

Preface

This work is a report of the thesis work, to get the title of "ir". The research is done in the Knowledge Based Systems (KBS) group, headed by prof. dr. H.Koppelaar. I would like to thank prof.dr. H. Koppelaar for given me the opportunity to graduate in the group knowledge based systems.

The supervisor of the work was dr. drs. L.J.M. Rothkrantz. I want to thank L. Rothkrantz for all the support, patience and the pointers during the research.

I wish to thank my parents for making my study possible and supporting me during my entire study. Further I want to thank my wife, Hsiao for all her patience.

Shao Cheng Woo,

Abstract

The usefulness of identifying a person from the characteristics of his voice is increasing. The thesis will describe an approach to speaker identification where a neural classifier is used to separate different speakers. Several possible solutions will be showed. We will select cepstrum parameters as speaker's feature and explore the ways for solving the problem of speaker identification; the artificial neural network is introduced. The random combination of isolated digits from 0 to 9 is specified as identification utterances. The system has been evaluated on a database of isolated digit utterance of 20 speakers.

Contents

1	Introduction	4
2	Speaker identification models	9
2.1	Basic model of a speaker identification system	9
2.2	Speech signal	11
2.3	Linear predictive coding	16
2.4	Artificial neural network	19
2.4.1	Supervised learning	21
2.4.2	Unsupervised learning	23
2.4.3	Advantages and limitations of neural networks	26
2.5	Speaker recognition	27
2.6	Neural models for speaker identification	28
3	Experimental design	32
3.1	Hardware and software	32
3.2	Database	33
3.3	Goals	33
3.4	Respondents	33
3.5	Pre-processing of the speech data	35
3.6	Feature extraction	37
3.7	Creation of the training and test data set	38
3.8	Experiment set up	39
3.9	Creation of network	40
4	Experiments	46
4.1	Experiment 1 (one syllable versus two syllables digits)	46
4.2	Experiment 2 (words versus digits)	50
4.3	Experiment 3 (multiple digits)	52

4.4	Experiment 4 (usage of numbers greater than 9)	53
5	Summary	55
5.1	Conclusions	55
5.2	Recommendations	56

Chapter 1

Introduction

Personal Identity Identification is an essential requirement for controlling access to protected resources. Personal identity is usually claimed by presenting a unique personal possession such as a key, a badge or a password. However, these can be lost, stolen, or counterfeited, thereby posing a threat to security. Hence, identification based on biometric features of a person can be a solution [Naik]. This can be attempted by examining an individual's biometric features, such as fingerprints, hand geometry, or retinal pattern, or by examining certain features derived from individual's unique activity, such as speech or handwriting. The speaker recognition task falls under the general problem of pattern classification. In each case, the features are compared with all the previously stored features of persons. If the comparison is favourable, based on a decision criterion, then the person is identified. Among these methods, speaker recognition based on a person's voice has special advantages for practical deployment. Speech is our most natural means of communication and, therefore, user acceptance of the system would be high. Speech conveys linguistic information, speaker-dependent (individual) information, and many more other kinds of information. Among these, individual information plays the most important role next to linguistic information. Individual information takes the form of voice quality, voice height, loudness, speed, tempo, intonation, accent, the use of vocabulary and so on. Voice quality and height, which are the most important of the auditory types of individual information, can be related mainly to the static and dynamic characteristics of the spectral envelope a fundamental frequency (pitch). Advances in digital signal processors and speech technology have made possible the design of fast, cost effective, high performance speaker recognition systems.

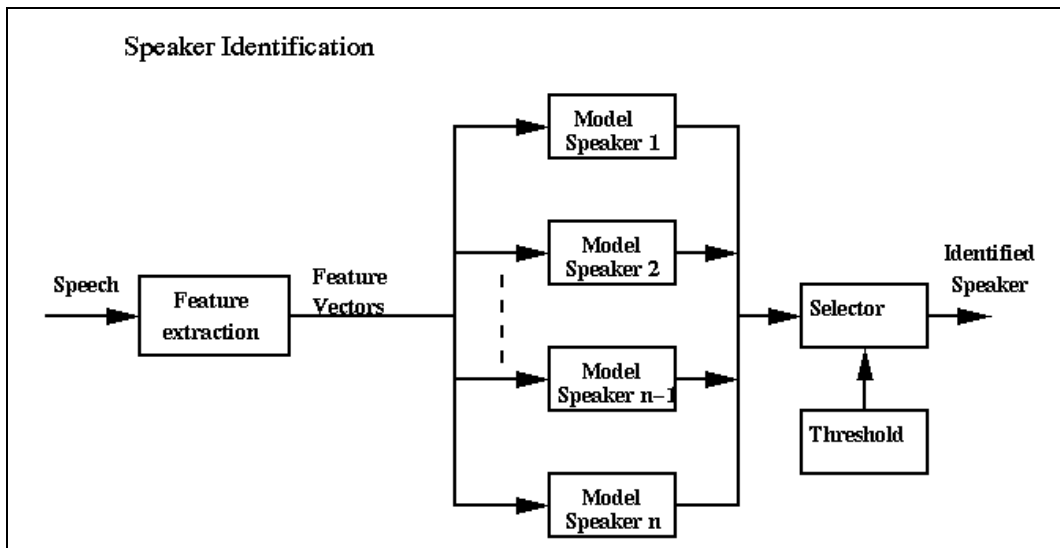


Figure 1.1: Basic model of Speaker Identification

In figure 1.1 we display a general model of speaker identification, where each block represents a function. The unknown speaker pronounces a password. In Feature extraction the characteristics of the speaker's utterance are extracted (*feature vectors*), the Model Speaker n consists out of reference feature vectors of the speaker n , in the selector the speaker is selected where the feature vectors (from Feature extraction) resembles the feature vectors (stored in the Model Speaker) most. When the resemblance expressed in a percentage is not equal or higher then the threshold then the speaker it not recognized.

Speaker recognition can be divided into *speaker identification* and *speaker verification*. Speaker identification is the process of determining from which of the registered speakers a given utterance comes (see figure 1.1). Speaker verification is the process of accepting or rejecting the identity claim of a speaker (see figure 1.2). In figure 1.29 ,each block represents a function. The unknown speaker claims an identity and pronounces a password. In Feature extraction the characteristics of the speaker's utterance are extracted (feature vectors), in Model Claimed Speaker the reference feature vectors are stored, the decision algorithm computes the resemblance between the feature vectors (from the feature extraction) when this is higher then the threshold

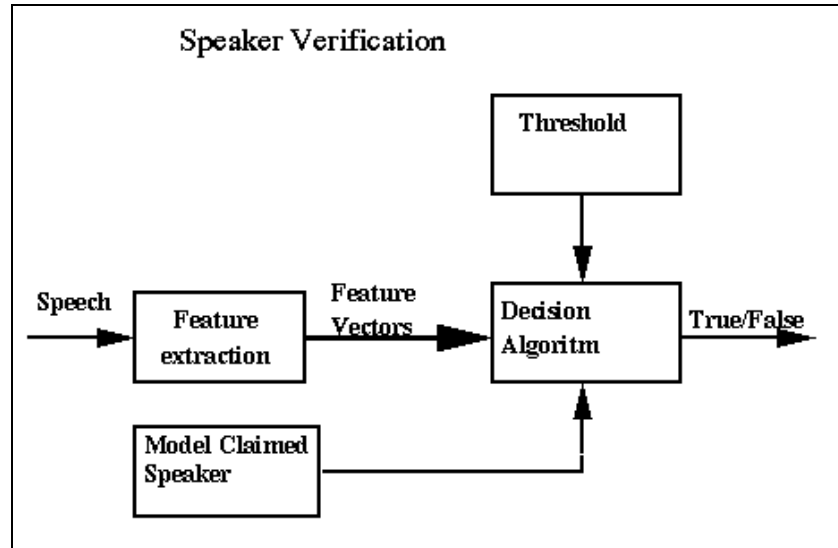


Figure 1.2: Basic model of Speaker Verification.

the claimed identity is accepted. The process of "getting to know" speakers is referred to as training and consists of collecting data from utterances of people to be identified. The second component of speaker identification is testing; namely the task of comparing an unidentified utterance to the training data and making the identification. The speaker of a test utterance is referred to as the *target speaker*. The terms *speaker identification* and *speaker recognition* are used interchangeably. In this thesis, only text-independent recognition is considered. By *text-independent*, we mean that the identification should work for any text in either training or testing. This is a different problem than *text-dependent* recognition, where the text in both training and testing is the same or is known. In the latter case, knowledge of the word sequence can be exploited to improve performance. In some cases text dependency is implicit, e.g., training and testing is done with digit strings although the digit strings may be different in training than in testing. Text dependent systems require the recitation of a predetermined text, thereby maintaining a high degree of user cooperation, whereas text-independent systems accept speech utterances of unrestricted text. In text-dependent systems, with adequate time alignment, one can make precise and reliable comparisons between two utterances of the same text. This is not easily ac-

completed with text-independent systems. Hence, text-dependent systems have a much higher level of performance than text-independent systems. There are two main reasons for wanting a speaker identification system to prompt the speaker with a new password phrase for each new occasion: (1) the client does not have to remember a fixed password and (2) the system cannot easily be defeated with replaying of recordings of the speaker's speech.

Speaker recognition can be subdivided in two further categories, closed-set and open-set problems. The closed-set problem is to identify the speaker from a group of N known speakers. Naturally, the larger N is, the more difficult the task is. The speaker that scores best on the test utterance is identified. Alternatively, one may want to decide whether the speaker of a test utterance belongs to a group of N known speakers. This is called the open-set problem, since the speaker to be identified may be not to be one of the N speakers. If a speaker scores well enough on the basis of a test utterance, then the target speaker is accepted as being known. Though the open-set task involves only binary decision (accept or reject), it is not necessarily easier than the closed-set problem, since it requires that a score be developed that has an absolute meaning; namely, a score that provides a calibrated measure of believe that the target speaker is known. The score is compared to a threshold for purposes of acceptance or rejection. The process of developing a calibrated score is referred to as *score normalization*. While this normalization process is not required for the closed-set problem, score normalization can play an important role in robust, closed-set, recognition procedures [Gish et al]. In addition, this normalization enables the scores from the robust procedures to be used directly for the open-set problem. *Speaker verification* see figure 1.2 is a special case of the open-set problem and refers to the task of deciding whether a speaker is who he or she claims to be. Often, however, speaker verification systems must not only verify the voice, but also the text with a speech recogniser in order to prevent impostors from using recordings. In this thesis, we will focus attention on the closed-set problem.

Potential applications: The potential for applications of speaker recognition systems exists any time speakers are unknown and their identities are important. In meetings, conferences, or conversations, speech technology makes automated identification of participants possible. If used in conjunc-

tion with continuous speech recognizers, automatic transcriptions could be produced containing a record of who said what. This capability can serve as the basis for information retrieval technologies from the vast quantities of audio information produced daily. In law enforcement [Klevans R.L, et al] [Anon], a speaker recognition system can be used to help identify suspects. In other words, security applications are abound. Access to cars, buildings, bank accounts and other services may be voice controlled in the future. Some existing applications use voice in conjunction with other security measures, perhaps a codeword, to provide an extra level of security. Speaker identification also has applications to other voice technologies. For example, speaker recognition can be usefully employed in speech recognition systems. Gender recognition, based on a variant of speaker recognition techniques, is already in use in many speaker independent speech recognizers to improve performance.

The problem definition is as follows

- A literature survey about speaker identification is performed
- Design a model for speaker identification.
- Create a workbench on which the test with the model can be performed.
- Design and implement a prototype.
- Test the prototype on its functionality.

This thesis focuses on the closed-set text-independent speaker recognition problem. The text, however, is restricted to numbers consisting of three digits where any combination of the words "nul", "een", "drie", "vier", "vijf", "zes", "zeven", "acht" and "negen" are allowed. However, the possibility of pronouncing numbers greater than nine and words has been investigated as well. Several existing techniques and methods are developed for solving any recognition problems are introduced. In the following, a brief overview of the organization of this thesis is presented. In chapter 2 the basic building blocks of a speaker identification system are described. Further in chapter 4 the Text-independent Speaker Identification system is tested with several experiments and chapter 5 holds the conclusions and the recommendations.

Chapter 2

Speaker identification models

2.1 Basic model of a speaker identification system

Figure 2.1 shows a block diagram of a basic Speaker Identification system. Next we will describe each block in more detail:

- **Speech signal:**
Here the utterance of a speaker is converted to a digital data format and stored in a file (which we will call a speech file).
- **Pre-processing:**
The speech signal in the speech file contains data that is not needed (like noise or non-speech) this has to be removed. The resulting speech signal will be passed through a filter that will amplify the speech signal and stored in a speech file (from this point this can be seen as speech data).
- **Feature Extraction:**
To extract the characteristics of the utterance of the speaker, the speech data is used. We will use an algorithm that is able to extract certain characteristics from the speech data, which will be discussed in section 2.2 (Vocal Signal). The extracted characteristics will be called the feature vectors.
- **Classifier:**
Here the identification of the speaker has to be established, on the basis

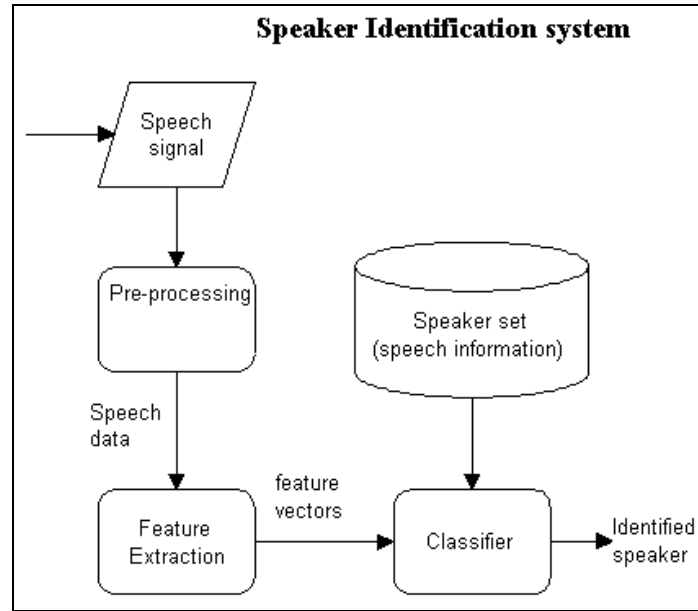


Figure 2.1: Basic model of a speaker identification system

of the given utterance of the speaker. There are several ways to take this decision on the basis of the feature vectors; this will be pointed out in section 4.

- **Speaker set:**

The speaker that is to be identified must belong to the Speaker-set. In the Speaker set the speech data of all speakers are stored.

In the recognition phase our system demands the speaker to pronounce three randomly chosen digits. However, instead of these digits we also investigated words and digits bigger than nine (for instance thirteen, twenty, etc.) and also the usage of words. The following basic problems have to be solved:

- What kind of input (like the number of digits, one or two syllables digits etc.) can to be used?
- How to choose a representation of the speaker's utterance (feature vectors)?
- How to select the type of classifier?

- how to configure the classifier (to get an optimal result)?

The aim is to develop a working system that is capable of recognizing a speaker on the basis of the speaker's utterance.

2.2 Speech signal

Feature extraction is the first important step in solving any recognition problem, and consists in our case, in obtaining a set of characteristics parameters with a high discriminating power between speakers to be used in signal classification.

What is it about the speech signal that conveys information about the speaker's identity? There are, of course many different sources of speaker identifying information. The speech signal conveys information about the speaker, these include "*high-level*" features (such as dialect, context, speaking style, emotional state of the speaker, etc.) These features are often used by human listeners to identify a person, however implementations to identify these perceptual bases have not been successful. The reason for this is because of the difficulty in acquiring and quantitatively measuring the speaker discriminating features used by humans. Hence, operational speaker identification systems use "*low-level*" parameters, such as pitch, spectral magnitudes, formant frequencies, energy profiles, etc. which can be derived from acoustic measurements of the speech signal.

These variables may be measured as a function of time or the statistics of long-term averages may be used as recognition variables. But the real question, the essence of the problem, is this : *How stable are these speakers discriminating features?* Given a speech signal, is the identity of the speaker uniquely decodable? The fact is that the speech signal is a complex function of the speaker and his environment. It is an acoustic signal generated by the speaker and which does not convey detailed anatomical information, at least not in any explicit manner. This distinguishes voice recognition from fingerprint identification, since fingerprint recognition uses fixed, static, physical characteristics, while speaker recognition uses dynamic "performance" features that depend upon an act.

Thus there exist inherent limitations in performance, which are attributable to the nature of the speech and its relationship to the signal generator

(the speaker). To appreciate these limits we must understand the source of speaker-discriminating information and how it is encoded in the speech signal. The speech signal, being a consequence of articulation, is determined by the vocal apparatus and its neural control. Thus there are two possible sources of speaker information; namely, the physical and structural characteristics of the vocal tract and the controlling information from the brain and articulatory musculature. This information is imparted to the speech signal during articulation along with all the other information sources. These other sources include not only the linguistic message but also the speech effort level (loud, soft), emotional state (e.g., anger, fear, urgency), health, age, and so on.

The characteristics of the speech signal are determined primarily by the linguistic message, via control of the vocal tract musculature and the resulting articulation of the vocal cords, jaw, tongue, lips and velum (which controls coupling to the nasal cavity). This articulation, in turn, produces the speech signal as a complex function of the articulatory parameters. The secondary speech messages, including speaker discriminates, are encoded as non-linguistic articulatory variations of the basic linguistic message. Thus the information useful for identifying the speaker is captured indirectly in the speech signal, a side effect of the articulatory process, and the speaker information may be viewed as "noise" applied to the basic linguistic message. Thus the problem with speaker recognition is that there are no known speech features or feature transformation which are dedicated solely to carrying speaker-discriminating information, and further that the speaker-discriminating information is a second-order effect in the speech features.

The fact is, however, that different individuals typically exhibit speech signal characteristics that are quite strikingly individualistic. We know that people sound different from each other, but the differences become visually apparent when comparing spectrograms from different individuals. The spectrogram is by far the most popular and generally informative tool available for phonetic analysis of speech signals. The spectrogram is a running display of the spectral amplitude of a short-time spectrum as a function of frequency and time. The amplitude is only rather crudely plotted as the level of darkness, but the resonant frequencies of the vocal tract are usually clearly presented in the spectrogram. Figure 2.2 demonstrates the degree of difference between spectrograms of 3 different speakers saying the number "neigen". The spectrogram is a display of the amplitude of a speech signal as a function of frequency and time. Note the differences between the individual

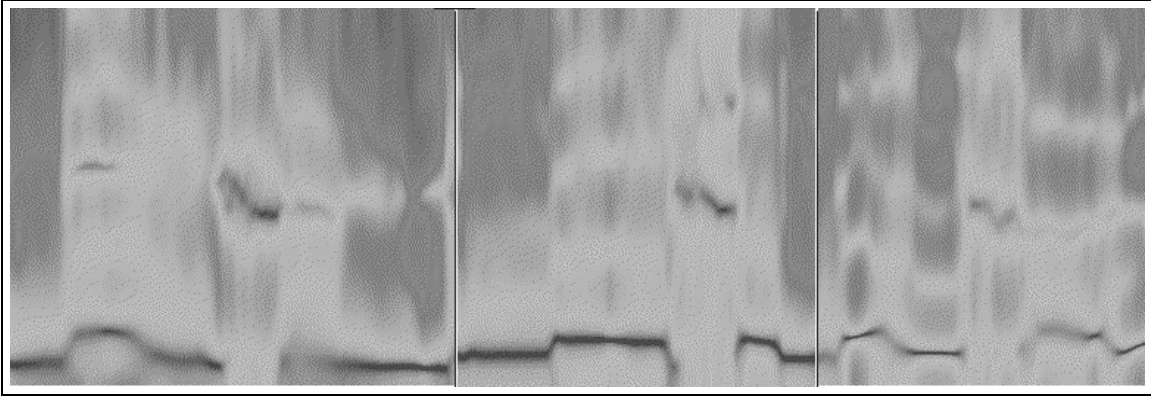


Figure 2.2: This figure exhibits 3 different spectrograms, one from each speaker pronouncing the number "negeen".

pronunciations. Segments duration's, formant frequencies, and formant frequency transitions, pitch and pitch dynamics, formant amplitude, all exhibit gross differences from speaker to speaker. Thus these speakers would be very easy to discriminate by visual inspection of their spectrograms.

But there are problems with this appealing notion of spectrographic differences. The primary difficulty lies not with the similarity between different speakers. Speakers usually sound very different from each other, and, in fact, the spectrograms in figure 2.2 show large differences between speakers. The real problem is that a single speaker also sounds (and looks, spectrographically) very different from time to time. We call this phenomenon "intraspeaker variability." This is illustrated in figure 2.3, which displays the spectrograms of three speakers (pronunciation of the number "zeven"). The spectrogram is a display of the amplitude of a speech signal as a function of frequency and time. One of the key issues in developing a text-independent speaker recognition system is to identify appropriate features and measures that will support a good recognition performance. The usage of the long-term average spectrum as a feature vector was discovered to have a potential for free-text recognition during initial exploratory studies of fixed-text recognition using spectral pattern matching techniques [Pruzansky]. Unfortunately, the long-term spectrum is not a good stable feature vector to use for speaker recognition. Long-term spectrum is sensitive to changes in the spectral response of any interposed communication channel. More important, the long-

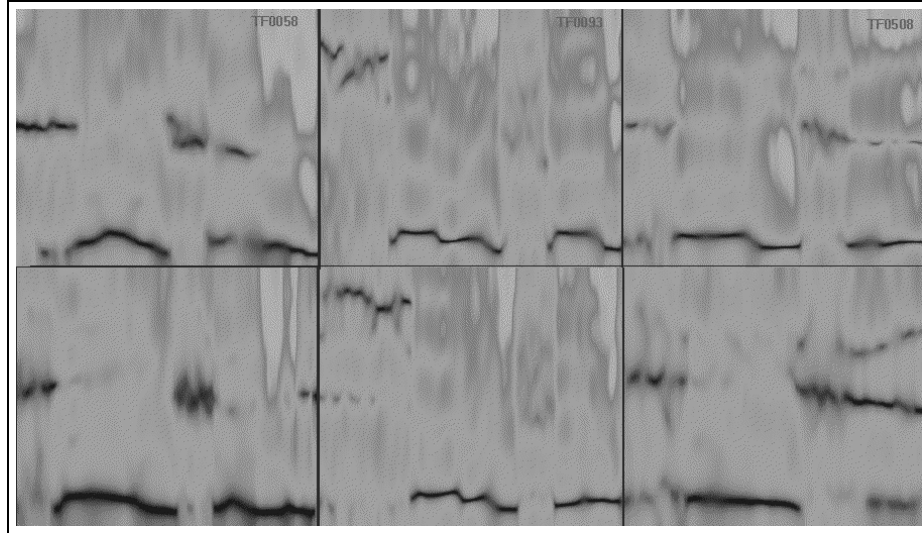


Figure 2.3: Spectrogram for three speakers of the pronunciation of "zeven".

term spectrum is not particularly stable across variations in the speaker's speech effort level. A number of increasingly more sophisticated approaches have been developed to overcome some of the more fundamental limitations of a simple Euclidean distance measure on a simple spectral amplitude vector [Cheung], [Wohlford] and [Shridar et al]. These approaches typically attempt to stabilize and statistically characterize the features that represent the speech spectrum. These features include statistically orthogonal spectral vector combinations, cepstral coefficients, and a variety of LPC-based parameters. Surprisingly, the primary measure of choice remains the spectral amplitude vector, and very little effort has been devoted to the development of other measure such as pitch, formant frequencies or statistical time functions. One reason for selecting the spectral amplitude vector is that it has typically produced performance superior to other features such as voice-pitch frequency [Markel].

Short time spectral features of speech signal have long been used successfully in speaker recognition applications, the spectral features hold speaker related information. The selection of the acoustic features is crucial for the effectiveness of the system. These features should have the following properties:

- Discriminate between speakers while being tolerant of intra-speakers variability's.
- Be easily measurable from the speech signal.
- Be stable over time.
- Not to be susceptible to mimicry by possible impostors.

There are several important issues involved in automatic speech and speaker recognition, including:

1. How to extract short-time spectral information from raw speech signals?
2. How to efficiently represent instantaneous spectral information at any time instant?
3. How to reliably characterize transitional spectral information associated with the time-varying properties of a speech signal in a compact form?
4. How to use instantaneous and transitional spectral features to measure the similarity (or dissimilarity) between two given running spectra?
5. How to make use of instantaneous and transitional spectral features in a complementary way?

Short-time spectral information of speech signal is usually extracted through a filter bank, a Fast Fourier Transform, or a LPC (Linear Predictive Coding) spectral analysis. Atal [Atal] compared several different spectral representation of speech spectra including LPC predictor coefficients, autocorrelation coefficients and LPC-derived cepstral coefficients, etc. and found that the LPC coefficients based spectral representation gave the best speaker recognition performance. Both text-dependent and text-independent experiments were conducted, and the performance of the text-dependent speaker recognition system was found to be better than that of the text-independent one. Furui [Furui] used both instantaneous and transitional spectral information in his LPC cepstrum-based speaker verification experiments to characterize a sequence-long utterance.

2.3 Linear predictive coding

Before any features can be extracted from the speech signal, some pre-processing has to be performed. The speech signal contains noise and unimportant data that has to be excluded. The noise can be filtered out by band limiting the speech signal (for instance between 100 Hz to 3.0 kHz). The resulting speech signal is then amplified. The parts of the speech signal where the energy of the signal is zero (silence) should be removed. LPC is a very important spectral estimation technique because it provides an estimate of the poles (hence the formants) of the vocal tract transfer function. The LPC algorithm is a n^{th} order predictor which attempts to predict the value of any point in a time-varying linear system based on the values of the previous n samples. The representation of the vocal tract transfer function, $H(z)$, can be represented by the following equation:

$$H(z) = \frac{G}{1 - \sum_{i=1}^p a(i)z^{-i}}. \quad (2.1)$$

The values $a(i)$ are called the *prediction coefficients* while G represents the amplitude, or gain, associated with the vocal tract excitation. The notation z^{-1} indicates a single, discrete-time delay in the domain of z-transforms. For discrete-time signals, the z-transforms can be considered a generalization of the Fourier transform. The poles of the transfer function in equation 2.1 are determined from the roots of the polynomial in the denominator. The LPC can only derive the resonant frequencies, or the formants, but not the zeros. The LPC does not adequately estimate signals that have no poles, such as some unvoiced speech noise. The non-linear signal components adversely affect the LPC estimates.

For the speech signal $s(n)$, the predicted speech sample $\tilde{s}(n)$ is a function of $a(i)$ and prior speech samples according to:

$$\tilde{s}(n) = \sum_{i=1}^p a(i)s(n-i). \quad (2.2)$$

LPC analysis involves solving for the $a(i)$ terms according to least squared error criteria. If the error is defined as:

$$\begin{aligned} e(n) &= s(n) - \tilde{s}(n) \\ &= s(n) - \sum_{i=1}^P a(i)s(n-i), \end{aligned} \quad (2.3)$$

then taking the derivative of the squared error with respect to the coefficients $a(i)$ and setting it equal to zero gives:

$$\begin{aligned} \frac{\partial}{\partial a(j)} [s(n) - \sum_{i=1}^P a(i)s(n-i)]^2 &= 0 \\ [s(n) - \sum_{i=1}^P a(i)s(n-i)]s(n-j) &= 0 \text{ for } 1 \leq j \leq P. \end{aligned} \quad (2.4)$$

Thus,

$$s(n)s(n-j) = \sum_{i=1}^P a(i)s(n-i)s(n-j) \text{ for } 1 \leq j \leq P. \quad (2.5)$$

A possible method for solving the matrix is called the *autocorrelation method*, which assumes that the signal is stationary within the analysis windows. The autocorrelation solution to 2.5 can be expressed as

$$R(j) = \sum_{i=1}^P a(i)R(|i-j|) \text{ for } 1 \leq j \leq P, \quad (2.6)$$

where $R(j)$ is an even function $R(j)=R(-j)$ and is computed from:

$$R(j) = \sum_{m=0}^{N-1-j} s(m)s(m+j) \text{ for } 0 \leq j \leq P. \quad (2.7)$$

Once the autocorrelation terms $R(j)$ have been calculated, a recursive algorithm, called *Durbin's recursion* [Makhoul], is used to determine the values of $a(i)$. The initial state of the recursion begins with an energy term, which contains the summed, squared energy in the windowed signal,

$$E^0 = R(0). \quad (2.8)$$

At each step in the recursion the following calculations are performed:

$$\begin{aligned} k(i) &= (R(i) - \sum_{j=1}^{i-1} a^{i-1}(j)R(i-j))/E^{i-1} \text{ for } 1 \leq i \leq P \\ a^i(i) &= k(i) \\ a^i(j) &= a^{i-1}(j) - k(i)a^{i-1}(i-j) \text{ for } 1 \leq j \leq i-1 \\ E^i &= (1 - k(i)^2)E^{i-1}. \end{aligned} \quad (2.9)$$

The final solution for $a(j)$ is given by $a^P(j)$ for $1 \leq j \leq P$. Given that the vocal tract does not produce a "purely" linear speech signal, the solution for $a(j)$ is optimal, but not exact. The most difficult part of the speech signal to predict is the glottal pulse because it contains a large amount of energy, which "instantaneously" appears in the signal.

One can calculate the cepstrum in 2 ways, one using simple recursion and the other with the Fourier transform.

Using the Fourier method: Speech wave $x(n)$ can be expressed as a convolution of speech wave $g(n)$ and vocal tract impulse response $v(n)$. In other words,

$$x(n) = g(n) * v(n). \quad (2.10)$$

Letting the logarithmic operation for the discrete Fourier transformation be D ,

$$D\{x(n)\} = D\{g(n)\} * v(n) = D\{g(n)\} + D\{v(n)\}. \quad (2.11)$$

The inverse discrete Fourier transform for $Dx(n)$ is called a cepstrum. In other words,

$$\begin{aligned} c(n) &= \frac{1}{2\pi} \int_0^{2\pi} \log|X(\omega)| e^{jn\omega} d\omega, \\ X(\omega) &= |X(z)|_{z=e^{j\omega t}}. \end{aligned} \quad (2.12)$$

The cepstrum for $X(n)$ turns out to be the sum of the cepstrum for $g(n)$ and the cepstrum for $v(n)$. The independent variable of the cepstrum has a time dimension (frequency). In the case of a voiced sound, $D\{g(n)\}$ appears as a component in the neighbourhood of $1/F_0$ (F_0 :fundamental frequency) on the time axis, and $Dv(n)$ as a component of the short time domain. Thus, a window is opened in the cepstrum and the short time range components extracted (this is accomplished by removing $g(n)$), and if a discrete Fourier transformation is performed in this, the spectral envelope is obtained (see Fig 2.4).

Using the Linear Prediction coefficient The LPC-derived cepstral coefficients are defined as follows, where c_i is the i th cepstral coefficient:

$$\begin{aligned} c_1 &= a_1 \\ c_i &= a_i + \sum_{k=1}^{i-1} ((1 - (k/i))a_k c_{i-k}), \quad 1 < i \leq N \end{aligned} \quad (2.13)$$

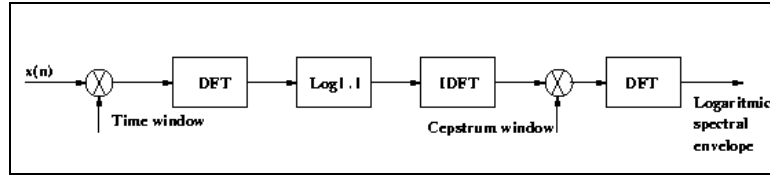


Figure 2.4: Cepstrum analysis

where c_i is the i^{th} cepstral coefficient and a_k are the prediction coefficients. Unlike LPC coefficients, cepstral coefficients are independent and the distance between cepstral coefficients vectors can be calculated with a Euclidean-type distance measure.

2.4 Artificial neural network

Artificial neural networks (ANN) are computational models that attempt to emulate the human brain by a topology that resembles interconnected nerve cells. NNs are capable of modelling non-linearity and can be used for many different tasks, such as classification, associative memory, and clustering. This versatility has allowed them to solve problems in areas as diverse as computer vision, process control, and medical diagnostic. The main drawback of a neural network is their long training time. Although knowledge about neural networks is still in an early stage, their application to automatic speaker recognition is significant.

ANN consists of a collection of neurons that are connected by weighted pathways. Each neuron is a processing element performing one function and producing one output (see figure 2.5). The computation performed by a typical neuron consists of taking the sum of its inputs (equation 2.14) and using that value as the argument to a (preferable) non-linear function (equation 2.15)

$$d_i = \sum_{j=1}^N w_{i,j} x_j \quad (2.14)$$

Where $w_{i,j}$ stands for the weighted pathway between neuron i and j , x_j the output of neuron j and d_i is the result of the sum.

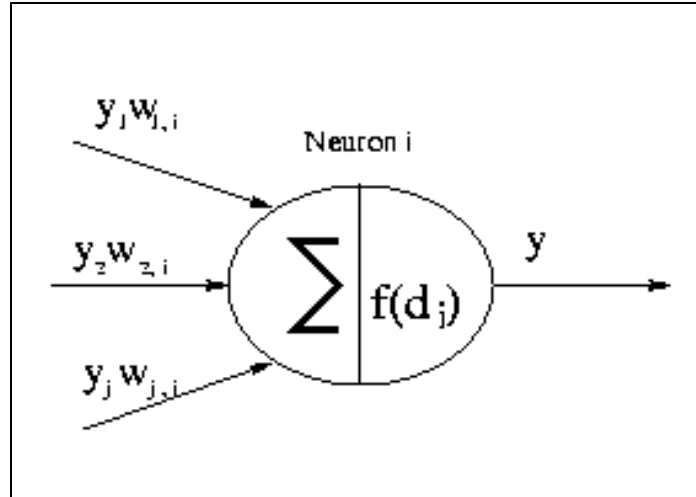


Figure 2.5: Basic model of a neuron

$$y_i = f(d_i) = \frac{1}{1 + e^{-\lambda d_i}} \quad (2.15)$$

This non-linear function (2.15) is called the activation function of the neuron. The most commonly used activation function is the sigmoid. The λ stands for the learning rate, it influences the slope of the sigmoid and the x is the result of the equation 2.14. For small values of *lambda*, the sigmoid approximates the linear activation function, while for large values the sigmoid approximates the step function. The output of the cell is computed with the function 2.15 ($f(d_i)$). The activation rule for the neuron has a geometric interpretation. The choice of which activation function to use in combination with a given similarity metric is typically guided by two constraints: differentiability and non-linearity. Activation functions that are differentiable are often selected because they facilitate analytic manipulation of the mapping produced by the ANN. In particular, *gradient-descent* learning algorithms rely on the calculations of the gradient of a global measure of the network activity with respect to the network's weights. The sigmoid is easy to manipulate analytically, and for this reason they are most commonly used activation functions. A principle advantage of ANNs constructed from neurons with non-linear activation functions is that they are able to compute more complex mappings than networks of neurons with linear activation functions. There are two types of neural networks that have the learning capabilities. The learning is accom-

plished through modification of processing element weights. It is important to develop a good model of the weight modification process. Most learning laws are formulated with a specific goal in mind. A commonly encountered type of goal is to move to position that yields a network that minimizes or maximizes some particular global neural network cost or performance function, such as mean squared error. Neural network adaptation always takes place in accordance with a *learning regimen*. Learning models can roughly be divided in two categories: *supervised learning* and *unsupervised learning* paradigms.

2.4.1 Supervised learning

Minsky and Papert acknowledged the computational potential of multi-layer network consisting of several layers of modifiable weights and non-linear processing elements [Minsky]. They argued that networks with one hidden layer could solve complex problems that cannot be tackled by a single layer perceptron. These types of networks are often called *multi-layer perceptrons (MLPs)*. From this point, Werbos[Werbos] defined a learning algorithm for multi-layer networks. This was rediscovered by other researches and popularised by Rumelhart, Hinton and Williams in the 90s [Rumelhart et al]. This algorithm is also known as *backward error propagation*. This algorithm for determining the weights in a multi-layer network is very popular due to its simplicity and efficiency. The term hidden layer as mentioned before can be seen as a transition between the input and the output layer. Remembering that x_{ij} (the input of neuron j from neuron i), d_j (is the sum of the input of neuron j), y_{ij} (the output of neuron i) and λ (stand for the learning rate). The training rules for the Backpropagation network can be summarized by three statements.

1. The change of weights between neuron i and j , $\Delta_p w_{ij}$ (during the training of the p th presentation of an input \output pair) is proportional to computed error say ∂_{pj} for the j th neuron. This can be expressed in the following equation :

$$\Delta_p w_{ij} = \beta \partial_{pj} y_{pi} \quad (2.16)$$

The calculation of Δ for the hidden layers is the important aspect of the Error Backpropagation network.

- For the output layer, the error is calculated based on the difference between the wanted output on x and the real output, the error for neuron j can be expressed as:

$$\partial_{pj} = (y_{target} - y_{pj})f'_j(d_{pj}) \quad (2.17)$$

From the expression $(y_{target}-y_{pj})$ it is clearly that the error is proportional to the difference between the actual output y_{pj} and the target output y_{target} . The term $f'_j(a_{pj})$ indicates the rate of change. Rumelhart et al. indicates that the activation function should be smooth function, when the error is small the weight change is small otherwise the weight change is greater.

- In case when the neuron is in the hidden layer then the error of this neuron can be defined as being proportional to the sum of the errors of all the neuron (say k neurons)that are connected to the output as modified by the weights. In symbols:

$$\partial_{pj} = \left(\sum_{\text{for all } k} \partial_{pk}w_{kj} \right) f'_j(d_{pj}) \quad (2.18)$$

The training consists of 2 steps, the *forward pass* and the *backward pass*. The *forward* pass during the input is applied and allowed to propagate to the output. The error values of the output neurons are calculated by using equation 2.17. During the *backward* pass these errors are propagated backwards and the weight changes made (see equation 2.18). This procedure will continue backwards until the weights in the input layer are adjusted. This then followed by another forward pass and a further backward pass, and so on.

The network is functioning as an input/output system, it receives an input vector \mathbf{x} and emits a vector \mathbf{y} . Supervised learning for such a system implies a regimen in which the network is supplied with a sequence of examples $(x_1,y_1),(x_2,y_2),\dots,(x_k,x_k)$. As each input x_k is entered into the neural network, the "correct output" y_k also is supplied to the network. When the response pattern of the network does not match the networks output, the network corrects by modifying the weights to reduce the difference between the output and y_k . The result that we want is that the neural network generalizes the training set examples to the entire problem environment, the weights of the neurons will define an almost flat surface. So that when the network is

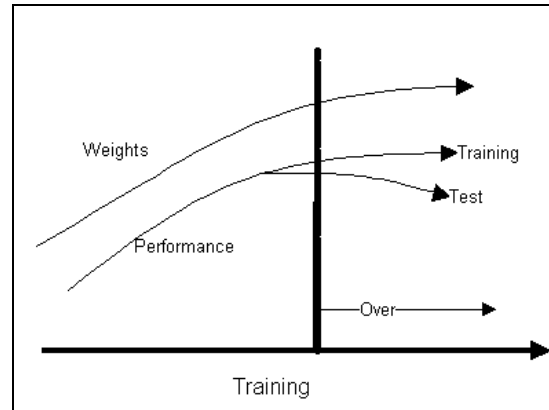


Figure 2.6: An interpretation of net training.

prompted with a slight different input (that lies within the surface) it will "interpolate" the output of the trained data to actual data. A problem that occurs when one is not careful with choosing the number of training iterations is the phenomenon of *overtraining*. The problem that arises here is that the network loses its important aspect namely generalization. With overtraining the network will adjust its weights so that it can match the input to the output, the result is that the surface is not flat but very wrinkled. This will cause a very bad interpolation between the points on the surface. So the number of training iteration is not dependent of one cost function but with another function (see figure 2.6).

2.4.2 Unsupervised learning

Self-learning network have particular importance because they can act as "optimal" vector quantifiers. They provide useful descriptions of the input signal in terms of self-generated primitives, arranged to form an ordered map, which has topological properties with a metric that is related to the similarity between the input signals. Neural network clustering algorithms have been employed for a large number of applications such as speech recognition and pattern recognition. It is possible that the representation of knowledge be in the particular form of a feature map that is geometrically organized. Kohonen showed that a set of interconnected adaptive units has the ability to change its responses in such a way that it will adapt to represent the characteristics

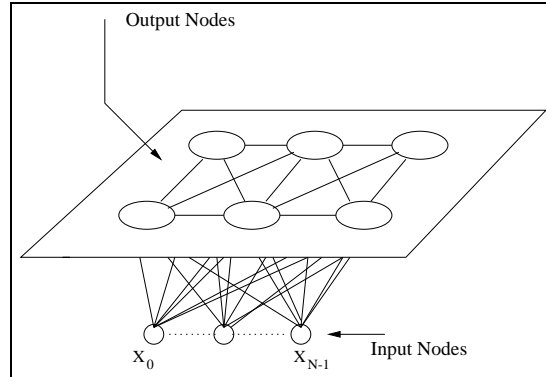


Figure 2.7: The architecture of the Kohonen feature map.

of the input signal. It is the same as the classification problem in classical pattern recognition such as vector quantization algorithm, where the feature vector space is to be partitioned into a set of non-overlapping regions, and where a reference vector represents each region.

A number of researches have studied models which develop feature maps [Kohonen82], [Scofield]. Kohonen has developed a model, which both emulates the feature maps observed in biological systems, and serves as a useful computational device for pattern classification [Kohonen82a]. In this model, the development of cell responses occurs in an unsupervised or *self-organising* fashion. Cells automatically develop stimulus specific properties, and the network self-organizes such that neighbouring cells are "tuned" to similar stimuli. In Kohonen's approach, a layer of cells is organized as a two dimensional grid, with each cell receiving input from a separate layer of input cells. The cells of the second, or feature-map layer, receive input from both the first layer and from neighbouring cells in the second layer. The lateral connectivity within the second layer is assumed to be distance dependent, but identical for all cells. Kohonen has selected a common distance-dependent relation in which neighbouring cells excite each other through positive connection weights, while more distant cells are mutually inhibitory. An example is illustrated in Figure 2.7. Lateral connectivity is not modifiable in Kohonen's model because the connections are employed only for the communication of activity levels within the network. In practical, the time-rate of change of a

cell's activation in the second layer is given by:

$$\dot{x}_i = -r_i(x_i) + \left(\sum_{j=1}^N (\omega_{ij} - f_j)^2\right)^{\frac{1}{2}} + \sum_{l=1}^L m_{il}x_l \quad (2.19)$$

In this equation, the first term is a non-linear loss, typically modelled by $r_i(x_i) = x_i^2$. The function $r_i(x_i)$ forces the network activity to decay to zero in the absence of external stimulation. This permits the activity to be dominated by external input and not locally induced activity. The second term is simply the input of f_j to cell i from input cells j in the first layer of N cells, mediated by modifiable weights ω_{ij} . The third term in equation 2.19 sums the activity levels of the L remaining cells in the network. The non-modifiable weights m_{il} are distance dependent so that $m_{il} = m(|i-l|)$. The effect of the lateral connectivity is a "clustering" of activity levels during processing. If a cell has been preferentially stimulated, then it will dominate and reduce the activity of cells within the inhibitory zone while exciting cells within the closer, excitatory zone. Thus the lateral connectivity results in a "winner-take-all" processing and a recruiting of neighbouring cells to the winner's activity. During training, the effect of the lateral connectivity may be replaced with a simple winner-take-all computation. In particular, a pattern vector $\vec{f}(t)$ is presented to each cell in the input layer and an initial activity, $x_i^0 = |\vec{f}(t) - \omega_i(t)|$, is computed. The cell that has the smallest distance is selected as the center of a modification region in the network. The weight vectors of cell i and each of its spatial neighbours in a region $R(t)$ are adjusted according to the rule:

$$\vec{\omega}_i(t+1) = \vec{\omega}_i(t) + \eta(t)(\vec{f}(t) - \vec{\omega}_i(t)). \quad (2.20)$$

The weight vectors of cells which fall outside of the winner-take-all region $R(t)$ are not modified. The quantity $\eta(t)$ is the learning rate and is selected such that $0 < \eta(t) < 1$. Typically $\eta(t)$ decays to zero after a pre-selected number of training set presentations. The exact decay schedule is not critical; however, Kohonen has noted that the convergence of the feature map consists of two distinct phases: initial formation of map order, and final convergence. The learning rate is usually chosen as a piecewise linear decay with the second phase lasting 0 to 100 times longer than the first phase [Kohonen82a]. An appealing aspect of the feature map learning procedure of equation 2.20 is that it may be easily converted to a supervised learning algorithm. Kohonen has called his algorithm *Learning Vector Quantization (LVQ)* [Kohonen88].

2.4.3 Advantages and limitations of neural networks

As pointed out before that with the use of a neural network one can exploit certain properties but one also have to face several problems that will arise. The advantages and its properties when one uses a Neural network are:

- **No explicit knowledge is required:** They can learn and classify input data without having an explicit knowledge of the application. This approach is well suited for applications where the topic is too complex to be explicitly formulated or where not all knowledge can be specified exactly.
- **Ability to generalize:** Neural networks can be useful to generalize from known data when different inputs are presented. Input like the training data are recognized, while cases outside the training set can be put into the closest match while the network has already learned.
- **Ability to adapt to changing environment and events:** Neural networks can learn from training data sets. During the learning stage, the weights change in response to the training data presented to the network. This data driven feature of neural networks allows a fast adjustment of the network to changing conditions. The neural network can be modified by retraining with updated training data sets.
- **Robust:** Neural network have the ability to deal robustly with poor structured data. Noise, incorrect and incomplete data may have enough information to recall or restore the complete stored information.

There are however (of course) limitations and disadvantages:

- **Training:** The network has to be trained with the appropriate training data set, determined structure and parameters. The task to find suitable, complete training sets to be a vital one. Furthermore, the convergence (of Root Mean Square Error) may lead to local minima. Finally, the training phase may take a very long time of course the danger of *over training* is resident.
- **Non-transparent:** A neural network does not use symbolic knowledge as used by humans to express a reasoning process. The knowledge is represented as stored patterns of numeric weights, it is not possible to inspect and/or examine the knowledge. Thus, a neural network can be seen as a *black-box* solution to problems.

- **Configuration:** Theoretically a neural network can solve any problem with one hidden layer, only the number of neurons in this layer is unknown. To find an optimal number of neurons, number of layers, learning rate parameters numerous set-ups are to be tested and where of course the time to train all the networks increases drastically.

2.5 Speaker recognition

Before addressing the various speaker recognition techniques and systems, it is appropriate to review the acoustical bases for speaker recognition. An excellent exposition in this subject is presented in a book by Francis Nolan [Nolan]; see also [Hecker]. Probably the most significant paper on speaker recognition, as judged by the amount of further research it has stimulated, was a paper by Kersta introducing the spectrogram as a means of personal identification [Kersta]. The term "voiceprint" was introduced in this paper, and 99-percent correct identification performance based upon visual comparison of these voiceprints (spectrogram) was reported in a voiceprint identification task using 12 reference speakers. The use of the term "voiceprint" has probably contributed to the popularity of voiceprint identification by analogy of the term "fingerprint." In the largest evaluation of voiceprints ever conducted, under the direction of professor Oscar Tosi at Michigan State University, [Anon], 0.5-percent identification error was achieved using voiceprints for nice clue words under the restrictive condition of isolated word utterances, closed trials, and contemporary speech. That is, the unknown speech tokens, and identification scenario, particularly any type of forensic model, which is the major application of voiceprint identification. The recognition reliability of voiceprints, relative to the reliability of a listener's judgment, is also an important consideration in the eyes of voiceprints (and in weighing voiceprints evidence in the courtroom). In previous studies comparing the performance of voiceprint identification with aural speaker discrimination by human listeners, the error rates for aural discrimination have always been smaller [Carbonell], [Stevens], [Clarke]. In the 1968 study by Stevens, for example, a closed-set identification test using a homogeneous group of eight reference speakers yielded 6-percent error for listening and 21-percent error for voiceprint. Thus the reliability of the voiceprint technique for speaker identification is clearly a fragile issue, because identification performance is sensitive to many acoustic, environmental, and speaker conditions. Further-

more, the use of the voiceprint technique is highly questionable, because better performance can likely be obtained through a listener's judgment. This brings up an important perspective on the development and evaluation of speaker recognition technology in general; namely, the comparative performance of a computational technique with respect to some generally accepted benchmark. Such a performance comparison seems to be a valuable step toward calibration of the absolute performance of any speaker recognition technique.

2.6 Neural models for speaker identification

A wide variety of pattern recognition processes have been applied to the task of speaker identification, most of which have their roots in speech recognition. The relative success of these methods has brought about commercial utilization of person recognition by speech.

Recently neural pattern classifiers have received a great deal of interest for tasks such as speech classification. The use of neural networks for speaker recognition has advantages, not only in terms of recognition performance but also in terms of computational tractability, and scalability, which are major issues in neural modelling. Currently little is known of the way speaker dependent characteristics are embedded in the speech signal. However, we expect to be able to extract some of the characteristics automatically, by training a neural model using unconstrained optimisation techniques.

There are several neural network configurations possible, the following four configurations can be considered as adequate:

1. One large neural network

The speaker classification is performed by one network, with three layers and trained with back-propagation algorithm. First layer is for the input, which is a vector with the length 10, one hidden layer and one output layer. The number of neurons in the output layer equals the number of speakers. When speech data of speaker n is presented then the n^{th} neuron in the output layer is 1 and all the other 0. This is the less favourite choice, the consequence of adding or removing a new speaker to the speakers set is that the neural network has to be re-trained and re tuned (this takes a lot of time), the number of layers

and the neurons will increase this will have effect on the complexity of the NN. The advantage is that the time to recognize is minimal.

2. Each speaker has a personalized network [Oglesby]

Each person to be recognized has a personalized network, with the output being active for the features associated with that person. An important aspect of the training is that the output is inactive for all speech not from the specified speaker. Each net is trained using back-propagation and conjugate gradient descent. The neural model used for automatic speaker identification comprises a 3 layer feed-forward net with standard sigmoid non-linearity's. The first hidden layer has 32 nodes, the second hidden layer 16. The input is a 10 dimensional feature vector, and the output is a single value that is active for the feature vectors associated with one of the speaker known to the system. Advantage is that adding a new speaker is simple; this can be achieved by adding a new personalized network. The disadvantage is the time to recognize the speaker, which is $n \cdot \text{time}(\text{network})$ ($n = \text{number of speakers}$) when using a sequential computer.

3. Binary network [Rudasi et al]

One big neural network is replaced with a large number of smaller networks. The binary-pair approach will need N times $(N-1)/2$ small neural classifiers to be trained, each to distinguish between two of the N categories (one might think of male and female or age). Each of these small binary neural nets is independent of the others as well as the training data. Each classifier would sort incoming data into one of two groups. At each step of classification half of the remaining possible categories are eliminated. The configuration of each classifier is a network with two layers, fully connected, memory less, feed-forward with sigmoid non-linearity. Network weights are initialised with random values uniformly distributed from -0.05 to 0.05. Each was trained with back propagation method with a fixed learning rate (0.1 - 0.3) and fixed momentum term (0.7). The output targets were 0.999 for the node corresponding to the target category, and 0.001 for the other(s). The training and testing data sets were the same in each experiment. The conclusion of Rudasi was that this partitioned approach performs comparably, or even better, than a single large network. For large values of N (>10), the partitioned approach requires only a fraction of the

training time required for the partitioned network would be about two orders of magnitude less than for the single large network.

4. Kohonen network [Ciobanu et al]

Vector Quantization (VQ) is an effective method of segregating data into clusters and determining the centroid of those clusters. VQ reduces a set of n k -dimensional vectors into a codebook of N centroids where $n \gg N$. There are several algorithms to create such codebook, but one can also create such codebook with a Kohonen network. The network is presented with the speech data of one speaker.

In a later report [Oglesby2] an improvement is made, they examined the variation in performance with the number of nodes for a single layer model, i.e. one hidden layer. Also two hidden layer models are investigated, the number of nodes in both layers being varied. The speech for the experiments is drawn from a speaker database and consists of 500 utterances from the digit set, 100 of which were used for training and 400 for recognition. In all experiments the same feature extraction process was used, namely 10^{th} order LPC-derived coefficients calculated on 256 samples. This resulted in the following conclusions: for the case of 16 nodes in the first layer a marginal improvement is seen as the number of second layer node is increased. This is despite the number of model parameters almost doubling from 249 to 465. When 32 first layer nodes are used performance actually drops in going from 8 to 16 second layer nodes. Above 16 a consistent improvement is seen, but again it should be noted the model size, in terms of free parameters, is growing rapidly as the number of second layer nodes is increased. A network with one hidden layer out-performs the two hidden layer models. The reason is that this is the consequence of the required decision surface for this specific task or the learning process.

For our speaker identification system we select the second configuration (see figure 2.8). The reason why the third option is not chosen is that the speaker set we are using is too small. The usage of a binary network would not be efficient. The type of neural network that is used is a Feed Forward neural network with one hidden layer, which is motivated by J. Oglesby [Oglesby2]. The neural network is first trained with the feature vectors of the speaker, every speaker owns a personalized neural network. The training of such personalized NN proceeds as follows, the NN will output a "1" when his own speech data (feature vectors) is presented as input and a "0" (speech data

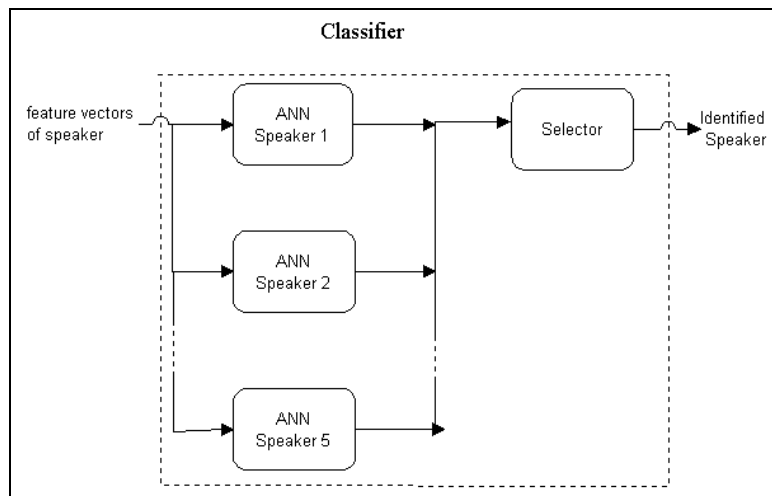


Figure 2.8: Neural classifier.

from other persons) otherwise. The activation function is a sigmoid and the generalized delta rule is used as update rule.

Chapter 3

Experimental design

Before we can proceed with experiments we have to create a workbench.

3.1 Hardware and software

The following hardware and software are used:

- Pentium III 450 MHz (hw)
- Sounblaster 16 (hw)
- CD-Rom player (Philips) (hw)
- Windows 95 (sw)
- Xwin32, this is needed to be able to use Snns
- Snns (sw), this program is used to create and train a neural network.
- soundforge (sw), used for editing the speech signal.
- cepstrum.exe (sw), it visualizes the cepstral vectors in a spectrogram.
- Shoten.exe (sw), it calculates the cepstral vectors of a speech signal.
- visualize.exe, it plots the average (sum of all the output divided by the number of cepstral vectors) of the output neural network.
- Borland c++, a c compiler.

Where "hw" stands for hardware and "sw" for software.

3.2 Database

The speech data is obtained via a corpus cd (Polyphone), on this cd 500 speakers are stored in arbitrary order. These speakers were recorded in several sessions over more than 3 months. This database is a telephone database recorded over local and long distance telephone lines using different types of handsets. Recordings took place from the speaker's office or from his/her home.

3.3 Goals

The experiments will be divided into two sections:

- proving that the neural network will converge to a stable point. A speaker identification (using ANN) will be configured and trained for several days, the MSE will be monitored.
- assessment of the impact of input (speech) on the recognition result, by changing certain variables (training iterations, number of neurons in the hidden layer, etc).

3.4 Respondents

From the polygon database 20 female speaker are selected (see figure 3.8) of the test set cd. The speech data of the selected speakers has the following properties:

- each digit (zero to nine) must be pronounced twice,
- the average speaker range is in the range of 20-30 years,
- each utterance of a digit is isolated from other utterances of digits.

The reason for the last properties is that the samples are hand selected, when the digits are close connected with each other then this will become a more difficult task to select only the needed digit without any interference of the other digit (see figure 3.2 versus figure 3.1).

The speaker-set consists out of the following respondents (table 3.1).

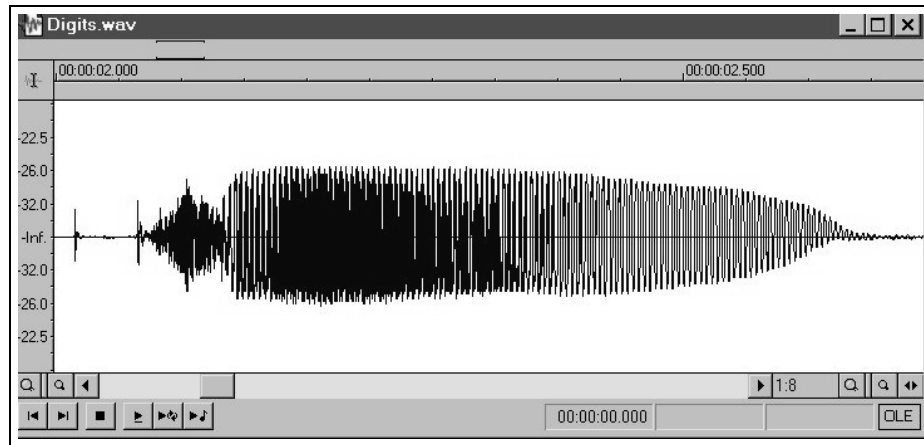


Figure 3.1: The speech wave of the pronunciation of the number 2 ("twee").

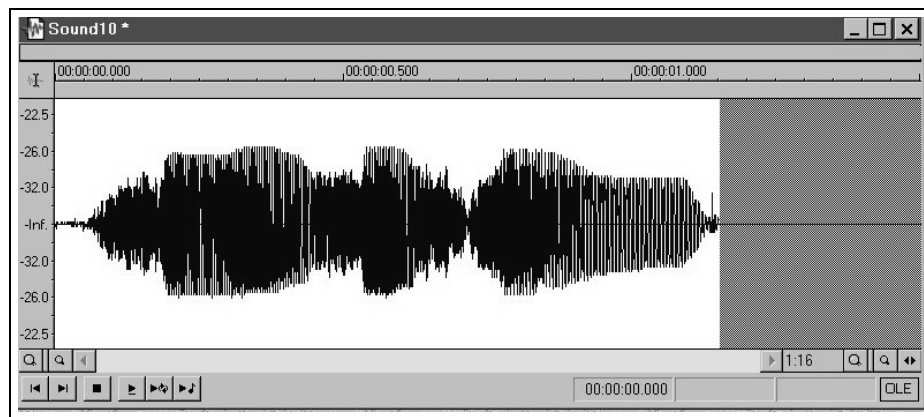


Figure 3.2: The speech wave of the pronunciation of numbers 4-6-1 ("vierzes-een").

Table 3.1: Overview of all the speakers in the speaker-set.

	code	age	domicile	sex
speaker 1	tf0058nh	36	Noord Holland	female
speaker 2	tf0093nh	33	Noord Holland	female
speaker 3	tf0106zh	52	Zuid Holland	female
speaker 4	tf0508zh	34	Zuid Holland	female
speaker 5	tf0122zh	30	Zuid Holland	female
speaker 6	tf0103ut	26	Utrecht	female
speaker 7	tf0107fr	33	Friesland	female
speaker 8	tf0109ge	25	Gelderland	female
speaker 9	tf0113zh	34	Zuid Holland	female
speaker 10	tf0114gr	31	Groningen	female
speaker 11	tf0115ge	29	Gelderland	female
speaker 12	tf0118ge	36	Gelderland	female
speaker 13	tf0124li	32	Limburg	female
speaker 14	tf0127zh	32	Zuid Holland	female
speaker 15	tf0130ov	34	Overijssel	female
speaker 16	tf0147nh	35	Noord Holland	female
speaker 17	tf0152nh	36	Noord Holland	female
speaker 18	tf0588zh	28	Zuid Holland	female
speaker 19	tf0722zh	30	Zuid Holland	female
speaker 20	tf0204nh	30	Noord Holland	female

3.5 Pre-processing of the speech data

The next step is to store the pronunciation of the digits zero to nine to a speech file. This must be done for all the respondents in the speaker-set (table 3.1).

The needed recordings from the digits zero to nine are selected from the following files that exists in each directory of each person: *digits.wav*, *number1*, *number2*, *number3*, *Number4* and *number5*. In these files the person is asked questions or the person has to pronounce prompted sentences that are related with the pronunciation of numbers. All files are stored in a NIST format, this consists out of a header (with information about the speech file) and the actual compressed speech data. To decompress the speech data an msdos program (*shorten.exe*) is used, *shorten -x <speech file> <output file>* (the "-x" switch stands for extraction) writes the decompressed speech file to

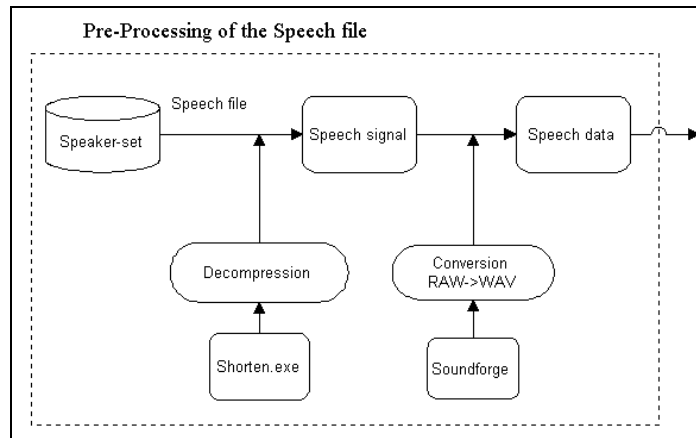


Figure 3.3: Model of pre-processing of a speech file.

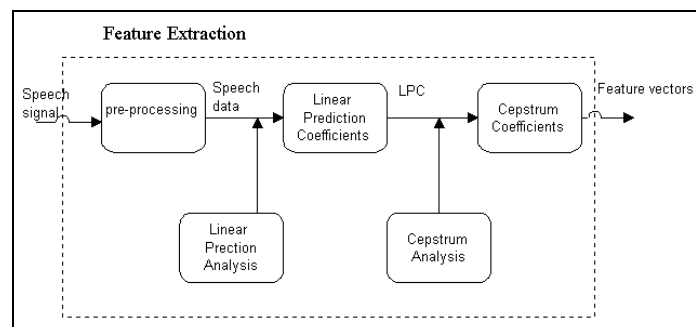


Figure 3.4: Model of feature extraction.

```

MS-DOS Prompt
12 x 16
D:\verslag>cepstrum
Extracting features from speech signals
Author: Jialong He, $Id: cepstrum.c,v 1.2 1996/04/09 11:27:15 jialong Exp $

Usage: cepstrum [options] speechfile

Where [options] can be any of the following:
-o feature vector file [stdout]          -b [512] starting sample
-w [256] window size                    -s [128] window moving step
-L LPC [off]                             -p [10] LPC order
-C LPCC [off]                             -n [10] LPCC order
-M MFCC [off]                             -r [10] MFCC order
-R RCEP [off]                             -g [2] RCEP order
-P PARCOR [off]                          -f pitch period [off]
-V Voiced segments only [all]            -t [0] Tolerance, 0 adaptive
-S *not* swap byte order                 -l [0] Label for this class

D:\verslag>

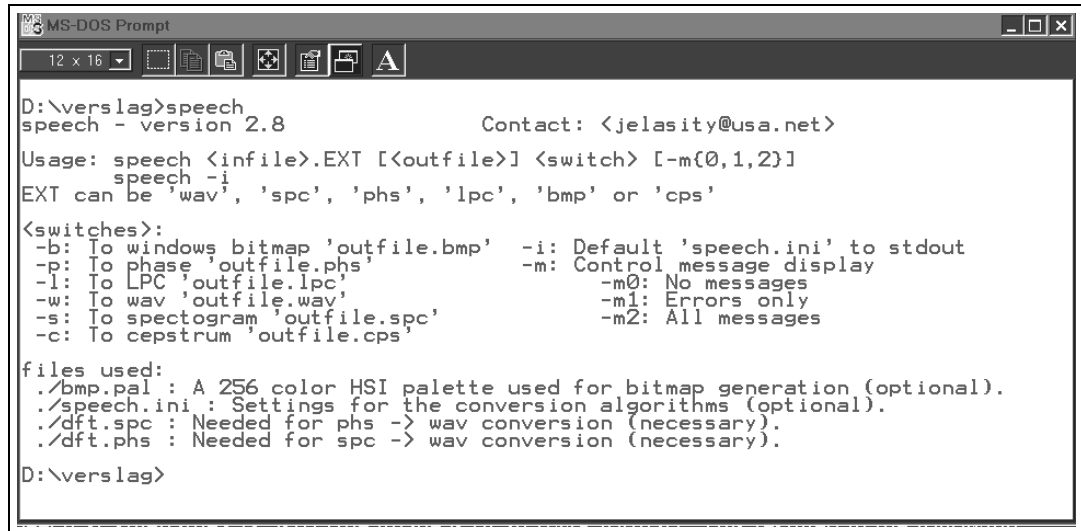
```

Figure 3.5: The Command line of `cepstrum.exe` with all the possible switches.

a specified output file. The decompressed speech data file is a RAW sound format, unfortunately this is not a standard sound format (like WAV). Using the sound utility program *Soundforge* all the speech data files of the type RAW format are converted to a WAV sound format. From the converted speech data files the needed digits are manually selected and saved for further usage (1.wav, 2.wav etc, where the number stands for the spoken digit).

3.6 Feature extraction

The next step is to extract from each sound file the feature vectors, this is achieved by calculating from each digit the cepstrum vectors. To perform the calculation of the cepstrum vectors, the program *cepstrum* is used. The author of the program *cepstrum.exe* is J. He, this program is used to calculate from a sound file the cepstrum coefficients. A short overview of the programs possible variables are given in figure 3.5 more information can be found in the appendix. The following configuration is selected: `cepstrum -C -V -n 10 <input file> <output file>`. Which will calculate the first 10 cepstrum coefficients and only the voiced parts of the input file. Further the window size is set to 256 samples and windows overlap is 128 samples. This setting is



```

D:\verslag>speech
speech - version 2.8                Contact: <jelacity@usa.net>

Usage: speech <infile>.EXT [<outfile>] <switch> [-m{0,1,2}]
      speech -i
EXT can be 'wav', 'spc', 'phs', 'lpc', 'bmp' or 'cps'

<switches>:
-b: To windows bitmap 'outfile.bmp'  -i: Default 'speech.ini' to stdout
-p: To phase 'outfile.phs'           -m: Control message display
-l: To LPC 'outfile.lpc'             -m0: No messages
-w: To wav 'outfile.wav'             -m1: Errors only
-s: To spectrogram 'outfile.spc'     -m2: All messages
-c: To cepstrum 'outfile.cps'

files used:
./bmp.pal : A 256 color HSI palette used for bitmap generation (optional).
./speech.ini : Settings for the conversion algorithms (optional).
./dft.spc : Needed for phs -> wav conversion (necessary).
./dft.phs : Needed for spc -> wav conversion (necessary).

D:\verslag>

```

Figure 3.6: The Command line of `speech.exe` with all the possible switches.

used for all the digits (1.wav, 2.wav etc these files are filled in <input file>) of each speaker, and stored in a file with the extension cep (1.cep, 2.cep etc. where the number stands for the spoken digit). To visualize the resulted cepstrum vectors, the program *speech* (see figure 3.6) is used, the image is created with the following command line: `speech <filename> -lsb`. First a spectrogram is created with the use of the cepstrum coefficients, then the image file is created (this is an BMP format). This resulted in the following image 3.7, in this image one can see the pronunciation of the digit 7 ("zeven"), the upper left to the upper right images are the utterances of TF0058, TF0093 and TF0508. The lower three images are a second pronunciation of the digit 7. These images are a function of time (x-axis) and the amplitude (y-axis), the colour intensity is an indication of the amplitude. The lightest colour stands for low amplitude and the brightest colour stands for high amplitude. One can see that the second utterance does resemble the previous utterance and differ from the other utterances of other speakers.

3.7 Creation of the training and test data set

There are two data sets to be created:

Training set A training set consists out of cepstral vector of all digits (zero

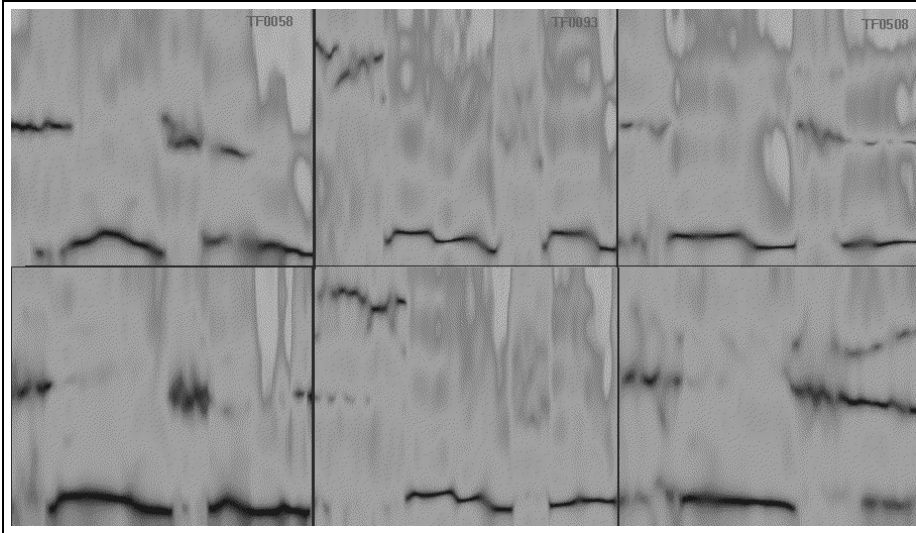


Figure 3.7: Spectrogram for three speakers of the pronunciation of "zeven".

to nine) of all speakers (speaker belongs to the speaker set). A file is created in a snns pattern format, with the extension *pat*. In the pattern file there are in total of 3277 cepstral vectors. For all cepstral vectors of speaker tf0058nh and one (target output) is inserted otherwise a zero is inserted. This file is used to train the personalized for speaker tf0058nh neural network.

Test set Several speakers are randomly selected from the speaker set, from these speakers some digits are selected. The cepstral vectors of these digits are used to test the recognition capabilities of the neural network. The cepstral vectors of each speaker are saved to a pattern file (in according to the snns format).

3.8 Experiment set up

We now have :

- recorded speech.
- respondent pronouncing three digits (between zero and nine).
- cepstral vectors of the digits of all respondents.

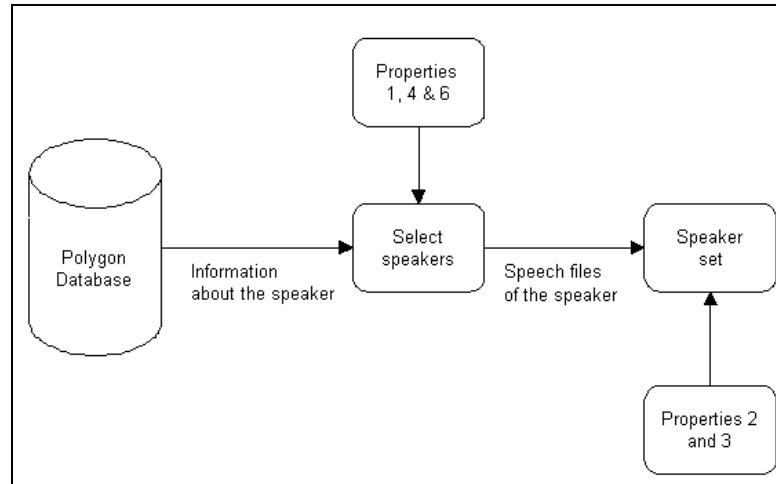


Figure 3.8: Speaker-set construction

- training and test data set.

The recognition will take place on the base of three spoken digits. Three digits are simple to use as a password and easily to remember, instead of using several words.

3.9 Creation of network

To create a neural network the program snns is used. There are some variables that are to be filled in:

- model of speaker recognition
- type of neural network.
- number of layers.
- number of neurons in the layers.
- number of training cycles.

The model that is chosen is a personalized neural network, each speaker will have a trained network. On the base of a paper of Oglesby ([Oglesby2]) we

use an Error Back propagation network and the number of layer is three. The first layer contains 10 neurons, this will function as an input layer and the third layer consist out of one neuron (output layer). On the base of the average output value (of all cepstral vector of speech data) of this neuron, the identity is established.

When unknown cepstral vectors of speech data are presented to the personalized network (of speaker tf0058nh) and the average output is high, then the speaker is probably tf0058nh. Otherwise when the average output is low then it is probably not speaker tf0058nh.

To calculate every time the average output of the neural network, a program (visualize) is written. The next time when "output of the neural network" is mentioned then this is the total sum of values (generated by the output neuron of the neural network) divided by the number of cepstral vectors. See figure 3.9, here a typically output file is shown. The line starting with *-0.7596* is the input pattern, this is one cepstral vector. The value *0.50683* is a value that is generated by the output neuron (in the output layer) of the neural network.

The number of neurons in the second layer and the number of training cycles is still open. We will test four network configuration, where the hidden layer varies from 60, 70, 80 and 90 neurons and select the network that performs best (good recognition, using less neurons in the hidden layer and with short training time).

Each network will default trained for 160.000 training cycles with the training data set, with the following set-up (see table 3.2). And the test data set is used to test the recognition capabilities. Every 100 training cycles the test pattern file is used to test the recognition performance. This option can be set in the *valid* textbox. By using the *graph* option the mean square error and the output of the test pattern can be monitored. On the x-axis the training cycles are shown and on the y-axis the error and output is shown (mostly between zero and one). The lower graph is the mean square error (between 0.02 and 0.04) and the other the output of the neural network (around 0.26).

During the training the following must happen:

1. The mean square error must decrease, but beware of overtraining.
2. The recognition performance must increase or stabilize.

```

MS-DOS Prompt - MORE
Auto
SNNS result file V1.4-3D
generated at Sat Aug 05 23:15:04 2000

No. of patterns   : 72
No. of input units : 10
No. of output units : 1
startpattern     : 1
endpattern       : 72
input patterns included

#1.1
-0.7596 -0.4516 -0.2344 -0.1527 -0.0089 -0.155 0.1015 -0.1132 -0.0632 -0.0415
0.50683
#2.1
-0.8463 -0.556 -0.0687 -0.1516 0.01 -0.0395 0.0253 0.0277 -0.0925 0.0269
0.49142
#3.1
-0.9037 -0.5453 -0.2486 -0.095 -0.1736 -0.072 0.1298 0.1038 0.028 0.105
0.48432
#4.1
-0.8666 -0.4716 -0.2694 -0.1401 -0.0267 -0.1663 -0.0202 0.0484 0.0494 -0.0783
0.49736
#5.1
-0.8203 -0.5142 -0.1882 -0.2466 -0.1364 -0.0055 -0.0114 -0.0697 0.1766 -0.1551
-- More --

```

Figure 3.9: Output of the neural network.

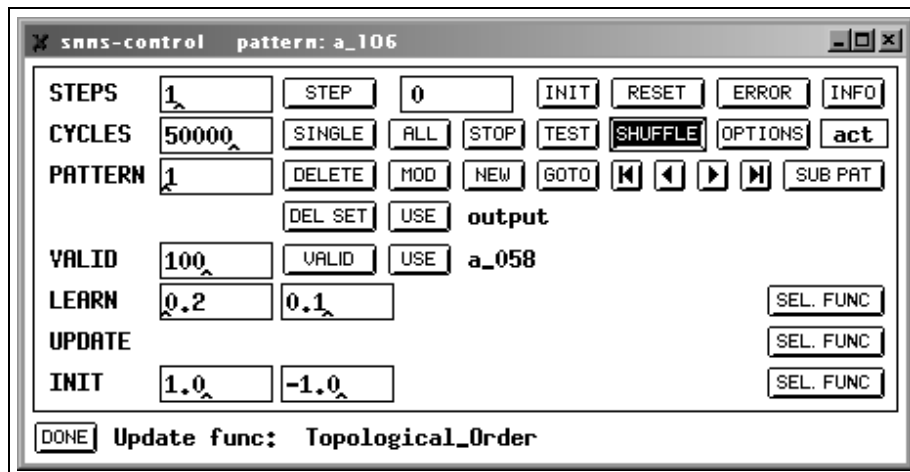


Figure 3.10: Control panel of Snns.

Table 3.2: Setup of snns.

variable	value
Learning rate	0.2
Maximun error	0.1
Learning function	standard backpropagation
Update function	topological order
Init function	randomise weights
Valid	100

The training is interrupted when the recognition performance worsens or when the mean square error did not decrease significantly. The following conclusions can be made:

1. When the training iteration is below the 100.000 the classification is very bad (for all three tested configuration), the reason for the bad performance is that the network has not yet reached the point where it has generalized the speaker's data.
2. When the training iterations exceed the 180.000 the recognition worsens (for the configuration with 80 and 90 neurons in the hidden layer), the network cannot recognize the speaker between the other speakers. The output of the network for all the speakers are close to each other, here the phenomenon arises which is called *overtraining*(see the section about classification for more details). The mean square error increases and the recognition performance worsens.
3. When one chooses 70 neurons in the hidden layer then the recognition is probably reasonable when the training iterations exceeds far over 1600.000 cycles, the output of all the speakers seems to be converse from each other but here the danger of *overtraining* exists.
4. Taken the factor of efficiency into account, the 80 neurons in the hidden layer is preferable above the 90 neurons. Here the number of training cycles is shorter and due to the number of neurons in the hidden layer less complex compared with 90 neurons in the hidden layer.

The resulting output of the neural network after 3 days continuous training is showed in figure 3.11. On the horizontal axis the number of training

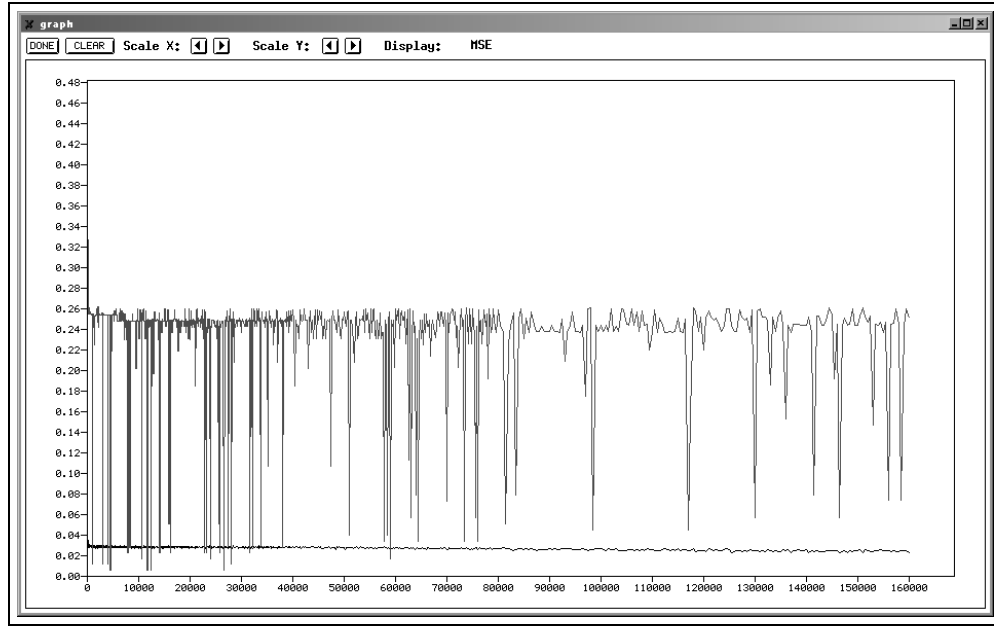


Figure 3.11: Result after 160.000 cycles of training.

cycles is shown and on the vertical axis the mean square error and the output of the neural network. We see that the mean square error stabilizes to a steady error in a very early stage. The reason that the training is not stopped is that the recognition performance is not good (the output varies too much). After 90.000 training cycles the recognition performance is stabilised. This trained network is prompted with several speech data of several speakers. this resulted in the following two graphs. The training time can be shorter, there are two neural networks selected. One neural network gave a high output for speaker tf0058nh (1368.000 training cycles) and one where the difference between speaker tf0058nh with the other speaker was maximal (1390000 training cycles). We now have a trained personalized neural network for speaker tf0058NH, the mean square error converts to a stable point and the recognition performance is good. The next step is to do some experiments with this trained network and tests its recognition capabilities.

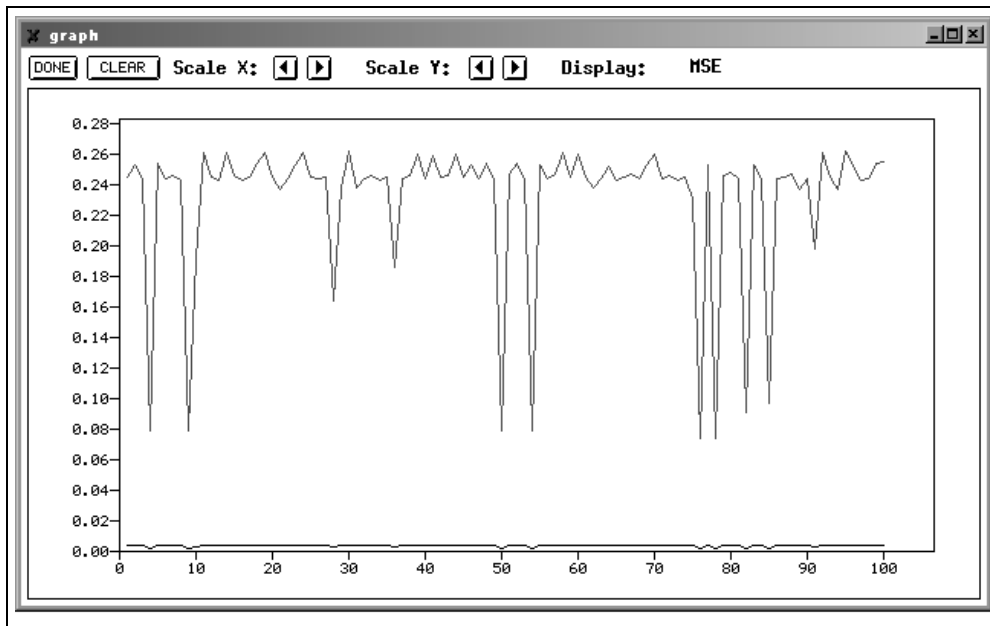


Figure 3.12: Output of the neural network, when input is speaker tf0058NH.

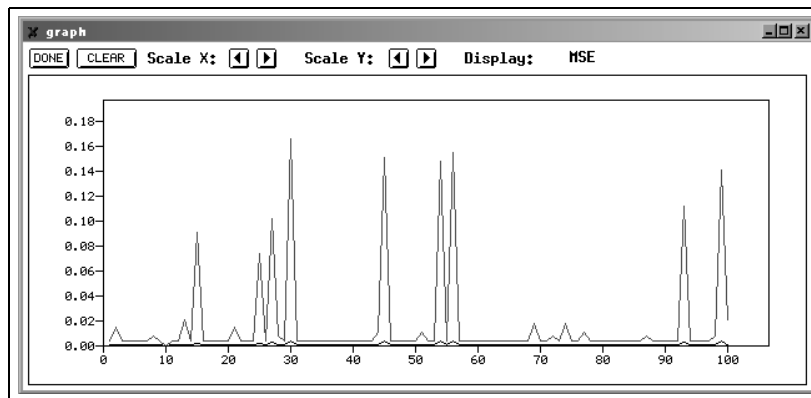


Figure 3.13: Output of the neural network, when input is not speaker tf0058NH.

Chapter 4

Experiments

We can now start experimenting the recognition performance by changing the input data. For speaker tf0058NH a personalized neural network is created and trained. With this neural network we will experiment. Further more we will only select the first five speakers who gave a high (average) output on the personalized neural network of speaker tf0058nh. Note that speaker one is always tf0058nh.

4.1 Experiment 1 (one syllable versus two syllables digits)

Of each speaker two sets of three digits are created, one other set consists out of one syllable numbers (the digits "1", "2" and "4") and the other set (the digits "7", "8" and "9"). The chosen digits that are used in the two sets are not the same as the one that are used in the training set. The two sets created for all speakers are stored in two separate files, namely 1klinker.nna (this contains the one syllable numbers) and 2klinkers (this contains two syllable numbers). In the file option in recall source window (see section "Creation of Network") these files are used to test the network. The results of the recognition capabilities are given in the table below, in this table the output value of the network is noted: From the resulting output of the network the difference of speaker 1 with all the other speaker is calculated, the result is shown in the table 4.3: Looking at the distance of speaker one with all the other speakers, we see that it decreases when two syllables is used also the overall output decreases. The pronunciation of the digit seven

4.1. EXPERIMENT 1 (ONE SYLLABLE VERSUS TWO SYLLABLES DIGITS)47

Table 4.1: The output of the personalized network of speaker 1 is given (1368000 training iterations).

input	speaker1	speaker2	speaker3	speaker4	speaker5
1 syllable (1-2-4)	0.456	0.321	0.320	0.346	0.383
2 syllables (7-8-9)	0.337	0.312	0.305	0.329	0.272

Table 4.2: The output of the personalized network of speaker 1 is given (1390.000).

input	speaker1	speaker2	speaker3	speaker4	speaker5
1 syllable (1-2-4)	0.467	0.342	0.329	0.385	0.417
2 syllables (7-8-9)	0.342	0.337	0.315	0.348	0.282

and nine in Dutch is :”ne-e-g-e-n” and ”z-e-e-v-e-n”. When we examine it closely, we see that the ”e-e” dominates and especially when the window size is 10 ms. The problem is that the utterance of ”e-e” is nearly the same for all the speakers, the consequence of this is that the feature vectors (of the ”e-e”) of all the speakers are close. When these feature vectors are fed into the network, it cannot see whether the utterance is from speaker 1 this can result in a ”mismatch” with another speaker. To see if this conclusion is correct two test sets are created one with 3 digits namely 1,2 and 7 (”een”, ”twee” and ”zeven”) and the second set: 3,4 and 8 (”drie”, ”vier” and ”acht”). From this we can see the following:

- The output for speaker one is higher when using digits 3, 4 and 8, here the vowels are shorter in contrast with digits 1,2 and 7 (where the output is lower).
- The output for all the other speakers is lower when using digits 3,4 and 8.
- When using the test digits 1,2 and 7 we can see that all the distance of the output of all speakers is small.

Here we can see that the supposed explanation is correct, the recognition is better when using numbers with short vowels. In figure 4.1 and in figure 4.2 a spectrogram is showed where the speaker pronounces one syllable digits (124 and 789)). The spectrogram is a display of the amplitude of a speech signal as a function of frequency and time.

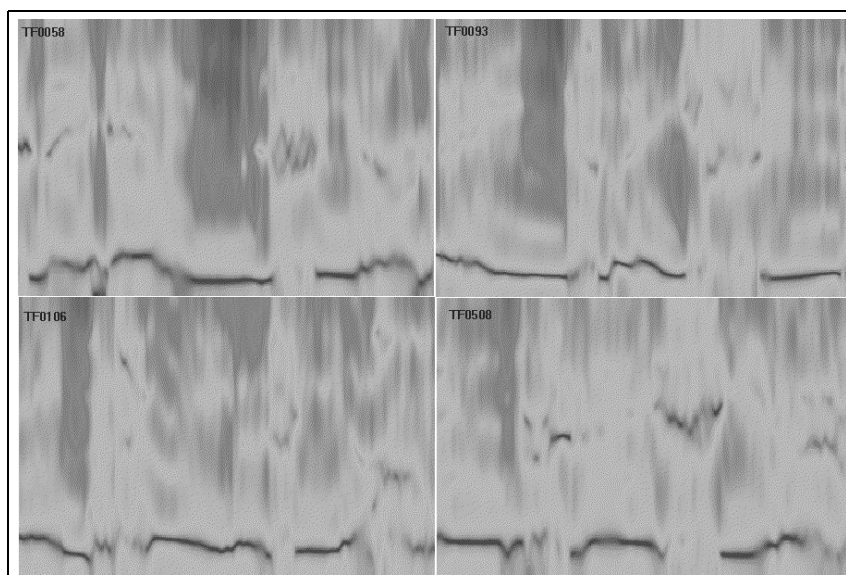


Figure 4.1: Each Speaker pronouncing the digits "een", "twee" and "vier".

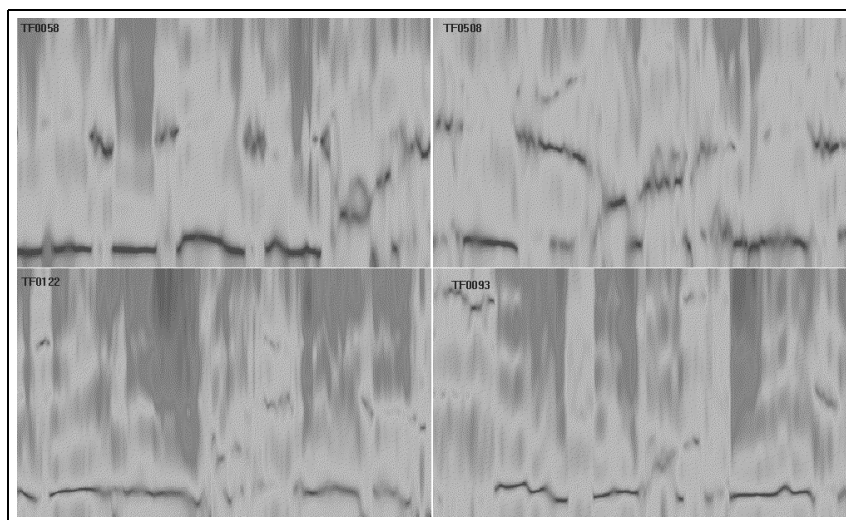


Figure 4.2: Each speaker pronouncing the digits "zeven", "acht" and "negen".

4.1. EXPERIMENT 1 (ONE SYLLABLE VERSUS TWO SYLLABLES DIGITS)49

Table 4.3: The difference of speaker 1 with other speakers (13680.000 training iterations).

input	speaker2	speaker3	speaker4	speaker5
1 syllable (1-2-4)	0.135	0.136	0.010	0.073
2 syllables (7-8-9)	0.025	0.032	0.008	0.065

Table 4.4: The difference of speaker 1 with other speakers (1390.000 training iterations).

input	speaker2	speaker3	speaker4	speaker5
1 syllable (1-2-4)	0.125	0.138	0.082	0.050
2 syllables (7-8-9)	0.005	0.027	-0.006	0.060

Table 4.5: The output of the personalized network of speaker one (1368.000 training iterations).

input	speaker1	speaker2	speaker3	speaker4	speaker5
"3-4-8"	0.379	0.255	0.292	0.369	0.322
"1-2-7"	0.344	0.343	0.320	0.322	0.360

Table 4.6: The output of the personalized network of speaker one (1390.000 training iterations).

input	speaker1	speaker2	speaker3	speaker4	speaker5
"3-4-8"	0.387	0.265	0.296	0.390	0.332
"1-2-7"	0.366	0.367	0.338	0.360	0.396

4.2 Experiment 2 (words versus digits)

A second set-up is created, to see what kind of effect it would have when one uses words instead of digits. The training set is gathered and recreated, again the speech data are acquired from the NIST database and from each person three words are selected. The words are selected from the following files *phonsnt1*, *phonsnt2*, *phonsnt3*, *phonsnt4* and *phonsnt5*. First the words for the training set are chosen, these have to be phonetically rich. The extraction of the selected words is done in a similar way as with the digits. After we

Table 4.7: Training set

# speaker	word 1	word 2	word 3
speaker 1	technisch	universiteit	voetbalwedstrijd
speaker 2	uitbetalingen	programmamakers	vakantie
speaker 3	diepgeworteld	filipijnen	februari
speaker 4	professioneel	debuterende	correspondentie
speaker 5	ziekenhuisspullen	uitsluitend	gedupeerde

have trained (similar way as with digits) the network with the training set, the testing can begin. From the same files several word are extracted, from each speaker three words (see table 4.8) are selected, namely words with one, two and three syllables. This resulted in the following output, which are

Table 4.8: The following words are selected.

# speaker	one syllable	two-three syllables	four-five syllables
speaker 1	park	rommel	vernielingen
speaker 2	dag	versliep	gewoontegetrouw
speaker 3	beide	getuigen	pessimisme
speaker 4	club	onderhoudt	gewonnen
speaker 5	staan	kwamen	gechanteerde

showed in table 4.9, and 4.10. The results are not spectacular, but this was to be expected. The reason is that when using words instead of digits the diversity of the sound will increase, the used network is configured for digits. A way to solve this bad performance is to change the configuration, one might think to increase the number of nodes in the hidden layer or add another hidden layer (due to the complexity of the sound). The phonetics

Table 4.9: The output of the personalised network of speaker one (1368000 training iterations)

# syllable(s)	speaker1	speaker2	speaker3	speaker4	speaker5
1 syllable	0.351	0.323	0.254	0.333	0.126
2-3 syllables	0.117	0.111	0.179	0.146	0.166
4-5 syllables	0.121	0.239	0.297	0.134	0.108

Table 4.10: The output of the personalised network of speaker one (1390000 training iterations)

# syllable(s)	speaker1	speaker2	speaker3	speaker4	speaker5
1 syllable	0.245	0.303	0.214	0.333	0.067
2-3 syllables	0.113	0.049	0.162	0.116	0.147
4-5 syllables	0.119	0.203	0.280	0.095	0.051

of the digits are : "ee", "ie", "ij", "e", "a" and "ie", in contrast with words. The number of phonetics in words is many times more than with digits.

4.3 Experiment 3 (multiple digits)

To see what kind of influence the number of digits can have on the performance of the network, the following experiment is created. From each speaker 4 sets are put together, the first set contains one digit, the second 2 digits, the third 3 digits and the last set contains 4 digits (of course all the selected digits are not from the training set). This resulted in the following table, in each column the output of the network is noted. The following digits are randomly chosen, 9824 (pronounce like "negen, acht, twee, vier"). Note

Table 4.11: The output of the personalized network of speaker one (1368000 training iterations)

# digit(s)	speaker1	speaker2	speaker3	speaker4	speaker5
one digit	0.226	0.286	0.329	0.290	0.229
two digits	0.342	0.304	0.343	0.361	0.268
three digits	0.339	0.303	0.344	0.358	0.276
four digits	0.369	0.313	0.346	0.345	0.302

Table 4.12: The output of the personalized network of speaker one(1390000 training iterations)

# digit(s)	speaker1	speaker2	speaker3	speaker4	speaker5
one digit	0.222	0.293	0.330	0.311	0.335
two digits	0.356	0.320	0.345	0.387	0.281
three digits	0.350	0.318	0.355	0.391	0.284
four digits	0.379	0.328	0.358	0.368	0.316

that a negative number indicates that the output of the speaker is higher than the output of speaker one. One can say that the recognition is more convincing when the number of digits is increased, the output of speaker one increases and the output of the other speakers slowly decreases. There is of course a limit of the numbers of digits, when it is too long then will not be user-friendly. The user has to pronounce a long password, which can result in agitation or making a slip of the mouth (like pronouncing the wrong number).

Additional research is done concerning the usage of numbers that are greater than 9 (like fourteen, twenty-one etc.).

Table 4.13: The difference of the output of the personalized network of speaker one with the other speakers(1680000 training iterations)

input	speaker2	speaker3	speaker4	speaker5
one digit	-0.060	-0.103	-0.064	-0.003
two digits	-0.038	-0.001	-0.019	0.076
three digits	0.036	-0.005	-0.019	0.063
four digits	0.056	0.023	0.024	0.067

Table 4.14: The difference of the output of the personalized network of speaker one with the other speakers(1390000 training iterations)

input	speaker2	speaker3	speaker4	speaker5
one digit	-0.071	-0.108	-0.089	-0.113
two digits	0.036	0.011	-0.031	0.075
three digits	0.032	-0.005	-0.041	0.066
four digits	0.051	0.021	0.011	0.063

4.4 Experiment 4 (usage of numbers greater then 9)

We have used till now digits in the training's and test phase of the speaker identification system, the question now arises is if numbers greater then nine has any influence on the performance. The next set-up is to get an answer on the question, we select 2 numbers that are greater then nine. The numbers are extracted from the files *number1* till *number5* of the Polygon database.

Table 4.15: The selected numbers for the test set.

	speaker1	speaker2	speaker3	speaker4	speaker5
number	32-64	32-67	37-76	21-71	45-71

The chosen numbers are not all the same, the reason is that each speaker pronounces different numbers. With these numbers the tests are performed. The results are a bit dissatisfactory, a possible explanation is that numbers greater then nine has more variation in tones than numbers smaller than 9. An important factor is that the neural network is trained with numbers below the nine.

Table 4.16: The output of the personalised network of speaker one (1368000 training iterations) when prompted with numbers greater than nine

input	speaker1	speaker2	speaker3	speaker4	speaker5
numbers > 9	0.311	0.349	0.285	0.306	0.255

Table 4.17: The output of the personalized network of speaker one (1390000 training iterations) when prompted with numbers greater than nine.

input	speaker1	speaker2	speaker3	speaker4	speaker5
numbers > 9	0.318	0.370	0.292	0.324	0.265

Chapter 5

Summary

We have seen the trajectory of the development of a Text-independent Speaker Identification system for a closed-set. A suitable method for feature extraction is selected. We have performed several tests with different type of input utterances (one or two syllable digits and words) of several speakers. A prototype of a speaker identification system is created and tested, we have seen that the prototype is able to recognize a speaker on basis of his speech characteristics.

5.1 Conclusions

We found that it is possible to use ANN in speaker identification in combination with cepstral analysis (feature extraction). Several ANN configurations (number of neurons in the layer and number of training cycles) are tested. Further more we have looked at the recognition performance by changing the input (number of digits, words and numbers above nine).

When one uses digits with a dominating "e-e" ("een", "twee" and "zeven") this will have a negative influence on the recognition. The usages of words and numbers greater then nine as input utterance in the speaker identification system resulted in a very poor performance. The main reason is that the system was designed for the usage of digits.

The result was that the recognition increases when the number of digits increases. One might be attempted to use more digits (five or six etc.) but this will not be user-friendly as pointed out on page 52, so the number of digits is best kept to four. The following configuration file is used for the

construction of the Speaker Identification system:

Configuration	chosen
Representation of feature vectors	LPC-Cepstrum Analysis
Type of Classifier	feed forward neural network (backpropagation)
Number of neurons in the hidden layer	80 neurons
Training iterations	1390.000 cycles
The number of digits	4

And with a note that when using the numbers: 3 ("drie"), 4 ("vier"), 5 ("vijf"), 6 ("zes") and 8 ("acht") will give a better recognition.

5.2 Recommendations

There are some points what can be looked into thoroughly, one could train the network (Feed Forward network) with numbers that are higher then nine. This will take another problem with it, for instance number as ten, eleven, twelve and thirteen should be included in the training set. One will have to look hard for these numbers on the Polygon database cd, which consists out of 500 speakers! When one wants to use words as input utterance then this will be a more difficult task, one cannot use short-term cepstral coefficients [Furui2][Markel], but long-term cepstrum coefficients.

Also the speaker set can be increased and this would results in a larger neural network (80 neurons in the layer will not be sufficient). Also one could use different types of neural network for speaker identification (Kohonen network or a binary network). A very intriguing approach is the combination of fuzzy logic and ANN. There are some research papers ([Yuan], [Yuan et al]) about this combination.

Bibliography

- [Anon] Voice identification research by Anon., Law Enforcement Assistance Administration, U.S. Department of Justice, Rep. PR 72-1, no. 2700-0144, 1972.
- [Atal] Automatic recognition of speaker from their voices by Atal. B, IEEE Transactionn ASSP vol.64. pp. 254-272 April 1976.
- [Assaleh] New LP-Derived Features for Speaker Identification by Assaleh K. T. and Mammone R. J., IEEE Transaction On Speech And Audio Processing, vol. 2, no. 4, october 1994.
- [Carbonell] Speaker authentication techniques by Carbonell J. R., BNN report 1296, Cambridge, MA, Bolt Beranek and Newman Inc., 1965.
- [Cheung] Feature selection via dynamic programming for text-independent speaker identification by Cheung R. S. and Eisenstein B., IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 397-403, 1978.
- [Ciobanu et al] Self-Organizing Maps versus Classical Vector quantization in Text-Independent Speaker Recognition by T. Ciobanu, S. Segărceanu, Dana Buscsa and I. Brailov, International Speech and Computers (SPECOM'98 St.-Petersburg. pp 239-244, 26-29 October 1998.
Our goal was to compare the efficiency of two methods of classification: The classical Vector Quantization and Kohonen's Self-Organizing Map.
- [Clarke] Comparison of techniques for discrimination among talkers by Clarke F.R. and Becker R. W., J. Speech Hearing Res., vol. 12, pp.747-761, 1969.

[Furui2] Research on individuality features in speech waves and automatic speaker recognition techniques by Furui S., IEEE Speech Communication vol. 5. Pp. 183-197 1986.

[Doddington] Speaker Recognition - Identifying People by their Voices by doddington G.R., Proceedings of the IEEE, VOL. 73, NO. 11, November 1985, pp 1651-1664.

This usefulness of identifying a person from the characteristics of his voice is increasing with the growing importance of automatic information processing and telecommunications. This paper reviews the voice characteristics and identification techniques used in recognizing people by their voices. A discussion of inherent performance limitations, along with a review of the performance achieved by listening, visual examinations of spectrograms, and automatic computer techniques, attempts to provide a perspective with which to evaluate the potential of speaker recognition and productive directions for research into and application of speaker recognition technology.

[Furui2] Research On Individuality Features In Speech Waves and Automatic Speaker Recognition Techniques by Furui S., Speech Communication 5 pp. 183-197, 1986. This paper presents an overview of Japanese research on individuality information in speech waves, which have been performed from various points of view. Whereas physical correlates having perceptual voice individuality have been investigated from psychological viewpoint, research from the engineering viewpoint is related to automatic speaker recognition, speaker-independent speech recognition, and training algorithms in speech recognition.

[Furui] Cepstral Analysis Technique for automatic Speaker verification by Furui S., IEEE Transaction ASSP vol.29. No.2 . April 1981

This paper describes new techniques for automatic speaker verification using telephone speech. The operation of the system is based on a set of functions of time obtained from acoustic analysis of a fixed sentence utterances. Cepstrum coefficient are extracted by means of LPC analysis successively throughout an utterance to form time functions and frequency response distortions introduced by transmission systems are removed.

[Gish et al] Text-Independent Speaker Identification by H. Gish and M. Schmidt. IEEE signal processing magazine pp 18-31 , october 1994.

- [Hattori] Text-Independent Speaker Recognition Using Neural Networks by Hattori H., ICASSP, pp 153-156, 1992.
- [Hecker] Speaker Recognition: An interpretative survey of the literature by Bolt R. H. et al., J. Speech Hearing Res., vol. 12, pp. 747-761, 1969.
- [Kersta] Voiceprint Identification by Kersta L. G., Nature, vol. 196, pp. 1253-1257, 1962.
- [Khaled] New LP-Derived Features for Speaker Identification by Khaled T. and R.J. Mammone, IEEE Transactionn ASSP vol.2. No.4 . October 1994
A new set of features is introduced that has been to improve the performance of automatic speaker identification systems. The new set of features is referred to as the A(daptive) C(omponent) W(eighting).
- [Klevans R.L, et al] Voice Recognition, Richard L. Klevans and Robert D. Rodman. isbn: 0-89006-927-1
- [Kohonen82] Self-Organised formation of topologically correct feature maps by Kohonen T., in Biological Cybernetics, vol. 43, 1982, pp. 59-69.
- [Kohonen82a] Clustering, taxonomy, and topological maps of patterns by kohonen T., in Proc. of the Sixth Int. Conf on Pattern Recognition, October 1982, pp. 114-128
- [Kohonen88] Learning vector quantization by Kohonen T., in Abstract of the First Annual Int. Neural Networks, San Diego, CA, vol I, July 1988, tutorial 10, p.13.
- [Lin Q, et al] Microphone Arrays and Speaker Identification by Lin Q. Jan E. and Flanagan J., IEEE Transaction on speech and audio processing, vol.2, no. 4, october 1994, pp. 622-629
The explores the influence of vector quantization techniques, codebook size, and order of cepstrum coefficients on the performance of the speaker identification system.
- [Lindberg et al] Text-prompted Versus Sound-prompted Passwords Speaker Verification Systems by J. Lindberg and H. Melin, ESCA Eurospeech97. Rhodes, Greece. ISSN 1018-4047, pp 851-854.
The problem of how to prompt a client for a password in an automatic,

prompted speaker verification system is addressed. Text-prompting of four-digit sequences is compared to speech-prompting of the same sequences, and speech-prompting of four digits is compared to speech-prompting of five digits.

[Makhoul] Linear prediction: A tutorial review. Proc. of the IEEE, vol 63 (4), 1975, pp. 561-580.

[Markel] Long-term feature averaging for speaker recognition by Markel J.D., Oshika B. and Gray H., IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, no. 4, pp. 330-337, 1977.

[Minsky] Perceptrons, by M. Minsky and C. Darken. MIT pres, Cambridge, MA 1969.

[Naik] Speaker Verification:A tutorial by J.M. Naik. IEEE Communication magazine pp 42-47, january 1990.
Personal Identity verification is an essential requirement for controlling access to protected resources.

[Nakagawa et al] Speech, Hearing And Neural Network Model, by Nakagawa S., Shikano K. and Tohkura Y. isbn 9051991789

[Nolan] The Phonetic Bases for Speaker Recognition by Noaln F., Cambridge, UK: Cambridge Univ. Press, 1983

[Oglesby] Speaker Recognition With A Neural Classifier by J. Oglesby and J.S Mason, IEE First int. Conf. on Artificial Neural Nets, pp. 306-309 1989.

Most current speaker recognition systems carry out training in an isolated manner in that each model has information relating only to one specific person This paper describes a new approach to speaker recognition where a neural classifier is used to separate the different speakers.

[Oglesby2] Optimisation of Neural Models For Speaker Identification, by J. Oglesby and J.S. Mason, IEEE Trans. on ASSP, pp. 261-264, 1990

A new approach to speaker recognition, based on feed-forward neural models, is investigated. Each person known to the system has a personalised neural net that is trained to be active for only that person's speech.

- [Pruzansky] Pattern-matching procedure for automatic talker recognition by Pruzansky S., J. Acoust. Soc. Amer., vol. 35, pp. 354-358, 1963.
- [Rosenblatt] Principles of Neurodynamics by F. Rosenblatt, New York Spartan Books, 1962.
- [Rosenblatt2] Automatic speaker verification: A review by Rosenblatt A. E., Proc. IEEE, vol 64, no. 4, pp. 475-487, 1976.
- [Rudasi et al] Text-Independent Talker Identification With Neural Networks by L. Rudasi and S.A. Zahorian. IEEE ICASSP pp. 389-392, 1991
This paper introduces a binary partitioned approach to classification which is applied to talker identification using neural networks.
- [Rumelhart et al] Learning representations by back-propagating errors, by D.E. Rumelhart, G.E. Hinton and R.J Williams. Nature, vol 332, 1986, pp. 533-536
- [Sambur] Selection of Acoustic Features for Speaker Identification by Sambur M. R., IEEE Transaction On Acoustic, Speech, And Signal Processing, april, pp.176-182, 1975.
- [Scofield] The Development of Selectivity and Ocular Dominance in a Neural Network by Scofield C.L., Ph.D. dissertation, Brown University, 1984
- [Shridar et al] Text-independent speaker recognition: A review and some new results by Shridhar M. and Mohankrishnan N., Speech Commun., vol. 1, pp. 257-267, 1982.
- [Stevens] Speaker authentication techniques by Stevens K. N., J. Acoust. Soc. Amer., vol 44, pp. 1596-1607, 1968.
- [Werbos] Beyond Regression: New Tools for Prediction and analysis in the Behavioural Sciences, by P.J Werbos. Ph.D. dissertation, Harvard University, 1974
- [Williams et al] Emotions and speech: Some acoustical correlates by Wilands C.E. and Stevens K. N., J. Acoust. Soc. Amer., vol. 52, pp. 1238-1250, 1972.
- [Wohlford] A comparison of four techniques for automatic speaker recognition by Wohlford R. E., Proc. ICASSP-80, pp. 908-911, 1980.

- [Yuan] A kind of fuzzy-neural networks for text-independent speaker identification, by Yuan Zhong-Xuan, Xu Bo-Ling and Yu chong-Zhi. IEEE Proc. ICASSP 1996, pp.657-660, 1996 (nr 0-7803-3192-3/96) A novel approach to establish membership functions of fuzzy states based on functional-link neural networks (FLNN) for text-independent speaker identification is proposed. A comparison of four techniques for automatic speaker recognition by Wohlford R. E., Proc. ICASSP-80, pp. 908-911, 1980.
- [Yuan et al] Text independent speaker identification using fuzzy mathematical algorithm by Yuan Zhong-Xuan, Yu Ching-Zhi and Fang Yuan. IEEE proc. ICASPP'93, pp II 403-406, 1993.