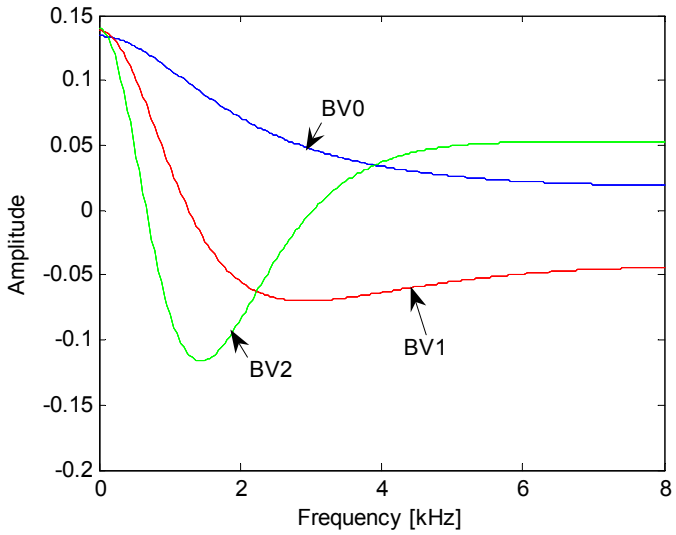

Figure 7: First three DCTC basis vectors [112]

The first three basis vectors, incorporating the warping function given in Equation III.7, are shown in Figure 7. The warping factor was set to 0.5.

3.2.3 Discrete Cosine Series Coefficients (DCSCs)

In order to create the features that represent the spectral evolution of DCTCs over time, thus encoding contextual information, a cosine basis vector expansion over time is performed using overlapping blocks of DCTCs. The DCSC features are computed so as to encode the trajectory of the smoothed short-time spectra, but typically with better temporal resolution in the central region than for the end regions.

Let the relation between linear time and perceptual time be given by:

$$t' = h(t), \tag{III.8}$$

where *h(t)* is a time warping function, chosen such that its derivative *dh/dt* determines the resolution for *t'*. For convenience, *t* and *t'* are again normalized to the range [0, 1]. The spectral feature trajectory *DCSC(i,j)* is encoded as a cosine transform over time using:

$$DCSC(i,j) = \int_0^1 DCTC(i,t')\cos(\pi j t')dt' . \tag{III.9}$$

Making the substitution:

$$t' = \frac{dh}{dt}dt \quad , \tag{III.10}$$

Equation III.9 becomes:

$$DCSC(i,j) = \int_0^1 DCTC(i,t)\cos(\pi j h(t))\frac{dh}{dt}dt . \tag{III.11}$$

Similar to DCTCs, modified basis vectors for DCSC can be defined as:

$$\Theta_j(t) = \cos[\pi j h(t)]\frac{dh}{dt}, \tag{III.12}$$

Thus, a rewritten form of Equation III.11 is:

$$DCSC(i,j) = \int_0^1 DCTC(i,h(t))\Theta_j(t)dt. \tag{III.13}$$

In practice, the integral is computed using the sum of all frames in the block. The calculation is repeated for each overlapping block, with the block spacing some integer multiple of frame spacing.

In this dissertation, a Kaiser window was employed as the time warping function *h(t)*. By varying the Kaiser β parameter, the resolution can be changed from uniform over the entire interval (β = 0), to a much higher resolution at the center of the interval than the end points (e.g., β = 5).

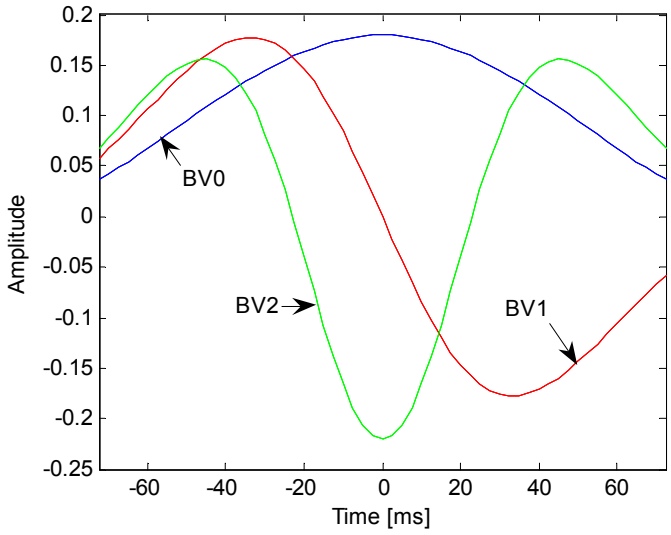

Figure 8: First three DCSC basis vectors [112]

Figure 8 depicts the first three DCSC basis vectors, using β = 5 for the Kaiser warping function. Using these modified basis vectors, feature trajectories are represented using the static feature values for each frame, but with varying resolution over a block

consisting of several frames. Therefore, DCSCs are the set of spectral-temporal features that represent speech for a block of frames.

Several parameters in the DCTC/DCSC analysis can easily be varied to examine tradeoffs between static and dynamic spectral information in terms of effects on recognition performance. For example, for increased emphasis of spectral information, a long frame length (e.g., 25 ms) and a large number of DCTCs (e.g., 15 terms) can be used to compute the spectra. To evaluate the effectiveness of purely static spectral information, the DCSC step can be eliminated and only DCTCs can be used as speech features. For increased emphasis on trajectory information, a short frame length and frame spacing (e.g., 5 ms length and 1 ms spacing) can be used along with a large number of DCSC terms (e.g., 10 terms) [112].

Figure 9 illustrates a speech spectrogram and two reconstructed spectrograms with different spectral and temporal resolutions. The first rebuilt spectrogram has high spectral resolution but low temporal resolution, which is rebuilt using 16 DCTCs and 4 DCSCs. The 16 DCTCs were computed using a frame length of 25 ms and a frame spacing of 10 ms, while the 4 DCSCs were computed using 50 frames per block. The second reconstructed spectrogram used 8 DCTCs with 5 ms frames for low spectral resolution, and 6 DCSCs with a frame spacing of 2 ms and 100 frames per block for high temporal resolution.

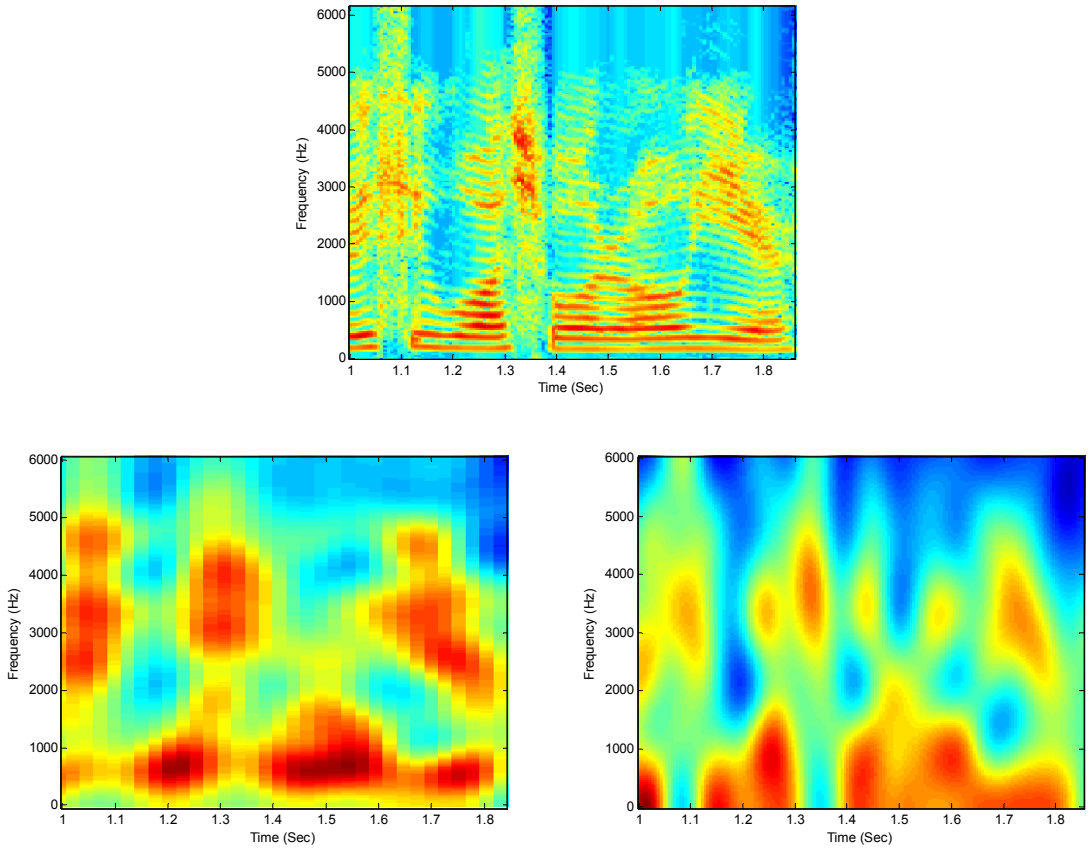

Figure 9: Original spectrogram (top), high-spectral low-temporal rebuilt spectrogram (bottom left), and low-spectral high-temporal rebuilt spectrogram (bottom right)

3.3 Continuous Density HMMs

3.3.1 Gaussian Mixture Models

As introduced in Section 2.3, the use of HMMs in speech recognition started as discrete HMMs, which assume discrete symbols in the input and require a quantization technique to convert continuous speech vectors into a finite size codebook. Alternatively, Continuous Density HMMs (CDHMMs) model the output distribution using continuous probability density functions such as a single Gaussian or Gaussian Mixture Models (GMMs) [106].

In a GMM based HMM, for a speech vector o at time t, the output probability $b_j(o_t)$ in state j is given by:

$$b_j(o_t) = \sum_{m=1}^{M} C_{jm} N(o_t; \mu_{jm}, \Sigma_{jm}) \tag{III.14}$$

where M is the number of mixture components and C_{jm} the weight of the mth component. $N(.; \mu, \Sigma)$ is a multivariate Gaussian with mean vector μ and covariance matrix Σ, which is computed as:

$$N(o; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(o-\mu)' \Sigma^{-1} (o-\mu)} \tag{III.15}$$

where n is the dimensionality of the vector o.

It is usually desired to use full covariance matrices to represent the Gaussian model since correlation exists among speech features. However, for complex HMMs with large numbers of states and mixtures, computational cost for estimating parameters make the use of full covariance matrices infeasible [37]. Therefore, the features are generally assumed to be uncorrelated and diagonal covariance matrices are generally used, including the work reported in this dissertation.

3.3.2 HMM Parameter Estimation

For one state HMMs, an approximate estimation of the mean and covariance matrices of GMMs can be obtained by taking the averages of the observed feature vectors. For the HMMs with multiple states, since all observation vectors contribute to the computation of the maximum likelihood for each state, each observation vector is assigned to every state in proportion to the probability of the model being in that state [106]. There-

fore, if $L_j(t)$ denotes the probability of being in state j at time t, the means and covariance can be determined as the following weighted averages:

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)} \tag{III.16}$$

and

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)} \tag{III.17}$$

where o_t is the speech vector o at time t, and the summations in the denominators are included for normalization.

Equations III.16 and III.17 are the Baum-Welch re-estimation formulas for the mean and covariance matrices of an HMM. The probability of state occupation $L_j(t)$ can be calculated using the following Forward-Backward algorithm.

In the forward procedure, a forward variable $\alpha_j(t)$ for a model M with N states can be defined as:

$$\alpha_j(t) = P(o_1, \ldots, o_t, x(t) = j \mid M), \tag{III.18}$$

where $\alpha_j(t)$ is the joint probability of observing the first t speech vectors and being in state j at time t. This probability can be calculated by the recursion:

$$\alpha_j(t) = \left[\sum_{i=1}^{N} \alpha_i(t-1) a_{ij} \right] b_j(\mathbf{o}_t) \tag{III.19}$$

with the initial condition:

$$\alpha_1(1) = 1, \quad \alpha_j(1) = a_{1j} b_j(\mathbf{o}_1) \tag{III.20}$$

The recursive calculation is terminated at time T, and then the required forward probability is:

$$P(O \mid M) = \alpha_N(T) = \sum_{i=1}^{N} \alpha_i(T) a_{iN} . \quad \text{(III.21)}$$

Similarly, a backward variable $\beta_j(t)$ is defined as the probability of the partial observation sequence from $t+1$ to the end:

$$\beta_j(t) = P(o_{t+1}, \ldots, o_T, x(t) = j \mid M) . \quad \text{(III.22)}$$

Using the following recursion:

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_j(t+1), \quad \text{(III.23)}$$

and the initial condition given by:

$$\beta_i(T) = a_{iN} . \quad \text{(III.24)}$$

Therefore, the probability of state occupation can be determined as the product of the forward and backward probabilities computed as:

$$L_j(t) = \frac{1}{p} \alpha_j(t) \beta_j(t). \quad \text{(III.25)}$$

3.3.3 Recognition and Viterbi Decoding

As described in the previous section, the forward probability is computed as the total likelihood $P(O|M)$ of a model. Therefore, this algorithm could also be used for recognition to find the model which produces the maximum value of $P(O|M_i)$.

However, it is more practical to compute the total likelihood based on the maximum likelihood state sequence rather than the total probability. The Viterbi algorithm follows this approach, which is the same as the forward probability calculation except that the summation is replaced by a maximum operation [106].

For a given model *M*, let $\phi_j(t)$ represent the maximum likelihood of observing speech vectors o_1 to o_t and being in state *j* at time *t*. This partial likelihood can be computed efficiently using the following recursion:

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\}b_j(o_t)\,. \tag{III.26}$$

where:

$$\phi_1(1)=1, \quad \phi_j(1)=a_{1j}b_j(o_1) \tag{III.27}$$

Therefore, the maximum likelihood *P(O|M)* is then given by:

$$\phi_N(T) = \max_i \{\phi_i(T)a_{iN}\}. \tag{III.28}$$

The recursion of Equation III.28 using log likelihood becomes:

$$\psi_N(t) = \max_i \{\psi_i(t-1) + \log(a_{ij})\} + \log(b_{ij}(o_t))\,. \tag{III.29}$$

After finding the maximum likelihood, the corresponding state sequence can be determined by tracking back from the maximum state at time *T*.

3.3.4 HTK Toolkit

The HMM Toolkit (HTK), developed by the Cambridge University Engineering Department, is a powerful software package which provides numerous modules for building Hidden Markov Models [106]. As shown in Figure 10, HTK is furnished with a variety of flexible tools for data preparation, HMM training and testing as well as result analysis in speech recognition.

For the task of data preparation, the tool HSLab can be used both to record the speech and to manually annotate it with any required transcriptions. HCopy copies speech waveform files, and a variety of mechanisms is provided for extracting speech

feature vectors such as MFCCs, LPCs. The script-driven label editor HLEd is designed to change transcription files such as reducing phone set and generating context-dependent labels.

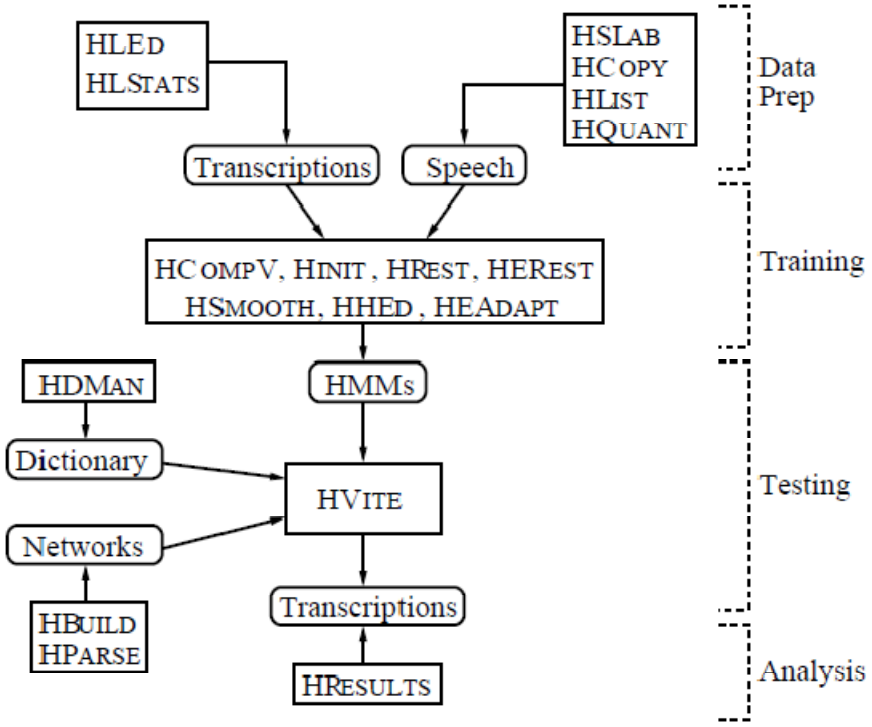

Figure 10: HTK tools [106]

Several HMM training strategies can be performed in HTK with different training tools. If the phonetic labeled training data is available, the training of each model can start with HInit to uniformly segment the data and iteratively compute an initial set of means and variances. These initial parameter values are then further re-estimated by HRest using the Baum-Welch algorithm. HERest performs a single Baum-Welch re-estimation of the whole set of HMM phone models simultaneously. For each training sentence, the corresponding phone models are concatenated and then the forward-

backward algorithm is used to accumulate the statistics of state occupation, mean and covariance matrices.

The recognition tool HVite performs the Viterbi-based decoding. The information required by the tool includes a network describing the allowable word sequences, a dictionary defining how each word is pronounced and a set of HMMs.

The analysis tool HResult uses dynamic programming to compare the two transcriptions and then counts substitution, deletion and insertion errors, as well as global performance measures. HResults can also provide speaker-by-speaker breakdowns, confusion matrices and time-aligned transcriptions.

The uses of HTK in this dissertation include:

1. Constructing HMM based phonetic recognizers using training tools HInit, HRest, and HERest; performing phone recognition with the Viterbi algorithm HVite for the evaluation of proposed methods; and obtaining state boundaries of the training data for creating the state level training targets for the neural networks used in this dissertation;
2. Analyzing recognition results using the tool HResult for both the training and test data in terms of phonetic recognition accuracy;
3. Extracting the MFCC features with HCopy for the comparison with the DCTC-DCSC features. The 62 phone set in the database was reduced to 48 using the transcription editor tools HLEd, and;
4. Language Model tools, such as HLStats and HBuild, were used to create phone bigram information as the language model of the recognition in the evaluation.

3.4 Experimental Evaluation

3.4.1 TIMIT Database

Several phonetic recognition experiments based on the TIMIT database [28,122] were conducted to evaluate the DCTC-DCSC features using the HMM based statistical modeling.

The TIMIT database contains a total of 6300 utterances, 10 utterances spoken by each of 630 speakers from 8 major regions of the United States, each with a different dialect. Of the text material in the database, two dialect sentences (SA sentences) were designed to expose the specific variants of the speakers and were read by all 630 speakers. There are 450 phonetically-compact sentences (SX sentences) which provide a good coverage of pairs of phones. Each speaker read 5 of these sentences and each text was spoken by 7 different speakers. A total of 1890 phonetically-diverse sentences (SI sentences) were selected from existing text sources to add diversity in sentence types and phonetic contexts. Each speaker read 3 of these sentences, with each text being read only by a single speaker. The database is divided into training and test sets. The training set consists of 4620 sentences spoken by 462 speakers and the test set consists of 1680 sentences spoken by 168 speakers.

In addition to the speech waveform files, time-aligned phonetic and word transcriptions are included for both the training and test data sets.

In this evaluation, the SA sentences were removed from the database, resulting in 3696 sentences (approximate 5 hours) for the training and 1344 sentences (approximate 2 hours) for the test. The original TIMIT 62 phone set was mapped to the reduced 48 phone

set described in the work of Lee and Hon [56], with the glottal stop q deleted from the TIMIT set. The TIMIT and reduced phone sets are listed in Table 2.

Table 2: List of reduced TIMIT phone set

Reduced Phone	TIMIT Phone	Reduced Phone	TIMIT Phone	Reduced Phone	TIMIT Phone
iy	iy	l	l	g	g
ih	ih	el	el	p	p
eh	eh	r	r	t	t
ae	ae	y	y	k	k
ix	ix	w	w	z	z
ax	ax-h	er	er axr	zh	zh
ah	ah	m	m em	v	v
uw	uw ux	n	n nx	f	f
uh	uh	en	En	th	th
ao	ao	ng	ng eng	s	s
aa	aa	ch	ch	sh	sh
ey	ey	jh	jh	hh	hh hv
ay	ay	dh	dh	cl	pcl tcl kcl qcl
oy	oy	b	b	vcl	bcl dcl gcl
aw	aw	d	d	epi	epi
ow	ow	dx	dx	sil	#h pau

Additionally, there are seven groups and within each ground confusions are not counted for recognition accuracy, resulting in 39 phone categories for the final evaluation: {sil, cl, vcl, epi}, {el, l}, {en, n}, {sh, zh}, {ao, aa}, {ih, ix}, {ah, ax}.

3.4.2 Experimental Setup

Left-to-right Markov models with 3-states were used and each state was represented as a GMM. A total of 48 monophone HMM models were created from the training data using the HTK toolbox. Each model was initialized with the training data and the phone level transcription, and the parameters including the mean and covariance matrices for GMMs were computed for each model state with uniformly segmented data. These initial

parameter values are then further estimated using the Viterbi alignment for 20 iterations. Finally, the Baum-Welch algorithm described in Section 3.3.2 was performed to re-estimate the parameters using the entire transcriptions for 8 iterations. In the recognition process, each test sentence was identified using the Viterbi algorithm, which determines the model with the highest likelihood of matching each phone.

The dictionary used for the phone recognition consisted of 48 phoneme pairs in addition to two silences for the start and end models. The language model, or the grammar, for these experiments was the phone bigram computed from the training data.

3.4.3 Experiment with DCTC features

This experiment evaluated the spectral only DCTC features. The number of DCTC terms and the frame size were also varied to explore how the frequency resolution of DCTCs affects recognition performance. The speech signal of the TIMIT database was pre-filtered with center frequency of 3200 Hz. The spectrum was processed for the frequency range 50 to 7000 Hz.

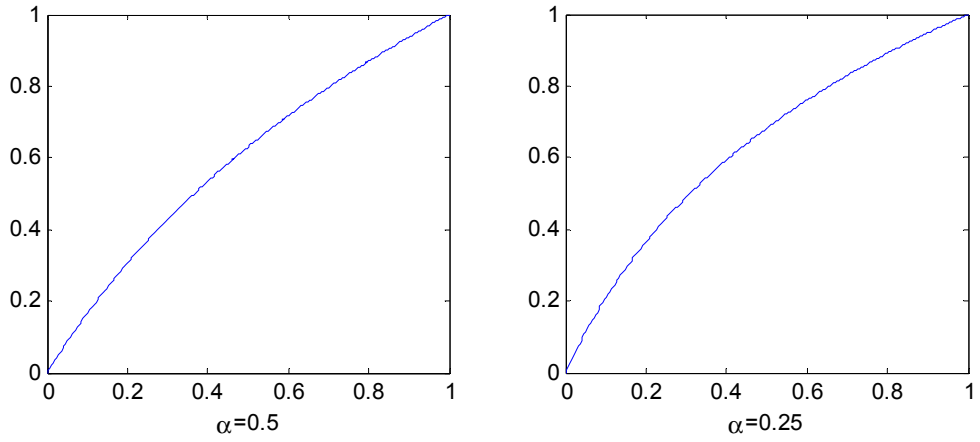

Figure 11: Mel frequency warping functions with warping factors of 0.5 (left panel) and 0.25 (right panel)

As shown in Figure 11, the Mel warping function as described in Equation III.7 was used with a warping factor of 0.5.

Figure 12 shows the recognition accuracies of the various numbers of DCTCs based on 3-state HMMs with 3 mixtures per state. The DCTC features were obtained using a frame of 8 ms with a spacing of 2 ms, a frame of 25 ms with the spacing of 10 ms and 2 ms.

The DCTCs obtained with the 8 ms frames performed considerably better than those of 25 ms frames using a 10 ms frame spacing. For example, the accuracy when using 13 DCTCs with 8 ms frames is 53.6%, approximately 18% higher than the same number of DCTCs with 25 ms frames. With the same spacing of 2 ms, the features of 25 ms frames show the similar performance as those of 8ms, indicating that the short term features could greatly improve the recognition performance. Meanwhile, the recognition accuracy increases with the expanding number of DCTCs, but with only a slight improvement being achieved with more than 13 DCTCs.

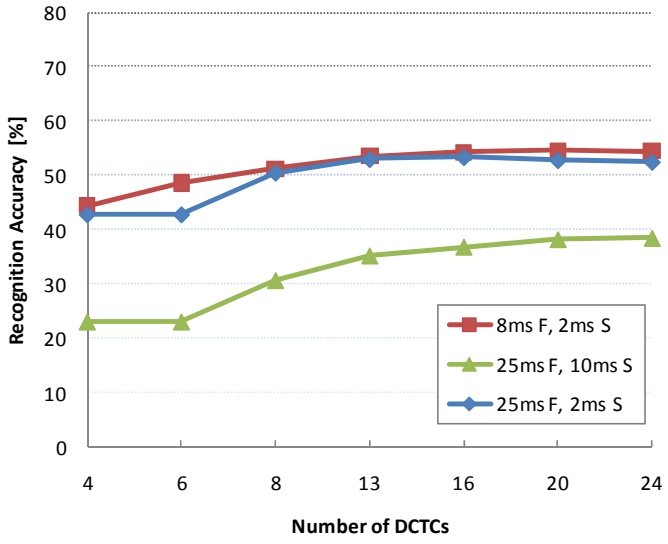

Figure 12: Recognition accuracies of using various numbers of DCTCs

3.4.4 Experiment with DCTC-DCSC features

The combination of the DCTC-DCSC features was tested in the experiments in order to evaluate the effectiveness of the spectral-temporal features for speech recognition. The 13 DCTCs, which performed the best in the previous experiment using 8 ms frames and 2 ms spacing, were integrated with various numbers of DCSCs. Therefore, the dimensionality of the feature set resulted in a large number. For example, in the case of using 6 DCTCs, a total of 78 features (13 DCTCs x 6 DCTCs) were computed.

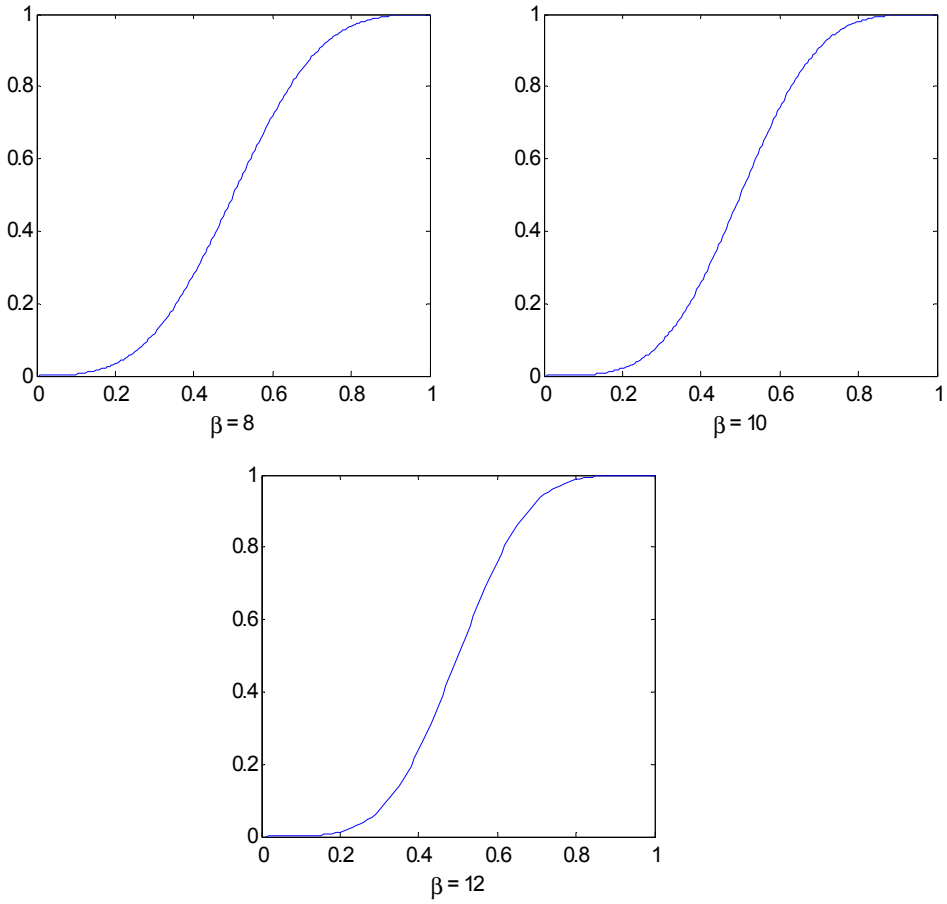

Figure 13: Time warping functions with warping factors of 8 (top left panel), 10 (top right panel), and 12 (bottom panel)

The block lengths used for DCSCS included 100 ms, 200 ms, and 500 ms. The time warping factors for a Kaiser function were 8, 10 and 12 respectively as plotted in Figure 13. The block spacing was 4 frames for all the cases.

As shown in Figure 14, the recognition accuracy increases substantially with the expanding number of DCSCs, although with more than 5 DCSC terms or 65 DCTC-DSCS terms the advantage decreases. The feature set of 500 ms frames outperformed the sets of shorter frame lengths. The highest result of 63.3% was obtained using 5 DCSCs with a block length of 500 ms.

These results imply that the temporal DCSC feature is able to capture more information from the speech, thus leading to a more efficient feature set for speech recognition. The recognition performance benefits from the longer block length used for computing DCSCs instead of higher time resolution, showing the importance of the long temporal context for speech recognition.

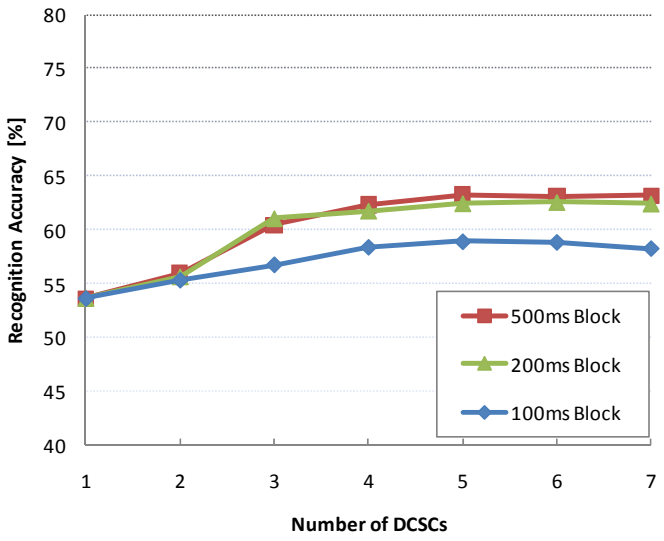

Figure 14: Recognition accuracies of using various numbers of DCSCs

Since it has been shown that the DCTC-DCSC features are able to capture the variations in both the frequency and time domain, more mixtures may be required in an HMM state to represent these features. Therefore, another set of experiments was conducted on DCTC-DCSCs with varying number of mixtures for the HMMs.

Table 3: DCTC-DCSC features for the evaluation

Feature Set	Frame length (ms)	Frame spacing (ms)	Freq. warping	Block length (ms)	Block spacing (ms)	Time warping
1	8	2	0.5	500	8	12
2	8	2	0.5	200	8	8
3	25	10	0.5	500	10	12

A total of 78 DCTC-DSCSs (13 DCTCs x 6 DCSCs) were extracted using the different conditions as listed in Table 3. For the comparison, 39-dimensional MFCC features (12 coefficients plus energy with the delta and acceleration terms) were also computed using the frame space of 10 ms and window length of 25 ms.

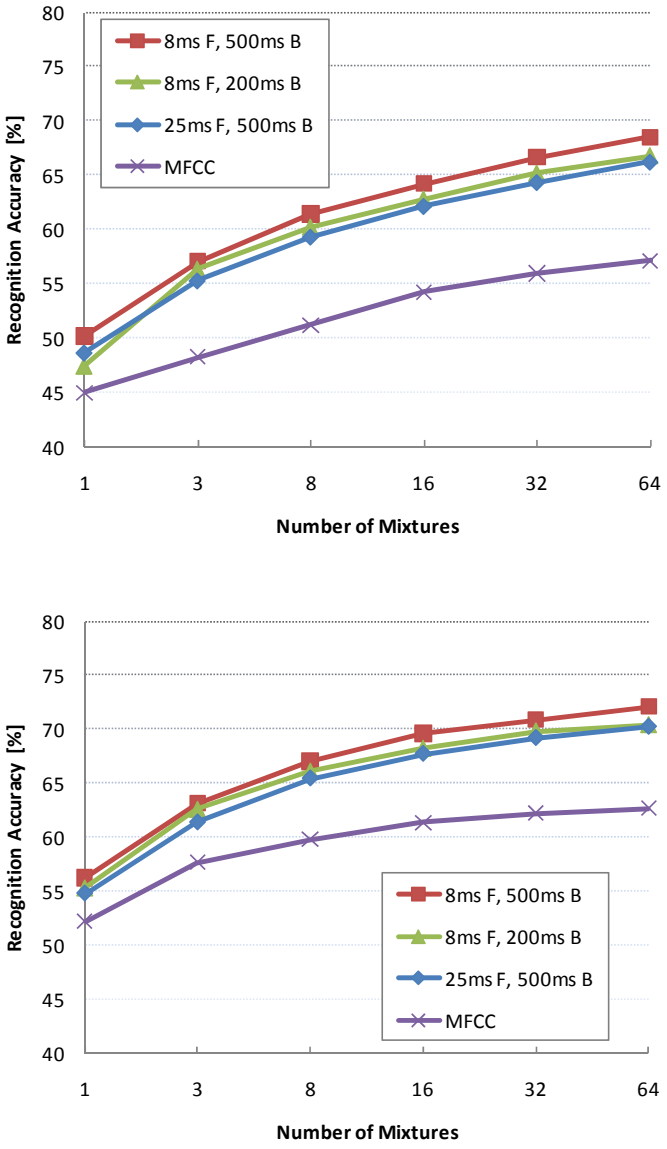

Figure 15: Recognition accuracies using 1-state (top panel) and 3-state HMMs (bottom panel) with various numbers of mixtures

As shown in Figure 15, the HMMs using more mixtures lead to higher accuracies. The highest accuracy of 72.1% was achieved for the features using 8 ms frames and 500 ms blocks based on 3-state HMMs with 64-mixtures. Similar to the experiment in Section 3.4.3, the feature sets derived from 8 ms frames performed better than those obtained

using 25 ms frame lengths. The best DCTC-DSCSs were obtained using 8 ms frames and 500 ms blocks.

The integrated DCTC-DCSC features incorporating a large number of mixtures in HMMs are able to substantially improve recognition performance, especially the feature set representing spectral information with medium spectral resolution and long temporal information.

3.5 Conclusions

In this chapter, the DCTC/DCSC features were investigated in the interest of obtaining a comprehensive feature set to capture both the spectral and temporal variations in speech. Continuous Density HMMs were also introduced as a powerful statistical modeling method for continuous speech features, especially temporal-spectral features such as DCTC/DCSCs.

From the experimental results, the following conclusions can be drawn:

1. Compared to standard features such as MFCCs which are generally extracted using 25 ms frames and 10 ms spacing, short term DCTCs (e.g. obtained with 8 frames and 2 ms spacing), which capture more temporal variations, are better sources for speech recognition;
2. The combination of spectral DCTCs and temporal DCSCs is significantly better than using DCTCs alone, showing the effectiveness of temporal-spectral features in speech recognition;
3. The DCSC features obtained with long block length (e.g. 500 ms) are able to well represent extended temporal context in order to improve recognition performance, and;

4. The integrated DCTC-DCSC features incorporating a large number of mixtures in HMMs are able to substantially improve recognition performance, especially the feature set representing spectral information with medium spectral resolution and long temporal information.

Chapter IV Hybrid NN/HMM Recognition Model

4.1 Introduction

Although HMMs have been successfully applied in a variety of ASR tasks for at least the past two decades, standard HMMs, which are trained with the forward-backward or Viterbi algorithms based on the Maximum Likelihood criterion, result in poor discriminative power among different models. In addition, the acoustic modeling of HMMs using GMMs depends on strong assumptions of statistical properties. For example, the speech is assumed to be Gaussian as the Gaussian distribution is used to represent a state of the HMMs and the random deviation of a speech vector in a state is assumed to follow a Gaussian distribution. Each dimension of the speech feature vector is assumed to be independent when the diagonal covariance matrices are used in GMMs. However, the speech signal does not necessarily follow these assumptions.

In contrast, neural networks make no assumptions about feature statistical properties and have several qualities making them attractive feature classifiers for speech recognition. When used to estimate the probabilities of a speech feature segment, neural networks allow discriminative training in a natural and efficient manner. Few assumptions on the statistics of input features are made with neural networks. Furthermore, neural networks have been found to cope well with highly correlated and unevenly distributed features such as spectral energy from several adjacent frames [20]. However, in spite of their effectiveness in classifying short-time units such as individual phones and isolated words [104,117], neural networks are rarely successful for continuous recognition tasks, largely because of their lack of ability to model temporal dependencies.

Over the past two decades, there has been a lot of research devoted to combining HMMs and neural networks with a single, integrated architecture, called hybrid NN/HMM speech recognition [94]. These hybrid systems attempt to take advantage of both HMMs and neural networks to improve flexibility and recognition performance. For example, the hybrid system proposed by Bourlard and Morgan [8] applied a neural network to estimate the posterior probabilities of HMM states. Rigoll et al. [78] combined context dependent discrete HMMs and neural networks for a hybrid system, in which neural networks are used as vector quantizers to map the speech feature vectors to discrete output labels. A self-organizing learning approach was also proposed to maximize the mutual information between the network output labels and the phone labels. Recently, the so-called TANDEM recognition approach [36,20] has resulted in a large improvement in recognition performance. This approach connects a neural network in tandem with HMMs and speech features are transformed by the neural network for discriminative features, which then become the input to the GMM based HMMs.

In this chapter, a combination of a neural network and HMM is presented for a phonetic recognition system. In order to obtain an effective speech representation for the HMM based recognition model, the neural networks employed in this approach are trained as feature classifiers to maximize discrimination of phones from speech features. Several nonlinear functions of the network nodes are also explored to find the optimal nonlinearity of the network for this hybrid approach. In addition, an established linear dimensionality reduction method, Principal Component Analysis (PCA), is also used to de-correlate and reduce the number of features. Portions of this chapter were given in the

works of Hu and Zahorian [39,41]. This chapter provides a comprehensive description of the approach including the investigation of various nonlinear functions.

The remainder of this chapter is organized as follows: In Section 4.2, several architectures of neural networks are introduced for speech recognition, followed by the descriptions of the recognition and network training algorithms. Section 4.3 summarizes hybrid NN/HMM recognition models and introduces the major categories of these models. A hybrid method, which uses a neural network as a feature transformer, is proposed in Section 4.4, as well as the details on network training. Section 4.5 describes the evaluation of the proposed approach based on the TIMIT database with various neural networks and HMM configurations. The conclusions are given in Section 4.6.

4.2 Neural Networks in Speech Recognition

4.2.1 Network Architectures

Neural networks provide an attractive feature modeling approach for speech recognition due to their strong discriminative training, ability for handling complex data, and flexible network configuration[101]. The first attempt to use neural networks for speech recognition was by Lippmann [59]in the late 1980s. Since then, a variety of neural networks with diverse structures have been proposed for different ASR tasks [58]. The most popular architectures introduced to date include Feed-forward Neural Networks, Time-Delay Neural Networks (TDNNs), and Recurrent Neural Networks (RNNs).

Feed-forward neural networks, also referred to as Multilayer Perceptrons (MLPs), consist of an input layer, one or more hidden layers, and an output layer. The nodes in a layer are only connected with their adjacent layers and there is no feedback in the network. The activation of a node is calculated as the weighted sum of the connected nodes

in its previous layer and then a nonlinear function is applied to determine the node output. The architecture of these networks is the one most commonly applied for pattern recognition and becomes the foundation of other network architectures [64].

TDNNs represent an effective attempt to construct a feed-forward network for time-sequence processing by integrating multiple temporal input vectors into a spatial sequence. As shown in Figure 16, the input layer has been enlarged to store the current as well as a number of delayed inputs, so that each hidden unit accepts inputs from both the current and the previous time-slot features. Hidden units at delayed locations accept inputs from the input layer which are similarly shifted. This architecture has been applied in a variety of ASR applications, mostly for phoneme recognition. For a speaker-dependent recognition of the phonemes "B," "D," and "G" in varying contexts, a significant recognition rate of 98.5% was achieved with TDNNs [96].

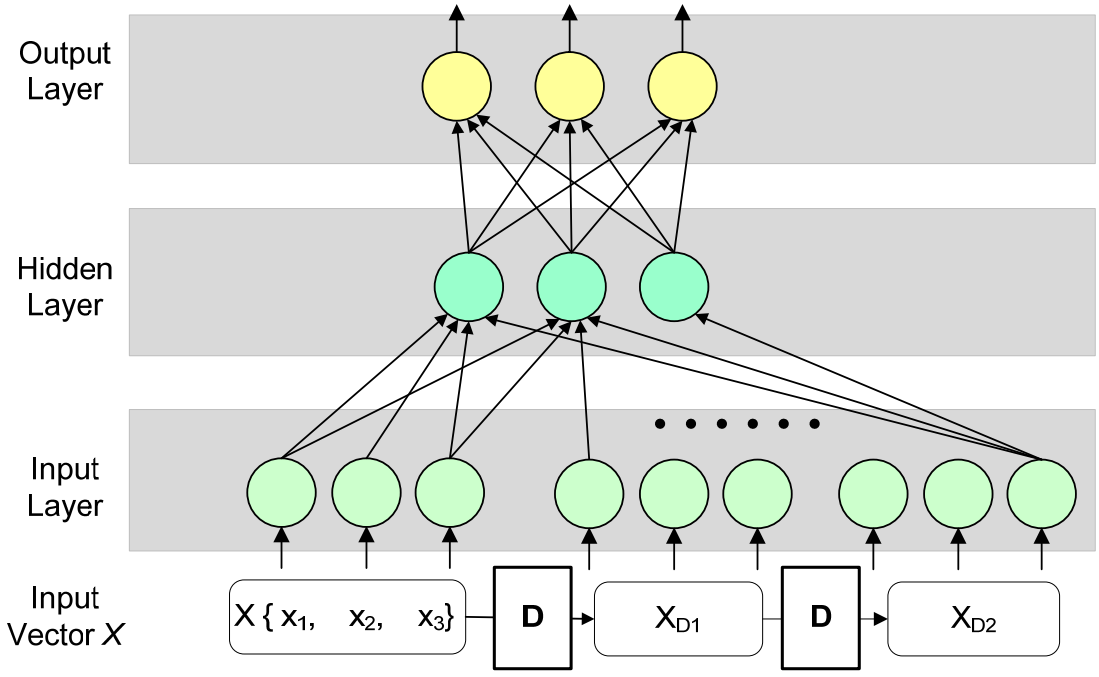

Figure 16: A Time Delay Neural Network [94]

RNNs allow the connections between arbitrary pairs of nodes, independent of their layer positions. These extended connections include the self-recurrent loop of a node and backward connections to previous layers [1,61,102]. Similar to TDNNs, RNNs were invented to deal with the temporal dependency of network inputs. Figure 17 illustrates an example of this network. The outputs of the hidden nodes are used as context inputs, and the input vector along with context inputs are input into hidden nodes in the next process cycle [64].

Remarkable results were obtained for the task of phoneme recognition based on this network. The phoneme recognition accuracy for the TIMIT database on a reduced 39 phone set is 69.8%, which compares favorably with the results of using standard HMMs [81].

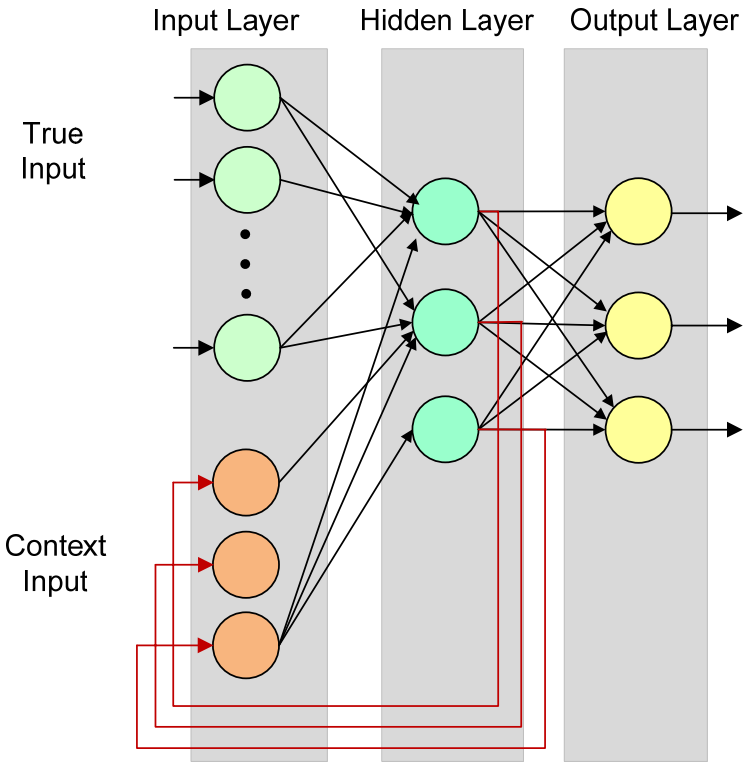

Figure 17: A simple recurrent neural network [64]

In this dissertation, feed-forward neural networks are investigated to explore the combination of these networks with HMMs as well as the possibility for feature dimensionality reduction. Therefore, the neural networks used in the remainder of this dissertation are feed-forward neural networks.

4.2.2 Recognition and Back-propagation Network Training

As described in Section 2.3.2, the outputs of a feed-forward neural network are calculated as the nonlinear function of the weighted sums of the hidden nodes, and the hidden nodes are further determined using the input nodes and the corresponding weights in the same manner. For such a network with input, hidden and output layers, the relationship between the output z_k for the kth output node and the inputs x can be generalized as:

$$z_k = f\left(\sum_{j=0}^{n} w_{kj} f(net_j)\right) = f\left(\sum_{j=0}^{n} w_{kj} f\left(\sum_{i=0}^{d} w_{ji} x_i\right)\right), \qquad \text{(IV.1)}$$

where w_{kj} denotes the weight from node k to node j and $f()$ is the nonlinear function used in each node. The number of nodes in the hidden layer is n and the number in the input layer is d. Furthermore, the summation operation of the hidden nodes in Equation IV.1 can be repeated to express a network with more than one hidden layer.

For recognition such as isolated phoneme classification, the input layer is generally designed to accept speech features, thus the number of the input nodes is the dimension of the feature vector. The number of the output nodes is mainly controlled by the number of recognition categories, with one node for each category. The node with the maximum output value indicates the category to which that input vector is classified.

The number of hidden layers and their nodes are independent of the input feature dimensions and the nature of the recognition categories. Although it has been shown that one hidden layer with a sufficient number of nodes is able to present any continuous function from input to output [19], in practice, the use of multiple hidden layers is favored by many ASR systems, especially for complex recognition tasks.

In general, the activation function is desired to be nonlinear, be continuously differential, and saturate. Other properites required for the function are continuity and smoothness so that it can be defined throughout the range of its argument. For the above reasons, the sigmoid is the most popular choice for the use as the activation function [19]. The basic form of a sigmoid function given an input *net* is:

$$f(net) = a\tanh(b\ net) = a\left[\frac{e^{+b\,net} - e^{-b\,net}}{e^{+b\,net} + e^{-b\,net}}\right], \qquad \text{(IV.2)}$$

where a and b are constant variables.

Moreover, the different activation functions can be used in a network. For example, the functions in the output layer are allowed to be different from the ones in the hidden layer, and even each individual node can be assigned a different activation function.

The training process for neural networks is designed to adjust weights to minimize error between output nodes and the neural targets for inputs. The most successful training algorithm is the Back-Propagation algorithm, which looks to minimize the error function in weight space using the method of gradient descent [83,19]. A brief explanation of the algorithm is presented below.

First, a training error function $J(W)$ between the desired output t_k and the actual output z_k is defined as:

$$J(W) = \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|T - Z\|^2 \ , \tag{IV.3}$$

where $T=\{t_1,\ldots, t_c\}$ and $Z=\{z_1,\ldots, z_c\}$ are the target and network output vectors of length c and W represents all the weights in the network.

The back-propagation learning rule is based on gradient descent. The weights are initialized with random values, and updated with a learning rate of η to reduce the error as:

$$\Delta W = -\eta \frac{\partial J}{\partial W} . \tag{IV.4}$$

The updating for a weight vector at iteration m is:

$$W(m+1) = W(m) + \Delta W . \tag{IV.5}$$

The calculation is repeated for each weight of the network until the weight updating drops below a threshold.

In the calculation of weight update ΔW, the hidden-to-output weights are first considered, the derivative of the error J in respect to the weights w_{kj} from the hidden layer to the output layer becomes:

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = -\delta_k \frac{\partial net_k}{\partial w_{kj}} \ , \tag{IV.6}$$

where net_k is the activation of the kth output node. The sensitivity δ_k of the node can be defined as the overall error against the net activation of the node as:

$$\delta_k = -\frac{\partial J}{\partial net_k} = \frac{\partial J}{\partial z_k}\frac{\partial z_k}{\partial net_k} = (t_k - z_k) f'(net_k) . \tag{IV.7}$$

The last derivative in Equation IV.6 can be rewritten as:

$$\frac{\partial net_k}{\partial w_{kj}} = y_j . \tag{IV.8}$$

Therefore, the weight update for the hidden-to-output nodes is:

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (t_k - z_k) f'(net_k) y_j . \tag{IV.9}$$

Similarly, the update for the input-to-hidden nodes can be derived using the chain rule to compute:

$$\Delta w_{ji} = \eta x_i \delta_j = \eta \left[\sum_{k=1}^{c} w_{kj} \delta_k \right] f'(net_j) x_i , \tag{IV.10}$$

where x_i is the input for the ith input node and c the number of hidden nodes.

4.3 Hybrid NN/HMM Models

In order to overcome the drawbacks of neural networks and HMMs, in the early 1990s, researchers began to explore the idea of combining HMMs and neural networks within a single novel model, widely known as hybrid HMM/NN models. Some researchers have thoroughly investigated these hybrid methods and have provided several good readings on this approach [94,66,7]. In this section, the following major uses of neural networks for HMMs are summarized.

4.3.1 Neural Networks to Emulate HMMs

Early approaches on hybrid models attempted to emulate HMMs by way of incorporating the dynamic programming algorithm into neural networks. Lippmann and Gold [59,57] implemented a recurrent neural network to behave as the Viterbi algorithm. However, the so-called Viterbi net was not assigned an actual training procedure and the

parameters were initialized through the corresponding HMMs. In a model called Alpha Net [9], a similar recurrent neural network was used to execute the forward-backward algorithm as in HMMs. This architecture resembles the forward computation of the alphas in the Baum-Welch algorithm, and adapts the backward procedure of the algorithm for the parameter learning [94].

Although these early works demonstrated that neural networks can be implemented to act as HMMs for speech recognition, the improvement in recognition performance was not significant.

4.3.2 Neural Networks as Vector Quantizers for Discrete HMMs

Since discrete HMMs require a clustering technique to convert continuous feature vectors into discrete symbols for the determination of the state output probabilities, a number of researchers applied neural networks to the problem of generating codebooks for discrete HMMs. Most of those works relied on Kohonen's Learning Vector Quantization (LVQ) as an effective neural alternative to standard clustering algorithms.

A good example of this approach is a hybrid system proposed by Rigoll et al. [78] for speaker independent continuous speech recognition in a large vocabulary task. The system is trained by an information theoretic based algorithm to maximize the mutual information between the network outputs and the phone descriptions. Neural networks are employed to perform vector quantization on the acoustic features. The statistical distributions of the VQ labels are then modeled using HMMs with discrete output probability distribution functions. A decrease of 35.5% in Word Error Rate (WER) over a classical VQ based HMM system was achieved in this hybrid system.

Jang and Un [42] presented a hybrid system for isolated word recognition based on the same approach. Time delay neural networks and HMMs are combined using the activations from the second hidden layer of the neural as the outputs of a fuzzy vector quantizer (FVQ). The HMM algorithm is modified to accommodate these FVQ outputs. The experimental results for an isolated Korean word database showed a 44.9% WER reduction with respect to standard discrete HMMs.

4.3.3 Neural Networks to Estimate State Posterior for HMMs

In this approach, neural networks are used to estimate state posterior probabilities for continuous density HMMs in place of Gaussian Mixture models. Bourlard and Morgan [8] first proposed this architecture and trained neural networks to perform a non-parametric estimate of the posterior probability of an HMM state. This hybrid system has been applied to the SRI's DECIPHER system and a 5.8% WER was achieved as opposed to an 11% rate with standard HMMs.

In the so-called Hidden Neural Networks (HNN) model introduced by Riss and Krogh [79,80], the standard HMM probability parameters are replaced by the outputs of neural network. To ensure a probabilistic interpretation, the HNN is normalized globally as opposed to the local normalization on parameters in standard HMMs. Furthermore, all parameters in the HNN are estimated simultaneously according to the discriminative conditional maximum likelihood criterion.

The advantages of these hybrids are the relative ease of implementation, due to a training scheme that relies on alternating between Back-propagation and a standard Viterbi alignment, as well as a discriminative training criterion for improved recognition

performance. However, the weakness of the approach is the lack of a well defined global optimization scheme at the system level.

4.3.4 Neural Networks as Feature Transformers

The continuous density HMMs assume that the speech features are uncorrelated and Gaussian distributed, but the actual features generally do not follow these assumptions. In this hybrid approach, neural networks are used to reduce the correlations among the features and transform the features into lower dimensional representations so as to be better modeled by HMMs. For example, in the work of Hermansky et al. [36], a neural network was combined with HMMs in a tandem structure as a preprocessing, called tandem acoustic modeling. Having nonlinear transformation ability, neural networks in this approach are able to transform a vector of acoustic features into a more effective representation for continuous HMMs.

Meanwhile, a number of researchers used the simultaneous estimation of all parameters for both the HMM and ANN according to global optimization criteria. Bengio et al. [3] attempted to perform global optimization by driving the neural network gradient descent with parameters computed in the HMMs. In the work of Johansen and Johnsen [44], the global optimization technique was used to train a neural network as an extractor of a joint input feature for an HMM relying on the Maximal Mutual Information (MMI) criterion [82].

Furthermore, Somervuo [90] has compared this nonlinear feature transformation based on neural networks with other linear transformation methods. Experimental results showed that the features following the neural network transformation were able to significantly improve HMM performance.

4.4 Tandem NN/HMM model

4.4.1 Structure Overview

As described in Section 3.3, the acoustic modeling of HMMs using GMMs assumes that speech features are Gaussian as a Gaussian distribution is used to represent each state. In addition, each dimension of the speech feature vector is assumed to be independent when diagonal covariance matrices are used in GMMs. Another limitation with HMMs is the use of non-discriminative training which results in poor discrimination among the models.

In order to overcome these limitations, this chapter focuses on the strong discrimination power of neural networks and investigates the use of a neural network as a feature transformer for an HMM based recognition system. A feed-forward neural network is employed to perform a nonlinear transformation of speech features. The discriminatively trained neural network assists in maximizing the discrimination of speech features, and the correlation among features can be reduced by using the network trained with orthogonal targets. Thus, the neural network outputs, which are used as the transformed features, form an uncorrelated and highly discriminative representation of the original ones for the recognition based on HMMs.

The architecture of the neural network based feature transformation for HMM phone recognition is illustrated in Figure 18. As the first step, a discriminatively trained multilayer neural network performs a feature transformation on the input speech features and transformed features are produced at the output layer. The transformed features are then fed into an HMM based phonetic recognizer and encoded into phonemes with the incorporation of a phone dictionary and a bigram language model.

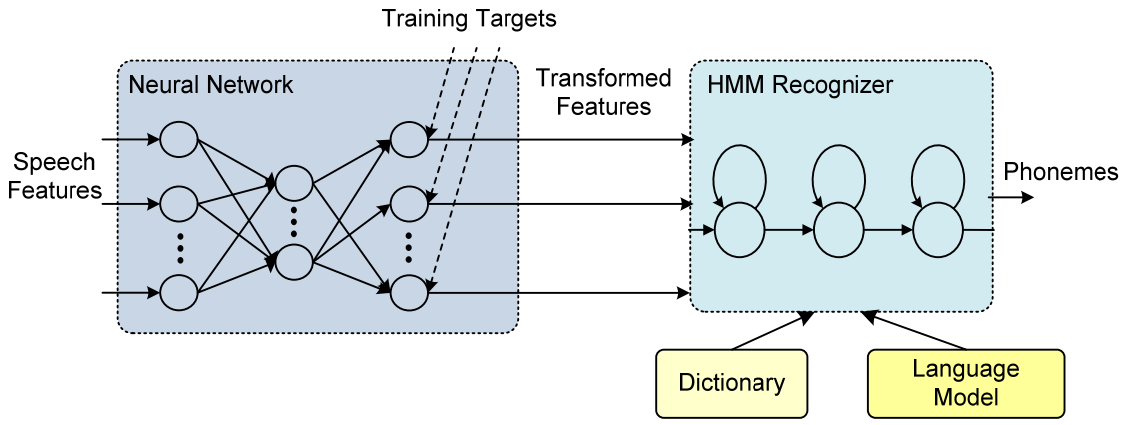

Figure 18: Overview of the neural network based discriminative feature transformation

This method uses the neural networks as a form of preprocessing. The learning of the neural network is conducted based on the training targets obtained from the data transcriptions, independent of the training process of HMMs. The feature transformation of the network is also independent of the HMM process. Thus, the neural network in this method results in a simple and fast process, and contains the flexibility and potential to combine the network with other processing methods.

The neural network includes an input layer, hidden layers and an output layer. The numbers of nodes contained in the input and output layers respectively correspond to the dimensions of the input features and the number of categories in the training target data. In the case for which there are fewer output nodes than input nodes, the network reduces feature dimensionality.

In optimizing the design of a neural network, an important consideration is how to determine the number of hidden layers and an appropriate number of hidden nodes in each layer. A neural network with no hidden layers can form only simple decision regions, which is not suitable for highly nonlinear and complex speech features. Although it has been shown that a neural network with a single hidden layer is able to represent any

function with a sufficient number of hidden nodes [19], the use of multiple hidden layers generally provides flexible configuration such as distributed deployment of hidden nodes, and diverse nonlinear functions for different layers. Of the number of hidden nodes, a small number reduces the network's computational complexity. However, the recognition accuracy is often degraded. The more hidden nodes a network has, the more complex a decision surface can be formed, and thus better classification accuracy can be expected [64]. Generally, the number of hidden nodes is empirically determined by a combination of accuracy and computational considerations, as applied to a particular application. Later in this chapter, experimental results reflect recognition accuracy as a function of the number of hidden layers and hidden nodes.

Phoneme HMMs are used as the recognizer for phonetic experiments, although other recognition units could be used. In the calculation of the emission probability for each state in an HMM, the transformed features are used instead of the original features. After emission and transition probabilities are estimated, the Viterbi algorithm is used to calculate the overall probabilities of a feature vector over HMMs and to determine the model with highest probability as the recognition result.

The parameters of the HMMs are trained by the Baum-Welch estimation algorithm as described in Section 3.3.2. Some global optimization training methods have been used in hybrid NN/HMM recognition models [3,77]. However, in this chapter, the weight training of neural network is conducted independently from the HMM training.

4.4.2 Network Training

A difficulty in neural network training is that the input data has a wide range of means and variances for each feature component. The variability in ranges can lead to

difficulties in training. In order to avoid this, the input data should be scaled so that all feature components have the same mean and variance.

In this dissertation, the input feature vector **x** at time *i* is scaled using:

$$\mathbf{o}_i = \frac{\mathbf{x}_i - \boldsymbol{\mu}}{r\boldsymbol{\sigma}}, \tag{IV.11}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\sigma}$ is the standard deviation vector of input features. The scale factor *r* was set to 5. By using this scaling, the mean of each scaled vector component of **o** is 0.0, and the standard deviation of each component of **o** is 0.2, thus resulting in a range of approximately [-1, 1] for all the scaled feature vectors.

The mean vector $\boldsymbol{\mu}$ and standard deviation vector $\boldsymbol{\sigma}$ were computed from the training data. However, both training and test data were scaled with the same mean and standard deviation vectors.

The training of the neural network requires category information for creating training targets. These targets are also desired to be uncorrelated and suitable for quicker convergence of weight update. In this work, phoneme category information is used to create training targets. The dimensionality of the targets is equal to the number of phone categories, with a value of "1" for the target category and "0s" for the non-target categories.

For the case of 48 phone categories used for the neural network training, the target vectors are 48 dimensions and each vector consists of only one peak value to indicate the category. Figure 19 illustrates a sequence of training targets for several phonemes using 48 categories.

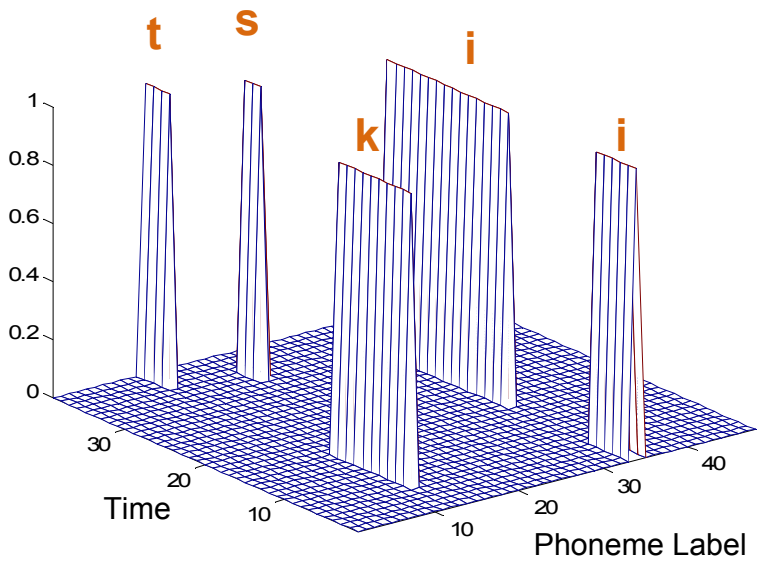

Figure 19: Training target vectors of the neural network

Different category information can be used for the NN training as compared to the categories used for HMM training since the training of the network is independent of the HMM training. For example, as an alternative to the use of phoneme categories, phone group (e.g. vowel, stop, fricative) categories can be used to train the network for maximizing the discrimination among these groups.

The weights of the neural network are estimated using the Back-Propagation algorithm to minimize the distance between the scaled input features and target data. The update of the weight in each layer depends on the activation function of that layer, thus the network learning can be designed to perform different updates when dissimilar activation functions are used.

Figure 20 illustrates an original feature set of 78 DCTC-DCSCs, the scaled features, the training target data and the outputs of the trained neural network.

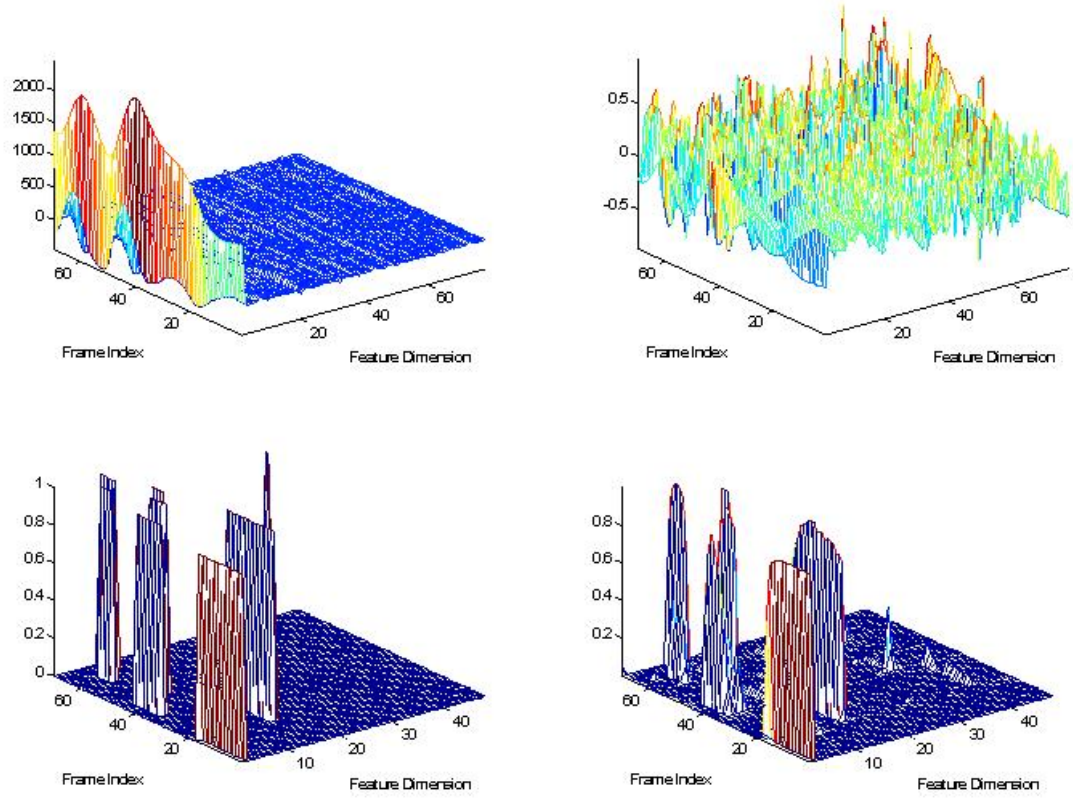

Figure 20: The illustrations of original features (top left), scaled features (top right), training target data (bottom left), and network outputs (bottom right)

4.4.3 Network Layer Nonlinearity

The nonlinearity is the activation function of a node. It performs a nonlinear transformation on the weighted sum of the inputs, and determines the output of the node. As described in Section 4.2.2, the activation function should be nonlinear, continuously differentiable, and it should saturate. In this section, the following popular activation functions are discused:

1. Linear activation function;
2. Unipolar sigmoid function;
3. Bipolar sigmoid function, and;

4. Softmax function.

The linear activation function, also referred to as an identity activation, produces an identical output to its input:

$$f_0(net) = net. \tag{IV.12}$$

The weight update is generally a constant value, for example, an update of 0.5 is used in this dissertation.

If the linear function is used throughout the network, then the network configurations are equivalent to some linear regression models [46]. As a result, it is uncommon to use linear functions in all nodes of a network.

The unipolar sigmoid function maps the node input to a range of values between [0, 1]:

$$f_1(net) = \frac{1}{1+e^{-net}}. \tag{IV.13}$$

The derivative of the function for weight update in the training is:

$$f_1'(net) = \frac{-e^{-net}}{(1+e^{-net})^2} = f_1(net)(1-f_1(net)). \tag{IV.14}$$

The bipolar sigmoid function maps the input into the range of [-1, 1] :

$$f_2(net) = \frac{1-e^{-net}}{1+e^{-net}}. \tag{IV.15}$$

The derivative of the bipolar sigmoid function is:

$$f_2'(net) = \frac{2e^{-net}}{(1+e^{-net})^2} = \frac{1}{2}(1-f_2(net)^2). \tag{IV.16}$$

The softmax function takes all the nodes in a layer into account and calculates the output of a node as:

$$f_3(net_i) = \frac{e^{net_i}}{\sum_{j=1}^{c} e^{net_i}}, \tag{IV.17}$$

where net_i is the activation of node i and c the number of nodes in the layer. By assigning a softmax activation function to the output layer of the neural network for categorical target variables, the outputs can be interpreted as posterior probabilities.

The weight update is based on the derivative of the function:

$$f_3^{'}(net) = f_3(net)(1 - f_3(net)) \ . \tag{IV.18}$$

A brief proof of Equation IV.18 starts with a rewritten form of the function:

$$f_3(net_i) = \frac{e^{net_i}}{A + e^{net_i}}, A = \sum_{j=1 \wedge j \neq i}^{c} e^{net_j} \ . \tag{IV.19}$$

Using the Quotient rule of the derivative:

$$f^{'}{}_3(net_i) = \frac{(e^{net_i})'(A + e^{net_i}) - (e^{net_i})(A + e^{net_i})'}{(A + e^{net_i})^2}. \tag{IV.20}$$

The equation can be rewritten as:

$$f^{'}{}_3(net_i) = \frac{e^{net_i}(A + e^{net_i}) - e^{net_i} e^{net_i}}{(A + e^{net_i})^2} = \frac{e^{net_i}}{A + e^{net_i}}(1 - \frac{e^{net_i}}{A + e^{net_i}}). \tag{IV.21}$$

It is clearly shown that Equation IV.18 is obtained by substituting Equation IV.20 into Equation IV.21.

Figure 21 shows the illustrations of the linear, and the two sigmoid functions.

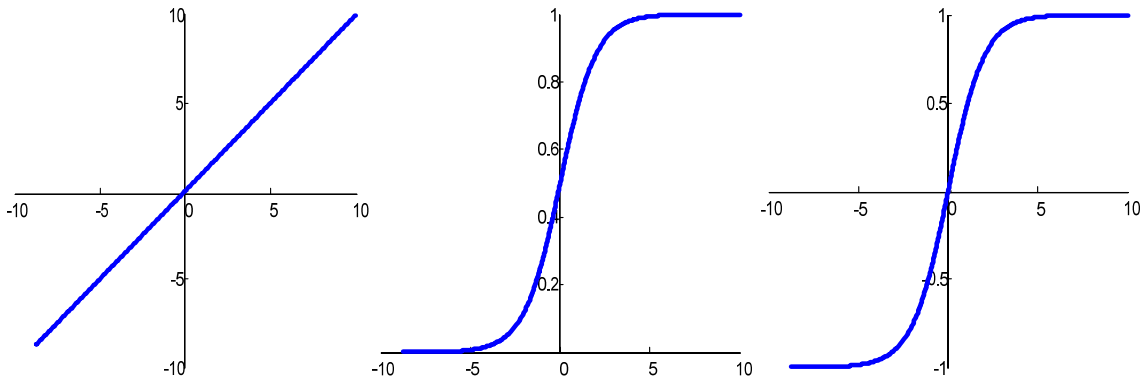

Figure 21: Illustrations of a linear activation function (left), a unipolar sigmoid function (middle) and a bipolar sigmoid function (right)

Another important consideration in selecting an activation function is that the activation function should match the characteristic of the input or output data. For example, with training targets assigned the values of "0" and "1", a sigmoid function with the outputs in the range of [0, 1] is a good candidate for the output layer. When the outputs of the network are to be used as transformed features for the HMM recognition, a linear function or a softmax function is appropriate to generate the data with a more diverse distribution, such as one that would be well-modeled with a GMM. Moreover, equipped with various nonlinearities, the neural network is expected to have a stronger discriminative capability and thus it is enabled to cope with more complex data.

The behavior of a neural network with nonlinear functions is difficult to prove theoretically. Therefore, experiments were conducted to investigate various nonlinearities in different layers and even different nonlinearities in the same layer depending on whether they are in the training or transform process.

4.4.4 Principal Component Analysis (PCA) for Feature De-correlation

As described in Section 3.3, for the covariance matrices of GMMs in HMMs, due to the computational cost for estimating parameters, diagonal covariance matrices are generally used instead of full covariance versions. Thus, the components in a feature vector are assumed to be uncorrelated with each other. In order to reduce the correlation of the network outputs and improve the match to a GMM, Principal Component Analysis (PCA) processing is often applied to the output of the neural network [20,36]

PCA analysis is a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for the majority of the variability in the data, and each succeeding component uncovers as much as possible of the remaining variability. The principal components are ordered in terms of variance, with the first principal component accounting for the most variance and thus generally considered the most important principal component.

Additionally, since PCA is a good dimensionality reduction method, the dimension of the network outputs can be reduced using PCA to obtain more compact and effective transformed features.

4.5 Experimental Evaluation

4.5.1 Experimental Setup

Several experiments based on the TIMIT database were conducted to investigate the proposed hybrid method using different network configurations and nonlinearities. As described in Section 3.4, the SA sentences were removed from the database, resulting in

3696 sentences for the training and 1344 sentences for the test. The original TIMIT 62 phone set was mapped to the reduced 48 phone set.

For both training and testing data, the best DCTC-DCSC features obtained in the experiments of Section 3.4, were 78 DCTC-DCSC features (13 DCTCs × 6 DCSCs) with medium spectral resolution and long temporal information, and were extracted as original features. These temporal-spectral features used DCTCs for representing spectral information and DCSCs for capturing spectral trajectories. The 13 DCTCs were computed using a frame of 8 ms with a spacing of 2 ms, and a warping factor of 0.5. The block length used for computing 7 DCSCs was 500 ms (250 fames), the block spacing was 8 ms (4 frames) and the time warping factor was 12.

Similar to Sections 3.4, left-to-right Markov models with no skip were used and a total of 48 monophone HMMs were created from the training data using the HTK toolbox. The phone bigram information extracted from the training data was used as the language model. Various numbers of states and mixtures were evaluated and the details are given in the description of the experiment in Section 4.5.4.

Neural networks with 1 hidden-layer and 3 hidden-layers were evaluated in the experiments to explore the effectiveness of neural network for feature transformation. The numbers of nodes in the input and output layers were 78 and 48 respectively, with 78 corresponding to the dimensionality of the original features and 48 determined by the number of phone categories used in the training targets. The various numbers of hidden nodes in each layer were tested in order to find the best configuration of neural networks. The weights of the networks were trained using the back-propagation algorithm in

Section 4.2.2 and the training targets in Section 4.4.2 created from the phone transcriptions of the training data.

4.5.2 Experiment with Various Nonlinearities

Since the behavior of a neural network is difficult to prove theoretically due to the use of nonlinear functions, this experiment was conducted to examine the network nonlinearity by using different combinations of activation functions.

Neural networks with 1 hidden-layer (500 nodes) and 3 hidden-layers (300-200-300 nodes) were used for performing a feature transformation. 3-state HMMs with each state represented by 3 mixtures were used to recognize phone using the transformed features. The PCA process was also applied to lessen the correlation of the network outputs and reduce the feature dimensionality.

The combinations of activation functions with the best accuracies based on the 1 hidden-layer network are listed in Table 4, and the results of the 3 hidden-layer network are listed in Table 5. The linear, unipolar sigmoid, and bipolar sigmoid functions as presented in Section 4.4.3 are indicated by "0", "1", and "2" respectively. The Softmax function, denoted by "4", was utilized in the output layer for producing posterior probabilities. For example, "1-0" represents using a unipolar sigmoid function in the hidden layer and a linear function in the output layer. "NN+PCA" means that the output of a neural network is further processed using PCA for the purpose of de-correlation.

For the 1 hidden-layer network, the nonlinearity of "1-1" in training and "1-0" in transformation produced the highest accuracy of 62.0%, which was much higher than "1-1" in both training and transformation. That implies that the use of linear output layers in the transformation results in more appropriate features for HMMs, or apparently features

better modeled as Gaussian mixture models. The use of the Softmax function in the output layer for feature transformation (i.e., "1-4") also outperformed the bipolar sigmoid function, but was not nearly as effective as using a linear output layer.

Table 4: Recognition accuracies with various nonlinearities based on 1 hidden-layer neural network.

Nonlinearities		Recognition Accuracies (%)	
Training	Transform	NN	NN+PCA
1-1	1-1	47.54	63.93
	1-0	**61.99**	**64.49**
	1-4	58.42	62.39
2-1	2-0	38.39	57.60
	2-1	60.67	62.35
	2-4	47.18	51.87

Table 5: Recognition accuracies with various nonlinearities based on 3 hidden-layer neural network.

Nonlinearities		Recognition Accuracies (%)	
Training	Transform	NN	NN+PCA
1-1-1-1	1-1-1-1	60.09	64.43
	1-1-1-0	**68.87**	**70.00**
	1-1-1-4	60.82	63.71
2-2-2-1	2-2-2-2	38.52	50.44
	2-2-2-0	64.67	65.23
	2-2-2-4	44.45	56.79

The neural network with the bipolar sigmoid function in the hidden layer achieved a similar performance to that of the unipolar version when the linear output layer was used in transformation, for example, 60.67% versus 61.99%. The use of the bipolar sigmoid function generally didn't show any material advantage.

The superiority of using linear output layers is also found in the results based on the 3 hidden-layer network. The highest accuracy of 68.87% was obtained in the case of "1-1-1-1" in training and "1-1-1-0" in transformation, where the output layer of the network was changed to linear in transformation.

Moreover, the recognition accuracies were further improved by approximately 3% with the incorporation of PCA, illustrating the effectiveness of PCA in reducing the correlation of speech features.

4.5.3 Experiment with Various Neural Network Configurations

The second experiment was conducted to explore various configurations of the neural network such as the number of hidden layers and nodes in each unique layer. Figure 22 shows the accuracies of 1 hidden-layer and 3 hidden-layer neural networks using various numbers of nodes in each hidden layer. The unipolar sigmoid function was employed in all layers in the training of the network, but both the sigmoid and linear functions were used in the feature transformation for the purpose of comparison.

The recognition accuracy increases with an expanding number of hidden nodes used in the network. The 3 hidden-layer network leads to better performance compared to the 1 hidden-layer version, even with a fewer total number of hidden nodes. For example, the 3 hidden-layer network with 100-50-100 nodes (a total of 250 nodes) contributed to an accuracy of 65.25%, which is higher than 63.92% obtained with the 1 hidden-layer network of 300 hidden nodes. The complexity of this 3 hidden-layer network in terms of number of weights is 22600 (78×100+100×50+50×100+100×48) versus 37800 (78×300+300×48) of the 1 hidden-layer network. Similar to the previous experiment, for both networks, the higher results were always obtained using linear output layers.

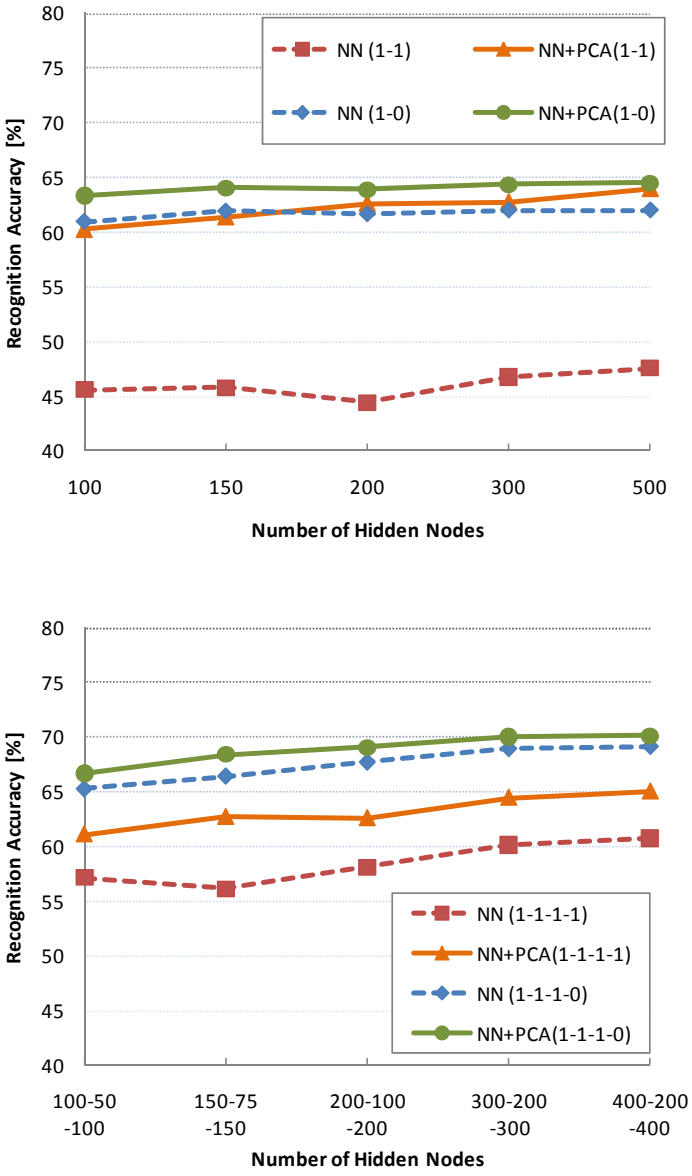

Figure 22: Accuracies of 1 hidden-layer (top panel) and 3 hidden-layer (bottom panel) neural networks using various numbers of hidden nodes

This experiment reveals that 3 hidden-layer neural networks give stronger transformation power and more flexible configuration than in the 1 hidden-layer case. Thus similar performance can be achieved with fewer nodes in a 3 hidden-layer network.

4.5.4 Experiment with Various HMM Configurations

In the third experiment, the transformed features obtained with the best nonlinearity in the previous experiments, which were "1-1-1-1" in training and "1-1-1-0" in transformation of the 3 hidden-layers network, were recognized with various HMM configurations. The PCA processed features were also evaluated as well as 32-dimensional PCA reduced features. The recognition accuracies based on 1-state and 3-state HMMs are shown in Figure 2 with respect to the number of mixtures in each state. "NN+P(k)" indicates that the network output features are further reduced to k dimensions using PCA.

For the case of 3-state HMMs, the transformed features after the PCA processing consistently lead to very high accuracies (above 69.0%) for all conditions, even when a very small number of mixtures are used in HMMs. Similar trends are observed using 1-state HMMs, where the PCA processed features exhibit much higher accuracies. The 32-dimensional PCA reduced features performed favorably with slight degradation when contrasted with the features without dimensionality reduction. The highest accuracy obtained with the PCA processed features is 72.04% using 3-state 64-mixture HMMs.

Compared with the original features, the direct outputs of the network lead to a large improvement using a small number of mixtures for both the 1-state and 3-state HMMs, although they didn't outperform the PCA processed features. The advantage decreases with an increasing number of mixtures. However, the superiority of the neural network diminished when a large number of mixtures (e.g., 64 mixtures) are used in HMMs.

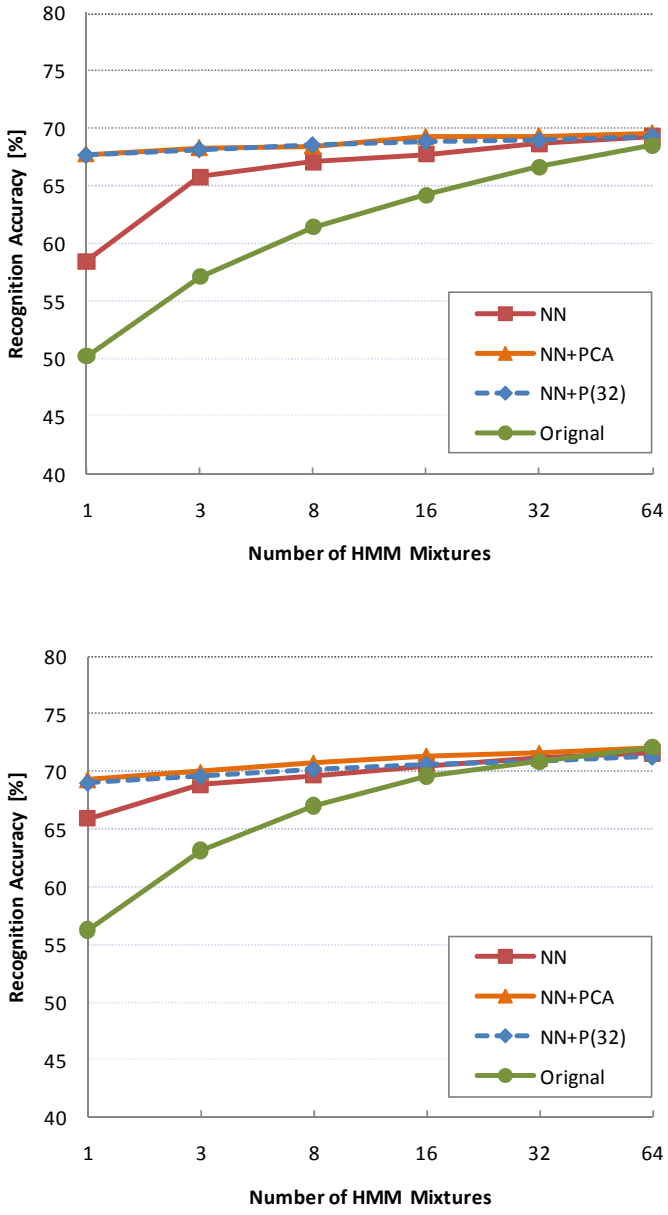

Figure 23: Accuracies of 1-state (top panel) and 3-state HMMs (bottom panel) using the transformed features with various numbers of mixtures

The results of this experiment imply that neural networks are able to transform speech features into a more effective representation, thus potentially simplifying the HMM configuration.

4.6 Conclusions

In this chapter, a neural network based feature transformation was presented in order to achieve a highly accurate HMM-based phonetic recognition system. In this hybrid NN/HMM approach, a neural network was employed to increase the discrimination and reduce the correlation of speech features so that the transformed features can be better modeled by GMM based HMMs. This feature transformation method applied various nonlinear functions in the layers of the network and even different nonlinearities in the same layer in different stages.

Experimental evaluations based on TIMIT database demonstrate:

1. The use of linear output layers produced better transformed speech features;
2. Compared with 1 hidden-layer neural networks, the networks of 3 hidden-layers showed more flexibility in deploying nodes and arranging nonlinear functions as well as stronger transformation power;
3. The incorporation of PCA is able to further reduce feature correlation as well as reducing dimensionality, and;
4. Very high recognition accuracies with the network transformed features were obtained, especially when a small number of states and mixtures were used for HMM phonetic models. In the case of a 3-state 3-mixture HMM recognition experiment, the accuracy obtained with the transformed features is about 70.0%, which is approximately 7% higher than that with the original features.

However, the superiority of the neural network was not observed when a large number of mixtures (e.g., 64 mixtures) were used in HMMs.

Chapter V Nonlinear Discriminant Analysis Based Dimensionality Reduction

5.1 Introduction

One of the main practical difficulties for ASR is the large dimensionality of acoustic feature spaces. Recent research has shown that the features with high spectral resolution and long temporal expansion, which result in high dimensional speech representation, contribute considerably to high recognition accuracy. Another cause for large feature dimensionality is that the combination of features from different sources can compensate for each other to construct a more robust feature set with the potential for higher accuracy and more resistance to noise. However, high dimensional features present many challenges and problems, collectively referred to as the curse of dimensionality [18]. One of the problems is that not all the feature components are important for understanding the fundamental phenomena of interest. In some cases of pattern classification, classification results on test data even decrease as the dimensionality increases. The high dimensionality of speech features also raises the computational cost of model training and restricts the choice of processing methods.

A statistically optimal strategy of dimensionality reduction is to project the data on a lower-dimensional subspace that captures as much of the variation of the data as possible. Many linear techniques, most notably Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) and several variants have been used to reduce dimensionality along with this strategy [51,55,84,114,98,23]. However, these orthogonal rotations of the feature space are suboptimal when data is distributed primarily on curved subspaces embedded in the higher dimensional feature spaces.

Nonlinear PCA (NLPCA) presented in the work of Kramer [54] is an extension of linear PCA, which allows for nonlinear mapping between data and its principal component. NLPCA is based on a feed-forward neural network to perform the identity mapping, where the network inputs are reproduced at the output layer. The activation of the internal bottleneck layer, which has fewer nodes than the input layer, is used as a compact representation of the input data. Although NLPCA has been demonstrated to be more powerful than linear methods in a variety of classification problems such as meteorology and oceanography [38], its applications to speech recognition are rarely used and reported.

In this chapter, two NonLinear Discriminant Analysis (NLDA) methods based on neural networks are presented for nonlinear dimensionality reduction of speech features. In contrast with NLPCA, the neural networks proposed are trained as feature classifiers to reduce feature dimensionality as well as maximize discrimination among speech features. The outputs of different network layers are used for obtaining transformed features. Moreover, the training of the neural networks uses the category information that corresponds to a state in HMMs so that the trained networks can better accommodate the temporal variability of features and obtain more highly discriminative features in a low dimensional space. Portions of this chapter were presented in the works of Hu and Zahorian [41,39,116].

The remainder of this paper is organized as follows; Section 5.2 describes linear dimensionality reduction methods including PCA and LDA. As an extension of linear methods, nonlinear NLPCA is presented in Section 5.3. In section 5.4, two NLDA dimensionality reduction methods are proposed. The details for the network training, such as using methods such as state-level training targets, are also presented. Section 5.5

summarizes the evaluation of the proposed approaches using the TIMIT database with various neural network and HMM configurations. A literature comparison and the conclusions are given in Section 5.6 and Section 5.7 in respectively.

5.2 Linear Dimensionality Reduction Methods

5.2.1 Principal Component Analysis (PCA)

As briefly introduced in Section 4.4.4, PCA is a well-established technique for feature extraction and dimensionality reduction, which is also known as singular value decomposition (SVD), the Karhunen-Loeve transform or the Hotelling transform in various fields [22]. The analysis is based on the assumption that most information about classes is contained in the directions in which the variations are the largest [45].

The derivation of PCA can be viewed as a linear projection which rotates the axes to new positions in the space defined by the original variables. In the new rotation, there will be no correlation among the variables. The first principal component is the linear combination of projected variables that contains the greatest amount of variation. The second component defines the maximum amount of variations independent and orthogonal to the first one. There can be as many principal components as there are original data dimensions, although generally fewer components are used than original dimensions.

For a given p-dimensional data set **x**, the *m* principal components **y** can be computed using a p×m transformation matrix **T** defined as:

$$\mathbf{y} = [y_1,...,y_m] = [\mathbf{T}_1^T x,...,\mathbf{T}_m^T] = \mathbf{T}^T \mathbf{x}, \tag{V.1}$$

where the principal axes $\mathbf{T}_1, \mathbf{T}_2, \ldots \mathbf{T}_m$ are orthonormal axes in the projected space. These axes are generally calculated by the eigenvectors of the covariance matrix of all the samples, which is defined as:

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T, \tag{V.2}$$

where N is the number of the samples and $\boldsymbol{\mu}$ the sample mean vector. The principal axes $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_m$ are the m leading eigenvectors of $\mathbf{S}$:

$$\mathbf{S}\mathbf{T}_i = \lambda_i \mathbf{T}_i, \quad i \in i,...,m, \tag{V.3}$$

where λ_i is the ith largest eigenvalue of $\mathbf{S}$ [98].

An assumption made for PCA based dimensionality reduction is that most information found in the observation vectors is contained in the subspace spanned by the first m principal axes, where $m<p$. Therefore, each original data vector can be represented by its principal component vector with dimensionality m.

Figure 24 shows an example of the PCA processing using 2-dimensional data. This demonstration used the MATLAB program presented in the work of Yen [105]. A set of data as plotted in the right panel was processed by PCA to compute the principal axes. In the left panel, the solid line, the first principal axis, represents the direction of the first principal component. The dashed line is the second principal axis and lies perpendicular to the first one. For the data with more than two dimensions, each succeeding component is always perpendicular to all the predecessors, and along the line of next greatest variation.

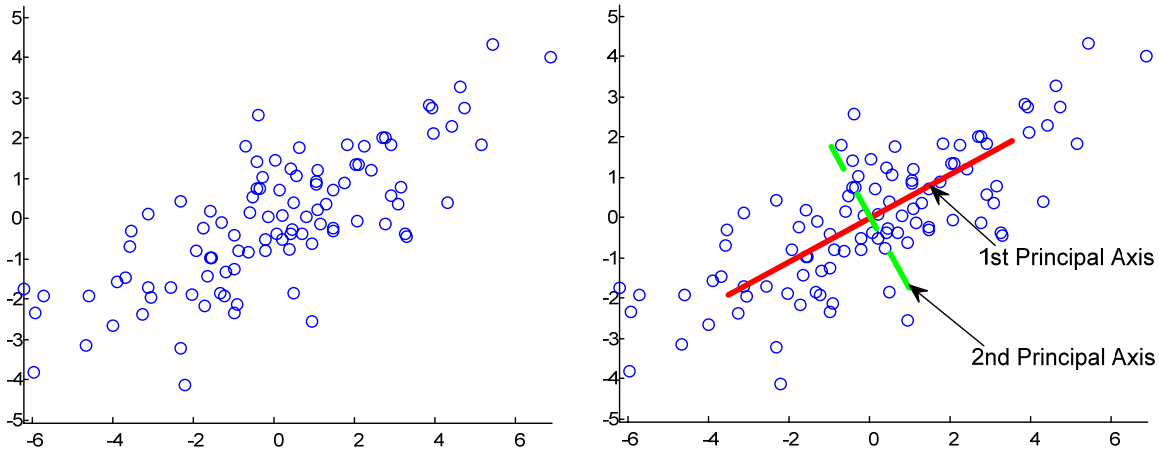

Figure 24: Plots of a set of 2-D data (left panel) and the data with principal axes obtained by PCA (right panel)

As a superior linear operator for dimensionality reduction, PCA is often used for signal representation and speech compression. For example, Takiguchi and Yasuo [92] applied PCA to MFCCs with the aim of removing additive noise and distortion for robust speech feature extraction. In this method, the main speech element is projected onto low-order features using PCA, while the noise or distortion element is transformed into high order features. It was shown that the use of kernel PCA instead of DCT provided better recognition performance for reverberant speech. In the work of Nhan and Lee [69], data-driven PCA was used to experimentally determine critical bandwidth and filter-bank coefficients in the calculation of MFCCs as described in Section 2.2.3. However, PCA is rarely very beneficial for speech recognition, since the PCA transformation is not intended for dealing with classification problems.

5.2.2 Linear Discriminant Analysis (LDA)

LDA is another commonly used technique for dimensionality reduction as well as pattern classification. In contrast to PCA, LDA preserves the discriminant information of

original features and linearly transforms them for the purpose of reducing feature variation in the same class and increasing the discrimination between classes. Instead of a global covariance matrix used in PCA, LDA requires two covariance matrices: a within-class covariance matrix and a between-class covariance matrix. The within-class covariance matrix is the local covariance of the samples from the same class, while the between-class matrix is the covariance mean of all the samples. Therefore, the ratio of the between-class variance to the within-class variance can be maximized in attempting to obtain the greatest separability among classes.

Suppose there are K classes, $C_1, C_2, \ldots, C_K$ and the ith data vector from the C_j is x_{ji}, where $0 < i < N_j$. N_j is the number of samples from class j.

The within-class covariance matrix $\boldsymbol{S}_W$ and between-class covariance matrix $\boldsymbol{S}_B$ are:

$$\mathbf{S}_W = \sum_{j=1}^{K} \mathbf{S}_j = \sum_{j=1}^{K} \frac{1}{N_j} \sum_{i=1}^{N_j} (\mathbf{x}_{ji} - \boldsymbol{\mu}_j)(\mathbf{x}_{ji} - \boldsymbol{\mu}_j)^T , \tag{V.4}$$

$$\mathbf{S}_B = \frac{1}{N} \sum_{j=1}^{K} N_j (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T , \tag{V.5}$$

where $\boldsymbol{\mu}_j$ is the mean vector of class j and $\boldsymbol{\mu}$ is the global mean vector.

A vector $\mathbf{x}$ on the transformed space is obtained using a linear projection matrix $\mathbf{T}$:

$$\mathbf{y} = \mathbf{T}^T \mathbf{x} . \tag{V.6}$$

Thus, the corresponding within-class and between-class covariance matrices become:

$$\widetilde{\mathbf{S}}_W = \mathbf{T}^T \mathbf{S}_W \mathbf{T} \tag{V.7}$$

$$\widetilde{\mathbf{S}}_B = \mathbf{T}^T \mathbf{S}_B \mathbf{T} . \tag{V.8}$$

A linear discriminant is then defined as the linear functions for which the following objective function $J(\mathbf{T})$is maximized:

$$J(\mathbf{T}) = \frac{\left|\widetilde{\mathbf{S}}_B\right|}{\left|\widetilde{\mathbf{S}}_W\right|} = \frac{\left|\mathbf{T}^T\mathbf{S}_B\mathbf{T}\right|}{\left|\mathbf{T}^T\mathbf{S}_w\mathbf{T}\right|}. \quad \text{(V.9)}$$

Thus, a generalized solution for the *i*th column of T is the eigenvector corresponding to the *i*th largest eigenvalue of the matrix $\mathbf{S}_W^{-1}\mathbf{S}_B$ [98].

The use of LDA in speech recognition started in the late 1970s in the interest of acquiring features suitable for classification in a low dimensionality space. For example, Bocchieri and Wilpon [5] applied LDA to feature selection on a frame feature space and achieved encouraging results for continuous speech recognition based on HMMs. In this work, the feature components were ordered according to the ratio of a between-class measure and a within-class scatter measure. Then the highest ranked components were chosen to form a compact set of highly discriminative features for recognition. In the work of Zahorian et al. [114], two variants of LDA were investigated for an isolated-word recognition task. Experiments demonstrated that LDA was able to enhance phonetic distinctions in speech features and thus improve recognition performance. Kumar and Andreou [55] presented heteroscedastic discriminant analysis (HDA), a model-based generalization of LDA derived in the maximum-likelihood framework, to handle heteroscedastic-unequal variance-classifier models.

In addition to PCA and LDA, other linear dimensionality reduction methods such as Joint Linear Discriminants (JLDs) [49], Minimum Classification Error (MCE) [98], maximum likelihood Discriminant [84] have been broadly explored for various recognition tasks with mixed results reported.

However, the common drawback of these linear methods is that only linear structures can be easily extracted from the data. For example, if the data represent the complicated

interaction of features, then the linear subspace may lead to a poor representation and a nonlinear subspace may be needed.

Figure 25 illustrates the limitation of linear PCA. The straight line fit to the data obtained by linear PCA does not provide much information about original curve shaped data and results in a poor representation of the original data [14]. As an alternative, a nonlinear method might discover the curve that the data lies on.

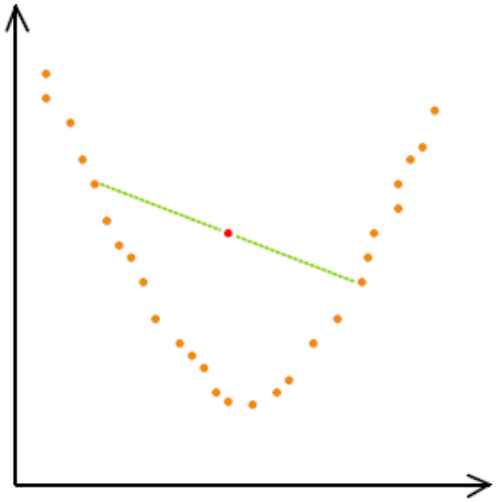

Figure 25: Straight line obtained using linear PCA applied to curve shaped data [14]

5.3 Nonlinear Principal Component Analysis (NLPCA)

As described in the previous section, when data are nonlinearly distributed such as on curved subspaces, linear PCA is suboptimal in capturing principal components. A natural extension of linear PCA is to introduce nonlinear functions so that the data with curved lines can be represented by its principal components through nonlinear mapping.

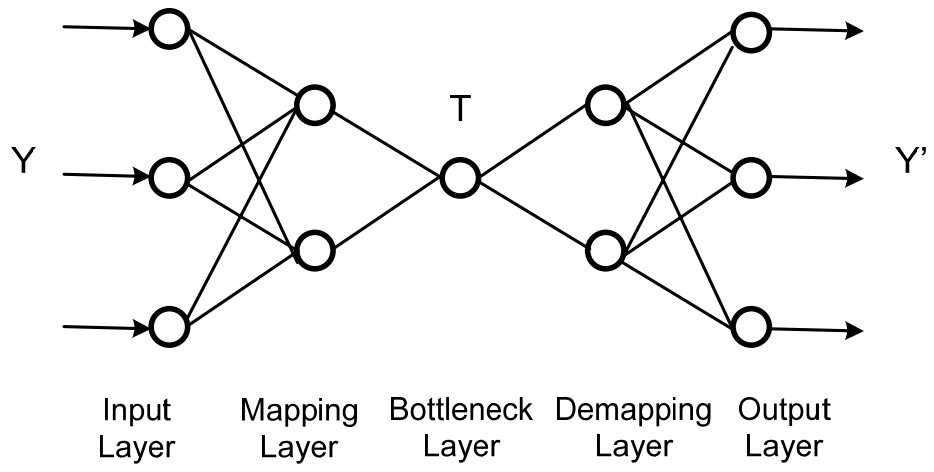

Figure 26: A general architecture of NLPCA [54]

Kramer presented a nonlinear principal component analysis (NLPCA) method based on a feed-forward neural network. As illustrated in Figure 26, the architecture of the proposed network consists of five layers including three hidden layers. The middle hidden layer, which contains fewer nodes than input and output layers, is used as a bottleneck layer. The network is trained to perform an identity mapping, where the input is approximated at the output layer. Thus, the output of the bottleneck layer results in an internal representation of the input data in a dimensionality reduced space [54].

In NLPCA, the mapping from an input **Y** into feature space T can be represented as:

$$\mathbf{T} = G(\mathbf{Y}) \tag{V.10}$$

where G is a nonlinear vector function composed of *f* individual nonlinear functions: $G = (G_1, G_2, \ldots, G_f)$. G_i is the *i*th nonlinear factor of *G* [54] .

The inverse transformation, restoring the original dimensionality of the data, is implemented by a second nonlinear function, $H = (H_1, H_2, \ldots, H_m)$;

$$\mathbf{Y}' = H(\mathbf{T}). \tag{V.11}$$

The objective function, or the loss function, used to train the neural network becomes

$$E = \sum_{j=1}^{n} \sum_{i=1}^{m} (\mathbf{X}_i - \mathbf{X'}_i)_j^2 \ . \tag{V.12}$$

The functions *G* and *H* are selected to minimize ||E|| using supervised training algorithms such as back-propagation as described in Section 4.2.2.

NLPCA and related nonlinear methods have been applied to a variety of nonlinear regression and classification tasks. For example, Hsieh presented a unified view of the NLPCA techniques and their applications to various data sets of the atmosphere and the ocean [38]. In the work of Malthouse [63], the theoretical properties, geometric interpretation, and parameter estimation of NLPCA were investigated for the tasks of feature extraction and dimensionality reduction.

In an earlier work [116], NLPCA was applied to an isolated vowel classification task, and the nonlinear method based on neural networks was experimentally compared with linear methods for reducing the dimensionality of speech features. It was demonstrated that NLPCA which minimizes mean square reconstruction error from a reduced dimensionality space can be very effective for representing data which lies in curved subspaces, but did not appear to offer any advantages over linear dimensionality reduction methods such as PCA and LDA, for a speech classification task. In contrast, as will be introduced subsequently, a nonlinear technique based on minimizing classification error, was quite effective for improving accuracy.

5.4 Nonlinear Discriminant Analysis for Dimensionality Reduction

5.4.1 Neural Network Based Nonlinear Discriminant Analysis (NLDA)

In this chapter, Neural Network Based NonLinear Discriminant Analysis (NLDA) is presented for the purpose of obtaining a set of compact, but highly discriminative speech features for an HMM-based speech recognition system. As described in Section 4.4, neural networks, which are trained by applying discrimination criteria subject to orthogonal targets, assist in maximizing the discrimination and lessening the correlation of speech features. The features transformed by neural networks thus result in a more effective speech representation for the acoustical modeling using GMM-based HMMs.

Another motivation of NLDA originates from NLPCA in which an internal representation of the input data to a neural network is acquired at the middle layer in a reduced dimensional space. However, NLPCA, trained to approximate the input data at the output layer for an identity mapping, is not intended to solve classification problems. In contrast, the neural network based approach which follows is trained as a feature classifier to reduce dimensionality as well as maximize discrimination among speech features.

This chapter presents two neural network based NonLinear Discriminative Analysis (NLDA) transformations, which obtain dimensionality reduced features from different layers of neural networks for speech recognition based on HMMs.

In the first approach, which is referred to as NLDA1, the transformed features are produced from the final output layer of the network. This approach is similar to the use of neural networks as described in Section 4.4, but the advantage of neural networks in dimensionality reduction is the focus of this approach.

In the second approach named NLDA2, the outputs of the middle hidden layer, with fewer nodes than the input layer, are used as transformed features to form a reduced but more discriminative dimension.

As illustrated in Figure 27, NLDA is based on a multilayer bottleneck neural network and performs a nonlinear feature transformation of the input speech features. The outputs of the network are further processed by PCA as described in Section 5.2.1 to create transformed features to be the inputs of an HMM recognizer [40,41].

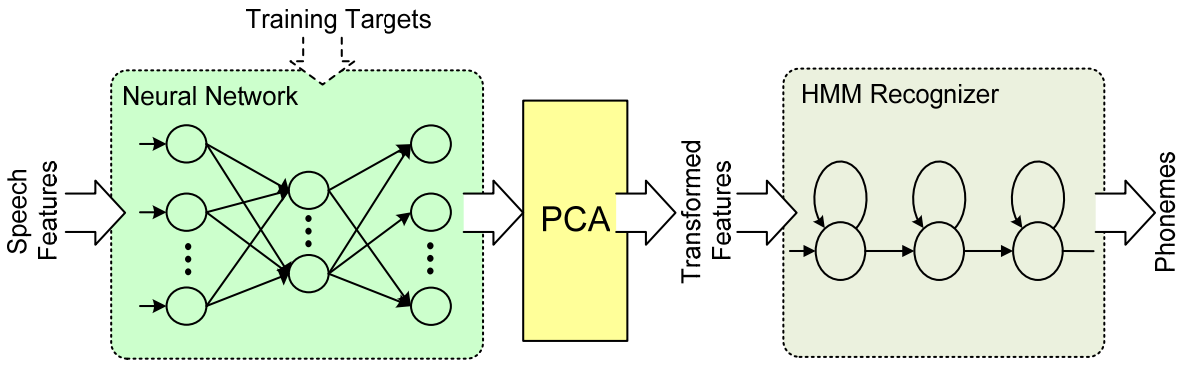

Figure 27: Overview of the NLDA transformation for speech recognition

The neural network used in NLDA includes an input layer, hidden layers and an output layer. The numbers of nodes contained in the input and output layers respectively correspond to the dimensions of the input features and the number of categories in the training target data. The number of hidden layers is experimentally determined as well as the number of nodes included in those layers.

Similar to the work described in Section 4.4.4 and the works of Hermansky et al. [36,20], the PCA processing is further applied to the output of the neural network, in order to reduce the correlation and improve the match to a GMM. In the NLDA1 approach, the dimension of the network outputs is also reduced using PCA to obtain

reduced features as the output of the network are subject to the number of output nodes, which are determined by the dimensionality of training targets.

The transformed features are finally recognized using HMMs with each state modeled as a GMM. Phoneme HMMs are used in this work as the recognizer for phonetic experiments, although other recognition units could be used. After emission and transition probabilities are estimated, the Viterbi algorithm is used to calculate the overall probabilities of a feature vector over HMMs and determine the model with the highest probability as the recognition result.

5.4.2 NLDA1

A straightforward use of neural networks for feature transformations is to produce features at the output layer, where the all trained layers are activated in the transformation and thus the full nonlinear ability of the network is engaged.

Following this idea, NLDA1 obtains transformed features at the output layer of a neural network which is trained to minimize the distance between its outputs and training targets. However, the transformed features, which are the approximation of training targets, are generally not suitable as the direct inputs for the GMMs used in HMMs. Another drawback is that the dimensionality of transformed features is equal to the number of training targets, that is, the number of phonemes for the work described in this dissertation. Therefore, PCA is used in NLDA1 to de-correlate the network outputs so that transformed features can be well-modeled with a GMM. Additionally, the transformed features can be reduced by PCA into a lower dimensionality space. Figure 28 illustrates the use of network outputs in NLDA1.

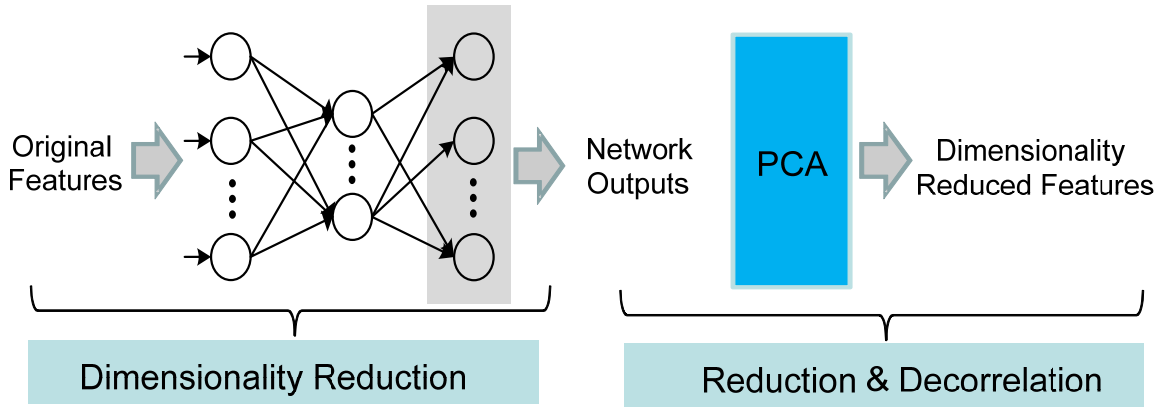

Figure 28: Use of network outputs in NLDA1

As described in Section 4.4.3, selecting a nonlinear function which is appropriate for the characteristics of the output data is of great importance. The experiments of Section 4.5 showed that a linear output layer in feature transformation was able to achieve data with a more diverse distribution, such as one that can be well-modeled with a GMM. Therefore, a linear output layer is used for the feature transformation, although all unipolar sigmoid nonlinear layers were used for the NLDA1 training.

5.4.3 NLDA2

Since the activations of the middle layer represent the internal structure of the input features, NLDA2 uses the outputs of the middle hidden layer in order to obtain low dimensional and highly discriminative features. The dimensionality of the reduced feature space is determined only by the number of nodes in the middle layer. Therefore, an arbitrary number of reduced dimensions can be obtained, independent of the input feature dimensions and the nature of the training targets. A lower dimensional representation of the input features is easily obtained by simply deploying fewer nodes in the middle layer than the input layer. This flexibility allows dimensionality to be adjusted so as to optimize overall system performance [39].

In contrast with NLDA1 where dimensionality reduction is assigned to PCA, for NLDA2, since the dimensionality reduction can be accomplished with the neural network only, the linear PCA is used specifically for reducing the feature correlation.

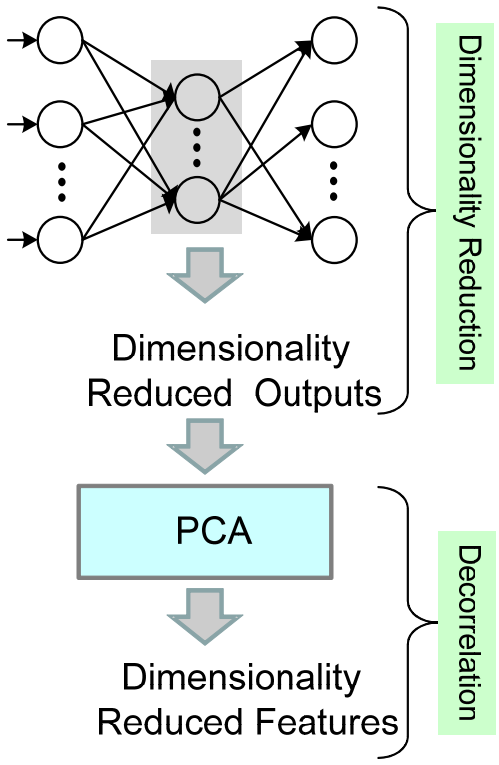

Figure 29: Middle layer outputs used as dimensionality reduced features in NLDA2

5.4.4 State Level Training Targets

The network training of NLDA follows the same procedure as described in Section 4.4.2. First, the original features are scaled using the mean and standard deviation vectors of the training data so that all components have the same mean and variance. Then, the weights of the network are estimated using the Back-Propagation algorithm to minimize the distance between the scaled input features and target data. The weight learning terminates when the weight updating drops below a threshold (or after a predefined number of weight updates).

The training of the neural network relies on the category information for creating training targets. In this chapter, a number of output nodes equal to the number of phone categories, with a value of 1 for the target category and 0 for the non-target categories, as presented in Section 4.4.2, were used as the baseline. In later experiments, the phone labeling information, which consists of the same 48 phone categories as that used for the HMM training, were used for the network training.

Due to the nonstationarity of speech signals, a speech signal varies even in a very short time interval (e.g. a phoneme). In the work of Chung and Un [16], in order to accommodate this variability, multiple neural networks are employed with each network corresponding to a state in an HMM. However, multiple networks increase the complexity of network training and ignore the relationship among states. Therefore, in this chapter, a different strategy is explored where training targets are designed to embed state information. Thus, a single neural network can be trained to produce state dependent outputs.

The following three types of training targets are discussed in this chapter:

1. Two-apex state training targets;
2. One-apex state training targets, and;
3. One-apex state training targets with "don't cares."

The two-apex state training target uses three extra components to indicate the target state in addition to the 48 components which represent phone categories. Therefore, there are two peaks with the values of "1" which respectively represent the phoneme category and the state target, resulting in a compact combination of both the state and phoneme information. Figure 31 shows an example of the two-apex state level target, which is an expansion of the phone level target illustrated in Figure 30. The peak at the fifth dimen-

sion represents phoneme 5, and the second peak in the last three components (e.g. 49 to 51) illustrates the state to which the target belongs.

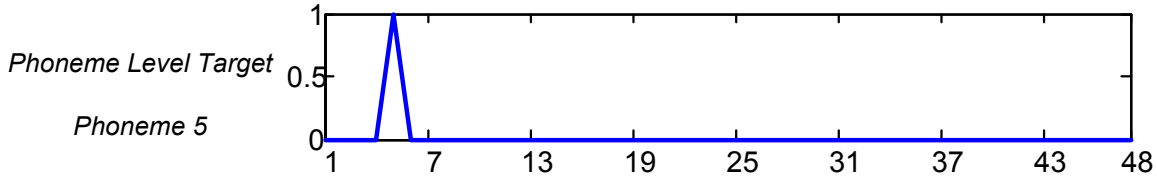

Figure 30: Illustration of a phoneme level target

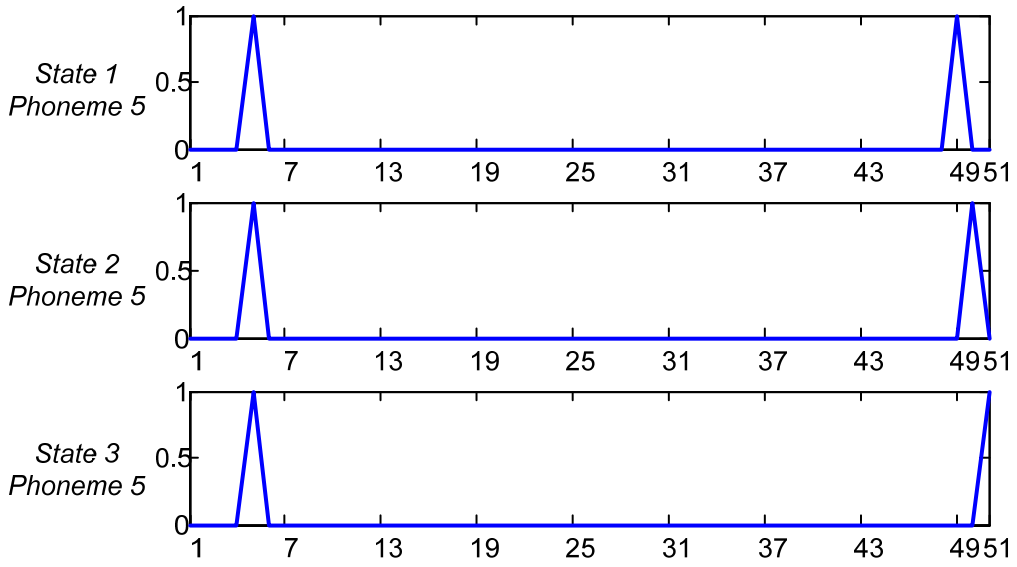

Figure 31: Illustration of the two-apex state level training targets

The one-apex state training target duplicates the phoneme specific target by the required number of the states in HMMs. For 3-state HMMs, there will be three times as many targets as there are distinct phones. There is a unique target for each state of each phone to represent the target phoneme. The duplicate, where the peak lies, indicates the target state. As illustrated in Figure 32, the peak "1" at the fifth component of the first duplicate shows that this target represents the first state of phoneme 5. There are no peaks in the other duplicates and the other components are all "0." Similarly, the target for the second state of the same phoneme is indicated by a peak at the fifth component of the

second duplicate. For the later experiments reported in this chapter, 48 phonemes were modeled with 3 state models, thus resulting in a neural network trained with 144 outputs rather than 48.

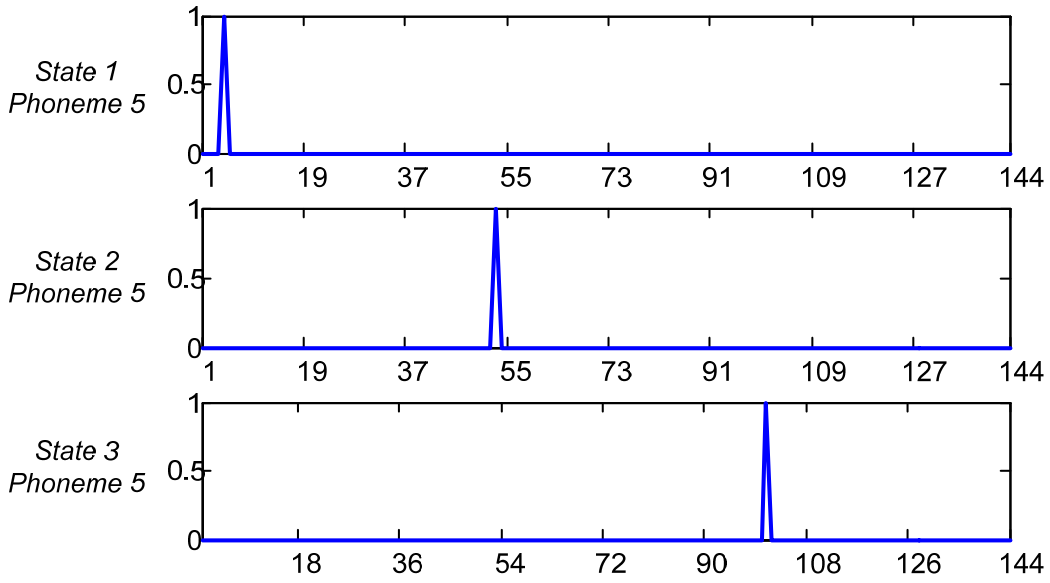

Figure 32: Illustration of the one-apex state level training targets

The One-apex state training target with "don't cares" uses "don't care" states for each phoneme model, so that one neural network trained with the targets can generate state dependent outputs. As illustrated in Figure 33, the phone-specific training targets are expanded to 144 dimensions in the same manner as those used for the one-apex state version. However, in the training process, for each point in time, one state target will be considered as a "1," and the other two state targets for that point in time will be considered as a "don't care" with the value of "-1", and the state targets for all other categories will be considered as "0" value targets. As time progresses during a phone, the "1" moves from state 1 to state 2, to state 3.

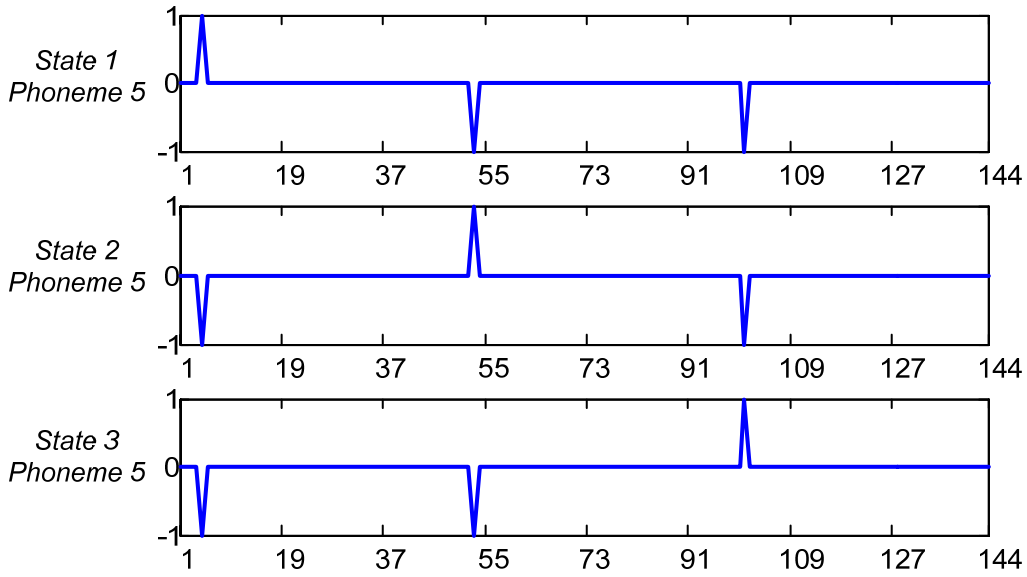

Figure 33: Illustration of the one-apex state level training targets with "don't cares"

The use of "don't cares" was based on the thinking that the features transformed by the neural network should be distinctive for each phoneme state, but the boundaries between states are likely to be indistinct. For example, the TIMIT training data used in the experiments of this dissertation provides only phoneme level transcriptions, thus the estimation of state boundary information is required, but often introduces errors due to the nature of unclear state boundaries and the lack of a reliable estimation approach. Therefore, the weights of the nodes corresponding to "Don't cares" will not be updated in the training so that there are no errors computed for the "don't care" output nodes.

Two approaches are used to expand a phoneme level label to a state level label. The first approach uses a fixed state length ratio for all phonemes. It assumes the first part of each phone is state 1, the central section state 2, and last part state 3. The second one determines state boundaries by using the HMM-based Viterbi alignment based on already trained HMMs.

The latter approach also provides global training between HMMs and neural networks by iteratively training the two components. As shown in Figure 34, a neural network is first trained with a fixed state length targets, then HMMs are trained, and then alignment based targets are used to train the neural networks again. The HMM training and neural network training steps are iterated until some point of convergence is reached.

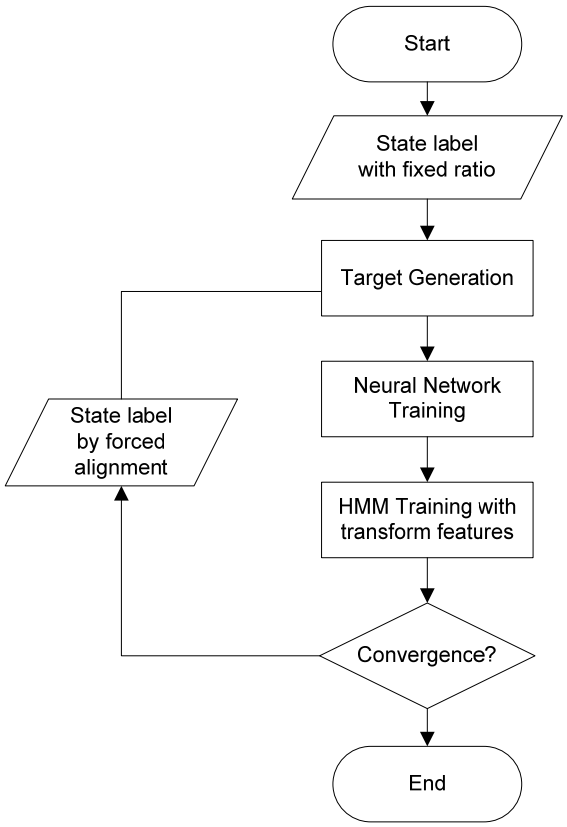

Figure 34: Flowchart of target generation based on the forced aligned state labels.

5.5 Experimental Evaluation

5.5.1 Experimental Setup

Several experiments based on the TIMIT database were conducted to investigate the two NLDA methods. Similar to the experiments in Section 4.5, the SA sentences were removed from the database, resulting in 3696 sentences for training and 1344 sentences

for testing. For all experiments, 13 DCTCs and 6 DCSCs were computed using 8 ms frames with a 2 ms frame spacing and a 1s block length, for a total of 78 DCTC-DCSC features. Left-to-right Markov models with no skip were used and a total of 48 mono-phone HMMs were created using the HTK toolbox (Ver3.4) [106]. The bigram phone information extracted from the training data was used to form the language model.

A neural network with 3 hidden-layers (500-36-500 nodes), experimentally determined, was used in NLDA1, while a 3 hidden-layer network with 500 nodes in the first and third hidden layers was used for NLDA2. The number of hidden nodes in the middle hidden layer in NLDA2 was varied from 6 to 48, according to the reduced dimensionality being evaluated. The number of nodes in the input layer was 78 corresponding to the dimensionality of the original features. The output layer used 48 nodes for the phoneme level targets, and 51 or 144 nodes for the state level targets.

5.5.2 Experiment with Various Reduced Dimensions

The first experiment was conducted to evaluate the two NLDAs with various dimensions in the reduced feature space with and without the use of PCA. As described in Section 5.4, the 48-dimensional outputs of the neural network were further reduced by PCA in NLDA1, while the dimensionality reduction was controlled only by the number of nodes in the middle layer in NLDA2. The features which are dimensionality reduced by PCA and LDA were also evaluated for the purpose of comparison.

Figure 35 shows recognition accuracies of dimensionality reduced features using 1-state and 3-state HMMs with 3 mixtures per state. Note that the NLDA1 features without the PCA process are always 48 dimensions. Compared to the PCA and LDA reduced

features, the NLDA1 and NLDA2 features performed considerably better for both the 1-state and 3-state HMMs.

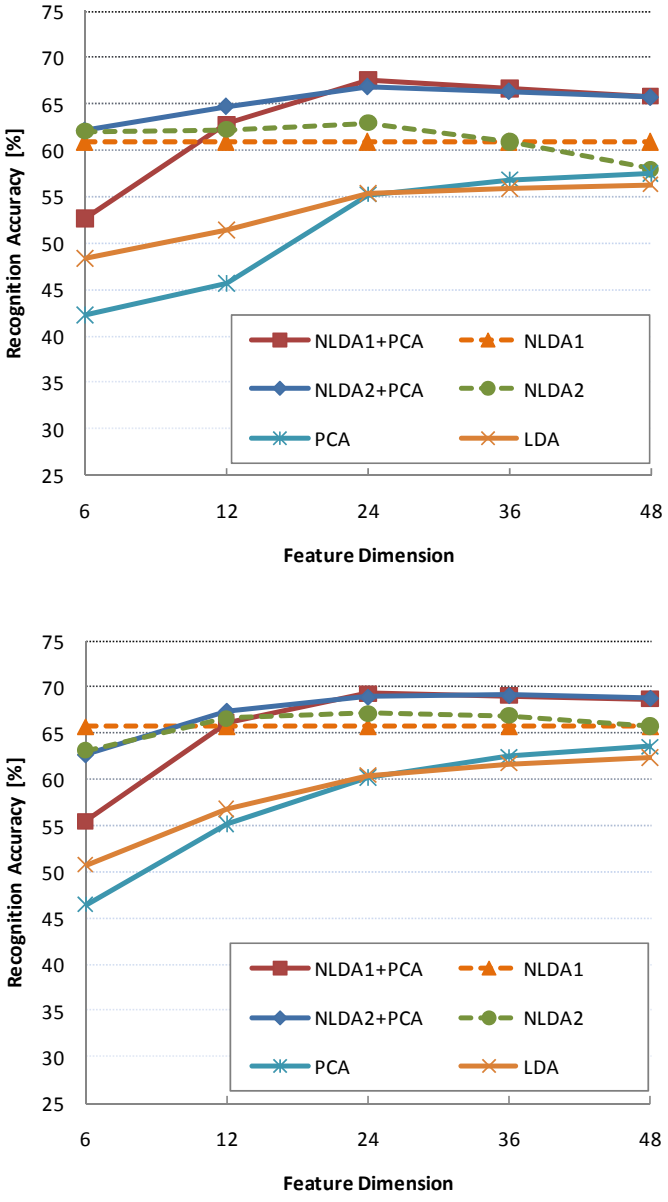

Figure 35: Accuracies of NLDA1 and NLDA2 with various dimensionality reduced features based on 1-state (top panel) and 3-state HMMs (bottom panel). The NLDA1 features without PCA are always 48 dimensions

In the case of 3-state HMMs, the transformed features reduced to 24 dimensions resulted in the highest accuracy of 69.33% for NLDA1. A very similar accuracy of 69.16% was obtained with NLDA2 by using 36 dimensional features.

The recognition accuracies were further improved by about 3% with PCA reduced dimensionality features versus the NLDA features for most cases, showing the effectiveness of PCA in de-correlating the network outputs.

5.5.3 NLDA1 and NLDA2 Experiment with various HMM configurations

The aim of the second experiment is the evaluation of NLDA1 and NLDA2 using a varying number of states and mixtures in HMMs. The 78 DCTC/DCSCs were reduced to 36 dimensions based on the findings in the previous experiment. The 48 phoneme level targets were used in the training of the network. The features which are the direct outputs of the network without PCA processing were also evaluated.

Figure 36 shows accuracies using 1-state and 3-state HMMs with a varying number of mixtures per state. NLDA2 performed better than NLDA1 for all conditions with approximately 2% higher accuracy. The NLDA2 transformed features resulted in the highest accuracy of 73.41% with 64 mixtures, which is about 1.5% higher than the original features for the same condition. The superiority of the NLDA transformed features is more significant when a small number of mixtures are used. For example, the NLDA2 features modeled by 3-state HMMs with 3 mixtures resulted in an accuracy of 69.37% versus 63.16% for the original features.

These results imply that the middle layer outputs of a neural network are able to better represent original features in a dimensionality-reduced space than are the outputs of

the final output layer. The configuration of HMMs can be largely simplified by incorporating NLDA.

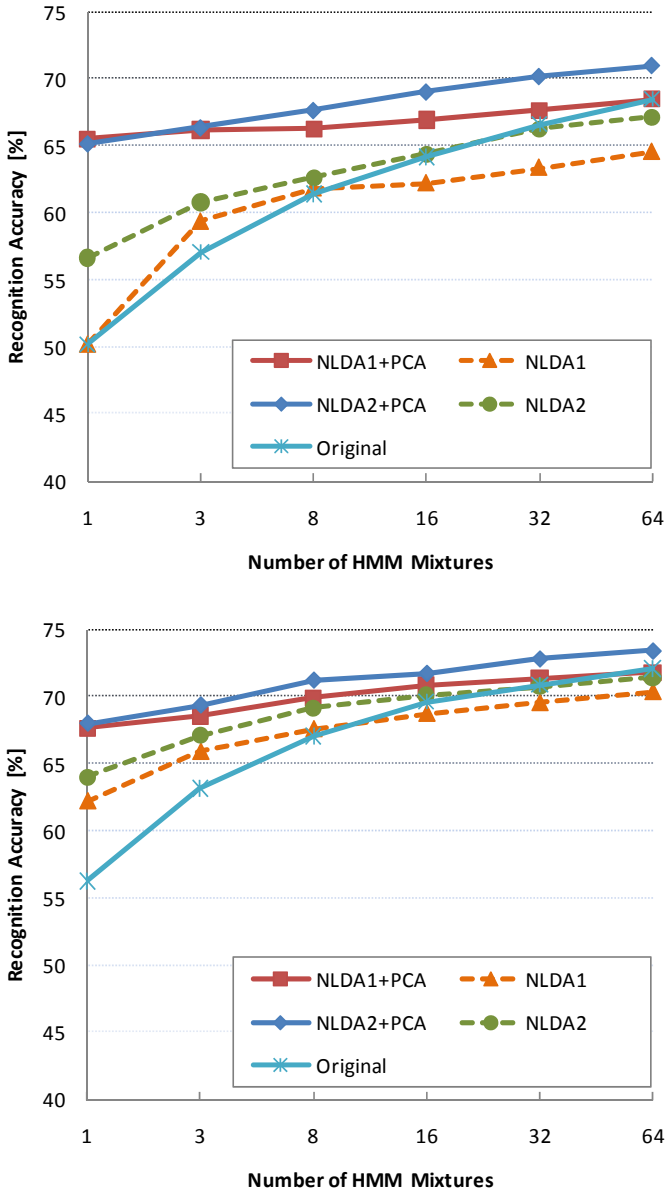

Figure 36: Accuracies of the NLDA1 and NLDA2 features using 1-state (top panel) and 3-state HMMs (bottom panel) with various numbers of mixtures

5.5.4 Experiment using State Specific Training Targets

As described in Section 5.4.4, three kinds of training targets were proposed so that a signal neural network can produce state dependent outputs. This experiment evaluated these training targets for NLDA1 and NLDA2 using 3-state HMMs with various numbers of mixtures. The neural network used 144 nodes in the output layers for the two one-apex targets, or 51 nodes for the two-apex state target. The network outputs were further processed with PCA for both NLDA1 and NLDA2. The state boundary for the expansion of these state targets was obtained using the approach of the fixed length ratio and experimentally determined to a length ratio of 1:4:1 for 3 states.

Recognition Accuracies of the NLDA1 and NLDA2 features using state specific targets are shown in Figure 38. Among three kinds of state-level targets, the one-apex target with "don't' cares" performed the best, especially for the NLDA1 case. Compared to with the phoneme level targets in Section 5.5.3, the features obtained with the state targets improved the recognition performance. For example, for the case of 64-mixture HMMs, the NLDA1 features using the one-apex targets with "don't' cares" is 72.54%, approximate 1.5% higher than the one with the phone level target.

However, the state level targets didn't appear to offer much advantage, such as for NLDA2 when more than 32 mixtures were used. It is speculated that the expanded networks, where the number of hidden-to-output weights were largely increased, were not sufficiently trained. Therefore, the following experiment with extensive network training and a massive number of HMM mixtures was designed and performed.

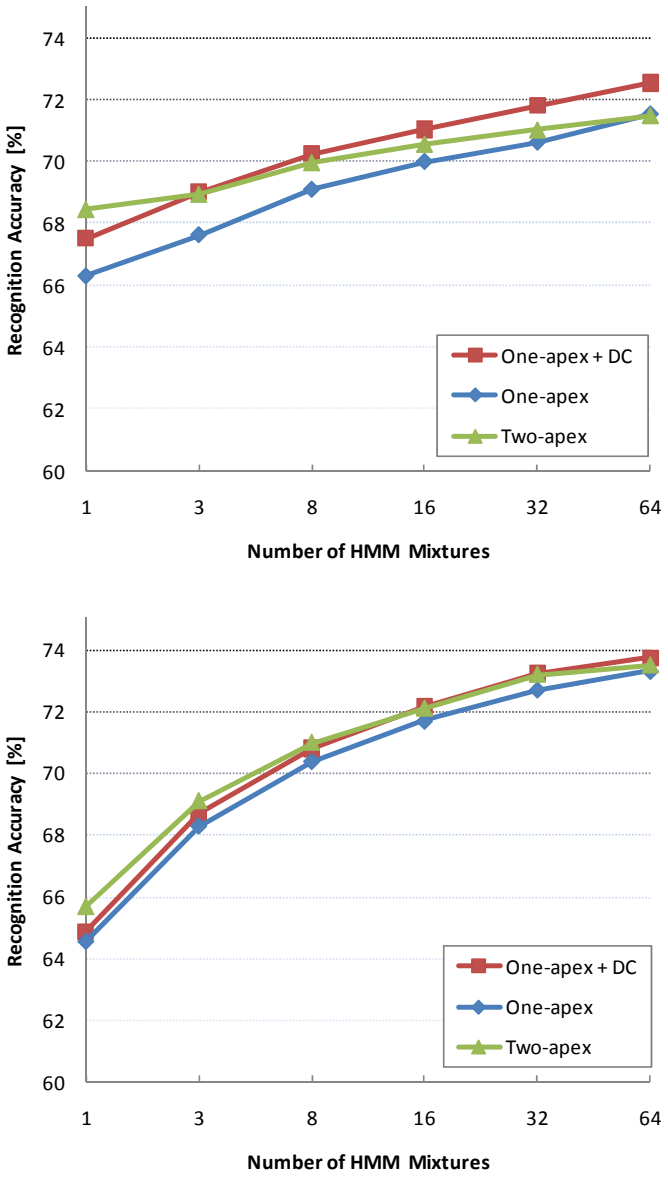

Figure 37: Accuracies of the NLDA1 (top panel) and NLDA2 (bottom panel) features using different state level targets for the network training. Note that "DC" is "don't care."

5.5.5 Experiments with Large Network Training

The NLDA methods trained using the one-apex state level targets with "don't cares" were investigated in this experiment. The state targets were expanded using either a constant length ratio (ratio for 3 states: 1:4:1) or a Viterbi forced alignment approach, as

described in Section 5.4.4. The expanded networks had 144 output nodes and were iteratively trained. The first iteration of the training started with randomly generated weights as used in Sections 5.5.3 and 5.5.4, and the next iteration updated the weights on the trained network with a decreased train update rate (i.e., smaller steps in gradient search of weight space). A total of five iterations ($4x10^7$ weight updates) were conduced.

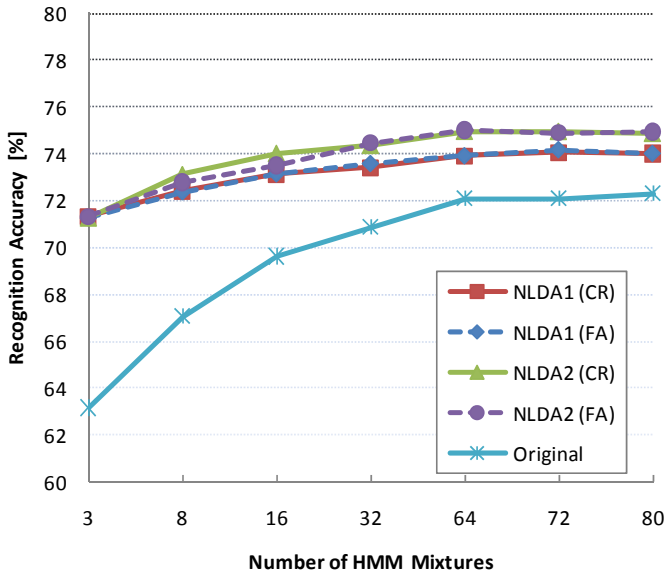

Figure 38: Recognition accuracies of the NLDA dimensionality reduced features using the state level targets. "(CR)" and "(FA)" indicate the training targets obtained with the constant length ratio and forced alignment respectively

As shown in Figure 38, both NLDA1 and NLDA2 using the expanded targets lead to an increase in accuracy approximately 2% higher than using the phoneme level targets reported in Figure 36. The use of forced alignment for state boundaries resulted in the highest accuracy of 75.0% with 64 mixtures. However, accuracies were only slightly lower using a fixed ratio of state lengths. These results imply that the use of "don't cares" is able to reduce errors introduced by inaccurate determination of state boundaries.

Comparing these results with those from Figure 36, the NLDA2 features in a reduced 36-dimensional space achieved a substantial improvement versus the original features, especially when a small number of mixtures were used. These results show the NLDA methods based on the state level training targets are able to obtain highly discriminative features in a dimensionality reduced space.

For comparison, 39-dimensional MFCC features (12 coefficients plus energy with the delta and acceleration terms) were reduced to 36-dimensions with the same configurations and evaluated. The results followed the same trend, but the accuracies were about 4% lower than those of the DCTC-DCSC features for all cases, for example, 70.7% with NLDA2 using forced alignment and 32 mixtures.

In order to provide the inside of the recognition results, Table 6 shows the confusion matrix of recognized phonemes based on NLDA1 using 64 mixtures in terms of seven phoneme groups as listed in Table 7. The correctness for each row was calculated as the number of correctly labeled phoneme instances divided by the total number of instances in the row. These values show that glides and affricates result in more difficult objects.

Table 6: Confusion matrix of phoneme groups

	Vowel	Stop	Glide	Nasal	Fricative	Affricate	Silence	Corr [%]
Vowel	14894	73	333	118	56	4	74	0.96
Stop	50	5519	30	34	155	26	69	0.94
Glide	411	116	4365	48	88	16	14	0.86
Nasal	177	82	40	3505	27	1	29	0.91
Fricative	69	223	27	39	5910	67	63	0.92
Affricate	1	44	2	2	24	453	12	0.84
Silence	136	66	24	69	48	6	9080	0.96

Table 7: Seven phoneme groups

Group	Phoneme
Vowel	iy eh ae ix ax ah uw uh ao aa ey ay oy aw ow er
Stop	b d dx g p t k
Glide	l el r y w hh
Nasal	m n en ng
Fricative	dh z zh v f th s sh
Affricate	ch jh
Silence	cl vcl epi sil

One more experiment was conducted to examine possible accuracy deviation due to different data. The TIMIT test data was equally divided in half so that the two groups include the same number of utterances from each speaker. Figure 39 shows the recognition accuracies of the NLDA dimensionality reduced features based on the two groups of test data. The results of this experiment illustrate that a stable performance of NLDAs can be expected independent of the nature of test data.

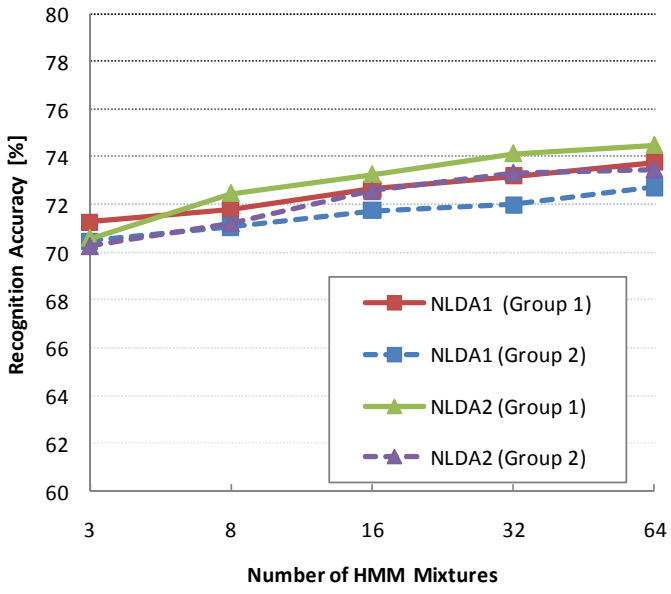

Figure 39: Recognition accuracies of the NLDA features based on different test data

5.6 Literature Comparison

The proposed approaches of this study are compared to those reported in literature using the TIMIT database as presented in Section 1.3. As listed in Table 8, the best result obtained in this study is higher than all others, except for that of the Tandem NN in which multiple neural networks and higher dimensional features were used.

Table 8: Accuracy comparison using the TIMIT database

Study	Feature	Recognizer	Accuracy (%)
Somervuo (2003) [89]	MFCC	NN/HMM	68.5
Ketabdar and Bourlard (2008) [53]	PLP	NN/HMM	71.5
Pinto and Hermansky (2008) [73]	LPC	HMM/MLP	74.6
Sha and Saul (2007) [87]	MFCC	HMM	70.0
Schwarz et al. (2006) [86]	MFCC	Tandem NN	78.5
Zahorian et al. (2009) [112]	DCTC/DCSC	HMM	73.9
This study	DCTC/DCSC	NN/HMM	75.0

5.7 Conclusions

Two feature dimensionality reduction methods based on neural networks were proposed in this chapter. In order to train neural networks with state dependent targets, three kinds of state level training target were introduced. In addition, a “don’t care” concept is used to represent states where boundaries are not likely to be distinct.

The NLDA methods were evaluated based on the TIMIT database in terms of phoneme recognition accuracy. The features reduced by NLDA1 and NLDA2 were compared with the PCA and LDA reduced features as well as the original 78 DCTC-

DCSC features. The neural networks were also trained with the state-level targets and a large number of weight updates.

The findings from the experimental evaluation include:

1. The NLDA1 and NLDA2 methods performed considerably better than PCA and LDA for the feature dimensionality reduction in speech recognition, especially with the incorporation of PCA in de-correlating the network outputs;
2. The middle layer outputs of a neural network (NLDA1) are able to better represent original features in a dimensionality-reduced space than are the outputs of the final output layer;
3. Recognition accuracies were improved using the state specific targets and a large number of iterations in the network training. The highest accuracy of 75.0% was obtained with the NLDA2 features using 3-state HMMs with 64 mixtures per state, and;
4. The NLDA features based on the state level training targets with "don't cares" are able to obtain highly discriminative features in a reduced space, which thus largely simplify HMM configurations and improve recognition performance.

Chapter VI Conclusions and Future Work

Automatic speech recognition has advanced to the point where state of the art speech recognition algorithms perform reasonably well even for large vocabulary continuous speech recognition in practical environments. Among speech recognition problems, feature extraction, which compresses a speech signal into streams of acoustical feature vectors, has become even more important for ASR since acoustical modeling methods have been well established and language modeling largely depends on the nature of the targeted language. The focus of this dissertation was the determination of effective speech features, where both spectral and temporal variations in speech are captured in a low dimensional representation, for speech recognition tasks.

In this dissertation, the DCTC/DCSC features were investigated as a spectral-temporal speech representation in Chapter III. Experimental evaluations showed that temporal variations are also of great importance for speech recognition, especially using long time context. In Chapter IV, a neural network is utilized to nonlinearly transform the DCTC/DCSC features to obtain uncorrelated and highly discriminative features for GMM based HMMs. Chapter V targeted feature dimensionality issues since high dimensional features, such as DCTC/DCSCs, increase computation cost considerably and greatly restrict performance improvement. Two Nonlinear Discriminant Analysis (NLDA) methods based on neural networks were presented for nonlinear dimensionality reduction of speech features. The first method (NLDA1) used the final outputs of the network to obtain dimensionality reduced features with the incorporation of PCA processing, while the second one (NLDA2) focused on the middle layer outputs. The

very high phone accuracy obtained with the TIMIT database was 75.0% with NLDA2 using a large number of training iterations based on the state-specific targets.

6.1 Contributions

The contributions of this work are summarized as follows:

1. The DCTC-DCSC features have been applied to a continuous speech recognition task to examine tradeoffs between spectral and temporal information of the features. In contrast to traditional features such as MFCCs where static information is essential and long frame length is desirable, the importance of temporal features was experimentally demonstrated for speech recognition as well as the significance of the short frame time (i.e., low spectral resolution) DCTC-DCSCs with long temporal context;
2. A tandem NN/HMM structure has been thoroughly discussed in this work for the purpose of seeking a hybrid recognition framework where a neural network based feature transformer was employed to reduce the limitations of HMMs. It was shown that the diverse nonlinear functions provided by multiple hidden layers considerably helped neural networks to create uncorrelated and highly discriminative features, thus improve the performance of speech recognition. The effectiveness of PCA in lessening the feature correlation was also verified through experimental evaluations;
3. In pursuit of a compact and highly discriminative speech representation, two Nonlinear Discriminant Analysis (NLDA) transformations based on neural networks have been proposed to reduce feature dimensionality while improving discriminations among phonemes. Experimental evaluation using the TIMIT da-

tabase showed that very high recognition accuracies with the NLDA dimensionality reduced features were obtained, especially when using the outputs of network middle layer, and;

4. The NLDA methods have been extended to incorporate state-dependent targets into the training of the networks so that short time variations can be considered. The extended networks with extensive training iterations have further improved the recognition performance and resulted in an accuracy of 75.0% in terms of phoneme recognition, which is better than most other reported state of the art approaches using the same database and configuration as used in this work.

6.2 Suggestions for Future Work

There are several suggestions for further work as follows:

1. The proposed approaches were evaluated using the HMM based recognition in terms of phoneme recognition. To further prove the effectiveness of the approaches, the evaluation based on a larger vocabulary database for a task of word level recognition is needed;
2. Neural networks are also popular techniques for noise compensation and speaker adaption in speech recognition, thus the proposed methods could provide benefits for noise additive speech and speaker-dependent recognizers. Future investigation and evaluation on these aspects may be desired;
3. The performance of the algorithms presented in this dissertation still could be improved. More studies, such as the use of multiple neural networks and the global training of neural networks and HMMs, are needed to develop more accurate and robust speech recognition algorithms;

4. The investigation of other neural networks such as a Generalized Regression Neural Network may allow the function of PCA to be merged into a neural network, and provide better accuracy overall, and;
5. Apart from experimental comparison, a more theoretical exploration on this neural network based nonlinear feature transformation is needed in the future. For example, a theoretical basis should be used to optimally determine the network configuration and choose layer nonlinearities for a specific recognition task.

Bibliography

[1] A. M. Ahmad, S. Ismail, and D. F. Samaon, "Recurrent Neural Network with Backpropagation," in *Intemational Symposium on Communications and Information Technologies*, 2004.

[2] J. K. Baker, "The dragon system - An overview," *IEEE Trans. Acoust. Speech Signal Process*, vol. 23, pp. 24-29, 1975.

[3] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global Optimization of a Neural Network - Hidden Markov Model Hybrid," in *International Joine Conference on Neural Networks*, 1991.

[4] M. Benzeghiba et al., "Automatic speech recognition and speech variability: A review.," *Speech Communication*, vol. 49, pp. 763-786, 2007.

[5] E.L. Bocchieri and J.G. Wilpon, "Discriminative analysis for feature reduction in automatic speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, 1992, pp. 501-504.

[6] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 27, pp. 113-120, 1979.

[7] H. Bourland and C. J. Wellekens, "Links Between Markov Models and Multilayer Perceptrons," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 1167-1178, 1990.

[8] H. Bourlard and N. Morgan, "Continuous speech recognition by connectionist statistical methods," *IEEE Transactions on Neural Networks*, vol. 4, pp. 893-909, 1993.

[9] J.S. Bridle, "Alphabets: a recurrent neural network architecture with a hidden Markov model interpretation," *Speech Communications*, vol. 9, pp. 83-92, 1990.

[10] M. Brookes, VOICEBOX: Speech Processing Toolbox for MATLAB, http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, 2010, last accessed: 03/01/2010.

[11] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

[12] R. Cardin, Y. Normandin, and R. De Mori, "High performance connected digit recognition using maximum mutual information estimation," in *Proc. ICASSP '91*, 1991, pp. 533-536.

[13] E. Chang, J. Zhou, S. Di, C. Huang, and K. F. Lee, "Large Vocabulary Mandarin Speech Recognition with Different Approaches in Modeling Tones," in *Proc. ICSLP '00*, 2000.

[14] L. Chapel, Dimension Reduction by Non Linear Methods, http://www.cs.unc.edu/Courses/comp290-90-f03/DimReduction2.pdf, 2003, last accessed in July 2007.

[15] B. Chen, Q. Zhu, and N. Morgan, "Learning long-term temporal features in LVCSR using neural networks," in *Proc. ICSLP*, 2004, pp. 612-615.

[16] Y. J. Chung and C. K. Un, "An MLP/HMM hybrid model using nonlinear predictors," *Speech Commun.*, vol. 19, pp. 307-316, 1996.

[17] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357-366, 1980.

[18] D. L. Donoho, "Aide-Memoire. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality," Department of Statistics, Stanford University, 2000.

[19] R. O. Duda, P. E. hart, and D. G. Stork, *Pattern Classification*, Richard O. Duda, Ed. New York: A Wiley-Interscience Publication, 2001.

[20] D. Ellis, R. Singh, and S. Sivadas, "Tandem Acoustic Modeling In Large-Vocabulary Recognition," in *Proc. ICASSP '01*, 2001, pp. 517-520.

[21] J. L. Flanagan, *Speech Analysis Synthesis and Perception*, James Loton, Ed. Berlin, New York: Springer Verlag, 1972.

[22] I. Fodor, "A Survey of Dimension Reduction Techniques," Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Technical Report 2002.

[23] V. Fontaine, C. Ris, J-M. Boite, and Multitel Site Initialis, "Nonlinear Discriminant Analysis for Improved Speech Recognition," in *Proc. Eurospeech-97, Rhodes*, 1997, pp. 4-2071.

[24] S. Furui, "50 Years of Progress in Speech and Speaker Recognition Research," *ECTI Transcations on Computer and Information Technology*, vol. 1, pp. 64-74, 2005.

[25] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition," *Computer Speech and Language*, vol. 12, pp. 75-98, 1997.

[26] M. J. F. Gales and S. J. Young, "Cepstral parameter compensation for HMM recognition in noise," *Speech Commun.*, vol. 12, pp. 231-239, 1993.

[27] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," in *Foundations and Trends in Signal Processing*, 2008, p. 2007.

[28] J. S. Garofolo et al., TIMIT Acoustic-Phonetic Continuous Speech Corpus, http://www.ldc.upenn.edu/Catalog/LDC93S1.html, 1993, last accessed 7/21/2009.

[29] J. l. Gauvain and C. h. Lee, "MAP Estimation of Continuous Density HMM: Theory and Applications," in *Proc. of DARPA Speech and Natural Language Workshop*, 1992, pp. 185-190.

[30] J. L. Gauvain and C. H. Lee, "Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291-298, 1994.

[31] R. A. Gopinath, "Maximum Likelihood Modeling With Gaussian Distributions For Classification," in *Proceedings of ICASSP*, 1998, pp. 661-664.

[32] E. Gouvea, CMU Sphinx - Speech Recognition Toolkit, http://cmusphinx.sourceforge.net/, 2010, last accessed 5/1/2010.

[33] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, pp. 1738-1752, 1990.

[34] H. Hermansky, "Should Recognizers Have Ears?," in *Proc. ESCA Tutorial and Research Workshop on Robust Speech*, 1997, pp. 1-10.

[35] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 578-589, 1984.

[36] H. Hermansky, D. P. W., and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP '00*, vol. 3, 2000, pp. 1635-1638.

[37] P. Hix, "Automatic Speech Recognition using LP-DCTC/DCS Analysis Followed by Morphological Filtering," Old Dominion University, Norfolk, VA, PhD Dissertation 2006.

[38] W. W. Hsieh, "Nonlinear multivariate and time series analysis by neural network methods," *Reviews of Geophysics*, vol. 42, pp. 10-1029, 2004.

[39] H. Hu and S. A. Zahorian, "A Neural Network Based Nonlinear Feature Transformation for Speech Recognition," in *Proc. INTERSPEECH '08*, 2008, pp. 533-1536.

[40] H. Hu and S. A. Zahorian, "Dimensionality Reduction Methods for HMM Phonetic Recognition," in *Proc. ICASSP 2010*, 2010, pp. 4854 - 4857.

[41] H. Hu and S. A. Zahorian, "Neural Network Based Nonlinear Discriminant Analysis for Speech Recognition," in *Proc. ANNIE 2009*, 2009.

[42] C. S. Jang and C. K. Un, "A new parameter smoothing method in the hybrid TDNN/HMM architecture for speech recognition," *Speech Communication*, vol. 19, pp. 317-324, 1996.

[43] C. R. Jankowski, H. D. H., and R. P. Lippmann, "A comparison of signal processing front ends for automatic word recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, pp. 286-293, 1995.

[44] F. T. Johansen and M. H. Johnsen, "Non-linear input transformations for discriminative HMMs," in *Proc. ICASSP '94*, 1994, pp. I/225--I/228.

[45] I.T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[46] E. R. Jones, "An Introduction to Neural Networks," Visual Numerics, Inc., San Ramon, CA, White Paper 2004.

[47] B.-H. Juang and S Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing,* vol. 40, pp. 3043-3054, 1992.

[48] B. H. Juang and Lawrence. R. Rabiner, "Automatic Speech Recognition - A Brief History of the Technology," in *Elsevier Encyclopedia of Language and Linguistics*.: Elsevier , 2005.

[49] S.S. Kajarekar, B. Yegnanarayana, and H. Hermansky, "A study of two dimensional linear discriminants for ASR," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, 2001, pp. 137-140.

[50] N. Kanedera, H. Hermansky, and T. Arai, "On Properties of Modulation Spectrum for Robust Automatic Speech Recognition," in *Proc ICASSP '98*, 1998.

[51] M. Karnjanadecha and S. A. Zahorian, "Signal Modeling for High-Performance Robust Isolated Word Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 647-654, 2001.

[52] M. Karnjanadecha and S. A. Zahorian, "Signal modeling for isolated word recognition," in *Proc. ICASSP '99*, 1999, pp. 293-296.

[53] H. Ketabdar and H. Bourlard, "Hierachical Integration of Phonetic and Lexical Knowledge in Phone Posterios Esitmation," in *Proc. ICASSP '08*, 2008.

[54] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, pp. 233-243, 1991.

[55] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, pp. 283-297, 1998.

[56] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1641-1648, 1989.

[57] R. P. Lippmann, "An introduction to computing with neural nets," *SIGARCH Comput. Archit. News*, vol. 16, pp. 7-25, 1988.

[58] R. P. Lippmann, "Review of neural networks for speech recognition," *Neural*

Comput., vol. 1, pp. 1-38, 1989.

[59] R. P. Lippmann and B. Gold, "Neural net classifiers useful for speech recognition," in *Proc. IEEE International. Conference of Neural Networks*, 1987, pp. 417-425.

[60] B. Lowerre, , A. Waibel and K. F. Lee, Eds.: Morgan Kaufmann Publishers, 1990, ch. Readings in Speech Recognition, pp. 576-586.

[61] N.U. Maheswari, A.P.Kabilan, and R.Venkatesh, "Speech Recognition System Based On Phonemes Using Neural Networks," *International Journal of Computer Science and Network Security*, vol. 9, pp. 148-153, 2009.

[62] J. Makhoul and R. Schwartz, "State of the art in continuous speech recognition," in *Proc. National Academy of Sciences*, 1994, pp. 165-198.

[63] E. C. Malthouse, "Some Theoretical Results on Nonlinear Principal Components Analysis," in *In Proceedings of the American Control Conference*, 1996.

[64] F. Meng, "Whole Word Phonetic Displays for Speech Articulation Training," Old Dominion University, Norfolk, VA, PhD Dissertation 2006.

[65] B. Milner, "Inclusion of temporal information into features for speech recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 1, 1996, pp. 256 -259 vol.1.

[66] N. Morgan and H. Bourlard, "Continuous Speech Recognition: An Introduction to the Hybrid HMM/Connectionist Approach," *Signal Processing Magazine*, pp. 25-42, 1995.

[67] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-vocabulary dictation using SRI's DECIPHER speech recognition system: progressive search techniques," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 2, pp. 319-322, 1993.

[68] K. Nagata, Y. Kato, and S. Chiba, "Spoken Digit Recognizer for Japanese Language," *NEC Research. Development*, 1963.

[69] V. D. Nhat and S. Lee, "PCA-Based Human Auditory Filter Bank for Speech Recognition," in *Proc. of International Conference on Signal Processing \& Communications (SPCOM)*, 2004.

[70] D. O'Shaughnessy, "Automatic speech recognition: History, methods and challenges," *Pattern Recognition*, vol. 41, pp. 2965-2979, 2008.

[71] P. Pedersen, "The Mel Scale," *Journal of Music Theory*, vol. 9, pp. 295-308, 1965.

[72] J. W. Picone, "Signal modeling techniques in speech recognition," in *Proceedings of the IEEE*, 1993, pp. 1215-1247.

[73] J. P. Pinto and H. Hermansky, "Combining Evidence from a Generative and a Discriminative Model in Phoneme Recognition," in *Proc. Interspeech '08*, 2008, Submitted for publication.

[74] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.

[75] Lawrence R. Rabiner and Biing Hwang Juang, "An introduction to hidden

[76] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*.: Prentice-Hall, Inc., 1993.

[77] G. Rigoll, C. Neukirchen, and J. Rottland, "A new hybrid system based on MMI-neural networks for the RM speech recognition task," in *Proc. ICASSP '96*, 1996, pp. 865-868.

[78] G. Rigoll, Ch. Neukirchen, and J. Rottland, "Large Vocabulary Speaker-Independent Continuous Speech Recognition With A New Hybrid System Based On MMI-Neural Networks," in *Proc. EUROSPEECH '95*, 1995.

[79] S. K. Riis, "Hidden Markov Models and Neural Networks for Speech Recognition," Technical University of Denmark, 1998.

[80] S. Riis and A. Krogh, "Hidden Neural Networks: A Framework for HMM/NN Hybrids," in *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 4*, vol. 4, 1997, p. 3233.

[81] T. Robinson and F. Fallside, "A recurrent error propagation network speech recognition system," *Computer Speech and Language*, vol. 5, pp. 259-274, 1991.

[82] J. Rottland, C. Neukirchen, and D. Willett, "Performance of Hybrid MMI-

Connectionist / HMM Systems on the WSJ Speech Database," in *Proc. ICASSP '97*, 1997, p. 1747.

[83] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, 1986.

[84] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *Proc. ICASSP '00*, 2000, pp. II1129--II1132.

[85] R. Schwartz et al., "The BBN BYBLOS Continuous Speech Recognition system," in *HLT '89: Proceedings of the workshop on Speech and Natural Language*, 1989, pp. 94-99.

[86] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical Structures of Neural Networks for Phoneme Recognition," in *Proc. ICASSP '06*, vol. 1, 2006, pp. I -I.

[87] F. Sha and L. K. Saul, "Large margin hidden Markov models for automatic speech recognition," in *in Advances in Neural Information Processing Systems 19*, 2007, pp. 1249-1256.

[88] J. O. Smith and J. S. Abel, "Bark and ERB bilinear transforms," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 697-708, 1999.

[89] P. Somervuo, "Experiments with linear and nonlinear feature transformations in HMM based phone recognition," in *Proc. ICASSP '03*, vol. 1, 2003, pp. I--52-5.

[90] P. Somervuo, B. Chen, and Q. Zhu, "Feature Transformations and Combinations for Improving ASR Performance," in *Proc. European Conference on Speech Communication and Technology*, 2003, pp. 477-480.

[91] J. Suzuki and K. Nakata, "Recognition of Japanese Vowelsâ€"Preliminary to the Recognition of Speech," *J. Radio Res. Lab*, vol. 37, pp. 193-212, 1961.

[92] T. Takiguchi and Y. Ariki, "PCA-Based Speech Enhancement for Distorted Speech Recognition," *Journal of Multimedia*, vol. 2, pp. 13-18, 2007.

[93] R. Togneri, A. M. Toh, and S. Nordholm, "Evaluation and Modification of Cepstral Moment Normalization for Speech Recognition in Additibe Babble Ensemble," in *Proceedings of the 11th Australian International Conference on Speech Science and Technology*, 2006.

[94] E. Trentin and M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomputing*, vol. 37, pp. 91-126, 2001.

[95] K. Vertanen, "Baseline WSJ Acoustic Models for HTK and Sphinx: Training Recipes and Recognition Experiments," Cavendish Laboratory, University of Cambridge, Technical Report 2006.

[96] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 328-339, 1989.

[97] A. Waibel and K. F. Lee, *Readings in speech recognition*, Alex Waibel and Kai-Fu Lee, Eds.: Morgan Kaufmann Publishers Inc., 1990.

[98] X. Wang and K. K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition," *Pattern Recognition*, vol. 36, pp. 2429-2439, 2003.

[99] Z. Wang, T. Schultz, and A. Waibel, "Comparison of Acoustic Model Adaptation Techniques on Non-Native Speech," in *Porc. ICASSP '03*, 2003, pp. 540-543.

[100] C. Wang and S. Seneff, "Improved Tone Recognition by Normalizing for Coarticulation and Intonation Effects," in *Proc. ICSLP '00*, 2000, pp. 83-86.

[101] B. Widrow, D. E. Rumelhart, and M. A. Lehr, "Neural networks: applications in industry, business and science," *Commun. ACM*, vol. 37, pp. 93-105, 1994.

[102] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, pp. 270-280, 1989.

[103] M. Wolfel and J. McDonough, *Distant Speech Recognition*, Matthias Wolfel, Ed.: Wiley, 2009.

[104] J. Wu and C. Chan, "Isolated Word Recognition by Neural Network Models with Cross-Correlation Coefficients for Speech Dynamics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 1174-1185, 1993.

[105] Y.F. Yen, "PCA - principal Component Analysis," Management Information System, Chuncheng University, Techincal Report 2010.

[106] S. Young et al., *The HTK Book (for HTK Version 3.4)*, Steve Young et al., Eds.: Cambridge University Engineering Department, 2006.

[107] K. Yu, "Adaptive Training for Large Vocabulary Continuous Speech Recognition," Cambridge University, Cambridge, PhD Dissertation 2006.

[108] D. Yuk, "Robust Speech Recognition Using Neural Networks and Hidden Markov Models," The State University of New Jersey at Rutgers, PhD Dissertation 1999.

[109] D. Yuk and J. Flanagan, "Telephone speech recognition using neural networks and hidden Markov models," in *Proc. ICASSP '99*, 1999, pp. 157-160.

[110] S. A. Zahorian, P. Dikshit, and H. Hu, "A Spectral-Temporal Method for Pitch Tracking," in *Proc. ICSLP '06*, 2006, pp. 910-913.

[111] S. A. Zahorian and H. Hu, "A spectral/temporal method for robust fundamental frequency tracking," *J. Acoust. Soc. Am.*, vol. 123, pp. 4559-4571, 2008.

[112] S. A. Zahorian, H. Hu, Z. Chen, and J. Wu, "Spectral and Temporal Modulation Features for Phonetic Recognition," in *Proc. INTERSPEECH '09*, 2009.

[113] S. A. Zahorian and Z. B. Nossair, "A Partitioned Neural Network Approach for Vowel Classification Using Smoothed Time/Frequency Features," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 414-425, 1999.

[114] S.A. Zahorian, D. Qian, and A.J. Jagharghi, "Acoustic-phonetic transformations for improved speaker-independent isolated word recognition," in *Porc. Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 0, 1991, pp. 561-564.

[115] S.A. Zahorian, P. Silsbee, and X. Wang, "Phone Classification with Segmental Features and a Binary-Pair Partitioned Neural Network Classifier," in *Proc. ICASSP '97*, 1997, p. 1011.

[116] S. A. Zahorian, T. Singh, and H. Hu, "Dimensionality Reduction of Speech Features using Nonlinear Principal Components Analysis," in *Proc. INTERSPEECH '07*, 2007, pp. 1134-1137.

[117] S. A. Zahorian, A. M. Zimmer, and F. Meng, "Vowel Classification for Computer-based Visual Feedback for Speech Training for the Hearing Impaired," in *ICSLP*,

2002.

[118] Y. Zheng et al., "Accent Detection and Speech Recognition for Shanghai-Accented Mandarin," in *Proc. EUROSPEECH '05*, 2005.

[119] Q. Zhu and A. Alwan, "Non-linear feature extraction for robust speech recognition in stationary and non-stationary noise," *Computer Speech \& Language*, vol. 17, pp. 381-402, 2003.

[120] A. Zolnay, D. Kocharov, R. Schluter, and H. Ney, "Using multiple acoustic feature sets for speech recognition," *Speech Communication*, vol. 49, pp. 514-525, 2007.

[121] V. Zue and R. Cole, *Survey of the State of the Art in Human Language Technology*, Ron Cole et al., Eds. New York: Cambridge University Press, 1997.

[122] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: Timit and beyond," *Speech Communication*, vol. 9, pp. 351-356, 1990.