A STUDY ON ISOLATED-WORD MYANMAR SPEECH RECOGNITION VIA ARTIFICIAL NEURAL NETWORKS

By

Nan Phyu Phyu Hsan
B.C.Tech. (Hons.)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Technology
(M.C.Tech.)

University of Computer Studies, Yangon
November 2018

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and sincere appreciation to all persons who have contributed directly or indirectly towards the completion of this thesis and helped make this dissertation possible.

Firstly, I would like to express my gratitude and sincere thanks to Dr. Mie Mie Thet Thwin, Rector of the University of Computer Studies, Yangon, for kindly allowing me to develop this thesis.

Secondly, I would like to express my gratitude to Professor Dr. Khin Than Mya, Head of the Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her kindness and administrative support throughout the development of the thesis.

Thirdly, I am deeply thankful to Professor Dr. Myat Thida Mon, Head of the Faculty of Computer Systems and Technologies, University of Information Technology, and Dr. Win Pa Pa, Associate Professor, Natural Language Processing Lab, University of Computer Studies, Yangon, for agreeing to be on my dissertation committee. They are more than generous with their expertise and precious time to review my thesis.

I would also like to express my sincere thanks to Professor Dr. Aung Htein Maw, the former dean of the Master course, for his administrative support throughout the development of the thesis.

I am also grateful to Dr. Thet Thet Khin, Associate Professor and Dean of the Master course, Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her invaluable guidance and administrative support during the development of my thesis.

I am also deeply thankful to my supervisor, Dr. Twe Ta Oo, Assistant Lecturer, Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her invaluable suggestion regarding the thesis topic and giving me detailed guidance, encouragement, and patient supervision throughout the preparation of the thesis.

I am also grateful to Daw Aye Aye Khine, Associate Professor and Head of the Department of Language, University of Computer Studies, Yangon, for kindly checking the grammar of my thesis book.

I would also like to express my heartfelt appreciation to all the teachers from the University of Computer Studies, Yangon, who attended the thesis seminars, for their support, valuable suggestions, helpful hints, and fair criticisms.

Last but not least, I also dedicate this thesis to all of my teachers who have kindly taught me everything and my colleagues from University of Computer Studies, Yangon, for their advice and fullest cooperation to the completion of this thesis.

ABSTRACT

Speech is an easiest way to communicate with each other. Digital processing of speech signals is very important for speedy and precise automatic speech recognition systems. Speech recognition is the capability of an electronic device to understand spoken words, i.e. the process of decoding an acoustic speech signal captured by a microphone or a mobile phone to a set of words. It is a technology that can be useful in many applications of our daily life, e.g. mobile communications, and has also become a challenge towards human-computer interfacing (HMI) technology.

This thesis aims to develop an efficient speech recognition system for isolated Myanmar words based on the theories of digital signal processing, speech processing, and artificial neural network techniques. The proposed system is intended to achieve speaker dependent recognition as well as speaker independent recognition.

A speech signal is combined with voice and unvoiced sounds. In addition, each word in the speech is typically surrounded with silence, which may be a hindrance for successful speech recognition. So firstly in this system, the input speeches are manually preprocessed by using the Audacity software in order to detect the start and end points of words and remove unwanted parts like silences in speeches. This system then extracts the acoustically representative features like Mel-Frequency Cepstral Coefficients from the preprocessed speech signals. Finally, those features are used to train a recognition model of neural network with the Backpropagation algorithm for classification and recognition of input speeches. Based on the knowledge learned during training, the recognition model is expected to recognize the same speech by untrained new speakers (i.e. speaker independent recognition).

The proposed system in this thesis is developed to recognize twenty isolated Myanmar words, which are the names of the cities in Rakhine state, Shan state, and Kachin state in Myanmar. This system consists of a database which is made up of training and testing data sets with 2400 and 400 utterances respectively. The training words are uttered by 10 speakers (4 males and 6 females) who are university graduate students. As for speaker independent recognition, testing utterances are the same words as in training but uttered by different speakers than the ones participated in training. The

proposed system is implemented in MATLAB and experimental results show that it achieved the recognition rate of about 93.5% for known speakers (i.e. speaker dependent) and 76.5% for unknown speakers (i.e. speaker independent).

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

This chapter firstly presents an analysis of the fundamentals of speech recognition. Following the basic concepts, it discusses the related works in the literature of speech recognition. Finally, this chapter is concluded by describing the objectives and organization of the thesis.

1.1 Speech recognition

Speech is the easiest way of communication for humans. The aim of speech recognition is to create machines that are capable of receiving speech such as spoken commands from humans and taking action upon those spoken information. It is the process of recording a person's voice by a microphone, converting the signal from analog sound waves to digital audio by hardware, and processing the audio data by software which interprets the sound as individual words. It also means understanding the voice by a computer and performing any required task and thus also called as “automatic speech recognition (ASR)” or “computer speech recognition”.

The ASR is a popular and challenging area of research in human computer interactions [11]. Speech communication is not only for face-to-face interaction but also between individuals at any moment and anywhere via a wide variety of modern technologies. Thus, a promising speech communication technique for human-to-machine interaction has come into being.

Although speech recognition was once thought to be a straightforward problem, many decades of research has revealed the fact that speech recognition is a rather difficult task to achieve, with several dimensions of difficulty due to the non-stationary nature of speech, vocabulary size, speaker dependency issues, etc. However, there have been quite remarkable advances and many successful applications in the speech recognition field, especially with the advances in computing technology beginning in the 1980s [27].

Speech recognition applications include voice based user interfaces such as voice dialing (e.g. “Call home”), call routing (e.g. “I would like to make a collect call”), demotic appliance control, search (e.g. find a podcast where particular words were

spoken), simple data entry (e.g. entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g. word processors or email), and aircraft (usually termed as Direct Voice Input) [40]. Speech recognition techniques enable us to use our voice as a command to execute an application. It is similar to using a mouse, a keyboard, and a phone keypad to give a command to run an application; we just talk to the system with voice recognition capability and it reacts accordingly. Additionally, recognition techniques help us to identify the speaker's identity based on his/her voice and to control access to services such as voice dialing, voice mail, banking by telephone, telephone commerce, database access services, security control for confidential information areas, and remote access to computers. The task of a speech recognizer is to automatically determine the spoken words regardless of the variability introduced by speaker identity, manner of speaking, and environmental conditions.

Speech recognition systems can be classified into two categories: speaker dependent and speaker independent systems.

Speaker dependent systems need to be trained by the individual who will be using the system. These systems are easier and cheaper to be developed, and also capable of achieving a high command count and better than 95% accuracy for word recognition [31]. This is the most common approach employed in software for personal computers. The drawback to this approach is that the system responds accurately only to the individual who trained the system.

Speaker independent systems are trained to respond to a word regardless of who speaks. The system must therefore respond to a large variety of speech patterns, enunciations, and inflections of the target word. These systems are the most difficult and expensive to be developed, and the command word count is usually lower than speaker dependent systems. However, high accuracy can still be maintained within the processing limits. Industrial requirements more demand for speaker independent voice systems, such as the AT&T system used in telephones [31].

Alternatively, speech recognition systems can also be categorized based on the fact that whether the speech is isolated or continuous.

An **isolated-word** system processes a single word at a time – there must be a pause between saying each word. This is the simplest form of recognition to perform because the end points are easier to find and pronunciation of a word tends not to affect others. Thus, the occurrences of words are more consistent and easier to recognize [41].

A **continuous speech** system processes speech in which words are connected together, i.e. not separated by pauses. Compared to isolated-word, continuous speech is more difficult to handle because of the problem of "coarticulation". In continuous speech, the production of each phoneme is affected by the production of surrounding phonemes and similarly the start and end points of words are affected by the preceding and following words. In addition, the recognition of continuous speech is also affected by the rate of speech (fast speech tends to be harder) [41].

One important fact concerning a speech recognition system is the accuracy, which is also the main constraint of such a system. Accuracy means the word recognition rate of a recognition model and evaluated based on the speaker dependent or independent testing. Speaker dependent testing means that the recognizer is tested with known speakers who have already participated in training, whereas speaker independent testing means that the recognizer is tested with unknown speakers who did not participate in training. In reality, it is very difficult to design a speech recognition system with 100% accuracy because of the grammar and punctuation, homonyms and unusual words, ambient noise, overlapping speech, and number of speakers. Most of the speech recognition techniques that have been proposed in the literature could recognize only with 70% to 80% accuracy [35]. The following section discusses some of the previous works in the speech recognition literature.

1.2 Related works

Nowadays, speech recognition technologies are very helpful for society because electronic devices with voice recognition capability can replace human in dangerous working environments. Researches for speech recognition have been started since 1930. Since then, there had been a lot of research experiments and achieved results in various languages throughout the world. This section introduces some of those research works.

B. Medhi and P. H. Talukdar [7] proposed an isolated Assamese speech recognition method that deployed the techniques of Artificial Neural Network (ANN) as a recognition model and Mel Frequency Cepstral Coefficients (MFCC), Zero Crossing Rate (ZCR), and Short Time Energy (STE) as features representing the Assamese speech. The system consisted of training, testing, and recognition phases, and used a database of 100 frequently used Assamese utterances whose syllable varies from 1 to 5 (monosyllabic to pentasyllabic). The utterances were spoken by twenty Assamese speakers with equal number of male and female (10 each), where each word was uttered by twenty times by each speaker. It was stated that the proposed system could achieve the speaker dependent recognition with 99% accuracy and speaker independent recognition with 93% accuracy.

M. D. Abdullah-al-MAMUN and F. Mahmud [21] presented the performance analysis of an isolated Bangla speech recognition system that used the Hidden Markov Model (HMM). That system proposed a recognition model for Bangla character set in which the MFCC and HMM were used as for feature extraction, and training and recognition respectively. A series of experiments were performed with ten talkers (5 males and 5 females) and 56 Bangla characters including the Bangla vowels, consonants, and digits with different conditions. That system was intended to use only for speaker dependent recognition and achieved 85.7% accuracy.

M. D. Ali Hossain et al. [22] also proposed a system of Backpropagation neural network model for isolated Bangla speech recognition in which the MFCC features of ten Bangla digits (0 to 9) recorded from ten speakers were used to train and test the network. It was stated that the system achieved the recognition rate of 96% for known speakers and 92% for unknown speakers.

Rady et al. [12] also proposed a speech recognition system based on Discrete Wavelet Transform (DWT) and ANN. The system used a data set of English digits (0 to 5) and other nine spoken words that were collected from four individuals in various time intervals. The feature vectors were formed by using the coefficients extracted by the DWT and then fed to the feed-forward Backpropagation neural network for classification. The proposed system achieved the speaker dependent recognition rate of 98.9%.

As per the knowledge gained from the previous works in the literature of speech recognition, it is found that the MFCC is the most widely used speech feature extraction

technique among others and ANN is also a widely used recognition method. Thus in this thesis, the MFCC and ANN techniques were deployed to implement a speaker independent recognition system for Myanmar isolated words.

1.3 Objectives of the thesis

This section lists the objectives of the proposed system in this thesis.

- To learn more about speech processing which is a branch of Digital Signal Processing
- To develop a speaker independent isolated words recognition system for our mother tongue (Myanmar)
- To study the MFCC which is a widely used speech feature extraction technique
- To study how to apply the ANN techniques to implement a speech recognition system
- To understand more about the supervised learning Backpropagation algorithm in real implementation

1.4 Organization of the thesis

This thesis is mainly composed of five chapters. Chapter 1 introduces the overview of the speech recognition systems and presents the objectives and organization of the thesis. The next chapter, Chapter 2 mainly discusses the theoretical background applied in this thesis. Then, Chapter 3 presents the implementations of the proposed system and Chapter 4 discusses the simulation results in detail. Finally, Chapter 5 concludes this thesis by highlighting the limitation and further extensions of the proposed system.

CHAPTER 2

BACKGROUND THEORY

This chapter focuses on the theoretical background of the proposed system in this thesis. It starts the discussion with introducing the fundamentals of digital signal processing. It is then followed by a review of the applications of the ASR systems, especially with some related works of Myanmar ASR. Finally, the chapter is concluded with the discussion of feature extraction and classification techniques, which are the main parts of the ASR systems.

2.1 Digital Signal Processing (DSP)

Digital Signal Processing (DSP) is concerned with the theoretical and practical aspects of representing information-bearing signals in a digital form and with using computers or special-purpose hardware and software to extract information, process it, or transform it in useful ways. It is characterized by the representation of discrete time or other discrete domain signals by a sequence of numbers and the processing of these signals [3].

Digital signal processing and analog signal processing are subfields of signal processing. DSP applications include audio and speech signal processing, sonar and radar signal processing, seismic data processing, biomedical signal processing, digital image processing, sensor array processing, spectral estimation, statistical signal processing, control systems, and signal processing for communication. DSP algorithms have long been run on standard computers as well as on specialized processors called digital signal processors and on purpose-built hardware like the application-specific integrated circuit (ASIC). Currently, there are additional technologies used for digital signal processing including more powerful general purpose microprocessors, field-programmable gate arrays (FPGAs), digital signal controllers (mostly for industrial applications such as motor control), and stream processors.

The goal of DSP is usually to measure, filter, and/or compress continuous real world analog signals. Usually, the first step is the conversion of an analog signal to a digital form by sampling and quantizing it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of discrete digital values. If the required

output signal is analog, it requires a digital-to-analog converter (DAC) [19]. Digital signal processing can involve linear or nonlinear operations. Nonlinear signal processing is closely related to nonlinear system identification and can be implemented in the time, frequency, and spatio-temporal domains.

DSP techniques have been well known in written language recognition in all its forms (on-line, off-line, printed, and handwritten). Essential and important methods for that kind of application include preprocessing techniques for noise removal, normalizing transformations for line width and slant removal, global transforms (e.g., Fourier transform, correlation), and various feature extraction methods. The local features include the computation of slopes, local densities, variable masks, etc., while others deal with various geometrical characteristics of letters (e.g., strokes, loops) [19]. There are additionally other important applicable areas of DSP techniques and some of them are discussed below.

2.1.1 DSP and its application

Application areas of digital signal processing have grown dramatically in importance in recent times, in parallel with the growth of powerful and low-cost processing circuits and reduction in price of computer memory. This has led, in turn, to many new applications including multimedia delivery and handheld communication devices with the convergence of computer and telecommunication technologies [19]. The followings list the most well-known applications of DSP, but not all.

- **Speech recognition** provides a more natural interface to computer systems and information retrieval systems (such as telephone voice response systems).
- **Audio enhancement and noise reduction** aims the improvement of audio quality, particularly in "acoustically difficult" environments such as vehicles. In cars and planes, for example, this is a desirable objective in order to improve passenger comfort and to enhance safety.
- **Digital music** in entertainment industry uses special effect and enhancements, for example, adding three-dimensional sound presence and simulating reverberation from the surroundings.

- **Image recognition** involves recognizing patterns in images, for example, recognizing faces for security systems, character recognition in scanned text, and handwriting recognition.
- **Image enhancement** is the improvement of the quality of digital images, for example, when degraded by noise on a communications channel or after suffering degradation over time on older recording media.
- **Communication and data transmission** heavily relies on signal processing. Error control, data synchronization, and maximization of the data throughput are prime examples.
- **Biomedical applications** like patient monitoring are indispensable in modern medical practice. Medical image processing and storage continues to attract much research attention.
- **Radar, sonar, and military applications** involve target detection, location of objects, and calculation of trajectories. Civilian applications of the Global Positioning System (GPS) are examples of the complex signal processing algorithms which have been optimized to operate on handheld devices.

Note that the above mentioned applications are all the subjects of ongoing research, and many unsolved problems remain. The following section especially discusses the speech recognition systems.

2.2 Automatic speech recognition systems

Automatic Speech Recognition (ASR) is also a branch of the DSP and an ongoing research field. It is the use of computer hardware and software-based techniques to identify and process the human voice. It is used to identify the words that a person has spoken or to authenticate the identity of the person speaking into the system.

Generally, there are three basic steps in a typical ASR system: preprocessing, feature extraction, and speech recognition. The voice from a speaker is recorded via a recording device such as a microphone and preprocessed to remove some undesirable features such as the background noise and room reverberation. Then, the features that identify the speaker or the speech are extracted from the preprocessed voice and based on them, the recognition model is built.

The ASR systems can be separated into different classes based on their generalization capability and what type of utterance they are able to recognize.

(1) Speaker dependent vs. independent system

Speaker dependence describes the degree to which a speech recognition system requires the knowledge of a speaker's individual voice characteristics to successfully process speech. A speech recognition engine can "learn" how a person speaks words and phrases and it can be trained to recognize a person's voice. Speech recognition systems that require a user to train the system to his/her voice are known as speaker dependent systems. If you are familiar with desktop dictation systems, most are speaker dependent. Because they operate on very large vocabularies, dictation systems perform much better when the speaker has spent the time to train the system to his/her voice [1].

Speech recognition systems that do not require a user to train the system are known as speaker independent systems. For example, consider the VoiceXML which is a digital document standard for specifying interactive media and voice dialogs between humans and computers. It is used for developing audio and voice response applications, such as automated customer service portals and banking systems. Speech recognition in the VoiceXML must be speaker independent. Think of how many users (hundreds, maybe thousands) may be calling into your web site. You cannot require that each caller train the system to his or her voice. The speech recognition system in a voice-enabled web application must successfully process the speech of many different callers without having to understand the individual voice characteristics of each caller [1].

(2) Isolated word vs. connected word recognition system

Isolated word recognizer requires single utterance at a time. It sets necessary condition that each utterance has little or no noise on both sides of sample window. Often, these types of speech have "Listen/Not-Listen states", where they require the speaker to have pause between utterances. Connected words require minimum pause between utterances to make speech flow smoothly. They are almost similar to isolated words [25].

(3) Continuous speech vs. spontaneous speech recognition system

Continuous speech is normal human speech, without silent pauses between words. It is basically computer's dictation. This kind of speech makes machine understanding

much more difficult. Spontaneous speech can be thought of as speech that is natural sounding and no tried out before. An ASR system with spontaneous speech ability should be able to handle a diversity of natural speech features such as words being run at the same time [20].

As previously discussed, automatic speech recognition is a challenging area of research in HMI technology and also a very desirable technology in many applications of our daily life, e.g. mobiles. There have been a lot of ASR researches carried out for different languages throughout the world and this thesis implements one for isolated Myanmar words. Thus, the following section introduces Myanmar language and presents some of the research works for Myanmar ASR systems.

2.2.1 Introduction to Myanmar language

Myanmar Language (formerly known as Burmese) is the official language of Myanmar. The Myanmar script was adapted from the Mon Script and descended from the Brahmi script of South India [33]. It is a syllabic script and one of the languages which have complex structures and unique. The Myanmar words are formed by collection of syllables and each syllable consists of up to seven different sub syllabic elements. The Myanmar script is written from left to right without any spaces between syllables or words. Nowadays, modern writing sometimes contains space after a sentence in order to enhance readability.

The Myanmar script has 34 consonants (known as “Byee”) as shown in Table 2.1. They serve as the base characters for Myanmar words [42]. Table 2.2 lists the Myanmar vowels, which are known as “Thara”. In Myanmar language, vowels are the basic building blocks of syllable formation. However, some syllable or words can be formed from consonants only. On the other hand, multiple vowel characters can exist in a single syllable like in other languages. In addition, Myanmar language has medial which are known as “ByeeTwe”. There are four basic medial and six combined medial in the Myanmar script [42]. Lastly, there are ten basic digits for counting the numbers in Myanmar language as shown in Table 2.3.

Table 2.1: Basic consonants

Basic Consonants				
က	ခ	ဂ	ဃ	င
စ	ဆ	ဇ	ဈ	ဉ/ည
ဋ	ဌ	ဍ	ဎ	ဏ
တ	ထ	ဒ	ဓ	န
ပ	ဖ	ဗ	ဘ	မ
ယ	ရ	လ	ဝ	သ
	ဟ	ဠ	အ	

Table 2.2: Vowels

Vowels				
◌	◌ါ	◌ါ်	◌ိ	◌ီ
◌့	◌ာ			

Table 2.3: Numerals

Numerals				
၀	၁	၂	၃	၄
၅	၆	၇	၈	၉

2.2.1.1 Related works for Myanmar ASR

Like other languages in the world, there has been some research works proposed for Myanmar speech recognition in the literature.

In 2003, Zaw Min Tun [43] proposed an ASR system for Myanmar language. In that system, MFCC was used as the front-end processing and then HMMs were constructed by Gaussian distribution function based on the MFCCs. That system did not discuss anything about the recognition result as its main focus was to introduce how to develop an ASR system for Myanmar language.

In 2009, Aung Tun Tun Lwin [5] also proposed an ANN-based isolated Myanmar digit recognition system. That system discussed how to use the Linear Predictive Coding (LPC) for feature extraction and the feed forward multilayer perceptrons trained by the Backpropagation for digit recognition. That system achieved an accuracy of more than 98% for speaker dependent isolated digit recognition.

In 2015, Ei Mon Kyaw [13] proposed a speaker dependent Myanmar speech command recognition system by using the MFCC and Dynamic Time Warping (DTW) techniques. That system was intended to recognize 10 commands from real-time microphone input. The accuracy was 90% in lower noisy speech condition, 80% in medium noisy speech condition, and 40% in high noisy speech condition.

In 2015, Su Myat Mon and Hla Myo Tun [32] also proposed a speech-to-text conversion system by using the MFCC and HMM. That system used the MFCC feature vectors extracted from the original speeches as the observation sequences of the HMM recognizer. That system only emphasized speaker dependent recognition and achieved the recognition rate of 87.6%.

Although there had been a lot of research done for Myanmar ASR, it is still far from a mature field. Therefore, this thesis focuses to develop a speaker independent recognition system for isolated Myanmar words. The following section discusses the two major steps of an ASR system: feature extraction and recognition models.

2.3 Feature extraction

Feature extraction is related to dimensionality reduction. Its main purpose is to reduce the resources required to describe a large dataset. When dealing with a large dataset with complex data, the number of variables involved is a major concern. Analysis of those complex variables generally requires a large amount of memory and computational power [36]. Thus, when an algorithm needs to process a large dataset which is suspected to be redundant (e.g. the same measurement in both feet and meters or the repetitiveness of images presented as pixels), the data should be first transformed into a more compact set of features called a feature vector. This process is called feature extraction. The extracted features are expected to contain the relevant information from the input data so

that the desired task can be performed by using this reduced representation instead of the complete initial data.

In machine learning, pattern recognition, and image processing, feature extraction starts from an initial set of the measured data and builds the derived values (features) intended to be informative, non-redundant, facilitating the subsequent learning and generalization steps; leading to better human interpretations in some cases.

For DSP systems such as speech recognition, feature extraction means identifying the components of a sound signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion, among others [18]. No need to doubt, a good feature may produce a good recognition result. As far as we know from the fundamental formation of speaker identification and verification systems, the number of training and test vectors needed for the classification problem grows with the dimension of the given input. Thus, a good feature extraction process for dimension reduction is inevitable in ASR systems. Some of the well-known speech feature extraction techniques are discussed below.

2.3.1 Principal Component Analysis

Principal component analysis (PCA) is a robust feature extraction method that could be used for feature extraction from speech signal [30]. It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. Often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data.

If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space, PCA can supply the user with a lower-dimensional picture, a projection of this object when viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced. The PCA more minimizes the data samples size as the correlation level goes deeper. In some cases where the data samples are originally uncorrelated, the number of PCA components will be the same as the original signal samples. Some common applications of PCA are quantitative finance, neuroscience, and data compression [30].

In 2007, Takiguchi and Ariki proposed the PCA-based speech enhancement for distorted speech recognition. That system described a new PCA-based speech enhancement algorithm using kernel PCA instead of DCT, where the main speech element was projected onto low-order features and the distortion element or noise was projected onto high-order features. Its effectiveness was confirmed by the word recognition experiments on distorted speech [34].

2.3.2 Linear Predictive Coding

Linear predictive coding (LPC) is a tool mostly used in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form, using the information of a linear predictive model. Linear prediction is a mathematical operation where future values of a discrete-time signal are estimated as a linear function of previous samples [30].

LPC is one of the most powerful speech analysis techniques, which can determine the basic parameter and computational model of speech. It is also a useful method for encoding quality speech at a low bit rate and provides extremely accurate estimates of speech parameters. The basic idea behind LPC is that a current speech sample can be estimated from a linear combination of past speech samples. LPC is a model based on human speech production. It utilizes a conventional source-filter model in which the glottal, vocal tract, and lip radiation transfer functions are integrated into one all-pole filter that simulates acoustics of the vocal tract. The principle behind the use of LPC is to minimize the sum of squared differences between the original speech and the estimated speech over a finite duration [39]. LPC analysis also involves the decision-making process of voiced or unvoiced.

LPC is generally used for speech analysis and resynthesis. It is used as a form of voice compression by phone companies, for example in the GSM standard. It is also used for secure wireless, where voice must be digitized, encrypted, and sent over a narrow voice channel.

In 2011, Thiang and Wijoyo proposed an implementation of the speech recognition system on a mobile robot for controlling movement of the robot. In that system, voice signals from the microphone were directly sampled and processed by the

LPC method to extract the voice features. Then, ANN was used as the recognition method. Experimental results showed that the highest recognition rate achieved by that system was 91.4% [37].

2.3.3 Perceptual Linear Prediction

Perceptual linear prediction (PLP) technique is identical to LPC except that its spectral characteristics have been transformed to match the characteristics of the human auditory system. PLP removes the unwanted information of the speech and thus improves speech recognition rate. The PLP analysis technique was originally designed to suppress speaker dependent components in features used for automatic speech recognition, but later experiments demonstrated the efficiency of their use for speaker recognition tasks [24].

The goal of the original PLP model is to describe the psychophysics of human hearing more accurately in the feature extraction process. Unlike pure linear predictive analysis, the short-term spectrum of the speech is modified by PLP by means of several psychophysically based transformations [30]. PLP coefficients are more often used than LPC coefficients because they approximate well the high-energy regions of the speech spectrum while simultaneously smoothing out the fine harmonic structure, which is often characteristic of the individual but not of the underlying linguistic unit. LPC, however, approximates the speech spectrum equally well at all frequencies, and this representation is contrary to known principles of human hearing. PLP incorporates critical-band spectral-resolution into its spectrum estimate by remapping the frequency axis to the Bark scale and integrating the energy in the critical bands to produce a critical-band spectrum approximation.

PLP, the idea of a perceptual front end for determining liner predictive cepstral coefficients, has been applied in different ways to improve speech detection and coding, as well as noise reduction, reverberation suppression, and echo cancellation.

In 2011, Kurian and Balakrishnan proposed an optimum speaker independent isolated digit recognizer for Malayalam language. That system employed the PLP cepstral coefficients for speech parameterization and HMM, the powerful and well accepted pattern recognition technique, for acoustic modeling. The system achieved an accuracy of 99.5% with the unseen data [8].

2.3.4 Discrete Wavelet Transform

Fast Fourier Transform (FFT) and discrete wavelet transform (DWT) are both linear operations that can be viewed as a rotation in function space to a different domain. For the FFT, this new domain contains basis functions that are sines and cosines. For the DWT, this new domain contains more complicated basis functions called wavelets. The basic functions in both transforms are localized in frequency, making mathematical tools such as power spectra (how much power is contained in a frequency interval) useful at picking out frequencies and calculating power distributions.

The most significant dissimilarity between FFT and DWT is that FFT only gives the frequency data of a signal; it doesn't provide the data about at what time which frequency is present. Thus, it is not suitable for analysis of non-stationary signals like speech. To solve this, the windowed short-time Fourier transform (STFT) provides temporal data concerning the frequency content of a signal. However, its disadvantage is fastened time resolution attributable to only fixed window length.

The wavelet transform, with its versatile time-frequency window, is an efficient tool for analysis of non-stationary signals [30]. Just as the Fourier transform decomposes a signal into a family of complex sinusoids, the wavelet transform decomposes a signal into a family of wavelets. Unlike sinusoids which are symmetric, smooth, and regular, wavelets can be symmetric or asymmetric, sharp or smooth, and regular or irregular. The wavelet transform computes the inner products of a signal with a family of wavelets. In contrast with sinusoids, wavelets are localized in both time and frequency domains, so wavelet signal processing is suitable for nonstationary signals, whose spectral contents change over time. The adaptive time-frequency resolution of wavelet signal processing enables you to perform multiresolution analysis on nonstationary signals.

The wavelet transform has been widely utilized in diversified applications of image/speech/video compression and digital communications. In addition, the properties of wavelets and flexibility to select wavelets make wavelet signal processing a beneficial tool for feature extraction applications.

Wavelet transform have been used for speech feature extraction [30] in which

- The energies of wavelet decomposed sub-bands have been used in place of Mel filtered sub-band energies because of its better energy compaction property.
- Wavelet transform-based features give better recognition accuracy than the LPC and MFCC.
- The wavelet transform has a better capability to model the details of unvoiced sound portions.
- Wavelet, that is using fast wavelet transform, is computationally very fast.
- Wavelets have the great advantage of being able to separate the fine details in a signal. Very small wavelets can be used to isolate very fine details in a signal, while very large wavelets can identify coarse details.

2.3.5 Mel Frequency Cepstral Coefficients

Up until the 1980s, the LPC and real cepstrum coefficients were the major parameters used to represent the utterances for speech recognizers. In 1980, Mel Frequency Cepstral Coefficients (MFCCs) were first used together with the DTW algorithm for a speech recognition system by Davis and Mermelstein [16]. Their study revealed the fact that the MFCCs outperform any other parametric representation such as LPC. Since then, the MFCCs have become the most popular features up to date.

The basic idea behind using the MFCCs is to obtain a feature representation which approximates the human perception [16]. To achieve this, the frequency bands in the Mel frequency cepstrum are linearly spaced at low frequency below 1 kHz and logarithmically spaced above 1 kHz. This spacing is done based on the Mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum [24]. This frequency warping also allows for better representation of sound, for example, in audio compression. Moreover, subjective pitch presents on Mel frequency scale to capture the important characteristic of phonetic in speech. Thus, MFCCs are widely used features in ASR systems these days.

The overall process of the MFCC is shown in Figure 2.1 and it consists of seven computational steps. Each step has its own mathematical approaches as discussed below.

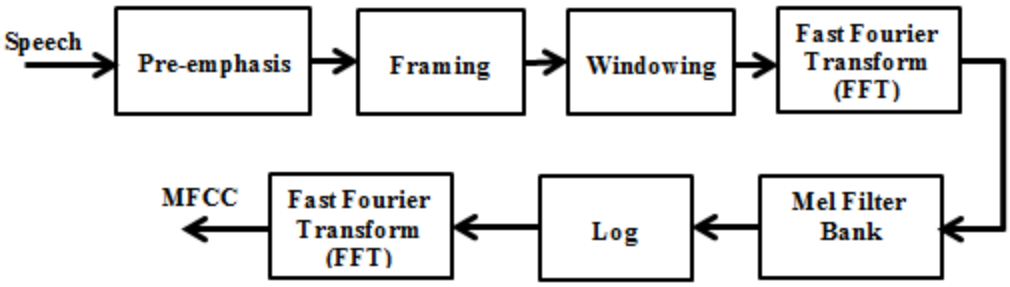


Figure 2.1: Flow diagram of the MFCC feature extraction

2.3.5.1 Pre-emphasis

Pre-emphasis step is carried out to increase the signal energy in high frequencies. This process is important because the high-frequency part was suppressed during the sound production mechanism of humans and yielded the unbalanced frequency spectrum. It can be evident on the spectrum for voiced segments like vowels, where there is more energy at the lower frequencies than at the higher frequencies. Boosting the high frequency energy makes information from these higher formats more available to the acoustic model and improves phone detection accuracy. To do pre-emphasis,

The speech signal $s(n)$ is sent to a high-pass filter.

$$s'(n) = s(n) - a * s(n-1), \tag{2.1}$$

where $s'(n)$ is the output signal and the value of a is usually between 0.9 and 1.0. The z-transform of the filter is

$$H(z) = 1 - a * z^{-1}. \tag{2.2}$$

2.3.5.2 Framing and windowing

Feature extraction process helps us build phone or sub phone classifiers by providing spectral features. It cannot be done on an entire utterance because the spectrum changes very quickly, i.e. speech is nonstationary and its statistical properties are not constant over time. Thus, the second step of MFCC is to split the speech signal into several frames such that each frame can be examined in short time. Although the speech is non-stationary, it is assumed that it is stationary for a short period of time. The width of a frame is generally 20~30 ms with an optional overlap of 1/3 ~1/2 of the frame size. If a

frame is too short, its few samples cannot get a reliable spectral estimate. If the frame is too long, the signal may vary widely throughout the frame.

In order to reduce the discontinuities at the start and end of a frame or to smooth the first and last points in a frame, windowing function is then used. A window that is non-zero inside some region and zero elsewhere is run across the speech signal and extract the waveform inside this window.

$$y[n] = w[n]*s[n], \tag{2.3}$$

where $w[n]$ is the window function, $s[n]$ and $y[n]$ are the input and output signals, respectively.

A windowing process can be characterized by three parameters: width of the window (in milliseconds), shape of the window, and offset between successive windows. Figure 2.2 suggests that the window shapes are rectangular as the extracted windowed signal is similar to the original one. Rectangular window is indeed the simplest window; however it can cause problems as it cuts off the signal abruptly at its boundaries. These discontinuities create problems when performing Fourier analysis. Thus, the MFCC chooses the Hamming window over rectangular window defined in eq. 2.4 that shrinks the values of the signal toward zero at window boundaries, thus avoiding discontinuities. Figure 2.3 compares the rectangular and the Hamming windows.

$$W(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L}\right), & 0 \le n \le L-1, \\ 0, & \text{otherwise}, \end{cases} \tag{2.4}$$

where L is the window length.

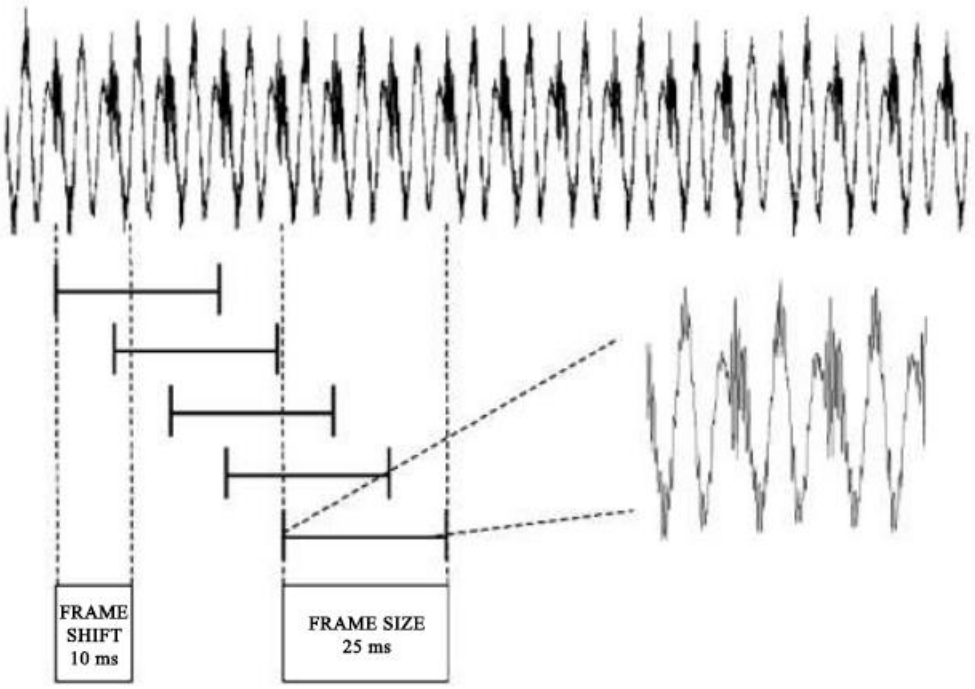


Figure 2.2: The rectangular windowing process

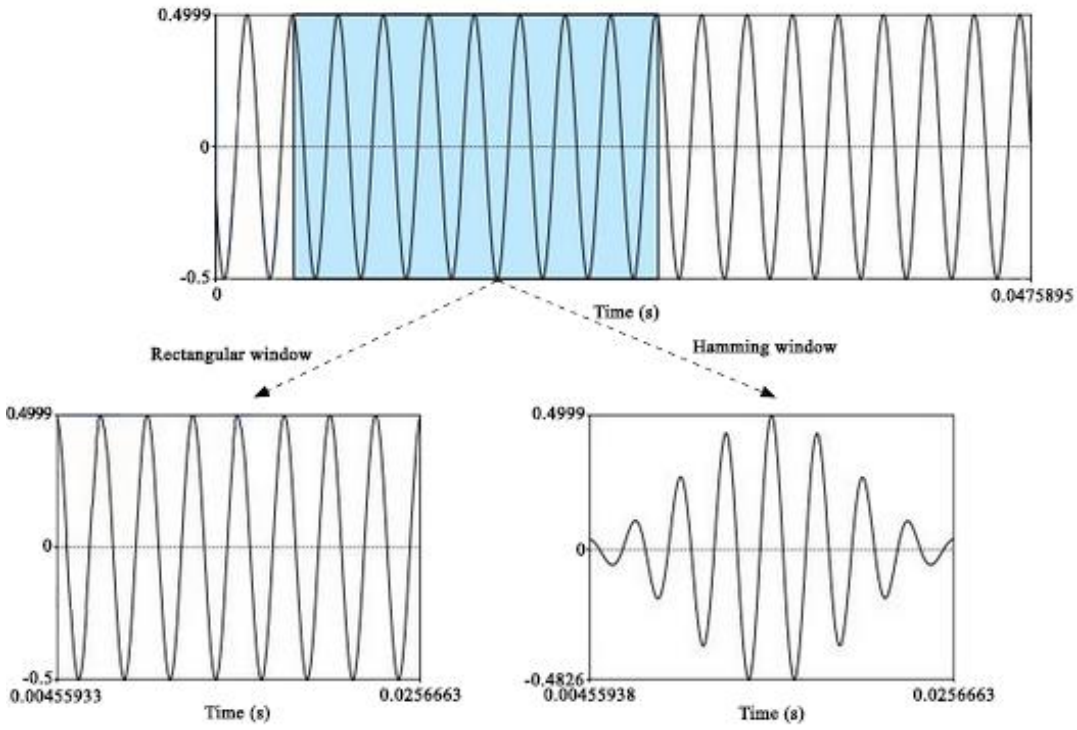


Figure 2.3: Comparison of the rectangular and the Hamming windows

2.3.5.3 Fast Fourier Transform (FFT)

The next step is to extract the spectral information from the windowed signal. It is needed to know how much energy the signal has at different frequency bands. Spectral analysis shows that different timbres in the speech correspond to different energy distribution over frequencies. Usually in the MFCC calculation, the signal within a frame is assumed as periodic and continuous when wrapping around, and the FFT is used to convert the convolution of the glottal pulse and the impulse response in the time domain. In other words, the FFT is performed to obtain the magnitude frequency response of the signal. The analysis and synthesis equations of FFT for a signal with length N are given by eq. 2.5 and 2.6 respectively.

$$X(k) = \sum_{n=1}^{N} x(n) W_N^{(n-1)(k-1)}, \tag{2.5}$$

$$x(n) = \left(\frac{1}{N}\right) \sum_{k=1}^{N} X(k) W_N^{-(n-1)(k-1)}, \tag{2.6}$$

where $x[n]$ is a time-domain signal, $X(k)$ is the frequency response of $x[n]$, and $W_N = e^{(-2\pi i)/N}$ is the N^{th} root of unity.

2.3.5.4 Mel filterbank

FFT yields the frequency response of a time-domain speech signal. However, the spectral envelope, a smooth curve outlining the extremes, is more important than the frequency response itself. Moreover, human hearing is differently sensitive at different frequency bands. It is more sensitive at lower frequencies, roughly below 1 kHz. Modeling this perception of the HAS during feature extraction turns out to improve the performance of speech recognition. During MFCC computation, Mel filterbank which is a triangular filterbank that collect energy from each frequency band is used to capture the spectral envelope. A Mel is a unit of pitch. By definition, pairs of sounds that are perceptually equidistant in pitch are separated by an equal number of Mels.

Mel filterbank is important due to the following reasons:

- It applies the Mel frequency scaling, which is a perceptual scale that helps to simulate the way human ear works. It corresponds to better resolution at low frequencies and less at high.
- Using the triangular filterbank helps to capture the energy at each critical band and gives a rough approximation of the spectrum shape, as well as smoothes the harmonic structure.

Figure 2.4 shows a bank of triangular filters used to compute a weighted sum of the filtered spectral components so that the output approximates a Mel scale. Each filter's magnitude frequency response is triangular in shape and equal to unity at the centre frequency and decrease linearly to zero at centre frequency of two adjacent filters. Then, the filter's output is the sum of its filtered spectral components. To compute the Mel for a given frequency *f* in Hz, eq. 2.7 is used.

$$F(mel) = 2595 * \log_{10}\left[1 + \frac{f}{700}\right]. \tag{2.7}$$

The mapping between the frequency in Hz and Mel scale is linear below 1000 Hz and logarithmic above 1000 Hz.

2.3.5.5 Logarithmic transformation

After the Mel-scale conversion, logarithmic transformation is applied to the absolute magnitude of the coefficients obtained. This operation discards the phase information, making feature extraction less sensitive to speaker dependent variations. Logarithmic is used because:

- Logarithmic scale can compress the dynamic range of values;
- Human ear response to signal level is logarithmic;
- Logarithmic scale can make the frequency estimates less sensitive to slight variations in input (power variation due to speaker's mouth moving closer to microphone);

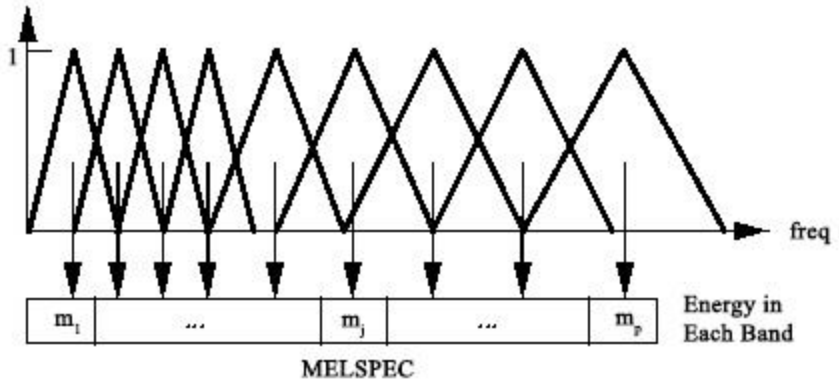


Figure 2.4: Mel filterbank

2.3.5.6 Discrete Cosine Transform (DCT)

While it would be possible to use the Mel spectrum by itself as a feature representation, the spectrum also has some correlation problems. For this reason, the next step in the MFCC feature extraction is the computation of cepstrum.

Cepstral coefficients have the extremely useful property that the variance of the different coefficients tends to be uncorrelated. This is not true for the spectrum, where spectral coefficients at different frequency bands are correlated. In this step, apply DCT on the 20 log energy obtained from the triangular bandpass filters to convert the log Mel spectrum into time domain. The result is a set of Mel frequency cepstral coefficients that is called acoustic vectors. MFCC alone can be used as features for speech recognition. However for better performance, other features such as pitch, zero cross rate, etc can also be taken into account.

2.4 Classification techniques

In addition to feature extraction, recognition models also play an important role in ASR systems. To classify the feature vectors into their relevant classes, the techniques like Artificial Intelligence, Cross-Correlation, Vector Quantization, Dynamic Time Warping, Hidden Markov Model, Artificial Neural Network, etc can be used. This section briefly discusses some of the widely used classification techniques.

2.4.1 Artificial Intelligence

Artificial intelligence (AI) approach attempts to mechanize the recognition procedure according to the way a person applies its intelligence in visualizing, analyzing, and finally making a decision on the measured acoustic features. AI approach is a hybrid of the acoustic phonetic approach and pattern recognition approach by exploiting the ideas and concepts of acoustic, phonetic, and pattern recognition methods. Expert system is widely used in this approach. Knowledge based AI approaches use the information regarding linguistic, phonetic, and spectrogram. Some speech researchers developed the recognition systems that used acoustic phonetic knowledge to develop classification rules for each speech sounds. Template based AI approaches have been very effective in the design of a variety of speech recognition systems. However, they provided little insight about human speech processing, thereby making error analysis and knowledge-based system enhancement difficult [38].

2.4.2 Cross-correlation

Even for the same speaker, different words have different frequency bands which are due to different vibrations of the vocal cord. And the shapes of spectrums are also different. Based on those facts, speech recognition systems can also be realized in which the spectrums of the query recorded signal and previously recorded reference signals have to be compared. By checking which of the reference signals better match the query signal, the system will give the judgment that which reference word is again recorded at the query time [38].

2.4.3 Dynamic Time Warping

Speech recognition can be complicated for a number of different reasons. When comparing a query sample against training data, the query may be of a different duration than the training sample. One way to solve this problem is to normalize the speech samples so that they all have the same duration. Another problem is that the rate at which the words are spoken may not always be constant, that would mean the optimal alignment between a test sample and the training sample may be nonlinear. An instance of dynamic

programming known as dynamic time warping (DTW) is an efficient method to solve the time alignment problem in speech recognition.

DTW is mainly used for measuring similarity between two time series which may vary time or speed. For instance, similarities in walking patterns could be detected by using DTW even if one person was walking faster than the other or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data – indeed, any data which can be turned into a linear sequence can be analyzed with DTW. It is also used to find the optimal alignment between the two time series if one time series may be "warped" non-linearly by stretching it along its time axis. This warping between two time series can then be used to find the corresponding regions or to determine the similarity between the two time series. A well-known application of DTW has been automatic speech recognition to cope with different speaking speeds. Other applications are speaker recognition and online signature recognition. Also it is seen that DTW can be used in partial shape matching application [2].

2.4.4 Hidden Markov Model

One of the most widely used and successful approaches to speech recognition is using a Hidden Markov Model (HMM). It is a mathematical model derived from Markov Model and essentially a collection of different states connected by transitions. The model begins in a designated initial state and at each step in the process, a transition is used to reach a new state where an output is generated. This system is referred to as hidden due to the fact that the sequence of states visited is hidden from the user while outputs are observed over the course of running the system.

A hidden Markov model can be considered as a generalization of a mixture model where the hidden variables which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. The challenge in HMM is to determine the hidden parameters/states from the observable data. In HMM, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Thus, the

sequence of tokens generated by an HMM gives some information about the sequence of states.

HMM is widely used in speech and speaker recognition systems. It creates stochastic models from known utterances and compares the probability that unknown utterance was generated by each model. It considers a speech signal as quasi-static for short durations and models these quasi-static frames for recognition. It breaks the feature vector of the signal into a number of states and finds the probability of a signal to transit from one state to another. This uses theory from statistics in order to (sort of) arrange the speech feature vectors into a Markov matrix (chains) that stores probabilities of state transitions. That is, if each of our code words were to represent some state, the HMM would follow the sequence of state changes and build a model that includes the probabilities of each state progressing to another state. The parameters of the model are the state transition probabilities, means, variances, and mixture weights that characterize the state output distributions [28].

2.4.5 Vector Quantization

Vector Quantization (VQ) is a classical quantization technique from signal processing that allows the modeling of probability density functions by distribution of prototype vectors. It works by dividing a large set of points (vectors) into groups having approximately the same number of point's closet to them. Each group is represented by its centroid point [15].

VQ is used for lossy data compression, lossy data correction, pattern recognition, density estimation, and clustering. It was also used in the 80s for speech and speaker recognition. In those applications, one codebook is constructed for each class using acoustic vectors of the speaker. In the testing phase, the quantization distortion of a testing signal is worked out with the whole set of codebooks obtained in the training phase. The codebook that provides the smallest VQ distortion indicates the identified speaker. Similarly for isolated word recognition, each vocabulary word gets its own VQ codebook, based on the training sequences of several repetitions of the word. The test word is evaluated by all codebooks and the word whose codebook yields the lowest distance measure is chosen.

The utility of VQ in speech/speaker recognition lies in the efficiency of using compact codebooks for reference models and codebook searcher in place of more costly evaluation methods. The main advantage of VQ in recognition applications is its low computational burden when compared with other techniques like DTW and HMM. The main drawback is that it does not take into account the temporal evolution of the signals such as speech because all vectors are mixed up [14].

2.4.6 Artificial Neural Networks

Speech recognition can be considered as a pattern recognition problem. Artificial neural networks (ANNs) perform well when attempting pattern recognition due to their pattern learning and distinguishing ability, parallel distributed memories, and error stability. Many researchers naturally applied the ANNs to speech recognition and the first attempts were nothing more than simple problems like classifying speech segments as voiced/unvoiced or nasal/fricative/plosive. When researchers succeeded in these experiments, they move on to phoneme classification. This section introduces the basic structure of neural networks and discusses Backpropagation learning, which is a kind of ANN, in detail.

ANNs are parallel computing devices consisting of many interconnected simple processors. These processors are quite simplistic, especially when compared with the types of processors found in a computer. Each processor in a network is only aware of the signals it periodically receives and the signals it periodically sends to other processors, and yet such simple local processors are capable of performing complex tasks when placed together in a large network of orchestrated cooperation.

ANNs were inspired by biological science which is a study of how the neuro-anatomical of the living have developed in solving problems. Researchers attempting to explain the human behavior and thinking process by modeling the human brain expounded the first ANN theories [9]. ANNs are also called:

- Parallel distributed processing models,
- Connectionism models,
- Adaptive systems,
- Self-organizing systems,

- Neurocomputing, or
- Neuromorphic systems.

Artificial neural networks have their roots in works performed in the early part of the twentieth century. However, only during the 1990s, after the breaking of some theoretical barriers and growth in available computing power, these networks have been widely accepted as useful tools. The word ‘artificial’ is sometimes used to make it clear that discussion is about an artificial device and not about the real biological neural networks found in humans. It is the human brain that has inspired the creation of artificial neural networks and no doubt will influence the further development. However, in comparison to the human brain, artificial neural networks are at present highly simplistic abstractions. It is common to drop the prefix ‘artificial’ when it is clear in which context these networks are being discussed.

Although neural networks can be implemented as fast hardware devices, much research is performed using a conventional computer running software simulations. Software simulation provides a somewhat cheap and flexible environment to develop research ideas and can also be effectively used in many real-world applications. For example, a neural network software package might be used to develop a system for credit scoring of individual who is applying for a bank loan. Though a neural network solution might have the look and feel of any conventional piece of software, there is a key difference in that most neural solutions are ‘learnt’ and not programmed: the network learns to perform a task rather than being directly programmed [9].

The application based for neural networks is enormous:

- Credit card fraud detection,
- Stock market forecasting,
- Credit scoring,
- Optical character recognition,
- Human health monitoring,
- Diagnosis,
- Machine health monitoring,
- Road vehicle autopilots,
- Learning to land damaged aircraft, etc.

2.4.6.1 Basics of ANNs

A neural network is a collection of units that are connected in some patterns to allow communication between the units. These units, also referred to as neurons or nodes, are simple processors whose computing ability is typically restricted to a rule for combining input signals and an activation rule that takes the combined input to calculate an output signal. Output signals may be sent to other units along connections known as weights. The weights usually excite or inhibit the signal that is being communicated. A neural network unit is illustrated in Figure 2.5.

There are many different types of neural networks, but a number of features that are common to all those are:

- A set of simple processing units, called neurons, and
- A pattern of connectivity.

Neurons are usually organized into layers, as shown in Figure 2.6. The input layer is not composed of full neurons, but rather consists simply of the values in a data record that constitutes inputs to the next layer of neurons. The input neurons exist to receive signals from the environment. The next layer is called a hidden layer; there may be several hidden layers. The final layer is the output layer, where there is one unit for each class. The output neurons exist to communicate back to the environment the result of computation.

Pattern of connectivity refers to a network's wiring detail, i.e. the detail of which units connect their direction of connections, and the values of their weighted connections. The task that a network knows is coded in the weights that connect units. In one network model, each unit may connect to every other unit; in another network model, units may be arranged into an ordered hierarchy of layers where connections are only allowed between units in immediately adjacent layers; other network models allow feedback connections between adjacent layers, or within a layer, or for units to send signals back to themselves. Based on the connectivity, neural networks can be mainly classified as feedforward or Backpropagation. In feedforward neural network model, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes, as shown in Figure 2.6. In Backpropagation neural network

model, the error information is calculated at the output layer and distributed back through the hidden and input layers, as shown in Figure 2.7.

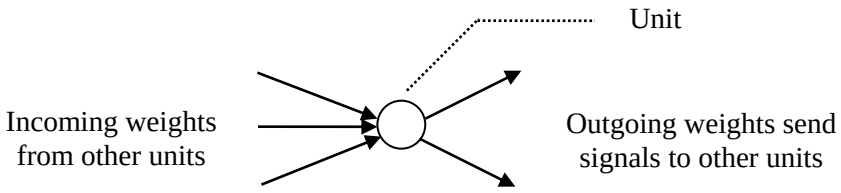


Figure 2.5: A single network unit

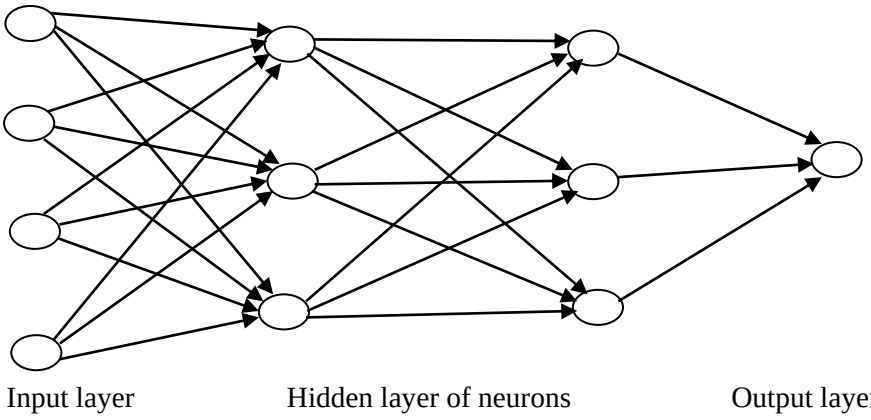


Figure 2.6: A neural network organized into different layers

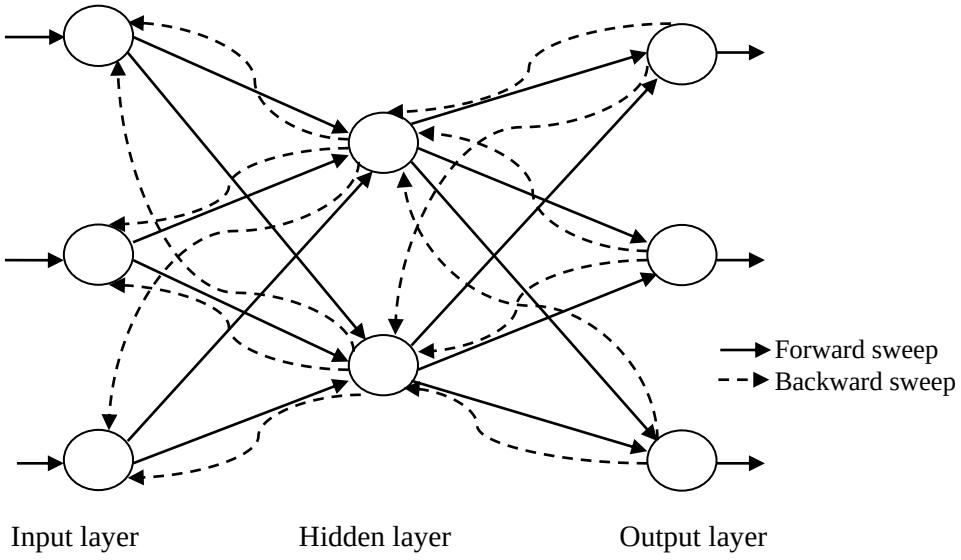


Figure 2.7: Backpropagation in a sample neural network

In addition to the above facts, other features that are common for all neural networks are:

- A rule for propagating signals through the network,
- A rule for combining input signals,
- A rule for calculating an output signal, and
- A learning rule to adapt the weights.

For a particular neural network model, some rules will exist to control when neurons can be updated, i.e. combining input signals and calculating an output signal, and when a signal can be sent on to other neurons. To get an insight of those rules, consider a nonlinear neuron, shown in Figure 2.8, which forms the basis for designing ANNs. Three basic elements of the neuron model are defined as follows.

1. A set of synapses or connecting links, each of which is characterized by a weight or strength of its own; specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. It is important to make a note of the manner in which the subscripts of the synaptic weight w_{kj} are written. The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. Unlike a synapse in the brain, the synaptic weight of an artificial neuron may lie in a range that includes negative as well as positive values.
2. An adder for summing the input signals, weighted by the respective synapses of the neuron; the operations described here constitutes a linear combiner.
3. An activation function for limiting the amplitude of the output of a neuron; The activation function is also referred to as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0, 1] or alternatively [-1, 1].

In mathematical terms, we may describe a neuron k by writing the following pair of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.8}$$

and

$$y_k = \varphi(u_k + b_k), \tag{2.9}$$

where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the synaptic weights of neuron k; u_k is the linear combiner output due to the input signals; b_k is the bias; $\varphi(.)$ is the activation function; and y_k is the output signal of the neuron. The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively.

The use of bias b_k has the effect of applying an affine transformation to the output u_k of the linear combiner in the model of Figure 2.8, as shown by

$$v_k = u_k + b_k. \tag{2.10}$$

In particular, depending on whether the bias b_k is positive or negative, the relationship between the induced local field or activation potential v_k of neuron k and the linear combiner output u_k is modified in the manner illustrated in Figure 2.9; Note that as a result of this affine transformation, the graph of v_k versus u_k no longer passes through the origin. A combination of eq. 2.8 to 2.10 may be reformulated as defined in eq. 2.11 and 2.12.

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{2.11}$$

and

$$y_k = \varphi(v_k). \tag{2.12}$$

In eq. (2.11), we have added a new synapse. Its input is

$$x_0 = +1 \tag{2.13}$$

and its weight is

$$w_{k0} = b_k. \tag{2.14}$$

Figure 2.10 shows the model of a nonlinear neuron, which is different in appearance but mathematically equivalent model of Figure 2.8. In that figure, the effect of the bias is accounted for by doing two things: (1) adding a new input signal fixed at +1, and (2) adding a new synaptic weight equal to the bias b_k.

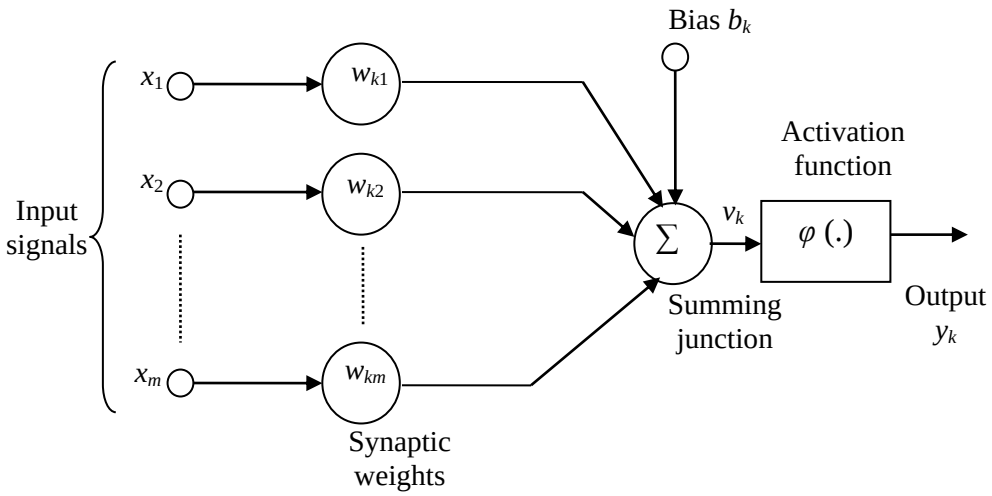


Figure 2.8: Nonlinear model of a neuron

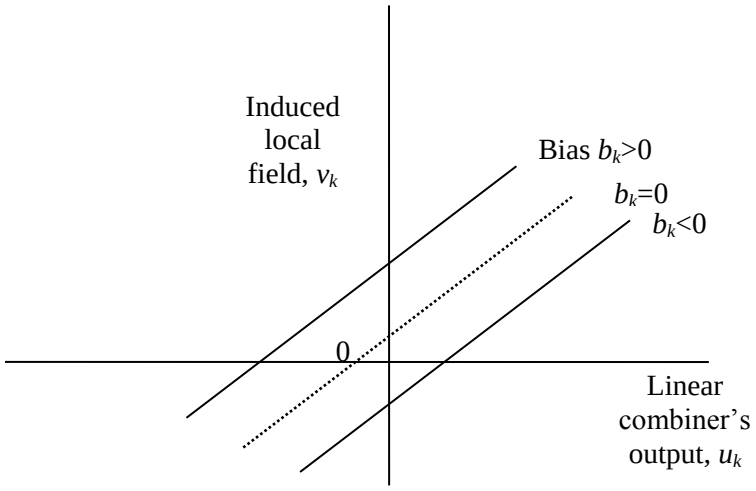


Figure 2.9: Affine transformation produced by the presence of a bias; note that $v_k = b_k$ at $u_k = 0$

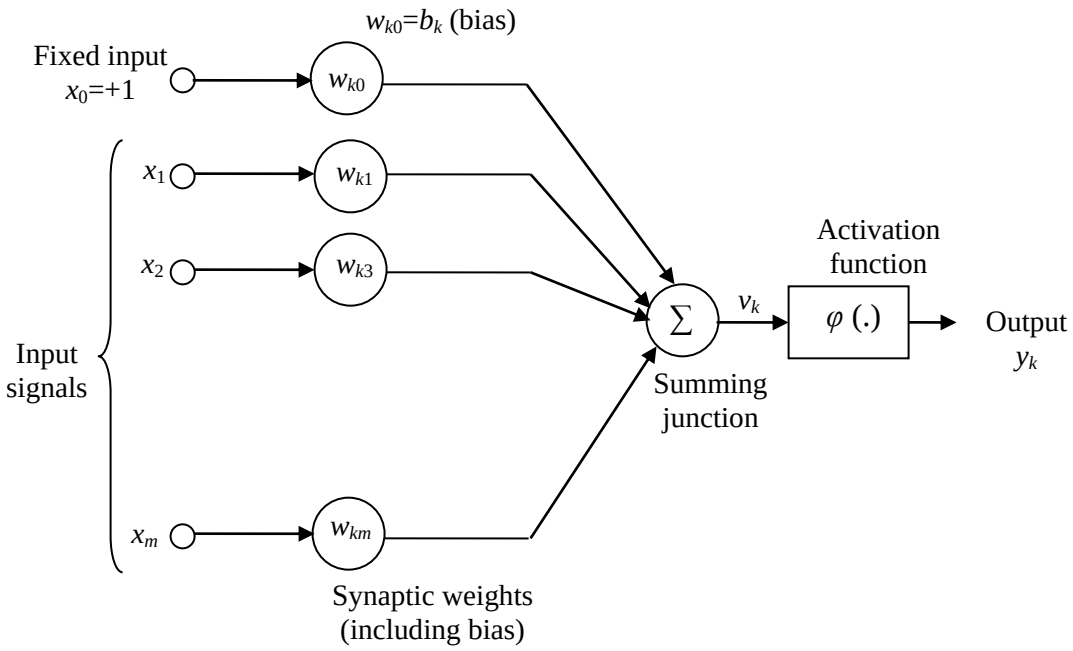


Figure 2.10: Another nonlinear model of a neuron

2.4.6.2 Transfer functions

A transfer function is a rule for calculating an output value that will be transmitted to other units or for presenting to the environment (if an output unit) the end result of computation. This rule is also known as an activation function and the output value is referred to as the activation for the unit. The activation may be a real number, a real number that is restricted to some interval such as [0, 1], or a discrete number such as {0, 1} or {+1, -1}. The value passed to the activation function is the liner combiner output, e.g. as defined in eq. 2.11. There are different kinds of transfer functions that can be chosen as per system requirement. Some of them are introduced below.

(1) Identity function

Identity function is an activation function for input units. As shown in Figure 2.11, it simply means that its activation is its net input, i.e. the linear combiner output. Input units really serve to distribute the input signals from the environment to other network units, and so we want the signal coming out of the unit to be the same as that going in. Unlike other network units, input units have only one input value, thus its net input is simply its input value.

(2) Binary threshold function

Most network models rely on nonlinear activation functions and one of them is binary threshold function. It will limit the activation to 1 or 0 depending on the net input relative to some threshold θ, as shown in Figure 2.12. Usually it is more convenient to subtract the threshold (known as a bias) from the net input and change the threshold to its mathematical equivalent form, shown in Figure 2.13.

(3) Sigmoid functions

One of the widely used nonlinear activation functions is sigmoid, S shaped functions. The output from a sigmoid function falls in a continuous range from 0 to 1. The sigmoid functions are extensively used in Backpropagation neural networks because it reduces the burden of complication involved during training phase. Logistic and hyperbolic tangent functions are commonly used sigmoid functions.

As an example, the logistic function is shown in Figure 2.14.

$$f(net) = \frac{1}{1+\exp(-net)}, \tag{2.15}$$

where *net* is the resultant combined input to unit.

The slope and output range of the logistic function may vary. The bipolar sigmoid, for example, has an output ranging from -1 to 1.

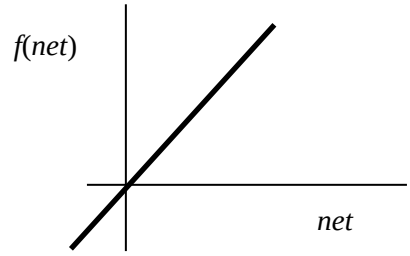
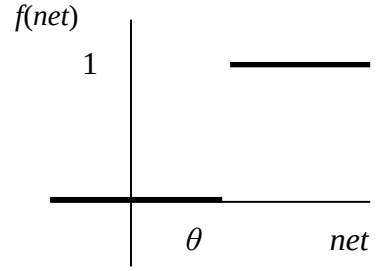


Figure 2.11: The activation is the same as the net input. Note that *f*(*net*) refers to the activation.



$$f(net) = \begin{cases} 1 & \text{if } net \geq \theta \\ 0 & \text{if } net < \theta \end{cases}$$

Figure 2.12: Binary threshold function

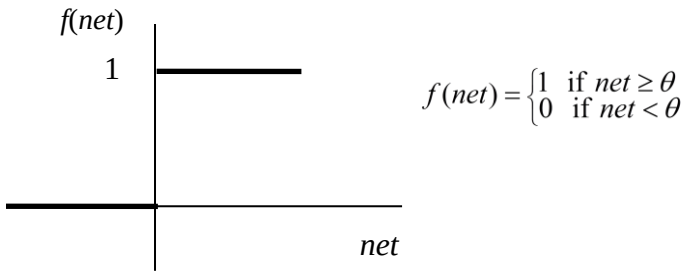


Figure 2.13: Binary threshold function with bias term added in

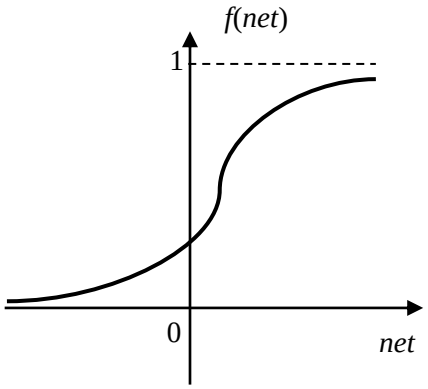


Figure 2.14: The logistic function

2.4.6.3 Training a neural net

Training often appears to be an ad hoc process in that most problems require much experimentation before acceptable results are attained. The performance of a neural network critically depends on training data. The training data must be representative of the task to be learnt. The purpose of training is to find the stable weights that can solve a particular problem at hand.

- Training a network consists of iteratively learning the appropriate weights that are required to recognize the input patterns.
- Training is required if the weights are not appropriate for correctly classifying the patterns.
- Training is achieved by feeding the network with a set of known sample patterns called training set (each pattern in the set has a known output), and then by comparing the network output with the expected (i.e. desired) output.

The training process is summarized as follows.

FOR all patterns in the training set

DO

present the pattern to the neural network

record the network response and compare it with the expected output

adjust the network weights as specified by the training law

END FOR

The designer of a neural network solution needs to:

- choose an appropriate network model;
- specify a network topology (i.e., number of units and their connections);
- specify learning parameters;

2.4.6.4 Strengths and weaknesses of ANNs

ANNs are easy to construct and deal very well with large amounts of noisy data. They are especially suited to solving nonlinear problems. They work well for problems where domain experts may be unavailable or where there are no known rules. ANNs are also adaptive in nature. This makes them particularly useful in fields such as finance where the environment is potentially volatile and dynamic. They are also very tolerant of noisy and incomplete data sets. Their robustness in storing and processing data earned them some applications in space exploration by NASA, where fault tolerant types of equipment are required. This flexibility derives from the fact that information is duplicated many times over in the many complex and intricate network connections in ANNs, just like in the human brain. This feature of ANNs is, in contrast to the serial computer where if one piece of information is lost, the entire information set may be corrupted.

The training process of an ANN itself is relatively simple. The pre-processing of the data, however, including the data selection and representation to the ANN and the post processing of the outputs (required for interpretation of the output and performance evaluation) requires a significant amount of work. However, constructing a problem with ANNs is still perceived to be easier than modeling with conventional statistical methods. There are many statisticians who argue that ANNs are nothing more than special cases of statistical models, and thus the rigid restrictions that apply to those models must also be applied to ANNs as well. However, there are probably more successful novel applications using ANNs than conventional statistical tools. The prolific number of ANN applications

in a relatively short time could be explained by the universal appeal of the relatively easy methodology in setting up an ANN to solve a problem. The restrictions imposed by many equivalent statistical models are probably less appealing to many researchers without a strong statistical background. ANN software packages are also relatively easier to use than the typical statistical packages. Researchers can successfully use ANNs' software packages without requiring full understanding of the learning algorithms. This makes them more accessible to a wider variety of researchers. ANN researchers are more likely to learn from experience rather than be guided by statistical rules in constructing a model and thus they may be implicitly aware of the statistical restrictions of their ANN models.

The major weakness of ANNs is their lack of explanation for the models that they create. Researches are currently being conducted to unravel the complex network structures that are created by ANN. Even though ANNs are easy to construct, finding a good ANN structure as well as the preprocessing and post processing of the data is a very time consuming process [26].

2.4.6.5 Backpropagation

The term Backpropagation network is used to describe a feed-forward neural network trained by using the Backpropagation learning algorithm, which is the modification of the least mean square algorithm. The Backpropagation learning algorithm modifies the network weights to minimize the mean squared error between the actual and desired outputs of the network. Thus, it is also sometimes called backward propagation of errors. In the context of learning, Backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function.

Backpropagation learning is known as supervised learning as it performs under the supervision of an external teacher [26]. The teacher provides the network with a desired or target response for any input vector. The actual response of the network to each input vector is then compared by the teacher with the desired response for that vector, and the network parameters are adjusted in accordance with an error signal which is defined as the difference between the desired response and the actual response. The adjustment is carried out iteratively in a step-by-step fashion with the aim of eventually

making the error signal for all input vectors as small as possible. When this has been achieved, then the network is believed to have built internal representations of the data set by detecting its basic features, and hence, to be able to deal with data that has not encountered during the learning process, that is, it can generalize its "knowledge". Supervised learning is by far the most widely used learning technique in ANNs. Before Backpropagation, there was no rule available for updating the weights of a multi-layered network undergoing supervised training.

A Backpropagation neural network is composed of forward and backward sweeps. The feed-forward sweep involves presenting an input pattern to input layer nodes that pass the input values onto the hidden layer nodes. The hidden layer nodes compute a weighted sum of its inputs and pass the sum through its activation function before presenting the result to the output layer. The backward sweep involves propagating the error at a higher layer of multi-layer network backwards to nodes at lower layers of the network. The gradient of the backward-propagated error is then used to determine the desired weight modifications for connections leading into the nodes of lower layers. In short, weights are modified in a direction corresponding to the negative gradient of an error measure.

Summary of the Backpropagation algorithm is as follows:

The first stage is to initialize the weights to small random values. The training is supervised by having a target pattern associated with an input pattern. Training continues as long as the change in the absolute value of the averaged squared error fails to be within some tolerance between one epoch and next epoch [9].

1. Read first input pattern and associated output pattern.

   CONVEREGE = TRUE.
2. For input layer, assign as net input to each unit its corresponding element in the input vector. The output for each unit is its net input.
3. For the first hidden layer units, calculate the net input and output:

$$net_j = w_0 + \sum_{i=1}^{n} x_i w_{ij}, \qquad o_j = \frac{1}{1+\exp(-net_j)}.$$

   If it is more than one hidden layer, repeat step 3 for all subsequent hidden layers.
4. For the output layer units, calculate the net input and output:

$$net_j = w_0 + \sum_{i=1}^{n} x_i w_{ij}, \qquad o_j = \frac{1}{1+\exp(-net_j)}.$$

5. Is the difference between the target and output patterns within tolerance? IF No THEN CONVERGE = FALSE.
6. For each output unit, calculate its error:

$$\delta_j = (t_j - o_j) o_j (1 - o_j).$$

7. For the last hidden layer, calculate error for each unit:

$$\delta_k = o_j (1 - o_j) \sum_k \delta_k w_{kj}.$$

   If it is more than one hidden layer, repeat step 7 for all subsequent hidden layers.

8. For all layers, update weights for each unit:

$$\Delta w_{ij}(n+1) = \eta(\delta_j o_i) + \alpha \Delta w_{ij}(n).$$

   After processing all training patterns through the above steps, if CONVERGE is true, i.e. if the difference between the target and actual outputs for all training patterns are within tolerance, training is completed. If CONVERGE is false, the training process must be repeated again.

2.4.6.6 Some important parameters in Backpropagation learning

The stability or convergence of the iterative learning process, i.e. Backpropagation learning, depends on a number of parameters, especially the ones used in the weight adaptation rule of the Backpropagation algorithm.

One of them is the η, the learning rate parameter. The learning rate determines the amount of correction term that is applied to adjust the neuron weights during training. Small values of the learning rate increase time but tend to decrease the chance of overshooting the optimal solution. At the same time, they increase the likelihood of becoming stuck at local minima. Large values of the learning rate may train network faster, but may result in no learning occurring at all. The adaptive learning rate varies according to the amount of error being generated, i.e. the larger the error, the smaller the η values and vice-versa.

If the ANN is heading towards the optimal choice of η, it will accelerate the training. Otherwise, it will decelerate the training. In summary,

- The learning rate is a positive number ($\eta > 0$) with the following impact:
  - If it is too small, then the convergence might be needlessly slow.
  - If it is too large, the convergence might overshoot (and miss the minimum).
- Optimal value of η leads to minimum error in one learning step.
- Recommended range: $0 < \eta \leq 1$.
  - If the training specimen has little or no noise, then $\eta \rightarrow 1$.
  - The noisier the training specimen, the lower should be the value of η.
- Design alternatives:
  - Learning rate is constant throughout the training session.
  - Learning rate is decreased as the training progresses.
- If learning rate is small enough to ensure convergence, then its only influence is on the speed at which the ANN error attains the minimum.

Another important parameter for Backpropagation learning is the momentum value, α. It determines how much of the previous corrective term should be remembered and carried on in the current training. The larger the momentum value, the more emphasis is placed on the current correction term and the less on previous terms. It serves as a smoothing process that brakes the leaning process from heading in an undesirable direction.

- Problem: occasionally, when the error slope is very small, the training error set stops decreasing and stalls at some value higher than the acceptable level; in this case, the Backpropagation network might not train within a reasonable period of time.
- Idea: recall the inertial behavior of physical objects that tend to preserve their motion state unless acted upon by an outside force.
- Solution: training failures can be avoided by adding a momentum term that will allow the weight vector to continuously change towards the error reduction (and move out of a local minimum).
- Momentum constant α, is a positive integer.
  - If $\alpha > 0$, it will enable weight changes even in the absence of error.

- The α should be kept high (0.5 to 0.9) to compensate for lower learning constant (and to speed-up the training). Increased above 0.9 may adversely affect the learning.

2.4.6.7 Considerations on the implementation of Backpropagation

The Backpropagation algorithm can be implemented in two different modes: *on-line mode* and *batch mode*. In the on-line mode, the error function is calculated after the presentation of each input pattern. This error function is usually the Mean Square Error (MSE) of the difference between the desired and actual responses of the network over all the output units. Then the new weights remain fixed and a new pattern is presented to the network and this process continues until all the patterns have been presented to the network. The presentation of all the patterns is usually called one *epoch* or one *iteration* [9]. In practice, many epochs are needed before the error becomes acceptably small.

In the batch mode, the error signal is calculated for each input pattern but the weights are modified only when all input patterns have been presented. Then the error function is calculated as the sum of the individual MSE errors for each pattern and the weights are accordingly modified (all in a single step for all the patterns) before the next iteration.

Gradient descent can also become stuck in local minima of the cost function. These are isolated valleys of the cost function surface in which the system may "stuck" before it reaches the global minimum. This is so because in these valleys every change in the weight values causes the cost function to increase and hence the network is unable to escape. Local minima are fundamentally different from temporary minima as they cause the performance improvement of the classification to drop to zero and hence the learning process terminates event though the minimum may be located far above the global minimum. Local minima may be abandoned by including a momentum term in the weight is a stochastic learning algorithm in nature. The momentum term can also significantly accelerate the training time that is spent in a temporary minimum as it causes the weights to change at faster rate. Other approaches include the modification of the cost function or the employment of techniques such as simulated annealing [29].

CHAPTER 3

SYSTEM IMPLEMENTATION

Speech recognition enables the recognition and translation of spoken language into text by computers. This technology has found its application on various fields of automatic phone response systems, robotics, automatic vehicle control systems, computer gaming, health care, military, etc. Although there had been a lot of researches done for speech recognition in various languages throughout the world, there are still very few systems for Myanmar language. In this thesis, an isolated Myanmar speech recognition system developed by using the Mel Frequency Cepstral Coefficients (MFCC) and Artificial Neural Network (ANN) techniques is proposed. This chapter mainly discusses the implementation of the system.

3.1 Generalized structure of ASR system

Figure 3.1 depicts the generalized structure of the proposed speech recognition system. It is composed of three major steps: preprocessing, feature extraction, and recognition/classification model. The first step is the preprocessing step carried out by using the Audacity software in this thesis. The main purpose of this step is to remove the unwanted parts, such as silence, from the input speech and to find out the start and end points of each word. The second step is to find the acoustically distinctive features representing each input speech. The MFCC technique is used to extract the features from the preprocessed isolated spoken words. Then, the final step is to train and test the Backpropagation recognition model by using the MFCC features.

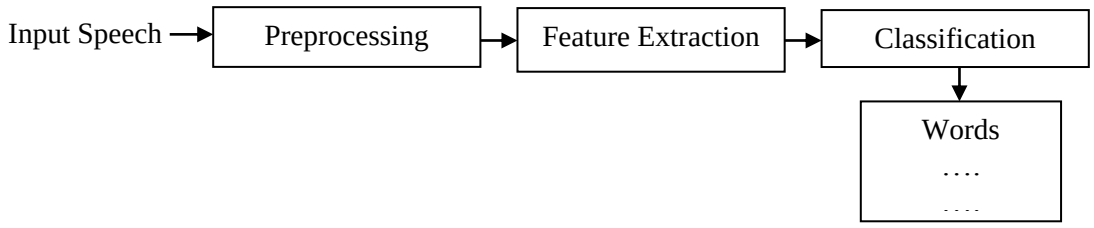


Figure 3.1: Generalized structure of an ASR system

3.1.1 Preprocessing

The speech production process involves generating voiced and unvoiced speech in succession, separated by what is called the silence region. Silences in speech can be due to hesitation, stutter, self-correction or a deliberate slowing of speech to clarify or aid the processing of ideas. During silence region, there is no excitation supplied to the vocal tract and hence no speech output. However, silence is an integral part of the speech signal. Without the presence of silence region between voiced and unvoiced speech, the speech will not be intelligible. Furthermore, the duration of silence along with other voiced or unvoiced speech is also an indicator of the certain category of sounds. Although silence region is unimportant from amplitude/energy point of view, its duration is very essential for intelligible speech [6].

Preprocessing of speech signals, i.e. separating the voiced region of the captured signal from the silence/unvoiced portion, is considered a crucial step in the development of a robust and efficient speech or speaker recognition system. This is because most of the speech or speaker specific attributes present in the voiced part of the speech signals. Extraction of the voiced part of a speech signal by marking and/or removing the silence and unvoiced regions leads to the substantial reduction in the computational complexity at later stages. Other applications of classifying speech into silence/unvoiced region and voiced region are fundamental frequency estimation, formant extraction or syllable marking, stop consonant identification, and end point detection for isolated speech signals [17].

One of the most commonly used speech preprocessing methods is short-time energy (STE). The STE is an acoustical feature that is correlated to samples' amplitudes within a frame, i.e. short period of time. It is used to measure the amount of energy in a sound at a specific instance in time and calculated as follows. A speech signal $x=\{x_1, x_2, \ldots, x_n\}$, where n is the total number of samples in the speech signal, is fragmented into small frames. Each frame is of ω samples, where $\omega<n$. The energy of speech is calculated frame-by-frame by using eq. 3.1 [23].

$$Energy = \sum_{i=1}^{\omega} x_i^2. \tag{3.1}$$

The STE value is high for voiced speech and low for unvoiced/silence speech. The STE methods for voiced-unvoiced separation are generally fast; however, these methods are hindered by the fact that the thresholds needed to implement them are chosen on an ad hoc basis. This means that the recognition system has to be retuned every time there is some change in the ambience.

Another commonly used speech preprocessing method is zero crossing rate (ZCR). The ZCR is a measure of the number of times in a given time interval/frame that the amplitude of the speech signals passes through the value of zero (sign changes). The ZCR is low for voiced part and high for unvoiced part. Like STE, the ZCR is calculated frame-by-frame as follows [23].

$$ZCR = \sum_{i=1}^{\omega} \frac{\left| sign(x_i) - sign(x_i - 1) \right|}{2} \tag{3.2}$$

According to the literature, the ZCR and STE methods are also widely used for end point detection of the isolated speech signals. However, it was found out from our experiments that the end point detection accuracy of ZCR depends on the words and is not always promising. Figure 3.2 shows the results of ZCR for two different signals. As shown in the figure, the ZCR result for the first signal seems correct, i.e. low ZCR values for voiced regions; whereas, the ZCR result for the second signal clearly contradicts the theory, the values of ZCR for some voiced regions are higher than the unvoiced regions. In addition, using the STE alone is not also good enough for feature extraction. By combining the ZCR and STE, the end point detection accuracy can be increased at the expense of increased computational complexity. Thus in this system, the utterances are manually preprocessed for end point detection and silence removal by using the Audacity software.

Audacity is a free, open source (cross-platform) digital audio editor, recorder, and mixer. Audacity is available under the GNU General Public License and it offers support for over 20 languages and runs on Microsoft Windows®, Mac OS X®, and Linux®. It is a sophisticated software application that comes with an extensive list of features such as:

- Recording from a microphone or mixer

- Import/export of WAV, AIFF, AU, FLAC, Ogg Vorbis, and MP3 files (via LAME encoder)
- Advanced editing (cut, copy, paste, delete commands with unlimited "Undo" and "Redo", and multi-track mixing)
- Digital effects (change the pitch, remove background noises, remove vocals, alter frequencies, create voice-overs for podcasts, etc.)

The full list of Audacity features can be found on its official homepage and for more detail of Audacity, refer reference [4].

Figure 3.3 shows how to use the Audacity for end point detection and silence removal of a speech signal. The upper figure shows the original speech surrounded by silence at the beginning and end (straight line in the figure). The "Cut" command symbolized by a scissor can be used to remove the unwanted parts by just selecting those parts with mouse and clicking that symbol. The lower figure shows the silence-free preprocessed signal.

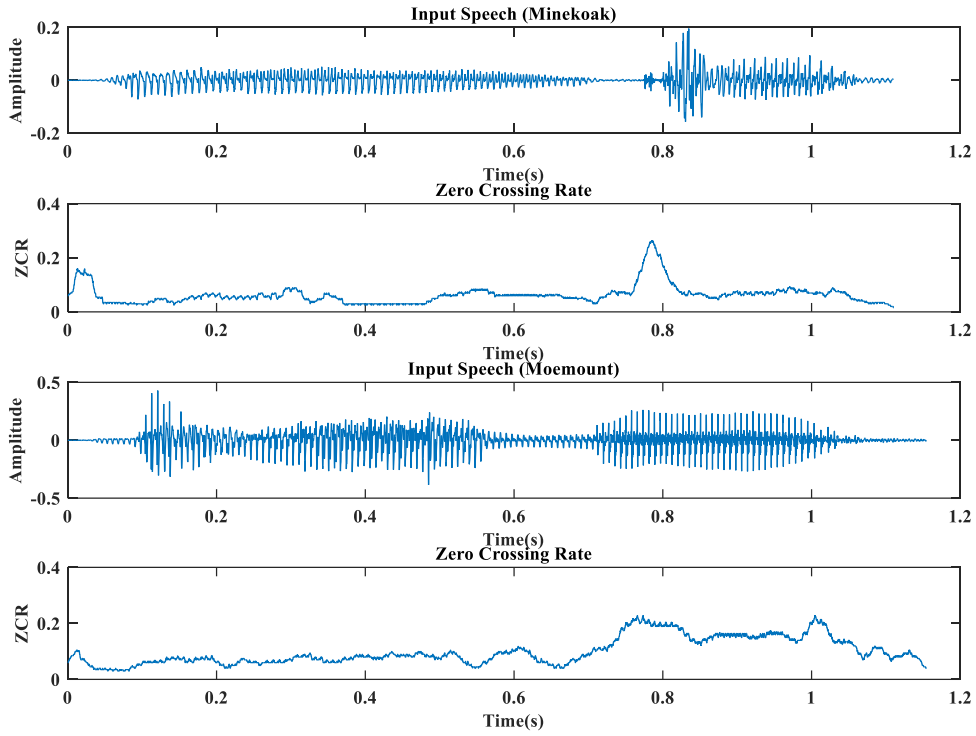


Figure 3.2: ZCR results for the word "မိုင်းခုတ်" and "မိုးမောက်"

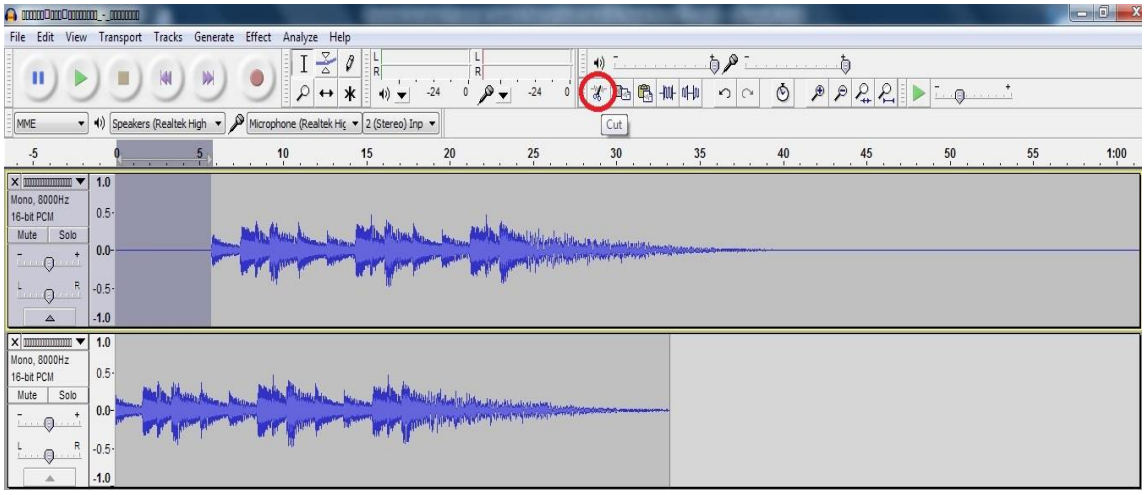


Figure 3.3: Silence removal in Audacity

3.1.2 Feature extraction

A speech signal is composed of samples which are highly redundant and the existence of some samples cannot be perceived by human ear. Thus, discarding those imperceptible samples and reducing the data size of speech without losing acoustically identifiable components is very desirable in speech/speaker recognition applications.

Feature extraction techniques extract the components of a speech signal that are good for identifying the linguistic content and discard all the other stuff that carries information like background noise, emotion, among others. There have been various speech feature extraction techniques such as MFCC, PCA, LPC, etc, as discussed in Chapter 2. Each technique has its own weak and strong points and among them, the MFCC outperforms any other parametric representation such as LPC and real cepstrum coefficients. The MFCC uses the Mel scale which is based on the human ear scale and it is one of the most powerful feature extraction techniques used in ASR systems. Thus in this system, Mel frequency spectral coefficients are considered as the features defining the input speech. The main steps of the MFCC calculation are summarized as follows.

Step 1: The input speech is passed through a high-pass filter with the aim of increasing the energy of the signal at high frequencies, which also contain signal information.

Step 2: The preprocessed speech is blocked into a number of frames (20-40ms long).

Step 3: In order to reduce discontinuities at the start and end of a frame or to be smooth of the first and last points in the frame, windowing function is used.

Step 4: FFT is executed to obtain the magnitude frequency response of each frame.

Step 5: Each frequency is then converted to Mel scale, which is a perceptual scale that helps to simulate the way human ear works. It corresponds to better resolution at low frequencies and less at high.

Step 6: Logarithmic operation is applied to the absolute magnitude of the coefficient obtained after Mel scale conversion. This operation discards the phase information, making feature extraction less sensitive to speaker dependent variations.

Step 7: The log Mel spectrum is finally converted to the time domain by applying the DCT to obtain the Mel frequency cepstral coefficients.

In general, the number of cepstral coefficients of the MFCC depends on the number of filters used in the Mel filterbank (refer step 5 above.) For example, if the Mel filterbank is made up of 20 filters, the result will be 20 feature vectors. The recommended number of filters is 20-40 filters. If we are dealing with the signals that contain a lot of closely spaced frequencies and want to resolve them, the number of filters might be increased. Among the resulting feature vectors, the first one represents the average power in the speech signal and it is not often used in recognition applications because the average power varies considerably depending on the microphone placement and channel. In the proposed system in this thesis, the last 19 feature vectors (resulting from 20 filters filterbank) are used to represent the input speech and to train the neural net.

Before applying the MFCC features to the neural network, there is a fact needed to be considered. The resulting MFCC features determine the number of input neurons of the neural net, *n* features means *n* input neurons. Traditional neural networks use the same network topology for all input patterns and thus the size of the MFCC features needs to be the same for all inputs. However, speech is a time-varying data. Even for the same word uttered by the same person, the size of the features vector may be different from time to time and yields time-varying input neurons.

For clear understanding, Figure 3.4 shows the time-domain waveforms of the word "မိုင်းခတ်", as an example, uttered by the same speaker at two different times. As shown in the figures, even for the same word spoken by the same person, the two waveforms have different number of samples (x-axis values) and different amplitudes (y-axis values). Similarly for the MFCC shown in Figure 3.5, the first word has 1197

features, whereas the second word has 1349 features. In order to solve this time-series problem in this system, the feature vectors of all training data are padded with zero, i.e. normalized, so that they are the same in length with the longest anticipated feature size (1800 features in this thesis).

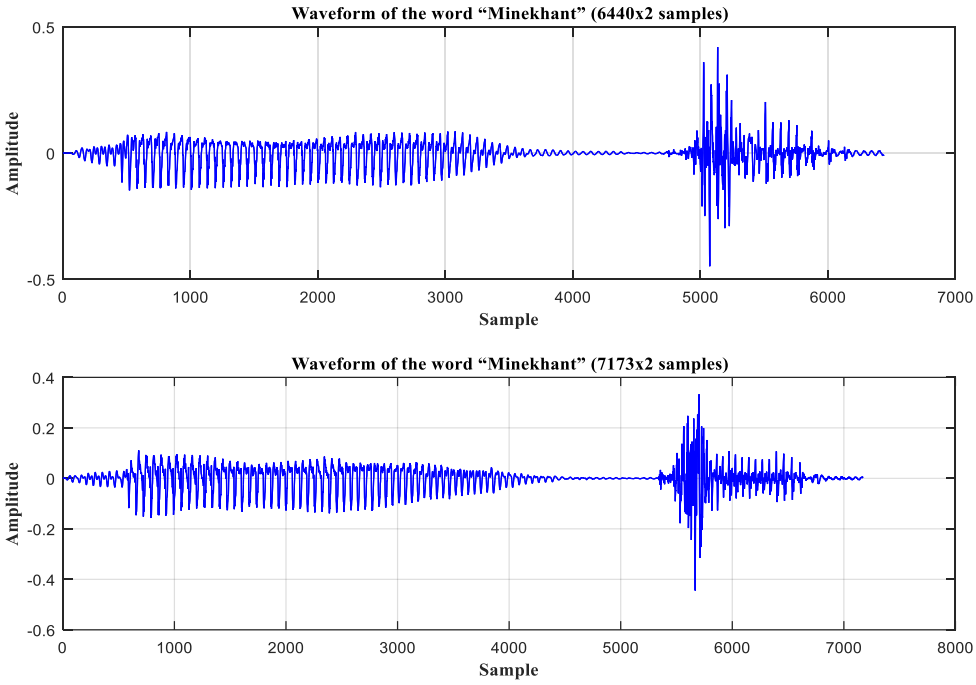


Figure 3.4: Waveforms of the words "မိုင်းခတ်" by same speaker but at different times

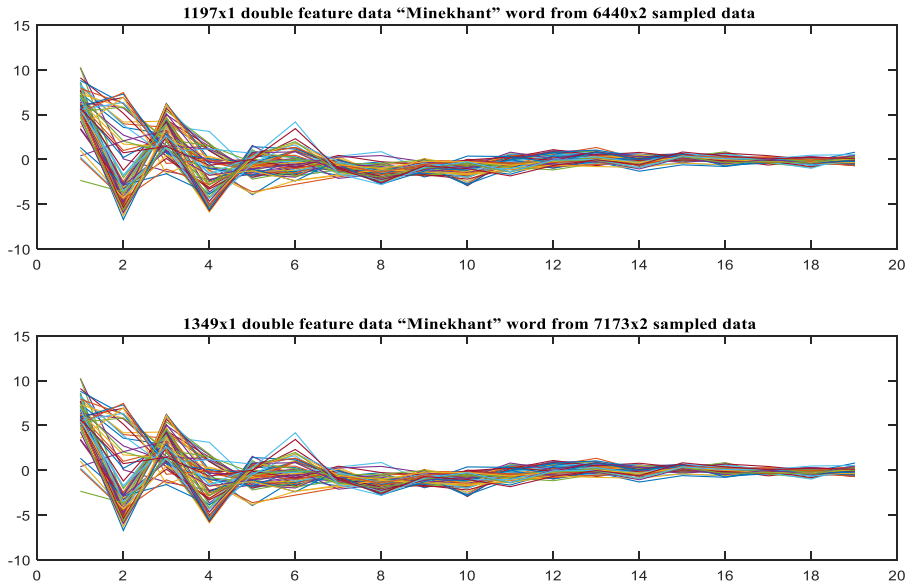


Figure 3.5: MFCC feature of the words "မိုင်းခတ်" by same speaker but at different times

3.1.3 Classification

As a classification/recognition model in this thesis, an artificial neural network is utilized. The ANNs are computing systems vaguely inspired by the biological neural networks that constitute human brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. In this thesis, a Backpropagation network model is trained and tested by using the MFCC features as input. Backpropagation is a supervised learning algorithm that distributes the error term back up through the layers by modifying the weights at each node. It more generally accelerates the training of multi-layer networks and can effectively solve the nonlinear problems like the exclusive-or.

The network topology used in this thesis is shown in Figure 3.6. At the input layer, there are 1800 input nodes, each of which represents an MFCC feature. The hidden layer consists of 100 hidden nodes, chosen as per the system requirement. The output layer consists of 20 nodes that represent the number of words supposed to be recognized by the recognition model, i.e. 20 names of the cities in Myanmar in this thesis.

The input data to an ANN are usually split into training and testing datasets. It is really only during testing that the true performance of a neural network is revealed since it is possible to train a network successfully only to find it underperforms during testing. During training, each input pattern will have an associated target pattern. The whole objective of training is to find a set of network weights that provide a solution to a particular problem at hand. The training phase of the proposed system is as follows:

Step 1: Before training commences, firstly, the weights are set to small random values. The learning rate and momentum values are also initialized.

Step 2: A pattern is presented to the network as input.

Step 3: The output of each neuron is computed by applying the sigmoid function, which is a nonlinear activation function, on the weighted sum of its inputs.

The above steps comprise the feed-forward sweep of the network.

Step 4: Error vector is calculated by comparing the actual and desired outputs of each neuron to determine how the weights should change; then the weights are changed accordingly through the backward sweep.

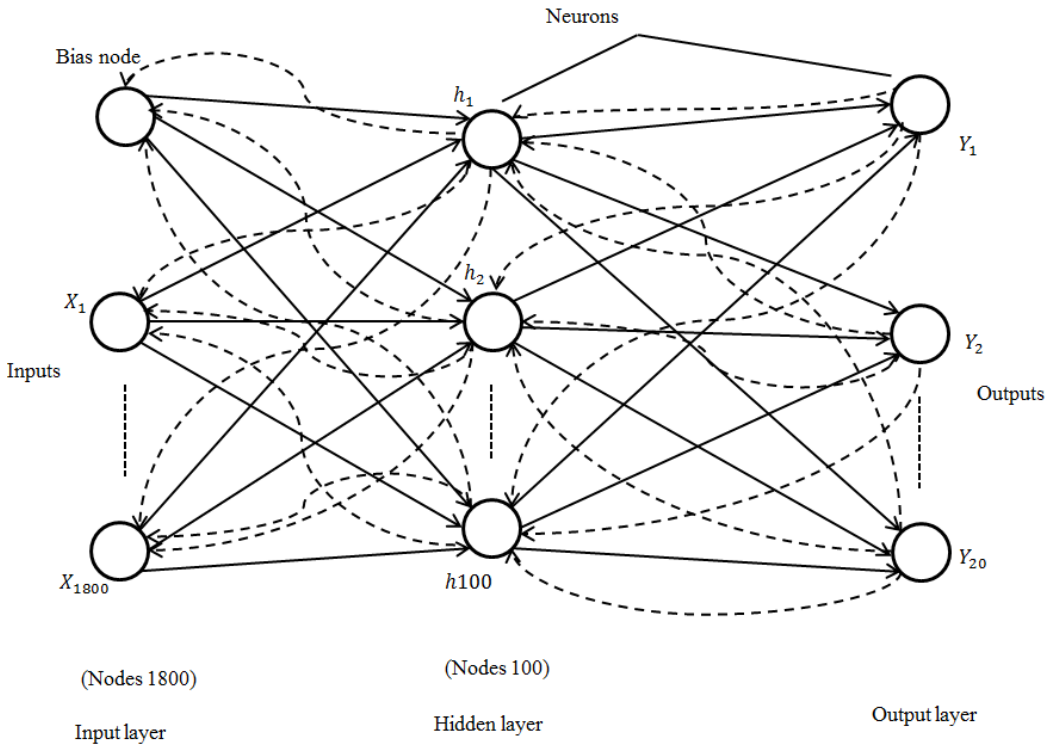


Figure 3.6: Neural network topology of the proposed system

The above steps (2 to 4) are repeated for all patterns. The patterns are continually presented to the network epoch after epoch until during one epoch all actual outputs for each pattern are within tolerance.

Step 5: Finally, the stable weights are stored as training data weight.

During the testing phase, a new pattern is presented to the network for testing the performance of the recognition model. Testing network consists of only the feed-forward sweep. The stable weights from training are used for calculating the classification/recognition result of the model.

The detailed system flow of the proposed speech recognition system is shown in Figure 3.7. For training phase,

Step 1: The speeches used in this system are recorded in mobile phones, which saved the sound in ".amr" format. The MATLAB software used to simulate the proposed system only accepts the sound files in ".wav" format. Thus, the ".amr" format files are converted to ".wav" format.

Step 2: Audacity software is used to detect the start and end point of each word.

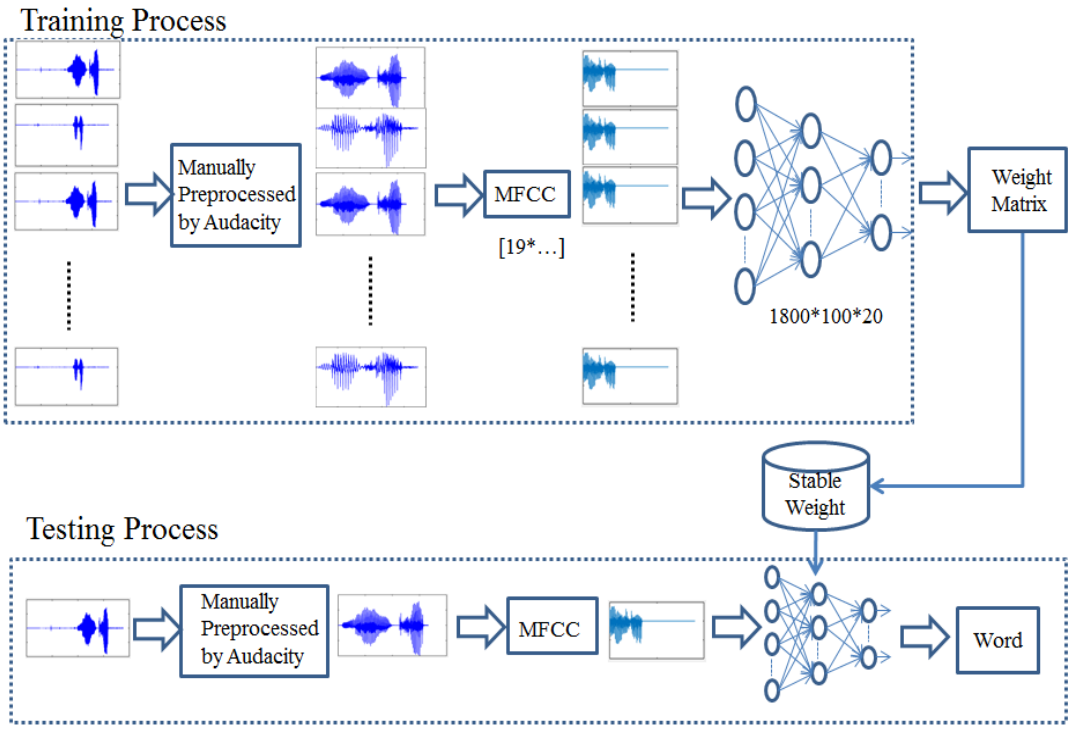


Figure 3.7: System flow of the proposed speech recognition system

Step 3: MFCC features are extracted from the preprocessed words and normalized if necessary so that the size is to be 1800 (the longest feature size in this system).

Step 4: The recognition model is trained by using the Backpropagation algorithm epoch after epoch until the stable weights can be obtained from the MFCC features.

Testing phase is exactly the same as the above mentioned training phase, apart from Step 4 in which the recognition model is tested with the MFCC features of the test speeches and by using the stable weights stored during training.

3.2 Speech database

The proposed system in this thesis implements a speaker independent recognition model for Myanmar Language. The system tries to recognize twenty names of the cities in Shan state, Kachin state, and Rakhine state in Myanmar. The word list is mentioned below.

မိုးညှင်း၊ မိုးကောင်း၊ မိုးမောက်၊ မြောက်ဦး၊ မိုင်းဆတ်၊ မိုင်းဖြတ်၊ မိုင်းရယ်၊ မိုင်းမော၊ မိုးနဲ့၊ မိုင်းပန်၊ မိုင်းကိုင်၊
မိုင်းခတ်၊ မောက်မယ်၊ မိုင်းယောင်း၊ မိုင်းလား၊ မိုင်းယန်၊ မိုင်းနောင်၊ မိုင်းငေါ့၊ မိုင်းယု၊ မိုင်းခုတ်။

The proposed system consists of a speech database that is made up of training and testing datasets with 2400 and 400 utterances each. All of the isolated words used in the training and testing phases are recorded by using the mobile phones in a normal room condition. The 10 speakers (4 males and 6 females) participated in training and speaker dependent testing say each word thirteen times. The duration of each recorded speech is 1-2 seconds and the whole database is approximately 1.57 hours long. The recording format is WAV and the sampling rate is 8000 Hz.

Of the 400 utterances of the testing dataset, 200 words are used for speaker dependent testing and another 200 words are used for speaker independent testing. For speaker dependent testing, the words are spoken by the same speakers who participated in training and whereas for speaker independent testing, the words are spoken by new speakers who did not participate in training.

3.3 Performance evaluation of an ASR system

The performance of an ASR system is generally measured in terms of the word recognition rate which is the ratio between correctly classified words and total number of tested words. In this system, the accuracy or recognition rate is computed by the following equation.

$$\text{Accuracy} = \frac{\text{No. of words correctly recognized}}{\text{Total no. of words}} \times 100\%. \tag{3.3}$$

CHAPTER 4

RESULT AND DISCUSSION

The proposed neural network model, shown in Figure 3.6, is trained with the Backpropagation algorithm by using the MFCC features of 2400 utterances, momentum value of 0.1, and learning rate of 0.4. During the training phase, the network becomes stable at 800-epoch with error of 0.00019 while the error tolerance is set to 0.0002. The simulation was done on a computer with Intel Core-i5 processor @ 2.2 GHz.

4.1 Speaker dependent testing

Table 4.1 shows the recognition result for speaker dependent testing with 200 test words. It can be seen from the table that the words မိုင်းငဲ့၊ မိုင်းပန်၊ မိုင်းရယ်၊ မိုင်းယန်၊ မိုင်းယု၊ မိုးကောင်း၊ မိုးမောက်၊ မိုးနဲ၊ မိုးညှင်း၊ မြောက်ဦး၊ and မိုင်းယောင်း are fully recognized with 100% accuracy. Some words such as မိုင်းခတ်၊ မိုင်းခုတ်၊ မိုင်းလား၊ မိုင်းမော၊ and မိုင်းဖြတ် are recognized with 90% accuracy and the remaining words appear as the least recognized ones with 80% accuracy. It is because the voices of the speakers who speak out မိုင်းကိုင်၊ မိုင်းနောင်၊ မိုင်းဆတ်၊ and မောက်မယ် are affected by little noise and their way of speaking is also noticeably different from while training. Averagely, the proposed system achieved 93.5% accuracy.

4.2 Speaker independent recognition

Regarding speaker independent recognition, the proposed system is tested with 200 test words uttered by new speakers different from the ones participated in training. It can be seen from Table 4.2 that the words မိုင်းကိုင်၊ မိုင်းယု၊ မိုးမောက်၊ မြောက်ဦး၊ and မောက်မယ် are fully recognized with 100% accuracy, whereas the words မိုင်းခတ်၊ မိုင်းရယ်၊ and မိုးညှင်း are recognized with 90% accuracy. And then, the words မိုင်းမော၊ မိုင်းနောင်၊ မိုင်းပန်၊ မိုးကောင်း၊ and မိုးနဲ are recognized with 80% accuracy. The words မိုင်းခုတ်၊ မိုင်းလား၊ မိုင်းငဲ့၊ မိုင်းဖြတ်၊ မိုင်းဆတ်၊ and မိုင်းယောင်း are recognized with 40%-70% accuracy. Unfortunately, the word မိုင်းယန် is the least recognized one with 30% accuracy. It is because the voices of the new speakers who speak out မိုင်းယန် are too quiet and the way of speaking is also noticeably different

from the trained speakers. As an average, the system achieved the speaker independent recognition rate of 76.5%.

Table 4.1: Speaker dependent recognition result

Myanmar word	No. of samples for testing	Testing result	
		No. of properly recognized words	Recognition rate (%)
မိုင်းခတ်	10	9	90
မိုင်းကိုင်	10	8	80
မိုင်းခုတ်	10	9	90
မိုင်းလား	10	9	90
မိုင်းမော	10	9	90
မိုင်းနောင်	10	8	80
မိုင်းဝေ့	10	10	100
မိုင်းပန်	10	10	100
မိုင်းဖြတ်	10	9	90
မိုင်းဆတ်	10	8	80
မိုင်းရယ်	10	10	100
မိုင်းယန်	10	10	100
မိုင်းယု	10	10	100
မိုးကောင်း	10	10	100
မိုးမောက်	10	10	100
မိုးနဲ	10	10	100
မိုးညှင်း	10	10	100
မြောက်ဦး	10	10	100
မောက်မယ်	10	8	80
မိုင်းယောင်း	10	10	100
Total	**200**	**187**	**93.5**

Table 4.2: Speaker independent recognition result

Myanmar word	No. of samples for testing	Testing result	
		No. of properly recognized words	Recognition rate (%)
မိုင်းခတ်	10	9	90
မိုင်းကိုင်	10	10	100
မိုင်းခုတ်	10	6	60
မိုင်းလား	10	5	50
မိုင်းမော	10	8	80
မိုင်းနောင်	10	8	80
မိုင်းငဲ့	10	7	70
မိုင်းပန်	10	8	80
မိုင်းဖြတ်	10	6	60
မိုင်းဆတ်	10	4	40
မိုင်းရယ်	10	9	90
မိုင်းယန်	10	3	30
မိုင်းယု	10	10	100
မိုးကောင်း	10	8	80
မိုးမောက်	10	10	100
မိုးနဲ့	10	8	80
မိုးညင်း	10	9	90
မြောက်ဦး	10	10	100
မောက်မယ်	10	10	100
မိုင်းယောင်း	10	5	50
Total	**200**	**153**	**76.5**

4.3 Discussion on the results

From the results in Table 4.1 and Table 4.2, it can be seen that the recognition rate of the proposed system is not satisfying for speaker independent testing. During the experiment, it was found that the performance of the recognizer depends on the speakers' way of speaking. If the new speakers sound alike the trained ones, the recognizer can recognize well; otherwise it fails. It is because the number of trained speakers in this system is too few and thus the recognizer fails to generalize some new speakers.

In addition, the speakers participated in this experiment are just graduate students who have no knowledge and experience about how to pronounce the words that will be helpful for ASR systems. Thus, their way of speaking such as rate of speaking and loudness might also be the problems. Moreover, the speeches used in this system were recorded with mobile phones in a normal room condition. Thus, the characteristics of the recording devices in mobile phones and background noises during recording might also hinder the performance of the system.

The choice of the preprocessing techniques for end point detection and silence removal is also very important. In this system, it was firstly experimented that the ZCR and STE to be used for end point detection. The preprocessed words by these techniques were applied to the MFCC feature extraction process; then, the resulting features were used to train the recognition model. The same network shown in Figure 3.6 was trained with the MFCC features of 2400 utterances by ten speakers (six females and four males), momentum value of 0.1, and learning rate of 0.4 (the same parameters used in the proposed system with manual preprocessing). During the training phase, the network got stuck at the error of 0.008 even at 5000-epoch while the error tolerance was set to 0.0002. Thus, it can be seen that training the network with the features of ZCR and STE takes longer time to converge for the same training data and for the same parameters. The recognition accuracy was also 54.5% for speaker independent testing and 87% for speaker dependent testing, less than the results of manual preprocessing shown in Table 4.1 and Table 4.2.

Moreover, the effect of learning parameters, learning rate and momentum, is also very significant in controlling the convergence of Backpropagation training and testing accuracy. As discussed in Chapter 2, the learning rate determines whether the neural

network is going to make major adaptation after each learning trial or if it is only going to make minor adaptation. The choice of the learning rate has the dramatic effect on generalization accuracy as well as on the training speed. If the learning rate is too small, it can require orders of magnitude more training time than the one that is in an appropriate range. As for the momentum, it helps control the possible weight oscillations that could be caused by alternately signed error signals. Those learning parameters have the most significant impact on training time and performance of a neural network. Most commercial Backpropagation tools provide a rich variable setting of those parameter values that can be chosen as desired by the user and as per system requirement [10].

In addition, the choice of network topology also determines the performance of the recognition model. There is no rule of thumb in choosing the network topology. Having too many hidden neurons is analogous to a system that is over specified and is incapable of generalization. Having too few hidden neurons, conversely, can prevent the system from properly fitting the input data, and reduce the robustness of the system.

CHAPTER 5

CONCLUSION

This thesis has proposed an isolated Myanmar speech recognition system that was developed by using the Mel Frequency Cepstral Coefficients (MFCC) feature extraction and the Backpropagation neural network algorithm. The system was designed and implemented in MATLAB simulation software.

The proposed system was designed with the purpose of being able to recognize twenty isolated Myanmar speech, which are the names of the cities in Myanmar. According to the experimental results, the proposed system achieved the recognition rate of 93.5% for trained speakers (i.e. speaker dependent testing) and 76.5% for untrained speakers (i.e. speaker independent testing). Generally, it is really difficult for a speech recognition system to get 100% accuracy. As discussed in Chapter 4, the accuracy of the proposed system was below 100% because of some real life problem such as using the poor microphone, noisy environment, poor utterance of speakers, and choice of the neural network parameters as well. In addition, the speech database used in this system is too small, just 2800 utterances, and the number of trained speakers is also very few. If the size of the database and the number of speakers were increased, the recognition model is supposed to achieve more generalization.

5.1 Further extension

Noise is a really big deal in speaker or speech recognition systems. It can increase the error rate of a recognition system. It is thus recommended to use an appropriate noise cancellation and normalization technique to reduce the channel noise and the environmental effects. In addition, using more suitable training data set can improve the performance of the recognizer. Additional to the above mentioned facts, we can also play the various parameters of neural network, for example - the error threshold, learning rate, and the number of hidden nodes in order to get a different, perhaps better result. Moreover, other recognition tools like Hidden Markov Model (HMM), Dynamic Time Warping, Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM) that can efficiently deal with time-series data like speech can also be tried out. By considering

the above facts, it is expected that the proposed system based on “isolated speech recognition” can be extended to “continuous speech recognition” in the future.

REFERENCES

[1] A. H. Mahmoud and S. Alzaki Ali, "Speech to text conversion," Sudan University of Science & Technology, December 2014.

[2] A. Kopanicakova, M. Vircikova, and P. Sincak, "Gesture recognition using DTW and its application potential in human-centered robotics," Department of Cybernetics and Artificial Intelligence, FEI TU of Kosice, Slovak Republic.

[3] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck, "Discrete-time signal processing," University of Massachusetts, Dartmouth.

[4] "Audacity: free, open source, cross-platform audio software,"
Date of access: July 2018.
https://www.audacityteam.org.

[5] Aung Tun Tun Lwin, "Speech recognition for Myanmar digits using neural networks with LPC approach," University of Computer Studies, Yangon, May 2009.

[6] B. M. S. Rani, A. Jhansi Rani, T. Ravi, and M. Divya Sree, "Basic fundamental recognition of voiced, unvoiced, and silence region of a speech," International Journal of Engineering and Advanced Technology (IJEAT), vol. 4, issue 2, December 2014.

[7] B. Medhi and P. H. Talukdar, "Isolated Assamese speech recognition using artificial neural network," Indian Institute of Technology Guwahati (IITG), May 2015.

[8] C. Kurian and K. Balakrishnan, "Perceptual Linear Predictive Cepstral Coefficient for Malayalam isolated digit recognition," International Journal of Advanced Information Technology (IJAIT), vol. 1, no. 5, October 2011.

[9] Clarence N. W. Tan, "An artificial neural networks primer with financial applications examples in financial distress predictions and foreign exchange hybrid trading system," School of Information Technology, Bond University, Gold Coast, QLD 4229, Australia.

[10] D. Randall Wilson and Tony R. Martinez, “The need for small learning rates on large problems,” International Joint Conference on Neural Networks, July 2001.

[11] “What is automatic speech recognition?,” Date of access: June 2009. http://docsoft.com/Resources/Studies/Whitepapers/whitepaper-ASR.pdf.

[12] E. R. Rady, A. H. Yahia, E. A. El-Dahshan, and H. El-Borey, “Speech recognition system based on wavelet transform and artificial neural network,” Egyptian Computer Science Journal (ECS), vol. 37, no. 3, May 2013.

[13] Ei Mon Kyaw, “Speech command recognition using Dynamic Time Warping,” University of Computer Studies, Yangon, July 2015.

[14] “Vector quantization,” Date of access: July 2018. https://en.wikipedia.org/wiki/Vector_quantization.

[15] G. Boopathy and S. Arockiasamy, “Implementation of vector quantization for image compression – a survey,” vol. 10, issue 3, April 2010.

[16] G. Bulbuller, “Recognition of in-ear microphone speech data using multi-layer neural networks,” Naval Postgraduate School, March 2006.

[17] G. Saha, S. Chakroborty, and S. Senapati, “A new silence removal and end point detection algorithm for speech and speaker recognition applications,” in Proceedings of Eleventh National Conference on Communications (NCC), pp. 291-295, January 2005.

[18] G. Sarosi, M. Mozsary, P. Mihajlik, and T. Fegyo, “Comparison of feature extraction methods for speech recognition in noise-free and in traffic noise environment,” University of Technology and Economics, Budapest, Hungary, 2007.

[19] John Leis, “Digital signal processing using MATLAB for students and researchers,” University of Southern Queensland.

[20] K. R. Ghule and R. R. Deshmukh, “Feature extraction techniques for speech recognition: a review,” International Journal of Scientific and Engineering Research, vol. 6, issue 5, May 2015.

[21] M. D. Abdullah-al-MAMUN and F. Mahmud, “Performance analysis of isolated Bangla speech recognition system using Hidden Markov Model,” International Journal of Scientific & Engineering Research, vol. 6, issue 1, January 2015.

[22] M. D. Ali Hossain, M. D. Mijianur Rahman, U. K. Prodhan, and M. D. Farukuzzaman Khan, “Implementation of Backpropagation neural network for isolated Bangla speech recognition,” International Journal of Information Sciences and Techniques (IJIST), vol. 3, no. 4, July 2013.

[23] M. R. Gamit and K. Dhameliya, “Isolated words recognition using MFCC, LPC and neural network,” International Journal of Research in Engineering and Technology (IJRET), vol. 4, issue 6, June 2015.

[24] N. Dave, “Feature extractions methods LPC, PLP, and MFCC in speech recognition,” Gujarat Technology University, India.

[25] N. Desai, K. Dhameliya, and V. Desai, “Feature extraction and classification techniques for speech recognition: a review,” International Journal of Emerging Technology and Advanced Engineering, vol. 3, issue 12, December 2013.

[26] N. Samsudin, “Speech recognition using Backpropagation neural network via Internet,” Universiti Technologi MARA.

[27] N. Seman, Z. Abu Bakar, N. Abu Bakar, H. F. Mohamed, N. A. Sia Abdullah, P. Ramakrisnan, and S. M. Syed Ahmad, “The optimal performance of multi-layer neural network for speaker-independent isolated spoken Malay parliamentary speech,” Faculty of Computer and Mathematical Sciences, vol. 1, issue 1, 2010.

[28] R. S. Chavan, “An overview of speech recognition using HMM,” International Journal of Computer Science and Mobile Computing, vol. 2, issue 6, June 2013.

[29] Richard P. Lipmann, “An introduction to computing with Neural Nets,” IEEE ASSP Magazine, April 1987.

[30] Smita B. Magre, Ratnadeep R. Deshmukh, and Pukhraj P. Shrishrimal, “A comparative study on feature extraction techniques in speech recognition,” Marathwada University, Aurangabad.

[31] "Speaker dependent / independent - build a speech recognition circuit," Date of access: July 2018. https://www.imagesco.com/articles/speech/speech-recognition-tutorial02.

[32] Su Myat Mon and Hla Myo Tun, "Speech-To-Text conversion (STT) system using hidden markov model (HMM)," International Journal of Scientific and Technology Research, vol. 4, issue 6, June 2015.

[33] T. T. Thet, J. Na, and W. K. Ko, "Word segmentation for the Myanmar language," Journal of Information Science, vol. 34, issue 5, October 2008.

[34] T. Takiguchi and Y. Ariki, "PCA-Based speech enhancement for distorted speech recognition," Journal of Multimedia, vol. 2, no. 5, September 2007.

[35] T. Yang, "The algorithm of speech recognition, programming, and simulating in MATLAB," Faculty of Engineering and Sustainable Development, January 2012.

[36] T. Yang and N. Rothpfeffer, "The algorithms of speech recognition programming and simulating in MATLAB," University of GAVLE.

[37] Thiang and S. Wijoyo, "Speech recognition using Linear Predictive Coding and Artificial Neural Network for controlling movement of mobile robot," International Conference on Information and Electronics Engineering, May 2011.

[38] Thomas M. Sullivan, "Multi-microphone correlation-based processing for robust automatic speech recognition," August 1996.

[39] U. Shrawankar, "Techniques for feature extraction in speech recognition system: a comparative study," SGB Amravati University.

[40] V. Malarmathi and Dr. E. Chandra, "A survey on speech recognition," International Journal of Computer Trends and Technology (IJCTT), vol. 4, issue 9, September 2013.

[41] "What is speech recognition," Date of access: June 1996. http://www.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.

[42] Z. Z. Tun and G. Srijuntongsiri, “A speech recognition system for Myanmar digits,” International Journal of Information and Electronics Engineering, vol. 6, no. 3, pp. 210-213, May 2016.

[43] Zaw Min Tun, “Automatic speech recognition for Myanmar language,” University of Computer Studies, Yangon, October 2003.

Publication

[1] Nan Phyu Phyu Hsan and Twe Ta Oo, “A study on isolated Myanmar speech recognition via ANN,” Parallel and Soft Computing Conference, University of Computer Studies, Yangon, Myanmar, 2018.