THESIS REPORT

Master's Degree

SYSTEMS
RESEARCH
CENTER

*Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
the University of Maryland,
Harvard University,
and Industry*

Neural Networks That Recognize Phonemes by Their Acoustic Features

by K. Wang
Advisor: S.A. Shamma

M.S. 91-1

ABSTRACT

Title of Thesis:	Neural Networks That Recognize Phonemes by Their Acoustic Features
Name of Degree Candidate:	Kuansan Wang
Degree and Year:	Master of Science, 1989
Thesis directed by:	Dr. S. A. Shamma Associate Professor Department of Electrical Engineering

The ability of the ear model and lateral inhibitory networks(LIN) to preserve and enhance acoustic features of speech signal is examined by training neural networks to recognize phonemes by their LIN outputs. Using the back propagation learning algorithm, networks that are specialized to recognize specific classes of phonemes are trained and tested. Experiments are conducted both in single and multi-speaker cases. By using single layer networks, we can show that the phonemes are identified by their acoustic features that have been known to linguists and phoneticians. The networks generally yield satisfying results when tested in experiments for a single speaker, where we focus on the performance against phoneme variation induced by the context, and in multi-speaker experiments where errors in recognition are due to speaker variation. These results convince us that the acoustic features picked by the networks are reliable cues for phoneme recognition.

Neural Networks That Recognize Phonemes by Their Acoustic Features

by

Kuansan Wang

Thesis submitted to the Faculty of The Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
December 1989

Advisory Committee:

Dr. S. A. Shamma	Chairman/Advisor
Dr. R. W. Newcomb	
Dr. N. Farvardin	

Contents

1 Introduction 1

1.1 An Overview 1

1.2 The Speech Processing Model 4

2 The Supervised Learning Algorithm 11

2.1 Introduction 11

2.2 Simple Gradient Descent Algorithm 12

2.3 Problems and Modifications 15

2.3.1 Convergence of the Algorithm 15

2.3.2 Local Minimum Problem and Speed of Convergence . . 19

2.3.3 Capability and Redundancy Effect 21

2.3.4 Performance of the Algorithm 22

3 Phoneme Recognition for Single Speaker 23

3.1 Introduction 23

3.2 Performance Measurements 26

3.3 Nasal Consonants 31

3.4 Stop Consonants 34

3.5 Fricative Consonants 41

3.6 Vowels and Semivowels 50

3.7 Summary and Remarks 56

4 Multi-speaker Phoneme Recognition 63

4.1 Introduction 63

4.2 Stop Consonants 65

4.3 Fricative Consonants 71

4.4 Nasals, Vowels and /l/ 77

4.5 Summary and Remarks 86

List of Figures

1.1 Output of the ear model for 800Hz monotone 5

1.2 A vertical cross-section of the above figure 5

1.3 The envelop of 800Hz signal 6

1.4 The LIN II output of 800Hz monotone signal 8

1.5 The ear model output of the mixture of a 400 and an 800 Hz monotone 9

1.6 The corresponding LIN II outputs of the signal with harmonics of 400 and 800 Hz 9

2.1 A two-layer, feed-forward network 12

2.2 (a,b) ideal gradient descent (c) run-away case 17

2.3 A simple two-layer network that can solve xor problem 18

2.4 The surface that solves generalized XOR problem 19

3.1 The LIN II outputs of the word 'snow' 28

3.2 The LIN II output of the word 'amnesia' 32

3.3 Time-average LIN II outputs for /m/ and /n/ 33

3.4 The weighted inputs for nasal /m/ and /n/ 33

3.5 Weights for discriminating nasals 34

3.6 LIN II output of 'maintain' spoken by a male speaker 36

3.7 Time-average LIN II outputs for the bursts of /t/,/k/ and /p/ 36

3.8 Time-average LIN II outputs for the bursts of /d/,/g/ and /b/ 37

3.9 Weighted inputs of unvoiced stops 38

3.10 Weighted inputs of voiced stops 38

3.11 Weights for discriminating unvoiced stops /t,k,p/ respectively 39

3.12 Weights for discriminating voiced stops /d,g,b/ respectively . . 40

3.13 Time-average LIN II outputs of /s/,/f/,/th/,/sh/ 42

3.14 Time-average LIN II outputs of /z/,/v/,/dh/,/zh/ 43

3.15 Weighted inputs of unvoiced fricatives /s,f,th,sh/ 45

3.16 Weighted inputs of voiced fricatives /z,v,dh,zh/ 46

3.17 Weights for discriminating unvoiced fricatives 47

3.18 Weights for discriminating voiced fricatives 48

3.19 Time-average patterns for vowels 52

3.20 Weighted LIN II patterns of vowels 54

3.21 The LIN II output of the word 'lemon' 58

3.22 A speech recognition system based on our single speaker experiments 59

3.23 The recognition of the word 'lemon' 60

3.24 The recognition of the word 'lemon'(Cont'd) 61

3.25 The recognition of the word 'lemon'(Cont'd) 62

4.1 The time-average patterns of /t/,/k/,/p/ of four male speakers 66

4.2 The time-average patterns of /d/,/g/,/b/ of four male speakers 66

4.3 The overall-average patterns of /t/,/k/,/p/ of four male speakers 67
4.4 The overall-average patterns of /d/,/g/,/b/ of four male speakers 67
4.5 The weighted inputs of unvoiced stops from the net trained on the overall-average patterns 68
4.6 The average of four speakers' weighted inputs for unvoiced stops 68
4.7 The weighted inputs of voiced stops from the net trained on the overall-average patterns 69
4.8 The average of four speakers' weighted inputs for voiced stops 69
4.9 The time-average patterns for /s,f,th,sh/ from four speakers . 72
4.10 The time-average patterns for /z,v,dh,zh/ from four speakers . 72
4.11 The overall-average patterns for /s,f,th,sh/ 73
4.12 The overall-average patterns for /z,v,dh,zh/ 73
4.13 The weighted inputs of unvoiced fricatives 74
4.14 The weighted inputs of voiced fricatives 74
4.15 The average of four speakers' weighted inputs of /s,f,th,sh/ . . 76
4.16 The average of four speakers' weighted inputs of /z/,/v/,/dh/ 76
4.17 Time-average patterns for vowels 78
4.18 Overall-average patterns for vowels 80
4.19 Weighted inputs of the vowels 81
4.20 The weights of multi-speaker vowels recognition network . . . 83
4.21 The average of four speakers' weighted inputs 84

List of Tables

1.1 Channel numbers with their resonant frequencies 7

3.1 The sentences from which our phonemes are extracted 26

3.2 Phonemes used in the single speaker experiments 27

3.3 Nasal consonants in English 31

3.4 Stop consonants in English 35

3.5 The confusion matrix of stop consonants recognition 40

3.6 The deferred decision performance of stop networks 41

3.7 Fricative consonants in English 44

3.8 The confusion matrix of fricatives 49

3.9 The deferred decision performance of fricatives 49

3.10 Vowels labeled in our database 51

3.11 Confusion matrix for vowels 55

3.12 The deferred decision performance of vowels network 57

4.1 Phonemes occurred in the four speakers' speech data 64

4.2 The confusion matrix on recognizing short-time-average stop patterns 70

4.3 The deferred decision performance on stops from four speakers 71

4.4 The confusion matrix of multi-speaker fricatives networks . . . 75

4.5 The deferred decision performance of multi-speaker fricatives networks 77

4.6 The confusion matrix for multi-speaker vowels recognition network 85

4.7 The deferred decision performance of multi-speaker vowels recognition network 86

Chapter 1

Introduction

1.1 An Overview

Speech is the major method for human beings to communicate with each other. Man is the only creature that has developed a sophisticated mechanism to transmit information by his voice. In spite of our great achievements in science and technology, we are still not able to build a machine that can fully emulate our speech processing system. For years, people have been trying to learn from Mother Nature how such a complicated system is developed and functioning. As a result, we have already had the vocal tract model accounting for the speech generating process[14,15].

Up to the present, many ear models have been successfully created based on biophysical experimental data[1,5,15]. We have been able to reproduce the activity patterns of the sensory nerves coming out of the cochlea through computer simulation. But how these signals are further processed in the central auditory system so that the encoded information can be recognized and understood is still unknown because the experimental data are difficult to obtain at this level of processing. People have been trying to build artificial neural networks to simulate what is really going on in the brain.

During the past decade, artificial neural networks have been reported to attain great success in applications of various fields besides signal processing. Among these examples, neural networks are proclaimed to achieve an outstanding performance especially in the field of pattern recognition. It is widely believed that employing massive parallelism like neural networks is the only way to tackle the traditional bottleneck induced by sequential processing. In addition to parallelism, people also believe that the potential benefit of neural network models is their ability in learning or adaptation[16]. Most neural network models are not parametric and make weak assumption of the underlying distribution of the input data. It is particularly useful when the input data are generated by nonlinear or non-Gaussian processes. The learning algorithms then become the key point for neural network models to work properly and excellently.

Several network topologies together with their learning algorithms have been proposed. The multi-layer feed-forward network with back propagation learning algorithm is undoubtably the most famous one that has been widely studied and used[6]. It basically implements a gradient descent algorithm to find a minimum of a given function which in our application is a predefined error function that characterizes the discrepancy between the real outputs and the desired outputs. With this algorithm, we are able to not only understand how sets of different patterns are properly classified but also see the generalization ability of the network after learning. We have taken advantage of this ability to achieve a rather high recognition rate in our experiments of single speaker phoneme recognition. Chapter 2 will describe the basic idea of the learning algorithm.

Every language is constructed by a set of basic linguistic units which have the property that if one replaces another in an utterance, the meaning of the word or sentence is changed. In other words, the information transmitted through speech can be represented by a concatenation of elements from a finite set. Such elements are called phonemes. To be identified correctly, each phoneme must have its own unique features. It has been suggested that the speech processing system described in the next section is good at preserving and even enhancing the features of the phonemes. To confirm this hypothesis and to discover the nature of the acoustic features, we have built and trained several artificial neural networks to identify the phonemes from several speakers in our speech database. The results illustrate that all the networks choose as a key to discriminate the patterns exactly the acoustic features which have long been known to linguistists and phoneticians. By using these features of phonemes, the networks are able to achieve a high recognition rate on the untrained patterns. We will describe our experiments on a single speaker in detail in the third chapter.

In the fourth chapter, we generalize the experiments from single speaker to four speakers. In spite of the significant variation among the speakers, the learning process is again able to detect the common features of phonemes with which our networks still yield a satisfying performance. By inspecting the resultant weights after learning, we can get a clear picture of the important features in the phonemes as well as how the learning process can possibly figure them out.

After these experiments, we are convinced that the lateral inhibitory networks(LINs)[4] suggested to process the cochlear model outputs do preserve and enhance the speech features in some sense. The basic ideas of the ear model and LINs used in our experiments are described in the next section.

1.2 The Speech Processing Model

Through the analysis of speech signals along the vocal tract, we can explain many aspects of the speech generating process. In an analogous manner, we may follow the signals up through the auditory system to find out how different acoustic features are extracted and recognized at different levels. After years of study, we have been able to create a model that emulates the performance of the ears of human beings and other mammals [1,3,5]. The input to the model is an ordinary speech waveform and the outputs are the average firing rates for bunches of auditory nerve fibers, each of which has different activity patterns for stimuli of different frequencies. Fibers in each bunch are taken from the vicinity of the same spot of the basilar membrane and should therefore have similar frequency responses. We shall call each bunch of fibers a 'channel' in the following text.

Figure 1.1 shows the outputs of the ear model when the input is an 800Hz monotone signal. The figure is arranged in a way that channels near the bottom of the figure are sensitive to high frequency signals and those on the

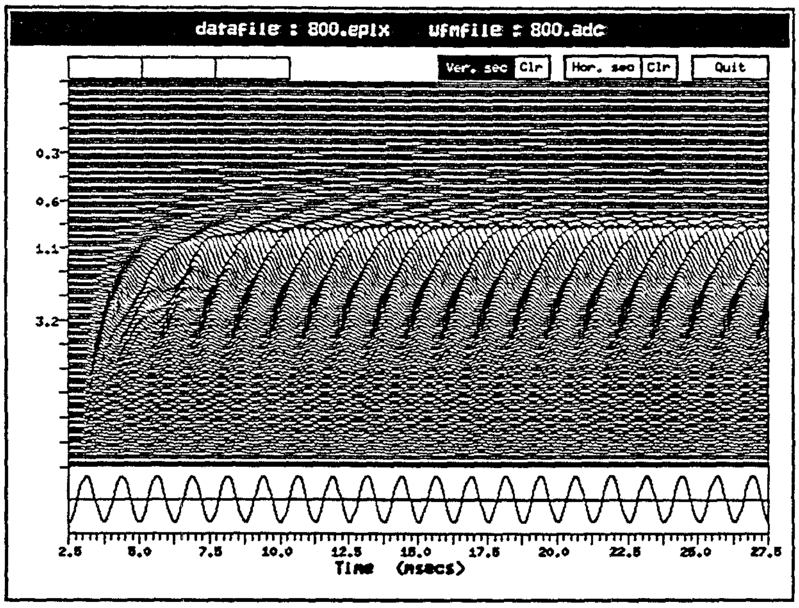

Figure 1.1: Output of the ear model for 800Hz monotone

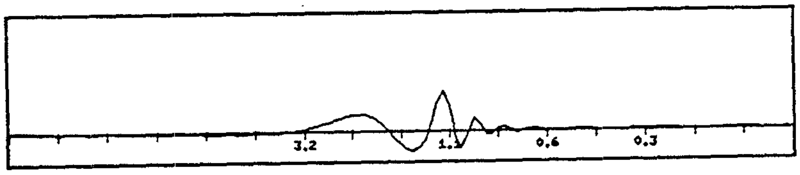

Figure 1.2: A vertical cross-section of the above figure

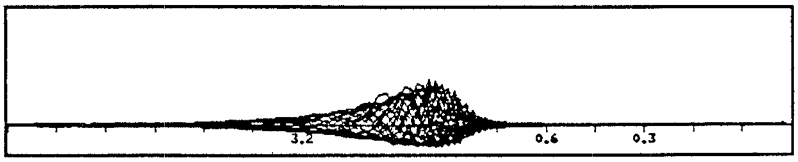

Figure 1.3: The envelope of 800Hz signal

top are sensitive to low frequency signals, which are corresponding to the beginning and the end of the cochlea respectively. It is known that every single tone signal will resonate at a particular point on the basilar membrane where the signal will generate highest firing rates on the nerve fibers around that point and decay rapidly thereafter. A list of the audible frequencies with their resonant channels is shown in table 1.1. Figure 1.3 shows the amplitudes of the channels by taking the vertical cross-sections in figure 1.1. It is interesting to note that channels before the resonant point basically act like half-wave rectifiers. Therefore, their firing patterns are also periodic and have the same frequency with the input signal. It can be seen clearly in figure 1.1.

A clear phase reversal can be seen among the channel responses near the resonant point, as shown in figure 1.2. The rapid decay in the response amplitude and the reversal in phase are the two most important characteristics of the ear model[3] that reveal the information of harmonics in the input signal. Occasionally the input signal consists of many components with different frequencies, it will be relatively hard to tell where the resonant points are by just looking at the amplitude envelope. In such cases, the phase re-

Frequency	Channel	Frequency	Channel	Frequency	Channel
400	99	500	92	600	87
700	83	800	79	900	76
1000	73	1100	71	1200	69
1300	67	1400	65	1500	63
1600	62	1700	60	1800	59
1900	58	2000	56	2100	55
2200	54	2300	53	2400	53
2500	52	2600	51	2700	50
2800	49	2900	48	3000	47
3100	47	3200	46	3300	45
3400	45	3500	44	3600	43
3700	43	3800	42	3900	42
4000	41	4100	40	4200	40
4300	39	4400	39		

Table 1.1: Channel numbers with their resonant frequencies

versal becomes the only reliable cue to tell where in the cochlea the signal resonates, i.e., what kinds of harmonics are there in the input.

Networks that are designed to extract the features described above have been created and tested in many aspects[4]. The so-called Lateral Inhibitory Networks consists of two stages of processing. The first one, called LIN I, uses the edge detection algorithm to examine whether a clear phase reversal and/or amplitude deterioration can be observed. The second stage LIN II then further enhances the results of LIN I. Figure 1.4 shows the LIN II output of an 800Hz monotone signal whose ear model outputs are shown in figure 1.1. Figures 1.5 and 1.6 show both the ear model and the LIN II outputs for an input signal which is composed of 400 and 800 Hz monotones and the duration of the 400 Hz signal is only half as long as that of 800 Hz. Conceptually, LINs generate output peaks only at locations corresponding to the harmonics of the input signal since LINs are basically edge detection processing and the 'edge' can be clearly seen only at the points where the harmonics resonate and decay quickly.

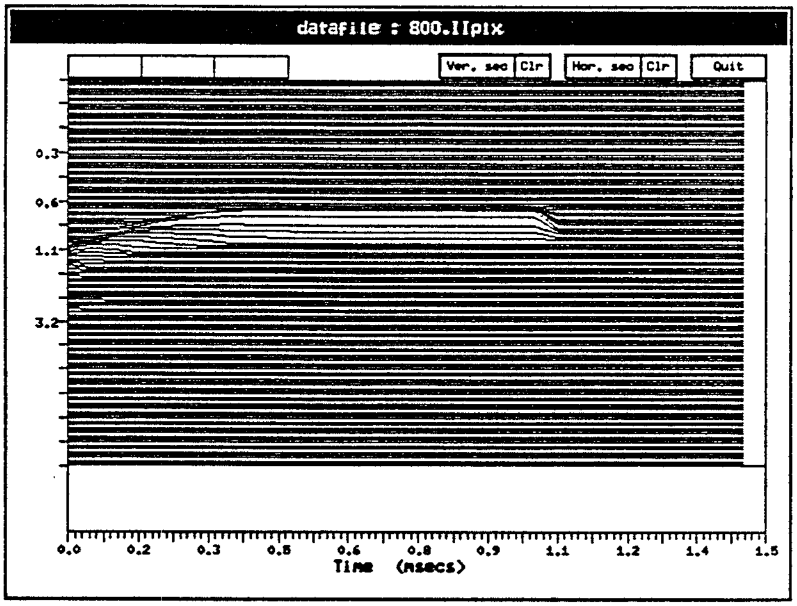

Figure 1.4: The LIN II output of 800Hz monotone signal

The amplitudes of LINs output, however, is not proportional to the relative intensity of the harmonics in the input signal because of the nonlinearity of the ear model. In our model, the high frequency components are usually amplified due to some pre-emphasis effect. The signal intensity and the output amplitudes are related in a rather complex way which is already far

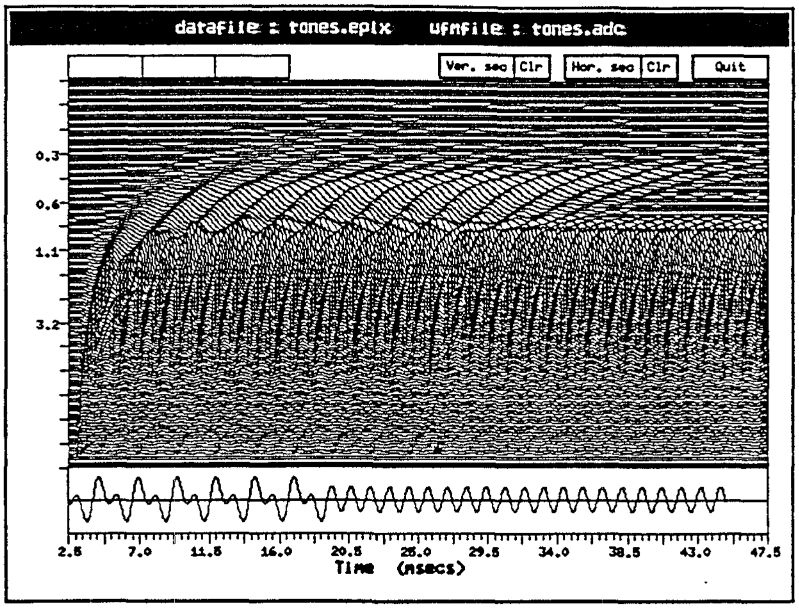

Figure 1.5: The ear model output of the mixture of a 400 and an 800 Hz monotone

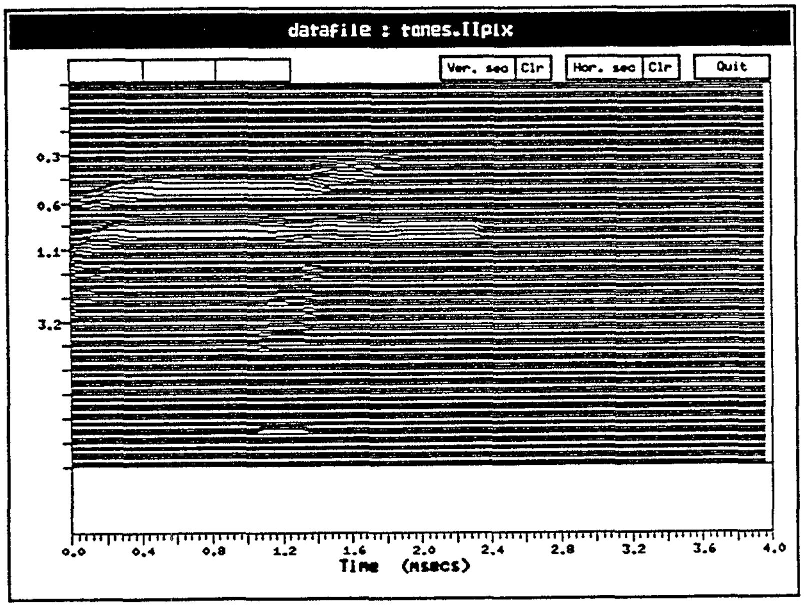

Figure 1.6: The corresponding LIN II outputs of the signal with harmonics of 400 and 800 Hz

beyond our scope of discussion.

Chapter 2

The Supervised Learning Algorithm

2.1 Introduction

The back propagation algorithm has become very popular and achieved great success in many applications[6]. It has been exclusively implemented on multi-layer, feed-forward networks based on the perceptron model. Each neuron in the network is assumed to have the same activation function which should be continuously differentiable. The logistic function is the most popular choice because it can be divided into regions corresponding to shut-off, quasi-linear and saturated states and thus behaves much more like that of a real neuron.

In consideration of computation load, the algorithm is seldom implemented as a real gradient descent as it should be. Besides, even if real

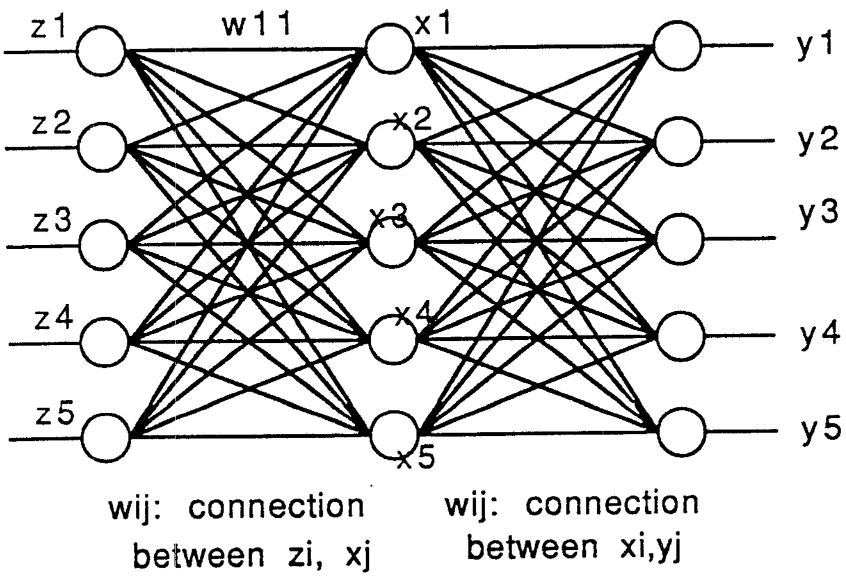

Figure 2.1: A two-layer, feed-forward network

gradient descent is embodied into the learning process, there are still some problems that should be carefully taken into account. In this chapter, we are going to show that it can actually be implemented to guarantee convergence if the problem is solvable. We will also argue that by properly choosing the parameters, the algorithm can more likely achieve a global optimal answer just like we have always got in our experiments described in the next two chapters.

2.2 Simple Gradient Descent Algorithm

Consider a two-layer network as shown in figure 2.1. We thus have

$$y_j = f(\sum_i w_{ij}x_i - \theta_j) \tag{2.1}$$

where

$$f(\alpha) = \frac{1}{1+e^{-\alpha}} \tag{2.2}$$

is the activation function and w_{ij} and θ_j are the weight and threshold respectively. Let d_j be the desired output of node j for a particular exemplar, we can define the error function as

$$E_1 = \frac{1}{2}\sum_j (y_j - d_j)^2 \tag{2.3}$$

which is a function in terms of the weights and thresholds in the network. The goal of the learning process is to adjust those parameters so as to minimize the error. The most efficient way to minimize a function is to adjust its variables along its gradient, i.e., we should let

$$\Delta w_{ij} = -\lambda \frac{\partial E_1}{\partial w_{ij}}$$

where λ is a positive number which, for the reason shown below, is called the learning rate of the process. The minus sign here indicates that we are going along the direction to minimize the function rather than to maximize it. However, since

$$\begin{aligned}\frac{\partial E_1}{\partial w_{ij}} &= (y_j - d_j)\frac{df}{d\alpha_j}\frac{\partial \alpha_j}{\partial w_{ij}} \\ &= (y_j - d_j)\frac{e^{-\alpha_j}}{(1+e^{-\alpha_j})^2}x_i \\ &= (y_j - d_j)(1-y_j)y_jx_i \end{aligned} \tag{2.4}$$

therefore,

$$\Delta w_{ij} = \lambda \delta_j x_i \tag{2.5}$$

where $\delta_j = y_j(1-y_j)(d_j - y_j)$. Similarly, we can obtain

$$\Delta \theta_j = \lambda \frac{\partial E_j}{\partial \theta_j} = -\lambda \delta_j \tag{2.6}$$

The same technique can be used to adjust parameters in hidden layers. From figure 2.1, we have

$$\begin{aligned} x_j &= f(\beta_j) \\ \beta_j &= \sum_i w_{ij} z_i - \theta_j \\ y_k &= f(\sum_j w_{jk} x_j - \theta'_k) \end{aligned} \tag{2.7}$$

and

$$\begin{aligned} \frac{\partial E_1}{\partial w_{ij}} &= \sum_k \frac{\partial E_1}{\partial y_k} \frac{\partial y_k}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial w_{ij}} \\ &= \sum_k \delta_k \frac{\partial \alpha_k}{\partial \beta_j} \frac{\partial \beta_j}{\partial w_{ij}} \\ &= x_j(1-x_j) \sum_k \delta_k w_{jk} z_i \end{aligned}$$

therefore,

$$\Delta w_{ij} = \lambda x_j (1-x_j) \sum_k \delta_k w_{jk} z_i \tag{2.8}$$

Similarly, we can obtain

$$\Delta \theta_j = -\lambda x_j (1-x_j) \sum_k \delta_k w_{jk} \tag{2.9}$$

Although these results are derived from a two layer network, they can be easily generalized for any multiple layer networks. To summarize, the adjusting formulae are

$$\begin{aligned}\Delta w_{ij}^{l-1} &= \lambda \delta_j^{l-1} x_i^{l-1} \\ \Delta \theta_j^l &= -\lambda \delta_j^{l-1} \\ \delta_j^{l-1} &= x_j^l (1 - x_j^l) \sum_k \delta_k^l w_{jk}^l \end{aligned} \tag{2.10}$$

where we use the superscript to denote the layer, i.e., x_j^l means the output of the jth neuron at the lth layer.

From equation 2.10 we can see that the error of the network is carried from the outer level toward the inner level by the term δ in which way the weights and thresholds of the network are adjusted. This is why the algorithm is known as the "back propagation" algorithm.

2.3 Problems and Modifications

In this section, we are going to discuss several problems induced by the algorithm described in the preceding section and exploit the possible solutions.

2.3.1 Convergence of the Algorithm

Equations 2.10 have been widely used for various problems although they are derived for the single exemplar case. For multi-exemplar problems, what has been usually done is, in each run of the training process, pick up an exemplar from the training set *randomly*, adjust the weights and thresholds according to the above formulae and then repeat the choosing and adjusting processes until the error is reduced below an acceptable value.

Intuitively, it may work fine when all the error functions associated with different exemplars share a common minimum. If this is not the case, the adjustment made to decrease error for a particular exemplar may not decrease the error for other exemplars. In the worst case, it may even increase the total error. Furthermore, it has been shown that the algorithm is doomed to fail if the error function of a particular exemplar possesses a very 'attractive' local minimum[8]. Under such cases, the total output error keeps going up and down and the algorithm fails to converge.

Since the point of convergence is quite crucial in our application, we choose to implement it as a 'real' gradient descent algorithm, i.e., by considering the overall error function

$$\varepsilon = \sum_i E_i$$

and taking advantage of the linearity of differentiation, we may obtain that the resultant adjustments of the parameters are simply the summation of adjustments for each exemplar. When implemented on the digital computer, we have to approximate the differential equations in the form of difference equations and, therefore, even with this modification, we still can not conclude that the error sequence will always be less and less after each run by the nature of gradient descent. The problem lies on the learning rate λ.

Theoretically, when the learning rate is small, the system should go along a path as shown in figure 2.2(a) and reach the minimum. Even if the rate is a little larger, we may have a path like figure 2.2(b) which still gives the desired

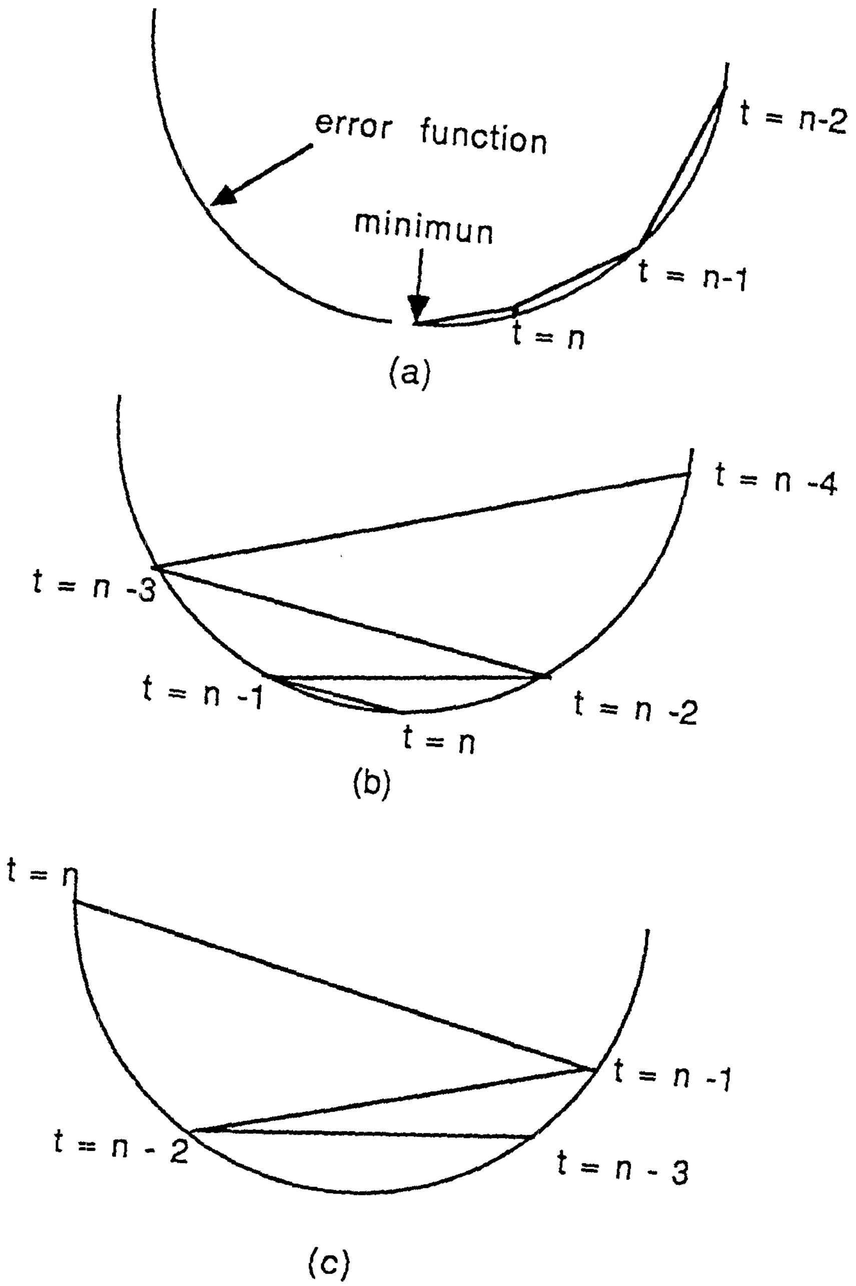

Figure 2.2: (a,b) ideal gradient descent (c) run-away case

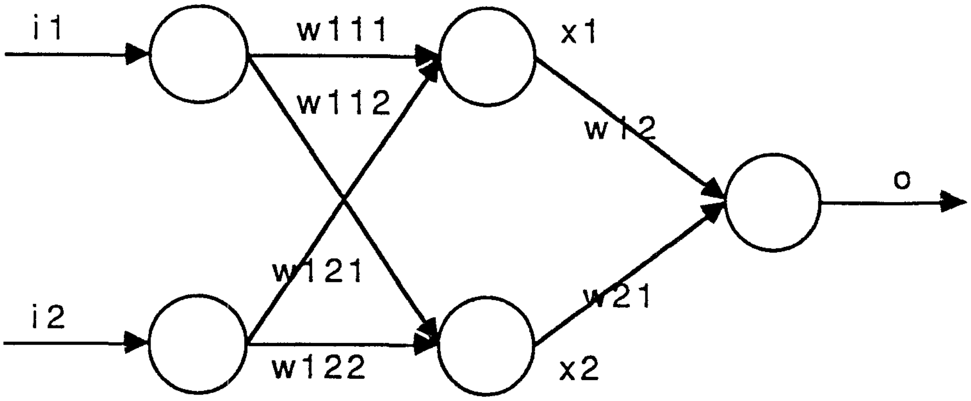

Figure 2.3: A simple two-layer network that can solve xor problem

result. However, if the learning rate is chosen too large, we might carry the movement too far and have a path as shown in figure 2.2(c). Unfortunately, the magnitude of the learning rate depends on the shape of the error function, which in turn depends on the problem itself. Since there has been no heuristic enlightening how we can choose the learning rate, it is usually chosen by trial and error.

With carefully chosen learning rate, we can guarantee that the error sequence is strictly decreasing if the problem is learnable, i.e., there exists a set of weights and thresholds that can yield the input-output relations recursively defined by eqs. 2.7. Take the generalized exclusive-OR problem for example. The two-dimensional plane is divided into eight parts instead of four as in the exclusive-OR(XOR) problem and every four disjoint parts are classified as the same group. The problem is considered hard to solve because the back propagation algorithm has to form a continuous surface to divide the 'discontinuous' space. Using a two-layer network like the one shown in figure 2.3, we can obtain the result as shown in figure 2.4.

The parameters for running the back propagation algorithm is very crucial in this example. The above result is solved with learning rate 0.2 and momentum 0.6. If a larger learning rate is used, our program will diverge. The effect of momentum is discussed in the following section.

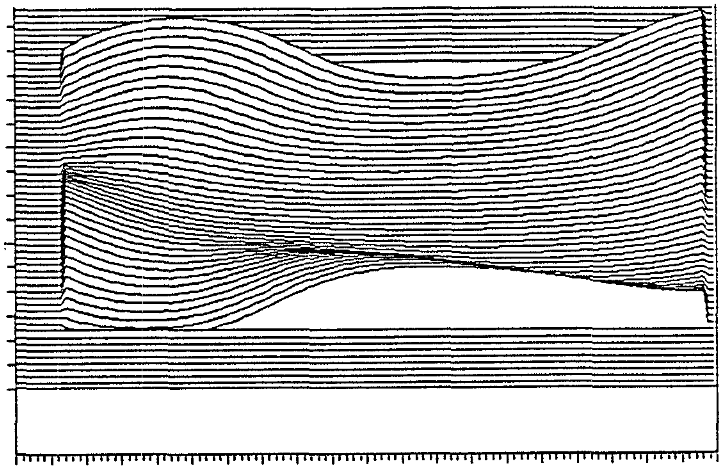

Figure 2.4: The surface that solves generalized XOR problem

2.3.2 Local Minimum Problem and Speed of Convergence

After carefully choosing the learning rate, we still have the problem that the algorithm can get stuck at a local minimum point and never reach the point that most minimizes the error. Furthermore, since the output of the neuron is between 0 and 1, the adjustments in equation 2.10 are therefore very small and result in a slow convergence.

These two problems can be solved by adding a 'momentum' term, i.e., we now let

$$\Delta w_{ij}(t) = \lambda x_j(1 - x_j) \sum_k \delta_k w_{jk}(t-1) z_i + m \Delta w_{ij}(t-1) \tag{2.11}$$

and

$$\Delta \theta_j(t) = -\lambda x_j(1 - x_j) \sum_k \delta_k w_{jk}(t-1) + m \Delta \theta_j(t-1) \tag{2.12}$$

where $w_{ij}(t)$ and $\theta_j(t)$ represent the weight and threshold at run t respectively and m is called the momentum.

The idea is to add some 'inertia' to the system so that it has a tendency to keep in the direction it has been moving. Therefore, when it is moving toward the minimum, it can be accelerated and reach the goal more quickly. On the other hand, if the vicinity of a local minimum is not so steep, the momentum can help to drive the process uphill and get out of that valley. As in the case of choosing a learning rate, there is still not a known systematic method to decide the magnitude of the momentum because they both depend on the problem itself very much. Typically the momentum gain m is chosen to be less than 1 so that the influence of the past moves will decay in time.

In the applications described in the next two chapters, we always choose a large momentum to get away from local minima. But a large momentum also has a tendency to prevent the searching process to stop at a global minimum if it does not reside in a deep valley. To avoid missing the global minimum, we keep monitoring the overall learning error sequence and once the error reaches zero, we claim that a global minimum is found and abort the searching process. Luckily enough, we always get the global minimum in our experiments.

2.3.3 Capability and Redundancy Effect

In a simple feed-forward network, we can associate with each training exemplar an equation in terms of the weights and thresholds of the network. For example, we can have the equations below for the network shown in figure 2.3:

$$\begin{aligned} o_k &= f(w_{21}x_1 + w_{22}x_2 - \theta_2) \\ x_1 &= f(w_{111}i_{k1} + w_{121}i_{k2} - \theta_{11}) \\ x_2 &= f(w_{112}i_{k1} + w_{122}i_{k2} - \theta_{12}) \end{aligned}$$

where the training exemplars in this example are specified as (i_1, i_2, o). The learning process is then to find proper values for the nine parameters: 6 weights and 3 thresholds. It is then very clear that there would be no solution if the number of exemplars exceeds nine, we will have more than nine equations to be solved while there are only nine variables. Remember that the number of equations should always be less or equal to the number of unknown variables. Therefore, a single layer network with 2 input and 1 output node can never learn the exclusive-OR(XOR) problem since there are only three variables but four restrictions to be satisfied. It turns out that the XOR problem should be solved by a network as shown in figure 2.3.

However, we have found that the number of nodes can neither be infinitely increased. In the XOR network, an additional node in the hidden layer will generate 4 more variables. Somehow when the number of nodes in the hidden layer is increased up to 5, the error sequence can no longer go directly downward as before and sometimes it even diverges. This is a typical problem when we use a discrete time algorithm to solve a continuous time differential equation. Therefore, in addition to lowering the computation load, it is also important to avoid having too many redundant nodes in the network.

2.3.4 Performance of the Algorithm

It is known[7] that a 2-2-1 network as shown in figure 2.3 can learn exclusive OR problem by back propagation algorithm, i.e., a set of weights and thresholds can be found for a network with 2 inputs, 1 output and 2 hidden nodes. Both the simplified and real-gradient-search algorithms are coded and executed on an HP-835 under UNIX system. The real-gradient-search algorithm does not take much more time as expected. Actually, the former algorithm was wandering in the beginning and wastes some time.

For a single layer network, like those used for phoneme discrimination described in the next chapter, we have tried to solve for the weights and thresholds directly by applying matrix algebra. Since what we deal with is a network with 128 input nodes, the dimension of the the matrices is so large that it takes no less time than the gradient descent algorithm. Actually, when dealing with so large a matrix, many additional procedures have to be employed to reduce truncation error. This makes the learning algorithm more attractive even when a single layer network is concerned.

Chapter 3

Phoneme Recognition for Single Speaker

3.1 Introduction

Recently, many successful experiments in speech recognition by neural network models have been reported(See for example,[10,11,13]). Most of them are done by training 2 or 3 layer feed forward networks to recognize a certain class of phoneme. After training, most achieve recognition rates over 95% for consonants and 65% for vowels. Nevertheless, people have not paid much attention to interpretate what the weights of neural networks really stand for.

However, in our experiments in which single layer, feed forward neural networks are trained to recognize LIN II outputs of the steady state bursts of phonemes, we find surprisingly that not only a single layer network is good enough to achieve satisfying performance but that the weights of the networks serve like a mask that extracts unique features of phonemes. Since our learning process only stops at the the global minimum of the error function, the significance of the results is that the best way to discriminate phonemes is to examine their characteristics in the frequency domain and label them with their unique features.

A phoneme is characterized by its voicing, manner and place of articulation. Sounds produced with excitement of the vocal chords are called voiced sounds. Reflected in the waveform, voiced sounds have larger energy than unvoiced sounds. The manner of articulation is related to the vocal tract dynamics in the sound producing process. While there are sounds that have to be produced with fixed shape of the vocal tract, there are others that strongly depend on the change in shape during their production. The temporal information in LIN spectra reveals their manner of articulation. In terms of these two characteristics, we can divide the phonemes in English into six classes: nasals, voiced/unvoiced stops, voiced/unvoiced fricatives, vowels and vowel-like sounds. Their characteristics will be described in detail in the following sections. Phonemes in each group then can be further discriminated by their spectral characteristic that is related to their place of articulation.

What we have done is basically to discriminate phonemes within the same group, that is, only the frequency information in the spectra is used. Specifically, we are trying to figure out whether there are important stationary features that can be used to identify uniquely each phoneme. We have built for each class of phonemes a specialized network and trained it to discriminate the phonemes with back propagation algorithm. In order to focus on the frequency information, the training and test data are taken explicitly from the steady state of the spectra. By steady state we mean the nasal murmurs for the nasals. For stops, it is the noise burst following the release from the closure. The diphthongs are not included because we believe they can be further divided and recognized as two concatenated vowels.

In the following text, we shall refer to three kinds of data patterns that are used as training or testing data. An *instantaneous pattern* refers to the channel response at a particular instant of time. Therefore, it is simply a vertical cross-section in the LIN II spectrum. A *short-time-average pattern* is the average response of the channel over the steady-state duration of a single phoneme. It is actually the average of the instantaneous patterns within that duration. Although we have tried to exclude all the transitions, the short-time-average patterns may still neglect the phoneme context. We further take the average of the short-time-average patterns of each phoneme extracted from different words of the same speaker and use it as the exemplar in our training process. This pattern is called the *time-average pattern* in the following text.

All our speech data are taken from the Ice-cream database which contains speech waveforms from both male and female speakers, each of whom spoke ten sentences in an ordinary way. The sentences were chosen in consideration of the balance of the phonemes. In this chapter, we are going to present the results of the experiments on single speaker's data. All the results in this chapters except fricatives are taken from the first male speaker in the database. The results of fricatives are from the data of the third speaker because there is no /zh/ in the first speaker's data. All the experiments have been repeated for 3 other male speakers and have yielded similar results although the phonemes are from different sentences and hence different context. The sentences and occurrence of phonemes are shown in table 3.1 and 3.2..

1	The birch canoe slid on the smooth planks.
2	A large size in stockings is hard to sell.
3	Glue the sheet to the dark blue background.
4	It's easy to tell the depth of a well.
5	These days a chicken leg is a rare dish.
6	Rice is often served in round bowls.
7	The juice of lemons makes fine punch.
8	The box was thrown beside the parked truck.
9	The hogs were fed chopped corn and garbage.
10	Four hours of steady work faced us.

Table 3.1: The sentences from which our phonemes are extracted

3.2 Performance Measurements

All of our networks are specialized to recognize a particular class of phonemes. An output node is assigned to turn on for a particular phoneme and all the rest output nodes are instructed to shut off. Ideally, when a test pattern is presented, only one output node will exhibit a high output. We then say that the test pattern is recognized as the phoneme that is trained to turn on this output node.

Symbol	Samples	Symbol	Samples	Symbol	Samples
aa	10	ae	2	ah	9
ao	7	ax	13	b	6
d	9	dh	8	er	4
ey	12	f	5	g	4
ix	11	iy	9	k	11
l	11	m	3	n	13
p	3	r	14	s	17
sh	2	t	8	th	3
uw	3	ux	2	v	4
z	12				

Table 3.2: Phonemes used in the single speaker experiments

For a rough measurement of our networks performance, we use *hard decision* rule for recognition, i.e., we always say that the input is recognized as the phoneme whose corresponding output node has the largest output value no matter how small it may be. The recognition rate is thus the percentage that the network makes the right choice on the testing or untrained patterns. The idea of this decision rule is directly from the ideal case in which only one node has a high output. If a decision has to be made in this level of processing without any other information, the hard decision rule is quite logical. Note that there is no not-recognized case under this decision rule.

To understand how errors are made, we also construct a confusion matrix for each network based on the recognition of the instantaneous patterns. The entity in the i^{th} row, j^{th} column of the matrix(for $i \neq j$) is the percentage that the j^{th} neuron has the largest output value when the testing patterns are supposed to turn on the i^{th} neuron. We can see from the confusion matrix how phonemes are mistaken for each other.

In most cases, however, there is noise in the input which boosts other output nodes. The noise may even come from the context interference. For example, figure 3.1 shows the nasal murmur of /n/ in the word 'snow'. It can be clearly seen that the burst of /n/ is deeply buried in the end of /s/ and the initial of /o/, both of which have higher responses and make the burst of /n/ very obscure. As a result, the hard decision rule will yield confusing recognition under such a condition. Furthermore, there are phonemes that are quite similar to each other in many aspects so that even a human being cannot always distinguish them very well without temporal or context

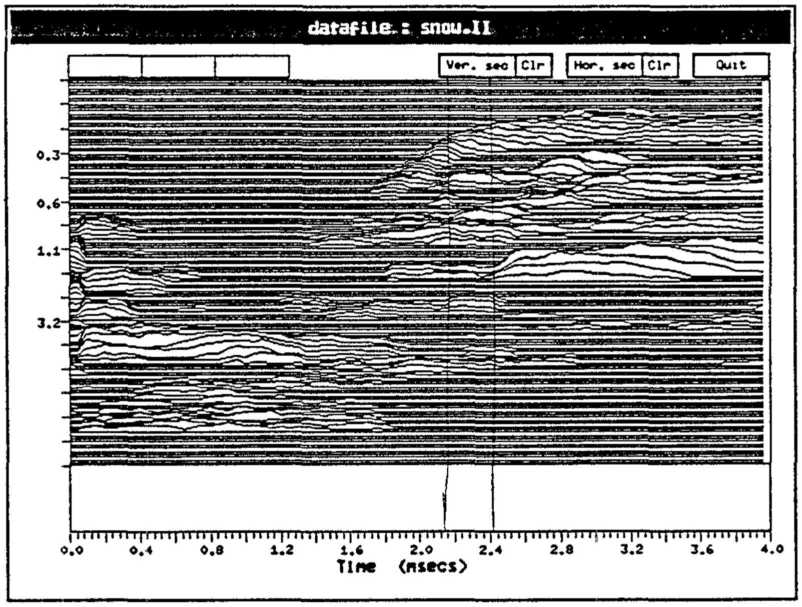

Figure 3.1: The LIN II outputs of the word 'snow'

information. This may corrupt the hard decision performance.

It is reasonable to assume that all phoneme recognition does not have to be made at a low but rather at a high level which takes both the spectral and temporal information into account. In such case, our networks may report the pattern of output node activation and let the higher level processing decide which one should be really chosen. However, to make the scheme work properly, the probability that each node is incorrectly turned on/off should be as low as possible.

For each node, two types of errors can be defined. A type 1 error, a *miss*, occurs if the node exhibits a low output when it is supposed to be high. Similarly, type 2 error, a *false alarm*, occurs when the node has a high output but should actually be a low one. By testing the networks on all the instantaneous patterns, we obtain the empirical distributions, f_{on} and f_{off}, of the output value for each neuron under the cases that it is supposed to turn on or off respectively. Then the empirical probabilities of error are

$$\begin{aligned} Pr(miss) &= \int_0^{\theta} f_{on}(x)dx \\ Pr(falsealarm) &= \int_{\theta}^{1} f_{off}(x)dx \end{aligned}$$

where θ is the threshold for on and off for the neuron. It is clear that the threshold plays a key role here. If the threshold is too high, we may suffer a huge amount of miss and if it is too low, then we may have too many false alarms. The probabilities of these two types of error must match the requirement set by the higher level processing unit. What is usually done is we define a cost, or penalty function in terms of the probabilities of error, i.e.,

$$\begin{aligned} C(\theta) &= \alpha Pr(miss) + \beta Pr(falsealarm) \\ &= \alpha \int_0^\theta f_{on}(x)dx + \beta \int_\theta^1 f_{off}(x)dx \end{aligned} \tag{3.1}$$

where α, β are the penalty coefficients for miss and false alarm respectively. The threshold θ is then chosen such that the cost $C(\theta)$ is minimized.

In the following text, we are also going to examine our network performance assuming the penalty coefficients are given generally according to the number of the output nodes. For example, if a network has 6 output nodes, we will assume that the penalty of miss is five times as large as the penalty of false alarm for every node in the network. We made this assumption just for demonstrative purpose. We shall refer to this kind of performance measurement as *deferred decision performance* and the previous one as *hard decision performance* in the following text.

In the rest of this chapter, we are going to discuss the performance of the networks that are trained to recognize different classes of phonemes in the order of nasàls, stops, fricatives and vowels. Nasals are first discussed independently because their radiation is totally different from any other phonemes but, since we cannot easily tell them from vowels simply by the voicing or the temporal information in the spectrum, we also add nasals to the vowels network and train them together.

3.3 Nasal Consonants

The nasal consonants are normally excited by the vocal cords and hence are voiced. Among all the phonemes, nasals are the only sounds that are radiated at the nostrils. The mouth cavity, on the other hand, is usually closed and serves like a zero which suppresses all spectral energy at its resonant frequency. Nasals used in English are summarized in the table 3.3. Due to this special way of radiation, in the spectrum nasals typically don't have significant outputs within the region where vowels have. This is because the all-pole model cannot be applied to nasals and zeros must be added to the vocal tract model due to the resonance in the mouth cavity. In the LIN II outputs, nasals have significant outputs in the higher frequency region corresponding to the upper(higher frequency) 'edge' of zero. Figure 3.2 shows the LIN II output of the word 'amnesia' in which typical LIN II patterns of nasals can be clearly seen. This may be the most important clue to distinguish nasals from the vowels by their LIN II outputs.

Place	Symbol	Example
Labial	m	ma'am
Alveolar	n	none
Palatal/velar	ng	sing (no initial form)

Table 3.3: Nasal consonants in English

An important difference between nasals spectral patterns arises from the size of resonant space in the mouth cavity. For example, the resonant cavity

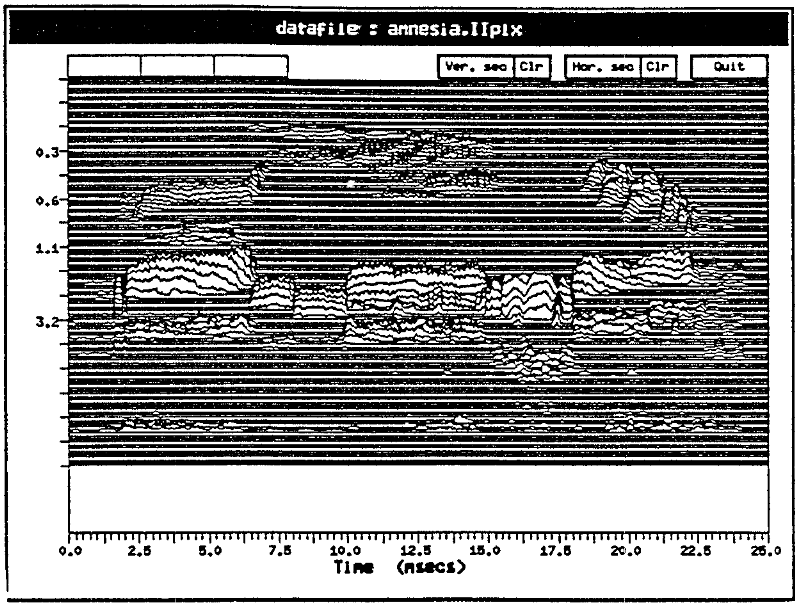

Figure 3.2: The LIN II output of the word 'amnesia'

of /m/ is longer than that of /n/ so that the resonant frequency of /m/ is lower. It should be reflected in the spectrum that /m/ has an edge of zero in the lower frequency than /n/ does. This difference between the nasals, which often is invisible in spectrograms is quite evident in the LIN II output. As an example, figure 3.2 shows the LIN II output for the word 'amnesia' in which the nasals are right next to each other and we can easily see the difference. The time-average LIN II outputs of nasals are shown in figures 3.3.

We created a network with only 2 output neurons, each of which is trained to turn on for one of the nasals. The weighted inputs and the weights after learning are shown in figure 3.4 and 3.5. It can be clearly seen that the network detects the difference in frequency of the edge of zero and use it as the key to discriminate these two phonemes. When we test the network with the short-time-average patterns, the recognition rate is 100%. The recognition

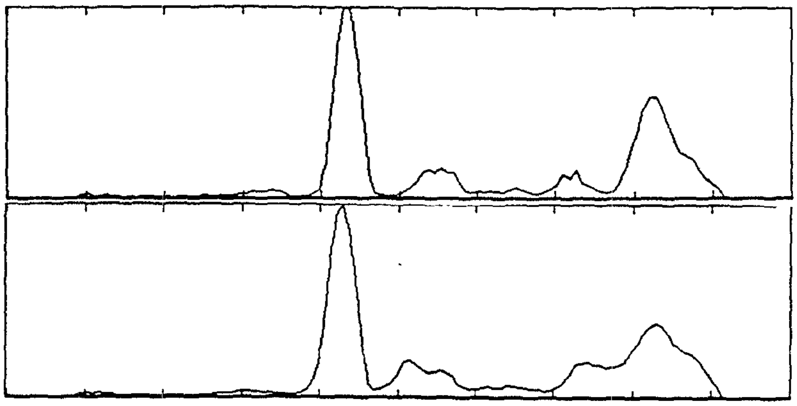

Figure 3.3: Time-average LIN II outputs for /m/ and /n/. The major peak of /m/ and /n/ are locateed at the frequency of 2100 Hz and 2200 Hz respectively

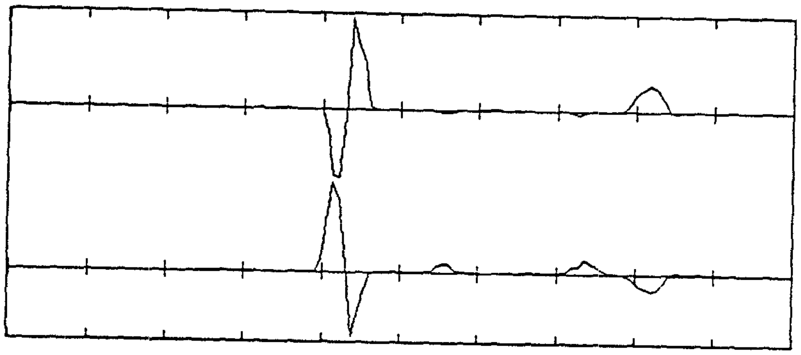

Figure 3.4: The weighted inputs for nasal /m/ and /n/

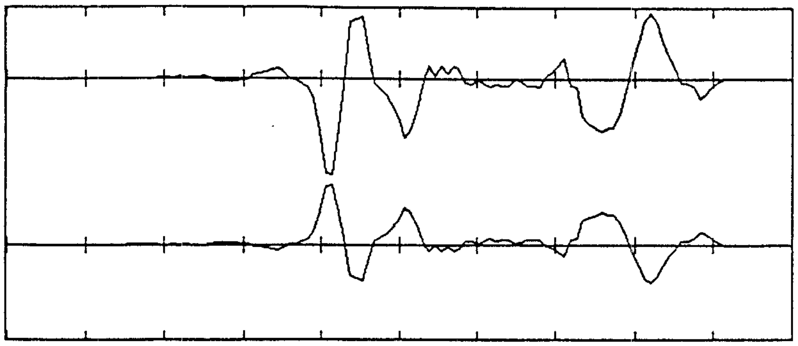

Figure 3.5: The sets of weights for discriminating nasals. The upper is the set of weights associated with /m/ neuron, the other is with /n/ neuron

here is based on hard decision rule described in the previous section. When the test patterns are chosen from instantaneous time slices, the recognition rate is still as high as 95%. This indicates the network does pick a reliable key to discriminate the nasals. We also have trained the nasals together with all the vowels in the vowel network in which the acoustic features of nasals can be seen more clearly. We shall describe this in the section 3.6.

3.4 Stop Consonants

Stop consonants, or stops, are a class of phonemes that depend heavily on the vocal tract dynamics for their creation. To produce these sounds, a complete closure has to be formed in the vocal tract. A pressure is then set up behind that point and suddenly released by an abrupt motion of the articulators. It is the explosion and the aspiration that characterize the stops. The stops can be produced with or without the vibration of vocal cords. Therefore, there exist both voiced and unvoiced stops. Consonants that can be grouped into voiced and unvoiced complementary pairs are called *cognates*.

The stop consonants used in English are listed in the table 3.4.

Place of	Voiced		Unvoiced	
articulation	Symbol	Examples	Symbol	Examples
Labial	b	be,crab	p	pop,pipe
Alveolar	d	day,good	t	tote,tit
Palatal/velar	g	go,dog	k	kick,cook

Table 3.4: Stop consonants in English

Since a complete closure should be formed before this class of phonemes can be produced, there exists a silent period immediately preceding the burst of each stop consonant. This temporal cue makes the stop consonants distinguishable from any other classes of phonemes. As an example, the LIN II output of the word 'maintain' is shown in figure 3.6.

The time-average patterns of the bursts of stop consonants are shown in figures 3.7 and 3.8. It can be clearly seen that the outputs of stops spread over different regions in the LIN II spectrum and therefore can be distinguished accordingly. As shown, labial stops like /b/ and /p/ have outputs in the lowest frequency and alveolar stops like /t/ and /d/ have outputs in the highest frequency. /g/ and /k/ have outputs in between. The spectra of /t,d/ and /k,g/ reflect the resonant effect of the frontal cavity which we will describe in more detail in the next section. It is interesting to see that the time-average patterns of voiced stops look quite similar to those of unvoiced stops and also have little response in the low frequency region. It results from

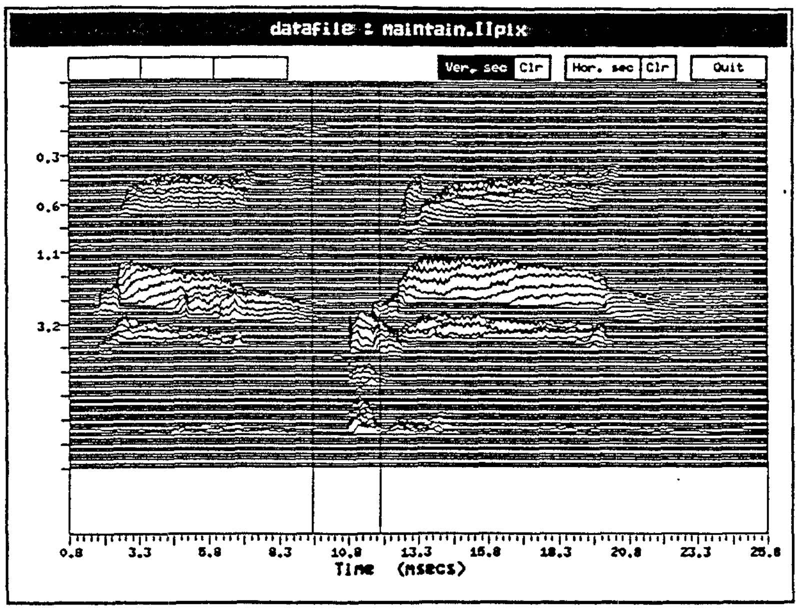

Figure 3.6: LIN II output of 'maintain' spoken by a male speaker

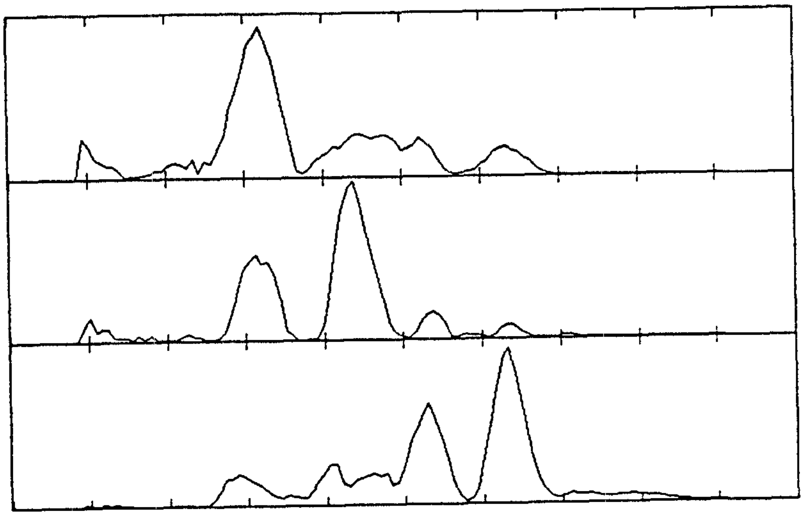

Figure 3.7: Time-average LIN II outputs for the bursts of /t/,/k/ and /p/, where the peaks are located at about 4000, 2000, and 780 Hz respectively.

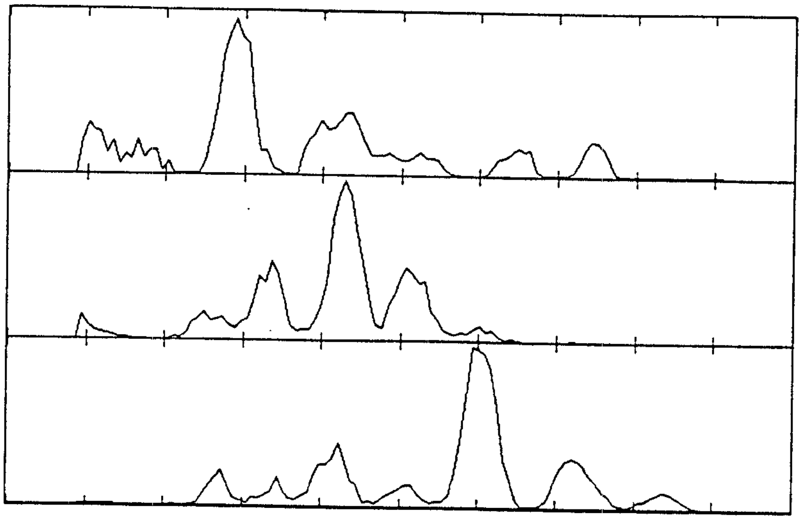

Figure 3.8: Time-average LIN II outputs for the bursts of /d/,/g/ and /b/, where the peaks are located at about 4700, 2200, and 880 Hz respectively.

the pre-emphasis in the cochlear model that suppresses the low frequency components. Nonetheless, the voicing information can be easily obtained even before the speech waveform is sent to the ear model. For example, the energy in the waveform can be used for this purpose. In our database, the energy of voiced phonemes can be higher by 3 dB to 20 dB than unvoiced phonemes and a threshold can be simply set up to detect voicing with more than 98% accuracy.

Two networks were trained to recognize this class of phonemes, one for voiced and one for unvoiced stops. After training, the weighted inputs of the stops are shown in figure 3.9 and 3.10 and the corresponding weights are shown in figures 3.11 and 3.12. It is not surprising that the weights for each phoneme are larger in the region corresponding to their major peak locations. The features that differentiate phonemes in this group can be even

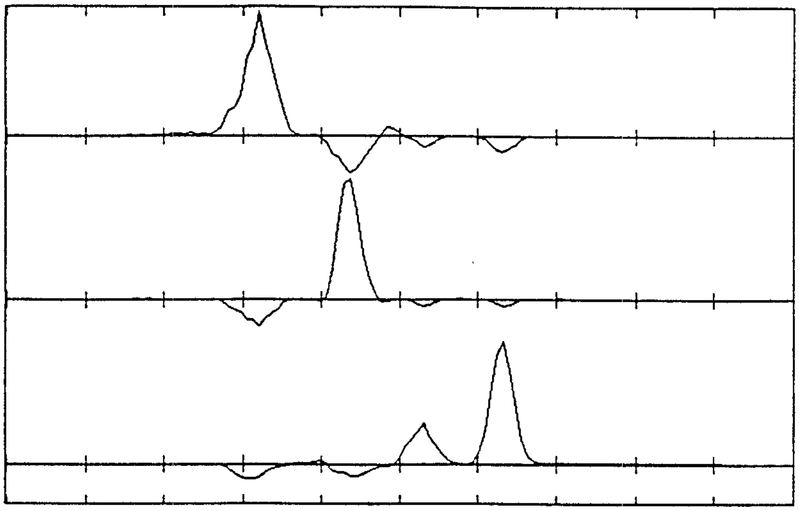

Figure 3.9: Weighted inputs of unvoiced stops

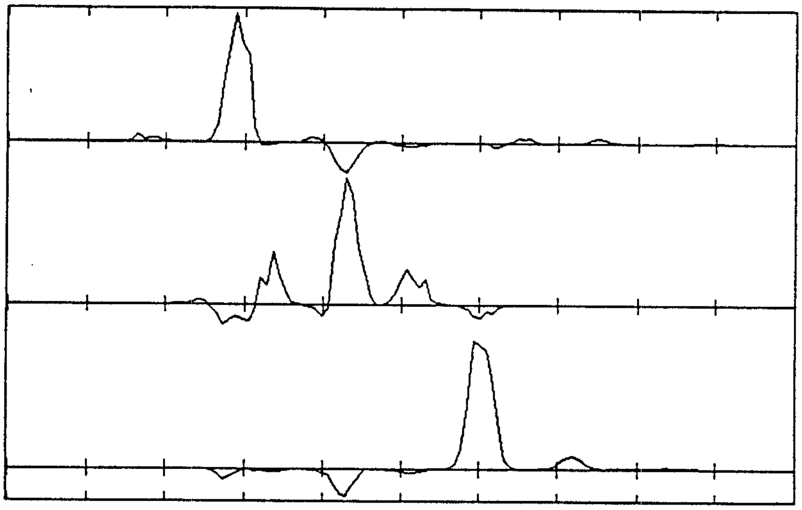

Figure 3.10: Weighted inputs of voiced stops

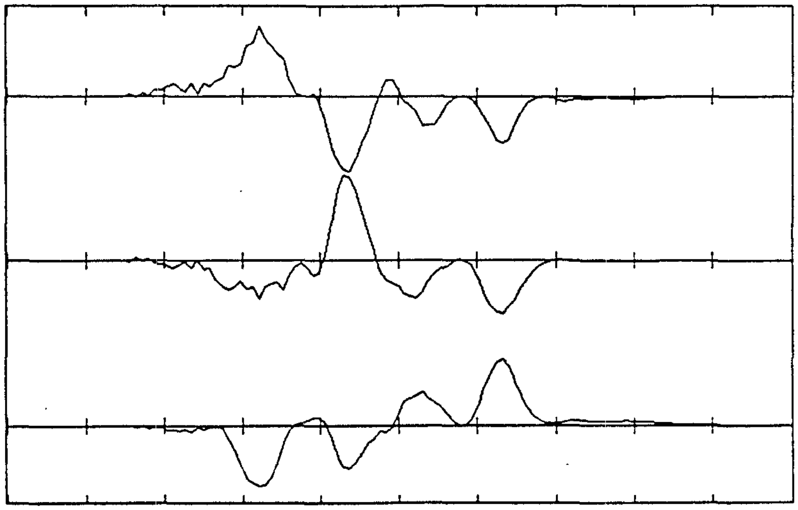

Figure 3.11: Weights for discriminating unvoiced stops /t,k,p/ respectively

more clear by looking at the weighted inputs as shown in figure 3.9 and 3.10. It can be seen, for example, although /d/ and /t/ have responses over a wide range of the LIN II spectrum, it is the the highest frequency output that differentiates them from other phonemes in this group. Actually, the lower frequency peak of /t/ or /d/ is negatively weighted in order to avoid confusion with /k/ or /g/, as can be seen in figure 3.9 or 3.10. When tested with short-time-average patterns, both the voiced and unvoiced networks yield perfect recognition. When tested with the instantaneous patterns, we can still have an average recognition rate of 76%. Table 3.5 summarizes the percentage of the mis-classification with respect to the input patterns. Significant amount of errors are made in recognizing /d/. This occurs because of the absence of high frequency outputs in many /d/ patterns, leaving their maximum peaks in the middle of the spectrum. Unfortunately that location overlaps the portion occupied by the major peak of /g/ and those patterns are therefore

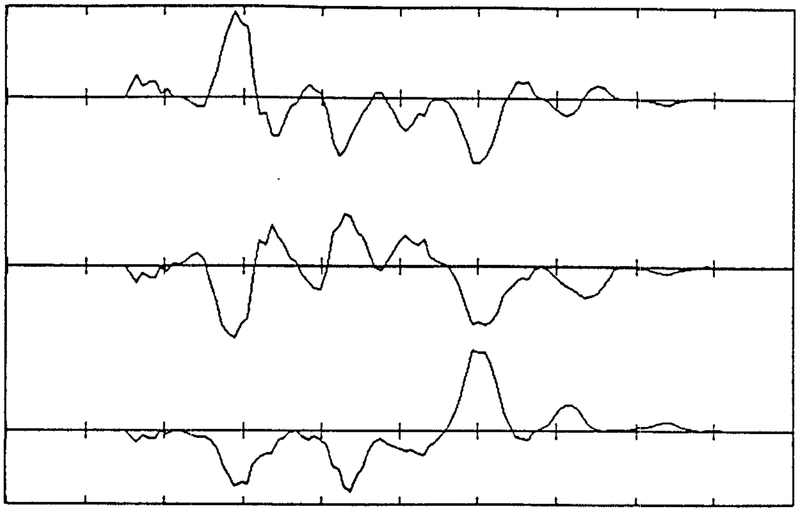

Figure 3.12: Weights for discriminating voiced stops /d,g,b/ respectively

	t	k	p	Samples		d	g	b	Samples
t	0.00	0.09	0.00	90	d	0.00	0.47	0.03	58
k	0.14	0.00	0.19	57	g	0.11	0.00	0.18	62
p	0.04	0.04	0.00	24	b	0.09	0.05	0.00	22

Table 3.5: The confusion matrix of stop consonants recognition

identified as /g/.

Table 3.6 shows the deferred decision performance with the penalty of miss is twice as much as that of false alarm. All the offsets are reasonably chosen around 0.5. Except for the probability of miss of /k/ is a little too high, the performance is in general acceptable. This suggests that the frequency information of the burst alone is suitable to discriminate the stop consonants from one another.

Output	threshold	Prob. of miss	Prob. of False Alarm
d	0.52	0.00	0.05
g	0.46	0.10	0.20
b	0.45	0.05	0.11
t	0.51	0.08	0.06
k	0.50	0.32	0.07
p	0.47	0.00	0.17

Table 3.6: The deferred decision performance of stop networks

3.5 Fricative Consonants

Fricatives are produced from an incoherent noise excitation of the vocal tract. The noise is generated by the turbulent air flow in a constriction in the vocal tract. Radiation of fricatives usually occurs at the mouth. There are both voiced and unvoiced fricatives just like there are both voiced and unvoiced stops. Both stop and fricatives consonants are cognates.

Fricatives used in English are listed in table 3.7. Their time-average patterns are shown in figures 3.13 and 3.14. It can be clearly seen that all the dominant outputs of fricatives are located in the high frequency and almost nothing in the low frequency. This is an important clue to distinguish fricatives from other classes of phonemes. Although the stops may have similar spectral shapes, they can still be distinguished from the fricatives by the longer burst duration of the fricatives.

Again, the LIN II output for voiced and unvoiced fricatives are quite similar to each other as in the case of stops. After training, the weighted inputs

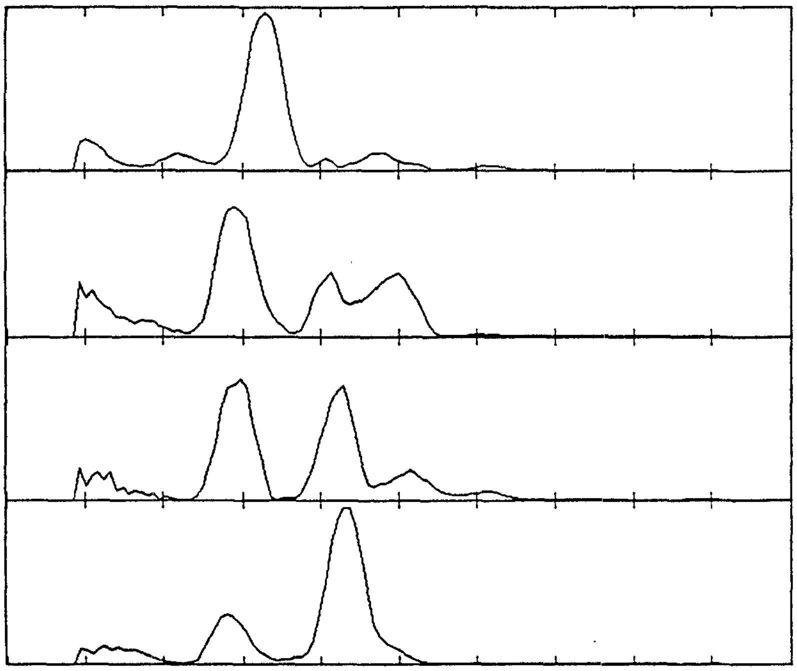

Figure 3.13: Time-average LIN II outputs of /s/,/f/,/th/,/sh/

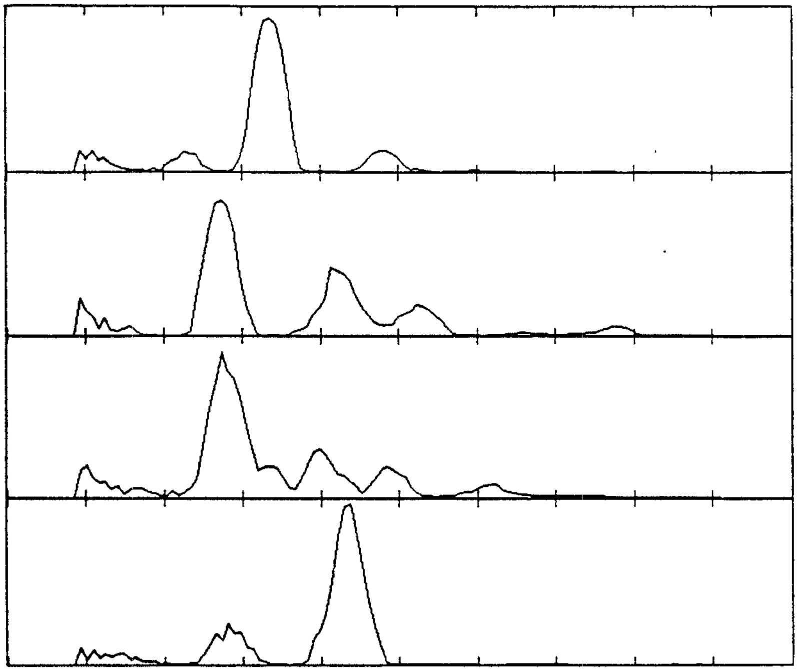

Figure 3.14: Time-average LIN II outputs of /z/,/v/,/dh/,/zh/

Place of	Voiced		Unvoiced	
articulation	Symbol	Examples	Symbol	Examples
Labio-dental	v	very,survive	f	for,beef
Dental	dh	then,clothe	th	thin,fifth
Alveolar	z	zero,analyze	s	see,less
Palatal/velar	zh	usual,mirage	sh	she,fish
Glottal			h	he (no final form)

Table 3.7: Fricative consonants in English

of the fricatives are shown in figure 3.15 and 3.16. The weights of the two networks are also shown in figures 3.17 and 3.18. With the weights shown above, we can achieve 80% recognition rate for unvoiced and 85% for voiced fricatives on short-time-average patterns. Errors can be seen exclusively on recognition /f/ versus /th/ and /v/ versus /dh/. It is rather understandable because these two fricatives have little difference in their LIN II outputs. It is even more clear by examining their weighted inputs as shown in figure 3.15 and 3.16. It is interesting to note that in the weighted input patterns of the fricatives, the peak location strongly reflects the point of constriction during its production. As the constriction point moves backwards toward the glottis, the peak location moves downward to the low frequency. This phenomenon is a result of lengthening in the frontal cavity of resonance which largely determines the dominant energy in the spectrum and its overall shape[2]. The peak locations of the weighted inputs for /f/ and /th/ are only three channels away from each other with about only 400 Hz difference in frequency. This

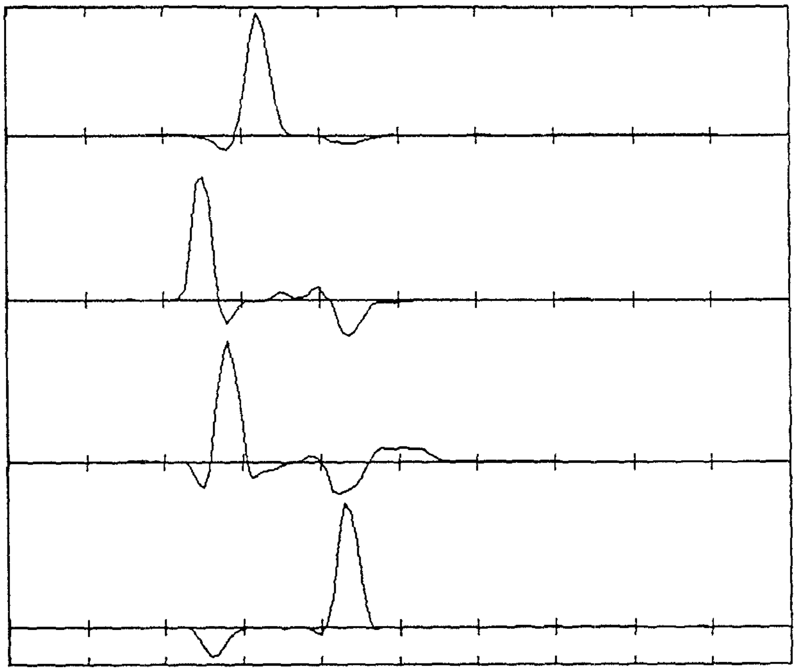

Figure 3.15: Weighted inputs of unvoiced fricatives /s,f,th,sh/

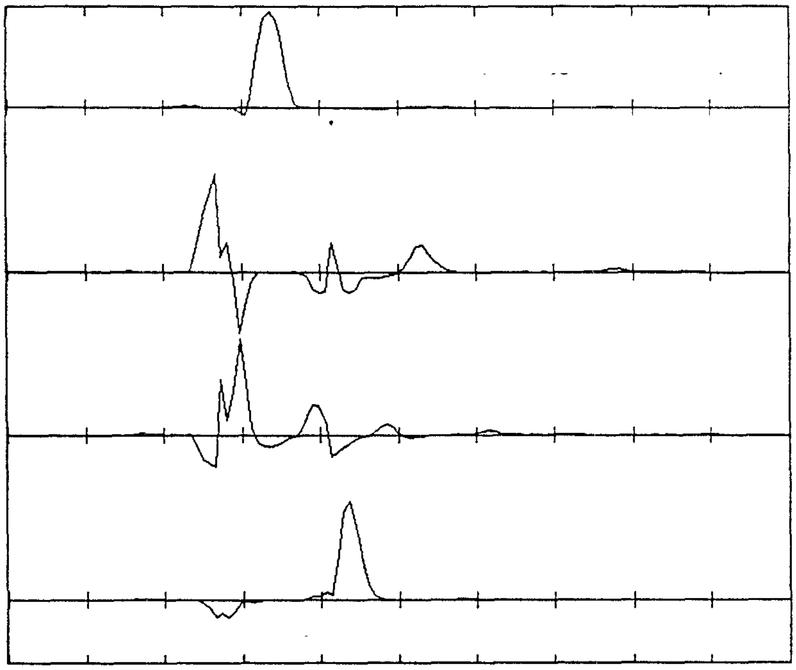

Figure 3.16: Weighted inputs of voiced fricatives /z,v,dh,zh/

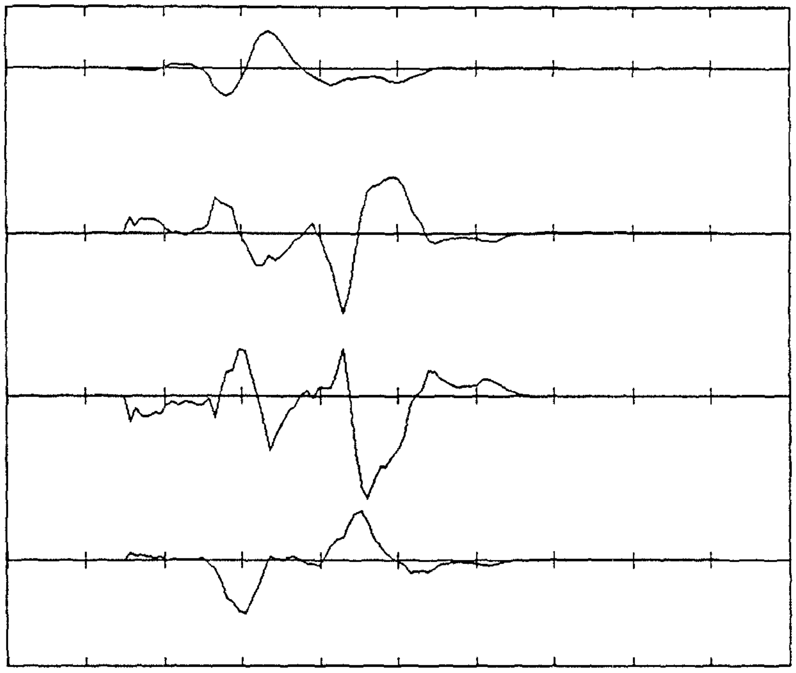

Figure 3.17: The weights for discriminating unvoiced fricatives /s, f, th,sh/ respectively.

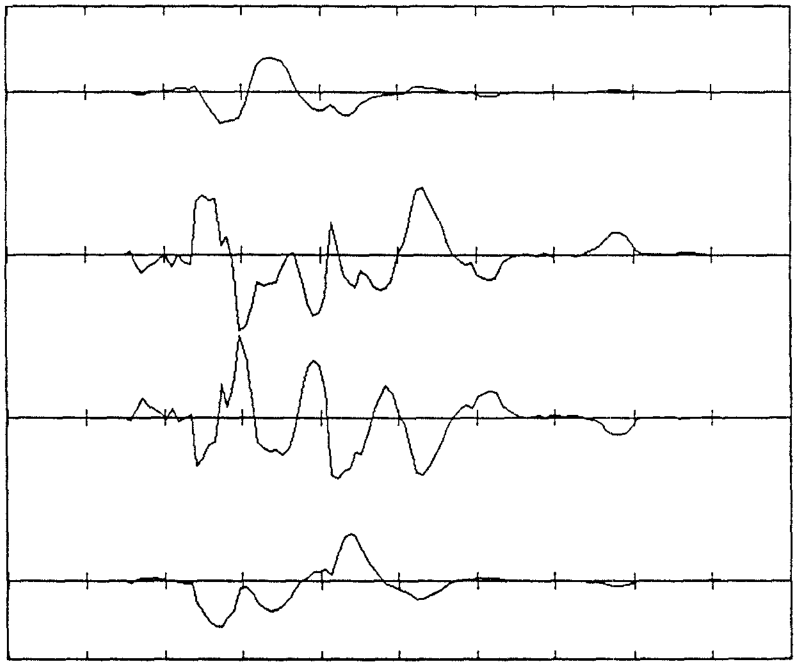

Figure 3.18: The weights for discriminating voiced fricatives /z,v,dh,zh/ respectively

is no surprising at all because the of the place of articulation is rather close for these two phonemes. The hard decision performance for instantaneous patterns is shown in table 3.8.

	s	f	th	sh	Samples		z	v	dh	zh	Samples
s	.00	.08	.14	.01	537	z	.00	.04	.00	.00	54
f	.03	.00	.30	.02	281	v	.01	.00	.16	.10	69
th	.01	.29	.00	.06	70	dh	.08	.42	.00	.03	191
sh	.06	.12	.17	.00	224	zh	.04	.11	.02	.00	46

Table 3.8: The confusion matrix of fricatives

Output	threshold	Prob. of miss	Prob. of false alarm
s	0.51	0.04	0.22
f	0.50	0.19	0.27
th	0.51	0.26	0.28
sh	0.49	0.09	0.22
z	0.50	0.03	0.16
v	0.52	0.25	0.15
dh	0.42	0.13	0.26
zh	0.54	0.24	0.04

Table 3.9: The deferred decision performance of fricatives

Table 3.9 summarizes the deferred decision performance with the penalty of miss is three times as much as that of false alarm. We believe that such an error rate is acceptable and correctable in a higher level processing. This suggests that the frequency information of fricatives is also suitable for their discrimination.

However, when analyzing data from other speakers, we notice that sometimes the highest peaks of /f/–/th/ and /v/–/dh/ do overlap a lot and the weighted input patterns look somewhat different in /f/ and /v/. We shall discuss this issue in the next chapter.

It is quite interesting to note that the weighted inputs of /s,z/ and /t,d/ match at high frequency. This goes with the knowledge that /t,d/ and /s,z/ are only different in manners of articulation. We also observe the similarity between /sh,zh/ and /k,g/.

3.6 Vowels and Semivowels

Vowels are produced exclusively by voiced excitation of the vocal tract. In comparison with the voiced fricatives, vowels have their dominant LIN II peaks exclusively within low frequency region and hence can be easily identified.

Vowels are usually classified by the shape of the vocal tract and the degree of constriction during their producing process. Table 3.10 lists the vowels labeled in our speech database.

In English, there are still two groups of sounds that resemble vowels very much. The first group contains /r,l/ that are called semivowels. Semivowel /l/ can form a syllable like vowels can, for example, consider /l/'s in the words like 'apple','double','castle'. The other group contains /w,y/ which are called glides. They are both produced with more constriction in the vocal

Degree of	Tongue hump position					
constriction	front		central		back	
High	/iy/	eat	/er/	bird	/uw/	food
	/ix/	kid			/ux/	good
Medium	/ey/	bed	/ah/	bus	/ao/	all
			/ax/	ago(unstressed)		
Low	/ae/	bad			/aa/	lock

Table 3.10: Vowels labeled in our database

tract and the tongue is not down. They are also characterized by voiced excitation, no nasal coupling, and sound radiation from the mouth – exactly the same characteristics as the vowels have. In our experiments, we trained the semivowel /l/ and nasals together with the vowels. The vowel /ax/ is not included because it is always unstressed and we usually cannot get a clear segmentation of it. Similarly, glides often do not have a stationary part in their bursts and therefore are not included in our experiments. However, for the reasons we have mentioned before, we include the nasals in the vowel network because the way to distinguish them is their spectral characteristics rather than the voicing or temporal information in the LIN II outputs.

The time-average patterns for this class of phonemes are shown in figure 3.19. It can be seen that, for vowels, only the first few formants dominate the LIN II outputs. As pointed out in [2], the information of the degree of constriction and tongue hump position as summarized in table 3.10 is strongly encoded in the LIN II outputs not only in the locations of peaks but also in their relative amplitudes. Generally speaking, the higher degree of

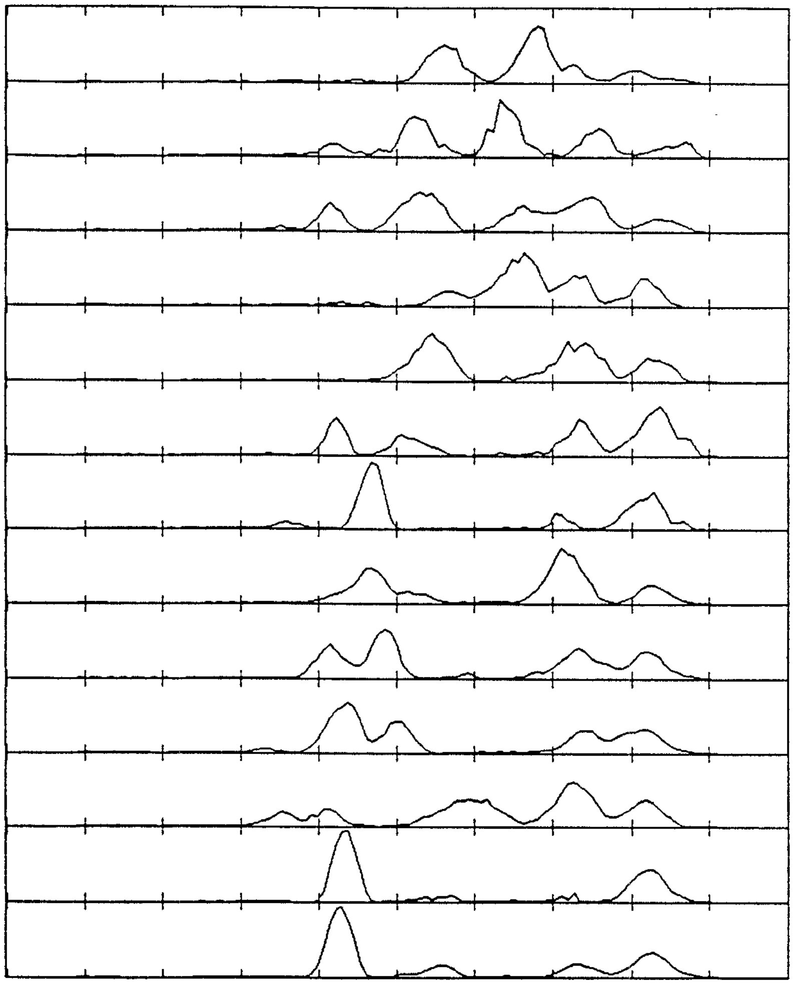

Figure 3.19: Time-average patterns for vowels, from top to bottom, /aa/, /ae/, /ah/, /ao/, /er/, /uw/, /ux/, /ey/, /ix/, /iy/, /l/, /m/, /n/

constriction yields higher ratio of peak amplitudes and as the tongue hump moving backward to the glottis, the dominating peak moves from high to low frequency. Therefore the amplitude ratios of 'close' vowels like /iy/ and /uw/ are larger than 'open' vowels like /ae/ and /aa/. Also, the dominant peak of /iy/ appears at rather high frequency(channel 57, around 1950 Hz) and the peak of /uw/ is at low frequency(channel 106, below 400 Hz) with the peak of /er/ at about 1200 Hz. The only exception is /ux/ has the high frequency component larger than its low frequency ones. We suspect that results from its short duration and hence more sensitive to the context so that what we've got is a polluted spectrum. Anyway, the features of vowels can be seen more clearly in the weighted input patterns as shown in figure 3.20. From this figure, we can see that the low frequency peaks of close front vowels like /iy/ and /ix/ are ignored and same for the high frequency peak of /uw/. For the open vowels, since their peaks are almost of the same magnitude and therefore equally important, the back propagation learning algorithm then conducts the give-and-take rule over the spectrum. For example, /ey/ has to take the low frequency one because its other peak is inhibited by the front vowels. /aa/, /ae/ and /ao/ then take their largest peaks and release the portion occupied by their other peaks. Since /ah/ has a rather flat spectrum, it can easily take the rest without conflicts. Such a give-and-take rule is rather fragile because the weighting heavily depends on the training set. For example, since /l/ is trained with all the vowels, its first formant(around 480 Hz) is totally ignored and its second formant(at 900 Hz) is boosted. In the next chapter, we'll see that experiments on other speakers yield somewhat different results. Nonetheless, the features of nasals

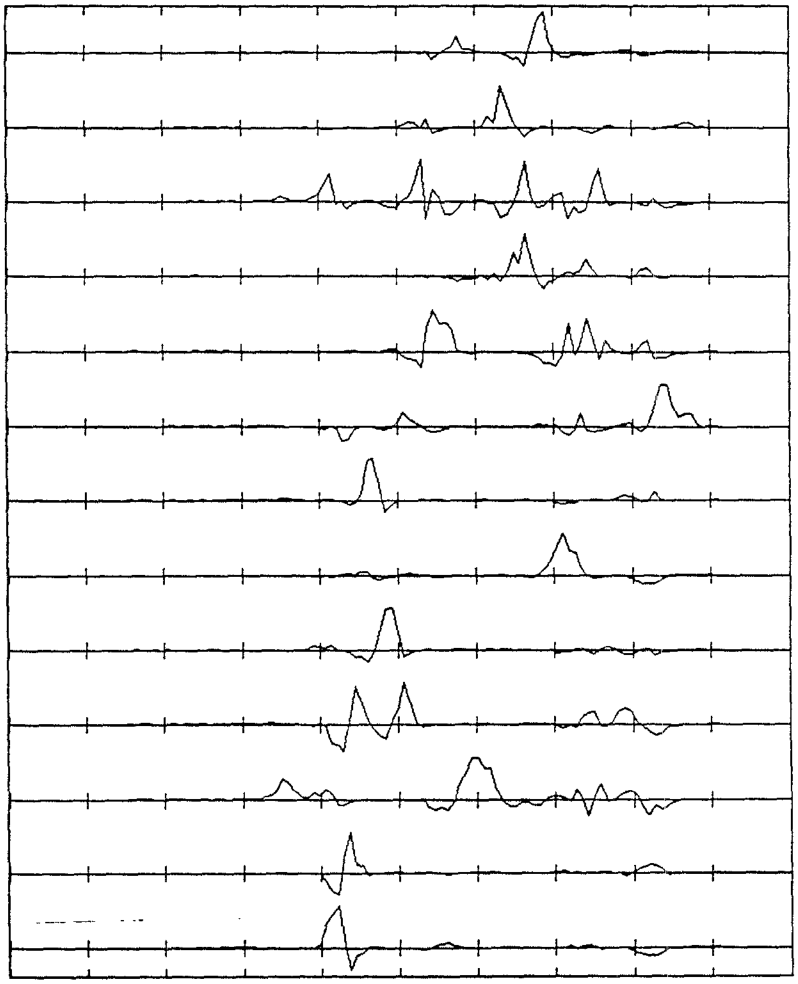

Figure 3.20: Weighted input patterns for vowels. From top to bottom, they are /aa/, /ae/, /ah/, /ao/, /er/, /uw/, /ux/, /ey/, /ix/, /iy/, /l/, /m/, /n/

	aa	ae	ah	ao	er	uw	ux	ey	ix	iy	l	m	n	Data
aa	.00	.00	.09	.12	.21	.00	.00	.06	.00	.00	.00	.00	.00	208
ae	.00	.00	.00	.09	.18	.00	.00	.00	.02	.18	.02	.00	.02	55
ah	.07	.05	.00	.01	.06	.03	.00	.12	.06	.16	.01	.02	.05	179
ao	.21	.01	.09	.00	.00	.00	.00	.01	.00	.00	.05	.00	.00	96
er	.05	.00	.05	.00	.00	.23	.00	.00	.01	.19	.01	.00	.00	135
uw	.00	.00	.03	.01	.00	.00	.00	.00	.00	.05	.00	.06	.06	108
ux	.00	.00	.01	.00	.00	.04	.00	.01	.17	.01	.00	.00	.01	78
ey	.04	.01	.04	.03	.01	.00	.04	.00	.03	.13	.00	.03	.01	211
ix	.01	.00	.00	.00	.02	.00	.11	.02	.00	.08	.00	.06	.12	166
iy	.00	.00	.02	.00	.00	.01	.01	.00	.15	.00	.00	.32	.07	238
l	.10	.03	.11	.00	.09	.02	.00	.07	.00	.03	.00	.04	.04	182
m	.00	.00	.00	.00	.00	.01	.00	.00	.00	.00	.00	.00	.13	68
n	.00	.00	.02	.00	.00	.04	.00	.00	.00	.02	.00	.23	.00	248

Table 3.11: Confusion matrix for vowels

and most vowels are not influenced at all.

The network can achieve 82% recognition rate on short-time-average patterns by using hard decision rule. Most errors come from mistaking /iy/ for /ix/ and /m/. Few errors are made in recognizing open from back close vowels. The hard decision performance on instantaneous patterns is shown in table 3.11. From this table, we see that basically the performance of this network is reasonable in that it makes mistakes among the vowels in the same group. Only a few open vowels will be mistaken for close vowels and front vowels are seldom mistaken for back vowels. The great amount of errors in confusing /ao/ as /aa/ is also not surprising after looking at their LIN II outputs. By using the time-average patterns as the exemplars and the short-time average patterns as the test patterns, the performance of our single layer vowels network is comparable to the performance of many multi-layer networks like [10].

In terms of deferred decision performance, we also achieve an acceptable results. Table 3.12 shows the thresholds and the probabilities of two types of errors with the penalty of miss is twelve times as much as that of false alarm for each output neuron. Again, the good performance suggests that the acoustic features of vowels are reliable cues to their identification.

3.7 Summary and Remarks

In this chapter, we have presented the ability of single layer networks in single speaker phoneme discrimination. We have found that the acoustic features of the phonemes are learned by the networks and used to discriminate phonemes. According to our experimental results, the acoustic features are generally a reliable key for phoneme identification even in a context free environment. We come to this assertion after noticing the high recognition rate in short-time-average patterns while the training exemplars are the overall average of them. Had phonemes not had stationary characteristics in the frequency domain that are invariant to the context, we could not have achieved such a good performance.

By confining our recognition network to a single layer, we can deduce the the acoustic features of each phoneme easily. All our results agree with the

Output	threshold	Prob. of miss	Prob. of false alarm
aa	0.40	0.22	0.25
ae	0.46	0.13	0.12
ah	0.39	0.09	0.29
ao	0.53	0.04	0.10
er	0.40	0.08	0.24
uw	0.59	0.21	0.10
ux	0.44	0.00	0.10
ey	0.47	0.06	0.16
ix	0.45	0.30	0.13
iy	0.38	0.08	0.33
l	0.43	0.09	0.12
m	0.47	0.07	0.16
n	0.43	0.12	0.22

Table 3.12: The deferred decision performance of vowels network

current knowledge in phonetics.

We have constructed the a confusion matrix for each class of phonemes which tells the network performance based on hard decision rule. Although we have achieved satisfying recognition rate, we have to note that it is very harsh a decision rule. For example, the results of recognizing the instantaneous patterns of the word of 'lemon'(Labeled as /l-ey-m-ax-n/) in sentence 7 are shown at the end of this chapter. The LIN II output of the word is shown in figure 3.21. We note that /n/ is not always correctly identified because there is a strong voicing in the low frequency which drives the /uw/

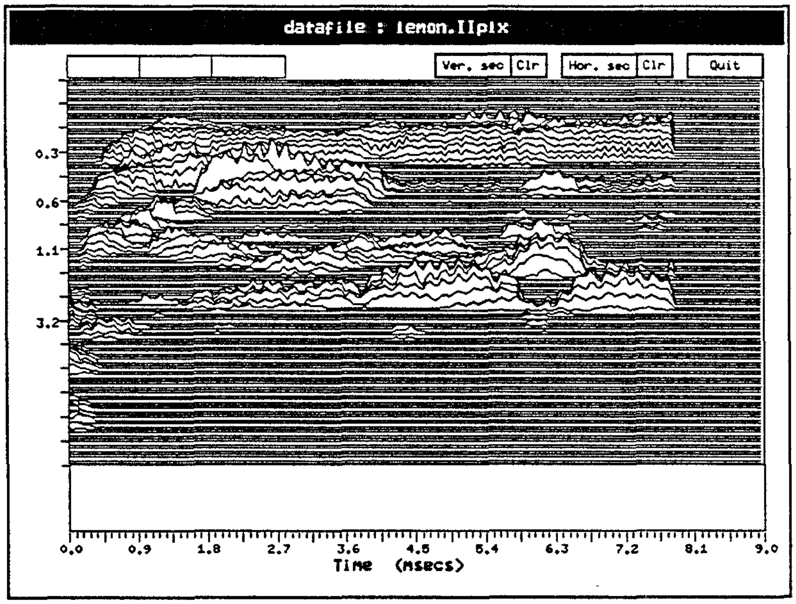

Figure 3.21: The LIN II output of the word 'lemon'

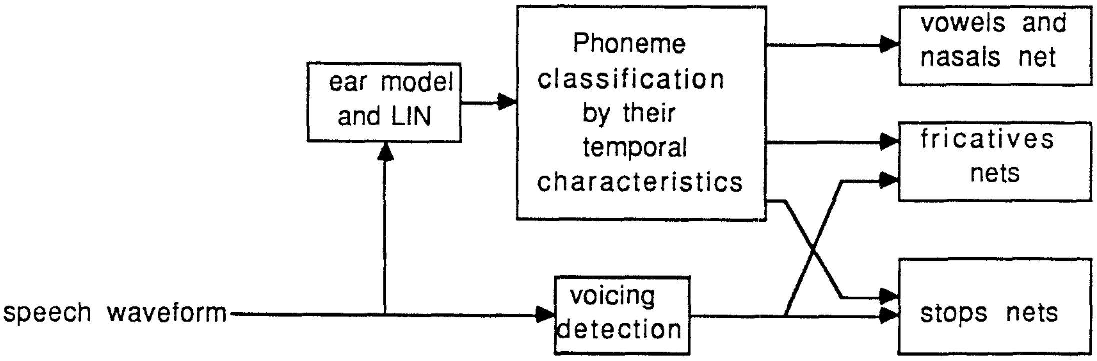

Figure 3.22: A speech recognition system based on our single speaker experiments

neuron to a higher output than /n/. Nevertheless, the 'correct' choice – /n/ neuron has always been exhibiting competitively high outputs which would be discarded by hard decision rule but may be coming out if the system is coupled with higher level processing.

In addition to inspecting the recognition rate based on hard decision rule, we also evaluate the networks' feasibility to be coupled with higher level processing. We have done so by calculating empirical probabilities of two types of error obtained from optimizing certain cost function which is hopefully given by higher processing units. With reasonable low probabilities of error, we are confident that the combined system should achieve very good performance.

In short, we can depict our system profile as shown in figure 3.22.

```
n 0.892   ix 0.568   l 0.494   | ey 0.572   n 0.460    ix 0.205
ah 0.705   l 0.666   n 0.580   | ey 0.762   uw 0.441    m 0.435
ah 0.673   n 0.671   ix 0.528  | ey 0.820   m 0.422    n 0.280
n 0.748   l 0.510   ix 0.491   | ey 0.743   m 0.590    l 0:223
ah 0.817   l 0.557   iy 0.408  | ey 0.772   m 0.635    l 0.269
ah 0.714   ux 0.553   l 0.549  | ey 0.808   m 0.684    er 0.275
aa 0.875   n 0.670   er 0.271  | ey 0.726   m 0.589    l 0.266
aa 0.878   l 0.737   n 0.432   | ey 0.789   m 0.717    l 0.298
l 0.815   ey 0.412   n 0.388   | ey 0.843   m 0.736    ix 0.200
l 0.819   ey 0.496   n 0.403   | ey 0.757   m 0.665    l 0.329
l 0.865   ey 0.417   m 0.365   | ey 0.829   m 0.725    l 0.282
l 0.857   ey 0.557   m 0.453   | ey 0.830   m 0.725    ux 0.168
l 0.870   m 0.453   uw 0.236   | ey 0.854   m 0.752    ix 0.200
l 0.884   m 0.369   ux 0.245   | ey 0.763   m 0.736    l 0.317
l 0.836   m 0.430   ix 0.248   | ey 0.840   m 0.774    ix 0.141
l 0.712   er 0.486   ey 0.478  | ey 0.837   m 0.764    l 0.386
l 0.748   er 0.465   ey 0.404  | ey 0.832   m 0.750    l 0.363
l 0.763   er 0.500   iy 0.418  | ey 0.711   m 0.684    l 0.296
l 0.707   er 0.502   m 0.430   | ey 0.785   m 0.703    l 0.319
l 0.665   er 0.486   uw 0.449  | ey 0.765   m 0.744    ux 0.198
l 0.649   uw 0.489   ey 0.454  | m 0.665   uw 0.569    l 0.303
ao 0.599   l 0.589   uw 0.501  | m 0.715   uw 0.541    l 0.296
er 0.707   uw 0.430   n 0.383  | m 0.790   ah 0.414    l 0.282
ao 0.807   er 0.557   n 0.409  | m 0.781   uw 0.410    l 0.313
ao 0.708   er 0.650   n 0.530  | m 0.699   ah 0.497    uw 0.443
ao 0.744   er 0.674   n 0.483  | m 0.760   uw 0.480    l 0.251
ao 0.739   er 0.701   n 0.540  | m 0.741   uw 0.511    l 0.319
ao 0.733   er 0.624   n 0.481  | m 0.707   uw 0.547    l 0.322
er 0.764   n 0.573   ix 0.287  | m 0.638   uw 0.537    ey 0.499
er 0.733   n 0.530   ix 0.311  | m 0.776   ah 0.486    uw 0.437
er 0.753   n 0.537   ux 0.298  | m 0.770   uw 0.369    l 0.281
er 0.704   n 0.473   ux 0.286  | m 0.777   ah 0.455    uw 0.355
er 0.721   n 0.554   iy 0.313  | m 0.715   ey 0.474    uw 0.297
ao 0.671   er 0.662   n 0.590  | m 0.796   ah 0.357    iy 0.231
er 0.629   n 0.543   ey 0.278  | m 0.767   ey 0.485    uw 0.339
n 0.611   ey 0.526   ix 0.248  | m 0.756   ey 0.428    uw 0.297
ey 0.547   n 0.524   ix 0.223  |
```

Figure 3.23: The recognition of the word ‘lemon’

```
m 0.720   ey 0.343   uw 0.281  |  uw 0.730   m 0.461    l 0.257
m 0.723   ah 0.332   ey 0.329  |  iy 0.568   m 0.353    l 0.290
m 0.594   er 0.415   uw 0.320  |  uw 0.721   iy 0.474   m 0.443
m 0.529   uw 0.296   l 0.237   |  m 0.525    l 0.349    ao 0.157
m 0.612   uw 0.241   n 0.216   |  iy 0.559   m 0.456    ix 0.167
m 0.691   n 0.240    l 0.170   |  iy 0.730   l 0.442    m 0.335
m 0.709   uw 0.295   n 0.209   |  uw 0.835   iy 0.760   m 0.482
m 0.708   n 0.231    l 0.176   |  iy 0.830   m 0.586    l 0.356
m 0.765   uw 0.325   ey 0.315  |  iy 0.916   m 0.507    l 0.396
m 0.750   uw 0.309   ux 0.255  |  iy 0.924   m 0.455    l 0.341
m 0.753   ux 0.264   ao 0.184  |  iy 0.933   m 0.537    l 0.364
m 0.722   ux 0.276   n 0.177   |  iy 0.913   m 0.513    l 0.397
m 0.800   iy 0.294   l 0.171   |  iy 0.922   m 0.406    l 0.380
m 0.767   ux 0.290   iy 0.286  |  iy 0.882   ae 0.458   ix 0.353
m 0.807   iy 0.308   l 0.209   |  iy 0.889   m 0.484    l 0.363
m 0.791   iy 0.291   l 0.196   |  uw 0.881   iy 0.797   m 0.414
m 0.837   iy 0.329   uw 0.190  |  uw 0.881   iy 0.817   ah 0.328
m 0.798   iy 0.307   l 0.193   |  uw 0.881   iy 0.806   ux 0.308
m 0.831   uw 0.307   ux 0.289  |  uw 0.857   iy 0.852   l 0.359
m 0.805   iy 0.288   uw 0.204  |  uw 0.874   iy 0.681   ux 0.296
m 0.827   iy 0.333   uw 0.178  |  uw 0.767   iy 0.565   ux 0.283
m 0.731   iy 0.386   l 0.179   |  uw 0.738   iy 0.460   m 0.334
m 0.750   ux 0.297   iy 0.274  |  m 0.537    iy 0.287   n 0.228
m 0.743   iy 0.392   n 0.213   |  m 0.615    n 0.344    l 0.165
m 0.742   iy 0.324   n 0.229   |  m 0.581    n 0.457    ey 0.220
m 0.716   iy 0.387   l 0.189   |  uw 0.503   n 0.487    ux 0.245
m 0.732   iy 0.304   n 0.209   |  m 0.525    n 0.376    l 0.211
m 0.755   ux 0.246   n 0.231   |  uw 0.569   m 0.473    n 0.443
m 0.758   ux 0.261   n 0.213   |  n 0.542    ux 0.291   l 0.234
m 0.781   ux 0.296   l 0.212   |  uw 0.642   n 0.579    ix 0.277
m 0.755   ux 0.264   n 0.205   |  uw 0.559   n 0.485    l 0.242
m 0.740   n 0.260    l 0.208   |  n 0.606    m 0.346    ix 0.264
m 0.757   uw 0.217   n 0.201   |  n 0.633    m 0.407    ix 0.241
m 0.736   iy 0.274   l 0.204   |  n 0.676    m 0.325    ix 0.264
m 0.640   n 0.284    l 0.222   |  m 0.497    n 0.478    ix 0.231
m 0.583   iy 0.292   n 0.239   |  n 0.615    m 0.406    ah 0.213
uw 0.654  m 0.499    l 0.234   |  n 0.623    m 0.440    ix 0.251
```

Figure 3.24: The recognition of the word 'lemon'(Cont'd)

```
uw 0.658  n 0.611  ux 0.256
uw 0.598  m 0.495  n 0.415
uw 0.722  n 0.612  l 0.204
n 0.581  m 0.432  l 0.211
uw 0.691  n 0.578  ux 0.261
uw 0.678  m 0.457  n 0.439
uw 0.786  n 0.586  ey 0.207
uw 0.728  n 0.473  ix 0.259
uw 0.929  ae 0.401  m 0.390
```

Figure 3.25: The recognition of the word ‘lemon’(Cont’d)

Chapter 4

Multi-speaker Phoneme Recognition

4.1 Introduction

In this chapter, we are going to show the results of the same experiments repeated for 3 other male speakers. We have found the acoustic features discussed in the previous chapter can be still seen, i.e., the overall shapes of the LIN II spectra for different speakers are similar to each other except the peak locations shift along the spectrum, which means the corresponding formant frequencies vary from speaker to speaker. Even so, our networks trained on the average of the time-average patterns of these speakers still achieve satisfying recognition rate in recognizing their short-time-average patterns. We have found that by taking time average, we can enhance the crucial features and lower down the confusion induced by speaker variation. Although

Symbol	Samples	Symbol	Samples	Symbol	Samples
aa	24	ae	21	ah	26
ao	27	ax	93	b	26
d	32	dh	59	er	20
ey	50	f	29	g	15
ix	30	iy	72	k	43
l	57	m	21	n	62
p	20	r	56	s	70
sh	11	t	54	th	9
uw	15	ux	11	v	20
z	35	zh	1		

Table 4.1: Phonemes occurred in the four speakers' speech data

the recognition in each individual's instantaneous patterns may have a large amount of errors, we are still convinced that there are certain important features that can be used reliably to identify phonemes by observing their weighted inputs.

The occurrence of all the phonemes in these four speakers' speech data is shown in table 4.1. In the following text, we shall refer to the averages of all the speakers' time-average patterns as overall-average patterns.

As before, we evaluate our networks in both hard decision and deferred decision performance in recognizing the instantaneous patterns. However, in this chapter, we use the short-time-average patterns instead as our test samples in this chapter. The reason is that in the single speaker's case, we were focusing on how our networks perform on the variant data induced by context change. By testing the instantaneous patterns, we can tell how well the features learned by the networks work. Here in the multi-speaker case, we are concerned more with the data variation caused by different speakers. In other words, we are examining whether there are cross-speaker acoustic features that are reliable enough for phoneme recognition. Henceforth, all our performance measures in this chapter are based on the short-time-average patterns except where explicitly stated.

4.2 Stop Consonants

The time-average patterns of these four male speakers are shown in figure 4.1 and 4.2. In contrast, figure 4.3 and 4.4 show their averages. As expected, the features seen for single speaker can be seen consistently in general. Among those, the unvoiced patterns are much more consistent across the speakers than their voiced counterparts because the production of unvoiced sounds does not excite the vocal cords and therefore less variant to speakers. It is thus not surprising that the weighted inputs of unvoiced stops from the network trained on the overall-average patterns almost overlap the average of weighted inputs trained on each speaker's time-average patterns, as shown in figure 4.5 and 4.6 respectively.

For voiced stops, we notice that except there is one strange high frequency peak in /b/, the general shapes of the overall-average patterns still reveal the frequency information of these three phonemes and bear much similarity to the patterns of single speaker and unvoiced stops. The upper peak in /b/

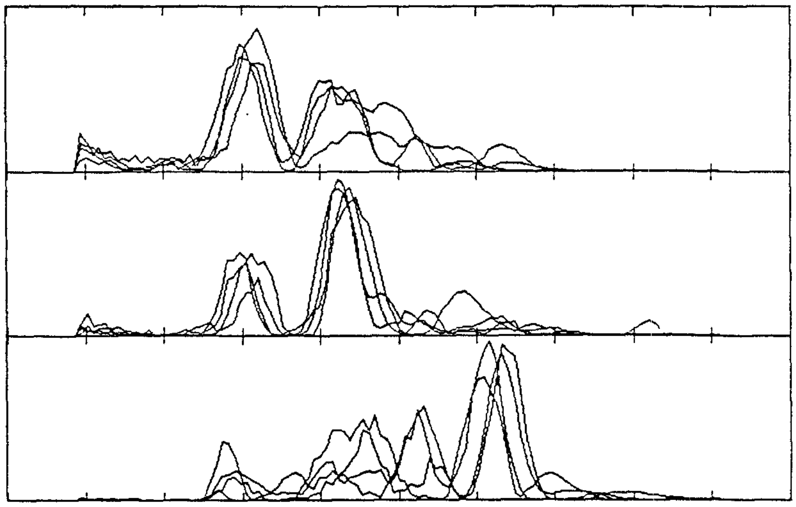

Figure 4.1: The time-average patterns of /t/,/k/,/p/ of four male speakers

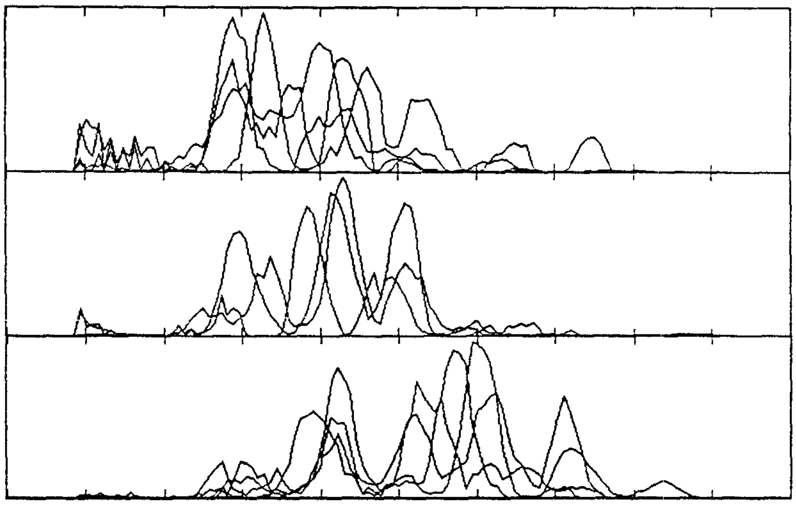

Figure 4.2: The time-average patterns of /d/,/g/,/b/ of four male speakers

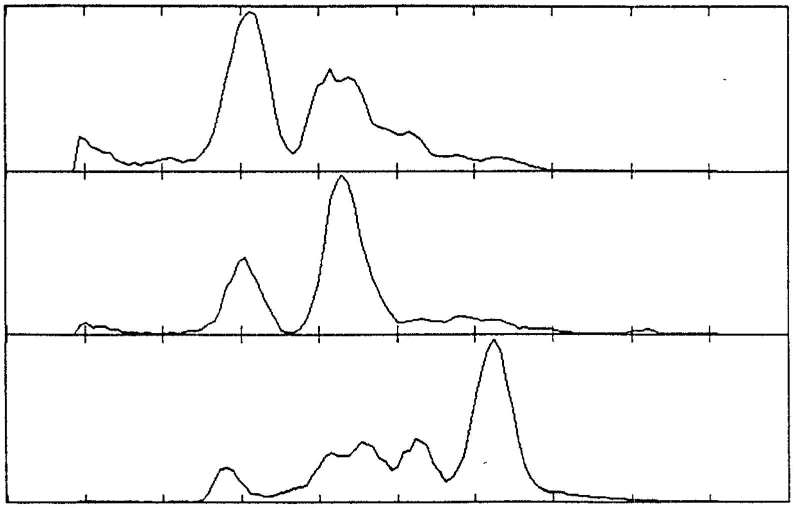

Figure 4.3: The overall-average patterns of /t/,/k/,/p/ of four male speakers

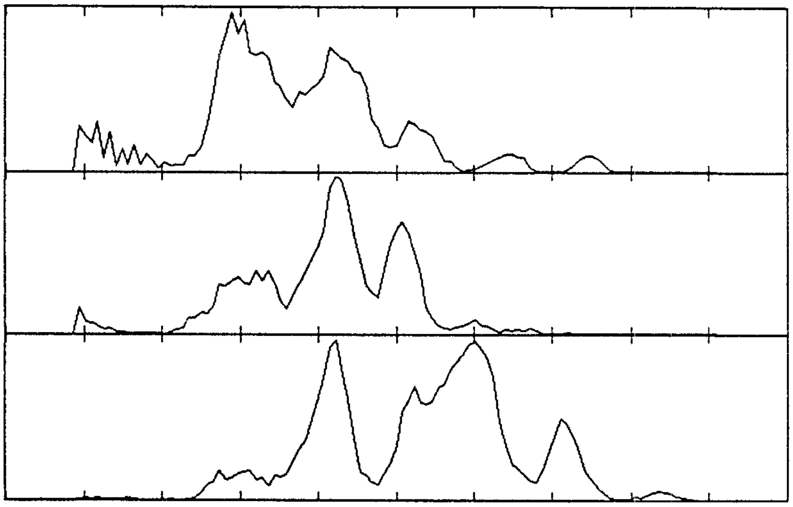

Figure 4.4: The overall-average patterns of /d/,/g/,/b/ of four male speakers

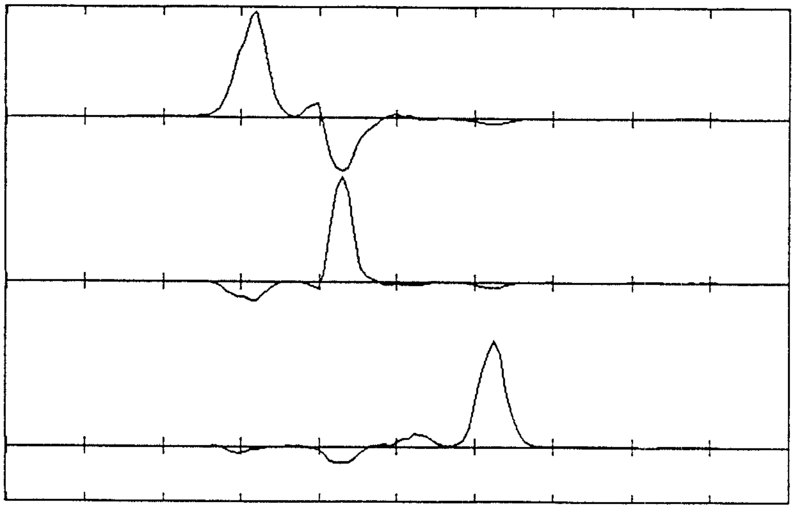

Figure 4.5: The weighted inputs of unvoiced stops from the net trained on the overall-average patterns

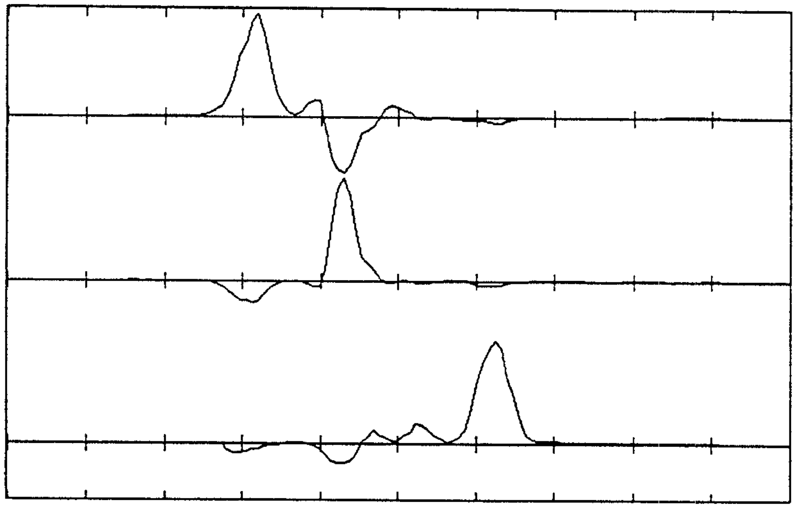

Figure 4.6: The average of four speakers' weighted inputs for unvoiced stops

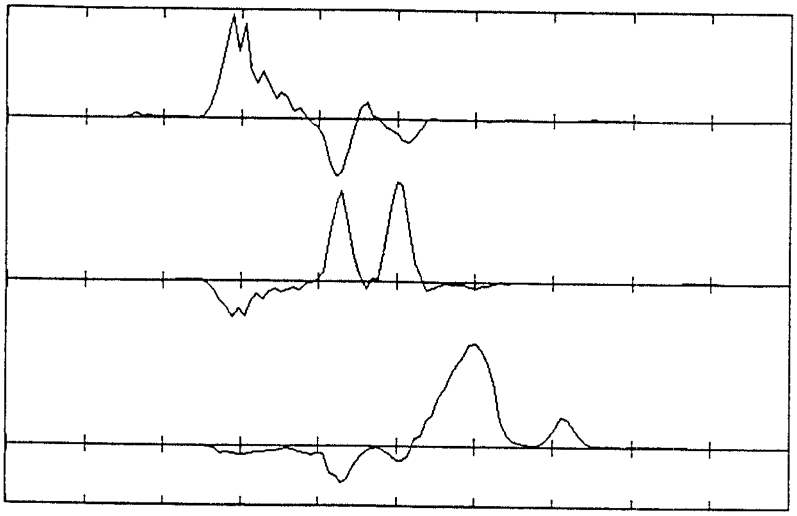

Figure 4.7: The weighted inputs of voiced stops from the net trained on the overall-average patterns

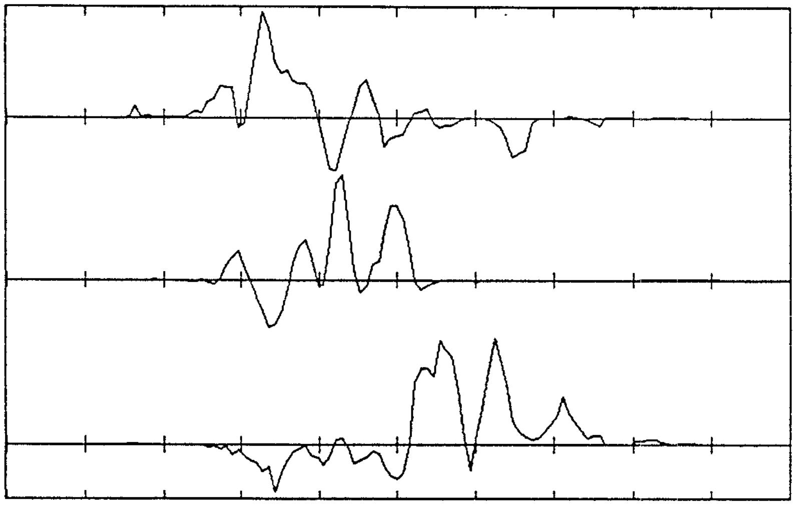

Figure 4.8: The average of four speakers' weighted inputs for voiced stops

Input	t	k	p	Input	d	g	b
t	0.00	0.09	0.09	d	0.00	0.10	0.00
k	0.11	0.00	0.06	g	0.15	0.00	0.00
p	0.00	0.10	0.00	b	0.00	0.09	0.00

Table 4.2: The confusion matrix on recognizing short-time-average stop patterns

results from the coincident overlapping of the small high frequency peaks in figure 4.2 and occupies the same region as the major /g/ peak. Although the overall-average patterns are therefore quite misleading in this case, the learning process has detected this phenomenon and made the right correction. In figure 4.7, we can see that in spite of the large amplitude of the spoiling peak, the network imposes insignificant weights over that region on the /b/ neuron and correctly extracts the features of each phoneme. The average on the four weighted inputs is more ragged as can be seen in figure 4.8. Since we are more interested in how and what acoustic features are enhanced, we'll focus on figure 4.7 in which we can get a more clear picture.

The hard performance on the short-time-average patterns is summarized in table 4.2. Again, we see that /g/ and /k/ are more difficult to recognize because their peak locations lie in between those of /b,p/ or /d,t/. On the other hand, the deferred decision performance is summarized in table 4.3 with penalty of miss is twice as much as that of false alarm. The good performance convinces us that the features picked by the networks that reveal the frequency domain information of the stop consonants are quite reliable cues for their identification even in the multi-speaker case.

Output	threshold	Prob. of miss	Prob. of False Alarm
d	0.49	0.00	0.16
g	0.51	0.15	0.10
b	0.48	0.00	0.00
t	0.48	0.13	0.08
k	0.50	0.11	0.14
p	0.50	0.00	0.05

Table 4.3: The deferred decision performance on stops from four speakers

4.3 Fricative Consonants

The time-average patterns of fricatives from the four speakers are shown in figure 4.9 and 4.10. The corresponding overall-average patterns are shown in figure 4.11 and 4.12. As in the case of stops, we see that the unvoiced patterns are more consistent than their voiced counterparts.

Although the overall-average patterns look quite similar to the ones of single speaker's as shown in figure 3.13 and 3.14, the weighted inputs(figure 4.13 and 4.14) are different. Due to the average operation in generating the overall-average patterns, the minor difference in frequency in the the highest peak of /f,v/ and /th,dh/ can be no longer explicit. The learning algorithm then again conducts the give-and-take rule which forces the /f,v/ give up their high frequency peaks and let their second large peaks dominate. The second large peaks reside in the location between the peaks of /s,z/ and /sh,zh/. A few phoneticians have asserted that the key peak of /f,v/ should lie between those of /s,z/ and /sh,zh/[2], which goes against the assumption of the reso-

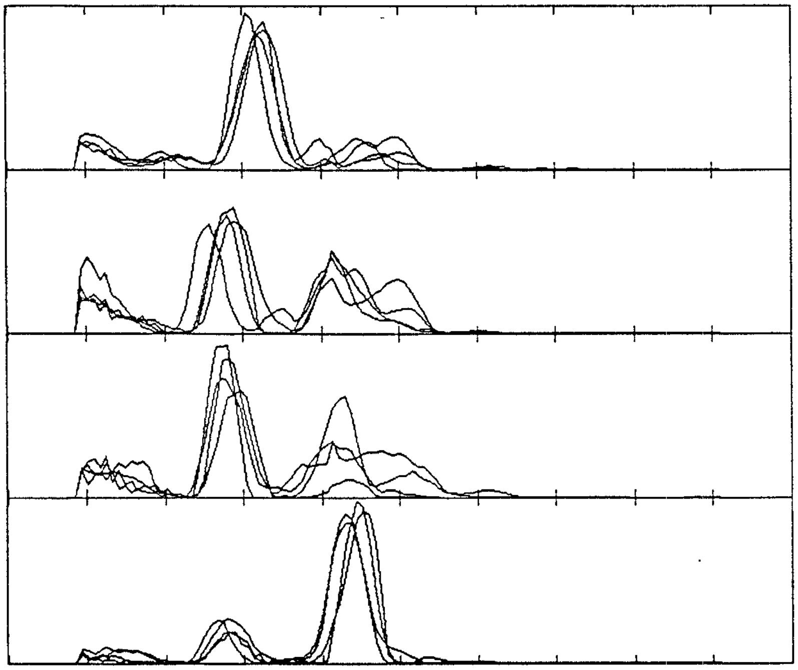

Figure 4.9: The time-average patterns for /s,f,th,sh/ from four speakers

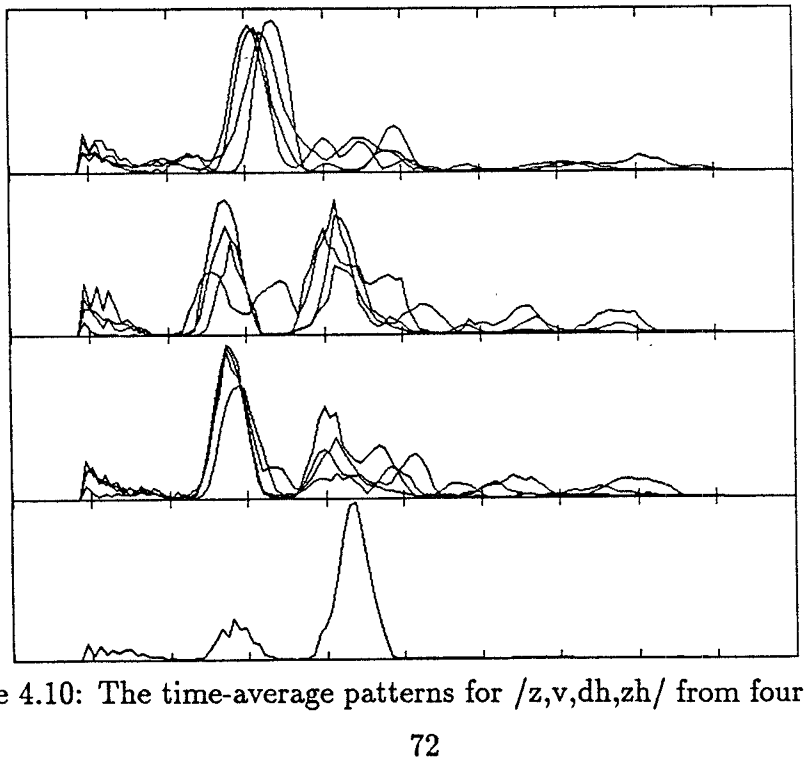

Figure 4.10: The time-average patterns for /z,v,dh,zh/ from four speakers

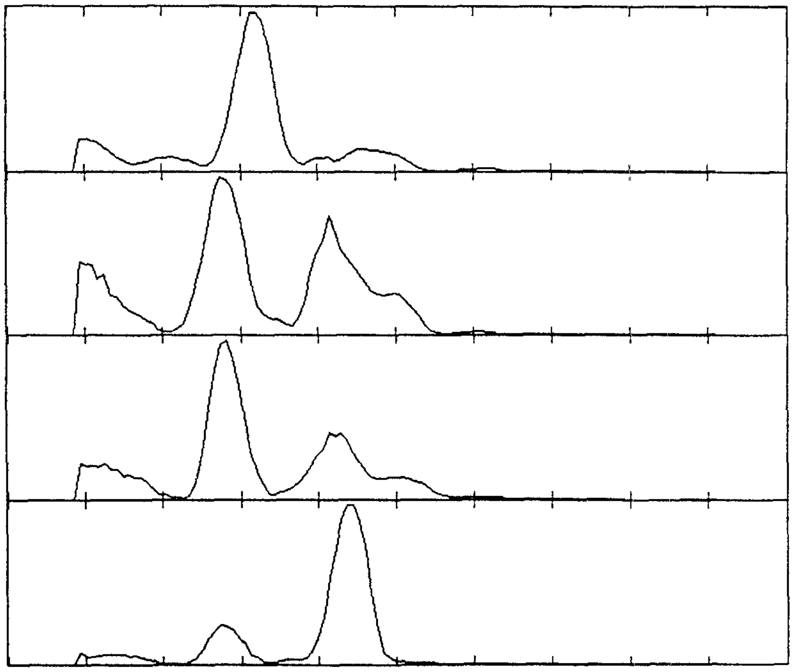

Figure 4.11: The overall-average patterns for /s,f,th,sh/

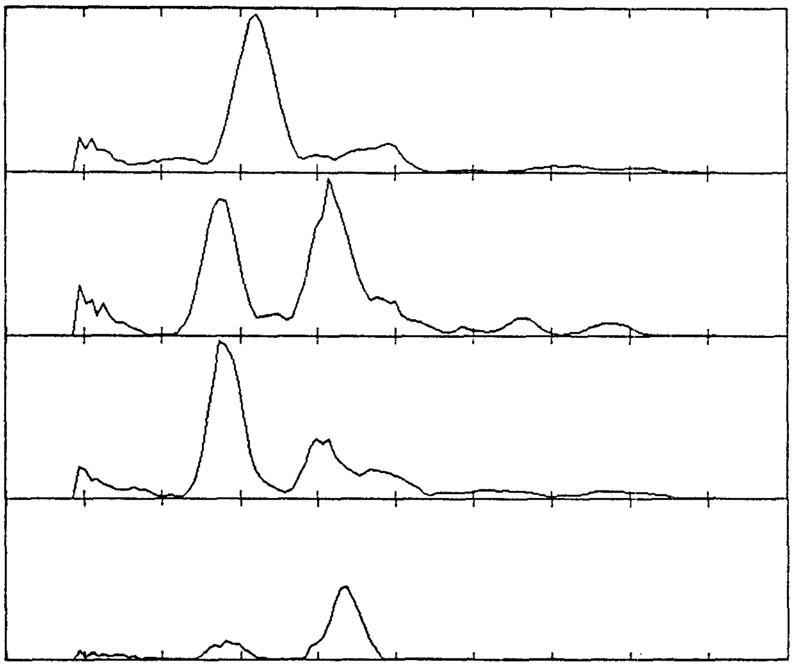

Figure 4.12: The overall-average patterns for /z,v,dh,zh/

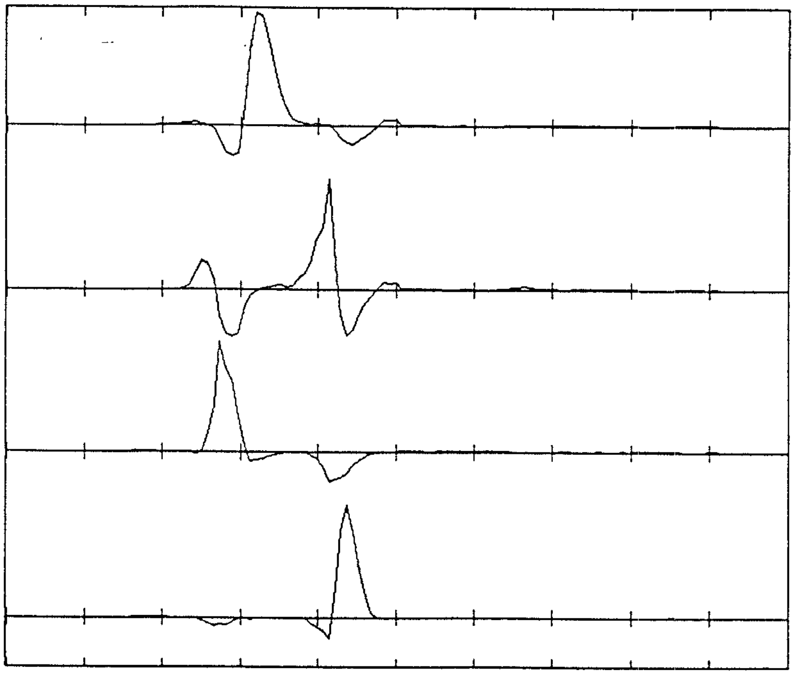

Figure 4.13: The weighted inputs of unvoiced fricatives

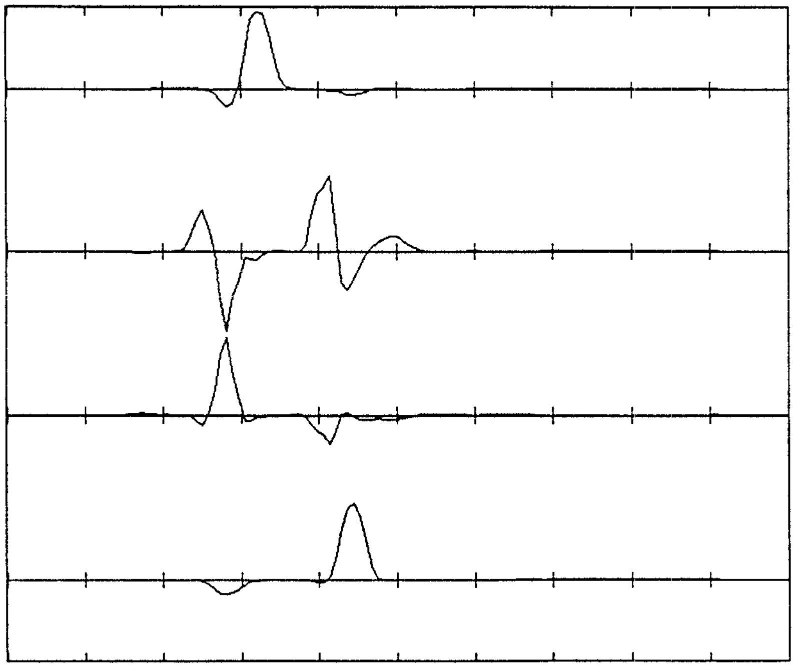

Figure 4.14: The weighted inputs of voiced fricatives

Input	s	f	th	sh	Input	z	v	dh	zh
s	0.00	0.00	0.06	0.00	z	0.00	0.10	0.10	0.00
f	0.05	0.00	0.47	0.00	v	0.10	0.00	0.40	0.20
th	0.00	0.29	0.00	0.00	dh	0.04	0.12	0.00	0.04
sh	0.00	0.05	0.00	0.00	zh	0.00	0.00	0.00	0.00

Table 4.4: The confusion matrix of multi-speaker fricatives networks

nance in the frontal cavity but matches our results here. As a matter of fact, this can be observed even in single speaker's experiments as we pointed out before because we also perform average operation in that case. This is shown in the average on the four speakers' weighted inputs in figure 4.15 and 4.16. Actually, for speakers other than the one we presented, all their second large peaks in /v,f/ dominate, which leads the average of their average weighted inputs looks more like the results of multi-speaker rather than those in figure 3.15 and 3.16. Interesting enough, since their time-average patterns of the fricatives have small deviation, their weighted inputs match the average of their individual weighted inputs very well. Figure 4.16 is nearly no different from figure 4.14.

The hard decision performance for multi-speaker fricative networks is summarized in table 4.4. Here again we see a lot of confusion made in distinguishing /f,v/ from /th,dh/ as in the case for single speaker recognition. It is important to note that the fricatives /f,th/ have very low energy in their waveforms and even the energy of /v/ is generally lower than other voiced sounds. This may be a major reason why we didn't get a very good result for these phonemes because their features are not quite clear in the speactra.

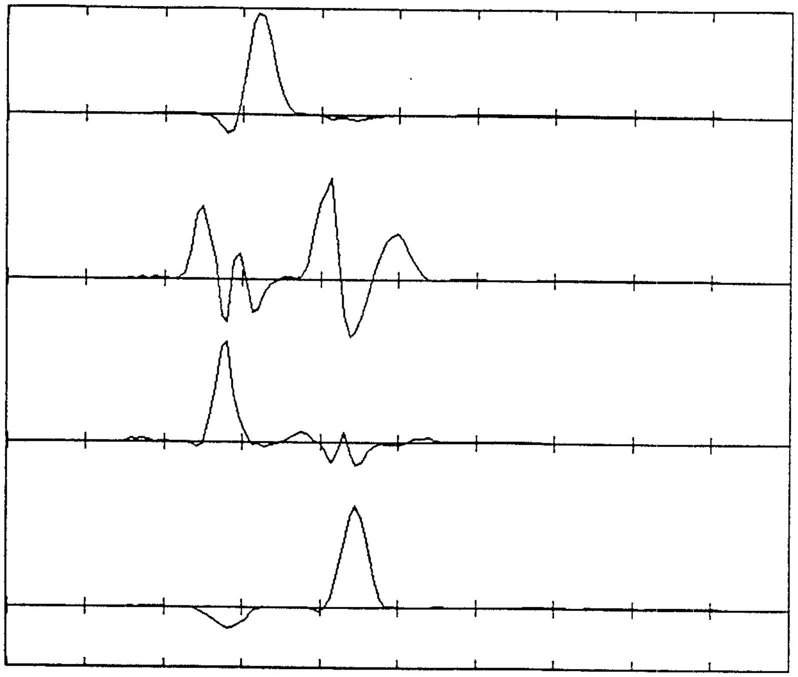

Figure 4.15: The average of four speakers' weighted inputs of /s,f,th,sh/

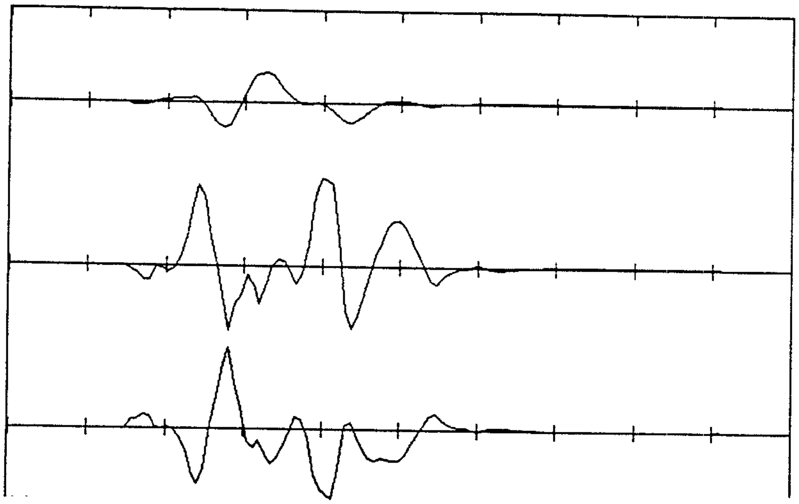

Figure 4.16: The average of four speakers' weighted inputs of /z/,/v/,/dh/

Output	threshold	Prob. of miss	Prob. of False Alarm
s	0.48	0.03	0.00
f	0.43	0.00	0.36
th	0.6	0.43	0.02
sh	0.50	0.00	0.03
z	0.48	0.00	0.07
v	0.46	0.10	0.12
dh	0.58	0.23	0.10
zh	0.63	0.00	0.02

Table 4.5: The deferred decision performance of multi-speaker fricatives networks

Except that, the recognition rate is still reasonable. The deferred decision performance is shown in table 4.5 with the penalty of miss is twice as much as that of false alarm. We are now suffering the high miss rate of /th/ and false alarm rate of /f/. Nonetheless, the overall performance is acceptable.

4.4 Nasals, Vowels and /l/

Generally speaking, to recognize this group of phonemes from several speakers is most challenging because their productions are most sensitive to the excitation of vocal cords and the shapes of vocal tract which are quite speaker dependent. In spite of the highly speaker dependent nature, we can still generally observe their acoustic features discussed in the preceding chapter in their time-average patterns(figure 4.17). For example, we can still see the

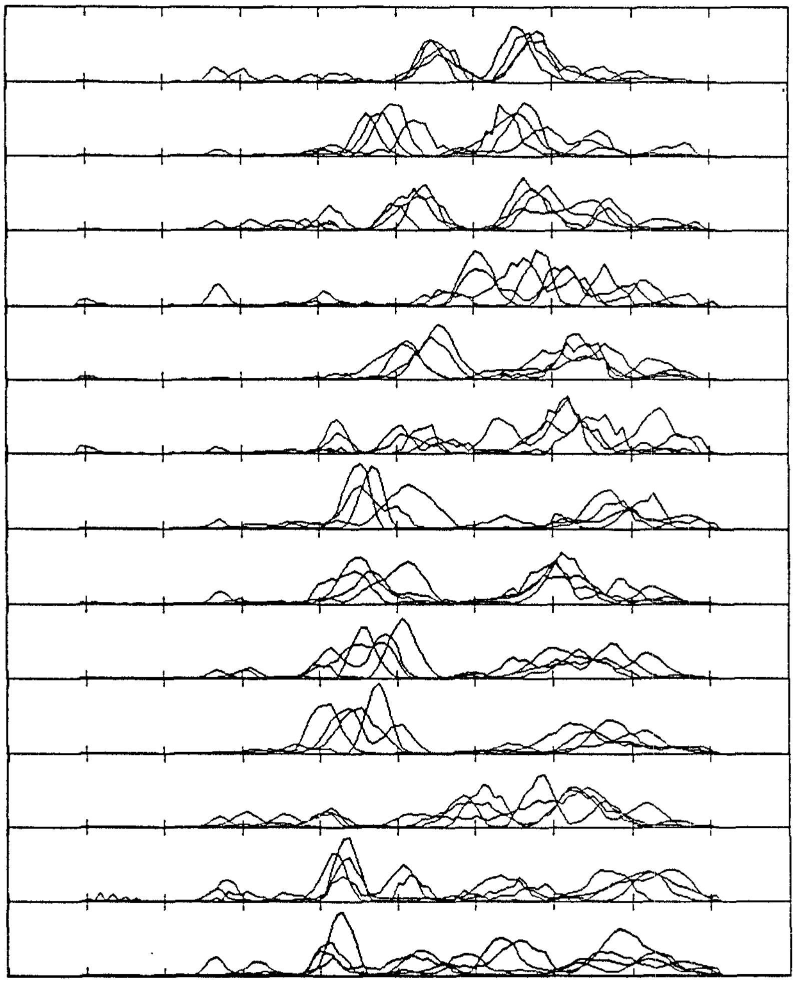

Figure 4.17: Time-average patterns of vowel /aa/, /ae/, /ah/, /ao/, /er/, /uw/, /ux/, /ey/, /ix/, /iy/, /l/, /m/, /n/ from four speakers(from top to bottom)

amplitude ratio varies according to their degree of constriction. The dominant peak also appears in the place according to their tongue hump position. For the nasals, the /n/ peak is usually higher in frequency than /m/, as we have seen in single speaker's case. Therefore, their overall-average patterns in figure 4.18 are quite understandable.

Looking at the weighted inputs in figure 4.19 of the overall-average patterns, we can reach almost the same conclusions as we did in the chapter 3. The close vowels' dominant peaks are enhanced although, for example, /ix/ and /iy/ both have significant outputs in the low frequency. The amplitude ratio of close vowel /er/ is now larger, which is the way it should be. The high frequency component of front vowel /ey/ is also more visible although the low frequency peak is still dominating. However, this dominating peak has moved to some higher frequency and in addition to the vanishing of /er/'s low frequency peak, the largest peak of /l/ is thus released. The open vowels are still playing the give-and-take rule. In comparison with figure 3.20, we see that for this time, /aa/ takes the higher frequency peak and gives up its largest one. /ae,ao/ take their largest peaks as before except the location of /ao/'s peak now shifts a little to the low frequency. As always, the nasals are differentiated from each other by their edges of zeroes.

It's intersting to note that the dominant peak of /uw/ now shifts to a higher frequency region but does not kill the /l/ peak. Furthermore, there are also many phonemes having significant response in the low frequency region but none of them have successfully prevented this /l/ peak from coming out, which is quite different from the case shown in the previous chapter. This can be explained first by examining the overall-average patterns(figure 4.18)

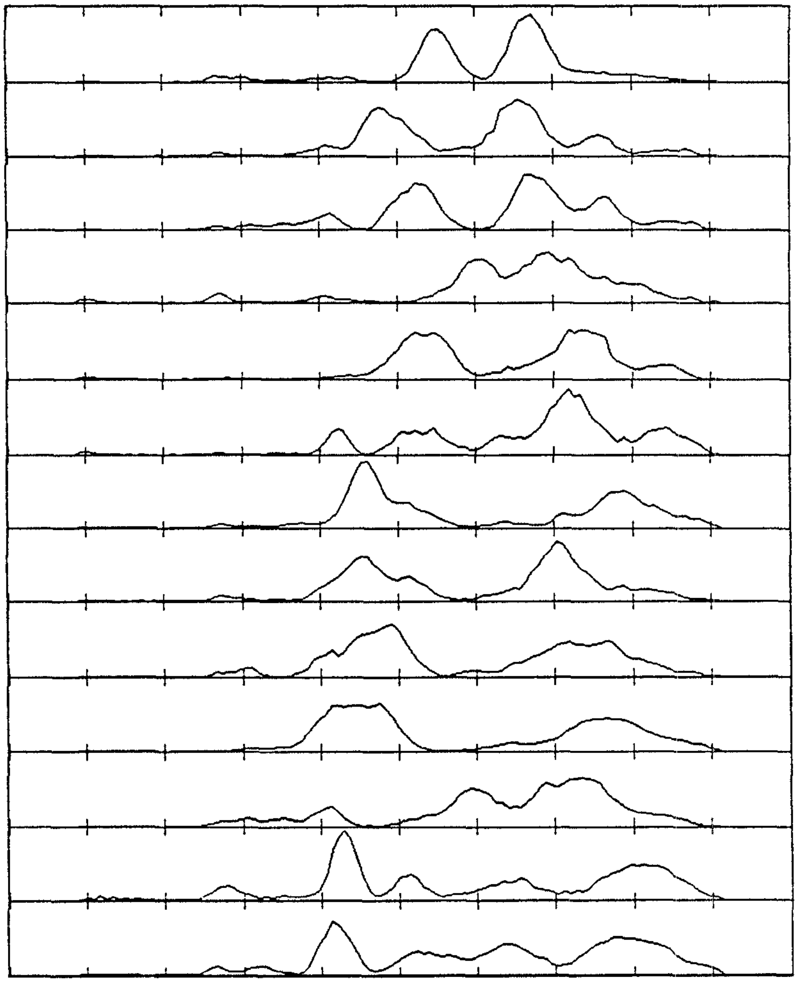

Figure 4.18: Overall-average patterns of vowel /aa/, /ae/, /ah/, /ao/, /er/, /uw/, /ux/, /ey/, /ix/, /iy/, /l/, /m/, /n/ from four speakers(from top to bottom)

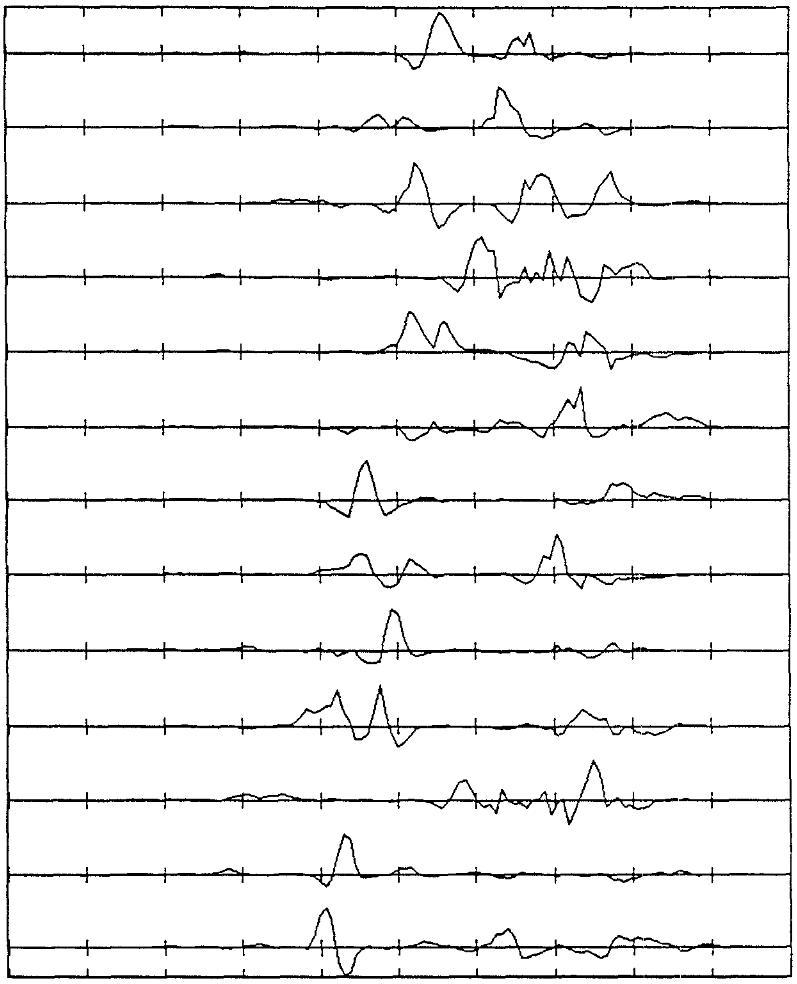

Figure 4.19: Weighted inputs of the vowels

and the time-average patterns of the single speaker in figure 3.19. We can see that in the latter figure, /l/ is the only one that has outputs over the region where its third peak resides. The learning process detects this unique feature of /l/ and puts strong weights on /l/ over this region. According to give-and-take rule, /l/ dominates this region and surrenders elsewhere. But in the multi-speaker case, this band is no longer occupied by /l/ only because now /ao/ also has outputs in this region. Therefore, the /l/ neuron has to find some other way to distinguish /l/ from other phonemes in this group. The learning process finds out that every one that shares the low frequency region with /l/ has almost nothing in the high frequency with the exception that /ao/ also has some large outputs in the high frequency. Therefore, high weights are assigned to the /l/ neuron's high frequency region while inhibiting the same region of /ao/. For better understanding, the weights of the network trained on overall-average patterns are shown in figure 4.20. The above observation reveals how the learning algorithm works and how the features are weighted and extracted.

Figure 4.21 shows the average of weighted inputs patterns from the four individuals. In comparison with figure 4.19 and 3.20, it reveals the deviation of the ways that each network chosen to recognize the phonemes. For example, we can see that the dominant features of /aa/ are now equally distributed between its two peaks. The features of /l/ now look like the hybrid of the two discussed above. What remains the same between the single speaker's and multi-speaker's case, like /ae,m,n/, is still the same in this one. Also important is the overall shapes and even the peak locations do not change significantly. This suggests that even for this very speaker-dependent group

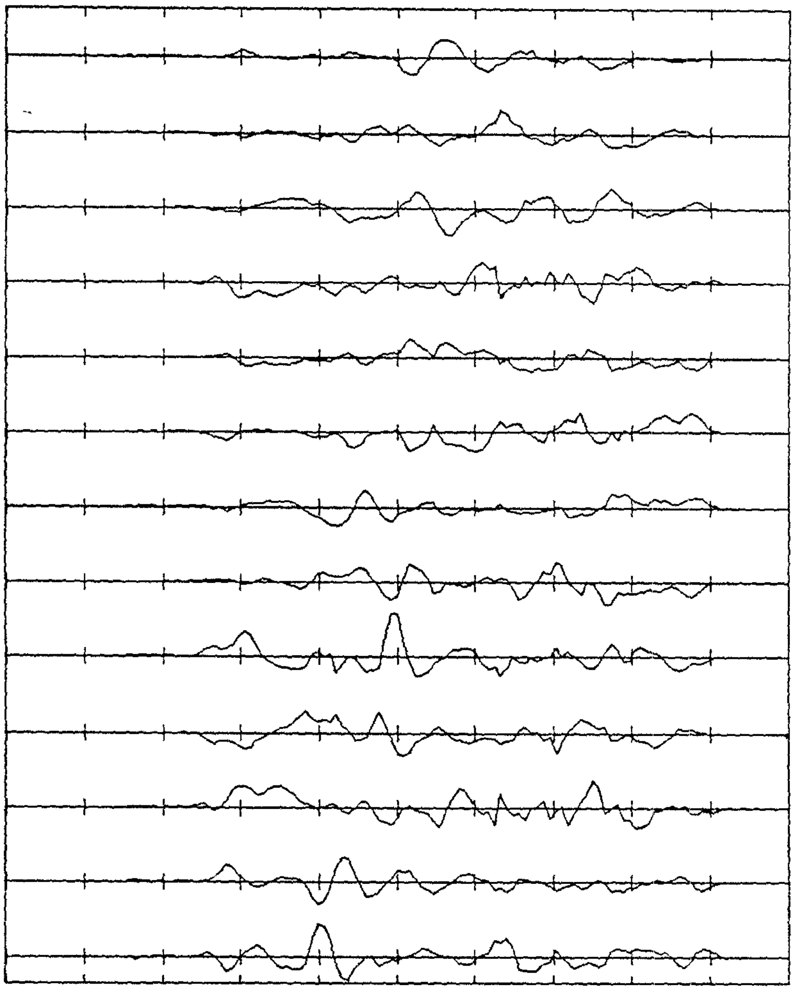

Figure 4.20: The weights of multi-speaker vowels recognition network. They are shown in the same order as in figure 4.18.

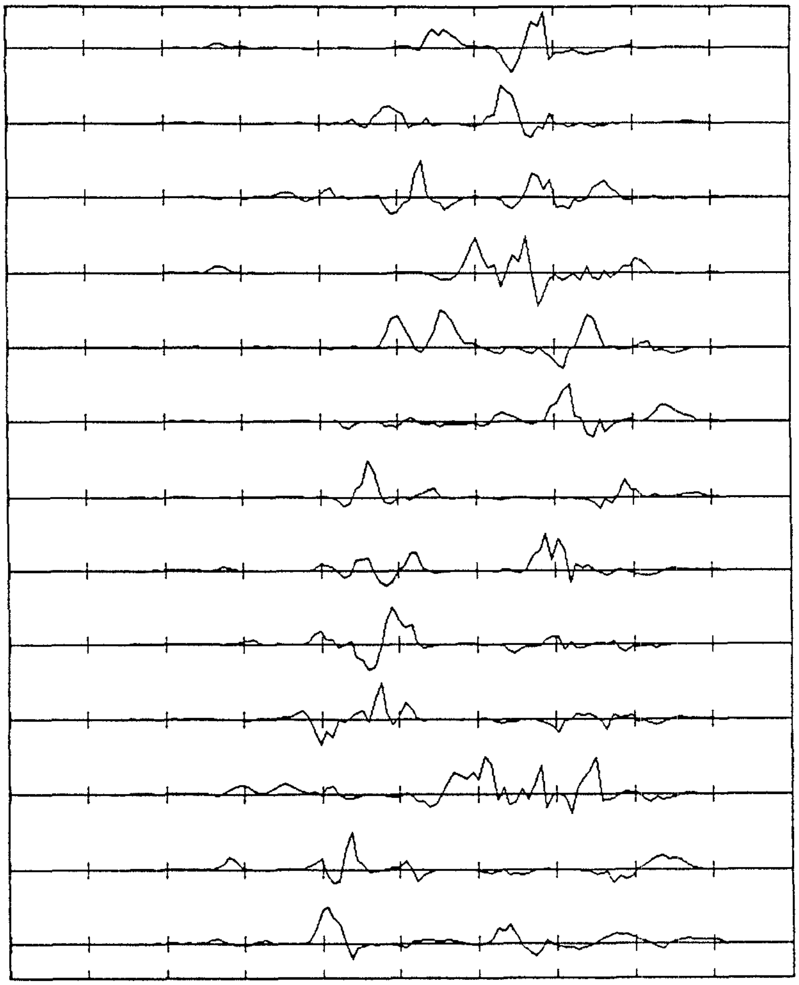

Figure 4.21: The average of four speakers' weighted inputs. They are shown in the same order as in figure 4.18.

	aa	ae	ah	ao	er	uw	ux	ey	ix	iy	l	m	n
aa	.00	.13	.00	.00	.00	.07	.00	.07	.00	.00	.00	.00	.00
ae	.08	.00	.00	.00	.08	.00	.00	.00	.31	.00	.00	.00	.00
ah	.00	.00	.00	.00	.07	.00	.00	.21	.14	.07	.00	.00	.07
ao	.09	.09	.18	.00	.00	.18	.00	.00	.00	.00	.00	.00	.00
er	.13	.00	.13	.00	.00	.07	.00	.00	.13	.07	.00	.00	.00
uw	.12	.12	.12	.00	.00	.00	.00	.25	.12	.00	.00	.00	.00
ux	.00	.00	.10	.00	.00	.00	.00	.00	.30	.00	.00	.00	.00
ey	.00	.05	.00	.00	.00	.14	.05	.00	.05	.10	.05	.00	.00
ix	.00	.00	.00	.00	.06	.06	.12	.12	.00	.06	.00	.06	.00
iy	.00	.00	.00	.00	.04	.04	.09	.13	.13	.00	.00	.22	.09
l	.09	.09	.04	.17	.04	.04	.00	.09	.00	.00	.00	.00	.04
m	.00	.00	.00	.00	.00	.20	.00	.10	.10	.10	.00	.00	.00
n	.07	.04	.00	.04	.00	.04	.07	.00	.00	.04	.07	.22	.00

Table 4.6: The confusion matrix for multi-speaker vowels recognition network

of phonemes, we can still anticipate that there are features in the frequency domain for the phonemes that can be well associated with them and used as reliable cues to identify them.

For our vowel recognition network, we also summarize the hard decision performance in table 4.6 and the deferred decision performance in table 4.7 with the penalty of miss is ten times as much as that of false alarm. Again, our performance is no worse than that by multi-layer networks like the one in [10]. This convinces us that the acoustic features can be used as a good cue to identify vowels.

Output	threshold	Prob. of miss	Prob. of false alarm
aa	0.70	0.27	0.04
ae	0.59	0.23	0.14
ah	0.62	0.36	0.18
ao	0.49	0.18	0.18
er	0.76	0.33	0.07
uw	0.43	0.12	0.25
ux	0.69	0.30	0.06
ey	0.61	0.33	0.17
ix	0.69	0.44	0.14
iy	0.43	0.22	0.27
l	0.54	0.26	0.15
m	0.53	0.10	0.14
n	0.51	0.33	0.18

Table 4.7: The deferred decision performance of multi-speaker vowels recognition network

4.5 Summary and Remarks

In this chapter, we have discussed how our single layer networks recognize phonemes. We have shown how the back propagation algorithm conducts a give-and-take rule to pick up features from the LIN II spectra. We have also shown that with these features, the networks can achieve acceptable performance.

In comparison with the similar experiments done in [10] in which ear model and back propagation algorithm are also used, we have found that our networks achieve better performance. The only difference is we use the time average in both exemplars and test patterns while they used the instantaneous patterns. As we pointed out in the previous chapter, sometimes the acoustic features of the phoneme can be covered or polluted by their context and therefore can not be as clear as it should be. By taking the average, we think the features should be able to outgrow from the noise and the experimental results suggest our guess is not wrong. The same idea also holds for cross-speaker variation.

Traditionally, people working on speech recognition usually focus on the formant frequencies and seldom pay much attention to the relative amplitudes of the formants. As seeing the difference in the LIN II outputs of open and close vowels, we feel that it is important to consider this piece of information in the frequency domain as it reflects the overall spectral shape. Our neural networks model changes the point of interests and achieve satisfying results.

Our model is also ready to be incorporated with higher level of processing such as word recognition. We do believe that a great portion of sounds are recognized and many errors are corrected by the higher level processing unit. We believe so by observing how human processes speech signals. Many differences in speech processing have been noticed among adults and children(especially infants) and experiments have been designed to figure out where these differences come from. Examples can be found in [20] and [21]. It is also known that even mother language can influence human's ability in phoneme recognition. For example, Japanese people usually have trouble in distinguishing /r/ and /l/ because there is no /r/ in their language. All these facts make us believe that a powerful system that combining information of context and semantics may be more urgent and crucial than achieving perfect results in phoneme recognition.

Bibliography

[1] J. B. Allen, "Cochlear Modeling", *IEEE ASSP Magazine*, pp.3-28, January 1985

[2] S. A. Shamma, "The Acoustic Features of Speech Sounds in a Model of Auditory Processing: Vowels and Voiceless Fricatives", *Journal of Phonetics*, Vol. 16, pp.77-91, 1988

[3] S. A. Shamma, "Speech Processing in the Auditory System I: The Representation of Speech Sounds in the Responses of the Auditory Nerve", *Journal of the Acoustical Society of America*, 78(5), pp.1612-1621, November 1985

[4] S. A. Shamma, "Speech Processing in the Auditory System II: Lateral Inhibition and the Central Processing of Speech Evoked Activity in the Auditory Nerve", *Journal of the Acoustical Society of America*, 78(5), pp.1622-1632, November 1985

[5] R. F. Lyon and C. Mead, "An Analog Electronis Cochlea", *IEEE Transaction on ASSP*, Vol. 36, No. 7, pp.1119-1134, July 1988

[6] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning Internal Representation by Error Propagation", in *Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. 1: Foundations.* MIT Press, 1986

[7] J. L. McClelland and D. E. Rumelhart, "Training Hidden Units: The Generalized Delta Rule", *Explorations in Parallel Distributed Processing*, MIT Press, 1988

[8] M. L. Brady, R. Raghavan and J. Slawny, "Back Propagation Fails to Separate Where Perceptrons Succeed", *Special Issue on Neural Networks, IEEE Transactions on CAS*, May 1989

[9] B. Widrow, R. G. Winter and R. A. Baxter, "Layered Neural Nets for Pattern Recognition", *IEEE Transactions on ASSP*, Vol. 36, No. 7, pp.1109-1117, July 1988

[10] H. C. Leung and V. W. Zue, "Some Phonetic Recognition Experiments Using Artificial Neural Nets", *Proceedings of ICASSP*, Vol. 1, April 1988

[11] A. W. 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, March 1989

[12] F. J. Fineda,"Generaliztion of Back Propagation to Recurrent and Higher Order Neural Networks", *American Institue of Physics*, pp.602-611, 1988

[13] R. L. Watrous, "Phoneme Discrimination Using Connectionist Networks", *Journal of the Acoustical Society of America*, May 1989

[14] L. R. Rabiner and R. W. Schafer, "Digital Processing of Speech Signals", *Prentice-Hall*, 1978, Chap. 2

[15] J. L. Flaganan, "Speech Analysis, Synthesis and Perception", *Springer-Verlag(Berlin)*, 1972

[16] R. Lippman, "Introduction to Neural Nets", *IEEE ASSP Magzine*, pp.4-22, Feb. 1987

[17] F. Hadjisgamatiou, "Two Parallel Processing Networks of Associative Memory that Learn Vowels", *Master Thesis, University of Maryland*, May 1987

[18] T. Kohonen, *Self-Organization and Associative Memory*, 2nd Ed., Springer-Verlag, Berlin 1988

[19] T. Kohonen, K. Makrsara, and T. Saramaki, "Phonotopic Maps – Insightful Representation of Phonological Features for Speech Recognition", *Proc. IEEE 7th International Conference on Pattern Recognition*, Montreal, Canada, 1984

[20] P. W. Jusczyk, "Toward a model of the development of speech perception", edited by J. S. Perkell and D. H. Klatt in *Invariance and variablity in speech process*, LEA publisher, 1986

[21] R. M. Dalston, "Acoustic characteristics of English /w,r,l/ spoken correctly by young children and adults", *Journal of the Acousitcal Society of America*, Vol. 57, No. 2, pp.462-469, February 1975