AN RBFN-BASED SYSTEM FOR SPEAKER-INDEPENDENT SPEECH RECOGNITION

by

Fakhralden A. Huliehel

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

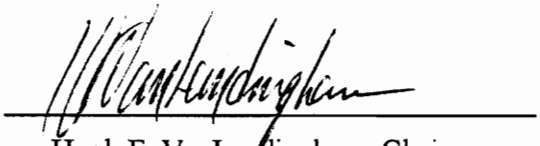
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

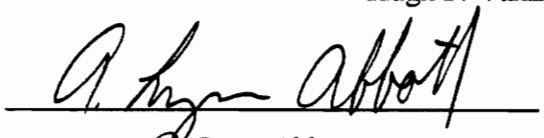
in

Electrical Engineering

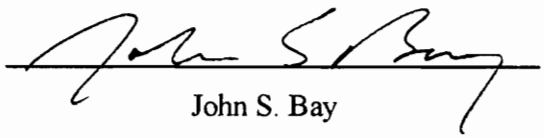
APPROVED:



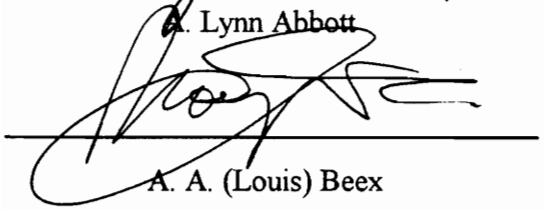
Hugh F. VanLandingham, Chairman



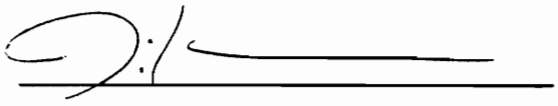
A. Lynn Abbott



John S. Bay



A. A. (Louis) Beex



Panickos N. Palettas

July 17, 1995

Blacksburg, Virginia

AN RBFN-BASED SYSTEM FOR SPEAKER-INDEPENDENT SPEECH RECOGNITION

by

Fakhralden A. Huliehel

Hugh F. VanLandingham, Chairman

Electrical Engineering

(ABSTRACT)

A speaker-independent isolated-word small vocabulary system is developed for applications such as voice-driven menu systems. The design of a cascade of recognition layers is presented. Several feature sets are compared. Phone recognition is performed using a radial basis function network (RBFN). Dynamic time warping (DTW) is used for word recognition. The TIMIT database is used to design and test the automatic speech recognition (ASR) system.

Several feature sets using mel-scale filter bank (MSFB), smoothed FFT, reflection coefficients (also called PARCORs), and cepstral features are extracted. The MSFBs outperform the other features considered in our study.

Multilayer perceptrons (MLPs) and radial basis function networks (RBFNs) are considered for phoneme recognition. RBFNs are easier to train than MLPs so that RBFNs were selected to perform phoneme classification.

Four RBFN's are compared: RBFN type-I is a single-layer RBFN, RBFN type-II is a two-layer net where the second layer consists of a vector of weights, RBFN type-III is a two-layer net where the second layer is a linear layer, and RBFN type-IV is a two-layer net where the second layer is a RBFN. RBFN type-II outperforms the others on the phone level where the phone recognition rate is about 44%.

Using clustering techniques, a suboptimal, iterative and interactive algorithm is developed to train the radial basis functions (RBFs). An algorithm is developed to reduce segmentation errors in TIMIT. The TIMIT 60 phone set is reduced to a 33 phone set by merging similar phones.

For 168 test speakers, 84% recognition rate is achieved on a vocabulary of 11 words from the sentence SA1 ("she had your dark suit in greasy wash water all year") in TIMIT. For applications such as voice driven menu systems, the vocabulary words can be selected to be separable and distinct. A 95% recognition rate is achieved when the confusing words in the 11 words vocabulary are excluded to get an 8-word vocabulary.

Real-time implementation of the proposed system can be achieved using a digital signal processor that can perform a multiplication within 100ns.

Acknowledgements

I would like to express my sincere gratitude to Dr. Hugh F. VanLandingham for accepting me as his student and opening the way to me to proceed in the speech and neural network research areas that are close to my heart. I also would like to thank Dr. VanLandingham for his enlightening academic guidance and valuable support during the last two years.

I would like to express my sincere gratitude to Dr. A. Lynn Abbott, Dr. John S. Bay, Dr. A. A. Beex, and Dr. Panickos N. Palettas, for serving on my advisory committee. In particular, I would like to thank Dr. Palettas for his valuable discussions about the statistical aspects of my dissertation.

I would like to thank the system managers of the workstation lab for their help in installing and maintaining the TIMIT CD-ROM, in particular Farooq Azam has been very helpful and nice.

I would like to express my sincere gratitude and thanks to my parents and my oldest sister Gazali for without their support this work would not have been possible.

I would like to express my sincere gratitude and thanks to my sister Khadra and her family for their caring and support during both my graduate and undergraduate programs.

My gratitude is also extended to Tammy Jo Hiner for her caring and moral support during my stay at Virginia Tech.

Table of Contents

Chapter 1: INTRODUCTION

1.1 Background and Motivation

A speech to text system (STTS) is a speech recognition system. Speech is one of the most natural forms of human communication. Hence, the purpose of automatic speech recognition (ASR) technology is to provide an easy and hands-free man-machine communication tools. The speech processing technology is relatively new. Significant advances in the field of speech processing have been achieved during the last three decades since the advent of digital computing in the 1960s [1]. Speech technology is a multi-disciplinary field that involves areas such as signal processing, electronics, computing science, linguistics, physiology, and psychology [1, 2].

The main areas in speech technology are speech analysis and processing, speech enhancement, speech transmission and communications, speech coding and compression, speech recognition, speech synthesis, speaker identification and verification [3].

The ideal speech recognition system would have the capability to recognize unlimited continuous speech utterances from any speaker of a given language with human accuracy. One of the main applications of speech recognition is voice input to computers for tasks such as speech to text and word processing (also called the talkwriter), database interrogation, voice control and command. Currently, the ASR technology is far from reaching human speech recognition capabilities. However, ASR systems are capable of recognizing isolated words from a limited vocabulary with reasonable accuracy. The performance degrades significantly when the size of the vocabulary is increased, the speech is continuous, or the number of speakers is increased. However, the performance of ASR systems is improving and currently there are some commercial products and applications. For example, the IBM Tangora speaker-dependent, isolated words voice type writer with vocabulary size of 20,000 has 94.6 % accuracy, and the AT&T Bell Labs speaker-independent connected speech recognizer with 11 word vocabulary has 99.6% accuracy [3].

Automatic speech recognition is basically a pattern classification task. The input to the ASR system is a speech signal waveform that consists of a sequence of speech units or patterns (i.e. words, sub-words, phonemes, etc.) that have to be classified. As will be presented in the next chapter, the ASR task is typically accomplished by two steps. First, the input signal is processed and segmented into a sequence of frames and certain features and characteristics are extracted for each frame (see the next chapter for more details about the feature extraction step). Second, the pattern classification is performed using the feature sequence obtained during the first step.

There are four main solutions for the pattern classification problem:

1. statistical modeling, such as hidden Markov modeling (HMM),

2. template matching using dynamic programming and dynamic time warping (DTW),
3. neural networks, and
4. knowledge-based systems.

A review of the four classification approaches is provided in the next chapter.

In the statistical approach to speech recognition, it is assumed that speech can be modeled by a statistical model; the hidden Markov model. However, the assumption of strictly Markov model is a rough approximation. Despite the approximations and invalid assumptions in the HMM method, it has been most successful in several difficult speech recognition applications [4].

In the template matching approach to speech recognition a typical template or characteristic pattern of each expected input is created. The pattern recognition is performed by selecting the closest reference pattern to the input speech. Because the input speech can have different length compared to the reference templates, the distance between the input and the template is measured using the dynamic time warping (DTW) method which is a widely used and efficient dynamic programming procedure for time alignment. Connected word recognition and a degree of speaker independence can be achieved using the template matching approach. The main limitation of the template matching approach is that each template represents only one speech sound because this approach is a deterministic one.

In recent years the neural network or connectionist approach has been applied to the speech recognition problem. Neural network classifiers which are biologically motivated are non-parametric, adaptive and robust pattern classifiers that have the ability to form complex decision regions. The multi-layer perceptron network (static or dynamic

network) is a popular neural network that has been applied to pattern classification using supervised back-propagation training algorithms. Recently, several neural network classifiers, with supervised and/or unsupervised training, have been applied to isolated word, sub-word (especially phoneme) and connected speech recognition systems. Some of the results obtained so far are not quite as good as that of DTW or HMM, however, there are some promising results that are as good or better than these from the HMM or DTW approaches. The research is still in its early stages, and there are still open research issues about the optimum network topology and the training algorithm and set-up [1, 4, 5].

In the knowledge-based system approach, a role-based expert system is applied to the speech recognition task. The results obtained by this approach are not competitive with the pattern matching techniques yet.

1.2 The Research Objectives

The research objective is to study the feasibility of applying neural network technology to the ASR problem. The ultimate goal of the research is to design an isolated-word, speaker-independent speech-to-text translator. The feasibility of the system will be demonstrated over a small vocabulary.

The final system is designed for applications such as voice-driven menu systems where a tree-like word recognition system is adopted (see Fig. 1.1 for example). At each level of the tree there is a small vocabulary (less than 20 words). It is assumed that the words at each level are well separated. The ASR system developed in our research can be used to

perform word recognition at each level of the tree (less than 20 words). The design and development of a full menu system is outside the scope of our research.

The main issues considered in this research are :

1. selection of speaker-independent features,
2. the neural network topology,
3. the training algorithm,
4. the training set-up and strategy,
5. system performance assessment.

1.3 Dissertation Outline

In the next chapter theory and previous work are reviewed. In Chapter 3 feature extraction and analysis of speech signals is performed. In Chapter 4 an RBFN-based system for speaker-independent speech recognition is developed. And in Chapter 5 the main conclusions of the research are presented.

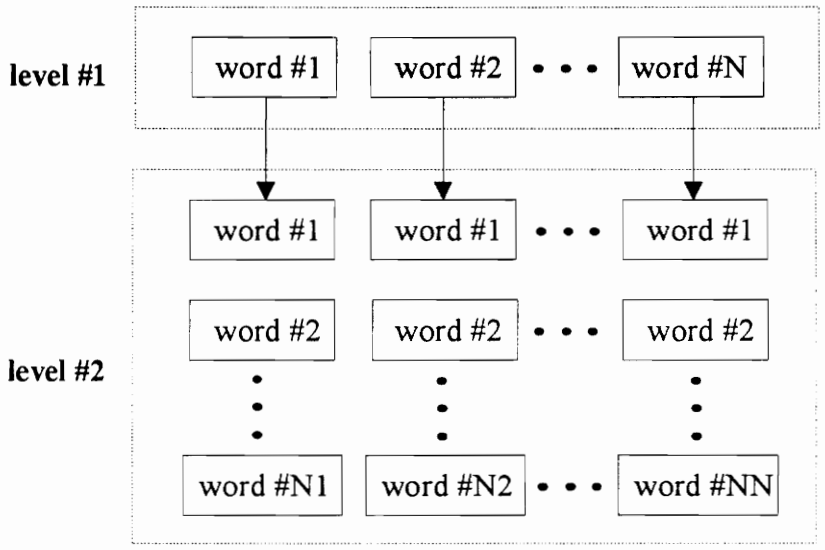


Fig. 1.1 An example of a tree-like word recognition system.

Chapter 2: THEORY AND PREVIOUS WORK

2.1 The Problem of Speech-To-Text Translation

The STT (speech-to-text) problem is a speech recognition problem. As shown in Fig. 2.1, the input to the STT system is a time waveform of a speech signal and the output is a sequence of text characters or words. The speech recognition task is typically performed in two steps (Fig. 2.1). First, the speech signal is processed and certain features and characteristics are extracted. Second, speech patterns or sounds are recognized. The design of a STT system requires understanding of the nature of speech signals, their auditory representation, and the problems and complexities associated with automatic speech recognition. In this section, a brief summary of the nature of speech signals, and the human auditory system is provided.

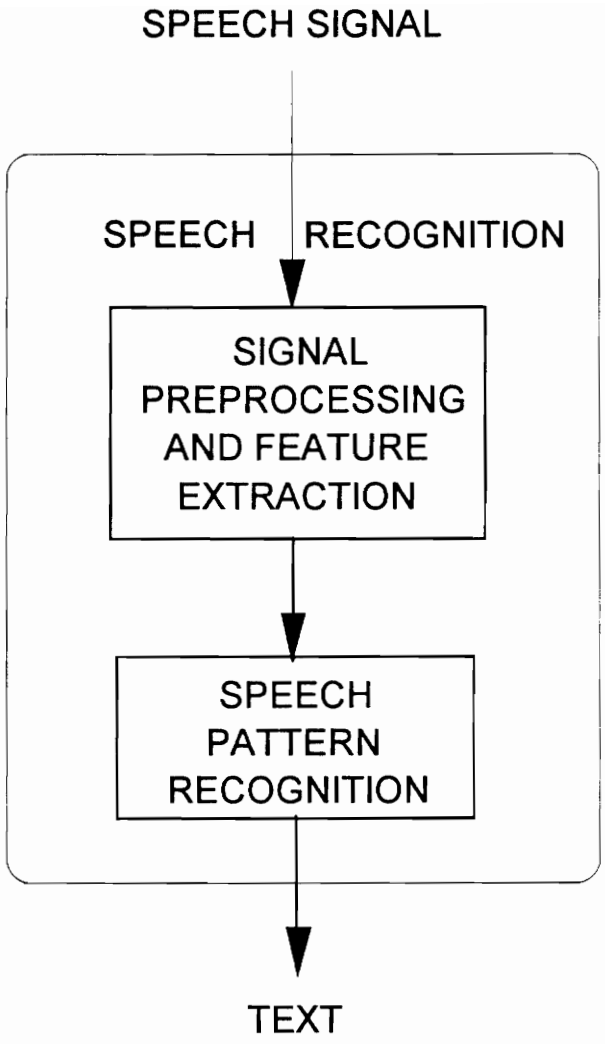


Fig. 2.1 Speech-to-text system.

2.1.1 The Nature of Speech

2.1.1.1 Symbolic Representation of Speech Sounds [1, 2]

One way of representing speech is the letter symbols of writing, however, the letter symbols of writing are unsuitable when they are used to represent different sounds in different contexts. For example the letter 'o' is pronounced differently in the words 'one' and 'bone'.

A convenient unit for representing speech sounds is the 'Phoneme'.

Phonemes

The phoneme is a linguistic unit defined such as if one phoneme is substituted for another in a word, the meaning of that word may be changed. For each language there is one set of phonemes, however, there is much overlap of the phoneme sets of all languages, and the total number of phonemes is limited. A consistent set of phoneme symbols for languages is provided by the International Phonetic Alphabet (IPA).

The phonemes of British English are shown in Fig. 2.2 and examples of words in which they are used are given in Table 2.1. The pure vowels are subdivided into front, middle, or back depending upon the position of the tongue hump when they are produced.

Allophones

Allophones (of a phoneme) are phones that correspond to different realizations of the

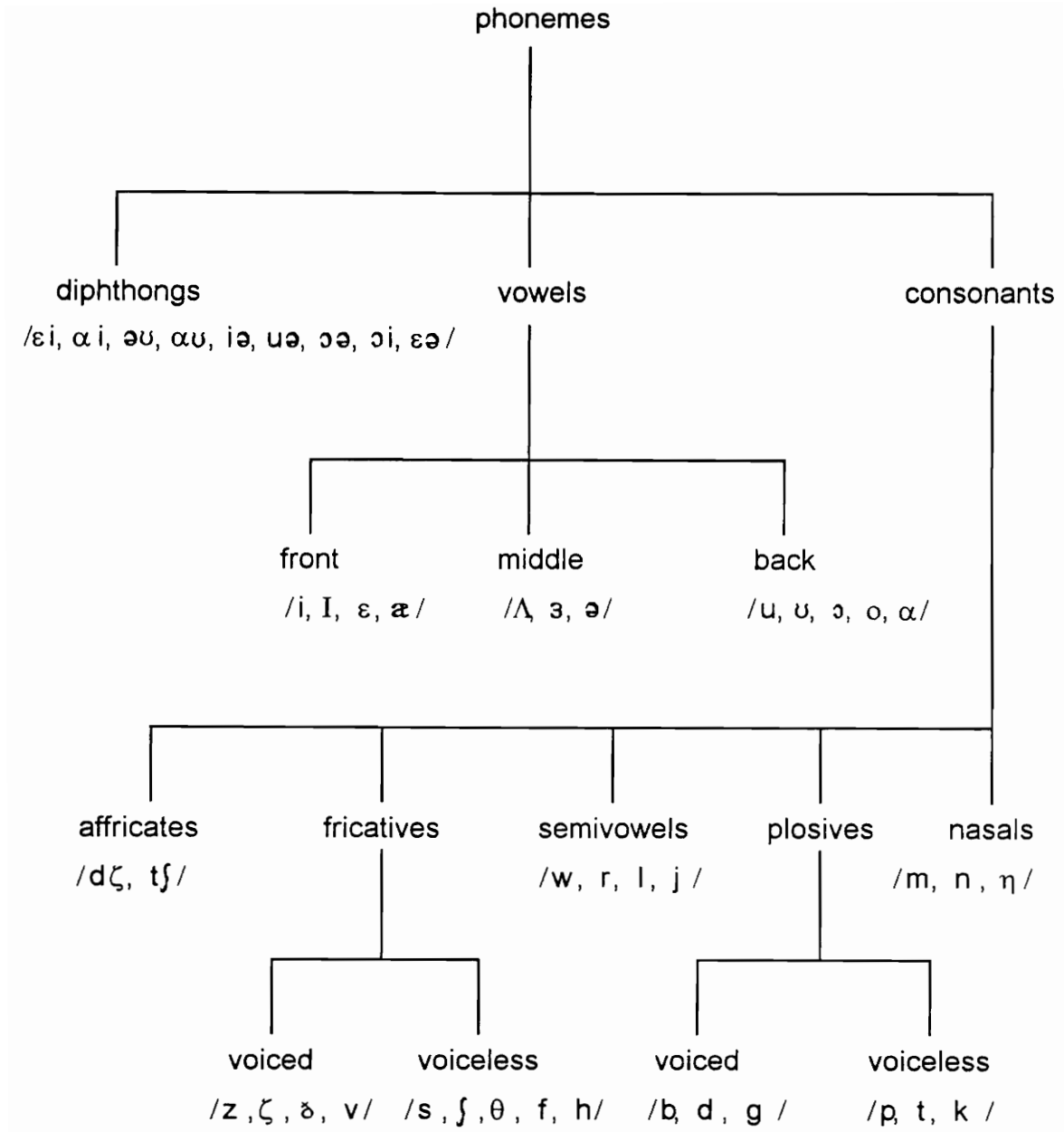


Fig. 2.2 Classification of the phonemes of British English.

Table 2.1 The International Phonetic Alphabet (IPA) symbols of the phonemes of British English and examples of words in which they are used.

Vowels		**Diphthongs**		**Semivowels**		**Nasals**	
/i/	feed	/ɛi/	ray	/w/	was	/m/	am
/ɪ/	did	/ɑi/	by	/r/	ran	/n/	an
/ɛ/	bed	/əʊ/	row	/l/	lot	/ŋ/	sang
/æ/	dad	/ɑʊ/	bough	/j/	yard		
/ɑ/	car	/iə/	deer			**Fricatives**	
/o/	rod	/uə/	doer	**Plosives**		/s/	sam
/ɔ/	road	/ɔə/	boar	/b/	bar	/ʃ/	ship
/ʊ/	wood	/ɔi/	toy	/d/	disc	/f/	fan
/u/	rude	/ɛə/	bear	/g/	goat	/θ/	thin
/ʌ/	but			/p/	poor	/h/	hum
/ɜ/	heard	**Affricates**		/t/	task	/z/	zoom
/ə/	the	/dʒ/	jam	/k/	kid	/ʒ/	azure
		/tʃ/	chum			/ð/	then
						/v/	van

same phoneme depending on the context, where phones are units for classifying speech sounds in terms of the way in which they are produced. There are many more allophones than phonemes. For example, the velarised 'l' occurs before consonants and at the end of utterances. The non-velarised 'l' occurs in other positions in English. Both are allophones of the phoneme 'l'.

Transcription

A transcription of an utterance is a string of symbols that represents the utterance. There are two levels of transcription: phonemic (or broad) transcription or phonetic (or narrow) transcription. The phonemic transcription consists of a string of phonemes. The phonetic transcription consists of a string of phones (allophones). The speaker needs to know the phonology of the language (i.e. he needs to know which allophone of each phoneme to employ in which context) to reproduce the utterance from its phonemic transcription .

Prosodics

The prosodic features of stress, rhythm, and intonation affect the way in which an utterance is spoken.

'Sentence stress' indicates the most important words in a sentence. 'Word stress' indicates the most important syllables in a word. The IPA allows three levels of stress to be recorded. ' is placed before a syllable with primary stress, and a blank space is placed before a syllable with secondary stress. Unstressed syllables are not marked.

Rhythm refers to the timing of an utterance. When the durations between stresses are equal the language is stress-timed. English is claimed to be a stress-timed language. Objective measurements have shown that there is merely a tendency in this direction. Other languages are syllable-timed, such as French.

Intonation, or pitch movement, affects the meaning of the sentence. The IPA provides symbols for representing high level, low level, high rising, low rising, high falling, and low falling pitch.

2.1.1.2 The Mechanism of Speech Production [1]

The vocal apparatus is shown in Fig. 2.3. In speaking, the lungs are filled with air by expansion of the rib-cage and lowering of the diaphragm. As the rib-cage contracts, air is expelled and is forced along the trachea through the glottis. The flow of air is the source of energy for speech generation. The vocal cords can be made to repeatedly blow apart and flap together as air is forced through the slit between them which is called the glottis. The oral tract is a non-uniform acoustic tube, approximately 17 cm long in an adult male, terminated at the back by the vocal cords or larynx. Its cross-sectional area can be varied from zero to about 20 cm^2 by muscular control of the speech articulators (lips, tongue, jaw, and velum). The nasal tract is a non-uniform acoustic tube of a fixed area and length (about 12 cm in an adult male). It is terminated at the front by the nostrils, and at the rear by a movable flag of skin, called the velum, that controls the acoustic coupling between the oral and the nasal tracts. The velum seals off the nasal tract during the production of non-nasalized speech. When nasalized speech is produced the velum is lowered and the nasal tract is acoustically coupled to the oral tract. However, in this situation the front of the oral tract is completely blocked and there is only one transmission path via the nostrils.

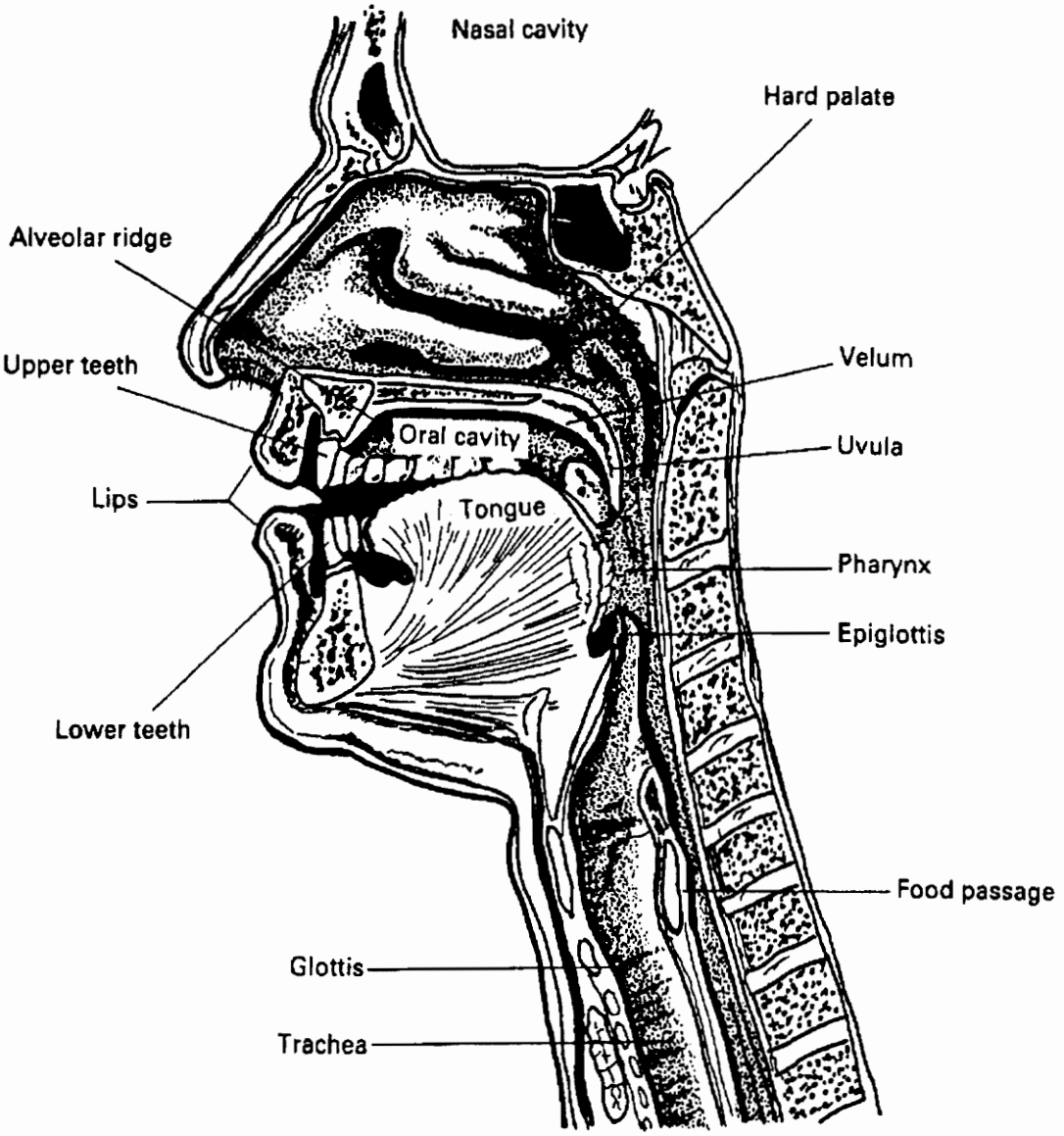


Fig. 2.3 The vocal apparatus diagram [1].

The speech sounds can be divided into three classes according to the mode of excitation.

Voiced Sounds

The excitation in voiced sounds is periodic or at least quasi-periodic. The vibrations of the vocal cords produce an airflow waveform which is approximately triangular and periodic (a train of glottal pulses is produced). The spectrum of the generated waveform is rich of harmonics of the fundamental frequency, which is called the pitch frequency, and decays at a rate of approximately 12 dB/octave. The vocal tract acts as a resonance cavity that amplifies some of the harmonics and attenuates the others. The range of the pitch frequency is from about 50 Hz to about 250 Hz, with an average value of about 120 Hz for adult males. For an adult female the pitch frequency can reach as high as 500 Hz.

Voiceless Sounds

The excitation for unvoiced sounds is a random noise source. A point of constriction is created at some point of the vocal tract, and as air is forced past it, turbulence occurs that causes a random noise excitation. For fricatives (such as 's'), the constriction point is created near the front of the mouth, so that the vocal tract resonances have little effect on the features of the fricative sounds. In aspirated sounds (such as 'h'), the excitation is generated at the glottis, so that the vocal tract resonances modulate the spectrum of the random noise. The modulation effect is clearly heard in the case of whispered speech.

Plosive sounds

The plosives (such as 'p' and 'b') have a transient excitation. For the plosive sounds, the vocal tract is closed at some point, the air pressure is allowed to build up and then suddenly released. The rapid release of this pressure provides a transient excitation of the vocal tract. The transient excitation may occur with or without vocal cord vibrations to produce voiced (such as 'din') or voiceless (such as 'pin') plosive sounds.

2.1.1.3 Source-Filter Model of Speech Production [1]

The speech production system can be approximated by a source-filter model (Fig. 2.4). The sound source is either a periodic pulse train in the case of voiced sounds or a random noise in the case of unvoiced sounds. The vocal tract is modeled by a time-varying filter.

The voiced source has a high frequency roll-off of approximately -12 dB/octave. The unvoiced source spectrum is relatively flat and broad band. The gains A_V and A_N control the intensity of the sound. Voiced sounds have higher intensity than voiceless ones. The vocal tract has an infinite number of resonance frequencies (also called formants). However, due to the lowpass property of the vocal tract, the filter order can be reduced to a few formants. For voiced sounds the significant frequency range is 100 Hz to 3-4 kHz. For voiceless sounds the frequency range of interest is extended up to 7-8 kHz. The filter model also includes the effect of radiation from the mouth. The effect of the acoustic radiation impedance can be approximated by a first order high-pass characteristic increasing at a rate of 6 dB/octave in the range of 0-3 kHz.

The filter-source model is a simplification of the speech production system. Even though there is some coupling between the sound sources and the vocal tract, it isassumed

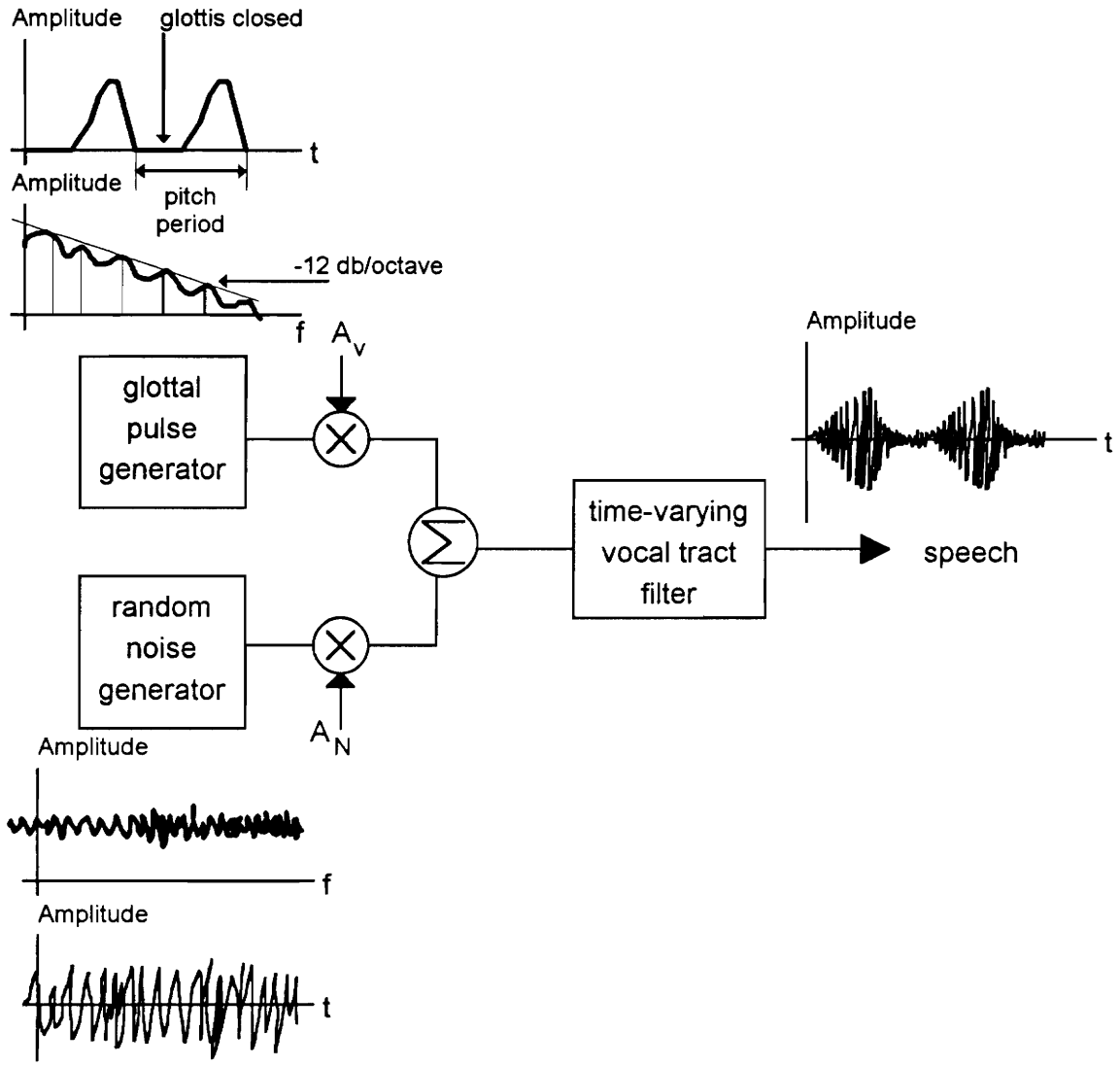


Fig. 2.4 Source-filter model of the speech production system.

that the source and filter are linearly separable and that there is no interaction between them. However the coupling between the source and vocal tract is secondary. The fricative sounds are not filtered by the vocal tract, so that the source filter model is not an accurate model for fricative production. However, very often, these secondary factors are ignored and the source-filter model is adequate.

2.1.2 The Human Auditory System

2.1.2.1 The Peripheral Auditory System [1]

The periphery of the auditory system is illustrated in Fig. 2.5. The ear is divided into three main regions - the outer ear, the middle ear and the inner ear. The outer ear consists of the pinna and the auditory canal or meatus, that leads to the eardrum or tympanic membrane. Sound waves impinge upon the eardrum and make it vibrate. The typical deflection of the eardrum is a few nanometers (very sensitive).

In the middle ear, a small bone called the hammer (malleus) is attached to the eardrum. When the eardrum moves, the hammer makes contact with another bone called the anvil (incus), causing it to rotate. The anvil is connected to another small bone called the stirrup (stapes), which is attached to the oval window of the inner ear. The three bones in the inner ear are the smallest in the human body and are called ossicles. Their function is to transmit the vibrations of the eardrum to the oval window of the inner ear.

The oval window is a membrane-covered opening in the bony wall of the spiral-shaped structure called the cochlea. The fluid-filled cochlea is divided along its length by two

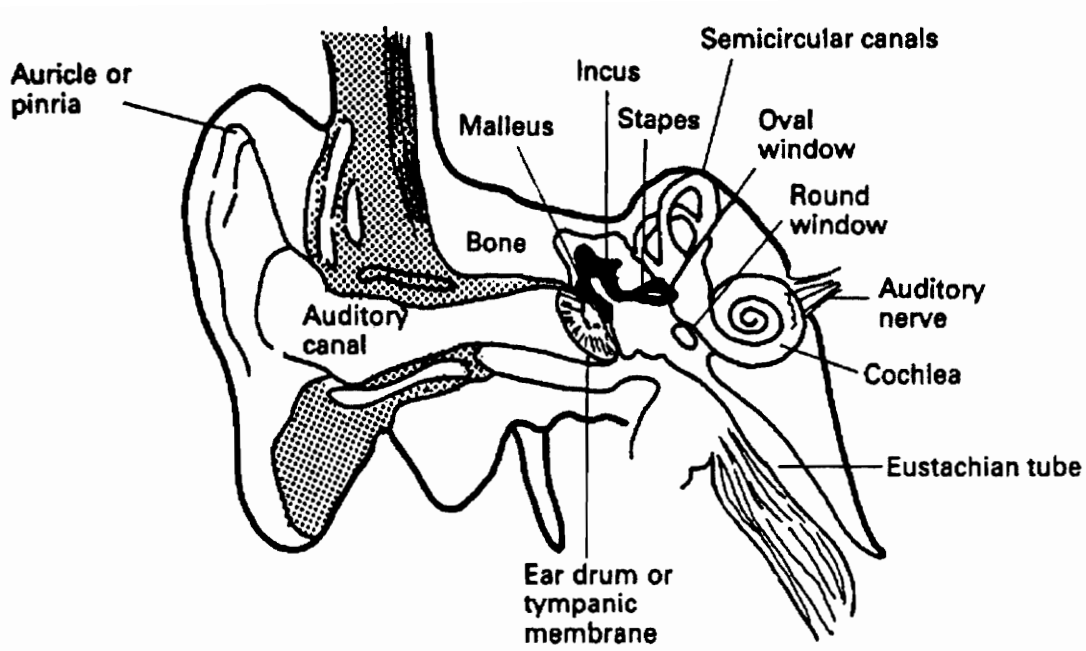


Fig. 2.5 The peripheral auditory system [1].

membranes, Reissner's membrane and the basilar membrane. The oval window vibrations result in pressure waves that propagate through the cochlear fluid and cause the basilar membrane to deflect at different points along its length. The basilar membrane is connected to the Corti organ that is a jelly-like organ which contains around 30,000 hair cells arranged as three rows of outer cells and one row of inner cells. Each hair cell has many tiny hairs protruding from it. The motion of the basilar membrane bends the hairs creating action potentials in the hair cells. The hair cells are connected to the nerve-endings (dendrites) of the neurons of the auditory nerve. The action potentials of the hair cells result in neural-firing (a series of electrical impulses) that are transmitted via the auditory nerve to the brain where speech perception takes place.

2.1.2.2 Processing of Acoustic Signals in the Auditory System [1, 6]

The auditory system processes sound signals applying several complex transformations. Several modeling efforts and studies of the auditory system exist. However, all the models can be reduced to three stages: analysis, transduction, and reduction as shown in Fig. 2.6 [6].

The Analysis Stage

As mentioned in the previous section, the sound waves impinging upon the eardrum result in pressure waves in the cochlear fluid, and the pressure wave produces mechanical displacements of the basilar membrane. There are two equivalent ways of viewing the patterns of the basilar membrane displacements [6]. The first is to focus on their spatial distribution along the length of the cochlea. The vibrations caused by a single-tone wave

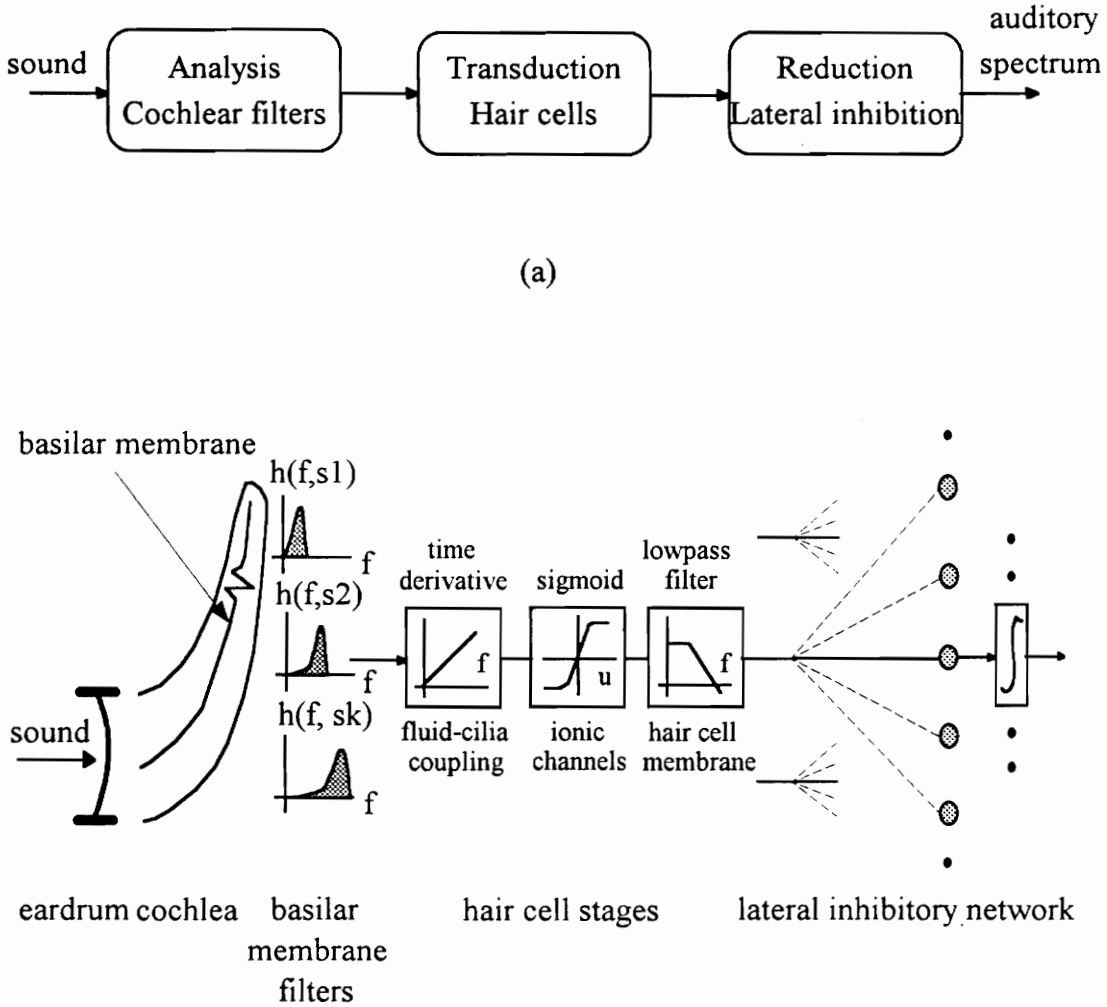


Fig. 2.6 Early stages of processing in the auditory system. (a) Block diagram of the three basic auditory processing stages. (b) Quasi-anatomical sketches of the auditory stages.

appear as traveling waves that propagate up the cochlea (from the base to the apex), reaching a maximum amplitude at a particular point before slowing down and decaying rapidly. The point at which the maximum displacement occurs depends on the frequency of the tone, with lower frequencies propagating further towards the apex of the cochlea. In this way, the cochlea segregates incoming frequencies onto different spatial locations in a tonotopically ordered manner along its length.

The second is to view the cochlea as a parallel bank of bandpass filters [1,6]. At each point along the basilar membrane, the displacement can be measured as a function of the tone frequency, i.e., a transfer function. In the cochlea, the transfer functions are moderately well-tuned, with center frequencies decreasing towards the apex of the cochlea. Above 800 Hz, the impulse responses of these 'filters' are related to each other by a dilation. Consequently, along a logarithmic frequency, the transfer functions appear approximately invariant except for a translation, i.e., they maintain constant Q-factor [1,6]. Therefore, the output of the cochlear filters can be viewed as an affine wavelet transform of the stimulus, and the continuous spatial axis of the cochlea as the scale parameter axis [6].

The above filter-bank and wavelet model of the cochlear processing is an approximate model. Several nonlinearities that enhance the sensitivity and tuning of the cochlear filters at lower sound levels have been ignored. The actual frequency scale of the cochlea is not purely logarithmic, especially below 500 Hz, but rather becomes progressively more linear. Measurements have shown sharper frequency selectivity than the frequency-selectivity curves of the basilar membrane filters [1]. Experimental measurements were performed to find the relation between the filter bandwidth and center frequency. It has been found that if a masking tone falls inside a certain frequency band, then the perception of the frequencies within this band will be affected by the masking tone. These frequency

bands are called critical bands. The critical bands of a wide range of frequencies have been determined from psycho-acoustic experiments and are plotted in Fig. 2.7. The critical bands in Fig. 2.7. are much narrower than those suggested by the observed vibrations on the basilar membrane.

The filter-bank model of the basilar membrane and the results in Fig. 2.7. are often the basis for the design of filter banks for speech signal processing.

The Transduction Stage

In this stage, the mechanical vibrations of the basilar membrane are transduced into electrical activity. At each point, membrane displacement causes a local fluid flow that bends the inner hair cells. The bending of the hair cells controls the flow of ionic currents through nonlinear channels into the hair cell. The ionic flow results in electrical potentials across the hair cell membranes. The electrical potentials are conveyed by the auditory nerve fibers to the central auditory system

The transduction stage can be approximated by three stages as in Fig. 2.6. [6]. First, a velocity coupling stage that can be modeled by a time derivative. Second, the ionic channel can be described by an instantaneous nonlinearity modeling the opening and closing of the ionic channel. Third, a lowpass filter (with a relatively short time constant, < 0.3 ms) can be used to model the ionic leakage through the hair cell.

The above transduction model ignores the effects of the adaptive mechanisms operative at the hair cell-auditory nerve junction, that might be significant in describing the responses to the onset of sound. They have been found useful in some phonetic segmentation algorithms.

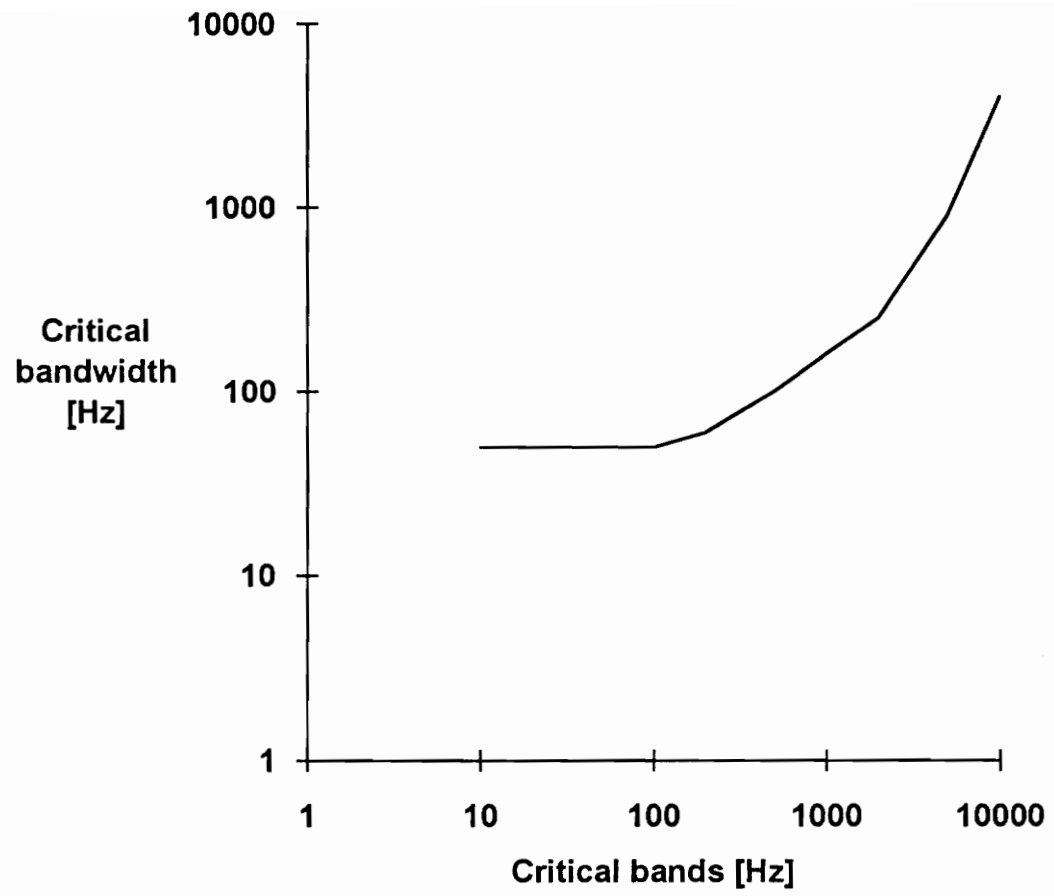


Fig. 2.7 The measured critical bands of the cochlear filters.

The Reduction Stage: Spectral Estimation

The auditory nerve transmits the electrical activity from the hair cells to the cochlear nucleus of the central auditory system. Several features and characteristics (such as pitch, timbre, location in space, etc.) are extracted and processed along parallel pathways. We elaborate only on the short-time spectral estimation of the auditory system due to its significant role in the recognition of different sounds and other fundamental auditory tasks.

The spectral estimation is implemented biologically by a neural network known as the lateral inhibitory network (LIN). The simplest model of the LIN consists of a one layer of nonlinear neurons that are mutually inhibited either in a feedback or a feedforward manner [6]. The LIN can be modeled by a three stage system. The first stage is a derivative with respect to the spatial axis of the cochlea. The spatial derivative models the lateral inhibitory influences among the LIN neurons. The second stage consists of half-wave rectifiers that model the nonlinearity of the neurons. The third stage is a long time-constant, 10-20 ms, integrator. This integration is applied to represent the fact that the central auditory neurons are unable to follow rapid temporal modulations (i.e. higher than a few hundred Hertz). Instead, they output a temporally integrated version of their instantaneous output.

It can be shown that the final output of the LIN approximately reflects the short-time amplitude spectrum of the sound stimuli. This can be called the final auditory representation of the sound signal. Experimental tests with automatic speech recognition systems have consistently shown that the auditory representation preserves all spectral information and may highlight more perceptually useful features [6].

In speech processing systems that are based on the human auditory system models, the auditory spectrum is strongly affected by the filter-bank parameters, so that it is possible

to adjust the design to meet the needs of the task at hand. The number of bandpass filters and their bandwidths could be chosen to get adequate accuracy of the sound signals' spectra.

2.1.3 Problems of Speech Recognition and Speech-To-Text Translation [2, 7]

When designing an ASR system, several problems need to be addressed. The optimal solution (if it does exist) for many of these problems depends on the application. In this section the common problems in ASR are considered. However, the problems that are significant to speech-to-text systems are emphasized.

The basic problem in ASR is that speech consists of a continuous stream of sounds without clear boundaries between the words, and yet the speech is perceived as a sequence of separated words. The problem is to segment the speech into linguistic units and to recognize these units. This problem is complex due to the wide variations of the acoustic signals that are accepted by the human listener as examples of the same speech unit [2].

2.1.3.1 Recognition Units [2]

The first step in solving the recognition problem is to choose the recognition unit. The choice of a recognition unit is a trade-off problem, and depending on the application, the designer should make his decision. Possible recognition units are word, syllable, demisyllable, diphone, allophone, phoneme, and distinctive features.

The main problem of using words as recognition units is the large number of words in the language. This is of the order of 100,000 words in English. The large number of words creates problems such as storage, slow recognition speed, lower recognition accuracy, and the sheer amount of effort and time to record and pronounce the recognition units. However, there are several applications that require the recognition of a limited number of words. In these applications words can be applied as the recognition unit.

Instead of using words, smaller units such as syllables may be considered. The advantage of employing syllables as recognition units is the reduction of the number of recognition units. There are about 10,000 syllables in English. However, it is more difficult to segment the speech stream into syllables than to segment it into words.

Further reduction in the number of units can be achieved by using the demisyllable as the recognition unit, which consists of half a syllable. There are about 2000 demisyllables in English.

Another recognition unit is the diphone. In English there are 1000-2000 diphones. However, the segmentation problem is much more difficult than in the case of syllables.

Phonemes could be used as recognition units. The main advantage of phonemes is their small number. There are only 40 to 60 phonemes. However, phonemes have a number of contextual variations known as allophones. There about 100-200 allophones. The very small number of phonemes makes them attractive recognition units. However, it is very difficult to detect the end points of phonemes, and as a result, the segmentation problem is difficult.

2.1.3.2 Variability [2]

There are many factors that cause variability in speech. These include the speaker, the context, the environment, and the transducer employed. The variability of the speech introduces higher recognition error rates and increases the ASR problem complexity.

The speaker's physical characteristics, sex and age affect the speech signal features. The vocal tract has a different size and characteristics depending on the speaker. This causes variability in the speech features, especially the formant frequencies. Even people from different parts of the country, or from different social and economic backgrounds speak with different dialects. This involves substitution of one phoneme for another in certain words. Second language speakers introduce a lot of variability, especially in the prosody and phoneme set. Even the same speaker produces different speech patterns of the same word on different occasions.

Co-articulation effects cause each word to be spoken differently depending upon context. Words also are pronounced differently depending on their position in the sentence and their degree of stress.

The speaking rate causes variability, especially in the duration of each phoneme. Moreover, the durations of all sounds in fast speech are not reduced proportionally compared to their duration in slow speech.

The amplitude of speech varies widely depending on the situation, topic, the emotional state, and the distance between the microphone and the speaker.

An additional source of variability in speech is the environment. Noise and background sounds, microphone and recording equipment also can cause variability in the speech signal.

The parametrization of speech introduces variability. The sampling rate, the quantization, and the pre-sampling filtering cause variability in the speech signal.

2.1.3.3 Ambiguity [2]

The problem of ambiguity is significant when the system is required to perform some actions as a result of the speech input that it receives. Ambiguity is caused by several factors, such as homophones, overlapping clauses, word boundaries, syntax, and semantics. Homophones are words that sound alike even though they have different spelling and meaning (for example 'to' , 'too' and 'two').

2.1.3.4 Type of Recognizer [7]

The input to a speech recognizer can be isolated-words or continuous speech. And the ASR system can be a speaker-independent or a speaker-dependent system. The ideal ASR is continuous-speech, speaker-independent system. Isolated-word recognition systems are cheaper and more accurate than continuous speech recognition systems. However, many people would prefer continuous speech recognition systems.

A speaker-independent system would not require enrollment, but it would pay a price in accuracy. Any speaker-dependent system requires some form of enrollment. For large vocabulary, and multi-user systems, the enrollment process is time consuming and tiresome. However, the recognition accuracy is higher and the price is less in the case of speaker-dependent than in the case of speaker-independent systems.

2.1.3.5 Vocabulary Size [7]

The vocabulary size is determined by the application. Some applications require a small vocabulary and others require a large vocabulary. Large vocabulary systems are more expensive and less accurate than small vocabulary systems. About 75% of typical text is covered by the 1000 most frequently used words. About 95% of text is covered by the 5000 most frequently used words. English has about 100,000 words. Increasing the vocabulary size does not increase the accuracy unless the recognizer is improved. In speech-to-text systems (or talkwriter) the vocabulary size is a trade-off problem, and different solutions will result in different price and performance of the talkwriter.

2.1.3.6 Speed and Accuracy [7]

The acceptable speed and accuracy of an ASR system depend upon the application. Some applications require very high accuracy and real-time speed. For speech-to-text systems, real-time recognition is desirable. Two to three times slower than real-time system can compete with conventional typewriters or word processors.

The accuracy of a system is difficult to estimate and it depends on several factors such as the speaker, the vocabulary size, the vocabulary itself, the context, the mode of speaking, the enrollment strategy, noise, syntax, and the time allowed for processing. One way of estimating the accuracy of a speech-to-text system is to compare it with conventional typewriters or word processors. An acceptable error rate for a talkwriter is about 10-15 %.

2.2 Preprocessing and Feature Extraction of Speech Signals

As was mentioned in the introduction, the speech recognition task consists of two main steps. First, the speech signal is processed and certain features and characteristics are extracted. Second, speech units are identified using the features and analysis results obtained in the preprocessing and feature extraction step. In this section the common processing and feature extraction techniques are summarized. The filter bank method for estimating the short-time spectrum of a speech signal is emphasized because it is suitable for neural network classifiers.

2.2.1 Discretization of Speech Signals

Most of the analysis and processing techniques of speech signals are digital ones. The speech signal has to be converted from analog to digital form. The discretization of an analog signal consists of two processes: the sampling and the quantization processes.

2.2.1.1 Sampling of Speech Signals [1, 2]

The main parameter that has to be determined is the sampling frequency. The sampling frequency must satisfy Nyquist's sampling theorem, which states that the sampling

frequency must be higher than twice the highest frequency component in the sampled signal to prevent the aliasing effect from showing up.

For speech signals the highest frequency component is not distinctly known. Therefore, it is necessary to use a lowpass filter, which is called the anti-aliasing or pre-sampling filter, to band limit the signal prior to sampling.

In ASR the frequencies of interest are those of the range of the human hearing. This range is about 20 Hz to 20 kHz, although the upper limit diminishes progressively with the age of the listener. A sampling frequency of about 40 kHz and an anti-aliasing filter of about 20 kHz is therefore more than adequate. However, most of the energy in speech sounds lives below 5 kHz, even though some of the fricative sounds have energies up to 10 kHz. Depending on the application, the sampling frequency can be in the range of 8 kHz to 20 kHz, and the pre-sampling filter bandwidth can range from 3.5 kHz up to 10 kHz [2].

2.2.1.2 Quantization [1, 2]

The second parameter which should be considered in the discretization process is the accuracy with which the signal is sampled. The accuracy of sampling is determined by the number of bits of the A/D converter. The intensity range of human hearing, from the threshold of hearing to the threshold of pain, is about 120 dB. The accuracy required to cover the human hearing range is then about 20 bits (because 120dB is about equal to $20\log_{10}(2^{20})$). However, speech signals range over about 70 dB, which corresponds to 12 bit accuracy. In this study, a 16-bit A/D converter is used.

Depending on the application, the sampling accuracy can be 8 to 16 bit if uniform sampling and quantisation, or pulse-code-modulation (PCM) is used. However, further

reduction of the number of bits can be achieved if other quantisation schemes, such as logarithmic PCM, adaptive PCM, differential PCM, adaptive differential PCM, or delta modulation, are employed [1].

2.2.2 Processing and Feature Extraction of Speech Signals [1, 2, 4, 8]

This section describes the basic techniques that are applied to extract features and acoustic characteristics from the speech signal. Most of these techniques are based on the source-filter model of speech production that was introduced in Section 2.1. Speech analysis is mainly the process of estimating the slowly time-varying parameters which characterize the speech production system and its excitation. Other goals include voiced/voiceless classification and pitch estimation [1].

The speech processing techniques can be classified as frequency-domain or time-domain approaches. Frequency-domain methods include Fourier transformation, filter banks, homomorphic or cepstral analysis. The time-domain methods include the autocorrelation function, zero-crossing rate, and signal energy. The linear prediction technique is a time-domain method, however it can be extended to estimate the spectrum of the signal [1].

2.2.2.1 The Concept of Short-Time Analysis

Speech signals are non-stationary, and the speech parameters vary relatively slowly with time. However, it can be assumed that speech signals are stationary over relatively short time intervals (10-30 ms). Thus most of the speech processing systems operate on a time-varying basis, using short-time uniformly spaced segments or frames of speech of typical duration of 10-30 ms. However, there are some adaptive segmentation techniques that divide the speech signal into short-time non-uniformly spaced segments, but these are relatively complex techniques.

Windowing [2]

Usually an arbitrary set of N points is taken as one frame of speech. This is equivalent to multiplying the signal by a uniform rectangular window in the time-domain. This distorts the spectrum of the signal by adding spurious high frequency components.

To reduce the spectral distortion, each frame or segment is multiplied by a smooth window function. Triangular, Gaussian and cosine-shaped windows have been applied, but the effects are much the same. A commonly used window is the Hamming window that is given by

$$w(n) = 0.5(1 - \cos(2\pi n/(N-1))) \quad , \quad n = 0, 1 \ldots, N- \qquad (2.1)$$

2.2.2.2 Pre-emphasis [1, 8]

In the spectrum of voiced sounds there is an overall -6 dB/octave trend, as frequency increases. This is a combination of a -12 dB/octave trend due to the voiced excitation source, and +6 dB/octave trend due to the radiation from the mouth. It is desirable to compensate for the -6 dB/octave roll-off so that the measured spectrum has a similar dynamic range across the entire frequency band. This is called pre-emphasis. The pre-emphasis can be implemented digitally as a first-order high-pass filter with a 3 dB cut-off frequency somewhere between 100 Hz and 1 kHz. The pre-emphasis filter can be described by:

$$y(n) = x(n) - ax(n-1) \tag{2.2}$$

where y(n) denotes the current output sample of the pre-emphasis filter, and x(n) is the current input sample, and a is a constant usually chosen between 0.9 and 1.

In the case of voiceless speech, there is no need to apply pre-emphasis, however, for simplicity, pre-emphasis is normally applied to unvoiced speech as well.

2.2.2.3 Short-Time Fourier Transform (FT, DFT, FFT) [1, 2, 4, 8]

The Fourier transform (FT) is employed in estimating the spectrum of speech signals. Due to the fact that speech signals are non-stationary, and are not known for all time, short-time analysis is normally applied. Analog implementation of short-time FT can be implemented by a band-pass filter followed a rectifier and a low-pass filter [1]. Sweeping the center frequency of the band-pass filter through the frequency range of the signal

causes the spectrum of the signal to appear on the output of the low-pass filter. This method is employed in the sound spectrograph; the instrument used to obtain sonograms.

Because most of the speech signal processing is performed digitally, the discrete Fourier transform (DFT) is needed. As in the analog case, the short-time DFT is applied to estimate the spectra of speech signals. The computational time required to calculate the DFT of a frame of N-points of data is of the order of N^2. To reduce the computational time, then a Fast Fourier Transform (FFT) is usually applied. The computational time required by an FFT is on the order of $N\log_2 N$.

The short-time spectrum obtained by the FFT can be helpful in extracting several features and characteristics of speech sounds, such as pitch and formant frequencies. However, selected samples of the spectrum can be used as a set of input features to the recognition stage. The human auditory system uses the short-time spectrum, along with other features, in the process of speech perception [6]

A drawback of FFT-based spectra is that the FFT of speech signals is not smooth and in many speech processing algorithms, where smooth spectra are required, additional stages of smoothing are needed.

2.2.2.4 Filter Banks and Wavelets [1, 2, 6, 9-13]

Filter banks and wavelets can be applied to estimate the short-time spectrum of speech signals. Wavelets and the filter bank processing approach are based on the human auditory processing system models. In the early stages of processing in the auditory system, the short-time spectrum is estimated with a system that can be approximated by a wavelet or filter bank system [1, 2, 6, 9]. As was mentioned in Section 2.1, the basilar

membrane of the cochlea can be modeled by a wavelet transform, and by an affine transform for frequencies above 500 Hz, or by a set of band-pass filters.

The continuous wavelet transform (CWT) is defined as:

$$CWT(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) h^{*}(\frac{t-b}{a}) dt \,, \tag{2.3}$$

where x(t) is the time-domain signal, a is called scale parameter, b is called time location, and h(t) is the wavelet 'prototype' function which can be thought of as a band-pass function.

For spectral estimation, the scale parameter is defined as

$$a = \frac{f}{f_o} \,, \tag{2.4}$$

where f_0 is the center frequency of the band-pass function.

The discrete wavelet transform (DWT) can be obtained by discretizing the scale parameter, a, and the time location parameter b. Discretizing the time location, b, corresponds to short-time analysis in speech processing. This means that the division of the speech signal into 10-30 ms frames and calculating the spectrum for each frame, is equivalent to discretizing the time-location b.

The DWT is equivalent to passing the processed signal through a finite set of band-pass filters and sampling the energies at the outputs of these filters.

A filter bank short-time spectrum analyzer is shown in Fig. 2.8. The band-pass filter center frequencies and bandwidths are chosen to cover the typical speech frequency band (0-5 kHz). As few as 4 and as many as 100 or more channels have been used depending

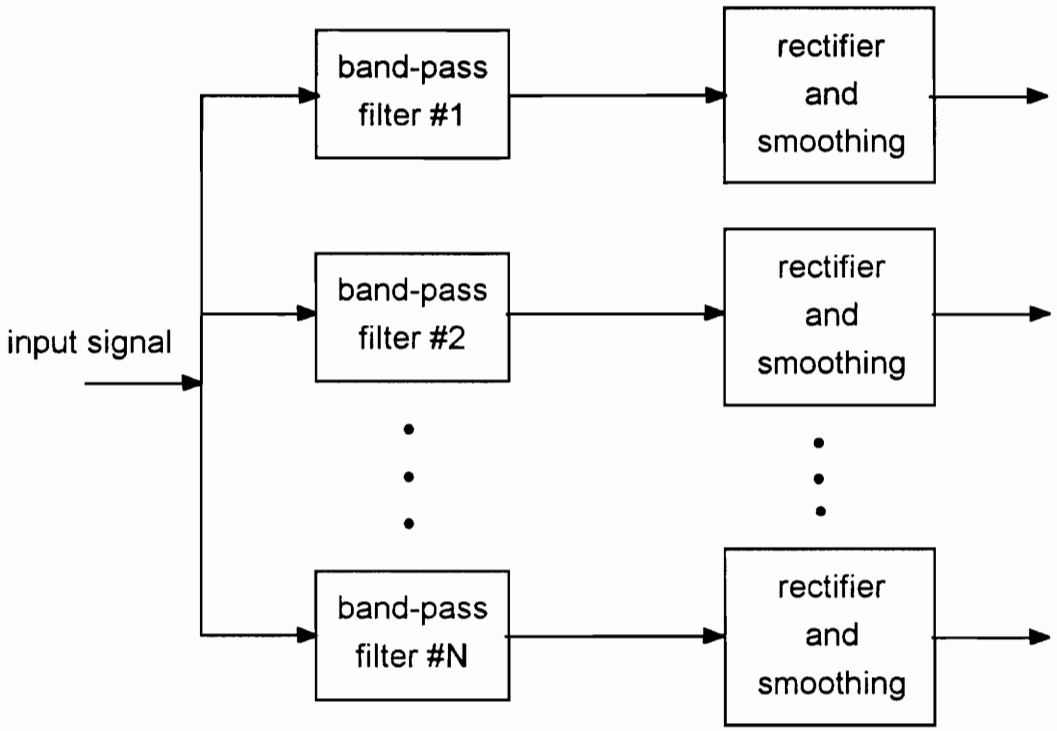


Fig. 2.8 Filter bank system for short-time spectrum estimation.

on the application and the required accuracy. Usually, the bandwidths and the frequency spacing of the filters increases with frequency in an attempt to approximate the operation of the human ear and due to the fact that the resolution of information in speech signals decreases with frequency. Table 2.2 provides the specification of a 19-channel filter bank due to Holmex [142] that has been used successfully in a channel vocodor, and as the front-end in an automatic speech recognition system.

An important issue is the choice of a bandpass filters set. The ideal filters are ones that have rectangular characteristics with the same constant gain and linear phase in their pass bands, and zero gain outside. Hence, the sum of the outputs of these filters is a perfectly reconstructed replica of the input signal. However, the realization of such ideal filters is impossible. When realizable filters are employed, the filters overlap with each other, especially when lower filter orders are employed. Detailed design procedures of filter banks and fast algorithms for implementing wavelet transforms are provided in [10-13].

Analog implementation of a filter bank for short-time spectrum estimation consists of a set of parallel channels, where each channel consists of a band-pass filter followed by a precision rectifier followed by a low-pass smoothing filter. The output of each channel is sampled every 5-30 ms (one frame of speech), and the corner frequency of the smoothing filter is about 20-60 ms.

There are several digital implementations of a filter bank system. The filters can be implemented using IIR or FIR digital filters.

In our research, the filter bank design is oriented towards getting good accuracy with the minimum possible number of channels and the minimum possible complexity of the filters. The outputs of the filter bank system are provided as inputs to the neural network classifiers, where a minimum number of inputs is required to reduce the size of the classifier. However, there is a trade-off between the accuracy of the spectral estimation

Table 2.2 Specifications of a 19-channel filter bank, after Holmex (1980).

Channel number	Center frequency	Analyzing bandwidth
1	240	120
2	360	120
3	480	120
4	600	120
5	720	120
6	840	150
7	1000	150
8	1150	150
9	1300	150
10	1450	150
11	1600	150
12	1800	200
13	2000	200
14	2200	200
15	2400	200
16	2700	200
17	3000	300
18	3300	300
19	3750	500

and the complexity of the filter bank system. One of the research tasks have been the optimization of the filter bank system for use as a front-end to a neural network based speech recognition system.

2.2.2.5 Cepstral and Homomorphic Analysis of Speech Signals [1, 2]

As mentioned in the introduction, the speech production system can be modeled by a source-filter configuration. The method of cepstral or homomorphic processing is used for separating the excitation signal of a speech waveform from the filter part. This makes it easier to extract both the pitch frequency of the excitation and the formant frequencies of the vocal tract.

The block diagram of a cepstral or homomorphic analysis system is shown in Fig. 2.9. First the short-time DFT of the speech signal is performed :

$$S(\omega) = E(\omega) \times H(\omega) \tag{2.5}$$

where $S(\omega)$ is the DFT of the speech signal, $E(\omega)$ is the excitation spectrum, and $H(\omega)$ is the frequency response of the vocal tract. Then the logarithm of $S(\omega)$ is calculated:

$$log\ S = \log E + \log H\ . \tag{2.6}$$

Then the inverse DFT of the transform is calculated :

$$C(t) = F^{-1}(\ log\ S\ ) = F^{-1}(\ log\ E\ ) + F^{-1}(\ log\ H\ ) \tag{2.7}$$

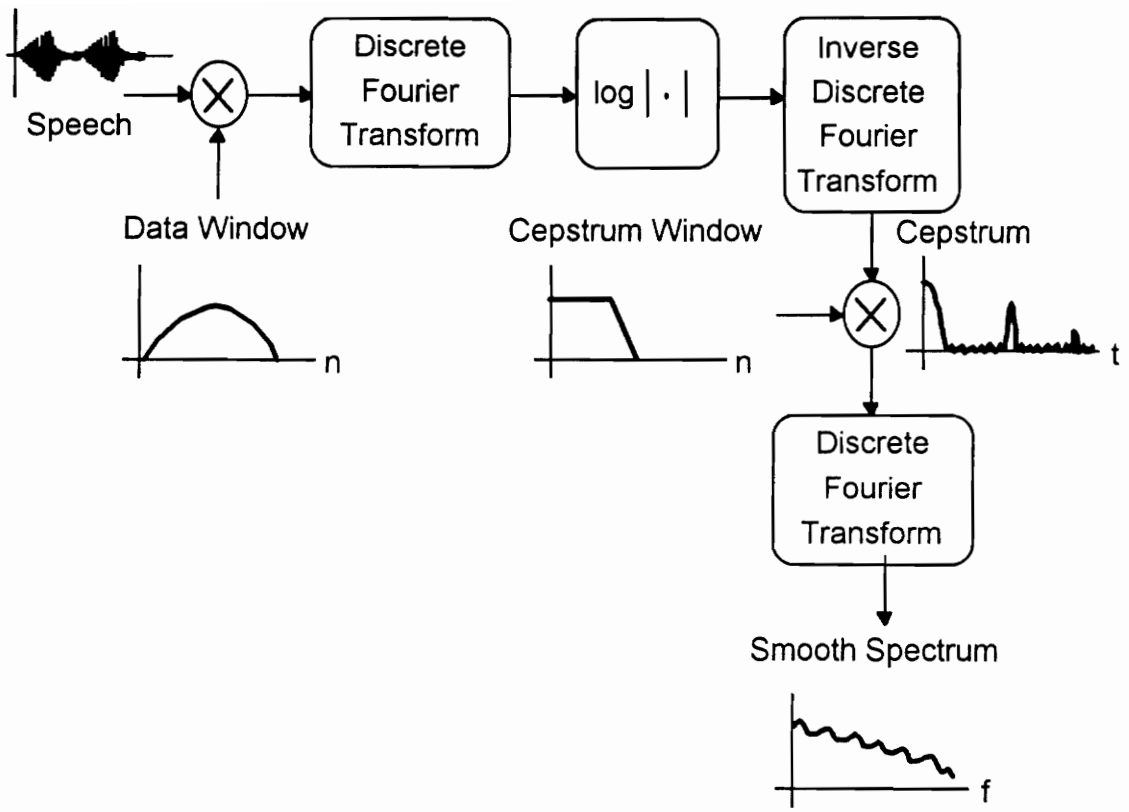


Fig. 2.9 Cepstral or homomorphic system for speech analysis.

C(t), which is the inverse DFT of the log of the spectrum of the signal, is called the cepstrum. The horizontal axis of the cepstrum is called the quefrency and has the unit of time.

In the cepstrum, the contribution due to the vocal tract occurs near the origin, and the contribution due to the excitation tends to occur far from the origin. If the speech is voiced, then sharp pulses occur at the pitch period and its multiples. If the speech is unvoiced, then there are no clear peaks in the cepstrum.

The position of the first peak in the cepstrum of a voiced speech can be used to determine the pitch frequency of the sound.

If the part that corresponds to the excitation is filtered out of the cepstrum, and then a DFT is performed (Fig. 2.9), the result is a smoothed spectrum of the vocal tract response. Similar results can be achieved by directly smoothing the DFT of the speech signal, however, the cepstral analysis provides both the smoothed spectrum of the vocal tract response and an estimate of the pitch period in the case of voiced speech.

The homomorphic (also called cepstral) analysis results can be employed to extract features, such as the pitch frequency and formant frequencies. Cepstral analysis can be used to distinguish voiced sound from unvoiced sounds. Samples of the cepstrum or cepstrally smoothed spectrum can be used as input features to the recognition stage.

2.2.2.6 The Short-Time Autocorrelation Function [1]

The autocorrelation function provides a measure of the correlation of a signal with a delayed copy of itself. The short-time autocorrelation function is defined as

$$R(k) = \sum_{n=0}^{N-1} [x(n)\cdot w(n)]\cdot [x(n+k)\cdot w(n+k)], \qquad (2.8)$$

where R(k) is the k-th sample of the autocorrelation function, x(n) is the n-th sample of the N-points frame of the speech signal, and w(n) is the n-th sample of the window function.

For voiced sounds, the autocorrelation function has peaks at time-shifts corresponding to the multiples of the pitch period. In the case of unvoiced speech, the autocorrelation function does not have clear peaks.

The autocorrelation function can be used for estimating the pitch of voiced speech. However, there are cases where it is difficult to detect the peaks in the autocorrelation function. Despite that, the autocorrelation function is used in some pitch detection algorithms.

2.2.2.7 Linear Prediction Analysis [1, 2, 8]

Linear prediction analysis assumes that the signal being analyzed was produced by passing an excitation through a suitable linear filter. As was mentioned in Section 2.1, the speech production system can be approximated by a source-filter model. Hence, linear prediction analysis is a particularly suitable method for the analysis of speech signals.

The linear prediction analysis is based on the idea that the current sample, x(n), of the speech signal can be predicted from the p previous samples, or

$$x(n) = e(n) + \sum_{k=1}^{p} a_k x(n-k), \tag{2.9}$$

where e(n) is the error between x(n) and its predicted value, and a_k, k=1..p, are constant coefficient that are called the LPC coefficients.

The main problem in linear prediction analysis is to estimate the LPC coefficients. There are algorithms for estimating the LPC coefficients based on minimizing the mean square error between the signal and its linearly predicted approximation. One commonly used method is the autocorrelation method that can be implemented using the Durbin-Levinson algorithm [1, 2, 8]. Another algorithm is the covariance method. The LPC coefficients can be transformed to another set of coefficients called the reflection coefficients. The reflection coefficients are the coefficients that describe a lattice filter implementation of an all-pole filter [8].

The filter obtained from the LPC analysis is an all-pole filter which is suitable for non-nasalised voiced sounds. Moreover, the error signal, e(n), can be viewed as the excitation signal and the LPC based filter can be viewed as the vocal tract filter. However, nasals and plosives can be modeled more accurately with a zero-pole filter than with an all-pole filter. The zero-pole filter model requires solving a set of nonlinear equations by an iterative numerical methods. This makes the implementation of the all-pole filter model much easier because only linear algebraic equations need to be solved.

The LPC coefficients can be used as codes in speech coding systems, as synthesis filter parameters in speech synthesis systems, and as features for speech recognition systems. The spectrum of the speech signal can be estimated using the LPC coefficients. Also, the excitation can be estimated from the signal by applying inverse-filtering to the signal.

The LPC coefficients (usually 8-12 coefficients) can be used as input features to the recognition stage of the ASR system.

It is important to notice that the LPC analysis of speech signals is performed as a short-time analysis. This means that the LPC coefficients are estimated for each segment or frame (10 to 30 ms) of the signal.

2.2.2.8 Short-Time Energy [1]

The short-time energy function can be obtained by splitting the speech signal into frames of N samples and computing the total squared values of the signal samples in each frame :

$$E(m) = \sum_{n=1}^{N} x(n)^2, \qquad (2.10)$$

where E(m) is the short-time energy of the m-th frame, N is the number of samples in each frame, and x(n) is the current sample in the current frame.

Usually voiced sounds have higher energies than unvoiced sounds, however there are occasions when the energy of strong fricatives is greater than that of weak vowels.

The short-time energy function can help in distinguishing voiced sounds from unvoiced ones, however, the energy feature alone will not provide accurate results. Additional features, such as zero-crossing rate and spectral features are employed along with the energy feature to get more accurate results.

2.2.2.9 Zero-Crossing Rate [1]

The zero-crossing rate is a measure of the number of times in a given time interval or frame that the amplitude of the speech signal passes a value of zero. Zero-crossing rates of voiced sounds are lower than the zero-crossing rates of voiceless speech due to the random nature of voiceless sounds.

The zero-crossing rate is an important feature for voiced/unvoiced classification and for end point detection. It is often used as a part of the front-end in ASR systems.

2.2.2.10 Endpoint Detection [1]

Endpoint detection is the problem of detecting the beginning and the end of an utterance. Endpoint detection is particularly difficult if the speech is accompanied by background noise.

Many endpoint detection algorithms are based on measurements of the short-time energy and zero-crossing rate and attempt to measure as accurately as possible the changes that these quantities undergo at the endpoints. A simple algorithm for endpoint detection can be implemented as follows: A small sample of the background noise is taken. The short time energy of the entire utterance is computed. A speech threshold is determined based on the noise energy and the peak energy. Initially the endpoint is assumed to occur where the signal energy crosses this threshold. Corrections to the initial estimates are made by comparing the zero-crossing rate around the endpoint and the zero-crossing rate of the silence. If detectable changes in the zero-crossing rate occur outside the initial threshold, the endpoints are redesignated to the points at which the change took place.

2.2.2.11 Pitch Extraction [1]

The pitch period of the excitation source is an important feature of voiced sounds. It is used in speech synthesis, analysis and coding. It may also be employed in ASR and speaker identification/verification systems.

Pitch extraction algorithms can operate either directly on the time waveform or on the spectrum of the speech. There are several algorithms for pitch estimation such as the Gold-Rabiner extractor, the SIFT extractor and autocorrelation-based extractors [1]. The

Gold-Rabiner algorithm operates directly on the speech waveform. The speech signal is passed through a bandpass filter with cut-off frequencies of 100 Hz and 800 Hz. Then peak/valley detection is performed and then the pitch period is estimated.

In the case of the SIFT (simplified inverse filter tracking) algorithm, the speech signal is lowpass filtered with a cut-off of 800 Hz and down-sampled to 2 kHz. A 4th order LPC analysis is carried out, and the LPC predictor is used to inverse filter the speech signal. The autocorrelation of the error signal (which is the inverse filtered signal) is computed, and the pitch period is estimated from the peaks in the autocorrelation function.

Post-processing of the pitch contours (i.e. the pitch as a function of time) can be done to reduce the error rates of the pitch detection. Smoothing algorithms, such as low-pass filtering and median of N filtering, are employed. Any unvoiced frame between two voiced frames is converted to a voiced frame with pitch value equal to the average of the surrounding frames.

Most good pitch detectors require relatively complex and time consuming algorithms.

2.2.2.12 Formant Tracking [1]

The formant frequencies are the vocal tract resonance frequencies. The formant frequencies, amplitudes, and bandwidths are important features of speech sounds. The formants appear as peaks in the spectrum of the speech signals. However, for many reasons, it is very difficult to track the formant frequencies, and simple peak-tracking algorithms provide poor detection accuracy. The short-time spectrum contains information both about the vocal tract formants and the excitation harmonics unless pitch-synchronous analysis (i.e. single pitch periods of the speech signal are identified and analyzed in isolation) has been done. This makes the formant tracking a difficult and complex task.

The contribution of the excitation is amplified when high resolution spectrum (high sampling rate) is provided. Smoothing algorithms, such as low-pass filtering and multi-pass filtering of the amplitude spectrum, can be applied to attenuate the excitation harmonics. However, the smoothing process can produce shifts of the formant frequencies.

A smooth spectrum can be estimated employing LPC or cepstral analysis. However, even in these spectra, the excitation is not eliminated totally. The filter bank approach can provide better detection of the formants if an adequate number of filters are used, however, good results require a large number of relatively well tuned channels.

The LPC coefficients can be used to calculate the formant frequency directly (not by peak-tracking of the spectrum) by calculating the poles of the vocal tract. However, some of the estimated poles might not correspond to formant frequencies.

Several algorithms have been developed for formant tracking, and usually there is a trade-off between the complexity and the computational expense and the accuracy of the algorithms. Formant tracking algorithms become more complex when the frequency range of interest is increased. Some simplified algorithms have been developed for tracking the first three formants.

The formants of the vocal tract are estimated for each frame (10-30 ms). The formant frequencies and amplitudes can be used as input features of the recognition stage.

2.2.2.13 Voiced/Unvoiced/Silence Classification [14-15]

The classification of speech signals into voiced, unvoiced, and silence (v/uv/s) provides preliminary segmentation of speech that is important for speech analysis and recognition. As was mentioned in Section 2.1, voiced speech is produced in a different way from

unvoiced speech. The voiced sounds have quasi-periodic excitations that are produced at the glottis, while the unvoiced sounds have random noise excitations that are produced at different points in the vocal tract.

The v/uv/s classification can be made using a single parameter, such as the short-time energy and the zero-crossing rate. However, the accuracy of such a method is limited due to the overlap of the parameter values of voiced and unvoiced speech. Hence, a set of features is needed to make high accuracy v/uv/s classification. In the simplest algorithms, a combination of the short-time energy and the zero-crossing rate are used.

Classification accuracy of more than 90% can be achieved if several parameter are used. One possible set of features [14-15] consists of the short-time energy, the zero-crossing rate, and 13 cepstral coefficients. In this case, a training algorithm is needed to set the decision boundaries between voiced and unvoiced speech. A statistical decision approach was implemented by Atal and Rabiner [15], and a neural network approach was introduced in [14].

In this research, the v/uv/s decision can be used in two ways. The first is to use it as a feature along with other features to a neural network classifier in the recognition stage. The second way is to use the v/uv/s classifier in a hierarchical (such as tree or two-level) classifier structures in a recognition system.

2.2.2.14 Vector Quantization [1]

Normally the speech signal is analyzed in a sequence of frames, and each frame is represented by a set of k numerical values (such as the 12 LPC coefficients and 16 filter bank outputs). Hence each frame can be viewed as a k-dimensional vector in a k-dimensional space.

The basic idea of vector quantisation is that the vector space is divided into a finite number, N, of non-uniform regions or bins, with each region being represented by a single vector giving the centroid of the bin. The collection of the N vector centroids is called the codebook. In the actual process of vector quantization the input vector is assigned to the nearest bin.

For spectral vectors, such as filter bank outputs, usually the Euclidean distance is employed in the vector quantization process. For the LPC vectors, usually the Itakura distance [143] is used.

When vector quantisation is performed, an adequate number of training vectors must be provided. A popular algorithm for codebook design is the LBG algorithm (Linde, Buzo and Gray algorithm) [144].

2.3 Processing and Recognition of Speech Patterns

A typical speech recognition system block diagram, that is based on the pattern matching approach, is shown in Fig. 2.10. The speech recognition task consists of two main steps. First, the speech signal is processed and certain features and characteristics are extracted. Second speech units are identified using the features and analysis results obtained in the processing and feature extraction step. In this section the common processing and recognition techniques of speech patterns are summarized. The neural network approach for pattern classification and recognition is emphasized because it is the approach that is studied in this research.

The output of the processing and feature extraction stage is a set of features, that can be spectral or LPC coefficients or acoustic or other parameters such as formant frequencies or amplitudes or pitch period. If there are n such features, each set of values may be represented by a point, or vector, in an n-dimensional space. Each vector can be regarded as a pattern.

Any speech utterance or unit can be viewed as a sequence of feature vectors that corresponds to a sequence of frames of the segmented speech utterance. This utterance or unit of speech is identified or classified in the recognition phase.

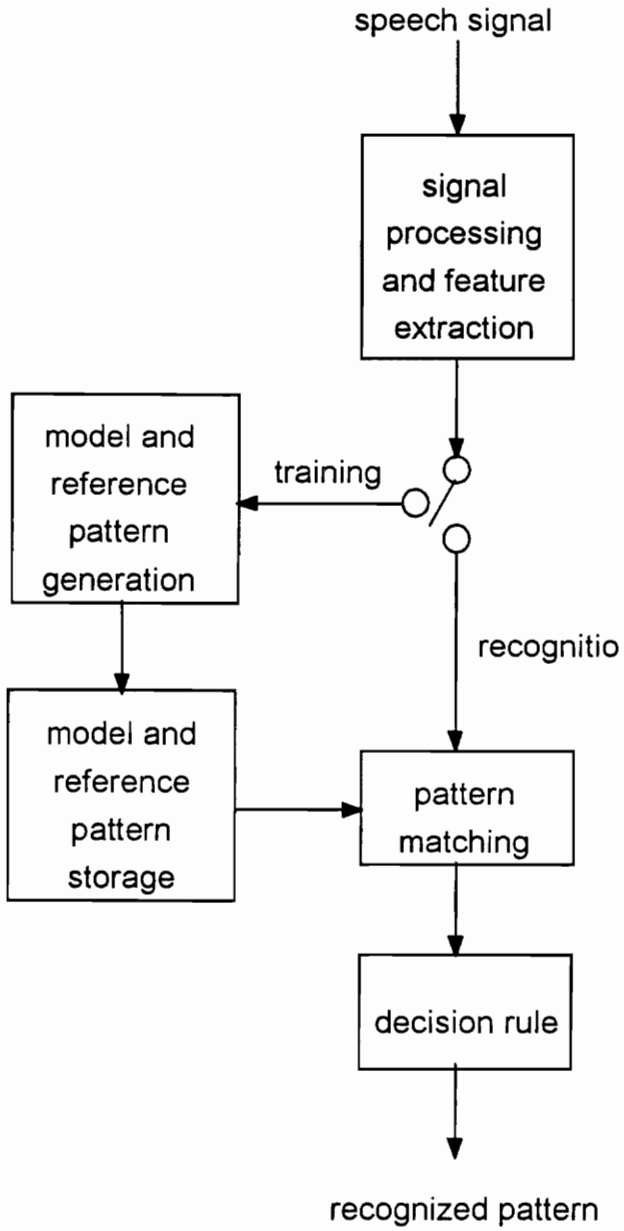


Fig. 2.10 Block diagram of a typical speech recognition system based on pattern matching.

2.3.1 Approaches to Pattern Classification [2, 16]

A pattern classifier is a system for identifying the class to which a pattern belongs. There are four main approaches for pattern classification. The existing classifiers include geometric or hyperplane, topological or exemplar, probabilistic and kernel or receptive field classifiers.

2.3.1.1 Probabilistic Classifiers

Probabilistic classifiers assume an a priori probability distribution of the input features. Usually Gaussian or Gaussian mixer distributions are employed. A probability density function is defined for each class. The decision that an unknown pattern belongs to a certain class is made by choosing the class for which the value of its probability density function is the greatest for the unknown pattern at the input of the classifier. The parameters of the distribution are usually estimated using supervised training. All the training data is assumed to be available simultaneously. The probabilistic classifiers provide optimal performance when the underlying distributions are accurate models of the test data and when there is sufficient training data. In many practical cases and with non-stationary systems these two conditions are often not satisfied.

2.3.1.2 Geometric or Hyperplane Classifiers

Hyperplane classifiers form complex decision regions using nodes that form hyperplane decision boundaries in the space spanned by the inputs. A well known example

of a hyperplane classifier is the multi-layer perceptron (MLP). Typically a weighted sum of the inputs is passed through a nonlinearity such as a sigmoid nonlinearity. The hyperplane classifiers do not require a priori models of the data. The memory and computation requirements for classification are low, however, the training time is long and the training algorithms are complex.

2.3.1.3 Exemplar, Topological or Nearest Neighbors Classifiers

Exemplar classifiers perform classification based on the identity of the training examples, or exemplars, that are nearest to the input. Exemplar nodes calculate the weighted Euclidean distance between inputs and centroids. Centroids are cluster centers formed during training, or previously presented labeled training examples. Examples of exemplar classifiers include the K-nearest neighbors [2], the learning vector quantizer [145], adaptive resonant theory [146], etc.

Exemplar classifiers can be trained much faster and easier than the hyperplane classifiers, however, the memory and computation requirements for classification are much higher than what is required with the hyperplane classifiers.

2.3.1.4 Kernel or Receptive Field Classifiers

Kernel classifiers create complex decision regions from kernel function nodes that form overlapping receptive fields. Kernel function nodes use a kernel function that provides the strongest output when the input is near a node's centroid. Typically Euclidean distance is employed, and the output of the kernel function decreases monotonically when the distance between the input and the centroid increases. Examples of kernel classifiers

are the potential function or radial basis function classifiers [141], and conventional classifiers that estimate probability density using the Parzen window approach or mixture distributions [147].

The kernel classifiers can be trained using combined unsupervised/supervised learning. The training time is shorter than in hyperplane classifiers. The kernel classifiers have intermediate memory and computation requirements.

2.3.2 Dynamic Programming and Dynamic Time-Warping [1-2, 7, 17-25]

Speech patterns or utterances can be represented by a sequence of feature vectors in an n-dimensional space. If the speech patterns have constant length or fixed number of feature vectors, then any input pattern can be recognized by identifying the minimum distance out of the measured distances between the input pattern and each of the templates in the stored database. The distance between two patterns can be defined as the sum of the distances between each two vectors in the input and template patterns.

However, when the input and template patterns consist of sequences with different numbers of vectors, the classification problem becomes complex. Linear time normalization can be applied in the simple case when uniform mapping of the time axis between the patterns exists (Fig.2.11(a)). Unfortunately, the mapping among speech patterns along the time axis is nonlinear (Fig.2.11(b)). This means that each internal sub-pattern or sub-unit in one pattern has a different time alignment when mapped to another pattern. Dynamic programming (DP) is a nonlinear normalization technique, and when the

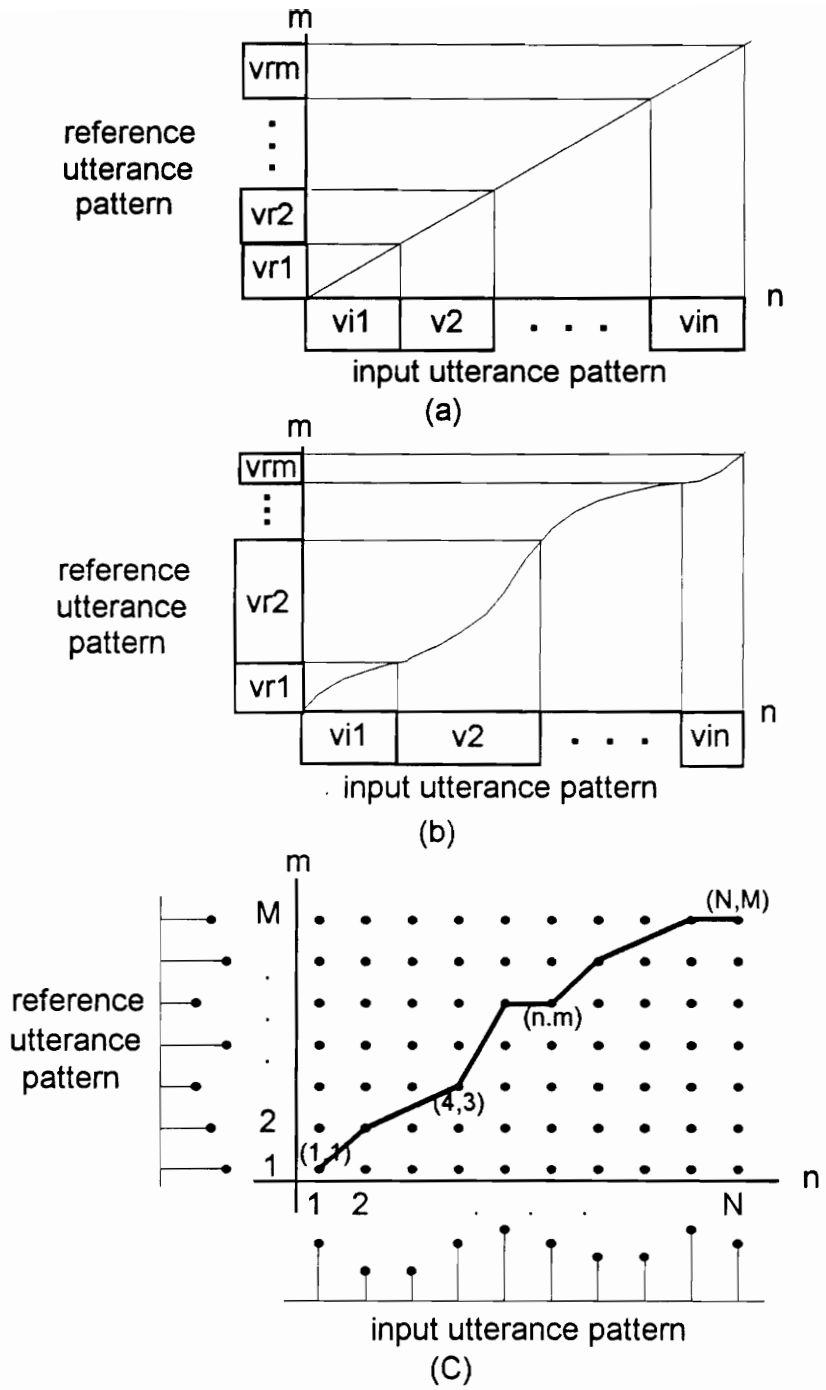


Fig. 2.11 Time alignment of patterns, (a) linear time alignment; (b) non-linear time alignment; (c) time alignment of two one-dimensional patterns.

normalization is performed along the time axis, dynamic programming is called dynamic time warping (DTW).

Dynamic programming is a technique used in multi-stage decision processes, where decisions are made at discrete time intervals during the course of some transaction in which the outcome of the transaction is a function of all the decisions. The dynamic programming technique is based on the **principle of optimality** that an optimal solution or policy has the property that, whatever the initial state or the initial decision may be, the remaining decision must constitute an optimal policy or solution with respect to the state resulting from the initial decision. To illustrate the principle of optimality in pattern matching framework consider the time alignment of two one-dimensional pattern as shown in Fig. 2.11 (c). The principle of optimality can be stated as follows : To find the best path from the grid point (1,1) to the grid point (n,m) that passes through the grid point (4,3), for example, then the best path from (4,3) to (n,m) must be selected as the best path from (4,3) to (n,m). In other words, and for pattern matching, the optimal global path can be obtained by always locally choosing the path which minimizes the distance between the two compared patterns. Hence the DP or DTW provides a method for measuring the minimum distance between two patterns that have different sizes.

2.3.2.1 Application of DTW to Isolated-Word Recognition

In isolated-word recognition, the speech patterns, words, are sequences of feature vectors where the endpoints of each pattern are known. Typically the number of words in the isolated-word system vocabulary is limited (<100). This can substantially increase the recognition rates. As far as dynamic programming is concerned, the fact that the endpoints of the patterns are known makes the implementation algorithm less complex.

In addition to the endpoint restriction, a number of local constraints are placed on the time-alignment path. These restrictions are added to prevent excessive compression or expansion of the time-scale, to insure convergence of the DP algorithm, and to reduce the complexity, memory, and computational requirements of the DP algorithms. For example, the slope of the path can never be negative. These local constraints are incorporated by specifying the full path in terms of simple local paths, that can be combined to form the full path. It is possible to enforce many different constraints on the local paths. One commonly used constraint is the Itakura [18] constraint shown in Fig. 2.12, that produces a DTW algorithm that performs very well and is used in many speech recognition systems.

The constraints on the alignment paths help in increasing the efficiency of the DTW algorithms and in reducing the storage requirements. These are very important because DTW algorithms are computationally expensive to implement and the storage requirements are large especially for large vocabularies. Most speech recognition systems based on DTW use some form of adjustment window to restrict the region in which the alignment path could pass. There are several methods for choosing the adjustment window width, but it is generally made inversely proportional to the difference in duration of the words being matched.

2.3.2.2 Application of DTW for Connected Speech Recognition

In connected speech, the main problem is that there are no distinct pauses between words. Hence, the endpoints of the words are very difficult to detect. One approach to solve this problem is to segment the connected words into a sequence of isolated words, and to then apply DTW algorithms used for isolated word recognition. However, the reliability of the segmentation process is not high. Another approach one might think of is

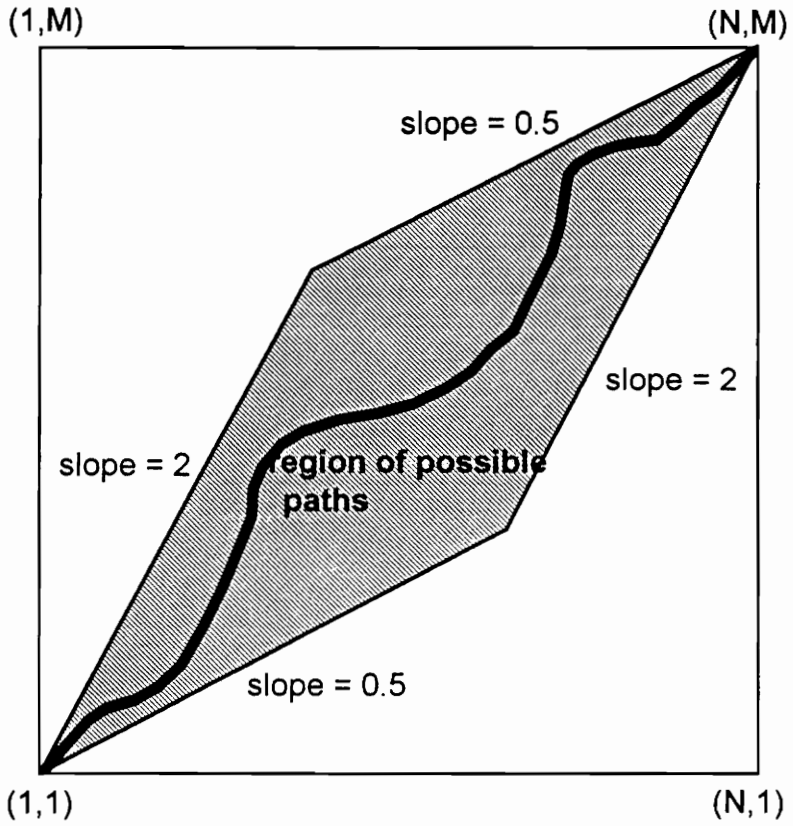


Fig. 2.12 Path region for Itakura local path constraint.

to concatenate isolated word patterns to form all possible combinations of connected word patterns, and to then apply DTW to find the sequence of words which best matches the connected input pattern. However, even for moderate vocabulary size and for small connected strings, this approach is not practical due to the amount of computation required. For example, a system that recognizes strings of at most four connected digits out of a ten digits vocabulary, has to match more than 10^4 one-digit patterns. Some algorithms that reduce the amount of computation to reasonable proportions have been developed. The **two-level DP algorithm [19]** is one of the first DTW for connected speech recognition. This algorithm consist of a word-level matching stage followed by a phrase-level matching stage. In the word-level stage, each stored word pattern is matched against all possible regions in the connected word input pattern. This gives rise to a matrix of partial distances. In the phrase-level stage, DTW is performed on the partial distances to obtain the sequences of words that minimize the total distance.

The **level-building algorithm** [20-21] is a more efficient algorithm than the two-level algorithm. In this algorithm, DTW is applied at a number of stages or levels up to the maximum number of anticipated words in the connected word string. The level-building algorithm has similar performance to the two level algorithm, but requires much less storage and computations.

The **Bridle or one-stage algorithm** [22] is a more efficient algorithm than the level building algorithm. This algorithm generates what is called a word-decision tree, which grows as the input is processed. The Bridle algorithm is based on what is called a **beam search**. In beam search, the unlikely interpretations of the input are avoided, but keeps the option open if there is some ambiguity. The beam search leads to a considerable reduction in computation. The Bridle algorithm has similar recognition performance to the level building and the two-level algorithms, however , the Bridle algorithm requires only about

25% of the computation required by the level-building algorithm and only about 4% of that of the two-level algorithm. The Bridle algorithm requires only about 10% of the storage required by the other two algorithms. All the algorithms have been implemented in real-time in a number of speech recognition systems.

The above mentioned algorithms are deterministic algorithms. The **Viterbi algorithm** [23], which is a DTW algorithm, has been proposed for stochastic pattern matching in hidden Markov model based ASR systems as will be presented in Section 2.3.3.

2.3.3 Hidden Markov Models (HMM) [1-2, 7, 26-35]

Markov modeling provides a mathematically rigorous approach to developing robust statistical signal models. A powerful technique for modeling the temporal structures and variability is the **hidden Markov modeling**. This is a probabilistic pattern matching approach that models a time-sequence of speech patterns as the output of a stochastic process. An example of a hidden Markov model (HMM) is shown in Fig. 2.13. The HMM consists of an underlying Markov chain where each circle represents a state of the model, and at a discrete time instant in time t, the model is in one of these states and outputs a certain speech pattern or observation. At time t+1, the model moves to a new state or stays in the same state, and outputs a new observation. This process is repeated until the complete sequence of patterns has been produced. The transitions among states is determined by a matrix of probabilities A, where A(i,j) is the probability of moving from state i at time t to state j at time t+1. In any state, the production of a speech pattern by

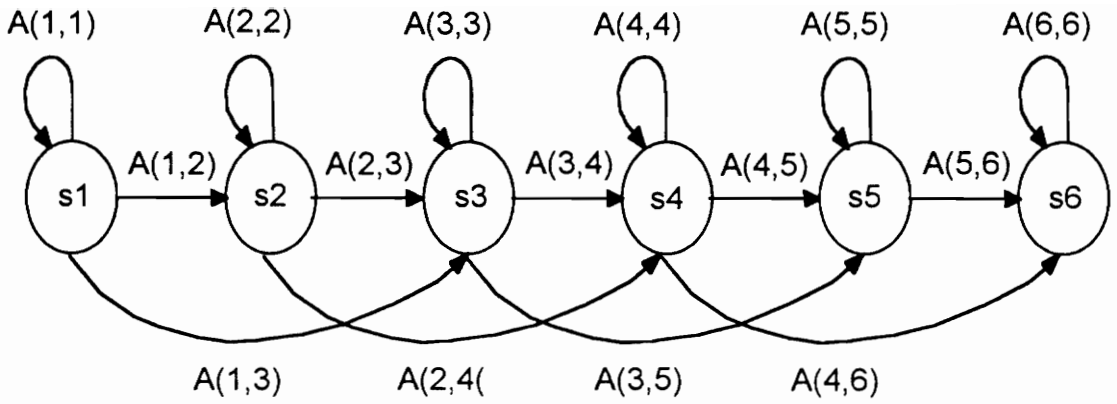


Fig. 2.13 An example of a hidden Markov model (HMM).

the model is governed by a matrix of probabilities B, where B(j,k) denotes the probability of producing pattern k when the model is in state j. The starting state of each model is represented by a probability vector p, where p(j) is the probability of the model being at state j at time t=0.

The model is said to be hidden because the state sequence that produces a certain sequence of patterns cannot be determined. In a general HMM, a transition from one state to any other state is possible, however typically a left-to-right model, which does not include backward transitions, is employed for speech recognition. This is necessary to model the temporal structures of speech effectively. Other left-to-right HMM models, with a different number of states and different topologies, can also be used, though they are not significantly better or worse than the one shown in Fig. 2.13.

HMM can be related to the speech production process by looking into the speech production process. Speech can be viewed as a sequence of different sounds, produced by the speech articulators taking up a sequence of different positions. If the articulatory positions corresponding to static sounds are considered as states, then speech can be viewed as the result of a sequence of articulatory states of different and varying duration. Hence, the transitions between the state can be represented by probabilities, and the overall Markov chain represents the temporal structures of the word. The acoustic patterns produced in each state correspond to the sound being articulated at that time. The production of these patterns can be described by probabilistic functions due to the variations in the shape of the vocal apparatus, pronunciations etc.

The recognition accuracy of an ASR system based on HMM is slightly better than that of an equivalent system based on dynamic programming though its storage and computation requirements are roughly an order of magnitude less. In a HMM system it is much easier to model speaker variability, however, this involves a lot of training

computations. The HMM can be employed in sub-word units such as phonemes, and as a result seems to have a potential for implementing large-vocabulary, speaker-independent and connected speech systems.

Examples of HMM-based ASR systems include the Tangora system [148], the SPHINX [149] system, and the AT&T connected digit recognizer [150]. The Tangora system was developed at IBM, and it is a speaker-dependent isolated-word ASR system scaleable from 5000 to 20,000 words. It has recognition rates above 94%. The SPHINX was developed at CMU, and it is a speaker-independent continuous speech ASR systems. For a 1000 word vocabulary, the recognition rates range from 67% to 97% depending on the recognition units employed. The connected digit recognizer was developed in AT&T, and it is a speaker-independent ASR system, and its recognition rate ranges from 92% up to 99.6%.

2.3.3.1 Isolated-Word and Connected Speech Recognition Using HMM

Before the HMM approach to speech recognition was introduced, it has been convenient to distinguish recognition systems based on the ability to recognize isolated or connected speech. Computational issue not withstanding, it is advantageous in an HMM framework to consider the isolated-word recognition problem using a connected speech framework [26]. The basic advantage of this approach is that a heuristic utterance detection/segmentation algorithm is no longer needed: the recognizer determines the optimal start and stop of an utterance. Connected speech recognition with HMM system is essentially the same as in the case of isolated-word recognition except for that in connected speech the state of non-speech patterns is treated as any other state of any speech pattern. As a result, any word can follow any other word with arbitrary duration of

non-speech occurring between any two words, and the pattern matching and training of connected speech ASR system are similar to the pattern matching of isolated-word ASR systems.

Each word in the vocabulary is represented by an HMM. Each input word consists of a sequence of frames of features. In the recognition phase, the probability or the likelihood of producing the unknown input pattern with each of the HMM word models is computed. The input word is recognized as that model which provides the maximum likelihood. Mathematically:

$$\text{The recognized word} = \underset{i=1,\ldots,W}{\operatorname{argmax}} \Pr_i\{O_t / M_i\}, \tag{2.11}$$

where $\Pr_i$ is the likelihood obtained from the ith HMM, W is the number of words in the vocabulary, M_i is the ith HMM, and O_t is the input pattern or the observation sequence.

The maximum of the likelihood function can be calculated in several ways [1]. One way is to consider all the possible state sequences that could have produced the observation sequence and then determine that sequence which maximizes the likelihood function. However, this method is unrealistic due to the large number of sequences involved. In general there will be N^T sequences, where N is number of states in the HMM of the word and T is number of frames in the observation. The likelihood function can be calculated using a recursive procedure that can reduce the amount of calculations to manageable proportions. Two algorithms have been developed: the **Baum-Welch algorithm** [1] and the **Viterbi algorithm** [23]. The Baum-Welch algorithm is based on calculating the so-called forward probabilities, that are the joint probabilities of producing a partial observations sequence and being in the a state at time t.

The Viterbi algorithm is an efficient algorithm for finding an optimal solution. It is based on the principle of optimality, and it has been employed extensively in dynamic programming based speech recognition systems. This imposes the restriction that cost or probability of any path leading to a given state can be computed recursively as the sum of the cost at the previous state, plus some incremental cost in making a transition from the previous state to the current state. This constraint integrates nicely well with the temporal constraints imposed by a HMM. For fast recognition, the so-called **Viterbi Beam search** [31] is employed, which is a fast search algorithm that produces a sub-optimal solution. This algorithm is useful particularly in large-vocabulary and connected speech recognition systems.

2.3.3.2 Training A HMM Based Speech Recognition System [1, 26]

The training process of an HMM based ASR system is performed in three steps. In the first step a seed model is generated where an initial set of prototype models are generated. In the second step, reestimation, the maximum likelihood method is used to reestimate model parameters. In the third step, the recognition performance is improved by enhancing the discrimination power of the reference models. In general, the third step attempts to identify utterances incorrectly recognized, called the confusion class, and builds statistical models that optimally discriminate between the correct class and the confusion class.

The seed model generation is often an iterative process. Models are trained and then successively refined and extended. The number of states in a model is often chosen to be proportional to the number of distinct acoustic events in the recognition unit (number of phones in a word for instance) [26]. The transition model probabilities are typically initialized to reflect an equiprobable distribution.

Obtaining a model for each vocabulary word is performed by iteratively adjusting the probability functions of the HMM so as to maximize the likelihood that the training sequence of patterns could be produced by that model. Large amounts of training data are required to get good word models. For speaker-dependent systems, several repetitions of each word are required, and for speaker-independent systems, several repetitions of each word by several speakers are required. Hence, the amount of computation required for training a HMM-based system is substantially higher than the amount of computation required during the recognition.

2.3.4 Neural Network Classifiers [1-2, 4-5, 7, 16, 36-126]

In recent years the neural networks approach has been applied to the speech recognition problem. Neural networks can be viewed as pattern matching devices with processing architectures that are based on the neural structure of the human brain. A neural network consists of simple processing interconnected processing units or neurons. The strength of the interconnections between units are variables and are called weights. Many architectures or configurations are possible, however, a popular structure is the multi-layer perceptron (MLP) that is shown in Fig. 2.14. In the MLP structure, the processing units are arranged in layers consisting of an input layer, a number of hidden layers, and an output layer. Weighted interconnections connect each unit in a given layer to every unit in the following layer. There are no interconnections between units within a layer, and there are no interconnections from outer layers back towards the input, hence

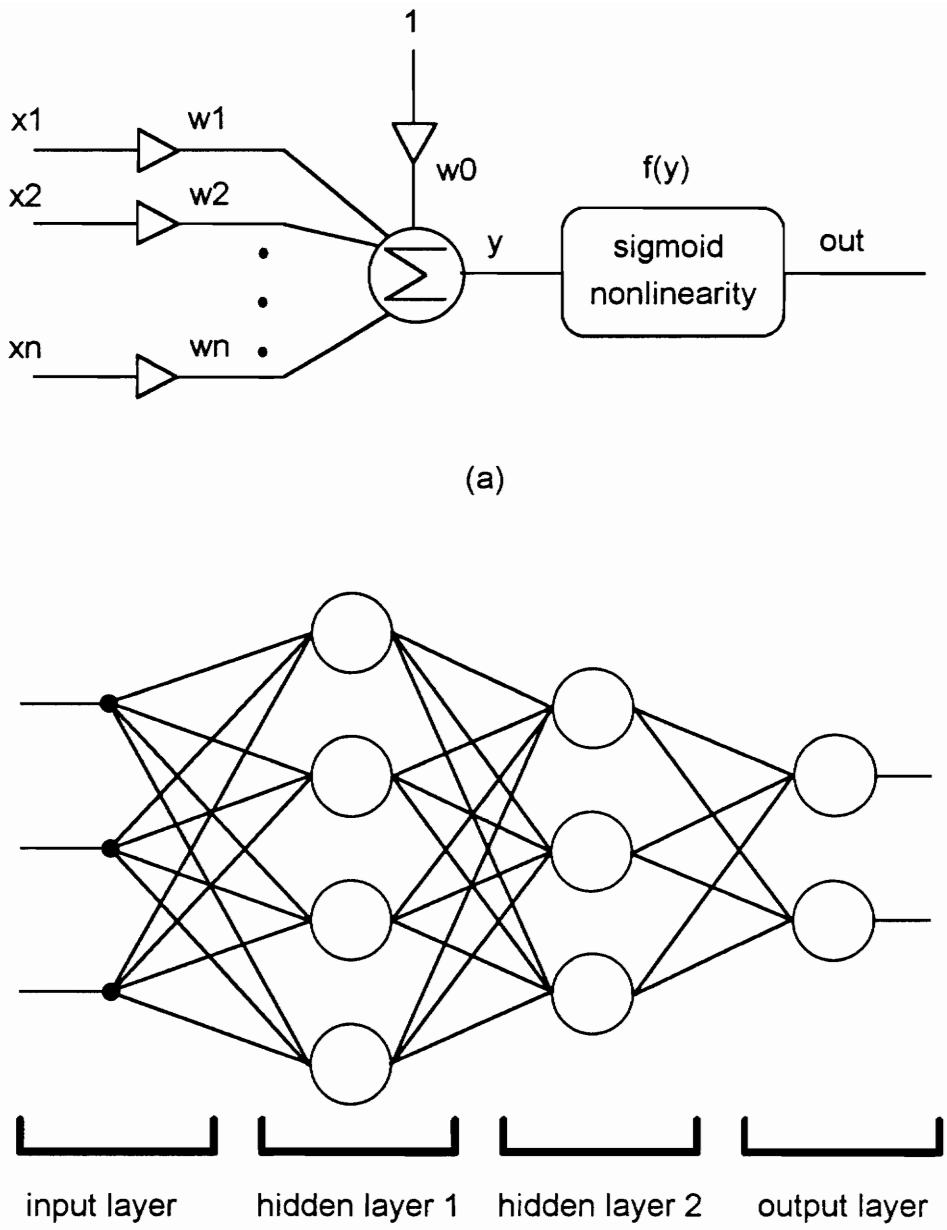


Fig. 2.14 Multi-layer perceptron, (a) a single perceptron model, (b) an example of a multi-layer perceptron.

the MLP is a feedforward network. The MLP is trained using the back-propagation algorithm which is a supervised learning algorithm.

Neural networks with various topologies and architectures have been applied to sub-word, isolated-word and continuous speech recognition, and analysis of speech signals [36-126]. Some researchers have reported that the performance of neural net-based ASR systems is not better than the performance of HMM-based or DTW-based systems. However, there are some promising results that have shown that neural net-based phoneme recognition systems can perform better than or equal to the HMM-based systems.

In this section, a brief review of neural network classifiers is provided. The recently proposed neural network classifiers, such as MLP, time-delay neural networks (TDNN), features map, learning vector quantizers (LVQ), recurrent networks, etc., for speech recognition are summarized. The recently proposed architectures of neural network based ASR systems, such as single-level networks, neural tree networks (NTNs), integrated neural network-HMM systems, and integrated DTW-neural network systems, are surveyed. The commonly used learning and training algorithms are also considered in this section.

2.3.4.1 Neural Network Pattern Classifiers [16, 36-40]

Several neural network classifiers have been proposed. In general, neural network classifiers can be divided into three categories depending on the training approach as shown in Fig. 2.15. The first category includes all the classifiers that are trained with supervised learning algorithms, such as the multi-layer perceptron and the decision tree classifiers. The second category includes all the classifiers that are trained with combined

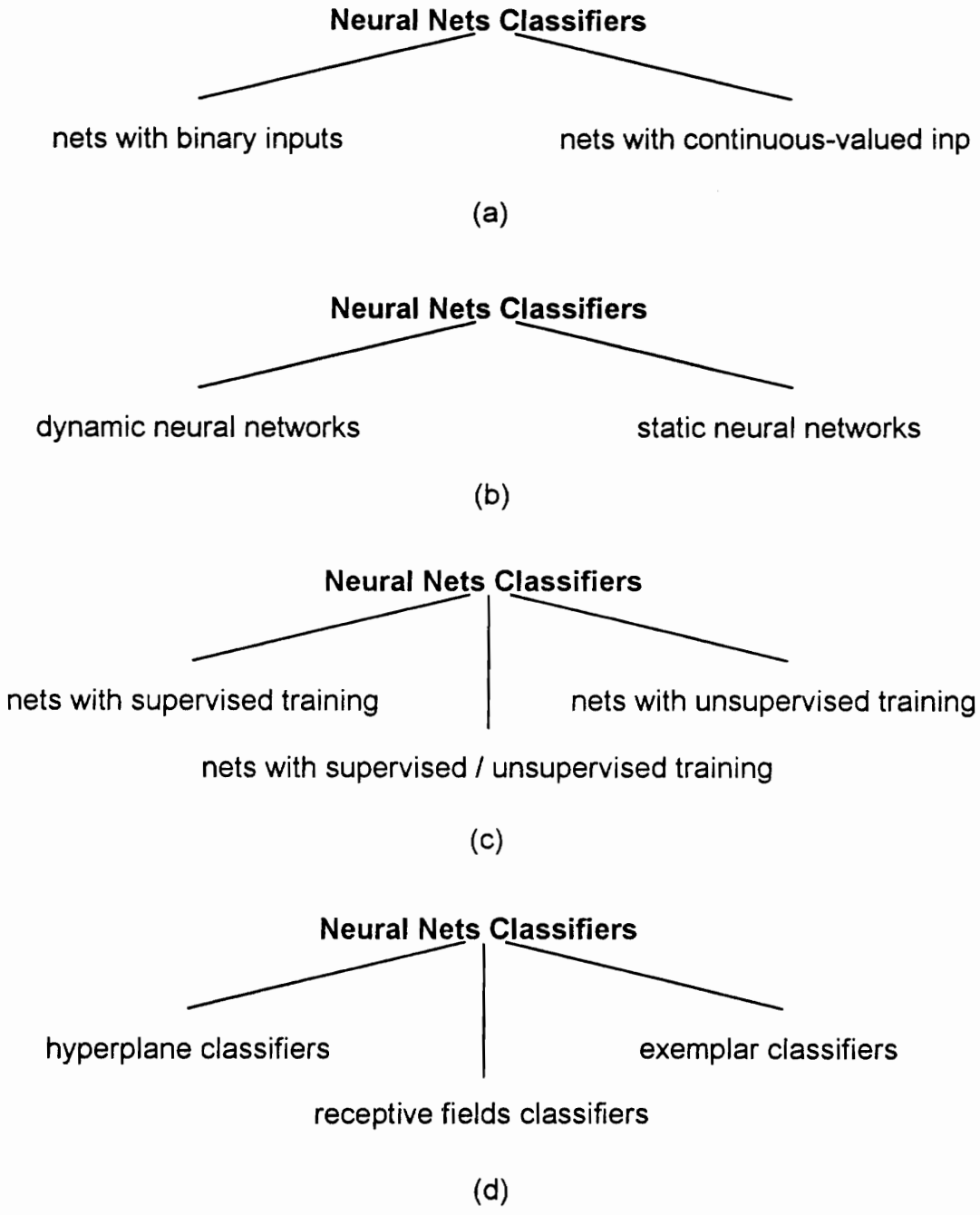


Fig. 2.15 Taxonomy of neural networks classifiers, (a) based on the input type, (b) based on the system dynamics, (c) based on the training approach, (d) based on the way they form decision regions.

supervised/unsupervised learning, such as the learning vector quantizer and features map classifiers. The third category includes all the classifiers that are trained with unsupervised learning algorithms, such as the k-nearest neighbor classifiers.

An additional way to categorize neural network classifiers is to consider the way in which classifiers form decision regions as shown in Fig. 2.15. The hyperplane classifiers, such as the MLP, form complex decision regions using nodes that form hyperplane boundaries in the space spanned by the inputs. The kernel classifiers, such as the radial basis function classifiers, generate complex decision regions from kernel function nodes that form overlapping receptive fields. The exemplar classifiers, such as k-nearest neighbor, feature map and LVQ, perform classification based on the identity of the training examples, or exemplars, that are nearest to the input.

Neural network classifiers can be divided into static classifiers and dynamic classifiers (Fig. 2.15). Static classifiers are feedforward or memory-less classifiers. Dynamic classifiers are networks with output or state feedback paths from the outputs or the internal nodes back towards the inputs.

Neural networks can be classified according to the type of the input values (Fig. 2.15). There are neural networks with continuous-valued inputs and neural networks with binary or discrete inputs.

Back-Propagation Classifiers

Back-propagation classifiers form complex decision regions using single or multi-layer perceptrons. In many cases, perceptrons with sigmoid nonlinearities are used. They are trained with supervision, using gradient-descent training methods that are called back-

propagation. Back-propagation classifiers have been applied successfully in many areas including speech recognition.

The training time of back-propagation classifiers is long especially when the number of layers and nodes is increased. An additional property of the back-propagation is that it is difficult to interpret and understand the solutions obtained after training. Also the number of layers and nodes need to be determined and optimized for each practical classification problem.

The back-propagation classifiers require less computation and memory for classification compared to other classifiers (see Fig. 2.16).

Decision Tree Classifiers

Decision tree classifiers are hyperplane classifiers. They require fewer computation and less memory than most of the other neural network classifiers (see Fig. 2.16). Their size can be easily adjusted to match their complexity to the size of the training data provided. They require complex training procedures that are not biologically motivated and that require simultaneous access to all the training data. The decision tree classifier's training time is much shorter than the training time of back-propagation classifiers. The decision tree classifiers have been applied successfully in many pattern classification applications.

GMDH and High-Order Nets

High-order and group method of data handling (GMDH) operate on high order products and powers of inputs in addition to the linear terms. If the proper number of high order terms is known then the number of hidden layers can be reduced and rapid training

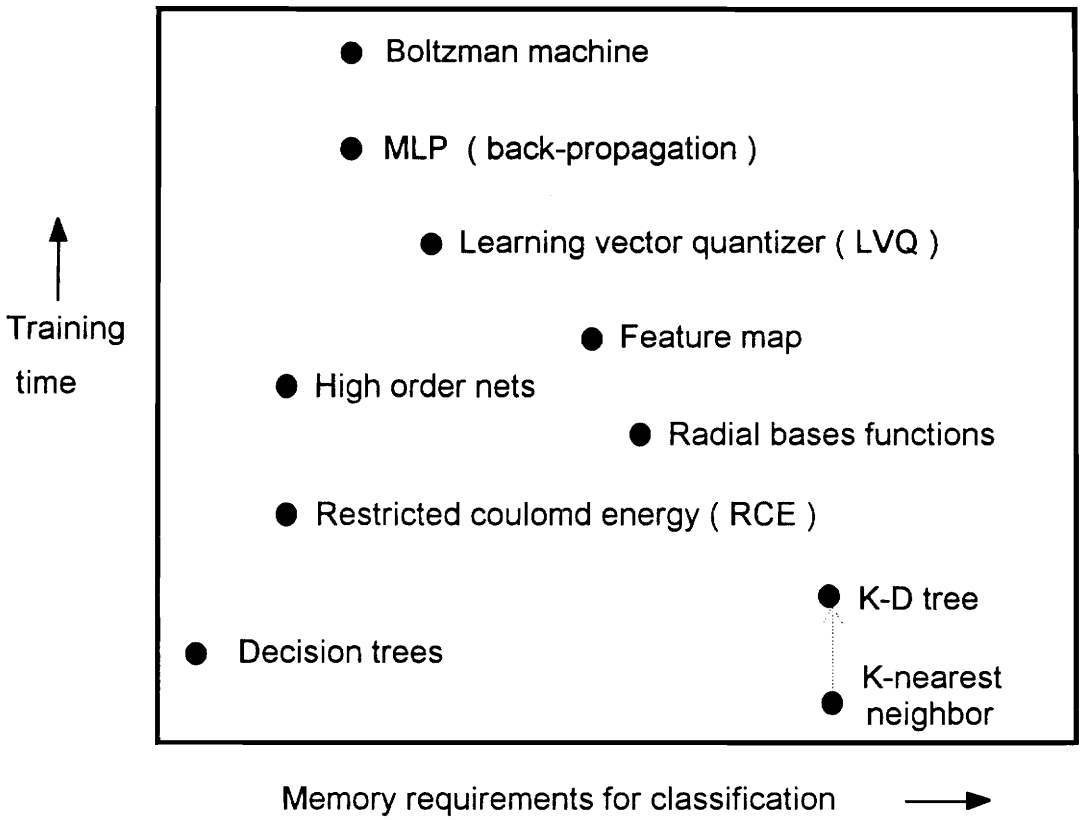


Fig. 2.16 Training time and memory requirements comparison among neural nets classifiers [16].

can be achieved. However, too many high order terms result in poor generalization and an excessive number of parameters.

GMDH networks, also called adaptive learning networks, provide a solution to the problem of matching the complexity of a multi-layer high order network to the amount of training data provided. GMDH nets provide many of the capabilities of back-propagation classifiers, but they use complex and efficient training procedures that require simultaneous access to the training data.

K-Nearest Neighbor Classifiers

K-nearest neighbor classifiers are exemplar classifiers trained without supervision. They can be trained fast, however, they require a large amount of memory and many computations. K-nearest neighbor classifier's performance is equal to or better than the back-propagation classifiers, though their training time is much less than the training time of the back-propagation classifiers (see Fig. 2.16).

The performance and the speed of the k-nearest neighbor classifiers depend on the distance metrics used. Recent work has focused on modified versions of the original algorithm to get good generalization and more effective algorithms, where specialized distance metrics are applied for classification instead of the common Euclidean distance.

The k-nearest neighbor classifiers are practical when large amounts of memory and sufficient computational power are available, and rapid learning is required.

Feature Map Classifier

The feature map classifier is an exemplar classifier that uses combined supervised/unsupervised training and requires less memory than the k-nearest neighbor classifiers. The feature-map classifier consists of an input layer, intermediate exemplar nodes layer, and an output nodes layer. The intermediate layer provides outputs that are equal to the Euclidean distance between the input and node centroids. The weights connecting the input to the intermediate layer are obtained by unsupervised training. The weights connecting the intermediate layer and the output layer are obtained by supervised training. The feature map classifier provides similar performance as the back-propagation classifiers (see Fig. 2.16), however, it reduces the number of supervised training trials (for example on vowel classification the number of training trials was reduced from 50,000 with back-propagation to 50 with feature-map classifier).

Learning Vector Quantizer (LVQ)

The LVQ is similar in structure to the feature map classifier, however, the LVQ classifier requires a final stage of supervised training that comes after the training used with the feature map classifier. Final training adjusts weights to the intermediate exemplar layer to shift node centroids slightly in a direction that attempts to improve performance, while maintaining the same number of centroids. The LVQ classifier has a similar performance to the back-propagation classifiers but often trains faster and requires more memory and computations during classification (see Fig. 2.16). The LVQ typically provides reduced error rates compared to the feature map classifiers, especially when the number of exemplar nodes is small.

Hypersphere Classifiers

The hypersphere classifiers are exemplar classifiers that create decision regions from nodes that form variable-size hyperspheres in the output space. These nodes have high outputs only if the input is within a given radius of the node's centroid. The classification decision is the label attached to the majority of nodes with high outputs. A "no decision"' response occurs if no nodes have high outputs. A recent version of the hypersphere classifiers is called Restricted Coulomb Energy (RCE) classifier. The RCE classifier can be trained fast as the nearest-neighbor classifiers, however, it typically requires many fewer exemplar nodes than nearest neighbor classifiers (see Fig. 2.16). The training time is much less than the classification time required by back-propagation classifiers.

Radial Basis Function Classifiers (RBF)

The radial basis function (RBF) classifiers are kernel classifiers that use the method of potential functions. RBF classifier consists of an inputs layer, a kernel nodes layer and an output nodes layer. Kernel nodes compute radially symmetric functions (typically Guassian shaped functions) that are maximum when the input is near the centroid of a node. Weights from kernel nodes to output nodes are obtained using the LMS algorithm or matrix based approaches that require simultaneous access to all training examples.

RBF classifiers have similar error rates compared to the back-propagation classifiers, however, they train much faster at the expense of requiring a few times as many connection weights (for example one study reported four minutes for RBF classifier versus three hour for back-propagation classifier, however, the number of weights in the RBF classifier was roughly five times the number of weights in the back-propagation classifier).

Time Delay Neural Net Classifier (TDNN)

Time delay neural network (TDNN) classifiers are dynamic back-propagation classifiers. The inputs to the TDNN consist of current and finite number of delayed samples of the time input signal.

The TDNN classifiers are capable of modeling systems where the output has a finite temporal dependence on the input. The TDNN classifiers have been applied successfully to speech recognition and to nonlinear time series prediction problems.

Recurrent or Feedback Neural Network Classifiers [37]

Recurrent networks are neural networks that have feedback loops from all or some of the nodes' outputs to all or some of the nodes' inputs. The feedback loops may or may not have time delay units. Recurrent networks can offer a great advantage over feedforward networks. For many problems, a small system with feedback is equivalent to a large or infinite feedforward system. Several recurrent networks have been introduced recently. They include neural networks with output feedback, networks with state feedback, continuous-time Hopfield nets [37], discrete-time Hopfield nets, continuous-time recurrent nets (CTRNN), discrete-time recurrent nets (DTRNN).

This class of networks is inherently recursive (so that they are called recurrent nets) because they incorporate feedback. It is difficult to develop meaningful learning algorithms for recurrent networks, and typically recursive and complex learning algorithms are needed.

Wavelet Network [40]

The wavelet network has been introduced recently. It has a structure similar to that of the multi-layer perceptron. The wavelet network is compatible with the wavelet transform. With wavelet networks it is possible to obtain the same performance as the MLP with a reduced number of nodes. Wavelet nets can be trained using a back-propagation type of supervised learning algorithm.

2.3.4.2 Neural Networks for Automatic Speech Recognition

Several neural network classifiers have been applied in speech recognition. Most of the work that has been done recently concentrates on phoneme, sub-word, and isolated-word recognition. Recently, recurrent networks have been applied to word spotting and continuous speech recognition. The classifiers that have been investigated include:

1. multi-layer perceptron,
2. learning vector quantizer,
3. feature map,
4. radial basis function classifier,
5. time-delay neural network classifier,
6. recurrent neural network classifier.

The performance of the neural network classifiers is comparable to the performance of the conventional HMM and DTW classifiers. Some authors reported slightly better performance than the HMM classifiers (about 1 to 3 %), especially in the case of the

TDNN classifier for phoneme recognition. However, some authors have reported slightly lower recognition rates than the HMM classifiers.

Similar performance among the various neural network classifiers has been reported, however, wide differences in the training time, memory requirements, computations power, and the complexity of the learning and the recognition algorithms have been observed.

2.3.4.3 Neural Network Speech Recognition System Architectures

Speech recognition systems have a relatively large number of inputs and outputs (i.e. a phoneme recognition system typically has 8 to 20 inputs, such as 12 LPC parameters or 16 mel-scale spectral components, and about 50 outputs that represent English phonemes). This makes the training time long, the error rates higher, and the recognition speed lower than in small systems. To improve the efficiency and performance of the neural network classifiers for speech recognition, several architectures have been introduced, where the system is divided into parallel and/or cascaded small sub-systems. Also integrated neural network-HMM, and integrated neural network-DTW classifiers have been introduced to combine the advantages of neural networks in sub-word and phoneme recognition and the advantages of the HMM and DTW classifiers in solving the time normalization in word recognition. The architectures that have been proposed include :

1. single network (such as one MLP),
2. neural tree network (see Fig. 2.17),
3. two-level hierarchical structure for phoneme recognition (see Fig. 2.18),
4. other integrated and hierarchical structures,

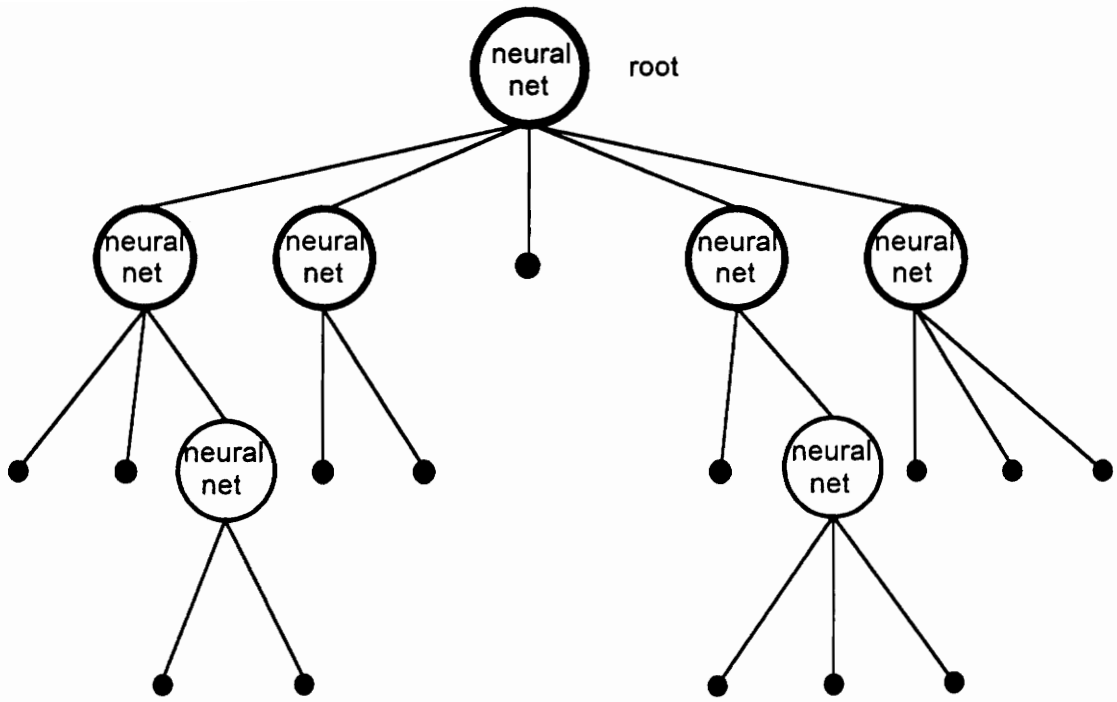


Fig. 2.17 Neural tree network.

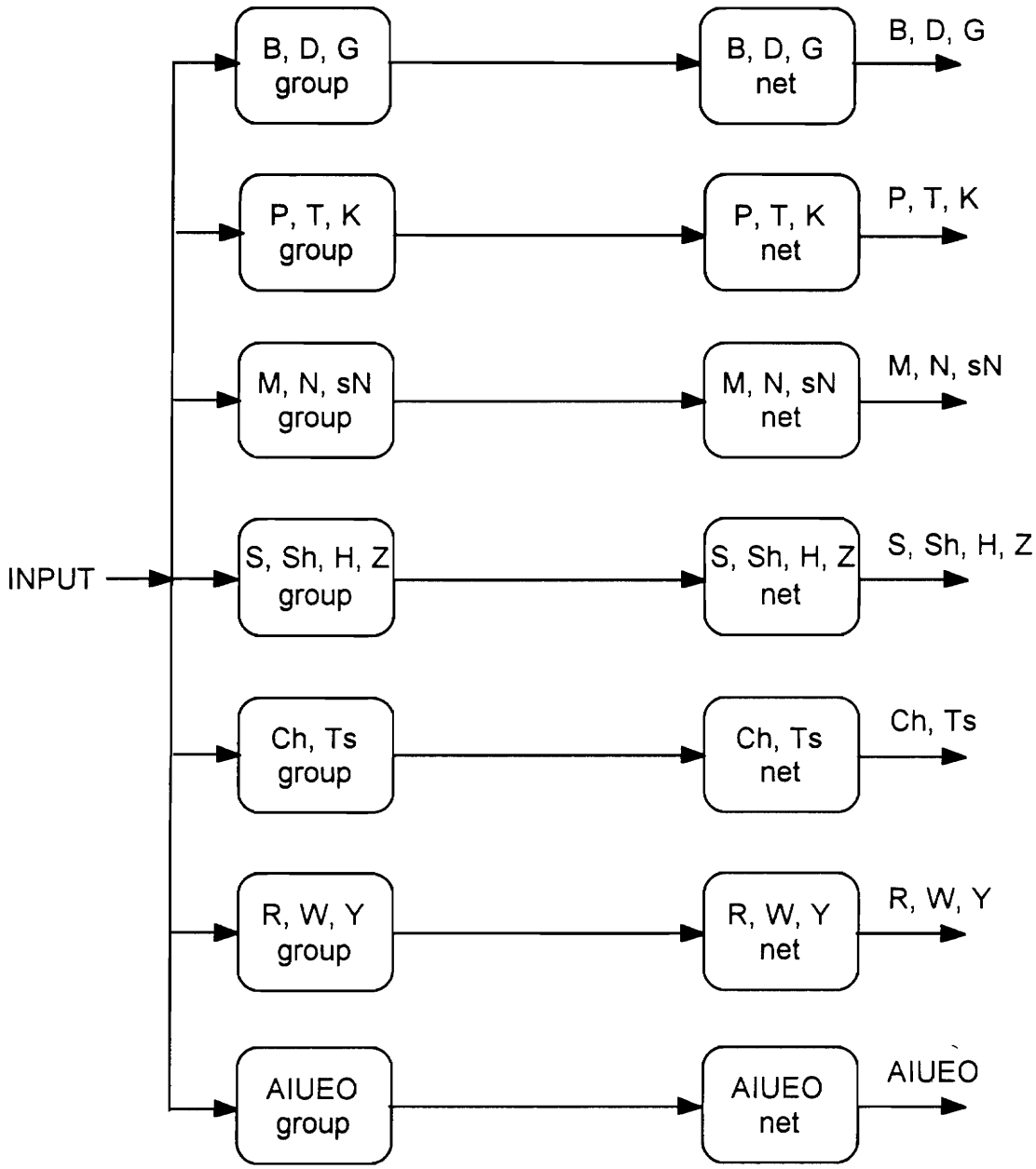


Fig. 2.18 A two-levels hierarchical structure for Japanese phoneme recognition.

5. integrated neural network-HMM system (see Fig. 2.19),
6. integrated neural network-DTW system (see Fig. 2.20).

2.3.4.4 Training and Learning Algorithms

Several learning algorithms have been introduced for neural network classifiers depending on the classifier used for recognition. These algorithms include :

1. supervised learning using back-propagation algorithms,
2. unsupervised learning using k-nearest neighbor clustering algorithms,
3. combined supervised/unsupervised learning, such as in the case of LVQ and feature map,
4. competitive learning (CL) and differential competitive learning (DCL) algorithms,
5. discriminative training,
6. fuzzy training.

Unsupervised learning, CL, and DCL algorithms are much faster than supervised learning algorithms, however, the unsupervised learning algorithms are more complex than supervised ones.

The generalization capability and the recognition rate of neural network classifiers depend on the training algorithm and on the training data. Typically back-propagation classifiers require long training times compared to DCL, CL, and unsupervised learning to get similar performance. However, the generalization capability of any classifier increases with the size and diversity of the training data.

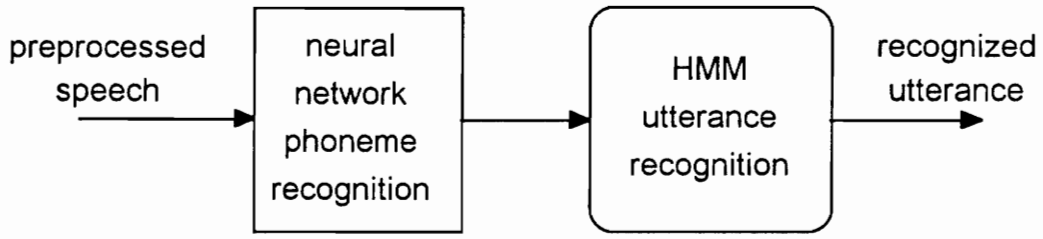


Fig. 2.19 Integrated neural network-HMM speech recognition system.

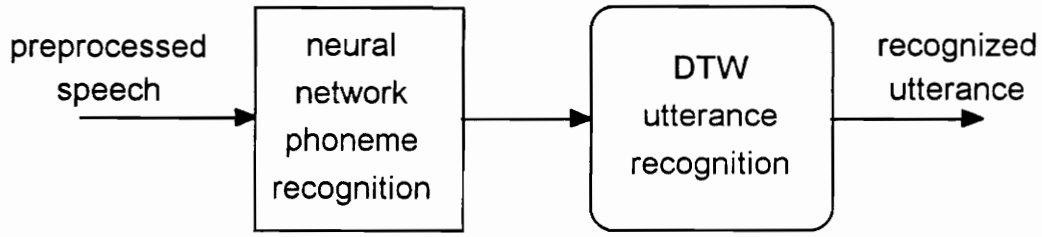


Fig. 2.20 Integrated neural network-DTW speech recognition system.

2.4 Recognition Performance Assessment [2, 7, 127-128]

Assessing the performance of a speech recognition system is not a simple task because the recognition accuracy depends on several factors such as the speaker, the vocabulary size and content, the noise level, and the syntax employed in a particular application.

In recent years a number of groups have been working on techniques, standards, databases and speech corpus for recognition applications, and guidelines for assessing the performance of speech recognizers.

2.4.1 Performance Measures

The basic measure of performance is the recognition rate that is defined as :

$$\text{recognition rate} = \frac{\text{No. of correctly recognized units} \times 100}{\text{No. of test units}}. \qquad (2.12)$$

To have a meaning, any report of the recognition rate of a system should also include a full description of the database used, the noise level and type, and any other details of the assessment test.

For many applications, the analysis of the error is important. Typically, there are three types of error rates. First is the substitution rate that is defined as :

$$\text{Substitution rate} = \frac{\text{No. of substituted units} \times 100}{\text{No. of test units}}, \qquad (2.13)$$

where the number of substituted units is the number of times one unit is confused with another (such as 'ten' is misrecognized as 'then').

The second type of error is the deletion rate that is defined as :

$$\text{Deletion rate} = \frac{\text{No. of deleted units} \times 100}{\text{No. of test units}}, \tag{2.14}$$

where the number of deleted units is the number of times one unit is confused with non-speech (such as the phone 'sh' is misrecognized as noise).

The third type of error is the insertion rate which is defined as :

$$\text{Insertion rate} = \frac{\text{No. of inserted units} \times 100}{\text{No. of test units}}, \tag{2.15}$$

where the number of inserted units is the number of times non-speech is confused with a speech unit (such as high level noise is misrecognized as 'sh').

The error rate is the sum of the substitution rate, the deletion rate and the insertion rate.

In many applications a reject facility is included. If any input does not belong to the legal inputs of the system then the input is ignored. When a reject capability is implemented, the rejection rate test is needed. The rejection rate is defined as :

$$\text{Rejection rate} = \frac{\text{No. of rejected responses} \times 100}{\text{No. of test units}}. \tag{2.16}$$

2.4.2 Databases

In this section, a brief summary of important guidelines for database construction and selection for speech recognition assessment and testing is provided. When constructing or choosing a database for performance assessment it is important to realize the differences among different applications and speakers. The database should represent the user population and the vocabulary must represent the vocabulary employed in each application.

The main guidelines for selecting and/or constructing a database for recognizer assessment are:

1. The database should contain a wide variety of speakers. Speakers from different sexes, ages, and speakers of different regional dialects should be included.
2. The conditions under which the database was recorded, such as signal to noise ratio and the hardware used for recording, should be specified.
3. The database should reflect as much as possible the vocabulary used in the application.

In recent years, several databases have been introduced for different applications ranging from connected digit corpus (database) to a very large switchboard corpus (240 hours of speech). In this research, the DARPA acoustic-phonetic continuous speech corpus, TIMIT [127], was chosen for recognition performance assessment.

The TIMIT database contains speech from 630 speakers from 8 major dialects of American English, each speaking 10 phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic, and word transcriptions as well as speech waveform data for each sentence-utterance. Text corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI), and Texas Instruments, Inc. (TI). The speech was recorded at TI under the following conditions:

1. 20 kHz sampling rate and downsampling to 16 kHz for distribution,
2. a Sennheiser head-mounted microphone in a quiet environment was used for recording.

The speech was transcribed at MIT; and verified and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST).

2.4.3 Guidelines For Performance Assessment

A set of guidelines for assessing the performance of speech recognizers has been provided by the IEEE and the US National Bureau of Standards. These have been summarized in [128]. The intent of these guidelines is to provide a benchmark for the comparison of performance characteristics. The main recommendations are summarized in the following:

1. Speaker population:

(a) The speakers should be representative of the user population.

(b) Speakers' characteristics, such as age, sex, dialect history, speaker training, speech science knowledge, and motivation and/or fatigue should be reported.

(c) The number of speakers should be large enough to statistically represent each significant characteristic of the population.

(d) The acoustic environment for the speakers and the channel over which data is transmitted should be relevant to the operational environment.

2. Vocabulary:

(a) The size and content of the vocabulary should be reported.

(b) In syntactically constrained tasks, complete description of the grammar, frequency of the transitions from each task to the next, and the average branching factor, should be reported.

3. Testing methodology:

(a) Enough data must be processed to provide statistically meaningful results.

(b) Complete separation of the training and test data should be maintained.

(c) Only one parameter should be varied at a time during the test if possible.

(d) Several measurements should be made to assess the recognition performance.

(e) The effect of factors such as vocabulary size, syntax, dynamic range and noise tolerance should be documented.

In general, complete assessment of a speech recognition system must include also hardware specifications and performance assessment, and system performance measures such as usability, acceptability and throughput. In this report we are interested in the recognition performance assessment, so that the hardware and system performance assessment are not considered. For information about hardware and system performance assessment see [2].

Chapter 3: FEATURE EXTRACTION AND ANALYSIS OF SPEECH SIGNALS

3.1 Introduction

It is well known that the selection of features greatly affects the performance of pattern classifiers. Several feature sets; spectral features, LP-based (linear prediction-based) feature sets, and cepstral features are extracted and used in the analysis and characterization of speech signals. In this chapter the feature sets employed in this research and the algorithms used to extract them are described.

The phonemes in the TIMIT database are analyzed to obtain a better understanding and modeling of speech sounds. The analysis is design oriented. There are two main objectives of the analysis stage: The first objective is to study the variability sources and characteristics caused by the speaker and the non-stationary nature of speech sounds. The second objective is to study the separability of phonemes and groups of phonemes using multilayer perceptron (MLP) and radial basis function networks (RBFNs). The main

results and conclusions of the variability and reparability studies are summarized in this chapter.

In this chapter, a reduced set of training and test data is used. The analysis results of Chapter 3 are used to design the isolated-word system in Chapter 4.

3.2 Preprocessing and Feature Extraction

Typically, the speech waveform is segmented into a sequence of short-time frames. Spectral features, LP features, and cepstral features are extracted. Spectral features are developed using two principal techniques: 1. Features are extracted from the smoothed FFT of speech frames. 2. Filter banks are employed to obtain spectral features. In Section 3.2.1-3, the feature sets and the algorithms used to extract them are described.

3.2.1 Spectral Features

Spectral features are extracted using two approaches: 1) Smoothed FFT, and 2) mel-scale filter banks. Important issues are the normalization and resolution of the spectral features.

A. Smoothed FFT

Spectral analysis can be done and spectral features are extracted from smoothed FFT of speech signals. The main steps of feature extraction from a smoothed FFT are:

1. Segment the speech signal into a sequence of short-time records or frames.
2. Multiply each frame by a Hamming window.
3. Compute the magnitude of the FFT of each frame.
4. Smooth the magnitude FFT using low pass filters (use FIR filters with 20 taps and 0.14 normalized corner frequency).
5. Normalize the magnitude of the smoothed FFT by its maximum peak value. When frequencies are taken as features, normalize the frequencies by the maximum peak frequency.
6. Extract spectral features (by either uniform or non-uniform sampling) from the smoothed amplitude FFT. (See part C of this section for details).

B. Filter banks

A 16-channel mel-scale [151] filter bank is implemented. Mel-scale is a frequency scale that is based on the perception of frequencies by humans (see Section 2.1.2.2 and Fig. 2.7), where the frequency resolution is high at low frequencies and decreases logarithmically when the frequency is increased. FIR filters have been used. Constant Q filters are implemented where the center frequencies are placed on a logarithmic scale.

Figure 3.1 shows the amplitude response of the filters. It has been found that relatively long FIR filters must be used to minimize the overlap between individual filters. In our design L=160, where L is the length of the FIR filter. The output of each FIR filter is down sampled from 512 samples to 20 samples to reduce the computation time.

MATLAB [129] is used to design an FIR filter bank when the center frequencies and the bandwidths are provided by the user. The filters' bandwidths and center frequencies are shown in Table 3.1.

The main advantage of using filter banks for spectral feature extraction is the ability to perform multi-resolution spectral analysis.

C. Averaging and normalization

A significant objective is to represent each phoneme by a minimum number of feature vectors. This can be achieved by averaging and normalizing a large number of spectra of each phoneme spoken by many speakers. A simple normalization scheme is employed. The main steps of the averaging/normalization of spectra are:

1. Calculate the smoothed FFT of many frames of each phoneme spoken by many speakers.
2. Calculate the average location of the maximum peak of the spectrum of each phoneme.
3. Normalize the frequency axis of each phoneme spectrum by the average peak frequency of the phoneme spectrum. Normalization is accomplished by decimation and interpolation of the spectrum.
4. Normalize the spectrum of each frame by its maximum peak.

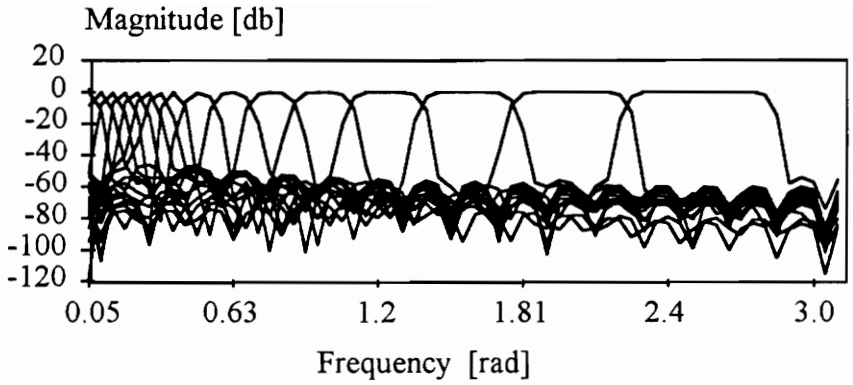


Fig. 3.1 16-channel mel-scale filter bank used to extract spectral features from speech signals.

Table 3.1 The FIR mel-scale filter bank bandwidths and center frequencies used to extract spectral features.

Channel	Center Frequency	Bandwidth
1	120	120
2	240	120
3	360	120
4	480	120
5	600	120
6	720	120
7	840	150
8	1000	150
9	1260	293
10	1590	370
11	2006	466
12	2530	588
13	3190	742
14	4024	936
15	5075	1180
16	6400	1488

5. Calculate the average spectrum of the normalized spectra of each phoneme.

Two normalized spectra of the phoneme 'sh' spoken by a male and a female speaker are shown in Fig. 3.2 to illustrate the normalization effect. At frequencies below the peak frequency, the distance between the two spectra is significantly reduced.

D. Resolution of spectral feature

Once the smoothed and/or averaged/normalized spectrum of a speech frame is obtained, samples are extracted as features. In the case of a smoothed-FFT, there are several possible ways of extracting features. The following feature extraction procedures are employed:

1. Amplitude samples, that are equally spaced on a logarithmic frequency scale, are selected.
2. Amplitude samples are obtained from non-uniformly spaced frequencies where the shape of the spectrum is captured.
3. Amplitude and normalized frequency samples are extracted as in 2. The frequency range is normalized to the location of the maximum peak of the spectrum.

The recognition rates using MLP for voiced/unvoiced classification are similar for the above three types of spectral features. However, further investigation of the non-uniform sampling procedure seems to offer some improvement potential.

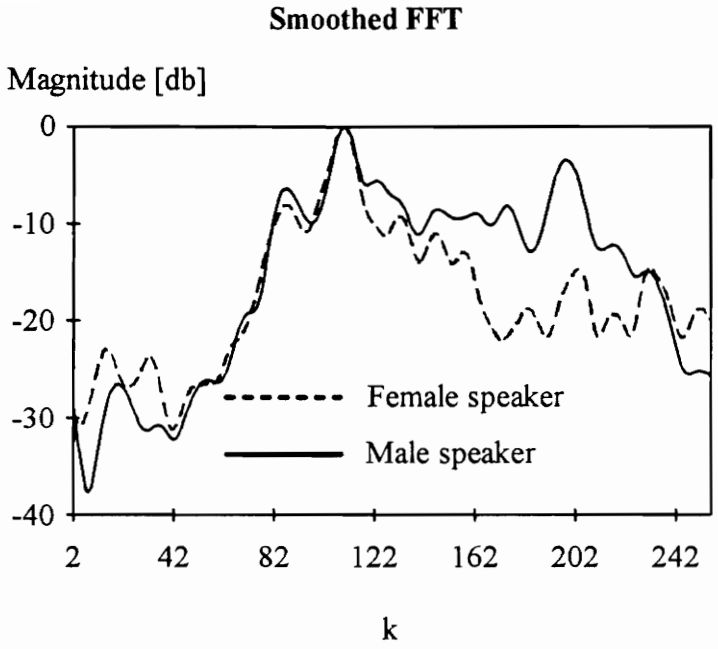


Fig. 3.2 Normalized spectra of the phoneme "sh" spoken by male and female speakers.

When filter banks are used to extract spectral features, the resolution is controlled by selecting the filters. Commonly mel-scale filter banks are used. However, by controlling the bandwidth and center frequency of each filter non-uniform spectral samples can be achieved.

3.2.2 Linear prediction features

LP features are one of the most common features used in speech coding and recognition. It has been observed that the partial correlation coefficients (PARCORs) result in improved recognition rates over the LPCs (linear prediction coefficients). Moreover, the biased autocorrelation method using Levinson-Durbin algorithm [130] provides better recognition rates than the Burg [130] or the covariance [130] algorithms.

In this study the Levinson-Durbin algorithm is used to extract 16 PARCORs for each 32ms frame of speech. Each frame is multiplied by a Hamming window prior to the LP analysis.

3.2.3 Cepstral features

The theoretical background and the main steps in extracting cepstral features were summarized in Chapter 2. However, as was reported in [131, 151], mel-scale cepstral features provide improved performance. The block diagram of a system for mel-scale cepstral feature extraction is shown in Fig. 3.4. A mel-scale smoothed spectrum of each 32ms speech frame is obtained using the 16-channel filter bank shown in Fig. 3.1. Then the logarithm of smoothed spectra is computed to obtain amplitude warping. A discrete

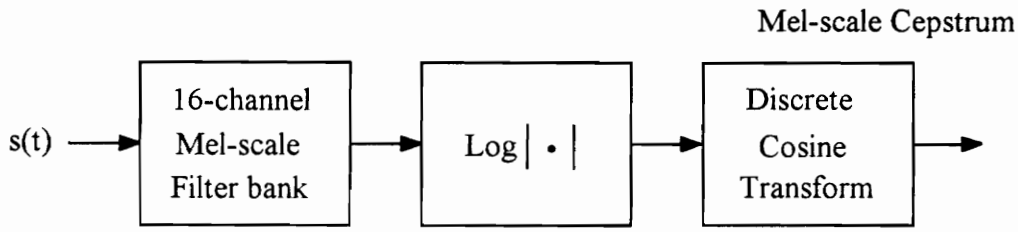


Fig. 3.4 Cepstral features extraction method.

cosine Foureir transform is used to transform the spectral samples to a set of 16 cepstral samples.

3.3 Variability

The variability study is performed on the phoneme level (not word level). The variabilities that are significant for the recognizer design are considered.

The dynamic patterns of each phoneme are investigated. It has been observed that phonemes can be divided into three main categories:

1. Single sound phonemes with one dynamic pattern, such as vowels and fricatives (Fig. 3.5).
2. Double sound phonemes with two dynamic patterns, such as the africates. For example the phoneme 'ch' can be thought of as a stop "t" followed by fricative "sh".
3. Stops and plosives that consist of three dynamic patterns. These are stops that have a closure transient, a silence interval, and a release transient (See Fig. 3.6). These transients reflect the way in which the stops and plosives are produced. The stops are generated by closing the mouth (closure transient) and building up pressure (silence interval) and then releasing the pressure (release transient) by abrupt opening of the mouth.

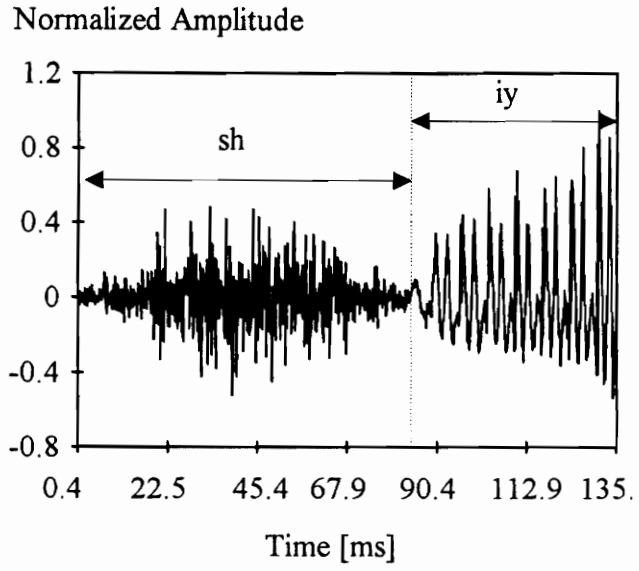


Fig. 3.5 An example of the phonemes "sh" and "iy" in the word "she".

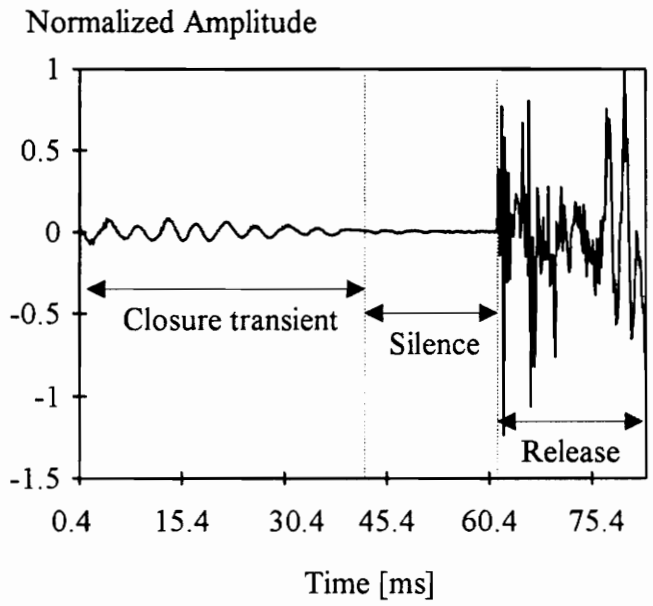


Fig. 3.6 An example of the phoneme "d".

The above observations have been reported in several publications such as [132]. However, we emphasize these dynamic properties due to their influence on the phone classifier design as will be shown in the next chapter.

The speech waveform is segmented into 32 ms frames, and one frame of features is used as the input to the classifier.

The smoothed FFT (magnitude only) of 32 ms speech frames has been used to study some of the characteristics of the TIMIT phonemes. The main characteristics of variability that have been observed are:

1. The variability from one speaker is of similar order of magnitude as the phoneme-specific variability. The smoothed spectra of frames at the beginning and center of the phonemes 'sh' and 'y' spoken by a male and a female speaker are shown in Fig.3.7. This leads to the conclusion that there is no need to split the speakers into several groups to reduce variability.
2. Voiced sounds have more variability at higher frequencies than at lower frequencies. The voiceless sounds have more variability than the voiced sounds (Fig. 3.8). This will affect the design and selection of radial basis function classifiers (see Section 3.4.2)
3. The boundaries between phonemes are not distinct. The transition from one phoneme to a other is rather continuous and gradual. In Fig. 3.9 the word "wash' is shown where the phoneme endpoints are the same as in TIMIT. Note that the transition from one phoneme to the other is continuous, and the endpoints are not distinct, especially between "w"

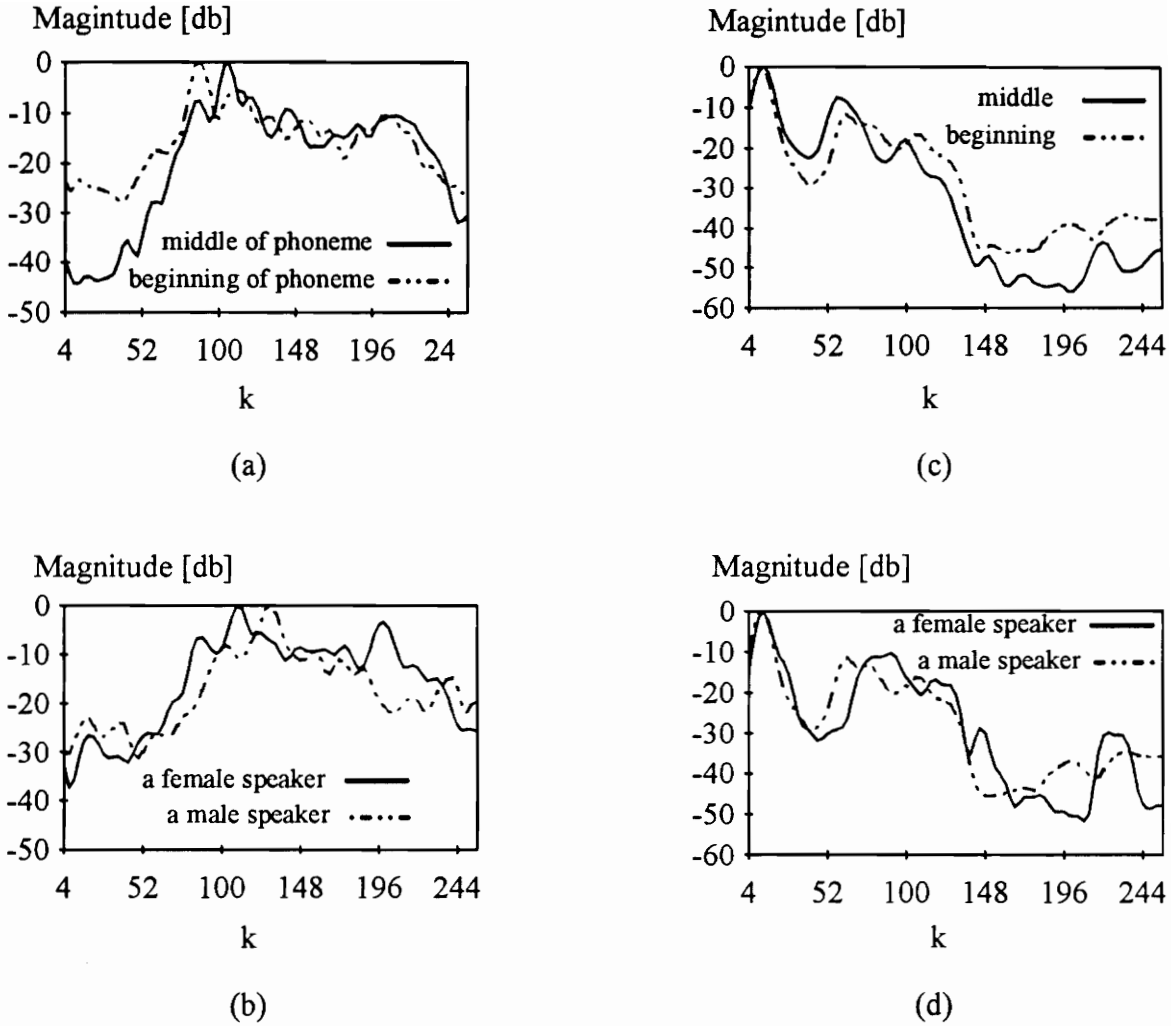
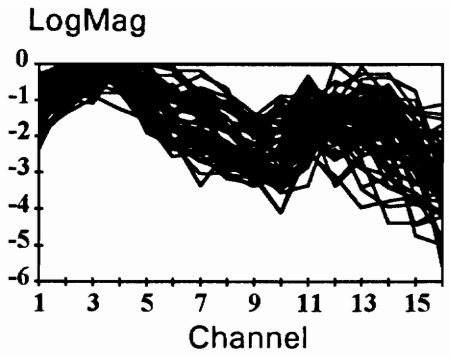
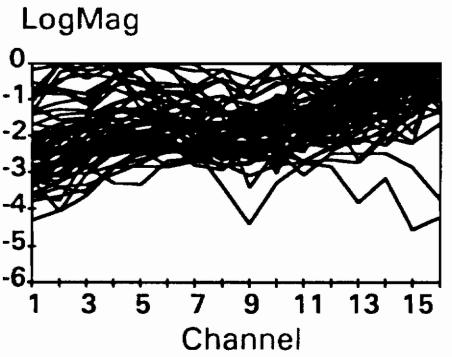


Fig. 3.7 Examples of smoothed FFT of "sh" and "ih". (a) Two frames of "sh"; one at the beginning of the phoneme and the other is at the middle. (b) Two frames of "sh" spoken by female and male speakers. (c) Two frames of "ih"; one at the beginning of the phoneme and the other is at the middle. (d) Two frames of "ih" spoken by female and male speakers.



(a)



(b)

Fig. 3.8 Sixty MSFB samples of the phonemes (a) "iy", and (b) "s".

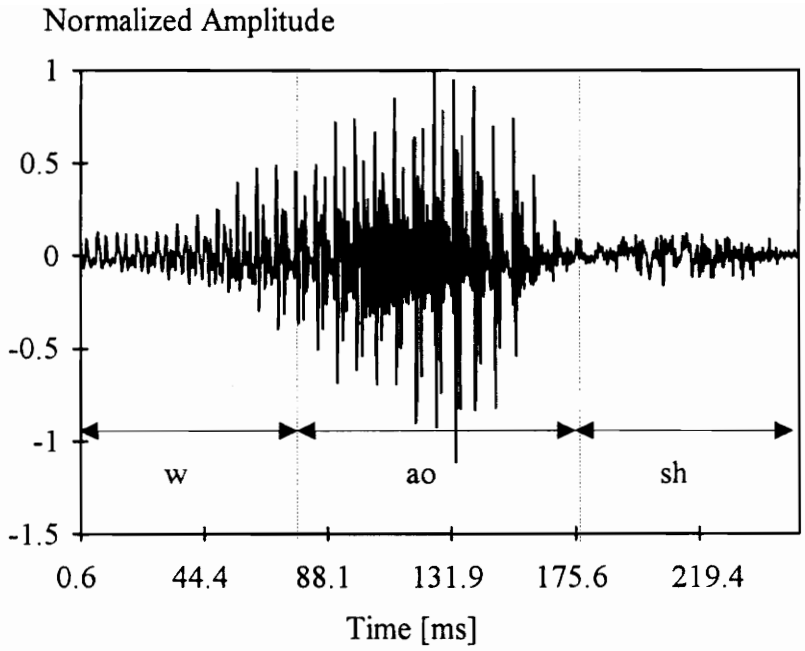


Fig. 3.9 An example of the transition between the phonemes "w", "ao", and "sh" in the word "wash".

and "ao". As a result phoneme labeling and endpoints are not accurate nor unique. This makes supervised training difficult.

4. There are many similar phones in the TIMIT. The spectral centers of "aa", "aw", and "ay" are shown in Fig. 3.10 as an example. As a result, high phoneme recognition rates are difficult ot achieve (see Chapter 4).

3.4 Separability

The objective of the separability analysis is to identify features and classifiers that provide increased separabilty among phonemes. MLP (multilayer perceptron) which is a non-parametric classifier, and RBFN (radial basis function network) classifiers are considered. At the beginning of the research MLP networks were trained to perform phoneme grouping. It was easy to train an MLP to perform voiced/unvoiced classification, but problems arose in the case of classifying similar phoneme groups such as vowels, semivowels, and nasals. As will be shown in Section 3.4.1 the training of MLPs was not successful and all the nasals and semivowels were classified as vowels. As a result the research was directed towards RBFNs where phonemes can be classified based on their statistical properties, such as the means and variances. Knowledge of the mean and variance of each phoneme also provides a way to assess the separability of various phonemes. In addition it was easier to train RBFNs than to train MLPs. RBFNs were thus selected to implement a phone classification system.

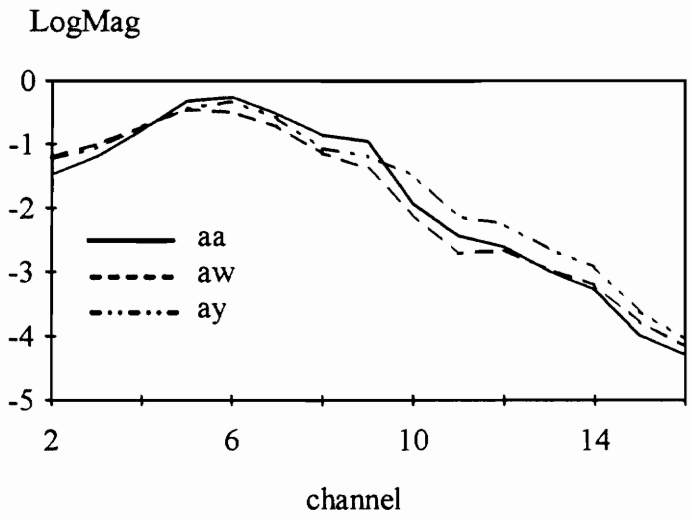


Fig. 3.10 The filter bank based spectral centers of the phones "aa", "aw", and "ay". These three phones have similar spectral centers.

Reduced data sets from the TIMIT corpus were used to perform the separability analysis presented in this section. The training data consists of the sentence "SA1", **"she had your dark suit in greasy wash water all year",** spoken by 8 speakers from dialect regions 5 to 8 in the TIMIT database, one male and one female speaker from each region. The test data consists of 4 speakers from dialect regions 1 and 2 of the TIMIT database, one male and one female from each region.

3.4.1 Separability Using Multi-layer Perceptrons

MLPs were trained to perform phoneme grouping using different sets of spectral features. The main objectives of this study were to select a set of spectral features that increases the separability of groups of phonemes, and to assess the ability of MLPs to perform phoneme grouping. A tree structure, that is based on the phoneme tree shown in Fig. 2.2, was assumed. First an MLP was designed to perform voiced/unvoiced classification (see part B of this section), and second an MLP was designed to perform vowel/nasal and semivowel classification (see part C of this section). The separability analysis using MLPs was performed as described in the following.

A. Feature selection

The following spectral feature sets were extracted:

1. 16 mel-scale filter bank features (MSFB) that were described in Section 3.2.1 part B.
2. 16 equally spaced smoothed-FFT samples on a logarithmic scale (MSSFFT). The first sample is at 62.5 Hz and the last sample is at 7937.5 Hz, and the ratio between each two successive frequencies is 1.38119. The MSSFFT are calculated as described in Section 3.2.1 part A.
3. 16 smoothed FFT samples, non-uniformly spaced to capture the shape of the smoothed FFT (NUSFFT). The NUSFFT are generated as described in Section 3.2.1 parts C and D. The shape of the spectrum is captured as follows:
    (a) The first sample is at the second sample of the smoothed FFT.
    (b) Find all the peaks and valleys in the smoothed FFT. If the ratio between the amplitudes of a valley and peak pair is more than 0.7 then delete them.
    (c) If the distance between the frequencies of a peak and a valley is more than 600 Hz, then insert equally spaced samples between the peak and the valley with a spacing of 500 Hz.
4. 8 amplitude samples and 8 frequency samples from normalized smoothed FFT (NSFFT). The NSFFT are extracted as described in Section 3.2.1 part C. The samples are equally spaced on a logarithmic frequency scale. The first sample is at the second sample of the normalized and smoothed FFT, and the last sample is at the 254th sample of the smoothed and normalized FFT.

B. Voiced/unvoiced classification

A 2-layer MLP network has been trained to make voiced/unvoiced classification. The MLP has 4 nodes in the first layer, 2 nodes in the second layer, and one output node. Further increase of the order of the MLP did not improve the performance of the classifier. The MLP was trained with 8 speakers from dialect regions 5 to 8 of the TIMIT database, one male and one female speaker from each region. The network was trained using the average or center of each phoneme spectral feature over all the training data. The highest recognition rate is 96% and is achieved using filter bank features as shown in Table 3.2. It is important to note that training of the system with a sequence of training data without averaging was not successful.

It is important to note that the response of the MLP to the training data follows closely the desired output as illustrated in Fig. 3.11.

The conclusion is that MLPs can be easily trained to perform voiced/unvoiced classification.

C. Vowels/Nasals, glides, and semivowels classification

Several 2-layer MLP networks have been trained to make vowel/non-vowel classification. The results for an MLP that has 4 nodes in the first layer, 2 nodes in the second layer, and one output node are shown in Table 3.2. Even though the recognition rate is 81% for all the feature sets the training of MLPs was not successful. All the phonemes from the training and test data were classified as vowels. The output response of the MLP to the input training data is shown in Fig. 3.12. The same output is produced for all the phoneme centers. The desired output of the MLP is one for vowels and zero for

Table 3.2 Phoneme grouping with MLP for various feature sets.

Feature	Recognition Rate (%)	
	voiced/unvoiced	vowel/nonvowel
MSFB	96	81.8
MSSFFT	94.7	81.8
NUSFFT	93.9	81.8
NSFFT	93.2	81.8

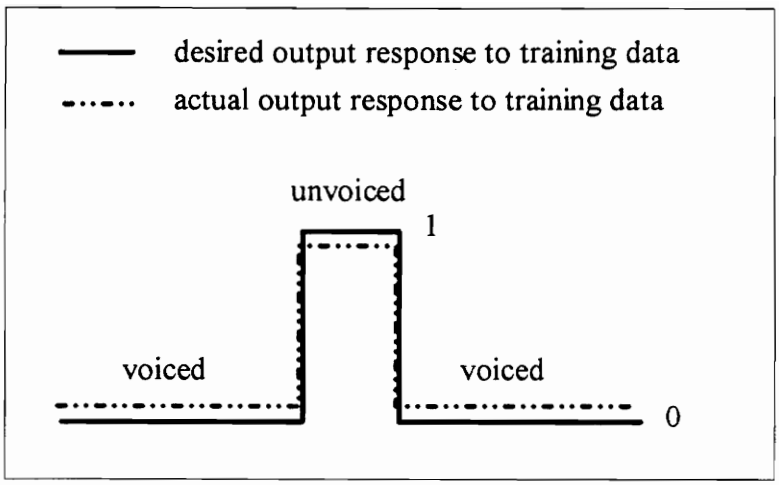


Fig. 3.11 The output response of the voiced/unvoiced MLP classifier to the training data. The MLP output closely follows the desired output.

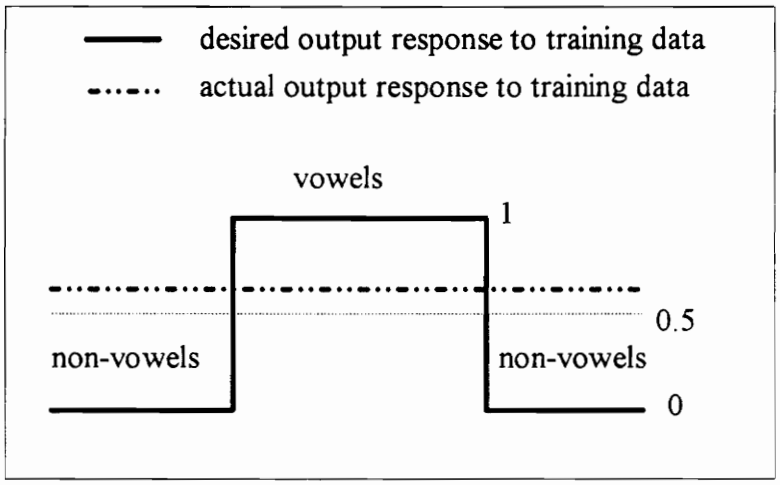


Fig. 3.12 The output response of the vowel/non-vowel MLP classifier to the training data. The MLP output does not follow the desired response.

nonvowels. If the output of the MLP is greater than 0.5 then input frame is classified as a vowel. Because the output of the MLP is greater than 0.5 for all the phonemes they were classified as vowels.

The conclusion is that the separation of vowels from non-vowels using MLPs was not successful. This indicates that voiced phoneme groups have significant overlap. As a result the training and optimization of MLPs for phoneme grouping are difficult. Hence, radial basis functions were selected as an alternative to the MLPs.

3.4.2 Separability Using Radial Basis Function Networks

One way to asses the separability of phonemes is to consider the statistical properties such as the mean and spread or variance of the data from each phoneme. A single layer RBFN (Fig. 3.13(a)) can be used to asses the separabilty of phonemes based on their statistical properties. Because the separability of phonemes is of interest, the parameters of each node in the single layer RBFN are estimated as the mean an variance of the training data from the corresponding phoneme. Clustering is not performed and a second layer is not incorporated.

In this section a group of 6 phonemes is used. The phonemes are "aa", "iy", "w", "r", "s", "sh". The analysis results will be used in designing the phoneme recognition sub-system in Chapter 4.

The separability analysis using RBFNs was performed as described in the following

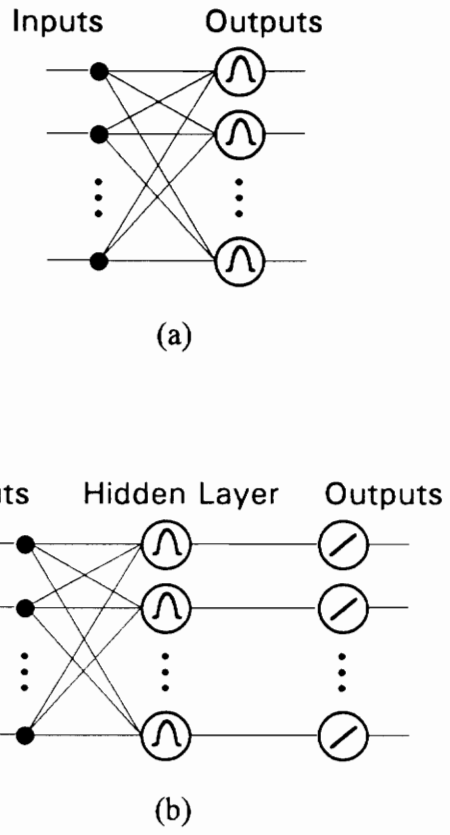


Fig. 3.13 Radial basis function networks: (a) single layer RBFN, and (b) two layer RBFN.

A. Feature selection

The following feature sets were extracted:

1. 16 MSFB (mel-scale filter bank).
2. 16 samples from smoothed FFT (SFFT).
3. 16 samples of smoothed phase of FFT (PSFFT).
4. 16 PARCORs (partial correlation coefficients).
5. Parallel combination of MSFB and PARCORs (FBPAR), where PARCORs are used for unvoiced phonemes, and MSFB features are used for voiced phonemes.
6. Linear combination of MSFB and PARCORs (16 LFBPARs), where

$$LFBPAR = \frac{1}{2}(MSFB + PARCOR). \tag{3.1}$$

7. Series combination of PARCORs and MSFB features (16 SFBPARs), where the first 8 SFBPARs are equal to the first 8 MSFB features, and the last 8 SFBPARs are equal to the first 8 PARCORs.
8. 32 MSFB features.
9. 32 PARCORs.
10. 10 PARCORs.

B. RBFNs design and training

The most common RBFN is a two layer network as shown in Fig. 3.13(b) with Gaussian basis functions. The general form of such a basis function is:

$$u_j = \exp\{-\frac{1}{2}(x-c_j)^T S_j^{-1}(x-c_j)\} \quad , \quad j = 1,\ldots,N_1, \tag{3.2}$$

where u_j is the output of the jth node in the hidden layer, x is the input pattern, c_j is the weight vector, or the center, for the jth node, and S_j is the normalization matrix for the jth node. The output layer can be described by:

$$y_j = w_j^T u \quad , \quad j = 1,\ldots,N_2 \ , \tag{3.3}$$

where y_j is the output of the jth node, u is the output of the hidden layer, and w_j is the weight vector for the jth node.

In general, the design process of a RBFN consists of selecting the network's configuration, the RBF parameters, c_j and S_j, and the number of nodes or the size of the RBFN. Two layer RBFNs are often employed for function approximation and control applications, and in many cases single layer nets are applied to pattern recognition problems. However, the RBFN configuration must be selected based on the application.

To study the separabilty of phonemes based on their statistical properties, a single layer RBFN can be used.

The variable c_j is selected to be equal to the mean of the training data from the corresponding phoneme. In many cases S_j is replaced by a normalization parameter, σ_j. In general, S_j can be of any form, but in practice, the choice of S_j is governed by the

number of nodes of the RBFN and the estimation accuracy of S_j. If S_j is simple it can be estimated with good accuracy, however, the simpler S_j is the more nodes are needed. For the separability study each phoneme is represented by one node. A diagonal normalization matrix can practically be used, where the estimation accuracy is reasonable and the number of nodes can be minimized. In addition speech patterns have different variances around their centers in different dimensions. Due to that the RBFN nodes' centers are taken as the phonemes means and not generated using clustering, the node normalization matrix might not be optimal. Therefore, the normalization matrix for each node is estimated as follows:

1. S_{j1} is estimated as the variance matrix of the data from the jth phoneme.
2. S_{j2} is estimated as a diagonal matrix based on the distances among the phoneme centers. Let $d_{j,i}$ denote the ith element of diag$\{S_{j2}\}$, and $c_k(i)$ denote the ith element of the kth phoneme center. Then

$$d_{j,i} = \min_k\{(c_k(i) - c_j(i))^2\}, \quad i = 1,\ldots,N_p, \tag{3.4}$$

   where N_p is the number of phoneme centers which is equal to 6.
3. S_j is a linear combination of S_{j1} and S_{j2}, where

$$S_j = bS_{j1} + (1-b)S_{j2}, \tag{3.5}$$

   where b is a constant scalar.

In the separability study six phonemes are used so that the number of nodes in the RBFN is six.

C. Experimental results

The training data consists of the sentence "SA1", **"she had your dark suit in greasy wash water all year",** spoken by 8 speakers from dialect regions 5 to 8 in the TIMIT database, one male and one female speaker from each region. The test data consists of 4 speakers from dialect regions 1 and 2 in TIMIT, one male and one female from each region. The phonemes "aa", "iy", "w", "r", "s", and "sh" are used in the analysis. Several feature sets are employed. For each feature set the recognition rate is evaluated for b=0, 0.1, 0.2, ..., 1. The highest recognition rate achieved and the corresponding values of b for each feature set are summarized in Table 3.3.

For each feature set there is a different value of b that maximizes the recognition rate. For PARCORs the highest recognition rate is achieved with b=1. However, for the other feature sets the optimal value of b is less than 1.

The highest recognition rate is achieved with the FBPARs where PARCORs are used for unvoiced sounds and MSFBs are used for voiced sounds. The smoothed phase of FFT samples provides the lowest recognition rate.

Features that consist of combinations of MSFBs and PARCORs provide recognition rates between the recognition rates achieved by MSFBs and PARCORs.

A frame length of 32ms provides slightly higher recognition rates than a frame length of 20ms, or 10 ms.

PARCORs provide better separation of "s" and "sh", while MSFBs provide better separation of "aa", "w", "iy", "r". Due to the fact that about 40% of the training data

Table 3.3 Phoneme recognition rates for various feature sets and using RBFNs. When the frame length is not equal to 32ms it is indicated in the tables.

Feature	b	Recognition Rate (%)
MSFB-16	0.9	71.83
PARCOR-16	1.0	75.35
SFFT (32ms)	0.7	71.47
SFFT (20ms)	0.7	70.02
SFFT (10ms)	1.0	67.27
SPFFT	0	23.94
FBPAR	0.9	81.69
LFBPAR	1.0	73.23
SFBPAR	0.7	72.53
MSFB-32	0.7	71.47
PARCOR-32	1.0	74.64
PARCOR-10	1.0	73.23

consists of "s" and "sh" frames, PARCORs provided higher recognition rates than MSFBs.

Mel-scale smoothed FFT provides similar or slightly lower recognition rates than MSFBs.

3.5 Summary

In this section several feature sets were used for phoneme recognition. MLPs and RBFNs were used to study the separability of phonemes. Some of the important characteristics of speech sounds that affect the design of speech recognizers were reviewed and emphasized.

FBPARs are the only combination of MSFBs and PARCORs that resulted in improved separation of phonemes.

MLPs can perform voiced/unvoiced classification with high accuracy, however it is difficult to train MLPs to separate similar groups of phonemes such as vowels and nasals.

RBFNs are easier to train and therefore they can offer greater separability of phonemes than MLPs.

For the full system design RBFNs will be used with MSFBs, PARCORs, FBPARs, and cepstral features (CEPs). FBPARs have offered the highest recognition rate on the reduced set of six phonemes, while other combinations of PARCORs and spectral features did not appear to add any significant increase of the recognition rate. Hence, FBPARs is the only combination of MSFBs and PARCORs that will be considered in the full system design process (see Chapter 4).

Chapter 4: AN RBFN-BASED SYSTEM FOR SPEAKER-INDEPENDENT SPEECH RECOGNITION

4.1. INTRODUCTION

Recently robust and/or speaker-independent automatic speech recognition (ASR) systems have been an area of increasing interest [133]. The performance of ASR systems has been improved using statistical modeling approaches which try to take some of the variability into account [133]. Dynamic time warping (DTW) and hidden Markov modeling (HMM) have been applied successfully to speech recognition. HMM is the most popular statistical technique that is employed in ASR systems. Artificial neural network (ANN) classifiers have been applied to speech recognition. The performance of ANN classifiers is comparable to that from HMM-based classifiers. However, the research is still

in its early stages, and there are still many open research issues regarding the selection of features, topology and architecture, as well as training strategy.

Currently speaker-independent systems are developed based on two main approaches: The first is based on speaker-invariant feature extraction and uses databases that are speaker independent [42, 125]. The second is based on speaker adaptation and normalization techniques that originally were used for speaker-dependent systems [34-35, 45].

To develop a completely robust ASR system, several variability sources, such as talker, environment, microphone, and communication channel must be addressed. In [134-137] the problem of ASR in a noisy environment has been addressed. Others have studied the ASR problem over a phone channel [32]. One of the main variability sources is the talker; however, current speech recognition systems are significantly non-robust, especially when the group of users is large.

The most robust speech recognizer known to us is the HSR (human speech recognition) system. The HSR system can be modeled by a bottom-up layered hierarchy, and it is believed that feedback is not common between the system layers [138]. Furthermore, the HSR system outperforms any artificial speech recognition system especially at the phone and syllable levels. Therefore, the bottom-up approach to automatic speech recognition system design is biologically motivated in addition to its simplicity compared to top-down and blackboard approaches. A typical bottom-up layered hierarchy of ASR system is shown in Fig. 4.1.

In our study we address speaker-independent ASR systems. Other sources of variability, such as noise, microphone, and channel, are not considered in this work. The TIMIT speech corpus is used to study and model the characteristics of the phonemes of

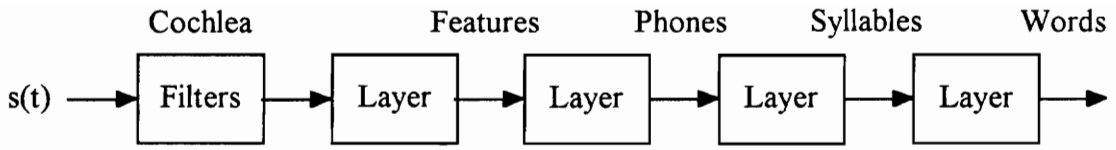


Fig. 4.1 Simplified model of the human speech recognition (HSR) system.

American English. Speaker-invariant features are selected and extracted. A hybrid RBFN-DTW system is employed to enhance the ASR system performance.

Several feature sets are extracted. The speech signal is segmented into a stream of 32ms frames. Mel-scale filter bank (MSFB), reflection coefficients (also called PARCORs), and cepstral features are extracted. A parallel combination of MSFB and PARCORs is also considered due to the fact that it has offered improved phone recognition rate on a set of 6 phones.

Radial basis function networks (RBFNs), that have been applied to speech recognition [117,139-141], are employed as pattern classifiers. The main objective of our study is to demonstrate the feasibility of using RBFN's to enhance speech recognition. Important issues, such as network structure, training strategy and set-up, generalization capability, and training time, are of main interest. In particular a guided or channel RBFN (Fig. 4.2) is introduced and used to make the training process easier. Moreover, simultaneous access to all the training data is not needed. In fact, the training and design of neural nets and the selection of features and training data are all related to each other.

This chapter consists of six sections. In Section 4.2 the ASR system structure is summarized. In Section 4.3 the RBFN design and training are described. In Section 4.4 the design and training of the DTW word recognition sub-system are presented. Some experimental results using the TIMIT speech corpus are provided in Section 4.5. The chapter is summarized in Section 4.6.

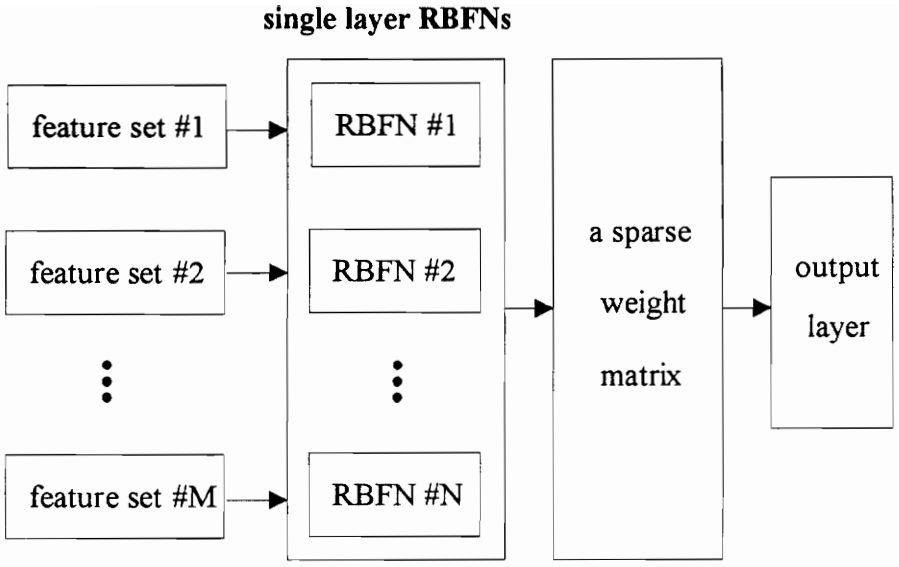


Fig. 4.2 A channel or guided RBFN. There are M feature sets and N single layer RBFNs where each feature set is connected to one or more RBFN. The outputs are connected to the RBFNs through a sparse weight matrix where the zeros in the weight matrix are permanent and not changed during training.

4.2 System Description

The system block diagram is shown in Fig. 4.3. The ASR system consists of three main cascaded layers: feature extraction, RBFN for phoneme recognition, and a DTW isolated word recognition sub-system. A brief description of the system components is given in the following section.

4.2.1 Feature Selection and Extraction

Based on the results of the analysis in Chapter 3, the following feature sets have been extracted:

1. 16 MSFBs (mel-scale filter bank spectral features).
2. 16 PARCORs (partial correlation coefficients).
3. 16 FBPARs (parallel combination of PARCORs and MSFBs).
4. 16 mel-scale cepstral features (CEPs).

We have observed that increasing the size of the phoneme set and changing the training and test data affects the recognition rate differently for different sets of features. Therefore the four feature sets are used and compared for the full system design, even though FBPARs have offered the best recognition rate with a reduced phone set as was shown in Chapter 3.

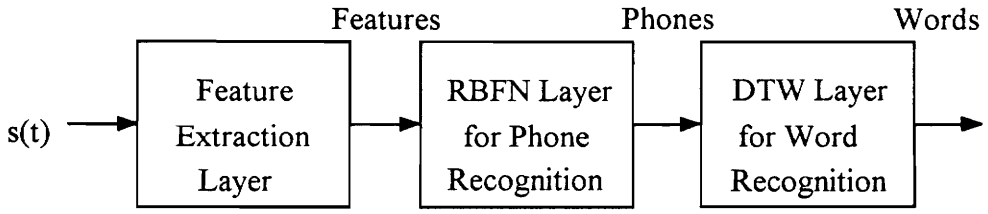


Fig. 4.3. Block diagram of the speech recognition system.

A description of the MSFBs, PARCORs, FBPARs, and CEPs and the algorithms used to extract them are provided in Section 3.2.

4.2.2 RBFN Phoneme Recognition Sub-System

As was mentioned in Chapter 3, the most common RBFN is a two layer network as shown in Fig. 4.4(c) with Gaussian basis functions. The general form of such a basis function is:

$$u_j = \exp\{-\frac{1}{2}(x - c_j)^T S_j^{-1}(x - c_j)\} \quad , \quad j = 1, \ldots, N_1, \tag{4.1}$$

where u_j is the output of the jth node in the hidden layer, x is the input pattern, c_j is the weight vector, or the center, for the jth node, and S_j is the normalization matrix for the jth node. The output layer can be described by:

$$y_j = w_j^T u \quad , \quad j = 1, \ldots, N_2 , \tag{4.2}$$

where y_j is the output of the jth node, u is the output of the hidden layer, and w_j is the weight vector for the jth node.

The variables c_j and S_j, are selected and designed as was described in Section 3.4.2. The mean of the training data from each phoneme is employed as c_j, and the variance is used as the normalization matrix S_j.

In this Chapter, four RBFN topologies, shown in Fig. 4.4, are considered for application to speaker-independent phoneme recognition. RBFNs of type-II and RBFNs of type-IV are compared to the common topologies in Fig. 4.4(a) and Fig. 4.4(c).

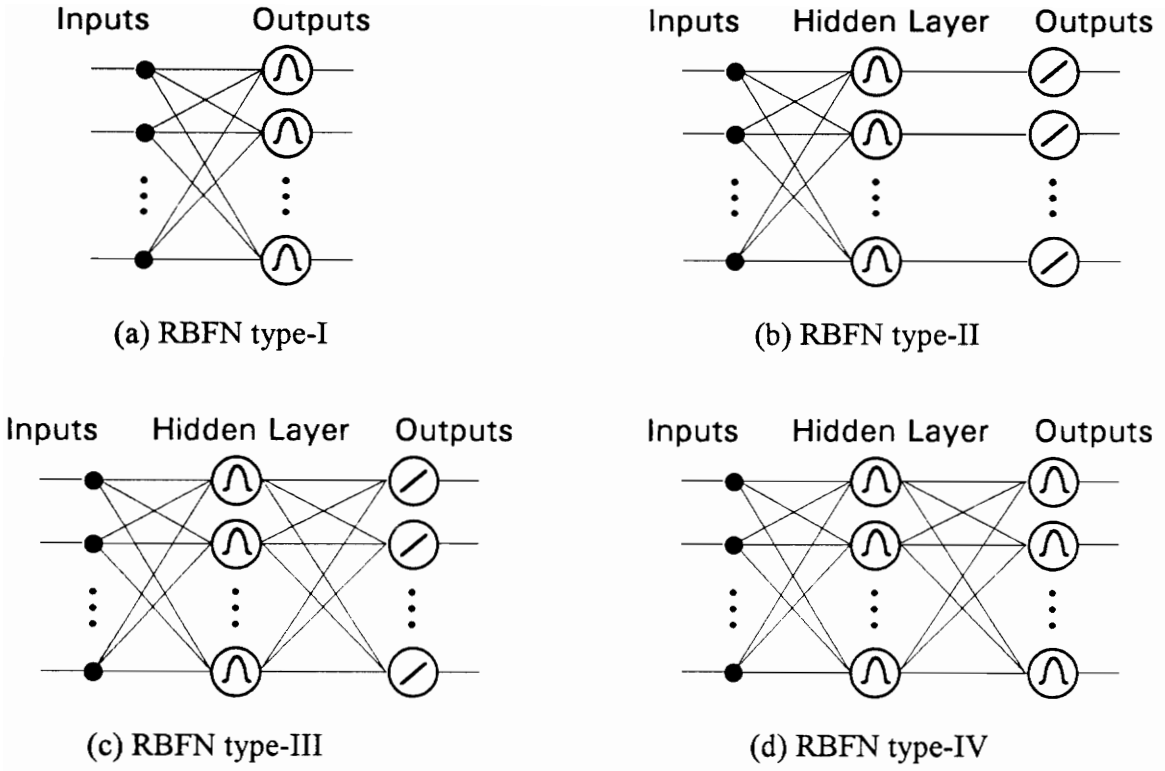


Fig. 4.4 Radial bases function networks used to implement the phoneme recognition layer.

The output layer consists of a vector of weights for RBFNs of type-II, and consists of a layer of RBFs for RBFNs of type-IV. RBFNs of type-II have less complexity and are easier to train than RBFNs of type-III. RBFs of type-IV are highly nonlinear.

The size of the RBFN is selected based on the application. For phoneme recognition the number of output nodes is equal to the number of the phonemes. In the TIMIT data there are 60 phone labels, which give rise to 60 output nodes. However, as will be shown in Section 4.3, we have defined a reduced set of 33 phone labels by merging similar phones. As a result 33 output nodes are needed. The speech patterns were selected as 16-channel MSFB, 16 CEPs, and 16 PARCORs to obtain high recognition rates, so that the RBFN has 16 input nodes. The number of nodes in the hidden layer is equal to, or greater than, the number of output nodes. As will be shown in Section 4.3 and Section 4.5, 33 nodes were used in the hidden layer.

4.2.3 DTW Isolated-Word Recognition Sub-System

DTW [17-25] was selected due to its capability for nonlinear time normalization. In addition the system is designed to perform isolated-word small vocabulary recognition, where DTW can be applied. Each word of the vocabulary is represented by a number of templates. DTW is used to measure the distance between the input word and code book templates. The label of the template closest to the input word is the recognition result. The DTW sub-system design is divided into two tasks: 1) the DTW algorithm, and 2) the code book design, and is summarized in Section 4.4.

4.3 Design and Training of The RBFN for Phoneme Recognition

Several training approaches and algorithms have been introduced for RBFNs [16, 37]. Most of the learning algorithms are based on training the hidden layer and the output layer separately. The RBF layer is first trained with a clustering algorithm to generate the centers and the normalization matrices or the spread parameters of the RBFs. The most popular clustering algorithm is the k-means algorithm [16, 37]. The linear layer is trained with supervision, typically employing the LMS algorithm [37]. In general the learning approach should fit the training data of the given application.

The training data for phoneme recognition has four main characteristics that affect the learning processes:

1. A large amount of training data is needed.
2. There are many similar phonemes.
3. The boundaries between phonemes in a stream of speech are not distinct, i.e. where the transition from one phoneme to the other is gradual and continuous, and must be separated by experts.
4. Experts also label the phonemes in the training data, where the labeling process outcome depends on the expert.

Unsupervised learning using clustering is an attractive choice for phoneme classification and labeling, however, clustering algorithms require simultaneous access to

all the training data, and are sub optimal algorithms, so that their performance degrades when the number of labels is large. In the case of phoneme recognition there are more than 30 labels and simultaneous access to all the training data could be impractical.

A sub optimal learning and labeling strategy for designing the RBF layer is introduced in this section. The training data is extracted from the TIMIT corpus. In the TIMIT data there are 60 phone labels (see Table 4.1), and the speech is labeled and segmented. The learning and labeling process is performed as in the following.

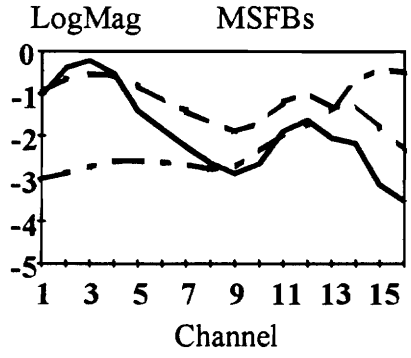
4.3.1 Phone Relabeling and Training of the RBF Hidden Layer

The relabeling and training of the RBF layer is accomplished as follows:

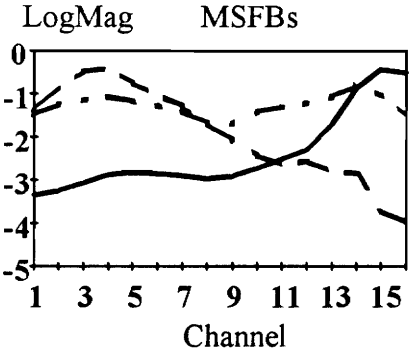
1. Each of the 60 phones in the TIMIT data is considered as a single cluster. Twenty seconds of each phone are extracted and segmented into 625 frames of 32 ms, and MSFB features are extracted. The MSFB features have been selected because they are believed to be rough approximations of the spectral features extracted in the human ear.
2. Each phone data is clustered into three clusters using the k-means algorithm. The cluster that has the maximum number of members is selected to represent the phone. The other two clusters represent segmentation and labeling errors. The clustering results of the phone "s" and "iy" are shown in Fig. 4.5, where we can see that there are voiced/unvoiced segmentation errors. The dotted-dashed line in Fig. 4.5(a) is very similar to the solid line

Table 4.1 The TIMIT phone labels vs. the reduced phone set labels.

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



(a)



(b)

Fig. 4.5. Cluster centers of the phonemes (a) "iy", and (b) "s". Solid lines represent the phones and dotted-dashed and dashed lines represent errors and boundaries.

in Fig. 4.5(b) that represent the unvoiced phone "s". This means that the cluster that is represented by a dotted-dashed line in Fig. 4.5(a) comes form unvoiced sounds that were labeled as "iy" due to segmentation and labeling errors in the TIMIT database. The dashed line in Fig. 4.5(b) represents voiced sounds, however, it is labeled as "s" due to segmentation and labeling errors in the TIMIT.

3. Similar phones are merged manually. Looking at the plots of the phone centers we have defined a new reduced phone set which is shown in Table 4.1. There are 33 phone labels in the reduced set.
4. Each of the 33 phones is considered as a single cluster. Forty seconds of each phone are extracted and segmented into 1250 frames of 32 ms, and MSFB features are extracted.
5. Each phone data is clustered into three clusters using the k-means algorithm. The cluster that has the maximum number of members is selected to represent the phone, and its center and variance are used as the corresponding RBF center and normalization matrix.
6. The iterative algorithm shown in Fig. 4.6 has been developed and employed to re-label and re-segment the needed training and test data from the TIMIT database. The vocabulary of the sentence "SA1" has been selected to design the word recognition sub-system. The boundaries between phonemes are modified using the cluster centers of the TIMIT phones generated in step 1. The label of a frame is changed to the label of its neighbor frame if the distance of the frame from the neighbor frame center is less than the distance of the frame from its own center (see Fig. 4.6).

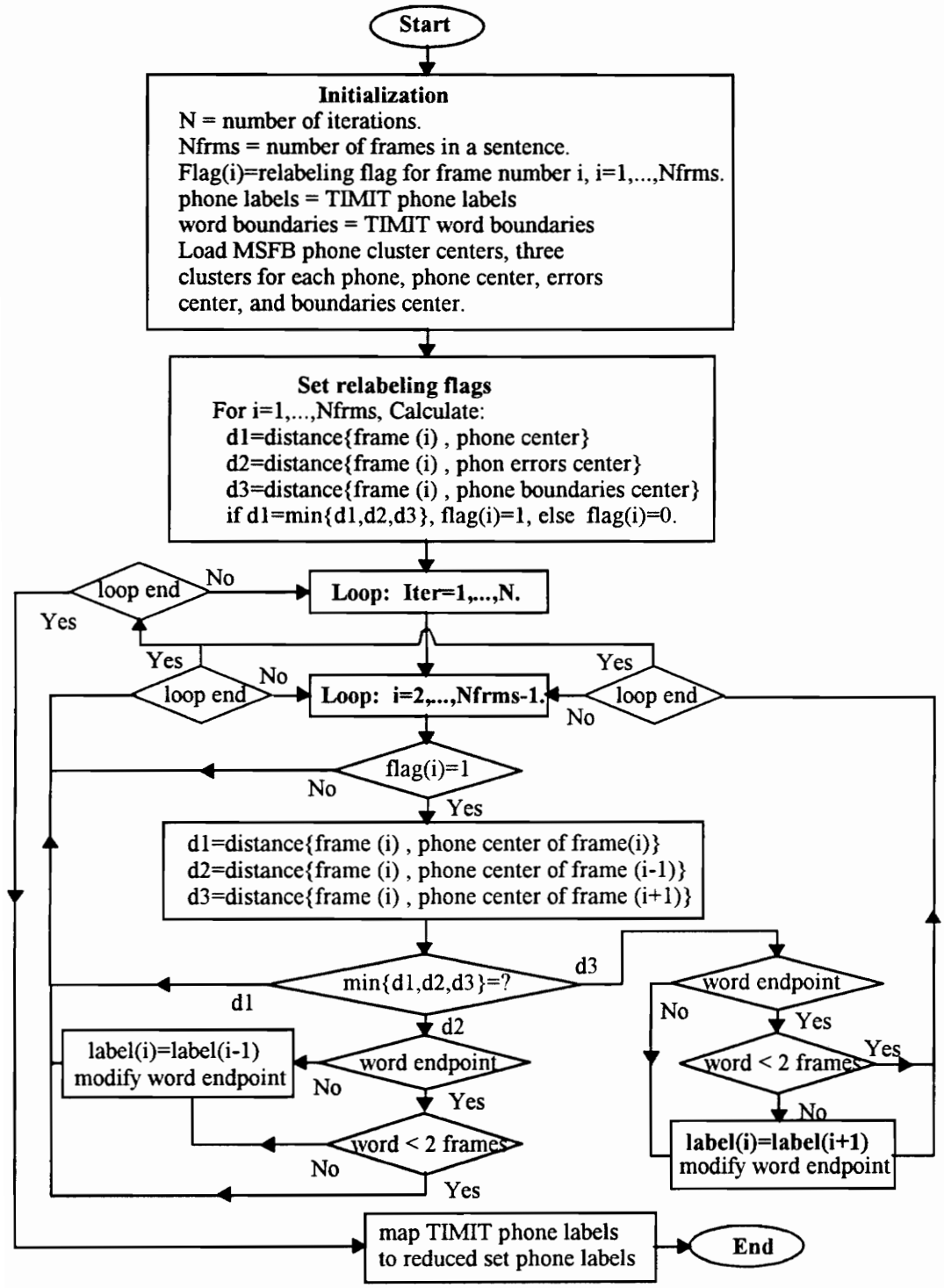


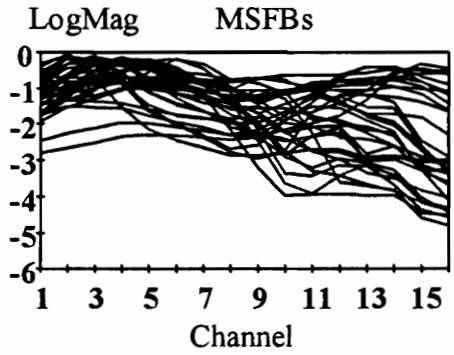
Fig. 4.6 Algorithm for re-labeling and re-segmenting the speech signals.

4.3.2 Training of the Output Layer of Type-II, III, and IV Nets

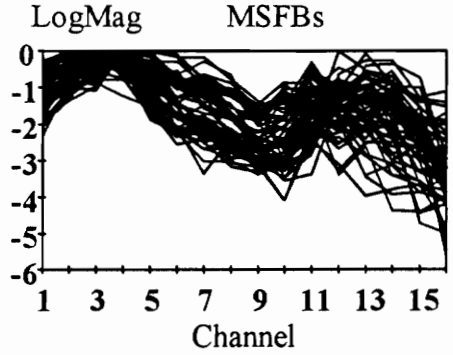
In general if patterns from different populations (from different phones) do not overlap, then a single layer RBFN would be sufficient to perform the classification task. Unfortunately, phonemes have significant overlap. The phone centers are close to each other as shown in Fig. 4.7(a), and the spread of the data from each phone is large. As an example, 60 samples of the phone "iy" are shown in Fig. 4.7(b). In this case a high recognition rate is difficult to achieve. Furthermore, the overlap regions can be divided among phonemes by supervised training of a second layer. The supervised training, if successful, will divide the overlap regions according to the frequency of occurrence of phones in the training vocabulary. This would increase the total recognition rate for the vocabulary used in training.

The output layers of the RBFNs are trained with supervision. For each type of RBFN we have used a different learning algorithm since the output layers differ from each other. In the case of nets of type-II, the output layer was trained using a simple iterative interactive method. The vector of weights was modified manually and iteratively in an interactive way to maximize the phone recognition rate. Although this algorithm is simple, it has yielded more than a 10% increase in the phone recognition rate over that of the LMS algorithm.

In the case of the RBFNs of type-III the LMS algorithm has been employed. The manual and interactive training algorithm becomes unattractive due to the large number of weights, and the learning rate was set to 0.8. To accelerate the convergence of the learning process, the initial values of the weight matrix were generated by adding random weights distributed between 0 and 0.1 to a unity weights matrix. The sentences "SA1"



(a) centers of the 33 phones



(b) sixty sample of "iy"

Fig. 4.7 The centers of the 33 phones and 60 samples of the phone "iy".

spoken by 200 speakers from the TIMIT data were used in the training; 10 training epochs were used.

In the case of a type-IV RBFN, the training was performed in the same way as for the hidden layer. Data from each phone was passed through the hidden layer to generate the input training data for the output layer, where the output from the hidden layer was considered as one cluster, and then steps (4-5) in training the hidden layer were repeated for the output layer. In this case we did not adapt the parameters of the second layer based on the vocabulary. This was to demonstrate two important points: 1) increasing the nonlinearity of the classifier could degrade the performance of the classifier, and 2) supervised training using the desired vocabulary results in better performance than supervised training using text-independent training data. As will be shown in Section 4.5, a type-I net produces better performance than a type-IV net.

4.4 DTW for Isolated-Word Recognition

There are several algorithms for implementing DTW [18-23] depending on the application (isolated-word, continuous speech, small or large-vocabulary). However, for small vocabulary, isolated-word recognition, where each word is considered as a full pattern (such as is the case for voice command and voice-driven menu systems), a direct implementation of the DTW sub-system can be applied.

In the ideal case of error free phoneme recognition, the code book can be generated from a standard dictionary transcription of the vocabulary. However, due to the low

phoneme recognition rates the code book is generated from the training data. Several ways of creating a code book from the training data could be used employing averaging and clustering. When the vocabulary is small, we can use samples of the vocabulary spoken by representative speakers to generate the code book. This way the code book generation is simplified with the expense of increasing its size and the search time. However, because the vocabulary is small, the size of the code book is still acceptable. Sixteen samples of the vocabulary (sentence SA1 in TIMIT) spoken by 16 speakers, one male and one female from each of the 8 dialect regions in TIMIT, have been used as the code book. Further increase of the size of the code book did not offer any significant increase of the word recognition rate.

4.5 Experimental Results

Sixteen isolated-word recognition systems have been designed using the four RBFN types and the four sets of features introduced in Section 4.2. The training data was extracted from the training data set of the TIMIT data. For each of the 60 phones in TIMIT 625 frames of 32ms were extracted. The data were distributed equally over male and female speakers from the 8 dialect regions of TIMIT. All ten sentences spoken by each speaker were used. Forty seconds from each phone of the 33 reduced phone set were extracted and divided into 1250 frames of 32ms. The MSFB, PARCORs, CEP, and the FBPAR feature sets were extracted for each phone. For the supervised training the data consisted of the sentence "SA1" spoken by 200 speakers, 13 males and 13 females from

dialect regions 1-6, 12 males 12 females from region 7, and 12 males and 8 females from region 8. The sentence "SA1", "**she had your dark suit in greasy wash water all year**", consists of 11 words vocabulary.

Two types of vocabulary were used: 1) the first vocabulary consists of all 11 words of the sentence "SA1", where there are similar words such as "year" and "your", "she" and "suit", 2) the second vocabulary consists of 8 words from the sentence "SA1" where the confusing words, "suit", "had", and "your" have been excluded.

The test data was extracted from the TIMIT test data. The sentence "SA1" was used. The number of speakers is 168, and the distribution of the speakers is described in Table 4.2.

Comparison of the performance of the various recognition systems is summarized in Table 4.3 and Table 4.4. The system with type-I RBFN and with MSFB has the best performance on the word level. Systems with MSFB and FBPAR have similar performance, and they outperform the systems with PARCORs and CEPs, although the CEPs yield better performance than PARCORs.

Systems with a type-II RBFN resulted in improved phone recognition over systems with type-I, III, and IV RBFNs. Systems with type-III RBFNs provided some improvement at the phone level over systems of type-I. Systems with type-IV RBFNs have very poor performance both at the phone and word levels. RBFNs of type-I and II are easier to train and implement than RBFNs of type-III.

The FBPAR features have shown improvement over the MSFB at a reduced phone set that includes the phones "s", "sh", "w", "y", "r", and "aa" from 8 speakers. However, on the full phone set the MSFB and the FBPAR provide comparable performances. The conclusion is that recognition results on reduced sets of phonemes may not be valid for the

Table 4.2. Distribution of speakers in the test set.

dialect	Number of Male	Number of Female	Total
1	7	4	11
2	18	8	26
3	23	3	26
4	16	16	32
5	17	11	28
6	8	3	11
7	15	8	23
8	8	3	11
Total:	112	56	168

Table 4.3. Word recognition rates for 168 speakers and the reduced phone set.

Features	Vocabulary size (words)	Word Recognition Rate (%)			
		RBFN Type-I	RBFN Type-II	RBFN Type-III	RBFN Type-IV
MSFB	8	95.31	93.75	91.36	64.70
	11	84.32	80.57	79.16	48.10
PARCOR	8	84.30	85.41	84.22	45.61
	11	70.45	72.34	70.40	36.47
CEP	8	92.26	93.37	91.04	43.89
	11	77.38	79.65	78.12	32.74
FBPAR	8	93.60	93.00	91.81	22.19
	11	82.19	77.81	80.46	18.99

Table 4.4 Phoneme recognition rates for 168 speakers and the reduced phone set.

Features	Phoneme Recognition Rate (%)			
	RBFN Type-I	RBFN Type-II	RBFN Type-III	RBFN Type-IV
MSFB	30.96	44.58	39.10	11.19
PARCOR	21.68	31.64	21.82	4.41
CEP	23.63	36.73	30.21	10.98
FBPAR	29.69	44.87	36.72	3.48

complete set of phonemes. Moreover, the design of a phone recognition system based on results from experiments on reduced sets of phonemes may not be an optimal design.

The conclusion is that the system with type-I RBFN and with MSFB should be used for word recognition, and when phoneme recognition is of interest, RBFNs of type-II should be used.

4.6 Recognition Computation Requirements

In this section an estimate of the computation requirements for implementing the isolated-word recognition system developed in this chapter is provided. An estimate of the number of operations needed to recognize one word is calculated. To recognize a word, MFSB features are extracted for each frame, then phone recognition is performed using an RBFN type-I, and then DTW is employed to recognize the word.

To extract 16 MSFBs for one frame of 32ms the number of multiplications and additions is of the order:

$$6 \text{ channels} \times 160 \text{ taps} \times 20 \text{ samples} + 20 \text{ samples} \times 16 \text{ channel} = 51520 \text{ operations} \quad (4.3)$$

.

To perform phone recognition for one frame the number of operation needed is of the order:

$$33 \text{ nodes} \times \{(16+16+16) + \text{operations for calculating } \exp(.)\} \approx 2000 \text{ operations.} \quad (4.4)$$

To recognize a word using DTW, the distance between the input word and each of the reference patterns in the codebook is calculated using DTW. For 11 words vocabulary and a codebook that consists of 16 samples of the vocabulary, and with an average of 10 frames for each word, the codebook contains **about 1800 frames**. Each frame in the codebook is represented by the first 7 principle components of its center. DTW is performed as in [1], so if the input word consists of N frames, then the number of operations needed to recognize the word is of the order:

$$9 \times 1800 \times N = 16200N. \quad (4.5)$$

If **N=10**, then about 160,000 operations are needed to recognize the input word with DTW. If the time that is needed to perform one operation is **100ns**, then the time needed for DTW is about 16ms, which is less than 10% of the length of the input word. However, extracting features from a frame of 32ms requires about 6-7ms, which is about 20-30% of the length of an input frame. The time needed for phone recognition is less than 0.3ms, which is about 1% of an input frame.

The conclusion is that the system developed in this chapter can be implemented in real-time using a fast DSP processor such as TMS320C25 [152] that can perform one multiplication every 100ns. In this case recognition could be done within about 40% of the input word duration. However, this is an approximated estimate, and exact estimate can be obtained only after complete implementation of the system with the necessary hardware.

4.7 Summary and Conclusions

In this chapter the design of a cascade of speech recognition layers was presented. The speech signal is preprocessed and features are extracted, then phoneme recognition is performed using a RBFN, then words are identified using DTW. The system performance was tested on a small vocabulary. The TIMIT speaker-independent continuous speech database was used to design and test the ASR system.

In the preprocessing stage the speech signal was segmented into a stream of 32 ms frames. MSFBs, PARCORs, CEPs, and a parallel combination of the MSFBs and PARCORs (FBPARs) were extracted and used as inputs for the ASR system. It has been shown that the MSFBs and the FBPARs provide better performance than the PARCORs and CEPs feature sets. Hence the MSFB is the feature set that is used in the final design.

In the TIMIT corpus 60 phone labels are used and the speech is transcribed and labeled. An algorithm based on data clustering was developed to reduce the labeling and segmentation errors. To enhance the recognition rate, a reduced set of 33 phones was generated by merging similar phones in the original set.

Four RBFNs were designed and compared. An RBFN of type-I was a single layer net, and it provided the best word recognition rate. An RBFN of type-II was a two layer net where the output layer was connected to the hidden layer by a vector of weights. This structure yielded the highest phone recognition rate. An RBFN of type-III was the common two layer net, and resulted in greater or equal phone recognition than nets of type-I, but lower word recognition than the nets of type-I and type-II. An RBFN of type-IV provided poor recognition results for both phone and word recognition.

The first seven principal components of the phone centers were used as inputs for the word recognition layer. A vocabulary of 11 words (from sentence SA1 in TIMIT) and a vocabulary of 8 words (from sentence SA1 with the most confusing 3 words excluded) were used. Isolated-word recognition was performed using DTW. A code book was generated using 16 samples of the vocabulary from one male and one female from each of the 8 dialect regions in TIMIT. The best word recognition rate was achieved using an RBFN of type-I with MSFB features.

Several systems for speech recognition have been proposed in the recent year. However, different training and test data sets were used. As a result we are unable to perform a precise comparison of the performance of the system developed in this chapter and other systems. As an example, a phoneme recognition rate of 50% was reported in [153]. Only 3 male or 3 female speakers were involved in the test and training. and a different text from a different database was used. Additional example is the AT&T connected digit speaker-independent recognizer [150] with 11 words vocabulary has a recognition rate of 92-99%. However, the vocabulary is different form the TIMIT sentence, SA1. Moreover, the recognition was performed based on matching sequences of digits where higher recognition rates than matching single words can be achieved. The sentence SA1 from TIMIT is designed to study acoustic phonetic variabilities due to speakers and dialects, and it is phonetically diverse. This makes it difficult to achieve very high recognition rates. Hence the conclusion is that the performance of the system developed in this chapter is comparable with other systems, and it has a reduced size and computation requirements due to the reduced size of the phone set and the codebook.

Chapter 5: CONCLUSIONS

The design of a cascade of speech recognition layers was presented. The speech signal was preprocessed and features were extracted, then phoneme recognition was performed, then words were identified using dynamic time warping (DTW). The system performance was tested on a small vocabulary. The TIMIT speaker-independent continuous speech database was used to design and test the automatic speech recognition (ASR) system.

In Chapter 3 several feature sets were used for phoneme recognition. Multilayer perceptrons (MLPs) and radial basis function networks (RBFNs) were used to study the separability of phonemes. RBFNs were selected to be used with mel-scale filter bank features (MSFBs), partial correlation coefficients (PARCORs), parallel combination of MSFBs and PARCORs where MSFBs are used with voiced sounds and PARCORs are used with unvoiced sounds (FBPARs), and cepstral features (CEPs). FBPARs have offered the highest recognition rate on the reduced set of six phonemes, while other combinations of PARCORs and spectral features did not appear to add any significant increase of the recognition rate. Hence, FBPARs is the only combination of MSFBs and PARCORs that was considered in the full system design in Chapter 4.

In the preprocessing stage, the speech signal was segmented into a stream of 32 ms frames. MSFBs, PARCORs, CEPs, and FBPARs were extracted and used as inputs for the ASR system. It has been shown that the MSFB and the FBPAR have provide better performance than the PARCOR and CEP feature sets. Hence the MSFB is the feature set that is used in the final design.

A suboptimal algorithm for training RBFNs for phoneme recogniton was developed. The algorithm is iterative and interactive and combines supervised and unsupervised learning techniques. To reduce the system size and to enhance the recognition rate, a reduced set of 33 phones was generated by merging similar phones in the original 60 phone set.

In the TIMIT corpus 60 phone labels are used and the speech is transcribed and labeled. An algorithm based on data clustering was developed to reduce the labeling and segmentation errors.

Four RBFNs were designed and compared. RBFN of type-I was a single layer net and it provided the best word recognition rate. RBFN of type-II was a two layer net where the output layer was connected to the hidden layer by a vector of weights, and yielded the highest phone recognition rates. RBFN of type-III was the common two layer net, and resulted in greater or equal phone recognition than a net of type-I, but lower word recognition than the nets of type-I and type-II. An RBFN of type-IV provided poor recognition results for the phone and word recognition.

The first seven principal components of the phone centers were used as inputs for the word recognition layer. A vocabulary of 11 words (sentence SA1, "she had your dark suit in greasy wash water all year", in the TIMIT database) and a vocabulary of 8 words (sentence SA1 where the most confusing 3 words were excluded) used. Isolated-word recognition was performed using DTW. A codebook was generated using 16 samples of

the vocabulary from one male and one female from each of the 8 dialect regions in the TIMIT. The best word recognition rate was achieved using the RBFN of type-I with MSFB features.

To conclude, a small vocabulary, speaker-independent, isolated-word recognition system was developed for applications such as voice-driven menu systems. A word recognition rate of 84% on 11 words vocabulary for test data from 168 speakers. It is important to notice that the vocabulary words still have similarities. Moreover, the sentence SA1 is read by the speakers in a continuous manner. This adds more variability to the data over a vocabulary that is spoken slowly and in isolated-word form. A word recognition rate of 95% was achieved when the most 3 confusing words in the 11 words vocabulary were excluded. Hence, if the vocabulary words are selected to be distinct from each other, and if the words are isolated and spoken clearly, the system will be suitable for practical applications such as a speaker-independent voice-driven menu system and a speaker-independent voice command system. The system can be implemented in real-time with a fast processor that can perform a multiplication in 100ns.

REFERENCES

[1] F. J. Owens, Signal Processing of Speech, McGraw-Hill, Inc., New York, 1993.

[2] W. A. Ainsworth, Speech Recognition By Machine, Peter Peregrinus Ltd., UK, 1988.

[3] IEEE Publications INDEX, 1979-1993.

[4] S. E. Levinson and D. B. Roe, "A perspective on speech recognition," IEEE Communications Magazine, pp. 28-34, January 1990.

[5] J. Mariani, "Recent advances in speech recognition," IEEE ICASSP'89, pp. 429-440, 1990.

[6] X. Yang, K. Wang and S. A. Shamma, "Auditory representations of acoustic signals," IEEE Transactions on Information theory, Vol. 38, No. 2, pp. 824-839, March 1992.

[7] G. Bristow, Electronic Speech Recognition, Collins Professional Books, William Collins Sons Co. Ltd., London, 1986.

[8] I. H. Witten, Principles of Computer Speech, Academic Press Inc., London, 1982.

[9] O. Rioul and M. Vetterli, "Wavelets and signal processing," IEEE Signal Processing Magazine, pp. 14-38, October 1991.

[10] M. Vetterli and C. Herley, "Wavelets and filter banks: theory and design," IEEE Transactions on Signal Processing, Vol. 40, No. 9, pp. 2207-2232, September 1992.

[11] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," IEEE Transactions on Information Theory, Vol. 38, No. 2, pp. 569-586, March 1992.

[12] A. K. Soman, P. P. Vaidyanathan and T. Q. Nguyen,"Linear phase orthonormal filter banks," IEEE ICASSP'93, Vol. 3, pp. III-209-III-212, 1993.

[13] J. Mau, "Perfect reconstruction modulated filter banks," IEEE ICASSP'93, Vol. 3, pp. III-225-III-228, 1993.

[14] Y. Qi and B. R. Hunt, "Voiced-unvoiced-silence classification of speech using hybrid features and a network classifier," IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 2, pp. 250-255, April 1993.

[15] B. Atal and L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-24, pp. 201-212, June 1976.

[16] R. P. Lippmann, "Pattern classification using neural networks," IEEE Communications Magazine, pp. 47-64, November 1989.

[17] H. F. Silverman and D. P. Morgan, "The application of dynamic programming to connected speech recognition," IEEE ASSP Magazine, pp. 6-25, July 1990.

[18] F. Itakura, "Minimum prediction residual principle applied to speech recognition," IEEE Transactions on ASSP, Vol. ASSP-23, No. 1, pp. 67-72, Feb. 1975.

[19] H. Sakoe and S. Chiba, "Dynamic programming algorithms optimization for spoken word recognition," IEEE Transactions on ASSP, Vol. ASSP-26, No. 1, pp. 43-49, Feb. 1978.

[20] C. S. Myers and L. R. Rabiner, "A levelbuilding dynamic time warping algorithm for connected word recognition," IEEE Transactions on ASSP, Vol. ASSP-29, No. 2, pp. 284-296, April 1981.

[21] C. S. Myers and L. R. Rabiner, "Connected digit recognition using a level-building DTW algorithm," IEEE Transactions on ASSP, Vol. ASSP-29, No. 3, pp. 351-363, June 1981.

[22] S. Bridle, R. M. Chamberlain and M. D. Brown, "An algorithm for connected word recognition," IEEE ICASSP'82, pp. 899-902, 1982.

[23] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Transactions on Information Theory, Vol. IT-13, No. 2, pp. 260-269, April 1987.

[24] T. Takara, "Isolated word recognition using continuous state transition probability and DP matching," IEEE ICASSP'89, pp. 274-277, 1989.

[25] J. Schroeter and M. M. Sondhi, "Dynamic programming search of articulatory codebook," IEEE ICASSP'89, pp. 588-591, 1989.

[26] J. Picone, "Continuous speech recognition using hidden Markov models," IEEE ASSP magazine, pp. 26-40, July 1990.

[27] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," IEEE ASSP Magazine, pp. 4-16, Jan. 1986.

[28] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of The IEEE, Vol. 77, No. 2, pp. 257-289, Feb. 1989.

[29] L. R. Bahl, P. F. Brown, P. V. DeSouza and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," IEEE ICASSP'86, pp. 49-52, 1986.

[30] C. H. Lee and L. R. Rabiner, "A frame synchronous network search algorithm for connected word recognition," IEEE Transactions on ASSP, Vol. 37, No. 11, pp. 1649-1658, November 1989.

[31] H. Ney, D. Mergel, A. Noll and A. Paeseler, "A data-drived organization of the dynamic programming beam search for continuous speech recognition," IEEE ICASSP'87, pp. 833-836, 1987.

[32] J. G. Wilpon, L. R. Rabiner, C. Lee and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models, " IEEE Transactions on ASSP, Vol. 38, No. 11, pp. 1870-1878, November 1990.

[33] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition, " IEEE Transactions on ASSP, Vol. 38, No. 12, pp. 2033-2045, December 1990.

[34] X. Huang and K. Lee, "On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition," IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 2, pp. 150-157, April 1993.

[35] Y. Zhao, "A speaker-independent continuous speech recognition system using continuous mixture Gaussian density HMM of phoneme-sized units," IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 3, pp. 345-361, July 1993.

[36] R. P. Lippmann, "An introduction to computing with neural nets," IEEE ASSP Magazine, pp. 4-22, April 1987.

[37] D. Hush and B. G. Horne, "Progress in supervised neural networks," IEEE Signal Processing Magazine, pp. 8-39, January 1993.

[38] S. Yoshimoto, "A study on artificial neural network generalization capability," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. 3, pp. 689-694, 1990.

[39] H. Drucker and Y. L. Cun, "Improving generalization performance using double backpropagation," IEEE Transactions on Neural networks, Vol. 3, No. 6, pp. 991-997, November 1992.

[40] Q. Zhang and A. Benveniste, "Wavelet networks," IEEE Transactions on Neural networks, Vol. 3, No. 6, pp. 889-898, November 1992.

[41] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang, "Phoneme recognition using time-delay neural networks," IEEE Transactions on ASSP, Vol. 37, No. 3, pp. 328-339, March 1989.

[42] J. B. Hampshire and A. H. Waibel, "A novel objective function for improved phoneme recognition using time-delay neural networks, " IEEE Transactions on Neural networks, Vol. 1, No. 2, pp. 216-228, June 1990.

[43] S. Kong and B. Kosko, "Differential competitive learning for centroid estimation and phoneme recognition," IEEE Transactions on Neural networks, Vol. 2, No. 1, pp. 118-124, January 1991.

[44] Y. Bengio, R. De Mori, G. Flammia and R. Kompe, "Global optimization of a neural network-hidden Markov model hybrid," IEEE Transactions on Neural networks, Vol. 3, No. 2, pp. 252-259, March 1992.

[45] R. L. Watrous, "Speaker normalization and adaptation using second-order connectionist networks," IEEE Transactions on Neural networks, Vol. 4, No. 1, pp. 21-30, January 1993.

[46] W. Liu, A. G. Andreou and M. H. Goldstein, "Voiced-speech representation by an analog silicon model of the auditory periphery," IEEE Transactions on Neural networks, Vol. 3, No. 3, pp. 477-487, May 1992.

[47] W. Y. Huang and R. P. Lippmann, "Comparison between neural net and conventional classifiers," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 485-493, 1987.

[48] R. P. Lippmann and Ben Gold, "Neural-net classifiers useful for speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 417-425, 1987.

[49] R. L. Watrous and L. Shastri, "Learning phonetic features using connectionist networks: an experiment in speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 381-388, 1987.

[50] M. Watson, "A neural network model for phoneme recognition using the generalized delta rule for connection strength modification," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 389-396, 1987.

[51] S. A. Shamma, "Neural networks for speech processing and recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 397-405, 1987.

[52] H. Bourland and C. J. Wellekens, "Multilayer perceptrons and automatic speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 407-416, 1987.

[53] B. Gold, R. P. Lippmann and M. L. Malpass, "Some neural net recognition results on isolated words," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 427-434, 1987.

[54] C. H. Rogers and W. J. B. Oldham, "Isolated word recognition with an artificial neural network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 435-442, 1987.

[55] M. A. Cohen, S. Grossberg and D. Stork, "Recent developments in a neural model of real-time speech analysis and synthesis," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 443-453, 1987.

[56] D. W. Tank and J. J. Hopfield, "Concentrating information in time: analog neural networks with applications to speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 455-468, 1987.

[57] R. T. Savely and L. B. Johnson, "The implementation of neural network technology," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 477-484, 1987.

[58] E. J. Smythe, "Detection of formant transitions by a connectionist network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 495-502, 1987.

[59] J. P. Lewis, "Creation by refinement: a creativity paradigm for gradient decent learning networks," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 229-233, 1988.

[60] S. Kurogi, "Abilities and limitations of a neural network model for spoken word recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 205-214, 1988.

[61] S. Nolfi and D. Parisi, "Learning to understand sentences in a connectionist network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 215-219, 1988.

[62] J. Y. Murdock, A. A. Husseiny and S. A. Abolrous, "Improvement on speech recognition and synthesis for disabled individuals using fuzzy neural net retrofits," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 252-258, 1988.

[63] T. Matsuoka, H. Hamada and R. Nakatsu, "Syllable recognition using integrated neural networks," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 251-258, 1989.

[64] H. Sawal, A. Waibel, P. Haffner, M. Miyatake and K. Shikano, "Parallelism, hierarchy, scaling in time-delay neural networks for spotting Japanese phonemes CV-syllables," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 81-88, 1989.

[65] B. R. Kämmerer and W. A. Küpper, "Design of hierarchical perceptron structures and their application to the task of isolated-word recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 243-249, 1989.

[66] M. A. Franzini, M. J. Witbrock and K. Lee, "Speaker-independent recognition of connected utterances using recurrent and non-recurrent neural networks," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 1-6, 1989.

[67] J. R. Hampshire II and A. H. Waibel, "A novel objective function for improved phoneme recognition using time-delay neural network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 235-241, 1989.

[68] M. D. Tom and M. F. Tenorio, "A spatio-temporal pattern recognition approach to word recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 351-355, 1989.

[69] A. Amano, T. Aritsuka, N. Hataoka and A. Ichikawa, "On the use of neural networks and fuzzy logic in speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 301-305, 1989.

[70] C. A. Kamm and S. Singhal, "Effect of neural network input span on phoneme classification," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 195-200, 1990.

[71] S. Danielsen, "Recognition of Danish phonemes using an artificial neural network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. III, pp. 677-682, 1990.

[72] A. Hirai and A. Waibel, "Phoneme-based word recognition by neural network - a step toward large vocabulary recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. III, pp. 671-676, 1990.

[73] T. Kitamura, W. Hui, A. Iwata and N. Suzumura, "Speaker-dependent 1000 word recognition using a large scale neural network "CombNET-II" and dynamic spectral features," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 1202-1207,Singapore, 1991.

[74] H. Hackbarth and J. Mantel, "Modular connectionist structure for 100-word recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 845-849, 1991.

[75] L. Y. Pratt and C. A. Kamm, "Improving a phoneme classification neural network through problem decomposition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 821-826, 1991.

[76] G. Thierer, A. Krause and H. Harckbarth, "Training speed-up methods for neural networks applied to word recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 865-869, 1991.

[77] P. Escande, D. Bè roule and P. Blanchet, "Speech recognition experiments with guided propagation," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 765-768,Singapore, 1991.

[78] C. Iooss, "From lattices of phonemes to sentences: a recurrent neural network approach," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 833-838, 1991.

[79] Y. Bengio, R. D. Mori, G. Flammia and R. Kompe,"Global optimization of a neural network - hidden Markov model hybrid," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 789-794, 1991.

[80] R. D. T. Janssen, M. Fanty and R. A. Cole, "Speaker-idependent phonetic classification in continuous English letters," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 801-808, 1991.

[81] X. Driancourt, L. Bottou and P. Gallinari, "Learning vector quantization, multi-layer perceptron and dynamic programming: comparison and cooperation," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 815-819, 1991.

[82] M. J. Palakal and M. J. Zoran, "Speaker-invarient phoneme recognition using multiple neural network models," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 839-844, 1991.

[83] T. Ghise-Crippa and A. El-Jaroudi, "Voiced-unvoiced-silence classification of speech using neural nets," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 851-856, 1991.

[84] F. S. Gurgen, K. Aikawa and K. Shikano, "Phoneme recognition with neural networks using a novel fuzzy training algorithm," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 572-577,Singapore, 1991.

[85] K. Aikwa, "Time-warping neural network for phoneme recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. III, pp. 2122-2127,Singapore, 1991.

[86] A. Sankar and R. J. Mammone, "Speaker independent vowel recognition using neural tree networks," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. II, pp. 809-814, 1991.

[87] C. Yongsheng, Y. Baozong and L. BiQing, "Real-time Chinese syllable recognition system with hierarchically structured neural network and transputer system," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 743-748, 1992.

[88] F. E. Shaudys and T. K. Leen, "Feature selection for improved classification," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 697-702, 1992.

[89] Y. Liu, Y. Lee, H. Chen and G. Sun, "Discriminative training algorithm for predictive neural network models," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 685-690, 1992.

[90] F. S. Gurgen, K. Aikawa and K. Shikana, "On the training strategies of neural networks for speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 749-754, 1992.

[91] H. Kinugasa, H. Kamata and Y. Ishida, "Recognition of Japanese words by neural networks using vocal tract area," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 637-642, 1992.

[92] J. Jianxin, H. Zheng and L. Feng, "A hybrid neural-fuzzy-neural framework for speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 643-648, 1992.

[93] S. A. Zahorian, S. Kelkar and D. Livingston, "Formant estimation from cepstral coefficients using a feedforward memoryless neural network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 673-678, 1992.

[94] X. Driancourt and P. Gallinari, "An empirical risk optimizer for speech recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 703-708, 1992.

[95] F. A. Unal and N. Tepedelenlioglu, "Dynamic time warping using an artificial neural network," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 715-721, 1992.

[96] S. C. Sivakumar, W. Robertson and K. Macleod, "Improving temporal representation in TDNN structure for phoneme recognition," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 728-733, 1992.

[97] H. Finster, "Automatic speech segmentation using neural network and phonetic transcription," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 734-736, 1992.

[98] N. M. Botros and S. Prennath, "Speech recognition using dynamic neural networks," in Proceedings of IEEE Joint International Conference on Neural Networks, Vol. IV, pp. 737-742, 1992.

[99] D. J. Burr, "Comparison of Gaussian and neural network classifiers on vowel recognition using the discrete cosine transform," IEEE ICASSP'92, Vol. II, pp. 365-368, 1992.

[100] M. A. Franzini, "A new connectionist architecture for word spotting," IEEE ICASSP'92, Vol. II, pp. 361-364, 1992.

[101] T. M. English and L. C. Boggess, "Back-propagation training of a neural network for word spotting," IEEE ICASSP'92, Vol. II, pp. 357-360, 1992.

[102] H. B. D. Sorensen and U. Hartmann, "Self-structuring hidden control neural network model for speech recognition," IEEE ICASSP'92, Vol. II, pp. 353-356, 1992.

[103] J. Kangas, K. Torkkola and M. Kokkonen, "Using soms as feature extractors for speech recognition," IEEE ICASSP'92, Vol. II, pp. 341-344, 1992.

[104] H. Bourland, N. Morgan, C. Wooters and S. Renals, "CDNN : a context dependent neural network for continuous speech recognition," IEEE ICASSP'92, Vol. II, pp. 349-352, 1992.

[105] M. G. Rahim, "A neural tree network for phoneme classification with experiments on the TIMIT database," IEEE ICASSP'92, Vol. II, pp. 345-348, 1992.

[106] Z. Wang and J. V. Hanson, "Code-excited neural vector quantizer," IEEE ICASSP'93, Vol. I, pp. 497-500, 1993.

[107] A. Basu and T. Svendsen, "A time-frequency segmental neural network for phoneme recognition," IEEE ICASSP'93, Vol. I, pp. 509-512, 1993.

[108] G. Zavaliagkost, R. Schwartz and J. Makhoul, "Elliptical basis functions for segment modeling," IEEE ICASSP'93, Vol. I, pp. 513-516, 1993.

[109] M. P. DeSimio and T. R. Anderson, "Phoneme recognition with binaural cochlear models and the stereausis representation," IEEE ICASSP'93, Vol. I, pp. 521-524, 1993.

[110] J. E. Diaz-Verdejo, J. C. Segura-Luna, A. J. Rubio-Ayuso, A. M. Peinada-Herreros and J. L. Pérez-Córdoba, "A new neuron model for an alphane-semicontinuous HHM," IEEE ICASSP'93, Vol. I, pp. 529-532, 1993.

[111] A. Mellouk and P. Gallinari, "A discriminative neural predition system for speech recognition," IEEE ICASSP'93, Vol. I, pp. 533-536, 1993.

[112] Y. Konig and N. Morgan, "Supervised and unsupervised clustering of the speaker space for connectionist speech recognition," IEEE ICASSP'93, Vol. I, pp. 545-548, 1993.

[113] M. Fanty, P. Schmid and R. Cole, "City name recognition over the telephone," IEEE ICASSP'93, Vol. I, pp. 549-552, 1993.

[114] Y. Kato and M. Sugiyama, "Speaker-independent features extracted by a neural network," IEEE ICASSP'93, Vol. I, pp. 553-556, 1993.

[115] C. Bregler, H. Hild, S. Manke and A. Wailbel, "Improving connected letter recognition by lipreading," IEEE ICASSP'93, Vol. I, pp. 557-560, 1993.

[116] P. Le Cerf and D. Van Compernolle, "Using parallel MLPs as labelers for multiple codebook HMMs," IEEE ICASSP'93, Vol. I, pp. 561-564, 1993.

[117] R. P. Lippmann and E. Singer, "Hybrid neural-network/HMM approaches to wordspotting," IEEE ICASSP'93, Vol. I, pp. 565-568, 1993.

[118] B. Petek and A. Ferligoj, "Exploiting prediction error in a predictive-based connectionist speech recognition system," IEEE ICASSP'93, Vol. II, pp. 267-270, 1993.

[119] Y. Anglade, D. Fohr and J. Junqua, "Speech discrimination in adverse conditions using acoustic knowledge and selectivity trained neural networks," IEEE ICASSP'93, Vol. II, pp. 279-282, 1993.

[120] T. R. Anderson, "Phoneme recognition using an auditory model and a recurrent self-organizing neural network," IEEE ICASSP'93, Vol. II, pp. 337-340, 1993.

[121] M. G. Rahim, "A self-learning neural tree network for recognition of speech features," IEEE ICASSP'93, Vol. I, pp. 517-520, 1993.

[122] T. Komori and S. Katagiri, "An optimal learning method for minimizing spotting errors," IEEE ICASSP'93, Vol. II, pp. 271-274, 1993.

[123] B. de Vries, L. Dias and J. Pearson, "Learning with target trajectory constraints for sequence classification," IEEE ICASSP'93, Vol. I, pp. 525-528, 1993.

[124] G. Jim and L. Ho Chung, "A multilayer perceptron postprocessor to hidden Markov modeling for speech recognition," IEEE ICASSP'93, Vol. II, pp. 263-266, 1993.

[125] H. Hild and A. Waibel, "Multi-speaker/speaker-independent architectures for the multi-state time delay neural network," IEEE ICASSP'93, Vol. II, pp. 255-258, 1993.

[126] J. Tebelskis, "Performance through consistency: connectionist large vocabulary continuous speech recognition," IEEE ICASSP'93, Vol. II, pp. 259-262, 1993.

[127] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CD-ROM," NISTIR 4930, February 1993.

[128] M. J. Baker, D. S. Pallett and J. S. Bridle, "Speech recognition performance assessment and available databases," IEEE ICASSP'83, pp. 527-530, 1983.

[129] MATLAB, The Math Works, Inc., 1986-92.

[130] S. M. Kay, Modern Spectral Estimation: Theory and Applications, Prentice Hall, Englewood Clifs, New Jersey, 1988.

[131] D. O'Shaughnessy, Speech Communication: Human and Machine, Addison Wesley Publishing Company, 1987.

[132] J. P. Olive, A. Greenwood, J. Coleman, Acoustics of American English Speech: A Dynamic Approach, Springer-Verlag, New York, 1993.

[133] R. A. Cole, L. Hirschman, et al., Workshop on Spoken Language Understanding," Center for Spoken Language Understanding, Oregon Graduate Institute of Science and Engineering, *Technical report* No. CS/E 92-014, Sep. 1992.

[134] P. Lockwood, J. Boundy and M. Blanchet, "Non-linear spectral subtraction (NSS) and hidden Markov models for robust speech recognition in car noise environment," IEEE ICASSP'92, pp. I-265-268, 1992.

[135] S. Dobler, P. Meyer and H. W. Ruehi, "A robust connected-words recognizer," IEEE ICASSP'92, pp. I-245-248, 1992.

[136] B. Mak, J. Junqua and B. Reaves, "A robust speech/non-speech detection algorithm using time and frequency-based features," IEEE ICASSP'92, pp. I-269-272, 1992.

[137] D. C. Bateman, D. K. Bye and M. J. Hunt, "Spectral contrast normalization and other techniques for speech recognition in noise," IEEE ICASSP'92, pp. I-241-244, 1992.

[138] J. B. Allen, "How do humans process and recognize speech ?," IEEE Trans. on Speech and Audio Processing, vol. 2, no. 4, pp. 567-577, 1994.

[139] R. P. Lippmann, E. I. Chang, and C. R. Jankowski, "Wordspotter training using figure-of-merit backpropagation," IEEE ICASSP'94, Vol. I, pp. 389-392, 1994.

[140] E. Singer, and R. P. Lippmann, "A speech recognizer using radial basis function neural networks in an HMM framework," IEEE ICASSP'92, Vol. II, pp. 629-632, 1992.

[141] S. Renals, and R. Rohwer, "Phoneme classification experiments using radial basis functions," in Proc. of IEEE Joint International Conference on Neural Networks, Vol. I, pp. 461-467, 1989.

[142] J. N. Holmes, "The JSRU channel vocoder," Proc. IEE, 127(F1), pp. 53-60, February, 1980.

[143] F. Itakura, "Minimum prediction residual principle applied to speech recognition," IEEE Transactions on ASSP, vol. 23, no. 2, pp. 67-72, 1975.

[144] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications, vol. 28, no. 1, pp. 84-95, 1980.

[145] T. Kohonen, "An introduction to neural computing," Neural Networks, vol. 1, pp. 3-16, 1988.

[146] G. A. Carpenter, and S. Grossberg, "ART 2: self organization of stable category recognition codes for analog input patterns," Applied Optics, vol. 26, 4.919-4.930, 1987

[147] K. Fukunaga, Introduction to Statistical Pattern Recognition, NY, Academic Press, 1972.

[148] A. Averbuch, L. Bahl, et al., "An IBM-PC based large-vocabulary isolated-utterance speech recognizer," IEEE ICASSP'86, pp. 53-56, 1986.

[149] K. F. Lee, Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System, Ph. D. Dissertation, Computer Science Department, Carnegie Mellon University, 1988.

[150] L. R. Rabiner, J. G. Wilpon, and F. K. Soong, "High performance digit recognition using hidden Markov models," IEEE Transactions on ASSP, vol. 37, no. 8, pp. 1214-1225, 1989.

[151] J. Deller, J. G. Proakis, and J. H. L. Hansen, Discrete-Time Processing of Speech Signals, pp. 380-384, Macmillan Publishing Company, New York, 1993.

[152] TMS320C2x User's Guide, Texas Instruments Inc., 1995.

[153] S. Furui, and M. M. Sondhi, Advances in Speech Signal Processing, pp. 419-452, Marcel Dekker, Inc., New York, 1992.

VITA

Fakhralden A. Huliehel was born on June 14th, 1964 in Safad, Israel. He received the B. Sc. and M. Sc. degrees in Electrical and Computer Engineering from the Ben-Gurion University of the Negev, Israel, in 1986 and 1990, respectively. He has been in the graduate program in the Department of Electrical Engineering, at Virginia Polytechnic Institute and State University, Blacksburg, Virginia since September, 1990.

From 1987 to 1989 he was a Teaching Assistant in the Department of Electrical and Computer Engineering of the Ben-Gurion University, Israel. From 1986 to 1989 he worked as a part-time teacher in the College of Technology, Sdaroat, Israel. From 1989 to 1990 he worked as a part-time teacher in the College of Technology Beer-Sheva, Israel. He was a Research/Teaching Assistant in the Department of Electrical Engineering, Virginia Polytechnic Institute and State University since September, 1990.

F. Huliehel