NOVIA
UNIVERSITY OF APPLIED SCIENCES

Joint Speech and Speaker Recognition Using Neural Networks

Xiaoguo Xue

May 9, 2013

Bachelor's Thesis

Electrical Engineering

Vaasa,2013

BACHELOR'S THESIS

Author:	Xiaoguo Xue
Degree programme:	Electrical Engineering, Vaasa
Specialization:	Automation Technology
Supervisor:	Dag Björklund

Title: *Joint Speech and Speaker Recognition Using Neural Networks*

May 9, 2013 Number of pages 60

Abstract

Speech is the main communication method between human beings. Since the time of the invention of the computer people have been trying to let the computer understand natural speech. Speech recognition is a technology which has close connections with computer science, signal processing, voice linguistics and intelligent systems. It has been a "hot" subject not only in the field of research but also as a practical application. Especially in real life, speaker and speech recognition have been used very frequently.

Even though the aims of speaker recognition and speech recognition are different, they have the same algorithms.

Neural network is a technology which tries to mimic human brain functions. With the development of neural network these past few decades, using neural network in speech and speaker recognition has become very popular and successful.

In this thesis the main procedures that include signal pre-processing, feature extraction, neural network design and implementation,are introduced. The Mel Frequency Cepstrum Coefficients(MFCC) is the best available approximation of human ear features. Back propagation neural network is used to design the recognition system. Moreover an implementation has been made in Matlab platform. The experiment results show that the system works well and it can be improved by using more training samples. This research gives a good foundation for future implementation on a realtime DSP system.

Language: English Keywords: speaker recognition, speech recognition,

Contents

Chapter 1

Introduction

1.1 About this Thesis

The work presented in this thesis was done in order to support the teaching of a course in intelligent systems (mainly neural networks) at the Novia University of Applied Science in Finland and to create a foundation for future work on e.g. voice controlled robots. Novia offers multidisciplinary higher education with a practical orientation, training professionals for expert and development posts.It has about 4000 studetns and a staff of 400. The supervisor of the thesis was Dag Björklund.

1.2 Background

As the computer technology develops, it promotes the development of society. On the other hand the development of human society entails a higher challenge to the computer development. The communications between humans and computers are wider and deeper and communication functions by using mouse, keyboard and touch screen can not satisfy the quick, accurate and efficient interchange of information. How to send information in a more natural, more efficient and quicker way has became an urgent question.

From technology research to daily life, computers are involved in every aspect of people's daily life. Computers are used to accomplish many tasks. Considering this situation, intelligent communication between computers and humans, human-computer interaction, becomes one of the most important research fields.

Speech is one of the natural forms of human communication. Since childhood people can express themselves by speech, recognizing others by distinguishing their voices and under-

standing others by their speech. People are very good at speaker and speech recognition. The human brain uses neurons and synapses, modified with experience and provides a distributed form of associative memory. Motivated by this, speaker and speech recognition systems have been developed.

- Speaker recognition is the technology of letting a machine distinguish different speakers from each other. Depending on the different speakers different actions are implemented.
- Speech recognition is the technology of letting a machine understand human speech and, according to the meaning of the speech, implement the intention of the human.

These technologies involve wide cross-disciplinary research; close connections to computer science, telecommunications, signal processing, voice linguistics, neural networks and intelligent systems. [1, 2, 3, 4] For the past 30 years already, speaker recognition and speech recognition have been used in industry, military, traffic, medicine and daily use, and especially in automatic control, information processing, telecommunications and electrical system. As voice control technologies, speaker recognition and speech recognition will definitely affect the technologies in automation and machine operation.

The time of multimedia is here, urgent requirements for the development of speech recognition will promote speech recognition technology to have a breakthrough both theoretically and in its applications.

Joint speech and speaker recognition system has various applications, which variable from health care, military to daily use applications. Automation of complex operator-based tasks is one of the most popular applications, e.g., customer care, dictation, form filling applications, provisioning of new services, customer help lines, e-commerce, etc. One application which enables big advantage to people's life is voice commanding car or robot. E.g RACO technology brings Google voice commands to car which achieves the GPS tracking by voice commands. Controlling robots which can be used with different robots follow its master's commands and gives the right reaction, like voice controlled wheelchair which makes the disabled people to control the chair much easier and convenient .This can be a further implementation based on the research of this project.

1.3 The history of speech recognition

In the middle of the 20th century, speech recognition was created as a new subject. At that time the AT&T Bell laboratory implemented the first speech recognition system, which could recognize ten English numbers, the "Audry System". In 1997 IBM developed a Chinese ViaVoice speech recognition system [5] and many other companies developed speech recognition systems e.g SpeechWorks has the phone automatically speech recognition system, Microsoft and SpeechWorks cooperated to integrate speech recognition technology in Office XP, and developed the Microsoft Speech SDK. All those indicate that speech recognition technology has become more and more well versed and it will be one of the most important technology trends in the future information field [1, 2, 4].

In the 1960s the applications of computer sciences promoted the development of speech recognition. At this time the most important achievements were: the generation of Dynamic Programming(DP) and Linear Predictive (LP) technologies.

In the 1970s LP technology had developed further, the Dynamic Time Warping(DTW) technology was well developed, especially Vector Quantization(VQ) and Hidden Markov Model(HMM) had been put forward. In practice, systems for people with isolated word recognition based on LP and DTW technologies were achieved.

In the 1980s the main achievement is the continuous speech recognition. Many algorithms for continuous speech recognition have been developed e.g in Bell laboratory, Myers, Rabiner and Lee's hierarchical construction algorithm and frame synchronization. At the same time the research direction changed from template matching to statistical modeling techniques. HMM became the main speech recognition technology. In 1988, Carnegie Mellon University(CMU) using VQ/HMM methods implemented the SPHINX system, which could recognize continuous speech for 997 words, independently of the speaker. This is the first system for speaker independent (see Chapter 1.6), large vocabulary and continuous speech. It pioneered a new stage of speech recognition.

In the 1990s speech recognition went through a rapid development, in some fields speech recognition has become mature. As the multimedia time came, there was an urgent demand to implement speech recognition systems in practice. Another main trend was that: speech recognition started to combine with other fields. In the beginning of the 90s there were researches about speech recognition and natural speech processing combininged. In the middle of the 90s speech recognition was combined with machine translation technology and

the development of translation between different languages started. [1, 2, 4]

1.4 Trends in speech recognition

Speech recognition technology is one of the most popular and potential technologies. Generally speech recognition has changed the computer into an "intelligent" device. Speech is the most natural communication medium. With the development of computer science and speech processing, translation between different languages will be the active part of speech processing research. The design of a natural speech database, feature extraction of speech,using speech material to do acoustical model research,speech recognition arithmetic research,language translation and conversation processing research will be the speech technology's hotspot direction.

Another development of speech recognition is multimedia human-machine interaction with a combination of human body language and oral language. Nowadays this kind of usage of combined audio and visual sensation information is being researched widely globally, and is becoming one of the important development and research directions. Zhang Yaqin [6] who is a professor of the IT field proposed eight prophecies , of which one is that speech will be the new human-machine interface. [4]

1.5 Speaker recognition

The ultimate goal of speaker recognition has two steps: the basic goal is to identify the speaker irrespective of what is being said. A future task is to decide whether the speech belongs to a claimed identity. Generally speaker recognition is the method of automatically identifying who is speaking. The speaker is recognized based on his individual speech wave information.

1.5.1 Classification of speaker recognition systems

According to the application area, speaker recognition systems can be divided into speaker identification systems and speaker verification systems.

Speaker identification

Speaker identification is the process of determining which registered speaker gave the given utterance. In the speaker identification system the test utterance is compared and scored with the registered speakers and the one whose module matches the best is selected. Based on whether test utterances are allowed to come from any unknown identity, speaker identification can be divided into two types:

- Closed set identification. In this process the system knows that the test utterance belongs to one of the registered speaker already, so the system will force itself to give an identity to this test utterance in any case.

- Open set identification. In this process the system does not know if the test utterance belongs to the registered speakers or not. So if the system does not find any register speaker who is matched with the test, the system will reject to assign the utterance to any of the speakers.

The performance of speaker identification depends on three main factors:

- Text-dependent or text-independent.Text-dependent requires the speakers to say exactly the given or the same speech. The system knows beforehand what the speakers will say and so it is easier and quicker to make the decision. Although higher performance could be achieved by knowing the utterance beforehand, such systems are more prone to cheating. Text-independent does not require the speakers to utter the same speech, the speaker can be recognized by any speech. Here the system does not know what the speaker will say, a more flexible system will be needed.

- Speaker number and confusability. With more speakers, a larger database will be needed and also longer time for training and scoring. Another point is that when the test utterances are similar in pronunciation or when text-dependent speakers' utterances are similar, a more flexible and powerful system is needed to complete this task.

- System design. With different feature extraction methods , different neural network algorithms, different systems can be designed and different performance can be reached.

Speaker verification

Speaker verification is the process of accepting or rejecting the identity claim of a speaker. The system extracts parameters from the input speech signal to represent vocal characteristics and uses these information to build representative speaker models. When a test utterance comes with a claim, it tests its parameters under the claimed speaker's model, and calculates a similarity score. The decision is made depending on this score. If the score is below the threshold then the system will reject the test utterance, otherwise the system will accept the test utterance.
The performance of speaker verification depends on three main factors:

- Text-dependent or text-independent. Text-dependent requires the speaker to say exactly the given or the same speech. As technology develops people can imitate others' voices by using some software, so this system's security needs to be considered. Text-independent does not require the speaker to utter the same speech, the speaker can be recognized by any speech.For example the system generates random speech, the test speaker reads it and the system makes the decision. This is popular in the application of speaker verification.

- Registered utterance. This is the training samples and the basis of calculating similarity score, it has great effect on the performance of the system. On the other hand, the amount of the registered utterance and the duration of the registered utterance will make the decision different.

- System design. The method of calculating similarity score, the error resistance, system algorithms all give great effects on the result.

- Acoustic variability: This is a big challenge towards achieving high performance.

[7]
In Figure 1.1.one can see the difference between speaker identification and verification clearly. Nowadays many daily used applications have been developed based on this theory. For example electric bank, security system etc.

Figure 1.1: Speaker verification and identification

1.6 Factors affecting speech recognition performance

In speech recognition systems, different systems have different characters and designs according to the task requirements. But in most cases few factors are the same. These factors play a significant roles for the system state of accuracy:

Vocabulary size and confusability

According to how much vocabulary can be recognized, speech recognition can be divided into three kinds of different scales vocabulary speech recognition:

- Small scale vocabulary speech recognition
- Medium scale vocabulary speech recognition
- Large scale vocabulary speech recognition

Small-scale can identify less than 100 vocabulary while medium-scale can identify more than 100 vocabulary and large-scale can identify more than 1000 vocabulary.
The situation that always happens is that when people are talking, the listener misunderstands what the speaker says. For the human brain, which is very good at recognition, this still happens quite often. This mostly because of the large scale vocabulary and the confusability. A general rule is that it is easy to discriminate among a small set of words. As the size of vocabulary differs, it is different to discriminate among a different set of words. Moreover, if the vocabulary contains confusable words, this will make the task more difficult.

Speaker dependence vs. independence

- Speaker dependence : This system is used by a exact speaker. So the system needs to recognize different speeches from one speaker.
- Speaker independence : This system can be used by any speakers. In this system, the features become tuned to the speaker that it was trained on, and these features tend to be highly speaker-specific.

Speech recognition as mentioned before the goal is to recognize what is being said irrespective of who is speaking. Different speakers have different voice,tone, frequency etc, so different features can be extracted although for the same speech. With the speaker-specific features, bigger challenges need to be conquered in order to get a good performance.

Isolated, discontinuous, or continuous speech

As we all know, a kid can recognize a separate word earlier than an entire sentence. It is the same situation with a speech recognition system. One question is always considered first: What kind of speech needs to be recognized? Is it a word or a sentence? So the speech that needs to be recognized can be divided into:

- Isolated speech (single words) e.g identifying ten single numbers from 0 to 9, place names, control commands or Chinese syllables.
- Discontinuous speech (full sentences) consists of words that are separated by silence.
- The ultimate purpose of speech recognition is to make the computer understand natural language. The biggest character of natural language is continuous. Continuous speech recognition is the most difficult task in speech recognition. For example, continuous speech recognition is needed in dictation, translation machines or human-computer speech conversations.

Read or spontaneous speech

The way the speech was given is also an important factor that influences the performance of the system.Normal speech can be given in many ways, generally it can be divided into:

- Read speech: Speakers read and are recorded from prepared scripts .

- Spontaneous speech: The utterance is spontaneous. Spontaneous speech is vastly more difficult, because it tends to be peppered with dis-fluencies, false starts, incomplete sentences, stuttering, coughing, and laughter. Moreover, the vocabulary is essentially unlimited, so the system must be able to deal intelligently with unknown words.

1.6.1 Other factors affecting both speaker and speech recognition

Other factors that affect the level of difficulty in speech recognition, and therefore also the reliability, are for instance:

- Variability in speakers : Gender, speed, regional and social dialects, speaking style, emotional, physical states
- Variability in environments: background noise, reverberation.
- Variability in transmission channels and microphones
- Variability in expertise: Speech physiology, acoustic phonetics, digital signal processing, statistical pattern recognition

1.6.2 The settings for this project

After researching of the basics of speaker recognition and speech recognition,the settings for speaker recognition of this project can be summarised as:

- Speaker recognition: three speakers.
- Speaker recognition: Speaker identification.
- Speaker recognition: Text-dependent and text-independent.

The settings for speech recognition of this project can be summarized as:

- Speech recognition: Small vocabulary(three speeches)
- Speech recognition: Isolated read speech
- Speech recognition: Speaker dependent and speaker independent.

This gives a clear task map for the next stage's theory research and experiment.

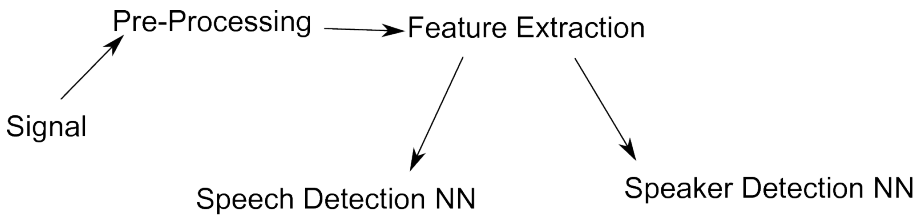


Figure 1.2: Speaker and speech recognition system using two neural networks

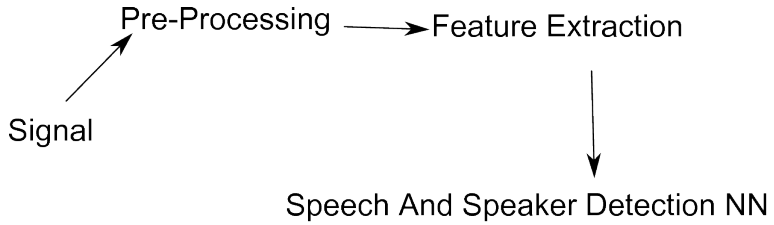


Figure 1.3: Speaker and speech recognition system using one neural network

1.7 Joint speech and speaker recognition, system overview

The system of joint speech and speaker recognition can complete the task of recognizing speaker and speech at the same time. Speaker recognition and speech recognition systems, are different in implementation details, but the basic technologies are similar.
For this project, the joint speech and speaker recognition systems have two different ways of implementation:

- With two neural networks for the system, each one for speaker and speech recognition respectively see Figure 1.2
- With one neural network which performances the speech and speaker recognition at the same time see Figure 1.3

In the recognition system, the signal is edited before it is fed to the neural network. First there is the pre-processing, which contains silence removal, normalization, pre-emphasis, framing and windowing. Then the windowed signal is sent to feature extraction, in which the needed features from the signal are extracted and compressed. After all that work, the signal is able to be fed to the neural network as inputs.
At this stage two different solutions are made according to the number of usable neural networks.

Chapter 2

Signal Capture and Pre-processing

Signal capturing is the first step that needs to be taken in this project. After this a pre-processing of signals will handle the signal processing before taking features from the signals. Normally the pre-processing of the speech signal contains pre-emphasis, framing and windowing. Normalization is also done at some point. A block diagram of the process is shown in Figure 2.1.

2.1 Capturing the signal

In the processing part the first step is to capture the signal we need. The signal from a human is an analog signal. When storing the speech to a computer, the analog signal needs to be digitized. So the first component of speech processing is speech signal measurement. When people talk to a microphone, the analog signal pressurizes the air, then the analog electric signals goes to the microphone. Analog speech digitization has two steps: Sampling and Quantification. The processing sequence is shown in Figure 2.2.

Sampling

The signals we got is analog signals. To process these signals in computers, we need to convert the signals to digital form which consists of digits 1 and 0. While an analog signal is continous in both time and amplitude, a digital signal is discrete in both domains. The

Figure 2.1: Pre-processing signal diagram

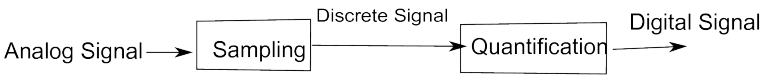


Figure 2.2: Speech signal digitization procedures [8]

values of the signal is measured at exact discrete time intervals [9, 10].

As for a human being, the adult male will have a fundamental frequency from 85 to 180 Hz, and the frequency of a typical adult female varies from 165 to 255Hz. [11]and children and babies have even higher fundamental frequencies.

The Shannon-Nyquist sampling theorem [12] states that: If the signal is band-limited, and the sampling frequency is higher than twice the signal bandwidth, the original continuous signal can be completely reconstructed from the samples. With lower sampling frequencies, aliasing will occur, which produces distortion from which the original signal cannot be recovered.

Human speech is generally below 5 kHz, so a sampling rate of over 10 kHz is required. In this thesis 11025 Hz sampling rate is used.

Quantification

Quantification means that the signal that is discrete in time domain but continuous in amplitude is turned into a signal that is discrete in amplitude. When doing this the amplitude values are cut into limited extent. According to the sampling precision, the samples that are in one extent will be given the same amplitude value. The dynamic range of the voice is limited by quantification, its unit is bit [8].

The prototype system we developed in this thesis used Matlab.

The setup was as follows:

- Three speakers: dag, lf and xxg.
- Three words: Left, Right and Forward
- Recording tools: Sound Recorder in Windows computer(64-bit operating system)
- Accessory: Microphone
- A/D converter doing the sampling and quantification

There are various pattern modeling/matching techniques can be used in joint speech and speaker recognition system. They include

- Dynamic Time Warping (DTW)
- Gaussian Mixture Model (GMM)
- Hidden Markov Modeling (HMM)
- Artificial Neural Network (ANN)
- Hybrid of ANN and HMM
- Fuzzy System
- Vector Quantization (VQ)

These are interchangeably used for speech, speaker modeling.Before the usage of HMM and ANN, DTW is very popular and successful in small vocabulary recognition system. Along the appearance of HMM, the development of recognition in large vocabulary, continual speech and speaker-independent system had great improvement. Using ANN in recognition system has big developing potential, but it has the common problems of long training time and poor generalization. So ANN has bigger challenge than other technologies. In this thesis NN is used as the recognition system. For NN system it needs to be trained to recognize the above speakers and words. For this we need to record a number of samples for training the system and for testing. We recorded 10 samples per speaker per word:

$$3speakers \cdot 3words \cdot 10samples/speaker/word = 90samples$$

So 90 samples were needed to be recorded. The training samples were recorded using a soundrecorder, but could also be recorded in Matlab using the `wavrecord` function:

```
1 Fs=11025; % sampling frequency 11025Hz
2 signal=wavrecord(5*Fs,Fs); %5*Fs samples
```

In a running system, that is, after training is completed and the system is used for recognizing new samples, the signals would of course be recorded in Matlab as above.
After the 90 samples the later processing started off-line (not realtime).
The stored test samples were read using the `wavread` function in Matlab, which can read the `.wav` files , get the information from the samples and store it in one vector variable which has the same name as the `.wav` file.

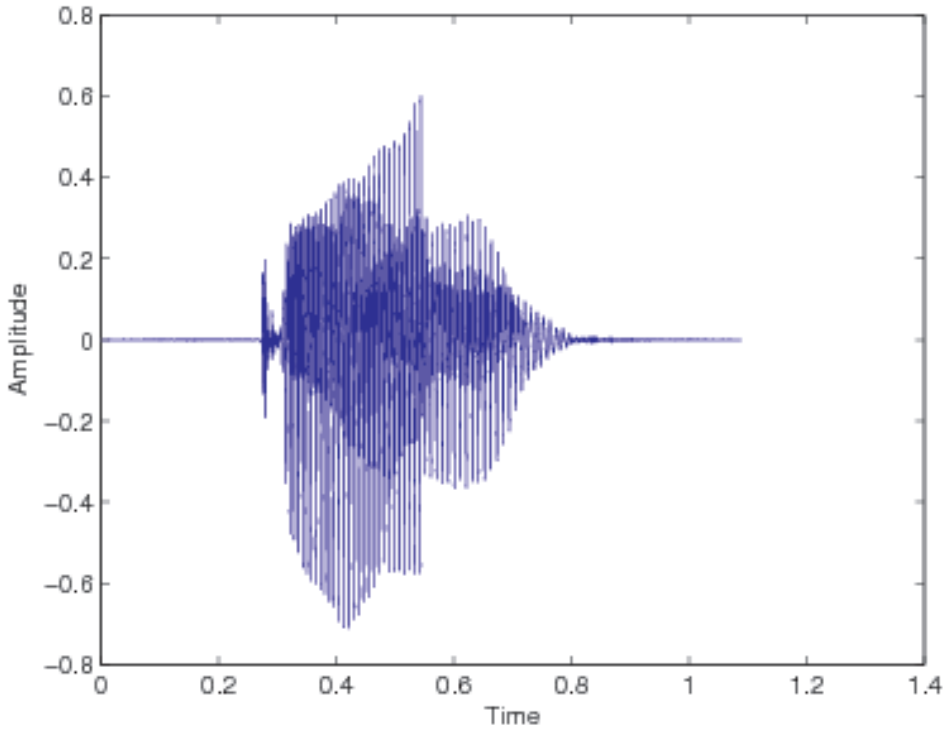


Figure 2.3: Original signal of "girl" in time domain

```
1 Speech_speaker_number=wavread('Speech_speaker_number.wav');
```

One example of the original signal of "girl" in time domain is shown in Figure 2.3:
Here we can see that at the beginning of the signal and the end of the signal silence takes a lot of space. But there is a very good boundary between silence and non-silence signal.

2.2 Pre-processing

2.2.1 Silence removal

The speech signals usually contain many areas of silence or noise. The silence signal is useless for recognition, because it contains no information. And if we keep the silence signal it will make the processing signal larger and take more time and space when getting information or features from the signal. So only the signal part that contains the actual speech segments is useful for recognition. There are many ways to do the silence removal:

- Physical way: Cutting the silence samples away and overwrites the original signal. With this method, if the speech signals or processing database have a small size, it is

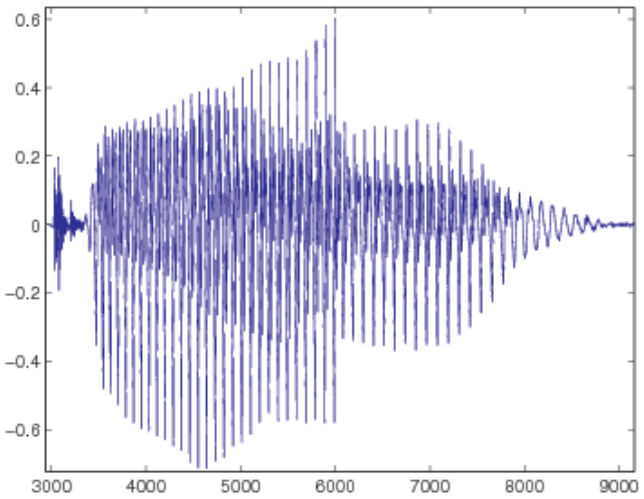


Figure 2.4: Signal after silence removal

feasible and time-saving. But for a large database this function is not a optimal choice.

- Processing way: Get some features of the signals and estimate thresholds for the features, finally according to the thresholds remove or keep the frames. The silence removal feature works by removing everything at the beginning and end of a signal that falls under a certain level or threshold. For this method, usually the signal energy and the spectral centroid are used as features [13].

In this thesis, as the training of neural network takes a lot of time, a physical or manual, method is used to do the task. In other words we have not implemented silence removal yet, as it is not so interesting, and can be done by hand quite quickly, as follows: In Matlab first read in the speech, and plot it (as in the previous figure). Then check the start sample and end sample of the signal, then cut the silence part, only keep the usable signal. Finally, rewrite the new signal to the old signal with sampling frequency 11025Hz, 24 NBITS parameters. In Matlab the code is like this:

```
1 signal=wavread('Speech_speaker_number.wav')
2 plot(signal)
3 signal=signal(start samples:end samples)
4 wavwrite(signal,11025,24,'Speech_speaker_number.wav')
```

After this procedure, the result is shown in figure Figure 2.4.

2.2.2 Normalization

Normalization is a method for adjusting the volume of audio files to a standard level, as different recording levels can cause the volume to vary greatly from word to word. The recording sound samples with different volumes and possibly some DC offset, should naturally not influence the detection system. A simple way of avoiding this is to normalize the signal in some way, e.g. scaling, and offsetting the signal so that it falls between levels -1 and 1. So a normalization is needed and can be applied before any other processing. The features that we will extract later on, e.g. the Mel-Frequency transform, depend on the power of the signal. This implies that speaking loudly will be seen differently than quietly. By normalizing the recording signal, this effect can be reduced [14].
In this thesis, the implementation of normalization is done by the `mapminmax` function:

```
1 signal=mapminmax(signal) ;
```

2.2.3 Pre-emphasis

Usually speech signal is pre-emphasized before any furhter processing. By looking at the spectrum of voiced segments we can see that the energy in the voice samples distributes more in the lower frequencies than in the higher frequencies. Thus in order to boost the amount of energy in the high frequencies, the goal of pre-emphasis is to compensate the high-frequency part that was suppressed during the sound production mechanism of humans. Speech that comes from the mouth will have a decay of 6dB per octave, a pre-emphasis filter is used to eliminate the -6dB per octave decay [15] of the spectral energy. A picture showing the signal "girl" in the frequency domain (amplitude spectrum) is shown in Figure 2.5:
As it shows that the energy in the voiced segments distributes more in the lower frequencies than in the higher frequencies. So the higher frequencies are suppressed.
Pre-emphasis is done by a first-order high-pass filter, which can be written as a difference equation (time domain):

$$EmphasisSignal(n) = Signal(n) - a * Signal(n-1)$$

Or as a transfer function (z-domain):

$$H(z) = \frac{1 - a \cdot z^{-1}}{1}$$

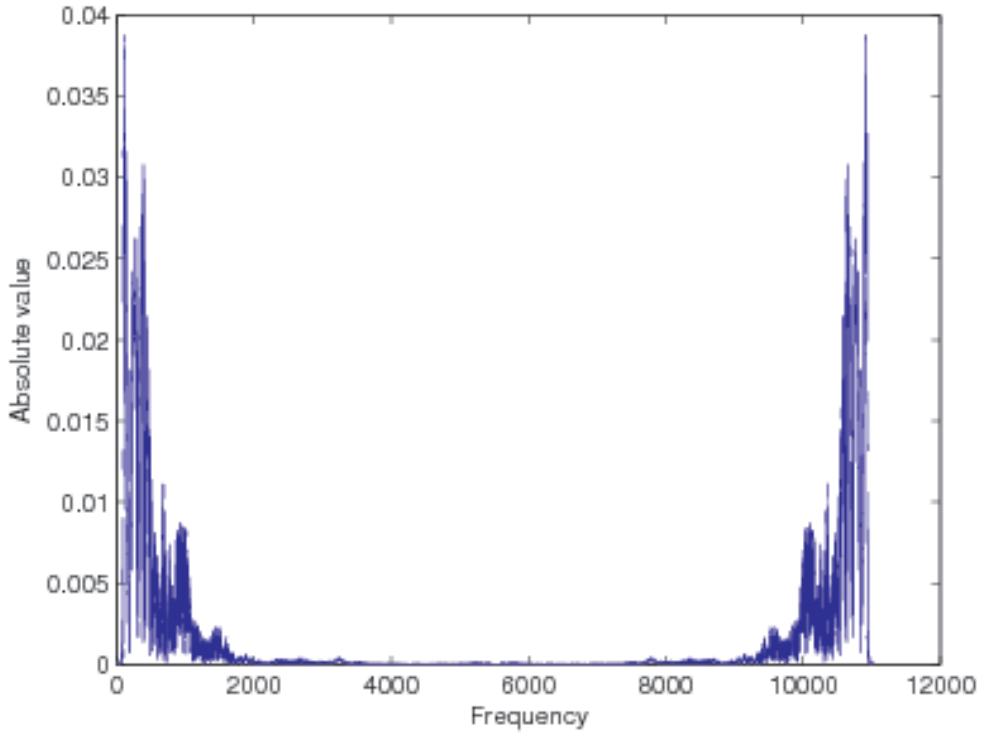


Figure 2.5: Original signal of "girl" in frequency domain

Here the value of the filter coefficient a is usually between 0.9 and 1.0, whereas in this thesis a is set to 0.9375, which is a common setting [16, 8]

In Matlab the filtering can be done as:

```
1 EmpahsisSignal=filter([1 -0.9375],1, Signal); % hp filtering
```

Where `[1 -0.9375]` is the nominator from the transfer function above, and 1 is the denominator.

Here the Signal is the target signal.

After the pre-emphasis the speech sounds sharper with a smaller volume, see Figure 2.6.

Here we can see that the high frequencies are boosted. The volume or amplitude is

$$7 * 10^{-3} = 0.007$$

It is much smaller than the original signal, which is 0.04.

So after the pre-emphasis the speech sounds sharper with a smaller volume.

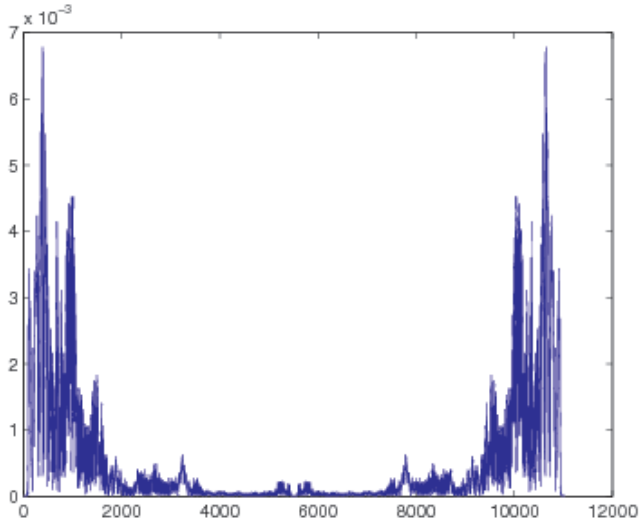


Figure 2.6: Emphasised signal in frequency domain

2.2.4 Windowing

An audio signal is an unstable signal, meaning that the statistical properties across time are not constant. However in a very short period of time the properties can be regarded as constant.

Then a short piece of signal is cut out of the whole speech signal. This is done by multiplying the speech samples with a windowing function to cut out a short segment of the speech signal. The time for which the signal is considered for processing is called a window, and the data acquired in a window is called a frame. Features are extracted once every M ms, which is called frame rate, while the window duration is N ms. Typically N is bigger than M. Thus two consecutive frames have overlapping areas. The overlapping segments are used for speech analysis. The choosing of frame length and frame shift are very important, as it can have different effect on eliminating noise as well. Normally the frame length is 256, when using a sample rate of 11025 Hz, and the frame shift is 128. But according to the previous research [17] , 3/4 frame overlap can get the best simulation result. So in this thesis the frame length is 256 and the frame shift is 64. Because the sampling frequency is 11025 kHz. So one frame is

$$256/11025 = 23.2 \text{ ms}$$

and the frame shift is

$$64/11025 = 5.8 \text{ ms}$$

As shown in Figure 2.7

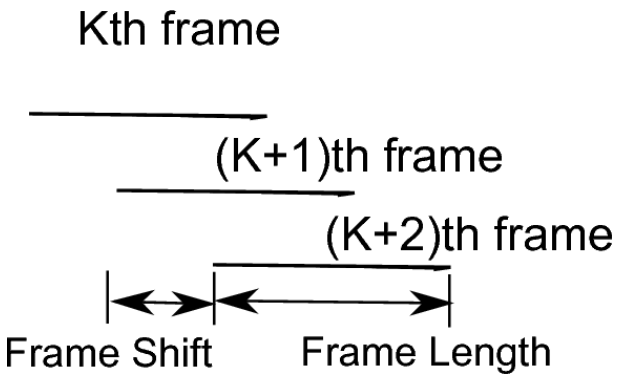


Figure 2.7: Framing

Many windowing functions can be used:

- Rectangular window
- Hann window
- Hamming window

The rectangular window (i.e., no window) is very smooth, but it can cause problems, when we do Fourier analysis; it abruptly cuts of the signal at its boundaries. It has a very high side lobe, easy to get frequency spectrum lost and missing the high frequency parts .The Hann window fades too fast and the low-pass characteristic is not smooth.A good window function has a narrow main lobe and low side lobe levels in their transfer functions, which shrinks the values of the signal toward zero at the window boundaries, avoiding discontinuities. Out of these, the most widely used window is Hamming window because of its smoothness in low-pass and very low side lobe. [18]
In this thesis:

- Frame length is 256
- Frame shift is 192
- Frame overlap is 3/4
- Hamming window is used as well because it introduces the least distortion.

The framing is shown in: Figure 2.8.
This picture shows how to do framing for the signal "girl". We are using the VoiceBox [19] toolbox for Matlab, which has a function function `enframe` that can do framing and windowing in one step:

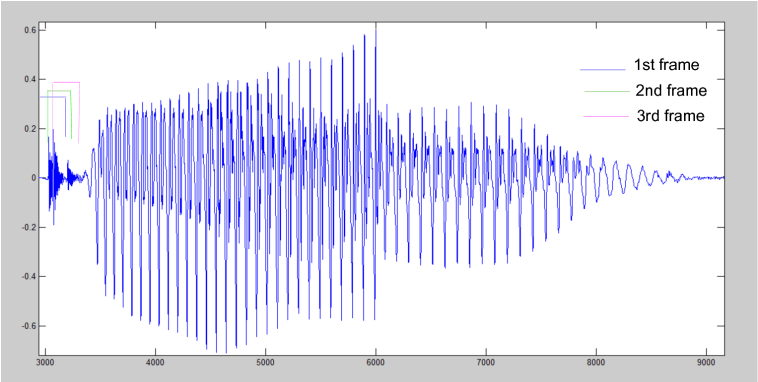


Figure 2.8: Framing signal

```
1 Frames=enframe(EmphasisSignal, hamming(FrameLength), Overlap);
```

Here the `EmphasisSignal` is the signal from the previous step, `hamming` is the windowing function, `FrameLenth` is 256, and `Overlap` is 192. The result is stored in `Frames`, which is a $N * 256$ matrix containing the frames extracted.

Chapter 3

Feature Extraction

After the pre-processing (skipping the framing even) we could have fed each 'raw' audio sample to a neural network, to let the neural network learn to recognize a speaker or speech based on that. This could work for speaker recognition at least, but this was not feasible for a number of reasons:

- Once a signal has been sampled, we have huge amounts of data every second.
- Feeding each sample to a neural network, that is one input to the neural network for each sample, which means a huge number of inputs to the NN.
- We can only capture the properties of the signal that are important for recognition instead of using all the samples.
- The spectrogram contains much of the information we need.
- By extracting the features, the same features can be used by multiple tasks.

The features are much lower in number, than the raw audio samples. They are an extremely more refined and compressed input to the neural network. So based on theoretical and practical operation, we will extract features from the frames.
The Merriam-Webster dictionary (online [20]) explains the word feature e.g. as: "a prominent part or characteristic", or by an example: 'Her eyes are her best feature". That is, we wish to extract some prominent characteristics in the frames of an audio sample of a speaker, or words (speech).
When doing speaker recognition the features need to reflect the information of every speaker as much as possible. In another way we can say that feature extraction is to transform

the input waveform into a sequence of acoustic feature vectors, each vector representing the information in a small time window of the signal. Feature extraction transforms high-dimensional input signals into lower dimensional vectors. That is, the large number of audio samples, in our case 11025 each second, is translated into a small number of features that are somehow characteristic of some speaker/speech [21].
Nowadays in speech recognition the commonly used features are:

- short-time energy
- short-time zero crossing rate
- short-time autocorrelation function
- LPC (Linear Predictive Code)
- LPCC (Linear Predictive Cepstrum Coefficients)
- LFPC (Log Frequency Power Coefficients)
- Sub-band Energy
- PLP (Perceptual Linear Prediction)
- MFCC (Mel-Frequency Cepstrum Coefficients)
- WT (Wavelet Transform)
- Frequency Spectrum Flux
- Speech Time, Formant and Pitch.

The choice of parameters is very important as it concerns the complexity of the neural network and the precision of the speech recognition. So different recognition systems normally have different features. But the standard of choosing right features are usually similar:

- The features can represent speech characteristics, including vocal tract characteristics and sense of hearing characteristics.
- Different order parameters have very good independence.
- It is convenient to calculate, the best situation is to have an efficient method of calculating.

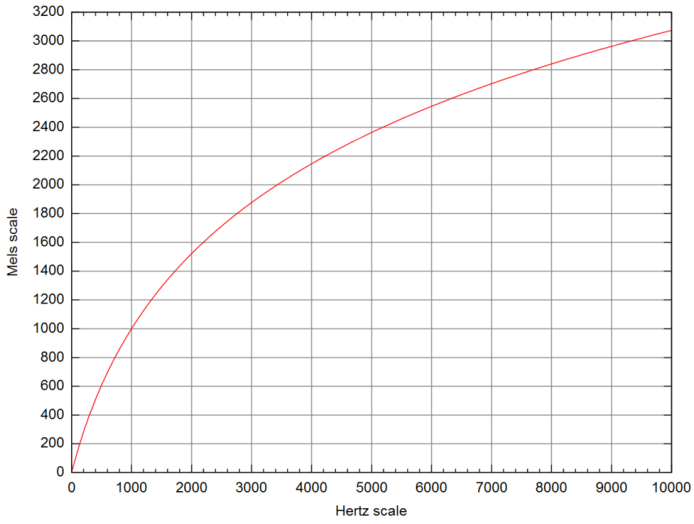


Figure 3.1: Mel-Hz scale [22]

- It should be stable .
- Easy to measure the extracted features.

Among these the most common features in speech recognition nowadays are: LPCC (Linear Predictive Cepstrum Coefficients) and MFCC (Mel-Frequency Cepstrum Coefficients) [21]. Those two methods both change the speech signal from time domain to inverted frequency parameters.

The human ear does not show a linear frequency resolution.In this thesis we use MFCCs as features, as is explained below.

3.1 Mel frequency scale

The Mel scale was named and invented by Stevens Volkman and Newman. It is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The name mel comes from the word melody to indicate that the scale is based on pitch comparisons. It is a logarithmic scale similar to the way the human ear perceives sound. Figure 3.1 illustrates the relationship between Hz and the Mel scale:

As the picture shows, when the frequency is 1000 Hz, in the mel scale it is also 1000 mel. When the frequency is below 500 Hz, the intervals are smaller compared with frequencies larger than 500 Hz.

Generally speaking the mel scale has a linear relationship with hertz when the frequency is below 1000 Hz, and a logarithmic relationship when the frequency is bigger than 1000 Hz.

Figure 3.2: MFCC Calculation [24]

There are many formulas to convert hertz into mel, but the most popular is:

$$f_{mel} = 2595 \cdot log_{10}(1 + f/700)$$

where f is the linear frequency.

3.2 MFCC

Wikipedia [23] states:

> "In sound processing, MFC which means mel frequency cepstrum, is a representation of the short term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency".

MFCC stands for Mel Frequency Cepstral Coefficients.The coefficients represent audio based on perception.They are derived from the mel frequency cepstrum. It is known that the human ear is more sensitive to higher frequency. The spectral information can then be converted to MFCC by passing the signals through band pass filters where higher frequencies are artificially boosted, and then doing an inverse Fast Fourier Transform (FFT) on it. This results in higher frequencies being more prominent. As the mel frequency cepstrum can represent a listener's response system better,MFCC is always considered to be the best available approximation of human ear.
As Figure 3.2 shows the whole MFCC processig procedure can be divided into three main steps:

- First the Fast Fourier Transform is used to convert speech signal from time domain to frequency domain.
- Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windowing. These filters follow the mel scale. In this thesis 24 filters are used. So the outputs after filtering are:

$$Signal_k, k = 1, 2, 3...24$$

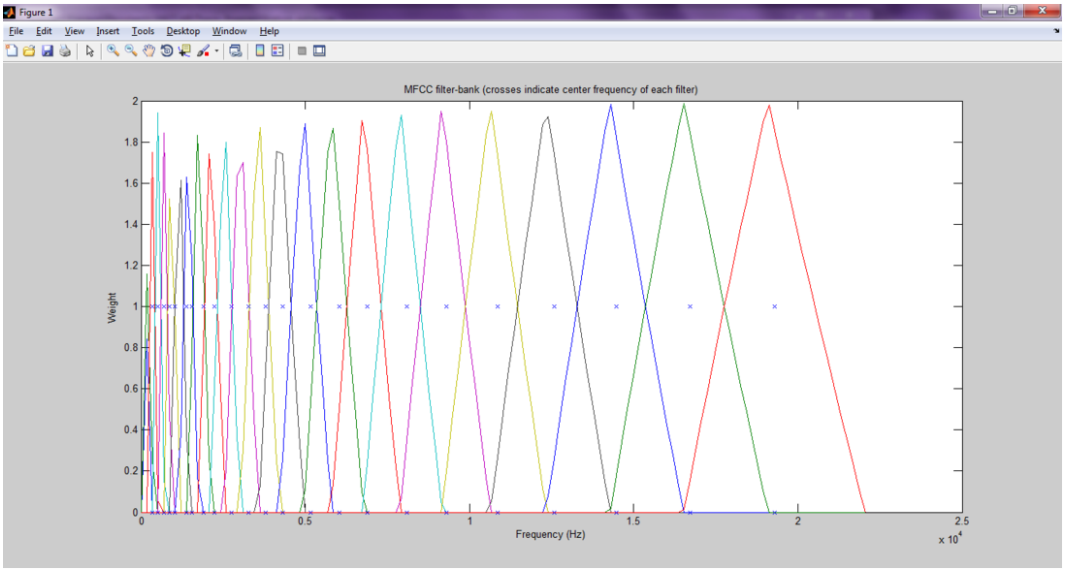


Figure 3.3: MFCC Calculation

A picture showing the 24 triangular filters :Figure 3.3 [24] Using the filterbank essentially mimics the way the human ear perceives certain frequencies.

- Finally,take the logs of the outputs, and take the discrete cosine transform(DCT) of the list of mel log powers. The MFCCs are the amplitudes of the resulting spectrum. [23, 24, 1] We use log because our ears work in decibels. Discrete cosine transform(DCT) will be applied to each Mel Spectrum to convert the values back to real values in the time domain. We take the DCT because it is good for compressing information. [24]

.

In Voicebox there is a function called `melcepst` that can calculate the mel cepstrum. Actually it can also do the framing and windowing, so the previously mentioned `enframe` function (described under pre-processing) is not even needed.

```
1 mfccs=melcepst(EmphasisSignal,11025,'M',12,24,256,192)
```

Where 11025 is the sampling frequency, 'M' means Hamming window and 12 is the number of cepstral coefficients, 24 the number of filter banks, 256 the frame length and 256-64=192 is the frame overlap. All these numbers have been discussed earlier. The result `mfccs` is a $N * 12$ matrix of 12 MFCCs per each N frame. The number N will naturally vary for each voice sample. The MFCCs are our features, that will be fed to the neural network. The NN cannot have a variable number N of inputs, so we need to do something about that, as

discussed in the next section.

3.3 Short-time energy and zero-crossing rate

The energy associated with speech is time varying in nature. Hence the intent for any automatic processing of speech is to know how the energy varying associated with short term region of speech. Short time processing method means frame by frame processing here. By the nature of production the speech signal consist of voiced, unvoiced and silence region. The short-time energy and short-time zero-crossing rate are important because they abstract valuable information about the speech signal, and they are simple to compute. The short-time energy is an indication of the amplitude of the signal. Further the energy associated with voiced region is large compared to unvoiced region and silence region will not have any energy. Thus short term energy is useful for voiced, unvoiced and silence classification of speech. During the unvoiced interval, the zero-crossing rate is relatively high compared to the zero-crossing rate in the voiced interval. Energy is measuerd, as usual, as:

$$E_n = \sum_{n=1}^{N} f[n]^2$$

Here $f[n]$ is the speech representation, i.e. the frame vectors of signal, and N is the length of the frame.

Similarly, the short-time zero-crossing rate is defined as the weighted average of the number of times the speech signal changes sign within the time window. The zero-crossing rate is defined as:

$$zcr = 1/(N-1) \sum_{n=1}^{N} \varphi \{f[n]f[n-1] < 0\}$$

Where the iindicator function φ is 1 if its argument is true and 0 otherwise.

3.4 Feature compression

In the above section the MFCC, zero-crossing rate, etc. features are calculated for each fram. But a speech signal has a strong randomness, which means that different signals may have different sizes of feature vectors. For example an utterance of the word left will have different duration, and thus different number of frames, each time. This will lead to non-static feature structures. Moreover, the feature vectors will be the inputs to the artificial neural network,

while basic artificial neural networks cannot handle these time-varying characteristics and non-static information. To this end it is necessary to merge some of the elements in the feature vector. Thus a method of compressing the different sizes of features into same size is needed.

3.4.1 K-means

K-means is one of the simplest and most popular way to cluster the vectors that contain the MFCCs to get compressed feature vectors. It is a unsupervised learning algorithm. This algorithm clusters the feature vectors based on attributes into K partitions.
It uses the k means of data generated from gaussian distributions to cluster the vectors. The main procedure is:

- Define k centroids, one for each cluster. These centroids shoud be placed in a cunning way because different locations causes different result. The better choice is to place them as far away as possible from each other.

- Take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters [25] of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. [26, 27]

- Repeat the previous step until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated. [26]

  In this step the algorithm aims at minimizing an objective function, in this case a squared error function. The objective function is:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^j - c_j||^2$$

where $||x_i^j - c_j||^2$ is a chosen distance measure between a data point and the cluster center. The result is an indicator of the distance of the n data points from their respective cluster centers. [26]

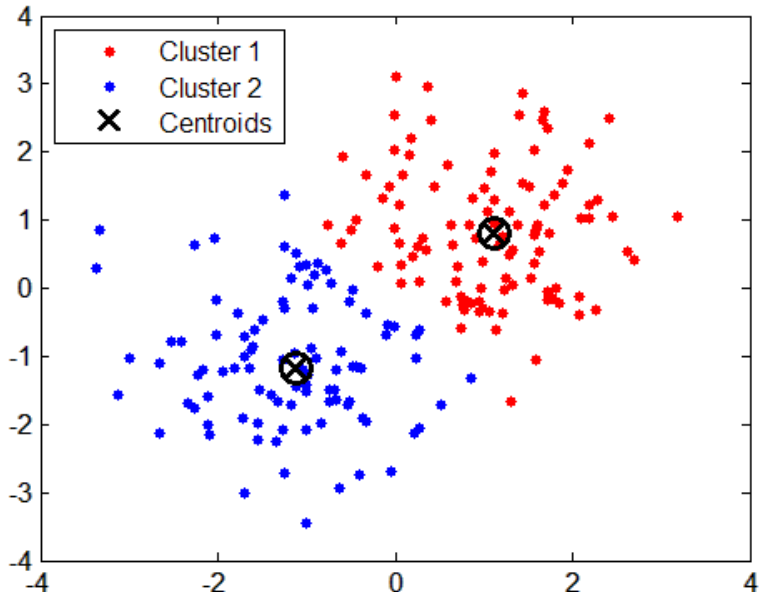


Figure 3.4: Clusters in K-means algorithm

One example of K-means cluster is shown in Figure 3.4: [28]

Although it can be proven that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. K-Means has been optimized for speed, and is suitable for very large data sets [29, 27, 26].

The k-means algorithm is implemented both as a built in function in Matlab, and in VoiceBox. We are using the one in Matlab. And a modification to the original `kmeans` function was made to order the centroids in the order of theirs clusters most frequent appearances. as follows:

```
1 features=kmeanss(mfccs,num_clusters);
```

Where `num_clusters` is a variable that we tune. The result is a `num_clusters` by 12 matrix, instead of the previous N by 12 matrix, where `num_clusters` is a known, chosen value. We further reshape the matrix into a 1 by `num_cluster`*12 vector, *which will be the input to the neural network(s)*:

```
1 features=features(:);
```

Chapter 4

Neural Network

As we all know the brain processes perception information much faster than modern computers, like visual and auditory information.
An artificial neural network or neural network is an information processing paradigm that is inspired by the way biological nervous systems, especially the brain, process information. The key elements in this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements working in unison to solve specific problems. A neural network is the same as a human brain which learns by examples. A neural network is normally trained so that a configuration for s specific application can be achieved.

4.1 Neural network history

The research of neural networks started at the end of the 19th century and the beginning of the 20th century. It came from the inter-discipline research of physics, psychology and neurophysiology. The main representative personage are: Herman Von Helmhots, Ernst Mach and Ivan Pavlov.
The development of the modern neural network started in the 1940s, by Warren McCulloch's and Walter Pitts's great work [1, 2, 3, 4]. They combine neurophysiology and symbolic logic to describe a neuron network's logical calculus. In this theory the neuron model is assumed to follow one rule,which is called " all-or-none". If this simple neuron has a big enough amount of neurons, and set of proper synapse connections and synchronous operation, McCulloch and Pitts proved that in theory, the network can calculate any computable functions. This is a significant result, which means the came of neuron network and artificial intelligent

subjects.

The second important development is found in 1949, in Donald Hebbe's book "The organization of Behavior", where he mentioned for the first time: In speech recognition many different training and recognition systems can be used: Dynamic Time Warping(DTW), Vector Quantization(VQ), Hidden Markov Model(HMM), and artificial neural network(ANN). In my thesis the back propagation neural network is utilized.

Speech recognition technology can be divided into traditional and modern speech recognition technologies. Traditional speech recognition technology is mainly done by template matching, while for the modern speech recognition technology, neural network is the main developing trend. [30, 1, 2]

As artificial neural network is modeled on the biological neural network, like the biological neural network, the ANN is an interconnection of nodes, analogous to neurons. Every neural network has three critical components: Node character, Network topology, and Learning rules. Node character determines how signals are processed by the node, such as the number of inputs and outputs associated with the node, the weight associated with each input and output, the activation function. Neural network topology determines the ways nodes are organized and connected. Learning rules determine how the weights are initialized and adjusted. [31]

4.2 Node charactersitics

Every node has multiple inputs from others via connections that have associated weights, analogous to the strength of a synapse in the brain. see Figure 4.1

The neuron has m inputs. The bias b_k can be used to adjust the output to a suitable range before the activation function. Mathematically the output of the linear combiner (summation element) is given by:

$$v_k = \sum_{j=0}^{n} w_{kj} x_j + b_k \tag{4.1}$$

the output of the neuron with the activation function applied becomes 4.1

$$y(k) = \varphi(v(k))$$

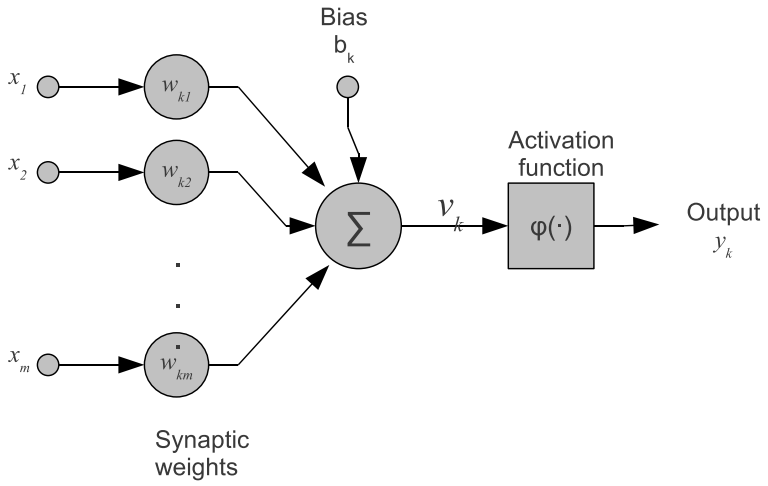


Figure 4.1: A basic model of a neuron [32]

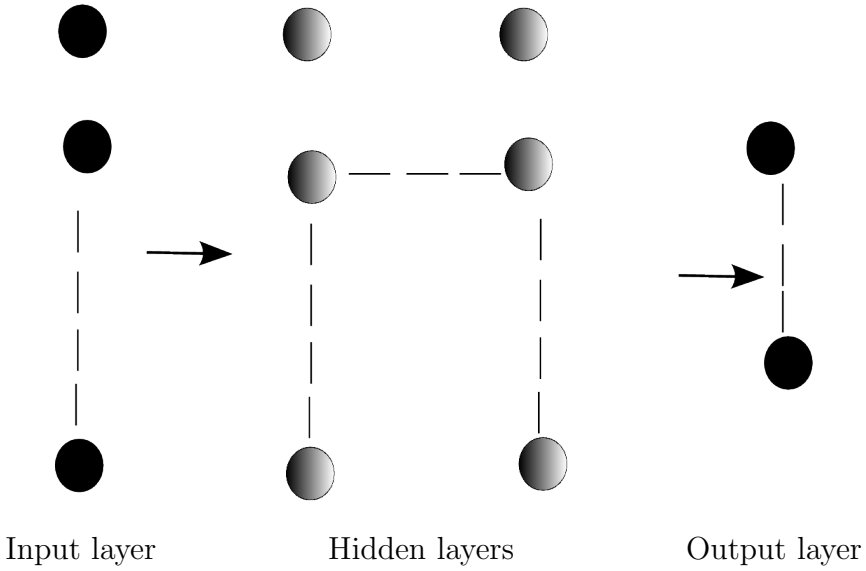


Figure 4.2: A general topology of ANN

4.3 Network topology

Usually there are the input layer, the output layer, and zero or more hidden layers. The network topology determines the number of nodes in each layers, the number of layers in the network, and the methods of connections among the nodes. see Figure 4.2

Generally, there are two types of connections between nodes. In my thesis only the *feed forward network*, which is a one way connection without loop back is used. It is static, so one input is associated with one particular output. as see Figure 4.3

It appears that the weights of a feedforward neural network are fixed after the training phase and it implanted fixed weight mapping from inputs to outputs. So the state of any neuron is solely determined by the input-output pattern and not the initial and past states of the

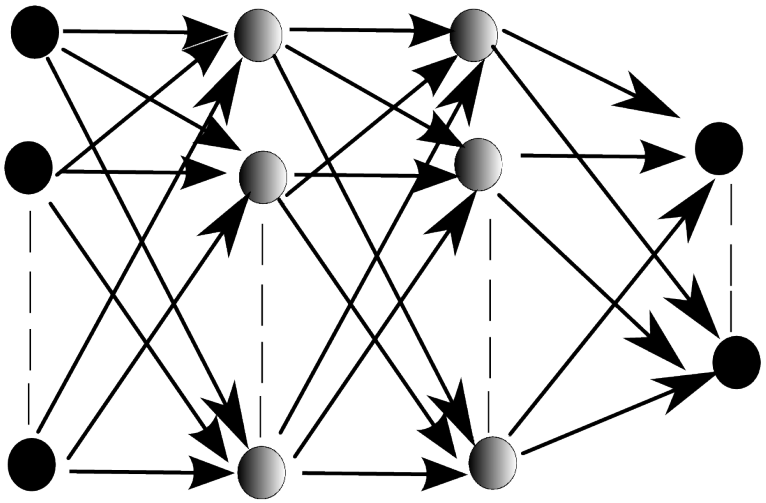


Figure 4.3: Feedforward ANN

neuron, that is, there is no dynamics involved. Consequently, this type of feedforward architecture is classified as a static neural network. The advantage of the staticfeedforward neural network is that the network can easily be built with a simple optimizing algorithm.It is the most popular neural network architecture in use today. Nonetheless, the static-feedforward neural network also has several drawbacks for some applications.

- First, it may fail to produce a satisfactory solution because the training data are insufficient in size.
- Second, the static-feedforward neural network cannot cope well with major changes that were never learned in the training phase.
- Finally, the static-feedforward neural network easily falls into a local minimum and the speed of convergence of the network can be very slow when the number of input in the data set is large.

The other type of network architecture, the *feedback network* ,has a loop back connection in which the outputs of the nodes can be the input of previous or same level nodes. It is dynamic compared with the feed forward network, so one input produces a series of outputs [31]. The Hopfiels network is one type of feedback network, see Figure 4.4
As the neurons have one or more feedback link,whose state varies with time, the feedback architecture is also called a dynamic neural network. The presence of a feedback link has a profound impact on the learning capability of the network and on its performance. Because the feedback neural networks have adjustable weights, the state of its neuron depends not

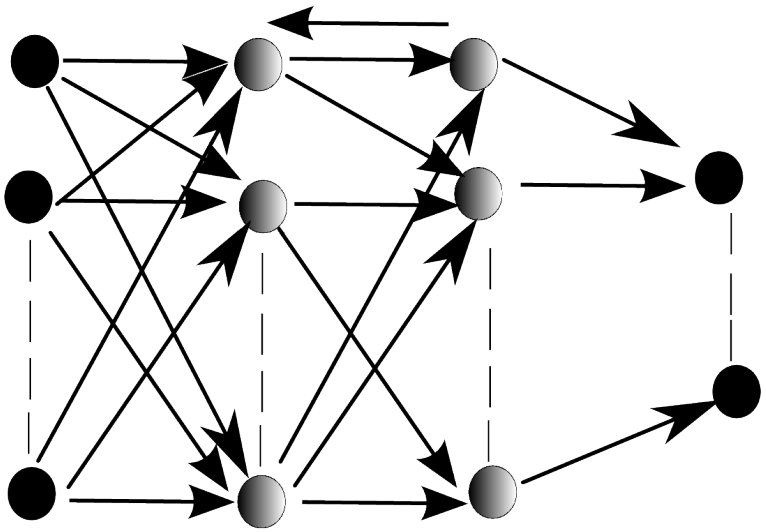


Figure 4.4: Feedback ANN

only on the current input signal, but also on the previous states of the neuron. The advantage of the dynamic-feedback neural network is that it can effectively decrease the network's input dimension and therefore the training time. However, due to the nonlinear nature of unit activation output characteristics and the weight adjustment strategies, the network stability is often difficult to ascertain.

4.3.1 The Topology Used

In this thesis a feedforward network is used. Here a description of all the components of the used network will be given.

- Inputs

  As the extracted features are compressed by k-means function, so we can get a a fixed number of features or inputs: `num_clusters` * 12 inputs. In order to get a better result different number of clusters are tested: 5,10 ,15 and 20.

- Hidden layer

  In this thesis only one hidden layer is used, but different number of nodes or neurons in hidden layer are tested: 16,32, 64 and 128.

- Activation function

  The activation function used for hidden layer is `tansig`, and for output layer is `satlins`. `tansig` is a sigmoid activation function and `satlins`is a stepwise linear

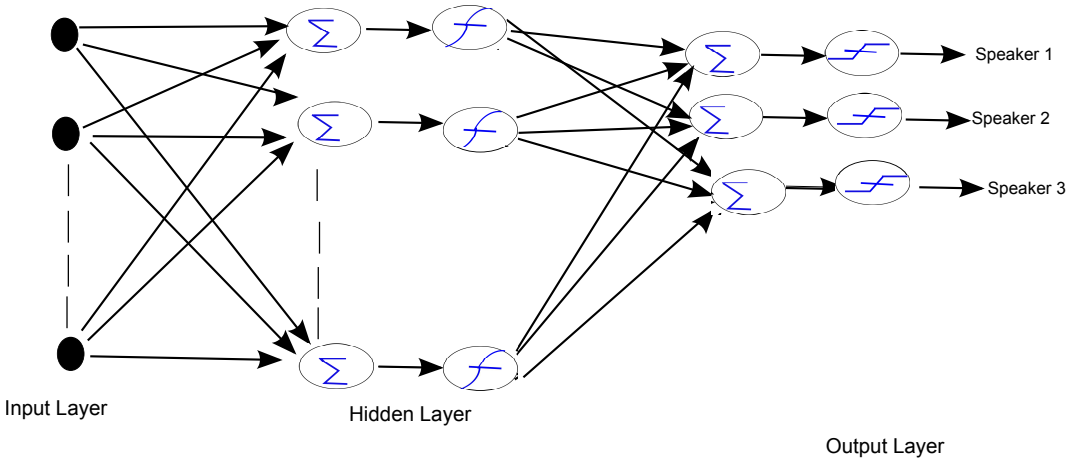


Figure 4.5: Topology of speaker recognition neural network

function.

- Output layer

  For different tasks different sizes of output layers are used. For speaker recognition, the output layer represents the speakers. There are two ways to do this task. One way is that we use one output, with a pure linear activation function. So if speaker one is detected,then the output will be 1. If speaker two is detected, the output will be 2 etc. Another way is using three binary outputs, if we have thhree speakers, which are either 1 or -1. For example is speaker two is detected, then the output will be -1, 1 and -1. There is one problem with the first method: for example if we have one linear output, and during the training process we present a voice sample of speaker 1, but the NN detects speaker 3, then the error is 3-1=2, that is a bigger number than if the NN believed it was speaker 2! There is thus a bigger penalty in confusing speaker 1 and 3 than 1 and 2. This error directs the training process which will not work properly. With three binary outputs this problem will not occur. So in this thesis *three* outputs with satlins activation function (stepwise linear) is used to correspond to every speaker.

  Figure 4.5 shows the topology of the whole neural network :

  For speech recognition, the output layer is the same as the speaker recognition. Because three words are used here.

  Figure 4.6 shows the topology of the whole neural network :

  For joint speaker and speech recognition, if one signal neural network is used, then six binary outputs will be used to represent three speakers and three words. If two neural

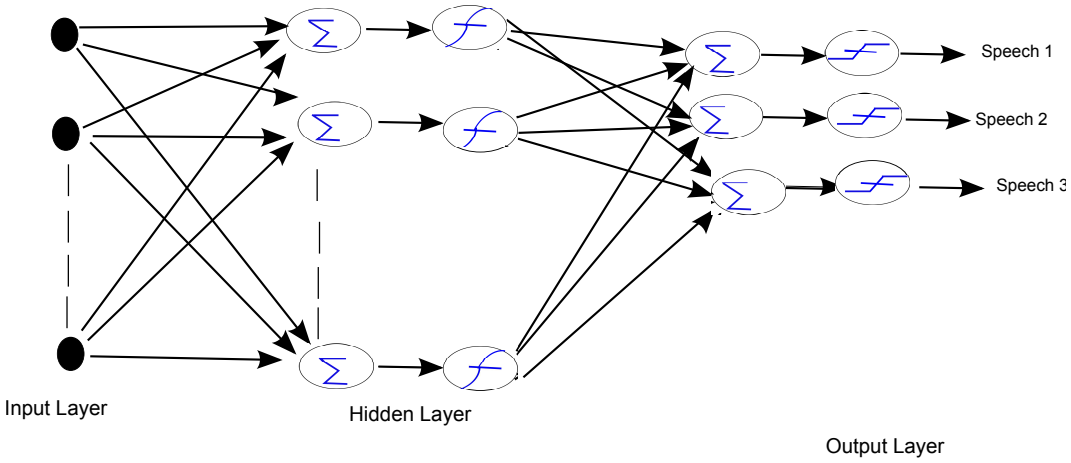


Figure 4.6: Topology of speech recognition neural network

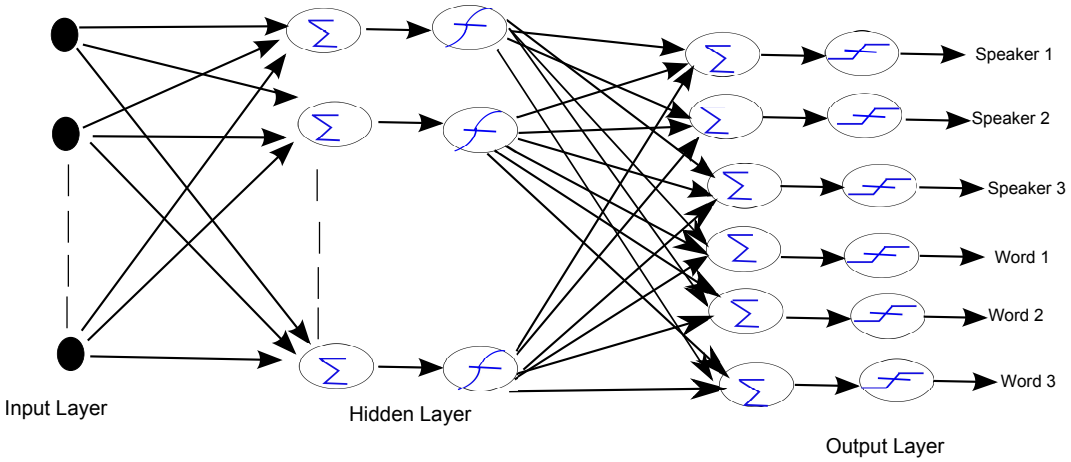


Figure 4.7: Topology of joint speaker and speech recognition neural network

network are used, then one neural network has three outputs for speaker recognition and another one has three outputs for speech recognition.

Figure 4.7 shows the topology of the whole neural network :

4.4 Learning

Learning is a process of training the network. During the training the weights are adjusted. The learning can be classified into two different categories:

- Supervised learning In supervised learning we have a teacher that gives the network desired outputs to example inputs. The weights are adjusted in order to minimize the error between the target and the network output. Figure 4.8 shows a process diagram of supervised learning : Here the target is the result that the teacher gives, the output

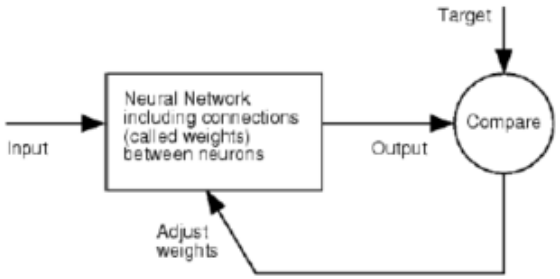


Figure 4.8: Supervised Leanring Process [32]

will compare with the target and according to the error adjust the weights.

- Unsupervised learning . In unsupervised learning there is no need to define the target output. It tries to figure out the underlying pattern or trend in the input data alone [32] [31]

4.5 Neural network data set analysis

For the operation of ANN, the data are usually divided into three subsets:

- Training phase. Before the neural network can realize a specific task, a learning process or training process must be done. During the training phase, the data is usually part of the whole database. After using all the training data once it is called a learn cycle or one epoch. The training data will be given to the neural network repeatedly until the weight values are determined.

- Validation phase. The main task doing the validation phase is to check the performance of the network and to determine the epoch at which training should be stopped in order to avoid over-fitting or over-training. Normally the best validation performance and validation check are measured when training the neural network. When the neural network is still learning or working correctly, the best validation performance normally decreases while the epoch increases. The validation checks is normally 0 when the neural network is still learning or working correctly, but will increase if the neural network is over-fitting or over-training. Then when the threshold is reached, the neural network stops. If the validation data set indicates that the network is overtrained, then the network should be retrained using a different number of neurons and/or parameter values.

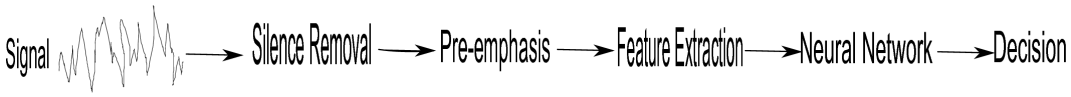


Figure 4.9: Speech recognition theory procedure flow chart

- Testing phase.

After the neural network is trained well, then a testing data set is fed to the neural network to evaluate the neural network performance. Normally the neural network uses all the data including "old data", which has been used before and "new data",which has not been used before or only use the "new data" which it has never used before as testing data set. With a different input data set, it will have different errors or errors rate. Here the inputs will be given to the neural network, and the output of the network is produced solely by input output mapping during the testing phase.

4.6 Properties of neural networks

The benefits of neural networks are that they are non-linear, adaptive and self-learning, and are good at associations and comparisons. They can be robust against interference and can easily be implemented on hardware. It can also approximate any nonlinear function, do parallel processing and be fault-tolerant. Therefore it provides a new way to solve the speech recognition which is a complex pattern classification problem [30].
Nowadays neural networks have become an interdisciplinary field involving: computer science, electrical engineering, mathematics, physics, psychology, and linguistics. For different recognition systems the requirements, the design and implementation methods are often similar. In most cases when speech recognition system uses neural network, the procedure is as shown in : Figure 4.9 [2]

Chapter 5

Neural network system implementation in Matlab

Computer simulation is a experimental method that is always used to test the design of a theory or a method's validity and operational performance. For a speech recognition system the simulation experiment follows the sequence: Capturing original speech signal, processing signal, extracting feature, training neural network, recognition processing and result output. The whole processing sequences is illustrated in Figure 5.1 and all the steps are gone through below.

Among all those procedures, there have many parameters which we can tune in order to get good result:

- Frame lengths
- Frame overlaps
- Number of MFCC's
- Cluster sizes
- Number of hidden layer neurons

In this project as the frame length and overlap are defined according to many of the previous researches results. So we can say these two are the best choices for this project. Normally people uses 12MFCC 12 Delta MFCC as the extracted features. In this project 12 MFCC and short-time energy and zero-crossing rate are used. So the parameters which are tuned during the whole project are: cluster size and number of hidden layer neurons.

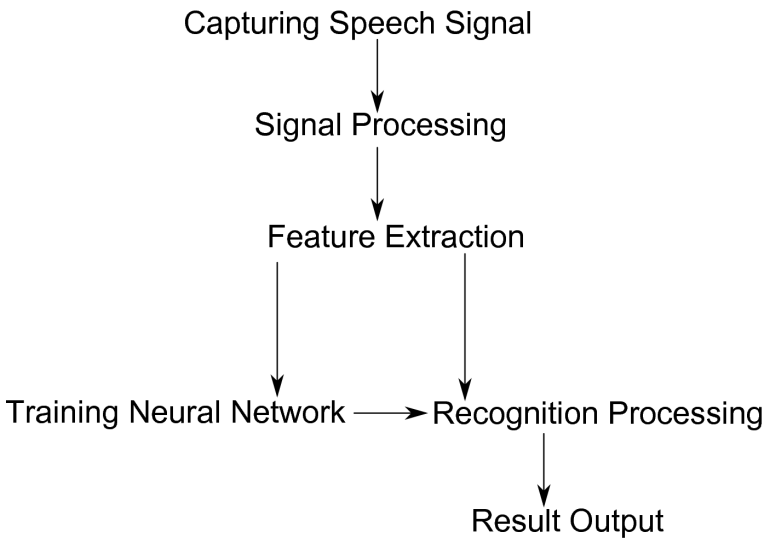


Figure 5.1: Neural Network emulation experiment sequences

First the `newff` function was used to create a feed-forward backpropagation network:

```
net = newff (p,t,[i],{'tansig','satlins'},'traingd');
```

Here p is the input matrix of the neural network and t is the target matrix. These matrices are only passed to `newff` for it to determine the dimensions of the input and output matrices. The variable i is the number of neurons in the hidden layer (we were only considering a single hidden layer). As usual sigmoid activation functions are used in the hidden layer, but in this thesis satlins transfer function is used for output layer.

Satlins is a neural transfer function. Transfer functions calculate a layer's output from its net input. and returns elements which are clipped to [-1, 1]. So that the output elements are consist of digits 1 which stand for "yes" and -1 stands for "no".

Speech signal conveys two important types of information, the primarily the speech content and on the secondary level, the speaker identity. Speech recognizers aim at extract the lexical information from the speech signal independently of the speaker by reducing the inter-speaker variability. On the other hand, speaker recognition is concerned with extracting the identity of the person speaking the utterance. So both speech recognition and speaker recognition system is possible from same voice input. In this project the same voice input is used for both speech recognition and speaker recognition.

As mentioned before the number of hidden layer neurons was tuned in this project.

There are some rules of thumb for tuning the number of hidden layer neurons for a neural

network:

- The number of hidden neurons should be less than twice the size of the input layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be between the size of the input layer and the size of the output layer. [33]

In our case the other parameter we concentrate on tuning is the cluster sizes produced by the K-means algorithm. The number of clusters determines the size of the input layer of the neural network and thus, as indicated by the rules of thumb above, affects the choice of hidden layer sizes.

5.1 Training, testing and parameter sweep

We perform a parameter sweep to try to find a suitable number of clusters and hidden layer neurons. The following numbers were iterated:

Hidden layer neurons: [16 32 64 128]
Number of clusters: [5 10 15 20]

We tested all combinations of the parameter values above, resulting in a 4x4 matrix of results. Five test scenarios were set up:

- Speech dependent speaker recognition (one word many speaker)
- Speaker dependent speech recognition (one known speaker many word)
- Speech independent speaker recognition (many speakers, many words, limited to registered words)
- Speaker independent speech recognition (many speakers, many words, limited to registered speakers and words)
- Joint speaker and speech recognition

And we ran the parameter sweeps for each scenario.

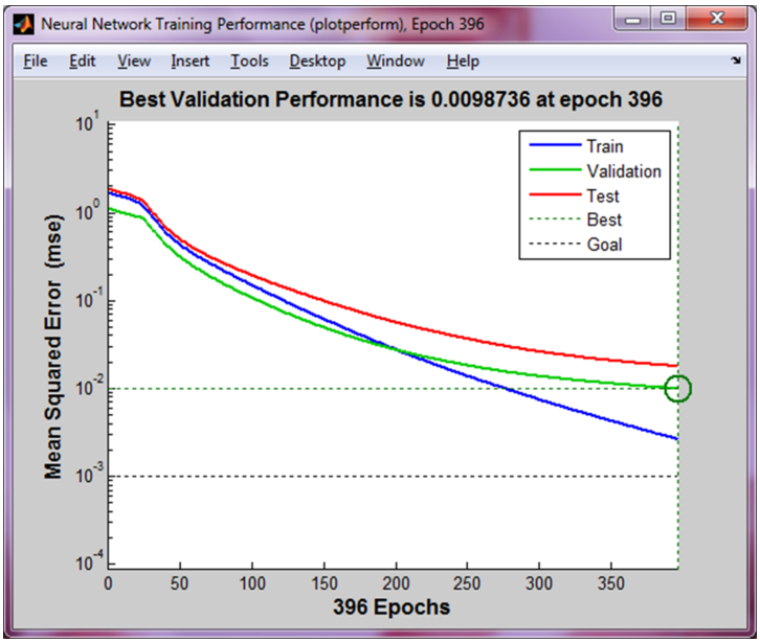


Figure 5.2: Matlab network training perforamnce plot: Speaker recognition with same word

5.1.1 Text-dependent (speech dependent) speaker recognition

In this test, 'left' is used as a voice sample to recognize who is speaking. Three speakers utter the word 'left' ten times each, giving us a training set of 30 voice samples. As per default Matlab settings, 60 percent of these samples are used for training, 20 percent for validation and 20 percent for testing.

The Matlab training performance results are shown in Figure 5.2: Here we can see that the training and validation performance are good, whereas the test set happens to fall somewhat behind. The important thing is that test and validation performance (or at least one of them) stay close to the training performance. This indicates that there is not bad over-fitting, which in turn indicates the network might be able to generalize to new test samples ,which it has not seen before. The training stopped in this case when the performance goal was met (training error 10^{-3}).A better result can still be reached just by lowering the goal. The resulting error rates are calculated both for the full data set of 30 sampels, as well as for the validation and test samples only. Testing with training and validation data only is a tougher test, since those voice samples where not used for training the network, i.e. tuning the weights. We thus test the generalization capabilities of the network. The results with all samples are shown in Table 5.1 and with only test and validation samples in Table 5.2. From the result we can see that for speaker recognition when only one word has been uttered by the speakers, the system works perfectly with no errors at all, no matter what cluster

Table 5.1: Error rate with all the samples

Hidden Layers	16	32	64	128
Cluster 5	0	0	0	0
Cluster 10	0	0	0	0
Cluster 15	0	0	0	0
Cluster 20	0	0	0	0

Table 5.2: Error rate result for the test and validation dataset

Hidden Layers	16	32	64	128
Cluster 5	0	0	0	0
Cluster 10	0	0	0	0
Cluster 15	0	0	0	0
Cluster 20	0	0	0	0

and hidden neuron structures were chosen.

5.1.2 Speaker dependent speech recognition

In this procedure the samples from speaker xxg: left, right and froward, were used for testing speaker dependent speech recognition, i.e. what has been said, i.e. a vocabulary size of four. The Matlab training performance results are shown in Figure 5.4:
Here we can see that the training performance is good, whereas the validation and test sets fall somewhat behind, which is just bad luck. Generally this result is not so good compared with text dependent speaker recognition.
The result for all the data set of speech recognition for one speaker is shown in Table 5.8.

Table 5.3: Error rate with all the data set

Hidden Layers	16	32	64	128
Cluster 5	2 2	9	3	16
Cluster 10	2	3	3	2
Cluster 15	3	1	2	7
Cluster 20	2	2	16	2

Evidently, and not surprisingly, the speech detection is more challenging than the speaker recognition.

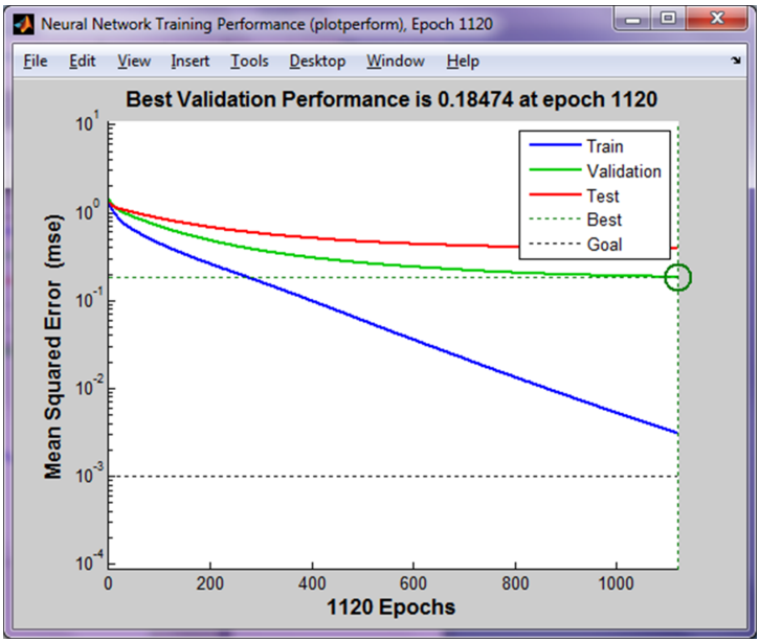


Figure 5.3: Matlab network training perforamnce plot: Speech recognition

From the parameter sweep we can see that there is no clear trend indicating the best combination of hidden layer size and number of clusters, but one suitable choice could be 5 clusters and 16 neurons in the hidden layer. Note that both the k-means algorithm, and the backpropagation algorithm are nondeterministic and give different results each time. Thus the numbers in this table will also change each time the test is run.
We pick the combination 5 clusters and 16 neurons and investigate the performance difference for complete data set and validation/test data samples. Two testruns give the error rates 2.2% and 1.1% for the complete data set, and 6.7% and 3.3% for the test/validation data. Two conclusions can now be made:

- The error rate for the whole data set is smaller than only test/validation data. This means that the neural network performs worse with the samples that it has never seen before. This is completely expected.

- For different sizes of the hidden layers and clusters there are naturally different results, but it is not quite clear which combination is the best. But generally speaking, a hidden layer size of 16 or 64 neurons seems good, while the cluster sizes do not matter so much.

The Matlab training performance results are shown in Figure 5.4:
The training performance plot already indicated pretty poor generalization capabilities of

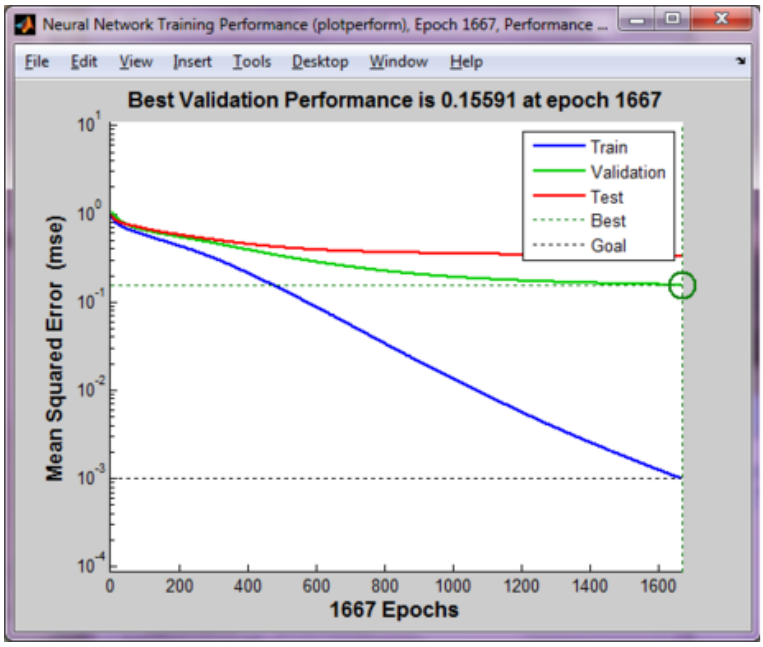


Figure 5.4: Matlab network training perforamnce plot: Speech recognition with 5 cluster and 16 neurons

our network. The mean square error of the training data set was relatively good, while the test and validation errors were higher. Poor generalization can be due to over-fitting, which is caused by a too powerful network (too many nodes in the hidden layer) or over-training a network. In our case the problem is not over-fitting. This we conclude since the problem is not only the gap between the blue line and the red and green in the plots, but the problem is that the green and red lines just never really decrease significantly. The network just does not see enough resemble between voice samples that it has not heard before and the ones used for training. There is such a big difference between each sample.

One remedy could be to present more training samples to the network. Then it might learn a wider range of voice samples that should be mapped to a certain word. A larger training set also prevents over-fitting, as it is just harder for the network to be over-fitted to a large number of different samples.

We test this by increasing the number of training samples from 10 per word to 20. Our small vocabulary size of three words then gives us 60 training samples.

The resulting error rates for two test runs (5 clusters and 16 nodes in the hidden layer) then become 1.1% and 0.6% for the complete data set, and 3.7% and 1.8% for the training and validation data sets. Both error rates drop. We would expect the training error rate to increase, as the possibility for the network to be (overly) fit for those samples decreases.

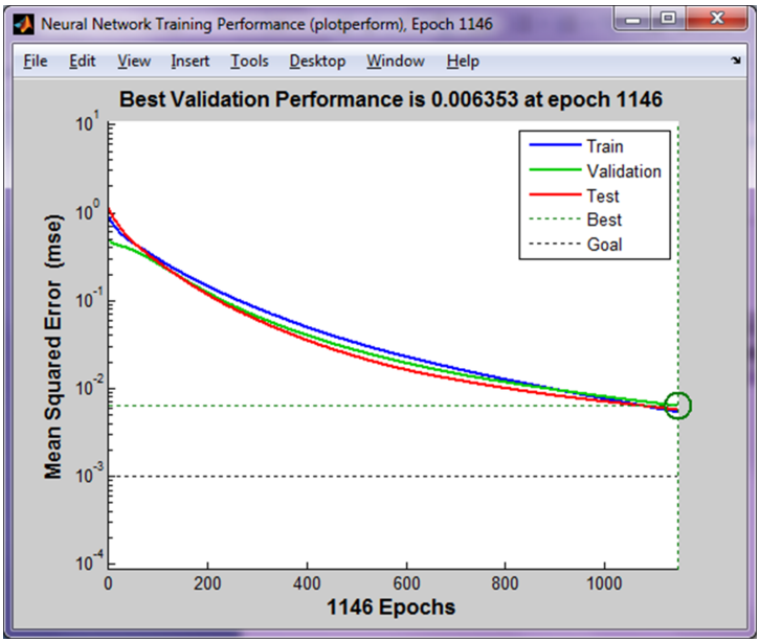


Figure 5.5: Speech independent speaker recognition

The test and validation data samples should be identified with better success. We remember that the complete data set also contains the test and validation sets. The decrease in error for the full data set is likely to be due to the decrease in the test/validation subsets.

5.1.3 Speech independent speaker recognition

In this case one neural network with 3 outputs to indicate three different speakers no matter which registered speeches are given (note that we do not test with previously unheard words, that would be completely speech independent).

The training performance from Matlab is plotted in Figure 5.5: The plot shows that the training performance reached the performance goal. And the training stopped. But the validation and test results were not so good because of poor generalization capabilities of network. Measured error rates for a parameter sweep are given in Table 5.8.

Table 5.4: Error rate with all the data set

Hidden Layers	16	32	64	128
Cluster 5	0	0	0	0
Cluster 10	0	0	0	0
Cluster 15	0	0	0	0
Cluster 20	0	0	0	6

Table 5.5: Error rate for test/validation data set

Hidden Layers	16	32	64	128
Cluster 5	0	0	0	0
Cluster 10	0	0	0	0
Cluster 15	0	0	0	0
Cluster 20	0	0	0	8

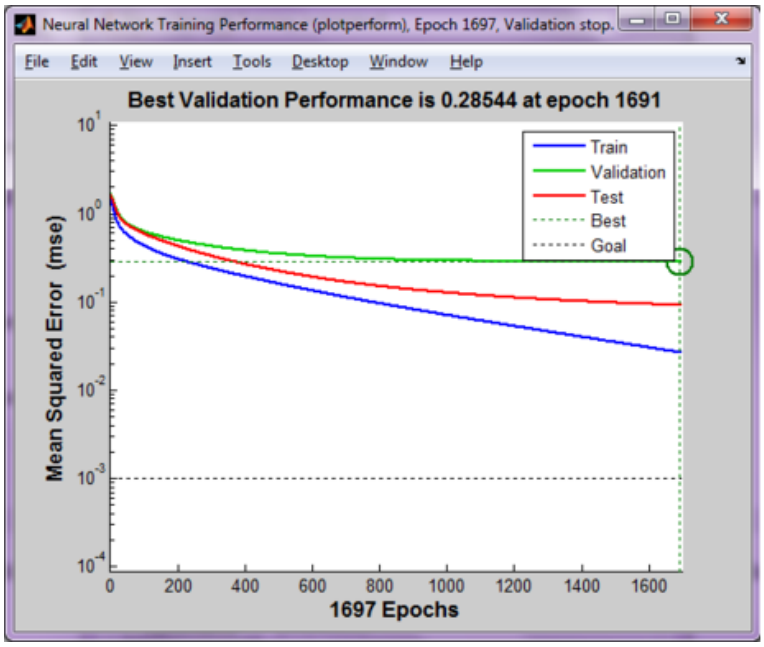


Figure 5.6: Speaker independent speech recognition

From these two tables we can see that even with independent speeches, the processing result of speaker recognition was almost with a error zero apart from one error with 11.9048. This was only because of bad luck. So in general the speaker recognition worked very well.

5.1.4 Speaker independet speech recognition

In this test case, three speakers and three words were used as samples. Each speaker has spoken each word 10 times, giving us a training data set of $3 \cdot 3 \cdot 10 = 90$ voice samples. This not really completely speaker independent, as all speakers are present in the training set. All speakers are thus registered and known, and we do not test with unknown speakers. Training performance from Matlab is plotted in Figure 5.6:
Again there are pretty high error rates on the test and validation sets, indicating poor generalization capabilities.

Measured error rates for a parameter sweep are given in Table 5.8.

Table 5.6: Error rate with all the data set

Hidden Layers	16	32	64	128
Cluster 5	1	1	2	2
Cluster 10	2	2	2	1
Cluster 15	2	1	2	1
Cluster 20	1	2	18	1

The error rate for the test and validation data sets is shown in Table 5.9.

Table 5.7: Error rate for test/validation data set

Hidden Layers	16	32	64	128
Cluster 5	3	3	5	5
Cluster 10	7	6	5	3
Cluster 15	7	5	5	3
Cluster 20	4	6	17	2

By analysing the results of error rate for all the data set and test/validation data set, one can see that:

- For joint speech and speaker recognition, the results are comparable to the speaker dependent speech recognition when the training data set is included in the test, but generalization capabilities are in fact poorer, which Table 5.9 shows.
- A higher number of neurons in the hidden layer seems to be beneficial.

What we are actually testing here is speaker independent speech recognition and text independent speaker recognition. This should be much harder than the text/speaker dependent cases. On the other hand we now have more samples for each word. Although spoken by three different speakers, this could help.
The error rates are in any case not satisfactory.

5.1.5 Joint speaker and speech recognition

As mentioned before, in this case two different network can be used. One is using two neural networks to realise speaker independent speech recognition and speech independent speaker

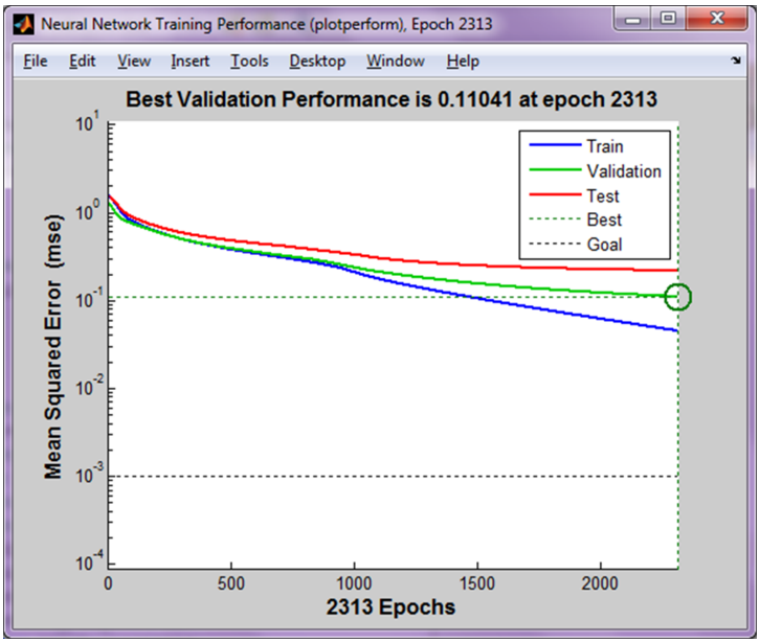


Figure 5.7: Joint speaker and speech recognition using one neural network

recognition seperately. This method is the combination of the previous sections. Another one is using one neural network to do those two tasks at the same time. For one neural network solution, it has six outouts in which three are used for recognizing speaker and another three for recognizing speech.

Now the result of using one neural network is shown in Figure 5.7:

Here we can see the plot result is good. The validation performance performance is very close to trainning performance.

Measured error rates for a parameter sweep are given in Table 5.8.

Table 5.8: Error rate with all the data set

Hidden Layers	16	32	64	128
Cluster 5	5	6	8	1
Cluster 10	2	3	5	1
Cluster 15	7	2	1	1
Cluster 20	1	2	7	2

For this case, the training time was extreme long. But the results was good. Compare the one neural network solution with the two neural networks solution one can see that:

- By using two neural networks, one can see clearly that speaker recognition worked perfectly with dependent and independent speech. While for speech recognition the

Table 5.9: Error rate for test/validation data set

Hidden Layers	16	32	64	128
Cluster 5	12	9	10	3
Cluster 10	5	10	8	3
Cluster 15	10	8	5	4
Cluster 20	5	6	10	6

system does not worke so well.

- By using one neural network, it took more time.
- As the results differed every time, so it was quite hard to say whcih solution is better. But two neural networks made it easier to analyse the results.
- With different neural network hidden layer neurons and clusters, the results differed a lot.

5.1.6 Scalability

In this section, a series of tests with 2,3,4,5 and 6 words, with 20 samples of each word, will be run. This we hope will show how the performance is degraded with an increased vocabulary size. Hopefully it will degrade gracefully, i.e. the method will be scalable for larger vocabulary sizes. Will will do this thoroughly with a parameter sweep for each vocabulary size, in order to also see if an increased vocabulary size will require more neurons in the hidden layer, for instance. Here first a two words recognition was done, right and left were chosen to be the words. The result is shown in Table 5.10.

Table 5.10: Error rate for two words recognition of all the data set

Hidden Layers	16	32	64	128
Cluster 5	0	1	0	1
Cluster 10	0	0	1	0
Cluster 15	0	0	1	0
Cluster 20	0	0	0	1

For three words test, right, left and forward are used to do the recognition. Here comes the result: Table 5.12. Table 5.13.

Table 5.11: Error rate for two words recognition of test data set

Hidden Layers	16	32	64	128
Cluster 5	0	5	1	3
Cluster 10	1	1	3	0
Cluster 15	0	3	3	0
Cluster 20	1	0	0	5

Table 5.12: Error rate for three words recognition of all the data set

Hidden Layers	16	32	64	128
Cluster 5	1	6	1	0
Cluster 10	0	0	0	1
Cluster 15	1	1	3	0
Cluster 20	1	0	7	0

Table 5.13: Error rate for three words recognition of test data set

Hidden Layers	16	32	64	128
Cluster 5	3	10	4	2
Cluster 10	1	1	1	3
Cluster 15	3	4	7	1
Cluster 20	3	2	8	1

For four words test, right, left, forward and run are used to do the recognition: Table 5.14. Table 5.15.

Table 5.14: Error rate for four words recognition of all the data set

Hidden Layers	16	32	64	128
Cluster 5	0	0	5	0
Cluster 10	5	1	1	0
Cluster 15	1	1	0	0
Cluster 20	1	0	5	0

Table 5.15: Error rate for four words recognition of test data set

Hidden Layers	16	32	64	128
Cluster 5	1	2	5	0
Cluster 10	10	3	3	2
Cluster 15	5	4	1	0
Cluster 20	4	0	6	0

For five words test, right, left, forward, run and stop are used to do the recognition: Table 5.16. Table 5.17.

Table 5.16: Error rate for five words recognition of all the data set

Hidden Layers	16	32	64	128
Cluster 5	2	0	0	0
Cluster 10	4	0	0	1
Cluster 15	0	1	0	0
Cluster 20	1	0	0	0

Finally a test with all the six words were done: Table 5.18 Table 5.19.

A plot of the result of different vacabularies results are also made in Figure 5.8, in order to have a easy comparison. Here only the error rate of all the data set result is considerer. From here we can see that generally, with hidden layer 32 and cluster 20 a better result is obtained. So a comparison of different vacabulary sizes is done in Figure 5.9:

From here we can see clearly that the error rate is not really growin much (at all according to these measurements) with growing vocabulary size. And in general the error rate is also quite low(maximum 2

Table 5.17: Error rate for five words recognition of test data set

Hidden Layers	16	32	64	128
Cluster 5	5	2	2	2
Cluster 10	6	1	2	4
Cluster 15	0	2	2	1
Cluster 20	4	2	2	2

Table 5.18: Error rate for six words recognition of all the data set

Hidden Layers	16	32	64	128
Cluster 5	4	2	2	1
Cluster 10	5	3	1	1
Cluster 15	1	1	1	1
Cluster 20	2	0	0	0

Table 5.19: Error rate for six words recognition of test data set

Hidden Layers	16	32	64	128
Cluster 5	7	4	5	2
Cluster 10	9	7	4	4
Cluster 15	3	4	3	4
Cluster 20	6	2	3	1

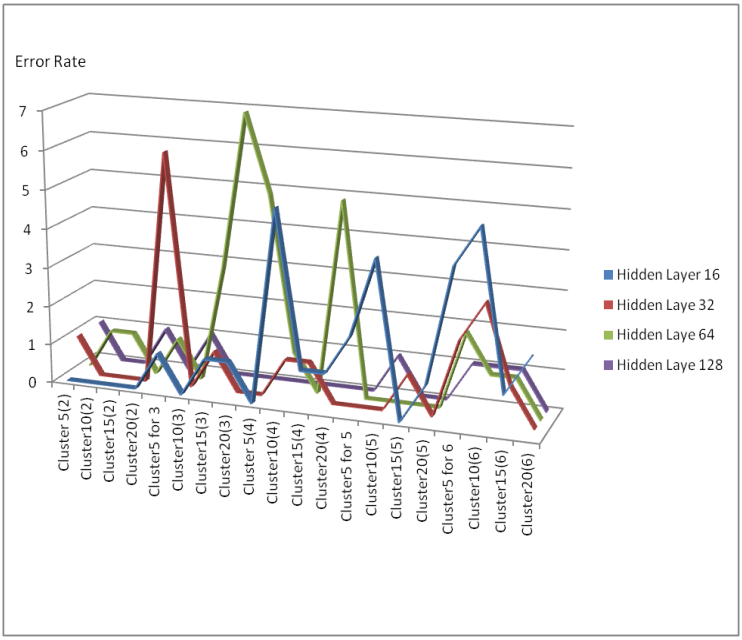


Figure 5.8: Scalability result show

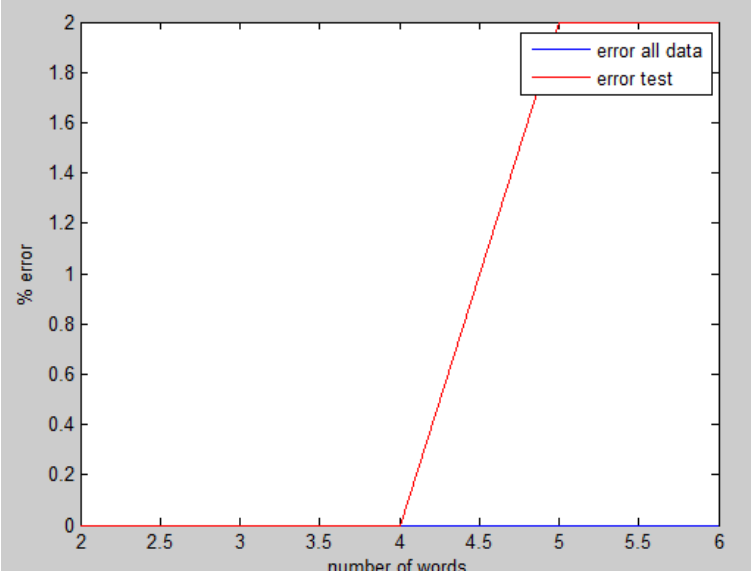


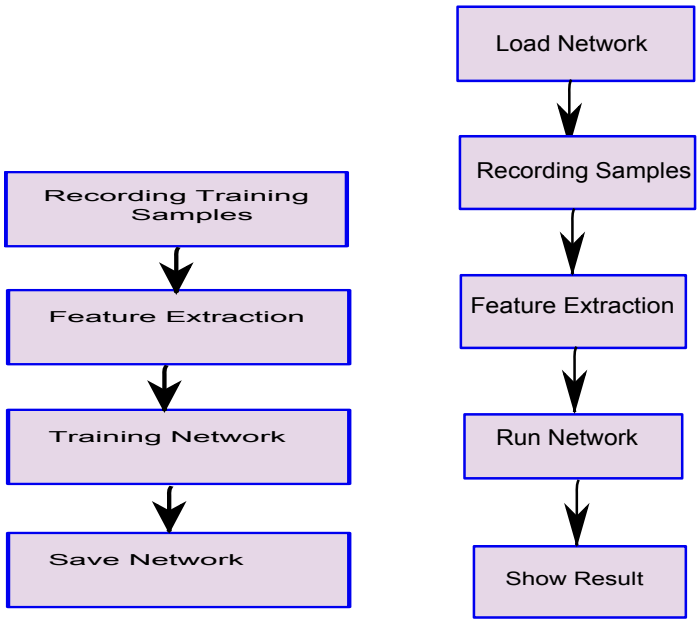
Figure 5.9: Error percentage with growing vocabulary size

5.1.7 Putting it all together, using the Network Online

In this section, we show how the trained network is used, and we present the full Matlab code for running the network "online". An utterance of a (known) word is recorded, and the recongized word is reported. A trained neural network `net`, using paramters found to be optimal in the tests above, has been stored to disk beforehand. the parameter sweep a online test is implemented, which means to recognize the speech online after a sample is given or recorded. Here the test is speaker independent speech recognition. Then online speech recording is done, and played in order to make sure the speech is recorded. Finally this recorded file is analysed the same way as the other voice samples. Only a feature vector is now fed to the neural network, representing one utterance, instead of a feature matrix containing many samples, as was done when the performance measurements were done. From the output vector we can read which word was recognized, and the recognized word is also reported in a popup message box. Here shows the training network processing chain in Figure 5.10:

Below we present a Matlab script used for the "online" recognition:

```
1 %%Load the net from the saved net file
2 load net;
3
4 %%Record  and play  the utterance
5
```



a) b)

Figure 5.10: Training network processing chain a) and Using the trained network b)

```
fprintf('Press any key to start %g seconds of recording...\n',duration);
pause;
fprintf('Recording...\n');
voice=wavrecord(duration*Fs,Fs);
signal=vadsohn(voice, Fs, 'a',pp); % vadsohn implements a voice activity detector
ind=find(signal); % find index of nonzero elements
cut_signal=voice(ind);
voice=cut_signal; % get the non-silence signal

fprintf('Finished recording.\n');
fprintf('Press any key to play the recording...\n');
pause;
wavplay(voice,Fs);

%% Feature extraction.

EmphasisSignal=filter([1 -0.9375],1, voice); % hp filtering
Framing=enframe(EmphasisSignal,hamming(FrameLength),Overlap);
Amplitude=sum(abs(Framing),2);
Energy=sum(Framing.*Framing,2);
 temp1=enframe(voice(1:end-1),FrameLength,Overlap);
 temp2=enframe(voice(2:end),FrameLength,Overlap);
 signs=(temp1.*temp2)<0;
 diffs=(temp1-temp2)>0.02;
Zcr=sum(signs.*diffs,2);
V=melcepst(EmphasisSignal,11025,'M',12,24,256,192);

[Comenergy]=kmeanss(Energy,num_clusters);
```

```
[Comzcr]=kmeanss(Zcr,num_clusters);
[V]=kmeanss(V,num_clusters);
feature=[V,Comenergy,Comzcr];
feature=feature(:);
V=feature;

%% Networking sim and report the recognized result.
y=sim(net,V);
[val,index]=max(y);
words={'left','right','forward','run','stop','turn'};
result=words(index);
msgbox(result,'Result');
```

Chapter 6

Conclusion

6.1 Conclusion

Speech recognition using Neural Networks are a hotspot of international academic circles . This thesis uses speech recognition as the research background, to research the whole recognition process system. To sum up, the main work and result of this thesis are:

- Research the theory of speech recognition, including its history, developing trend , signal processing, feature extraction and compression. MFCCs were used as the features,different clusters were tested to get the best cluster number which was chosen for the final use. This gives a detailed information,not only a theoretical one on how to do the whole process ,but also gives the specific functions for practical execution.

- Research about neural networks, from its developing history, elements and theory of recognition. The Backpropagation neural network was used in the recognition system as the recognition method, different numbers of hidden layer nodes were tested and the best hidden layer was chosen for the final use. The result shows that BP neural network works well when being used in speech recognition system.

- All the processes and procedures were implemented in Matlab. From the experiment level to testing the pre-emphasis, framing,windowing, feature extraction, compression and recognition, and then the performance analysis and result description. The presentation of errors with all the test samples and test/validation data only gave a more accurate and objective result.

6.2 Prospect

Even though this thesis has got some results, there are still many problems that need to be further investigated. The prospects for future work are:

- All the signals that were used in this thesis are read and recorded in very good condition, so they can be regarded as very pure signals. This will lead to a difference between experiment and practical application. In practice noise will make a difference for the signal features. This will influence the result of recognition, so a research about features or algorithms that are immune to noise will have a significant meaning for practical application.

- Features are very important for the recognition system. Future research about the nonlinearity of the speech is needed, so that better features can be used to get a better result.

- Utilizes this technology together with the computer science technology and speech recognition software and hardware system in practical application.

Bibliography

[1] Rong Rong. Speech recognition using artifical neural networks. master thesis, 2005.

[2] Wu Wei-ye. Arithmetic research based on neural network speeche recognition. Master thesis, 2009.

[3] Zou Da-yong. The research of the system of speech enhancement based on bp neural network. master thesis, 2006.

[4] Wang Wei-Zhen. Research of speech recognition based on neural network. master thesis, 2008.

[5] He Xiang-Zhi. Speech recognition researching and development. *Computer and Modernization*, 3, 2002.

[6] Bill gates appointed zhang yaqin as microsoft's global vice president. "`http://english.peopledaily.com.cn/200401/09/eng20040109_132249.shtml`", 2012.

[7] Speaker recognition. "`http://www.busim.ee.boun.edu.tr/~speech/Speaker_recognition.html`", 2012.

[8] Wen Lin. Based on retrofited mfcc speech recognition system research and design. master thesis, 2011.

[9] Amit-Degada. Digital coding of analog signal. lecture notes, Sardar Vallabhbhai National Institute of Technology.

[10] Thomas-Zawistowski and Paras-Shah. An introduction to sampling theory. "`http://www2.egr.uh.edu/~glover/applets/Sampling/Sampling.html`", 2012.

[11] Voice frequency. "`http://en.wikipedia.org/wiki/Voice_frequency`", 2012.

[12] Nyquist & shannon sampling theorem. "http://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem", 2012.

[13] Theodoros Giannakopoulos. A method for silence removal and segementation of speed signals,implemented in matlab. report, University of Athens, Greece and NCSR DEMOKRITOS, Greece.

[14] Scott Chin, Kelvin Lau, and Lindsey Leu. A speaker verification system. report, Department of Electrical and Computer Engineering ELEC499a, 2002.

[15] B. Plannerer. An introduction to speech recognition. tutorial, University of Munich, 2005.

[16] Li hui Fu. The key technology in speech recogn ition. *1994-2010 China Academic Journal Electronic Puslishing House*, 117, 2010.

[17] Yang Guo rong and Jin Li jun. The research of speech signal enhancement algorithm based on matlab simulation. *Science Technology and Engineering*, 10, 2010.

[18] Han Yi, Wang Guo yin, and Yang Yong. Speech emotion recognition based on mfcc. *Journal of Chongqing University of Posts and Telecommunications*, 20, 2008.

[19] Voicebox: Speech processing toolbox for matlab. "http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html".

[20] English dictionary. "http://www.merriam-webster.com/dictionary/feature", 2012.

[21] Liu bo xia and Chen Jian feng. Environment acoustic event recognition algorithm based on feature analysis. *Computer engineering*, 37, 2011.

[22] Mel-scale. "http://en.wikipedia.org/wiki/Mel_scale", 2012.

[23] Mel-frequency cepstrum. "http://en.wikipedia.org/wiki/Mel-frequency_cepstrum", 2012.

[24] Aldebaro-Klautau. The mfcc. Notes, 2005.

[25] Center of mass. "http://en.wikipedia.org/wiki/Center_of_mass", 2012.

[26] A tutorial on clustering algorithms. "http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html", 2012.

[27] Kiri Wagstaff and Seth Rogers. Constrained k-means clustering with background knowledge. 2001.

[28] Kmeans produc document. "http://www.mathworks.se/help/toolbox/stats/kmeans.html", 2012.

[29] A tutorial on clustering algorithms. "http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html", 2012.

[30] Sun Ning, Sun Jin guang, and Sun Yu. Research of speech technology based on neural network. *Computer and Digital Engineering*, 34, 2006.

[31] Jin ming zou, Yi Han, and Sung Sau Sa. *Overview of Artificial Neural Networks*. UK, 2009.

[32] Dag Björklund. Intelligent systems. lecture notes, Novia University of Applied Science, 2011.

[33] The number of hidden layers. "http://www.heatonresearch.com/node/707", 2012.