Robust speaker identification using artificial neural networks

Madhavan Sivathanu Pillai

University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation
https://digitalscholarship.unlv.edu/rtds/2068

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
ROBUST SPEAKER IDENTIFICATION USING
ARTIFICIAL NEURAL NETWORKS

by

Madhavan Sivathanu Pillai
Bachelor of Engineering
University of Madras, India
2003

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2006
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
The Thesis prepared by

MADHAVAN SIVATHANU PILLAI

Entitled

ROBUST SPEAKER IDENTIFICATION USING ARTIFICIAL NEURAL NETWORKS

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative
ABSTRACT

Robust Speaker Identification using Artificial Neural Networks

by

Madhavan Sivathanu Pillai

Dr. Evangelos Yfantis, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

This research mainly focuses on recognizing the speakers through their speech samples. Numerous “Text – Dependent” or “Text – Independent” algorithms have been developed by people so far, to recognize the speaker from his/her speech. In this thesis, we concentrate on the recognition of the speaker from the fixed text i.e. “Text – Dependent”. Possibility of extending this method to variable text i.e. “Text- Independent” is also analyzed. Different feature extraction algorithms are employed and their performance with Artificial Neural Networks as a Data Classifier on a fixed training set is analyzed. We find a way to combine all these individual feature extraction algorithms by incorporating their interdependence. The efficiency of these algorithms is determined after the input speech is classified using Back Propagation Algorithm of Artificial Neural Networks. A special case of Back Propagation Algorithm which improves the efficiency of the classification is also discussed.
# TABLE OF CONTENTS

**ABSTRACT** ............................................................................................................................. iii

**ABBREVIATIONS** ................................................................................................................ vii

**ACKNOWLEDGEMENTS** ........................................................................................................ viii

**CHAPTER 1 INTRODUCTION** ............................................................................................. 1
  Introduction to Speech ............................................................................................................ 2
  1.1 Speech Production ....................................................................................................... 3
    1.1.1 Physical Model .................................................................................................... 3
    1.1.2 Mathematical Model ......................................................................................... 4
  1.2 Speaker information in a speech signal ....................................................................... 5
  1.3 Principles of Speaker Recognition ............................................................................. 7
    1.3.1 Speaker Recognition by Humans ...................................................................... 7
    1.3.2 Issues in Speaker Recognition ......................................................................... 8
    1.3.3 Categories of Automatic Speaker Recognition ................................................ 10
  1.4 Artificial Neural Networks (ANN) ............................................................................ 12
    1.4.1 What is a Neural Network? ............................................................................. 12
    1.4.2 What is a Neuron? ......................................................................................... 12
    1.4.3 Historical Background .................................................................................... 13
    1.4.4 Why use Neural Networks? ............................................................................ 16
    1.4.5 Neural Networks Vs Conventional Classifiers ................................................ 16
  1.5 Organization of Thesis ............................................................................................... 18

**CHAPTER 2 LITERATURE REVIEW** ................................................................................. 19
  2.1 Approaches for Speaker Recognition ...................................................................... 19
    2.1.1 Parametric Approach ...................................................................................... 19
      2.1.1.1 Hidden Markov Models ............................................................................. 19
    2.1.2 Non Parametric Approach ............................................................................... 21
      2.1.2.1 Vector Quantization ................................................................................ 21

**CHAPTER 3 PRE-PROCESSING AND FEATURE EXTRACTION** ........................................... 22
  3.1 Linear Predictive Coding .......................................................................................... 23
  3.2 Pre-Processing ........................................................................................................... 23
    3.2.1 Pre-emphasis .................................................................................................... 24
    3.2.3 Windowing ........................................................................................................ 26
    3.2.4 Autocorrelation Analysis ............................................................................... 27
  3.3 Feature Extraction ..................................................................................................... 29

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Physical Model of Speech Production</td>
<td>3</td>
</tr>
<tr>
<td>1-2</td>
<td>Mathematical Model of Speech Production</td>
<td>4</td>
</tr>
<tr>
<td>1-3</td>
<td>Testing Phase of the Process</td>
<td>9</td>
</tr>
<tr>
<td>1-4</td>
<td>Basic Structure of a closed set identification system</td>
<td>11</td>
</tr>
<tr>
<td>1-5</td>
<td>Basic structure of a speaker verification system</td>
<td>12</td>
</tr>
<tr>
<td>1-6</td>
<td>Basic Structure of a Neuron</td>
<td>13</td>
</tr>
<tr>
<td>1-7</td>
<td>Multi-layer Network</td>
<td>15</td>
</tr>
<tr>
<td>3-1</td>
<td>“Three” plotted in time</td>
<td>24</td>
</tr>
<tr>
<td>3-2</td>
<td>Pre-emphasized “Three” plotted in time</td>
<td>25</td>
</tr>
<tr>
<td>3-3</td>
<td>Energy Distributions of 30 samples of each number, in order from top:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;One&quot;, &quot;Two&quot;, &quot;Three&quot;, &quot;Four&quot;, &quot;Five&quot;</td>
<td>26</td>
</tr>
<tr>
<td>3-4</td>
<td>Energy distribution after alignment, in order from top:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;One&quot;, &quot;Two&quot;, &quot;Three&quot;, &quot;Four&quot;, &quot;Five&quot;</td>
<td>27</td>
</tr>
<tr>
<td>3-5</td>
<td>Frame of a signal</td>
<td>28</td>
</tr>
<tr>
<td>3-6</td>
<td>Windowed frame</td>
<td>28</td>
</tr>
<tr>
<td>3-7</td>
<td>Coarse Root Finding and Root Finding Refinement representing the method to find the LSPs.</td>
<td>34</td>
</tr>
<tr>
<td>5-1</td>
<td>Learning curve with constant learning rate (α)</td>
<td>41</td>
</tr>
<tr>
<td>5-2</td>
<td>Learning curve with variable learning rate (α)</td>
<td>42</td>
</tr>
</tbody>
</table>
ABBREVIATIONS

ANN – Artificial Neural Network(s)
DTW – Dynamic Time Warping
GMM – Gaussian Mixture Model
HMM – Hidden Markov Model
LP – Linear Prediction
LPC – Linear Predictor Coefficients
LSP – Line Spectral Pairs
LSF – Line Spectral Frequencies
BPA – Back Propagation Algorithm
BPNN – Back Propagation Neural Network
L- Linear Layer
N- Non- Linear Layer
MSE – Mean Squared Error
ACKNOWLEDGEMENTS

I am indebted to many people whose help and support made this research effort possible.

Primarily, I would like to express my deep gratitude to my advisor, Dr. Evangelos A. Yfantis for all his untiring support, and seemingly unlimited belief on me throughout, on all stages of my thesis. He has brought the best in me and I have learnt a great deal from him. I am deeply grateful to Dr. Laxmi P. Gewali for all his support and encouragement that has made my masters degree possible. I thank Dr. Yoohwan Kim for his valuable time and advice. I would like to express my sincere thanks to Dr. Georg F. Mauer for having accepted to serve as a graduate college representative for my thesis. I am particularly thankful to him for the initial support which he gave me during the first year of my study in UNLV. I am also thankful to Dr. Shashi Nambisan for financially supporting me during the final phase of my thesis.

I take this opportunity to thank my colleagues Chris Jones, Dean Curtis, Jae Adams, Trenton Pack, Kristal Sauer, Jia Tse, Jacob Zodieru, John Bunch, Niveen and my friends Naveen Karla, Magesh Kumar Padmanabhan, Hina Jain, Meriam and Manisha Chawla for their kindly support.

Last but not the least; I owe everything to my parents for their enormous love, support and the sacrifices they have made for me. I am also grateful to my brother Natarajan for his love and encouragement throughout.

viii
And, most importantly, I acknowledge the Grace of the almighty God which has been the guiding light behind all my endeavors and without whose help this would not have been possible.
CHAPTER 1

INTRODUCTION

Speaker Recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. Humans use voice recognition everyday to distinguish between speakers and genders. Other animals use voice recognition to differentiate among sounds sources. For example, penguin parents can tell which baby chick is theirs by the baby's distinctive call. Similarly, a blind person can accurately classify speakers based solely on their vocal characteristics. In general, speaker recognition can be subdivided into speaker identification (Who is speaking?) and speaker verification (Is the speaker who we think he or she is?). In addition, speaker identification can be closed-set (The speaker is always one of a closed set used for training,) or open-set (Speakers from outside the training set may be examined.). Also, each variant may be implemented as text-dependent (The speaker must utter one of a closed set of words.) or text-independent (The speaker may utter any type of speech) [1].

Speaker identification systems can be closed set or open set. Closed-set speaker identification refers to the case where the speaker is known a priori to be a member of a set of N speakers. Open-set speaker identification includes the additional possibility where the speaker may be from outside the set of N speakers. Open-set speaker identification and speaker verification often use thresholds to determine if a speaker is out of the set.
In this thesis, we explore the ability of a back propagation neural network to perform text-dependent as well as text-independent speaker identification on both open and closed sets of speakers. Our neural network is trained on different sets of acoustical parameters extracted from samples obtained from a closed set of speakers uttering a set of known words. Our primary feature extraction tools are Levinson Durbin Algorithm to extract linear predictor coefficients (LPC), cepstral analysis from LPC coefficients, and computation of line spectral frequencies (LSF) using chebyshev polynomials [4].

Introduction to Speech

Speech signal not only carries the information regarding the message, but also the information about the speaker, the physical and emotional state of the speaker etc. The task of speaker identification has been an active research area from 1960's. Many features have been extracted from the speech signal in the past and used for speaker identification.
1.1 Speech Production

1.1.1 Physical Model

Figure 1–1  Physical Model of Speech Production

When we speak:

Air is pushed from our lung through our vocal tract and out of our mouth comes speech. For certain voiced sound, our vocal cords vibrate (open and close). The rate at which the vocal cords vibrate determines the pitch of our voice. Women and young children tend to have high pitch (fast vibration) while adult males tend to have low pitch.
(slow vibration). For certain fricatives and plosive (or unvoiced) sound, our vocal cords do not vibrate but remain constantly opened. The shape of our vocal tract determines the sound that we make. As we speak, our vocal tract changes relatively slow (on the scale of 10msec to 100msec). The amount of air coming from your lung determines the loudness of your voice.

1.1.2 Mathematical Model

Figure 1-2  Mathematical Model of Speech Production

![Diagram of speech production model]

Table 1-1 Comparison between Physical and Mathematical Models

<table>
<thead>
<tr>
<th>Physical Model</th>
<th>Mathematical Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocal Tract</td>
<td>H(z) (LPC Filter)</td>
</tr>
<tr>
<td>Air</td>
<td>u(n) (Innovations)</td>
</tr>
<tr>
<td>Vocal Cord Vibration</td>
<td>V (voiced)</td>
</tr>
<tr>
<td>Vocal Cord Vibration Period</td>
<td>T (Pitch Period)</td>
</tr>
<tr>
<td>Fricatives and Plosives</td>
<td>UV (Unvoiced)</td>
</tr>
<tr>
<td>Air Volume</td>
<td>G (gain)</td>
</tr>
</tbody>
</table>
The model Figure 1-2 is often called the LPC Model. The model says that the digital speech signal is the output of a digital filter (called the LPC filter) whose input is either a train of impulses or a white noise sequence. The relationship between the physical and the mathematical models is given in the Table 1.1.0.

The LPC filter is given by:

\[ H(z) = \frac{1}{1 + a_1z^{-1} + a_2z^{-2} + \ldots + a_{10}z^{-10}} \]

This is equivalent to saying that the input-output relationship of the filter is given by the linear difference equation.

\[ s(n) + \sum_{i=1}^{10} a_i s(n-i) = u(n) \]

1.2 Speaker Information in a Speech Signal

Speech is the result of a tone varying tract system. The objective of human speech is to convey messages. Speech signal not only carries the message but also encompasses into it the identity of the speaker, the language, the physical and emotional state of the speaker, etc. Of these, recognizing the speaker from his/her speech signal receives a significant attention as it can be put to use in many practical applications where there is need for secure to personal data or in any security based system. The Physiological and behavioral characteristics of a speaker help us to identify a speaker.

The variations in sizes and shapes of the vocal tracts, vocal cords, velum and nasal cavities (physiological characteristics) and their movements (behavioral characteristics) during speech production create differences in the spectra of speech signals.
The Phoneme (basic unit of sound) articulation and the phoneme - phoneme transition, pitch contours, etc. are referred to as “low - level” behavioral information. "Segmental" information refers to the information extracted from individual speech sounds or phones and “super - segmental” information refers to speech phenomena involving consecutive phones. "High - level" behavioral information takes into account characteristic words or phrases of a speaker and also speaking styles.

Humans are able to recognize both speech and speaker because of their ability to capture the temporal changes in speech. The change in formants or the resonant frequencies make the speech signal into meaningful words or phrases. Most of the existing methods for speech and speaker recognition try to extract features from speech, find the average and use it as a template for testing. These methods do not take into account the underlying temporal characteristics, which will lead to ambiguity for words like “task” & “cast” in word recognition because of the same phoneme contents being present in both the words. Human ability of recognizing speakers also likewise depends on the temporal variations in speech.

A Speaker can be identified with more accuracy based on a phrase rather than just a single sound unit or phoneme. This is because, the speaker’s dialect and the variations produced by him will also be captured in the case of a word or a phrase where as the above pieces of information will not be present in a phoneme. The static methods (like vector quantization or some other methods which average the feature vectors) try to find an averaged feature vector for all the phonemes in a word or a phrase, where as temporal methods keep track of the temporal changes.
1.3 Principles of Speaker Recognition

1.3.1 Speaker Recognition by Humans

People can reliably identify familiar voices. About 2-3 seconds of speech is sufficient to identify a voice, although performance decreases for unfamiliar voices [25]. Even if duration of the utterances was increased, but played backward (which distorts timing and articulatory cues), the accuracy decreased drastically. Widely varying performance on this backward task suggested that cues to voice recognition vary from voice to voice, and that voice patterns may consist of a set of acoustic cues from which listeners select a subset to use in identifying individual voices.

Recognition often falls sharply when speakers attempt to disguise their voices [26]. This is reflected in machines, where accuracy decreases when mimics act as impostors. Humans appear to handle mimics better that machines do, easily perceiving when a voice is being mimicked [27]. If the target (intended) voice is familiar to the listener, he/she often associates the mimic voice with it. Certain voices are more easily mimicked than others, which lends evidence to the theory that different acoustic cues are used to distinguish different voices.

Speaker recognition is one area of artificial intelligence where machine performance can exceed human performance – using short utterances and a large number of speakers, machine accuracy often exceeds that of human [27]. This is especially true for unfamiliar speakers, where the training time for humans to learn a new voice is normally very long compared to that for machines. Human performance in adverse conditions was also reviewed in [24], where it was reported that human listeners are adept at using various cues to verify speakers in the presence of acoustic mismatch.
1.3.2 Issues in Speaker Recognition

Speaker recognition by a machine involves three stages.

They are

(1) Extraction of features to represent the speaker information present in the speech signal.

(2) Modeling of speaker features.

(3) Decision logic to implement the identification of verification task. The issues involved in each of these stages are discussed below.

The primary task in a speaker recognition system is to extract features capable of representing the speaker information present in the speech signal. It is known that human beings use high-level features such as speaker dialect, style of speech and verbal mannerisms (for example, use of particular words and idioms, or a particular kind of a laugh) to recognize speakers. Intuitively, it is clear that these features constitute important speaker information. Difficulty arises due to existing feature extraction techniques [24]. Current speaker recognition systems use segmental features such as shape of the vocal tract to represent the speaker-specific information. These features show significant variation across speakers, but they also show considerable variations from time to time for a single speaker. In addition to this, the characteristics of the recording equipment and transmission channel are also reflected in these features [24].

Once a proper set of feature vectors is obtained, the next task in speaker recognition is to develop a model (prototype) for each speaker. The development of speaker modeling is called the training phase. Feature vectors representing the voice characteristics of the
speaker are extracted and used for building the reference models. The performance of a speaker recognition system depends primarily on the effectiveness of the models in capturing the speaker's specific information, and hence this phase plays a major role in determining the performance of a speaker recognition system.

Figure 1–3  Testing Phase of the Process

The final stage in the development of a speaker recognition system is the decision to either accept or reject the claim of a speaker is taken based on the result of matching technique used. Matching techniques are of two types, template matching and probabilistic modeling. Matching generally gives a score which will be measure of how well the test feature vector matches with the reference feature vector. A decision can be taken based on these scores by fixing some threshold appropriately. The block diagram of testing phase and decision logic is shown in the Figure. 1.3.0.
1.3.3 Categories of Automatic Speaker Recognition

Speaker recognition is the general term used to include many different ways of discriminating people based on their voices. The main categories are: Speaker identification and Speaker verification.

In speaker identification, a speech utterance from an unknown speaker is analyzed and compared with models of all known speakers. The unknown speaker is identified as the speaker whose model best matches the input utterance. Figure 1.3.1 shows the basic structure of speaker identification system. Speaker identification can be a closed set identification or an open set identification. In a closed set identification, it is assumed that the test utterance belongs to one of N enrolled speakers (N decisions). In case of open set identification, there is an additional decision to be made to determine whether the test utterance was uttered by one of the N enrolled speakers or not, that is, there are N+1 decision levels.
Speaker verification aims to accept or reject the claim of the speaker based on the samples of his speech. If the match between test and reference is above a certain threshold, the claim is accepted. A high threshold makes it difficult for impostors to be accepted by the system, but at the risk of rejecting the genuine person. Conversely, a low threshold ensures that the genuine person is accepted consistently, but at the risk of accepting impostors. Figure 1.3.2 shows the basic structure of a speaker verification system.
1.4 Artificial Neural Networks (ANN)

1.4.1 What is a Neural Network?

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (Artificial neurons) working in unison to solve specific problems. ANN, like people, learns by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANN as well.

1.4.2 What is a Neuron?

An artificial neuron (also called a "node" or "Nv neuron") is a basic unit in an ANN. Artificial neurons are simulations of biological neurons, and they are typically functions
from many dimensions to one dimension. They receive one or more inputs and sum them to produce an output. Usually the sums of each node are weighted, and the sum is passed through a non-linear function known as an activation or transfer function. The canonical form of transfer functions is the sigmoid, but they may also take the form of other non-linear functions, piecewise linear functions, or step functions. Generally, transfer functions are monotonically increasing.

For a given artificial neuron, let there be m inputs with signals $x_1$ through $x_m$ and weights $w_1$ through $w_m$.

The output of neuron $k$ is

$$y_k = \varphi \left( \sum_{j=0}^{m} w_{kj} x_j \right)$$

where $\varphi$ (Phi) is the transfer function.

Figure 1-6 Basic Structure of a Neuron

1.4.3 Historical Background

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major
setback and several eras. The Development of ANN was first reported in the early forties by McCulloh and Pitts. In this model, a neuron fires if the sum of its excitatory inputs exceeds its threshold. This happens, as long as it receives no inhibitory input. Using this model, it is possible to construct a network that can compute any logical function. Rosenblatt found that the McCulloh - Pitts model was unbiologic al. In order to overcome the deficiencies in the McCulloh – Pitts model, he found out a new model, namely the perceptron model, which could be utilized to learn and generalize. Further, he investigated several mathematical models, which included competitive learning or self-organization, and forced learning which is somewhat similar to reinforcement learning.

In addition to the above two types of learning, the concept of supervised learning was developed and incorporated in the adaptive linear element model (ADALINE). The ADALINE was found by Widrow and Hoff. The ADALINE is a single neuron, which uses a method to descend the gradient of the error, by using the supervised learning.

The ADALINE is a linear neuron, and it is helpful to discriminate the patterns, which are linearly separable. The Concept of multi layer ADALINES or multi layer network was developed for patterns which were not linearly separable. The structure of a multi layer network is shown in Figure 1.4.1. The training of the multi layer network was first explained by Werbos as Back-Propagation Algorithm in his PhD dissertation. His work did not become popular. Rumelhart and his group published the parallel processing, a two-volume collection of studies on a broad variety of neural network configurations. Through these books, the concept of back-propagation algorithm became popular for training a multi layer network. Lippmann briefed the concept of different algorithms in his tutorial paper, and he still made neural networks more popular.
In most of the supervised training methods, the patterns are presented in a predetermined sequence in a cycle. Normally, the order of presentation of the patterns is maintained in all the cycles. Ridgway found in his thesis that cyclic presentation of patterns could lead to cyclic adaptation. These cycles would cause the weights of the entire network to cycle, by preventing convergence. Various error criteria have been tried by Zakai and by Walsch and Widrow, for better convergence of the network. BPA have been analyzed by shoemaker, Carlin and Shimabukuro. Analysis of BPA with respect to mean weight behavior was done by Bershad, Shynk and Feintuch (1993).

Figure 1-7    Multi-layer Network
1.4.4 Why use Neural Networks?

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

1.4.5 Neural Networks Vs Conventional Classifiers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to
problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.
1.5 Organization of Thesis

Chapter 2 provides a brief insight of previous research in this area. Chapter 3 explains the different preprocessing techniques followed by feature extraction algorithms used in speaker recognition. In chapter 4 we discuss about the database used and propose a way in which several individual features can be combined using the Back propagation neural network. We also discuss the results of the algorithm with the existing database. Conclusions and further research recommendations are discussed in chapter 5.
CHAPTER 2

LITERATURE REVIEW

2.1 Approaches for Speaker Recognition

2.1.1 Parametric Approach

Models which assume a Probability density function are called as Parametric. Some of the parametric approaches are Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM). Here we briefly review about HMMs.

2.1.1.1 Hidden Markov Models

The Hidden Markov Models (HMM) is a doubly embedded stochastic process where the underlying stochastic process is not directly observable. HMMs can be used as probabilistic speaker models for both text-dependent and text-independent speaker recognition [10] [11] [12]. An HMM not only models the underlying speech sounds but also the temporal sequencing of the sounds. This temporal modeling is advantageous for text-dependent tasks. For text-dependent speaker recognition task, HMM based methods have achieved significantly better recognition accuracies than Dynamic Time Warping – based methods (DTW) [13] [14]. But this temporal modeling does not aid in the case of text-independent system, because the sequence of sounds in the test utterance need not be the same as that in the training utterance.
In training phase, an HMM for each speaker is obtained by estimating the parameters of the model using feature vectors from the training data. The parameters of HMM are [2]:

- **State –transition probability distribution**: It is represented by \( A = [a_{ij}] \), where
  \[
a_{ij} = P(q_{t+1} = j | q_t = i) \quad 1 \leq i, j \leq N
\]
defines the probability of transition from state \( i \) to \( j \) at time \( t \).

- **Observation symbol probability distribution**: It is given by \( B = b_j(k) \), in which
  \[
b_j(k) = P(\bar{o}_t = \bar{v}_k | q_t = j) \quad 1 \leq k \leq M
\]
defines the symbol distribution in state \( j \), \( j = 1,2,...N \).

- **The initial state distribution**: It is given by \( \Pi = [\pi_i] \), where
  \[
  \pi_i = P(q_1 = i) \quad 1 \leq i \leq N
\]

Here, \( N \) is the total number of states, and \( q_i \) is the state at time \( t \). \( M \) is the number of distinct observation symbols per state and \( O_t \) is the observation symbol at time \( t \).

In the testing phase, \( P(O | \lambda) \) for each model is calculated, where \( O = (o_1,o_2,o_3,...o_T) \) is the sequence of the test feature vectors. The goal is to find the probability, given the model, that the test utterance belongs to that particular model. The speaker model that gives the highest score is declared as the identified speaker. GMM corresponds to the single-state continuous ergodic HMM [15].
2.1.2 Non Parametric Approach

2.1.2.1 Vector Quantization

In the Vector Quantization-based method, codebooks are generated for each speaker. Codebooks consist of a small number of representative feature vectors to characterize a speaker [16] [17]. In the training phase, the feature vectors are grouped into certain fixed number of clusters. The representative data is the centroid vector of the cluster. A set of such centroid vectors is known as codebook. Assuming that each speaker's feature vectors will have different distributions, the codebook generated for each speaker will be unique to him. A separate codebook is generated for each speaker. In the testing phase, the test utterance is vector quantized using the codebook of each reference speaker. The VQ distortion, which is nothing but the distance of a test codebook. The distortion is accumulated over the entire test utterance. The accumulated distance is used to arrive at a decision for speaker recognition. In [18] weights are given to the distortion score for individual codebook elements, assuming that different codebook elements encode different levels of speaker-specific information. Moreover, there can be more than one codebook for each speaker [19].

VQ-based speaker recognition systems are easy to build and are shown to give good results. VQ based speaker recognition system can be evaluated in both text-dependent and text-independent modes [20]. VQ can be considered as a degenerate case of single state HMM with observation probability being replaced by the distance measure [15].
CHAPTER 3

PRE-PROCESSING AND FEATURE EXTRACTION

Ordinarily, the speech signal is a highly complex signal and significant up to 4 kHz frequency. In order to preserve data up to 4 kHz, it needs to be sampled at more than 8 kHz. The frequency response of the speech signal shows three or four dominant frequencies otherwise called resonant frequencies or formants and corresponds to the vocal tract resonances. The difference between a voiced and unvoiced speech is that the vocal cord vibrate with certain frequency called “Pitch frequency” for voiced speech while the vocal cords do not move for an unvoiced speech. It is found out that whether the speech is voiced or unvoiced, the frequency response contains peaks at the vocal tract resonances. The continuous movement of the formants makes the speech intelligible and enables human beings to understand. The movement of the formants represents the temporal characteristics of the speech. The vocal tract at a fixed position or the static feature characteristics, although used by many researchers in the past for speaker recognition [Atal (1974), Burton (1987), Dante (1979), Sambur (1976), Bogner (1981)] may not provide a good representation for speaker characteristics than capturing its temporal variations because of the fact that features corresponding to the movement of vocal tract can characterize a speaker more than the features at fixed positions. In order to extract features from the speech signal, there are many analysis methods, of which, the one based on the estimation of future values, the Linear predictive coding analysis
method and the other based on the separation of the pitch information and the vocal tract information, the cepstrum analysis are important.

3.1 Linear Predictive Coding

The linear prediction model [1] is one of the oldest and the best speech production model for speech/speaker analysis. The popularity of the model is due to the following [2]. The model is an all pole model and suits very well for the voiced speech signal. The separation of the effect of glottis, vocal tract and the mouth makes the model close to real production of speech. The unvoiced speech is taken care by modeling it as an effect of random signal on the voice tract.

(Refer to Figure 1-2)

The poles of the model extract the vocal tract resonances whose movements are a function of the speech being produced and also of the speaker. Conjugate pole pairs and real poles, represent the dominant frequencies and spectral slopes respectively, and these are captured in the model parameters.

The model is analytically tractable and the algorithms already available make the extraction of the model parameters easy. Many other parameters which are useful in recognition purposes can be derived from LPC parameters.

3.2 Pre-Processing

Pre-Processing is a set of operations which is performed on the input speech signal so that feature extraction algorithms can make the maximum use of the speaker dependent characteristics present in the digitized signal.
3.2.1 Pre-emphasis

The Input Speech signal is filtered with a first order FIR filter to spectrally flatten the signal. We used one of the most widely used pre emphasis filter of the form.

\[ H(z) = 1 - a z^{-1}, \text{where } a = 15/16. \]

Figure 3–1 “Three” plotted in time.

Fig [3.0] shows the signal before Pre-emphasis and Fig [3.1] shows the signal after Pre-emphasis is performed on the signal. The signal was sampled at a frequency of 8 kHz. Observe that the filter removes the DC Component of the signal.
3.2.2 Normalization

After Pre-emphasis, each word has its energy normalized. Based on the energy distribution along the temporal axis, it is computed the center of gravity, and this information is used as reference for temporal alignment of the words. Fig [3.2] and Fig [3.3] show examples of temporal alignment. The energy of each word was computed using 60 non overlapping windows.
Figure 3-3 Energy Distributions of 30 samples of each number, in order from top:
"One", "Two", "Three", "Four", "Five"

3.2.3 Windowing

Each individual frame is windowed to minimize the signal discontinuities at the borders of each frame. If the window is defined as $w[n]$, $0 < n < N-1$, then the windowed signal is

$$X_1[n] = X_0[n] \cdot W[n], \text{ where } 0 < n < N-1.$$  

We used a Hamming Window, a typical window used for the autocorrelation method of LPC. Fig [3.5] and Figure [3.6] show an example of a non-windowed and windowed frame. Observe the borders of the signal.
3.2.4 Autocorrelation Analysis

Each frame of windowed signal is next auto correlated to give

\[ y_j = \sum_{k=0}^{N-1} x_k^* \cdot x_{j+k} \]

Where \( j = 0, 1 \ldots p \) where \( p \) is the order of LPC analysis.

where the highest autocorrelation value, \( p \) is the order of the LPC analysis. Typically, values of \( p \) from 8 10 16 have been used, with \( p=8 \) being the value used for the system to be described in [2]. A side benefit of the autocorrelation analysis is that the zero\textsuperscript{th}
Figure 3–5  Frame of a signal

![Frame of a signal](image)

autocorrelation $R_t[0]$, is the energy if the $t^{th}$ frame. The frame energy is one of the important parameter for speech-detection systems.

Figure 3–6  Windowed frame

![Windowed frame](image)
3.3 Feature Extraction

Feature Extraction is a general term for methods for constructing combinations of the variables which get around classification problems but still describe the data sufficiently and accurately. Here, in this thesis we considered the following coefficients as the features required to identify speaker.

1. LP Coefficients extracted using Levinson-Durbin algorithm [1]
2. Cepstral Coefficients extracted using the LP coefficient set [8]
3. LSF extracted from LSP’s which are found using LP coefficient set [9]

3.3.1 LPC analysis

In applying time series analysis to the output of the previous system, each continuous signal \( s(t) \) is sampled to obtain a discrete-time signal \( S_n \), also known as time series, where \( n \) is an integer variable and \( T \) is the sampling interval. The sampling frequency is then \( f_s = 1/T \).

A major concern of time series analysis has been the estimation of power spectra, cross-spectra, coherence functions, and autocorrelation and cross-correlation functions. A more active concern at this time is that of system modeling. It is clear that if one is successful in developing a parametric model for the behavior of some signal, then that model can be used for different applications, such as prediction or forecasting, control, and data compression.

One of the most powerful models currently in use is that where a signal \( S_n \) considered to be the output of some system with some unknown input \( u_n \) such that the following relation holds:
The gain $a_k$ and $G$ is the parameter of the hypothesized system. Equation (1) says that the "output" $S_n$ is a linear function of past outputs and present and past inputs. That is, the signal $S_n$ is predictable from linear combination of past outputs and inputs. Hence the name linear prediction. [1]

This is of the form $Ra = r$

$$\tau = [\tau(1) \tau(2) \ldots \tau(3)]^T$$

is the autocorrelation vector.

$$a = [a_1 a_2 \ldots a_p]^T$$

is the filter coefficients vector and

$$R = \begin{bmatrix}
\tau(0) & \tau(1) & \tau(2) & \ldots & \tau(p - 1) \\
\tau(1) & \tau(0) & \tau(1) & \ldots & \tau(p - 2) \\
\tau(2) & \tau(1) & \tau(0) & \ldots & \tau(p - 3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\tau(p - 1) & \tau(p - 2) & \tau(p - 3) & \ldots & \tau(0)
\end{bmatrix}$$

is the autocorrelation matrix.

This matrix is non-singular and gives the solution.

Here, in this Thesis linear predictor coefficients are found using Levinson – Durbin Recursion. [2]

3.3.2 Levinson – Durbin Algorithm

The Durbin's algorithm is used to solve equation systems where the elements across the diagonal are identical and the matrix of coefficients is symmetry (Toeplitz matrix).
The complexity of the method, consists in solving $p^2 + O(p)$ operations and the memory required is only $2p$ locations. The equations to solve are in the form:

$$\sum_{k=1}^{p} a_k R_n(i-k) = R_n(i)$$

with $1 \leq i \leq p$.

The complete Durbin algorithm is:

$$E^{(0)} = R(0)$$

$$k_i = \left[ R(i) - \sum_{j=1}^{i-1} a_j^{i-1} R(i-j) \right]/E^{i-1}$$

$$a_i^{(i)} = k_i$$

$$a_j^{(i)} = a_j^{(i-1)} = k_i a_{i-j}^{(i-1)}$$

$$E^{(i)} = (1-k_i^2)E^{(i-1)}$$

This equations are solved recursively for $i=1,2,...,p$ and the final solution is given by:

$$a_j = a_j^{(p)}$$

where $1 \leq j \leq p$.

3.3.3 LPC Parameter Conversion to Cepstral coefficients

The LPC cepstral coefficients, $c_m$, are a very important LPC parameter used in speech/speaker recognition. They can be derived directly from the set of LPC coefficients $a_i$ for $i=1,...,p$, using the recursion.

$$c_0 = r(0),$$

$$c_m = a_m + \sum_{k=1}^{m-1} \frac{k_c}{m} c_k a_{m-k},$$

where $1 < m < p$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Where $m > p$,

The Cepstral Coefficients, which are the coefficients of the Fourier transform representation of the log magnitude of the spectrum, have been shown to be more robust for speech recognition that the LPC coefficients. Generally, it is used as a Cepstral representation with $Q>p$ coefficients, where $Q \sim (3/2) p$.

3.4. LPC Conversion to Line Spectral Frequencies (LSF)

Line Spectral Frequencies (LSF’s) are an alternative to the direct form predictor coefficients or the lattice form reflection coefficients for representing filter response [4]. The direct form coefficient representation of the LPC filters is not conducive to efficient quantization. These parameters are preferable because they have a relatively low spectral sensitivity. LSF’s are an alternative parameterization of the filter with a one to one correspondence with the direct form predictor coefficients. The concept of LSF was introduced by Itakura [9]. This was efficiently computation of line spectral frequencies. This involves an iterative root finding algorithm for a series representation in Chebyshev polynomials. The algorithm developed in [4] is simple in structure and constraints the maximum number of function evaluations. These considerations are important as they are used in a real time environment.

The starting point for deriving the LSF’s is the response of the prediction error filter with $P$ coefficients.

$$A(z) = 1 - \sum_{k=1}^{P} a(k) z^{-k}.$$
The \{a (k)\} are the direct form predictor coefficients. The corresponding all-pole synthesis filter is \(1/A (z)\). A minimum phase prediction error filter (i.e., one with all its roots within the unit circle) has a corresponding synthesis filter which is stable.

\[
F_1(z) = A(z) + z^{-(p+1)} A(z^{-1}),
\]

\[
F_2(z) = A(z) - z^{-(p+1)} A(z^{-1}).
\]

A symmetric polynomial \(F_1 (z)\) and an ant symmetric polynomial \(F_2 (z)\) related to \(A(z)\) are formed by adding and subtracting the time-reversed system function. The roots of these two auxiliary polynomials determine the line spectral frequencies. The two polynomials also have the interpretation of being the system polynomials for a \(p+1\) coefficient predictor derived from a lattice structure.

3.4.1 Root Finding Method using Chebyshev polynomials

LSPs, which are the roots of two particular polynomials, always lie in the range \((-1, 1)\). (The guaranteed roots at 1 and -1 are factored out.) The block finds the LSPs by looking for a sign change of the two polynomials' values between points in the range \((-1, 1)\). The block searches a maximum of \(k^{(n-1)}\) points, where

- \(n\) is the value of the "Root finding coarse grid points" parameter
- \(k\) is the value of the "Root finding bisection refinement" parameter.
Figure 3-7  Coarse Root Finding and Root Finding Refinement representing the method to find the LSPs.

Coarse Root Finding: LSPs are roots of two particular polynomials related to the input LPCs. Check signs of the two polynomials at evenly spaced points to find all intervals containing a sign change. Output any roots (LSPs) found.

Root Finding Refinement: Whenever Coarse Root Finding identifies an interval containing a sign change, repeatedly bisect the interval to better approximate the root (LSP value).

Once the roots are determined, the corresponding LSF’s are given by \( \omega_i = \arccos(x_i) \) where \( x_i \) are LSPs which maps the semi circle in the z-plane to the real interval [-1, +1].
CHAPTER 4

CLASSIFICATION AND RECOGNITION ALGORITHM

4.1 Back propagation Algorithm

A Back propagation network consists of at least three layers of units: an input layer, at least one intermediate hidden layer, and an output layer. In contrast to the IAC and Hopfield networks, connection weights in a Back propagation network are one-way. Typically, units are connected in a feed-forward fashion with input units fully connected to units in the hidden layer and hidden units fully connected to units in the output layer. When a Back propagation network is cycled, an input pattern is propagated forward to the output units through the intervening input-to-hidden and hidden-to-output weights.

With Back propagation networks, learning occurs during a training phase in which each input pattern in a training set is applied to the input units and then propagated forward. The pattern of activation arriving at the output layer is then compared with the correct (associated) output pattern to calculate an error signal. The error signal for each such target output pattern is then back propagated from the outputs to the inputs in order to appropriately adjust the weights in each layer of the network. After a Back propagation network has learned the correct classification for a set of inputs, it can be tested on a second set of inputs to see how well it classifies untrained patterns. Thus, an important consideration in applying Back propagation learning is how well the network generalizes. The backpropagation network was probably the main reason behind the...
repopularisation of neural networks [23]. The original network utilised multiple layers of weight-sum units of the type $f = g(w'x + b)$, where $g$ was a sigmoid function. Training was done by a form of stochastic steepest gradient descent. The employment of the chain rule of differentiation in deriving the appropriate parameter updates results in an algorithm that seems to 'backpropagate errors', hence the nomenclature. However, BPNN is essentially a form of gradient descent.

The BPNN currently used in this thesis consists of 36 nodes in the input layer, 32 nodes in the 1st hidden layer, 6 nodes in the 2nd hidden layer, 32 nodes in the 3rd hidden layer and 2 nodes in the output layer. This can be represented as 36 L, 32 N, 6 N, 32 N, 2L. The learning rate was set to 0.09 initially and momentum was set to 0.25. Number of Iterations was set to 100,000.

The reason for smaller number of nodes in hidden layer is to cause some compression to the input vectors to a lower dimension like in the principal component analysis (Diamantras & Kung, 1996). The dimension compression layer in the middle is used primarily for capturing the distribution of input vectors in the feature space.

4.1.1 Decoding the Output layer

The Output layer consists of 2 nodes each of them having a binary output. So one particular configuration of output nodes point to one speaker. For instance, Output node 1 can either point to 1 or 0. Similarly it's the same with Output node 2. So when the network is trained with feature vector of each speaker, the speaker one would point to the output node pairs of (0,0), speaker two to (0,1) speaker three to (1,0) and speaker four to (1,1) respectively. Number of output nodes can be changed depending upon the number of speakers required. Generally, it's not a good practice to point the nodes to 0 or
1 as the sigmoid curve never converges to either of the values. Hence binary "1" was replaced with 0.75 and binary "0" was replaced with 0.25 for the output nodes.

4.2 Database

4.2.1 Training Data

Since, our objective was to identify the speakers with a fixed text; we chose the fixed text to be "Thesis". Initially 4 speakers were considered 3 Male and 1 female. 10 samples of the same word was recorded from each speaker. 5 samples from each speaker were used for training. Similarly different samples for training out the case of "variable text" were recorded.

4.2.2 Testing Data

5 samples from each speaker uttering the word "Thesis" is stored separately. Similarly 5 samples of variable text were collected from each speaker to test the case of "Text – Independence". Voice of one speaker is considered whose samples are not used for training the database. This was used to check the behavior of the network when an unknown speaker is introduced.

4.3 Recognition Algorithm

Once the Network is trained with samples provided by the speakers, it is tested by passing the vector (12 LPCs + 12 Cepstral Coefficients + 12 LSFs) of each frame into the Forward pass of the Back Propagation Neural Network. The output of the sample used for testing is compared with actual output of the speakers in the data set. Small value of tolerance was set for each node while testing. This value depends on the mean squared error obtained after training the data.
While testing, every frame of the test sample is input through the Forward pass of the Back Propagation Neural Network. Each frame is classified according to the output using a methodology similar to "Maximum Likelihood Estimation". When the complete wave sample is tested, the speaker with a maximum likelihood is termed as the Probable speaker of the tested sample.
CHAPTER 5

RESULTS

5.1 Fixed text results

After training with all the speakers, all the ten samples were tested. Some interesting results were obtained. The Speakers were Madhavan, Naveen, Hina and Chris

Table 5-1 Confusion matrix for the word “Thesis”

<table>
<thead>
<tr>
<th></th>
<th>Madhavan</th>
<th>Naveen</th>
<th>Hina</th>
<th>Chris</th>
<th>Unknown Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Madhavan</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Naveen</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hina</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chris</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Unknown Speaker</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>
### 5.2 Variable text results

Table 5-2  Confusion matrix for variable text

<table>
<thead>
<tr>
<th></th>
<th>Madhavan</th>
<th>Naveen</th>
<th>Hina</th>
<th>Chris</th>
<th>Unknown Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Madhavan</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Naveen</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hina</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chris</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Unknown Speaker</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

### 5.3 Fixed Test Results with LPC and Cepstra as Source Features.

Table 5-3  Confusion matrix for fixed text

<table>
<thead>
<tr>
<th></th>
<th>Madhavan</th>
<th>Naveen</th>
<th>Hina</th>
<th>Chris</th>
<th>Unknown Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Madhavan</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Naveen</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hina</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Chris</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Unknown Speaker</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
5.4 Effects of simulated annealing

Simulated annealing is a method for searching a landscape of possibilities for the best solution to a complex problem, as expressed by the lowest (or highest) point on that landscape. The name comes from the process of annealing, during which a material is first heated and then slowly cooled to enhance ductility and strength. During annealing, the component atoms of the metal are allowed to settle into a lower energy, more stable arrangement than prior to the process.

Similarly, learning rate ($\alpha$) of the Back Propagation Network was subjected to simulated annealing and the corresponding change in the results was noted. Some interesting observation was found.

Figure 5–1   Learning curve with constant learning rate ($\alpha$)
Many previous works have suggested having a small learning rate to train the network to give a good convergence on the learning curve. This was applied to the Back Propagation Network used in this application. But, with a constant learning rate of $\alpha = 0.09$, the network didn't seem to take the path of a sigmoid learning curve [Fig. 5.1]. The network seemed to lose the knowledge gained repeatedly after some epochs.

Figure 5–2  Learning curve with variable learning rate (\(\alpha\))

Hence, the concept of simulated annealing was applied on the learning rate and the result can be observed from the figure. Initially the learning rate was kept as 0.54 and reduced by 0.01 every 100 epochs until the learning rate reached to 0.18.
5.4.1 Comparison of the results

With a constant learning rate, the Mean Squared Error (MSE) seemed to drop less than 0.05 for 10,000 epochs whereas with variable learning rate, the Mean Squared Error (MSE) started with the value of 0.062364 and followed the path of sigmoid curve down to 0.06235 for less than 4000 epochs. Moreover, there were no local minima observed in any part of the curve when the learning rate was variable.
In this thesis, we have found an effective way of recognizing the speakers by combining different features namely LPCs, Cepstral Coefficients and LSFs of a particular speaker and classifying and recognizing them with 36 L, 32 N, 6 N, 32 N, 2 L Back Propagation Network. After all the extensive data acquisition, network training, and network testing, we have drawn out two interesting conclusions for this thesis. First of all, LPC, Cepstral Coefficients and LSFs together form a robust feature to characterize speakers very well, even in noisy conditions. It is also found that LPC’s and Cepstral Coefficients perform very well than LSFs in identifying speakers with limited amount of data. However, this might change as the size of the data base increases. Our network obviously worked very well for text dependent speaker identification; however, it also showed very promising results for text independent speaker identification. If this network is provided with more extensive list of training data containing a diverse group of source feature information, then by using our algorithm, our network can definitely identify the speakers independent of what he or she says. Initially, the aim was to tackle the challenging problem of text dependent speaker identification. As we obtained fairly good results for a text dependent speaker recognition system, we extended even more and delved into the problem of text independent speaker recognition. Although our results were limited by the amount of training data we had, we still found the results to show
potential. However, the problem of automatic speaker/voice recognition is very broad field with many problems yet to be solved. Further development on this work includes training the network to recognize unauthorized speakers; however, again this depends on the training data. There is a good chance of improvement in the results for text independent speaker identification when a separate network is used for each speaker. This will reduce the dimensionality of the problem. Moreover, the time spent on training a new speaker will be less and hence increased number of epochs can be spent on characterizing a particular speaker's information in the network. This should help the network to point out the unauthorized speakers.
REFERENCES


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


VITA

Graduate College
University of Nevada, Las Vegas

Madhavan Sivathanu pillai

Local Address:
4248 Chatham Circle, Apt 1
Las Vegas, NV 89119.

Degree:
Bachelor of Engineering, Computer Science, 2003
University of Madras, India

Thesis Title: Robust Speaker Identification using Artificial Neural Networks

Thesis Examination Committee:
Chairperson, Dr. Evangelos A.Yfantis, Ph. D.
Committee Member, Dr. Laxmi P. Gewali Ph. D.
Committee Member, Dr. Yoohwan Kim Ph. D.
Graduate faculty Representative, Dr. Georg F. Mauer Ph. D.