# Speaker-Independent Consonant Classification with Distinctive Features

Giovanni Flammia
School of Computer Science
McGill University, Montréal

revised version September 1991

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of Master of Science

## Abstract

We study the problem of classifying stop and nasal consonants in continuous speech independently of the speaker. We consider some acoustic parameters computed from the auditory spectrogram, and other parameters computed from the speech waveform. The classification algorithm uses a recurrent multi-layer perceptron (MLP) with localized connections. The design of the classifier is motivated by knowledge in phonetics and in pattern recognition. We report experiments for the TIMIT database, using 343 speakers in the training set and 77 different speakers in the test set. Good performance is obtained when many acoustic parameters are fed to the MLP, and when the MLP desired outputs represent context-dependent articulatory features. Classification is performed by Principal Component Analysis of the MLP outputs. Refinement of the design parameters yield increasingly better performance on the test set, ranging from 45% errors for a perceptron to 23.3% errors for the best MLP.

## Résumé

Nous étudions le problème de la classification des consonnes plosives et nasales dans la parole continue, indépendamment du locuteur. Nous étudions des paramètres acoustiques calculés à partir du spectrogramme et d'autres paramètres calculés à partir du signal. L'algorithme de classification utilise un réseau multi-couche (MLP) récurrent à connections localisées. La conception du classificateur est guidée par des connaissances de phonétique et de reconnaissance des formes. Nous faisons rapport de expériences sur la base de données TIMIT, utilisant 343 locuteurs pour l'entraînement et 77 différents pour le test. Une bonne performance est obtenue lorsque les entrées du MLP sont plusieurs paramètres et les sorties représentent des traits articulatoires qui dépendent du contexte. La classification est faite par analyse en composantes principales des sorties du MLP. Des rafinements ont amené graduellement une meilleure performance sur le test, de 45 % d'erreurs pour un réseau sans nœuds cachés à 23.3 % d'erreurs pour le meilleur MLP.

## Acknowledgements

This thesis is the fruit of one year of work at McGill University. During my very enjoyable stay in Montréal, several people have contributed to increasing my knowledge in the multi-disciplinary field of speech communication. First of all, I would like to thank my advisor Renato De Mori, who supported my work with many ideas, suggestions, and continuous encouragement. In particular, he suggested to me to embed knowledge in experimental phonetics in the design of the MLP based classifier. I found him to be very congenial, and I had the opportunity to discuss with him life, the universe and everything. Next, I would like to thank Yoshua Bengio and Ralf Kompe, with whom I have collaborated for extending phonetic classification to phonetic recognition over entire sentences. Yoshua and Ralf have contributed much theoretical and practical advice, as well as, software for starting up the experiments described in this thesis. Gianni Lazzari and Maurizio Omologo shared their expertise during their visit at McGill. The comments of Victor Zue are greatly appreciated as well.

All along my stay in Montréal, Allyn Takahashi was always ready to help whenever I had a problem, either big or small. I was able to pursue my studies at McGill thanks to a Government of Canada Award. I would like to thank John Peck and Doug Pollitt at the Department of Mining. Together, we discovered that recognizing rock types while drilling is similar to recognizing a spoken sentence. I would like to thank also Philippe Wieczorek and Florent Pfirsch, who developed the software for visualizing acoustic parameters. Last, but not least, I would like to thank Katy, for opening a new perspective in my life.

# Contents

# Chapter 1

# Introduction

## 1.1 The Acoustic Modeling Problem

Automatic speech recognition and speech synthesis are open to many challenging applications. Among other examples, one may suggest:

- Providing communication tools for hearing impaired persons.
- Writing of diagnosis and reports without using the typewriter keyboard.
- Translating words and sentences from one language to other.
- Controlling the operation of a mechanical tool by voice.
- Convenient accessing of large information systems.
- Providing a teaching tool for a language course.

On the other hand, a few applications are commercially available at present. In a speaker-independent mode, current systems can recognize continuously spoken sentences with a simple syntax and a small vocabulary[1] In a speaker-dependent mode, current systems are able to recognize sentences with words belonging to a large vocabulary of a certain domain[2], provided that each word is separated by pauses. Continuous speech, speaker independent systems for medium size specialized vocabularies[3] are under devel-

[1] A small vocabulary has between 10 and 100 words. For example: numbers, letters, keywords, and commands

[2] A large vocabulary has at least 10,000 words. Some current applications reach more than 50,000 words Domain examples medicine, business, administration, naval resources

[3] of the order of 1000 words

opment [DARPA]. In order to build better recognition systems, research faces two main problems [Makhoul 90]:

**The Acoustic Modeling Problem** Today, machines utilize an approximate knowledge about the acoustic and phonological rules governing the spoken language. Research is directed towards a more detailed representation of the acoustic signal in terms of its phonetic relevance, and towards a significant integration of phonetic and phonological knowledge in computer models of speech.

**The Language Modeling Problem** Research should develop powerful tools for modeling and integrating information about syntax, semantics, pragmatics and dialogue, so that the machine is able to accept *unconstrained* sentences from the user.

Although we will not deal with the second problem in this thesis, it should be noted that the acoustic signal *per se* should not be the only source of information processed by the system when a sentence is spoken. The understanding of running speech requires an integrated solution to both the above problems, a solution in which each source influences the other. [White 90, Young 90] However, the construction of a reliable baseline acoustic-phonetic decoder is an important step towards the creation of robust recognition systems. In the following, we present a brief overview of the acoustic modeling problem.

Given a sentence to be recognized, most of the current systems model the acoustic signal as a sequence of linguistic units, usually phonemes. At this point, one make the assumption that a spoken sentence is a sequence of elementary sound units, by analogy with a written sentence which is a sequence of letters or graphemes. The problem is then to evaluate what is the most likely sequence of units given the sequence of acoustic observations. The acoustic-to-phonetic decoding problem is difficult, for three main reasons:

**Environment Variability** Other signals might be recorded by the microphone of the speech recognition system. They include background noise and other speakers voices (the so called *cocktail party* effect). Also, the speaker might move with respect to the position of the microphone.

**Speaker Variability** One may deal with many speakers from different geographical regions, and with sentences spoken with different rates. The recognizer has to take into account many sources of variability that influence the acoustic signal: physiology (size and shape of the speech organs), culture (dialect, accent, education), psychology (is the speaker nervous, is he bored?), and physical conditions (does the speaker have a cold?). The pronounciation can also vary depending whether the sentence is read aloud or spoken spontaneously.

**Phonetic Variability** One is trying to decode a *continuous* signal into a *discrete* stream of linguistic units. In particular, the acoustic realization of each sound is not discrete,

and depends on the neighbor sounds and on the intonation contour of the entire sentence. This problem fall under the general term of *coarticulation* [Lindblom 83].

We will not discuss the first problem, which is an entire field of research in itself. The second problem can be made more managable in two ways. The first approach is to design a system that adapts its internal acoustic parameters to one speaker at the time (i.e. *speaker-dependent* recognition). If the goal is *speaker-independent* recognition, the approach is to utilize a very large multi-speaker database for training the acoustic component of the system. These kind of databases are available to the research community for the American-English language. For example, the TIMIT database that we use in this thesis contains thousands of different sentences[4] read in a quiet room by more than 500 speakers belonging to 8 different regions in the United States [Seneff 88b, Zue 90] Similar databases are being created for specialized vocabularies, for spontaneous speech and in Europe and Japan for other languages. In general, the creation and the analysis of large standardized databases is a key factor for improving knowledge about speech communication.

The third problem, phonetic variability, is open and deserves attention. For any naturally spoken sentence, we should expect strong coarticulatory effects in the pronunciation of each sound. Consider the speech signal during two successive time instants $t$ and $t+\delta t$. The movements of the speech articulators[5] do not switch from the *a priori* target configuration representing phoneme $p(t)$ to the following target $p(t+\delta t)$. Instead, the speaker tries to minimize the effort by coarticulating successive sounds into a smooth melodic movement, that may never reach ideal target configurations. As a result, the phoneme $p(t)$ may be realized in many different ways, depending on the preceding and the following phonemes, and on the intonation contour of the entire sentence. A spoken sentence is *not* a discrete sequence of idealized units, but rather a melody of articulatory gestures. The goal of these gestures is not to convey an unambiguous acoustic signal, but rather to communicate an unambiguous semantic and emotional message, that may have many possible acoustic realizations. For this reason, one may argue that the problem is *ill-posed.*

In spite of its simplifying assumptions, the use of a limited number of discrete units is still attractive because it is a parsimonious mean of representing the spoken sentence in a computer system. Spoken words and sentences can be represented by a sequence of phonetic symbols belonging to a small alphabet. These symbols provide a practical interface between the acoustic-phonetic decoder and the lexical access component of the recognition system. At each time interval, the acoustic-phonetic decoder computes a

[4]The TIMIT corpus is a collection of sentences such that every speech sound of the language is adequately represented By *adequately represented* we mean (1) some sentences were designed such that the frequency of occurrence of each phoneme is equal to the a-priori estimated frequency of that phoneme in the spoken language, and (2) other sentences are such that each phoneme appears in many different contexts, in order to represent significant acoustic realizations of coarticulatory effects

[5]such as the vocal folds, the velum , the tongue, the lips and the jaw

probability for the occurrence of each phonetic symbol, given the acoustic evidence. These probabilities can be integrated in the computation of probabilities of occurrence for all the words of the sentence by the other components of the recognition system. For this reason, in the technical literature, phonemes are now a standard unit for evaluating the performance of different decoding algorithms. In order to model adequately acoustic phonetic details and the effect of coarticulation, one may study the use of speech units that are longer than the phoneme. Since the acoustic realization of a phoneme may depend on its left and right context, the use of units made by phonemes plus context, pairs and triplets of phonemes may be considered [Shwartz 85] [Lee 89, pages 91-97]. The number of English phonemes is about 40, and when we consider phonemes in context, phonetic pairs, syllables and phonetic triplets the number of units increases exponentially, and we face the problem of estimating reliable acoustic parameters from a limited database. However, phonemes may be defined by a limited set of distinctive articulatory features describing the characteristics of the vocal tract during the production of each sound [Jakobson 61, Stevens 83]. There are about 25 such features for the American-English language. The problem can be studied by representing phonemes by distinctive features, and by analyzing the relationship between these features and the acoustic signal, and between these features and the word sequence.

## 1.2 Methodology and Objectives

This thesis explores some issues related to the acoustic modeling problem, and is devoted to the subject of acoustic parameter selection and phonetic classification in the framework of speaker-independent continuous speech recognition. To start, we would like to make an extensive use of the available knowledge in experimental Phonetics. The use of domain-specific knowledge to solve the problem of acoustic decoding has been advocated by many researchers (among others, we refer to [Zue 85, Cole 86]). The underlying motivation for such an approach is that the integration of knowledge about the phonetic details of the speech communication process will improve the performance of any recognition system.

There are many possible ways to incorporate phonetic and phonological knowledge in a speech recognition system. One approach is to define appropriate abstract acoustic data representations and to compile probabilistic or deterministic rules for decoding such information in a Artificial Intelligence (AI) system (see, for example [Bush 83, De Mori 87]). This methodology is based on the use of phonetic knowledge, but is difficult to apply to very large tasks because of the complexity of the required AI system. Another approach is to embed knowledge in the structure and the constraints of a statistical decoder based on a hidden Markov model of the speech signal [Shwartz 85, Lee 89, Bartkova 87, Deng 90, Deng 91]. In this thesis, we embed implicit knowledge in the definition of acoustic parameters and in the modeling of phonetic classifiers based on

multilayer perceptions (MLP). This approach is inspired by recently published works [Leung 88, Leung 90, Bengio 90, Bimbot 90, Meng 91]. MLP appear to be a very flexible tool for speech recognition tasks where the distribution of the many observed parameters is difficult to be described by simple linear models, and when it is required to classify sequences of statistically correlated observations rather than only one observation at the time. The structure of MLP classifiers can be designed using domain specific knowledge.

In this thesis, we study some transformations of the speech signal into some acoustic parameters that should carry useful information regarding its phonetic identity, and some specific MLP topologies that should make the best use of these input parameters. The parameters that we consider are inspired by studies in experimental Phonetics and by signal processing strategies. Concerning the design of the MLP classifier, we take a *divide and conquer* or *modular* approach. Different topologies, different input parameters, and different MLP desired output encoding can be used depending on the features to be recognized. It may be convenient to represent the output layer of the MLP by distinctive phonetic features describing the place and manner of articulation and the degree of voicing rather than phonetic units. It is also possible to encode different instances of one distinctive feature depending on the context in which each phoneme is pronounced. The outputs of one or many MLP classifiers may be integrated in time at a higher stage in order to recover the sequence of spoken phonemes. In this thesis, we use Linear Discriminant Analysis and Principal Components to combine the network outputs. We find this a convenient way to interpret the outputs of the MLP in a probabilistic framework. This approach does not require the MLP outputs to estimate probabilities. Instead, the MLP are used to compute a feature vector from the speech signal. The frame-by-frame sequence of MLP outputs, or the compact representation given by the first principal components or linear discriminants can be treated as a sequence of observations for popular recognition algorithms based either on dynamic programming [Silverman 90] or on hidden Markov models [Picone 90]. The appeal of such methodology is twofold. First, one or many MLP can be fed with several heterogeneous acoustic parameters that span a time interval that is longer than one analysis frame. Second, this acoustic information is processed in a non-linear fashion in order to extract relevant phonetic features, without making restrictive assumptions about the underlying distribution of the observed parameters. In this thesis, we will discuss in detail issues relating with the choice of input parameters, the design of the MLP architecture and the appropriate encoding for the output vector of the MLP.

The specific problem addressed by this thesis concerns the distinction between all stop and nasal sounds in American-English. This problem represents a moderate size discrimination task (10 classes). At first sight, this may seem a limited goal with respect to the problem of recognizing all phonemes in continuous speech. On the other hand, it is a significant problem because speaker-independent automatic recognition of these particular sounds in continuous speech is difficult.

Many words may be distinguished or confused by them, for example *dry / try, more*

*/ nore tower / power.* Automatic recognition systems tend to confuse them with other consonants with the same place of articulation, like affricates (for unvoiced stops) and liquids (for voiced stops and nasals). Also, voiced stops and unvoiced stops with the same place of articulation are often confused (like /*p*/ and /*b*/ or /*t*/ and /*d*/). Problems arise because stops may be short in duration, and they may be skipped by the recognizer. Concerning nasals, there may be large differences in the shape of the nasal tract from one speaker to the other, and the acoustic realizations of nasals may vary greatly. Often, nasalization occurs (and is perceived) at the neighboring vowel. It is also difficult to discriminate between nasal sounds with different places of articulation, like /m/ and /n/, because the intensity of a nasal is low, and the vocal tract articulators are relatively free to move, depending on the place of articulation of the neighboring vowel. The spectrum is determined mostly by the nasal tract resonances and antiresonances, independently from the place of articulation. [Glass 86].

Such a problem allows one to complete an extensive comparative study of many experiments in different conditions, with a parsimonious use of computing resources. We will also report some preliminary experiments concerning the integration of such MLP based classifiers into an acoustic-phonetic decoder based on hidden Markov models. This work should be seen as a pilot experimental study.

The thesis is organized as follows. The second chapter reviews some experimental Phonetic studies concerning stop and nasal consonants in American-English. The linguistic concept of distinctive feature and its relationship with the acoustic signal are explained. In chapter 3 we define and justify the acoustic parameters that we take into consideration. Chapter 4 reviews some important statistical preliminaries that are necessary for the understanding of classifiers based on MLP. In particular, we focus our attention on the properties of Principal Component Analysis, Linear Discriminant Analysis and Logistic Regression. Chapter 5 describes how an MLP can be used as a phonetic classifier. The presentation stresses (rather informally) some links between MLP and the statistical procedures presented in the preceding chapter. Chapter 6 reports the recognition experiments on the TIMIT database. In particular, comparative experiments have been run to investigate the proper choice of the input parameters, of the output representation, and of the integration of the network outputs considered as a vector of phonetic features. We also report some other experiments that have been run at our laboratory for extending this work to integrate such MLP classifiers in an acoustic-phonetic decoder for continuous speech. The concluding chapter discusses the results and outlines some of the open problems that we would like to address in the future.

# Chapter 2

# Phonetic Overview

When we want to represent the time-varying speech signal, we look for acoustic parameters that are relatively invariant between speakers, and that carry discriminant information concerning the phonetic identity of the signal. If we represent phonemes by distinctive articulatory features, we look for acoustic cues that discriminate between these features. A promising approach is to use many parameters, based on signal processing strategies *and* phonetic knowledge. This chapter reviews some important studies in Phonetics that are relevant for the recognition problem that we wish to solve. The review describes the articulatory features that we want to recognize and their acoustic correlates. Apart from the papers referenced in the chapter, this review is based on the following books. [Fant 70, Fant 73] provide a comprehensive analytical theory about the production and the acoustic correlates of speech sounds. [Handel 89] is a study about auditory perception. [OShaughn 87] covers both Phonetics and automatic speech recognition. Last but not least, [Borden 84] textbook provides a comprehensive introduction from the linguistic and physiological points of view.

## 2.1 What is Experimental Phonetics?

Research in Phonetics is devoted to the study of speech sounds. Usually, the phonetician designs and evaluates experiments involving the study of natural speech samples and perceptual tests. From the point of view of speech articulation, research is carried out by studying the spectrogram, the waveform and sometimes direct measurements of speech articulator movements[1], muscle activity (EMG), and air pressure, volume and flow. These measurements are recorded during the pronunciation by one or more speakers of different words and sentences. From the point of view of auditory perception, the research usually

[1]Such as X-ray and Nuclear Magnetic Resonance pictures of the vocal tract

involves listening experiments during which some subjects are asked to evaluate different synthetic stimuli according to some criterion. In some experiments, the brain wave activity (EEG) may be recorded. The synthetic stimuli differ in some acoustic parameter such as duration, intensity, phase or spectral quality (i.e., different formant transitions). In general, the study of such experiments focuses on the identification of acoustic cues that are relevant for the production and the perception of different speech sounds. In order to improve the performance of a recognition system, the engineer can design acoustic parameters inspired by the work of the phonetician, aiming to prove that findings that are relevant for the production and the perception of human speech are also relevant for such a technical application as automatic speech recognition. However, the acoustic parameters that are designed by the engineer for a specific application are not meant to be psychological or neuro-physiological plausible. Rather, they should be considered as inspired artificial tools.

## 2.2 Distinctive Articulatory Features

Phoneticians classify speech sounds in space according to distinctive features [Jakobson 61, Stevens 83]. In theory, each sound can be represented by a point in the space. In general, the space is defined by three independent directions: *manner* of articulation, *place* of articulation and *voicing*. An additional distinction is between *vocalic* and *consonant*.

The term vocalic refers to speech sounds that are always pronounced with the vibration of the vocal folds towards an unconstricted vocal tract, while all the consonants involve some kind of vocal tract constriction. The constriction makes consonant sounds transitory and sometimes weaker than vocalic sounds. Vocalic sounds constitute the syllabic nucleus in many languages, including American-English.

The manner of articulation relates to the degree of constriction of the airflow through the vocal tract during the production of each sound. The airflow is stopped by an occlusion in the pronunciation of stop sounds /p,t,k,.../ and in the initial portion of affricates /ch,jh/. For nasal sounds /m,n,ng,.../ , the airflow is directed through the nasal tract by lowering the velum, and the vocal tract is occluded in a way that is similar to the stop sounds. The degree of constriction of the airflow decreases gradually when we consider stop, nasal and fricatives /f,v,z,s,.../, liquids /l,r/, glides /y,w/ and vowels /a,i,u,.../. For the vowels, the manner of articulation refers to the degree of openness of the vocal tract, set by the general position of the jaw and the tongue. The manner ranges from close (or high) /i,u,.../ to mid /ae,er,ao, .../ to open (or low) /ah,aa,.../.

The place of articulation is the location of the more constricted part of the vocal tract, where the upper wall of the vocal tract is closer to the upper part of the tongue. Concerning American-English consonants, the following six categories are ordered from a forward place of articulation (closer to the lips and the teeth) to a backward place

(closer to the velum and the back of the palate): labial /p,b,m/; labio-dental /f,v/; dental /th,dh/; alveolar /s,z,t,d,n,l/; velar-palatal /k,g,sh,jh/; glottal /hh,hv/. The liquid /r/ has a retroflex place of articulation, and is similar to the central vowels /er,axr/. The glides or semi-vowels /y,w/ have a front and back place of articulation, respectively. The place of articulation of the vowels ranges from front /i,ae,eh,.../ to central /er,.../ to back /u,aa,ao,.../.

Voicing refers to the absence or presence of vocal folds vibration. Table 2.1 summarizes the articulatory features that we take into consideration for consonant sounds in American-English. The reader should be aware that many distinctive features may be defined by different linguistic theories and by the desired detail in the description of speech sounds. In defining distinctive features for speech recognition, one keeps in mind two motivations: (1) choosing the minimum number of features that are necessary to distinguish the application vocabulary [Vernooij 89]; (2) choosing those features for which clear acoustic cues can be detected in the speech signal, taking into account the current state of research in Acoustic-Phonetics.

For each phoneme, the table reports its two most common symbols, as well as an example word and the features describing the place and the manner of articulation and the degree of voicing. The allophonic variations defined in the TIMIT acoustic-phonetic corpus are also reported.

In the following sections, we begin our review by considering the production of stop and nasal sounds. Then we consider the acoustic cues related to the manner and the place of articulation of these sounds.

## 2.3 Production of Nasal Consonants

In this section we refer to [Fujimura 62, Fant 70]. During nasal closures the soft palate (velum) is lowered, the airflow passes mostly through the nasal tract, and the oral cavity is occluded at the lips or by the tongue against the palate. The vocal folds provide a periodic excitation to the nasal and the vocal tract. The nasal tract is a large and long resonator of fixed dimensions with a large surface area compared to its volume. Therefore, it contributes to the acoustic spectrum with a single well dampened low frequency resonance. The constricted oral cavity and the large nasal cavity surface absorb much of the energy produced by the vibration of the vocal folds, and create antiresonances in certain frequency ranges. In general, the resonator made by the parallel oral and nasal cavities results in a spectrum with broad band low frequency resonances followed by antiresonances[2]. The

[2]The resonant frequencies of an acoustic tube are inversely proportional to its length, while the energy losses and the bandwidth of the resonances depend on the friction between the air and the walls of the tube and on the heat conduction through the walls, therefore they are proportional to the surface of the walls that is exposed to the air flow. Antiresonances appear when the air flows from the source to more than one path, and this happens for all consonants

| *Ascii* | *IPA* | *Example* | *Manner* | *Place* | *Variation* | *Voicing* |
|---|---|---|---|---|---|---|
| p | p | pop | stop | labial | | no |
| t | t | tie | stop | alveolar | | no |
| k | k | kick | stop | velar | | no |
| b | b | buy | stop | labial | | yes |
| d | d | did | stop | alveolar | | yes |
| g | g | guy | stop | velar | | yes |
| dx | ɾ | ladder | stop | alveolar | flapped | yes |
| m | m | my | nasal | labial | | yes |
| n | n | none | nasal | alveolar | | yes |
| ng | ŋ | king | nasal | velar | | yes |
| nx | ɾ̃ | winner | nasal | alveolar | flapped | yes |
| em | m̩ | bottom | nasal | velar | syllabic | yes |
| en | n̩ | button | nasal | alveolar | syllabic | yes |
| eng | ŋ̩ | washington | nasal | velar | syllabic | yes |
| ch | č | church | affricate | alveo-palatal | | no |
| jh | ǰ | judge | affricate | alveo-palatal | | yes |
| hh | h | hay! | fricative | glottal | aspiration | no |
| hv | ɦ | he | fricative | glottal | aspiration | yes |
| s | s | sister | fricative | alveolar | | no |
| z | z | zoo | fricative | alveolar | | yes |
| th | θ | thief | fricative | dental | | no |
| dh | ð | them | fricative | dental | | yes |
| f | f | fire | fricative | labio-dental | | no |
| v | v | very | fricative | labio-dental | | yes |

Table 2.1: Articulatory classification of American-English consonants.

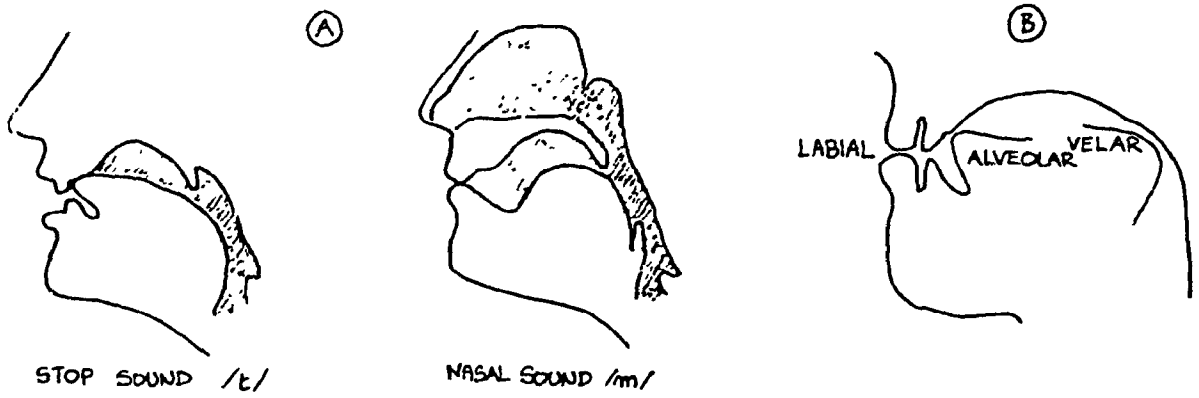



Figure 2.1: A: Schematic X-ray tracings of nasal and stop sounds. Adapted from [Fant 70]. B: Position of the place of constriction in the vocal tract for labial, alveolar and velar sounds. Adapted from [Borden 84].

place of constriction in the oral cavity differs for the three nasal sounds. For the labial /m/ the lips are touching, for the alveolar /n/ the tip of the tongue touches the front of the palate close to the teeth, and for the velar /ng/ the back of the tongue touches the back of the palate close to the velum. Therefore, the length of the oral cavity resonator decreases progressively for /m,n,ng/ and the frequency ranges for the resonances and the antiresonances increase accordingly (see figure 2.4). Nasal production is characterized by low pressure in the vocal tract above the glottis and behind the closure, and articulators that are not required by the nasal sound are free to move. For example, the jaw is free to move to or stay in the position required for the pronunciation of a neighboring sound. Also, if the velum does not contrast for the pronounciation of a neighboring vowel, it can be lowered during the pronounciation of the vowel.

## 2.4 Production of Stop Consonants

In this section we refer to [Fant 70, Fant 73]. Stops consist of three events: the closure, the burst release and sometimes the aspiration. During the closure the glottis is open. The air coming from the lungs increases the pressure in the oral cavity above in the vocal tract the glottis. Since the oral cavity is occluded completely at the place of articulation, it expands until it suddenly opens, releasing the air at the constriction. No air flows

through the nasal tract. During the release, the vocal tract is excited primarely at the constriction which contributes with a turbulent noise source. The constriction of stop (and nasal) sounds is at the lips for labial /p,b/ , at the hard palate for alveolar /t,d/ , and at the back of the palate for velar /k,g/.

Seen from the glottis, the production of stop sounds is a complex event. During the initial part of the closure, the vocal folds may or may not vibrate. If the preceding sound is voiced, and the stop is also voiced, the vocal folds vibrate. In such a case, some low frequency energy is dissipated through the walls of the vocal tract, while the occlusion causes the build up of a certain amount of air pressure (a lower pressure than for an unvoiced stop). If the preceding sound is voiced, and the stop is unvoiced, the vocal folds may continue to vibrate at the beginning of the closure, but they do not vibrate immediately before the release of the burst, allowing the pressure above the glottis to increase significantly.

Right after the release of an unvoiced stop, the vocal folds adduct (without vibrating) creating a turbulent noise source. The resulting sound is called aspiration. During the release of a voiced stop, either the vocal folds were already vibrating, or they will start vibrating sooner than during an unvoiced stop release (with a shorter or null aspiration phase). In general, in American-English there is a significant delay between the burst release and the voicing onset, if we consider an unvoiced stop followed by a vowel compared to a voiced stop.

It should be noted that the distance between burst release and voice onset (VOT) also depends on the context. For example, when a unvoiced stop is preceded by a fricative (like in *spin* vs. *pin*), this distance is shorter and is similar to the distance for the voiced cognate /b/.

The frequency location of the high frequency broad band resonances during the burst release depend on the place of the constriction of the vocal tract, i.e. the place of articulation determined by the position of the lips or the position of the tongue.

Labial resonances are associated with one long back cavity that is open on one end (the glottis) and closed on the other end (the lips). The main resonance is a relatively low second formant.

Alveolar resonances depend on two different cavities, a back cavity shorter than the labial one and constricted between the palate and the teeth, and a short front cavity that is open at the lips. The main resonances depend on the length of the back cavity. As the length of the back cavity decreases from the labial to the alveolar configuration, the second formant peak increases.

Velar resonances are influenced by the place of articulation of the neighboring vowel, because the place of the constriction is forward in the palate if the vowel is front, and towards the back if the vowel is non front. Therefore the front and back cavity resonances may vary. If the vowel is front ( the occlusion is forward) the second formant ( determined

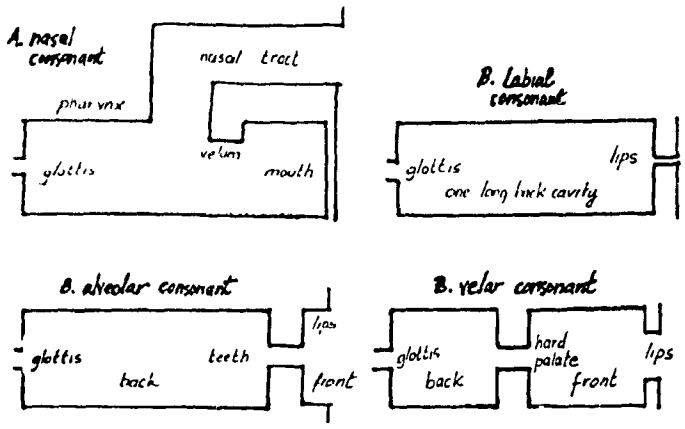



Figure 2.2: Simplified acoustic tube models. A: Nasal sounds. The nasal and the vocal tract are coupled. B: Labial, alveolar and velar configurations of the vocal tract. Adapted from [Fant 70].

---

by the longer back cavity) is close to the third formant peak ( determined by the shorter front cavity). If the vowel is back ( the occlusion is backwards) the second formant is determined by the front cavity and is further away from the third formant determined by the back cavity.

## 2.5 Acoustic Correlates of Stop and Nasal Sounds

### 2.5.1 Perceptual Cues

Absolute spectral properties such as formant peak locations, their transitions from or to the consonant and the relative duration of acoustic events are the main cues for the perception of stop and nasal sounds, although it is a matter of current research how these cues are integrated or alternated by our perceptual system, depending on the context in which the consonant is perceived [Handel 89].

Perception of nasality depends on the detection of the low frequency murmur and on the decrease in intensity with respect to the neighboring vowel. Stop perception depends mainly on duration and energy cues. A silence followed by a short spectral transition towards a steady state vowel is perceived as a stop rather than another consonant with the same place of articulation. The distance in time between the end of the silence and

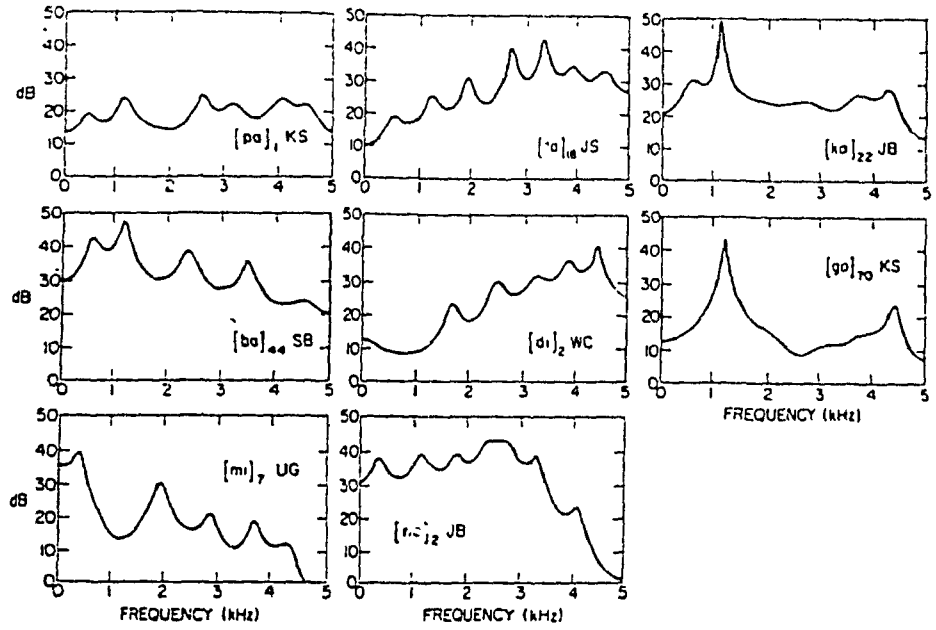



Figure 2.3: Examples of spectra at the release of stop and nasal sounds. The spectra are smoothed by a linear prediction algorithm. From [Stevens 80].

---

the voice onset (VOT) is a strong perceptual cue for voicing, although depending on the context, the detection of voicing during the preceding closure, as well as the burst amplitude, may integrate or substitute VOT in voicing perception.

The perception of the place of articulation depends on the frequency location of the major peaks in the spectrum, but also on the difference between these locations and the resonant frequencies of the neighboring vowel. Several acoustic cues might be integrated over time in order to perceive the exact place of articulation.

In general, different acoustic stimuli might be perceived as belonging to the same category, as long as some fundamental features are maintained, either in the spectral or in the time domain. For vowels, these features might be the distances between successive perceived formant peak locations [Chistovich 79], and for consonants like stops it might be the gross shape of the spectrum [Blumstein 80]. Since the gross shape of the spectrum depends from the distance between formant peaks, consonant and vowel perceptual theories are consistent. This effect is known as *categorical perception* and is much debated [Handel 89].

### 2.5.2 Outline of the Acoustic Cues

In the following we outline the acoustic correlates for the 10 stop and nasal sounds.

**Manner of Articulation** The acoustic cues correlated with the *manner of articulation* are:

- Stops are represented by a sequence of distinct acoustic events visible in the speech signal (closure, burst, aspiration). As a consequence, there are abrupt changes in the spectrum and in the amplitude of speech waveform. These changes are not so abrupt in the production and the perception of other consonants sharing the same place of articulation, such as the nasals, the liquids and the fricatives [Stevens 75, Stevens 81].

- The spectrum of a nasal murmur shows a low broad band first formant peak. The spectrum changes slowly compared to other consonants, while the speech waveform has low energy. The nasal resonance may appear during the pronunciation of the neighboring vowel.
  [Fujimura 62, Mermelstein 77, Glass 86]

**Voicing for Stop Phonemes** The acoustic cues for voicing of stop phonemes are multiple, and each can be present or not depending on the voicing manner of the contextual sounds. [Stevens 74] The acoustic cues for the distinction of *voicing* are:

- The movement of the first formant peak between the consonant and the neighboring vowel tend to be more pronounced for voiced sounds.

- The presence of some energy in the low frequency bands, visible during the closure, if the vocal folds do not stop vibrating during the closure.

- The distance between the burst release and the voice onset (VOT) is usually shorter for voiced stop than for unvoiced stop.

- The burst amplitude is usually greater for unvoiced stops.

**Place of Articulation for Stop Phonemes** There are two main acoustic cues for the distinction of the *place of articulation* of stop consonants.

- The general shape of the spectrum during the burst release, which is determined by the relative distance between the frequencies of resonance of the tubes constituting the vocal tract [Blumstein 79, Blumstein 80, Stevens 81].

- The movement of the second formant peak[3] between the consonant and the neighboring vowel. This cue is dependent on the place of articulation of the neighbouring

[3] Either rising or falling in frequency during a short interval of time

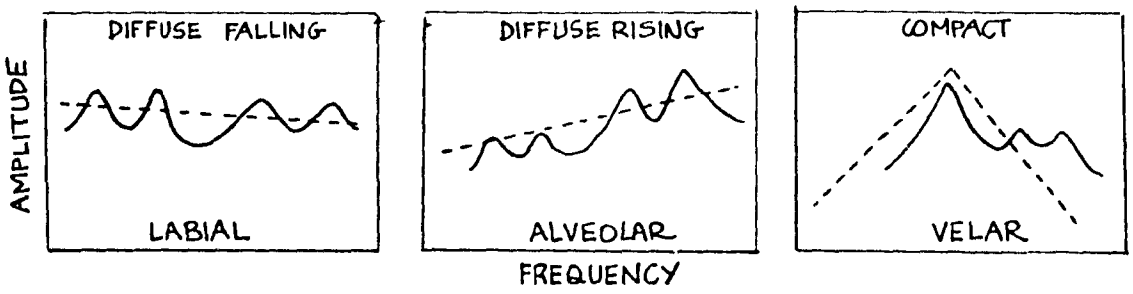



Figure 2.4: Spectral templates for labial, alveolar and velar bursts. Adapted from [Blumstein 81].

vowel. It is not clear what is the exact perceptual relationship between this cue and the above one. [KewleyP 82, KewleyP 83, Kuroski 84, Suomi 85]. Recently, [Nathan 91] classified stops in VC syllables in a $\Delta F_2/F_2$ feature space, where $F_2$ is the 2nd formant measured at the closure of the glottis prior to release and $\Delta F_2$ refer to the 2nd formant transition from the vowel to the closure.

**Place of Articulation for Nasal Sounds** The acoustic cues for the place of articulation of nasal sounds are:

- The second formant transitions between the consonant and the vowel, in analogy with the stop sounds. This one appear to be the primary acoustic cue.
- The position of the first formant peak increases gradually from the labial /m/ to the velar /ng/. This one seems to be a secondary cue (it is less evident when considering several different speakers).

### 2.5.3 Details about the Acoustic Cues

Consider now in detail the acoustic correlates for the unvoiced stop */p,t,k/* [Fant 73, Blumstein 79, Stevens 81, KewleyP 82, KewleyP 83]. The spectrum of the labial */p/* is

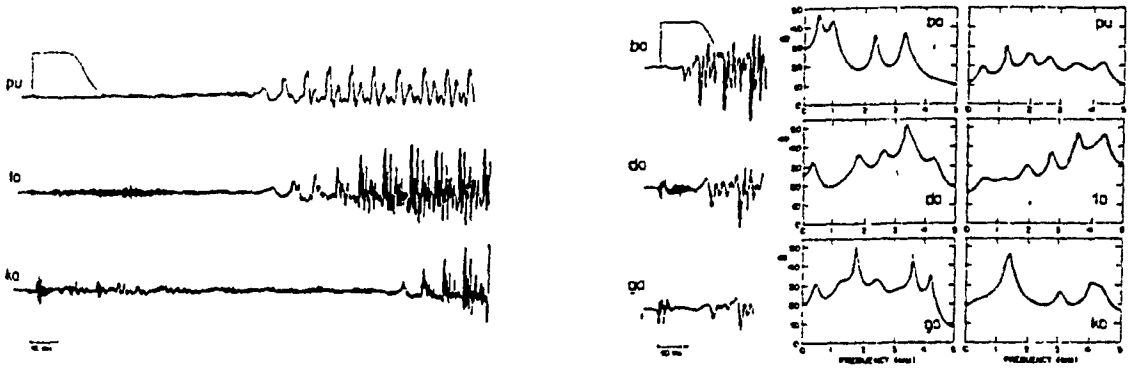



Figure 2.5: Examples of waveforms and spectra at the release of stop sounds. The spectra are smoothed by a linear prediction algorithm. From [Blumstein 79].

spread out or diffuse at the burst, with a falling or flat slope towards higher frequencies. The burst amplitude is generally smaller than the high frequencies amplitude of the neighboring vowel. The second formant peak is a resonance of the long back cavity of the vocal tract and is lower (in frequency) than the second formant peak of the vowel, and therefore it rises from the burst to the vowel onset if the stop precedes the vowel, and falls if the stop follows the vowel.

The spectrum of the alveolar */t/* is diffuse and rising at the burst. The amplitude of the burst is as high or higher than the amplitude in the following vowel, The second formant peak is a resonance of the back cavity of the vocal tract. Since the back cavity is shorter than for the labial */p/*, this resonance is at a higher frequency, and it rises (falls) moderately from the burst to the following front (non front) vowel. These movements will be inverted if the stop follows the vowel.

The spectrum of the velar */k/* depends on the place of articulation of the following vowel. If the vowel is front, the spectrum is concentrated or compact around a high frequency broad band peak determined by two close resonances of the back and front cavities of the vocal tract. This peak is located at a frequency close to and higher than the second formant of the neighbouring vowel. If the vowel is non front, the constriction is backwards, and the second formant peak is at a lower frequency, more distant from the third formant peak. The second formant is generally falling (rising) from the burst to the following (preceding) vowel. The burst and the aspiration of velar stops are often longer

in duration than for the other stops, and the spectrum changes in a slower fashion

The time-varying spectrum of the voiced stops */b,d,g/* and the nasals */m,n,ng/* obeys the same formant transition rules discriminating labials */b,m/*, alveolars */d,n/* and velars */g,ng/* [Borden 84]. However, the voiced burst is shorter and with less energy than the burst of unvoiced stops, and the aspiration is absent, while nasal murmurs correspond to energy dips.

It is important to note that these findings always refer to phonemes carefully pronounced in *isolated words*. In this research we are addressing the problem of *continuous speech*. Therefore, we expect strong coarticulatory effects between the consonant and the neighboring vowel, and a great variability in the duration and the amplitude of each sound.

### 2.5.4 Allophonic Variations in American-English

We have to consider several allophonic variations that have been labeled on the TIMIT database. The voiced alveolar flapped variation */dx/* is considered here as a distinct stop phoneme. Indeed, the acoustic realization of this sound is different from either */t/* or */d/* when pronounced between two vowels , due to the contact between the tongue tip and the alveolar ridge during the closure. This contact produces energy losses and increases the bandwidth of the formant peaks. The spectrogram of the flapped */dx/* looks somewhat like a short and weak voiced fricative [Zue 79]. Nasals can also be flapped (it is the case for */nx/*), but the most common variation is the syllabic form */em,en,eng/* where there is no evidence of a boundary between vowel and nasal on the spectrogram, the velum being lowered during the vowel pronunciation. In this case the spectrum looks like a very long nasal. For classification purposes, we merge the few syllabic nasal allophones with their respective non-syllabic labels. We also merge the nasal flap with */n/* because there is not as much difference between the two acoustic realizations, compared to the difference between flapped and non flapped stop realizations. This convention is used by other published work on the TIMIT database [Lee 89].

## 2.6 Summary

In summary, in order to distinguish stop and nasal consonants, we can take into consideration the temporal evolution of several acoustic cues, ranging from the fine spectral detail (e.g. formant peak trajectories, distance between resonant peaks) to the broad spectral shape (e.g. slope of the high frequency burst, major peaks and valleys location on the frequency axis, low and high frequency energy variations on the time axis). These acoustic cues are dependent on the left and right context, in particular on the place of articulation of the neighboring vowel and the voicing manner of the preceding sound, especially for

features such as velar, nasal, and voicing. In particular, nasal and velar articulations put loose constraints on the position of some speech articulators. It is then appropriate to assume different acoustic realizations and different allophonic labels for each phonetic class. In the next chapter we will define analytically some acoustic parameters that will be used for the recognition of stop and nasal sounds.

# Chapter 3

# Acoustic Parameters

In the past years, research has been devoted to the search of adequate parameters in order to improve recognition scores. Usually, the input parameters for speech recognition systems are sequences of feature vectors representing the spectrum and the energy of the speech signal in successive short-term analysis windows. For hidden Markov model based algorithms, performance has been shown to improve when time differential parameters are added to the standard spectrum based parameters[Lee 89]. For this thesis, we investigate the use of some other parameters in conjunction with the spectrogram. This research is motivated by two reasons. First, the neural networks that we will use as phonetic classifiers are able to handle a rather large number of heterogeneous and/or correlated input parameters without trouble, so we are not seriously limited by the number and the nature of input features to choose. Second, we look for acoustic parameters that are inspired by knowledge in experimental phonetics, and we wonder whether using more of these parameters will improve the recognition performance.

In the following sections we review the acoustic parameters that will be used in this thesis. We will not consider any parameter that rely explicitely on a *a priori* segmentation of the speech signal. In other words, no effort is made to detect automatically specific acoustic events, such as the burst or the vowel onset. Some parameters are derived from the auditory spectrogram, and other are directly computed from the speech waveform. Chapter 6 will report about comparative experiments using different sets of acoustic parameters.

## 3.1 Review of Spectral Analysis for Speech Recognition

First of all, we need a spectral estimation of the speech signal. For speech processing, popular spectral estimation methods are based on the Discrete Fourier Transform (DFT), homomorphic or cepstral analysis, linear prediction, and time-domain filtering according to an analytical model of the ear.[1] While time-domain filtering is performed sample by sample, all the other methods require the application of a short-term analysis window to the speech signal. The analysis step can be either fixed or adapted to the measured pitch of the acoustic signal.

Linear prediction [Makhoul 75] models the speech signal as the output of an all-pole filter excited by a sequence of pulses of short duration. These pulses are either periodic (for voiced sounds) or randomly distributed (for unvoiced ones). This technique is used widely in coding and synthesis of speech, and in some recognition systems. We have seen in the preceding chapter that consonant sounds, in particular nasals, are represented by resonances (poles) as well as anti-resonances (zeros), so at first sight we would not use an all-pole model. On the other hand, it is well known that any transfer function with poles and zeros can be modeled with a high enough number of poles. The problem is then to set the appropriate number of poles before the acoustic analysis is performed. However, the location of some formant peaks might not correspond with the pole location estimated by linear prediction[Makhoul 75]. To solve this problem, a more accurate type of all-pole modeling has been proposed [ElJaroudi 91]. Another problem is that the simple source-filter model is inapproriate when the vocal tract configuration changes rapidly from a stop consonant to a vowel. To solve this other problem [Nathan 90] uses a short-term pitch synchronous analysis to estimate the parameters of the all-pole filter. Recently, in [Hermansky 90] linear prediction has been applied to approximate the auditory spectrum rather than the standard spectrum. We decided not to investigate further the use of linear prediction in this thesis, considering also that it requires many more computations than the DFT based method.

The cepstrum is a linear transformation of the spectrum. More precisely, cepstral coefficients can obtained by projecting the spectral coefficients on a set of orthogonal cosine functions. The first cepstral coefficients are related to the global shape of the spectrum, like the tilts towards higher or lower frequencies. Cepstral coefficients with a higher index are related to details of the spectrum, and have a small variance. For speech recognition, it is possible to drop the coefficients with the smallest variance without degrading the performance [Davis 80]. Therefore, cepstral coefficients provide a compact set of uncorrelated parameters. For this reason, the use of cepstral coefficients is popular

[1] The interested reader should refer to the tutorial by [Rabiner 78] concerning digital representations of speech signals, and the *Journal of Phonetics* special issue dedicated to the subject of computational models of auditory speech processing [Greenberg 88]

for statistical phonetic decoders, because the complexity of such algorithms is negatively affected by the number and the correlation between the input parameters. For speech recognition applications based on artificial neural networks, recent published works showed that the performance did not change significantly in going from the FFT based spectral representation to the cepstral representation [Robinson 90b, Meng 91].

In general, we prefer spectral analysis to cepstral analysis, because the latter method provide parameters that are difficult to interpret visually, except for the first two. Comparing different spectral estimations, we find the method based on time-domain filtering attractive because the signal processing does not suffer from loss of information due to the fixed windowing of the speech signal independently from the variations of the fundamental period and of the duration of acoustic events. Usually, a bank of band-pass FIR (finite impulse response) filters is applied to the speech signal sample by sample, and no windowing is required. In addition, non linear computations can be applied to the outputs of the filters in order to represent adequately relevant spectral variations in time and frequency. This non linear behavior is inspired by neurophysiological studies [Greenberg 88]. [Meng 91] showed that a particular model [Seneff 88a] outperformed FFT based spectral and cepstral analysis, especially in a noisy environment, in a vowel recognition task. Another comparative study [Robinson 90b] based on another ear model, did not show much improvement by using this last method. It should be noted that for this latter study a very large analysis window length and analysis step (32 msec. and 16 msec. respectively) has been applied to the speech signal for all the experiments, *including* for the filter bank spectral analysis. This fact might have biased the results towards the same average performance rate.

A method based on an ear model is computationally very expensive unless directly implemented on digital signal processors. This is why in this thesis we settled for a simple and computationally inexpensive method based on the DFT, that will be described in the next section.

## 3.2 FFT Based Spectral Analysis

Every 5 msec a Fast Fourier Transform of 20 msec length is computed from the Hamming windowed and pre-emphasized (with a factor of 0.98) speech signal. A smoothed window of 20 msec. represents a compromise value, allowing for enough resolution in the frequency domain to track formant peaks in a large range of the fundamental period for male and female speakers, and enough resolution in the time domain to avoid missing short acoustic events. In a few cases, such a window may contain two acoustic events, such as a very short plosive burst followed by the initial portion of a vowel. In these cases, the spectrum will contain some information regarding both events, and coarticulation effects may be emphasized. Some other times, the average amplitude of the spectrum of two successive

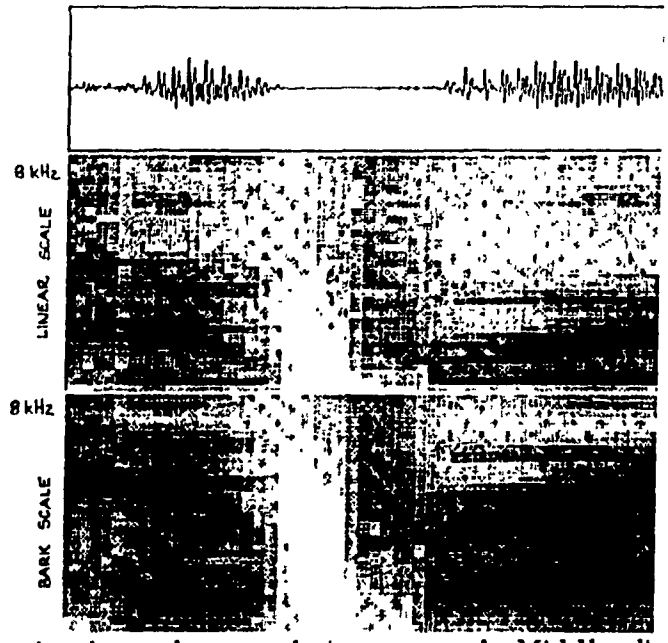



Figure 3.1: Top: signal. spoken word: 'recuperate'. Middle: linear scale spectrogram, Bottom : Bark scaled spectrogram

frames will vary because the analysis step is not synchronous to the fundamental period of the analyzed speech segment. For these limitations, we regard the FFT based spectrum as a baseline acoustic analysis that can be certainly improved at the expenses of a higher computational load.

The power spectrum is then smoothed by 32 overlapping triangular filters equally spaced on the auditory (Bark) scale from 100 to 7000 Hz. This scale compresses logarithmically the frequencies above 1200 Hz, according to the following formulae [Zwicker 80, Seneff 88a]:

$$\begin{aligned} f < 500Hz \quad & B(f) = 0.01f \\ f < 1220Hz \quad & B(f) = 0.007f + 1.5 \\ f \geq 1220Hz \quad & B(f) = 6\log f - 32.6 \end{aligned} \tag{3.1}$$

From the formulae we see that spectral information at low frequencies has a higher resolution than at high frequencies. The sequence of 32 smoothed spectral coefficients $X$ (i.e. the spectrogram computed on the auditory scale) is the basic and most important input parameter set for the phonetic classifier used in this thesis. Figure 3.1 illustrates the differences between the linear scale FFT and the Bark scaled spectrogram.

## 3.3 Spectrogram Based Parameters

In the following, we describe some other parameters derived from the spectrogram. We hope that these parameters will improve the performance of the classifier. Comparative experiments will be reported in Chapter 6.

Before introducing some differential parameters derived from the spectrogram $X(f,t)$ it is useful to remind the definition of the linear regression coefficient $R$. Consider a function $f(n)$ measured at some discrete samples around point $n$. For example, $f$ may be the spectrum $X$ and $n$ can be either in the frequency domain or in the time domain. The variations of the function $f$ over the interval $(n-\Delta n, n+\Delta n)$ can be expressed by the coefficient $R$:

$$R[f(n-\Delta n), f(n+\Delta n)] = \frac{\sum_{i=0}^{2\Delta n} i f(n-\Delta n+i) - \frac{1}{2\Delta n+1}\sum_{i=0}^{2\Delta n} i \sum_{i=0}^{2\Delta n} f(n-\Delta n+i)}{\sum_{i=0}^{2\Delta n} f^2(n-\Delta n+i) - \frac{1}{2\Delta n+1}\sum_{i=0}^{2\Delta n} i \sum_{j=0}^{2\Delta n} j} \tag{3.2}$$

We can apply this coefficient to the computation of a frequency slope of the spectrum, as follows:

$$\frac{\Delta X(f,t)}{\Delta f} = R[X(f-\Delta f,t), X(f+\Delta f,t)] \tag{3.3}$$

This parameter measures the variations of the spectrum $X$ along the frequency axis. We have chosen an interval $\Delta f = 4$ frequency samples, while the parameter is computed every other 4 samples. This way, we represent global rather than detailed spectral variations. This parameter describe the spectral shape of the spectrum, and therefore should discriminate such features as compact vs. diffuse, and rising vs. flat vs. falling. Remind that labial, alveolar and velar burst are diffuse-rising, diffuse-falling and compact, respectively [Blumstein 79]. Figure 3.2 illustrates this parameter.

When we look at the spectrum $X(f,t)$ in the time domain, we are interested in the pattern followed by the peaks. For example, we would like to detect a rising vs. a falling second formant during the transition between a consonant and a vowel. This information is implicit in the sequence of spectral frames, provided that we examine the spectrogram over a time window of adequate duration. An approach to the problem of measuring formant transitions is the following, directly inspired by a Phonetic study by [Stevens 75] We concentrate our attention on the pattern followed by the energy in different bands during a short time interval. More precisely, a gradient operator is the following:

$$G(f,t) = X(f-1,t-1) + X(f+1,t+1) - X(f+1,t-1) - X(f-1,t+1) \tag{3.4}$$

If the first order derivatives of the function $X(f,t)$ are defined by differences between values one interval apart, rather than by a regression coefficient:

$$\frac{\delta X}{\delta f} = X(f+1,t) - X(f,t)\,;\ \frac{\delta X}{\delta t} = X(f,t+1) - X(f,t) \tag{3.5}$$

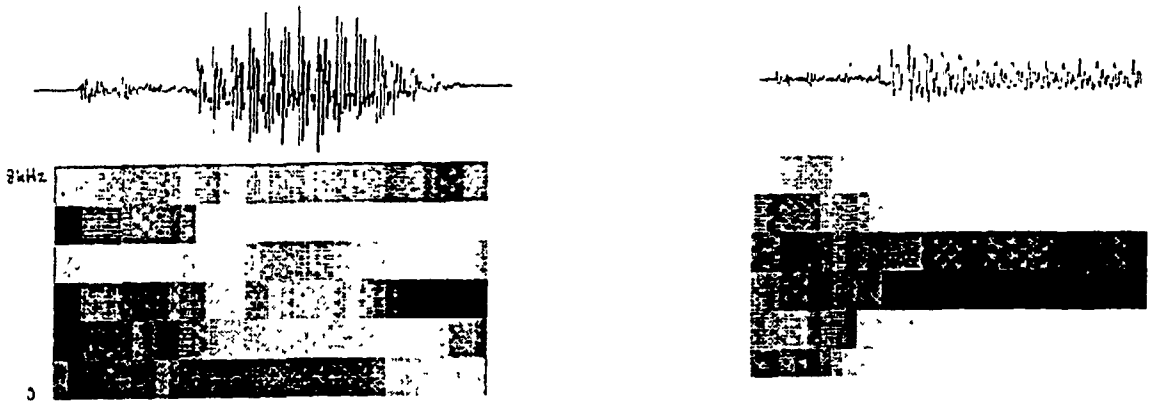



Figure 3.2: Top: signal. Spoken word: cartoons Bottom : Frequency derivative of the spectrogram, or slopes.

---

then $G(f,t)$ as in equation 3.4 approximates a second order derivative of $X(f,t)$ in both the time and frequency domain.

$$G(f,t) \simeq \frac{\delta^2 X}{\delta f \delta t} \tag{3.6}$$

If we compute this gradient for all the filters of the spectrogram, we get a picture that is difficult to evaluate. We define a smoother gradient operator that integrates the information in a larger window, and is computed only for the spectral peaks between 300 and 4000 Hz. First, all the local peaks are located in that frequency band, and the gradient is computed as follows:

$$G(f,t) = \begin{cases} X(f-1,t-2)+X(f-1,t-1)+X(f+1,t+1)+X(f+1,t+2) \\ -X(f+1,t-2)-X(f+1,t-1)-X(f-1,t+1)-X(f-1,t+2) \\ \qquad \text{if } E(t) > \textit{thresold} \text{ and } X(f,t) \text{ is a spectral peak} \\ 0 \qquad \text{otherwise} \end{cases} \tag{3.7}$$

$E(t)$ is the total energy of the signal in the window $t$, and the *threshold* discriminates between speech sounds and silence. Second, the gradient $G(f,t)$ is smoothed by averaging over the nine neighbours of the point $(f,t)$ in the spectrogram. Figures 3.3 and 3.4 illustrates this parameter. Note the activity of the gradient at the boundary between consonants and vowels, and for the liquid /r/.

During the time interval of 5 frames around frame $t$, if the energy is rising from filter

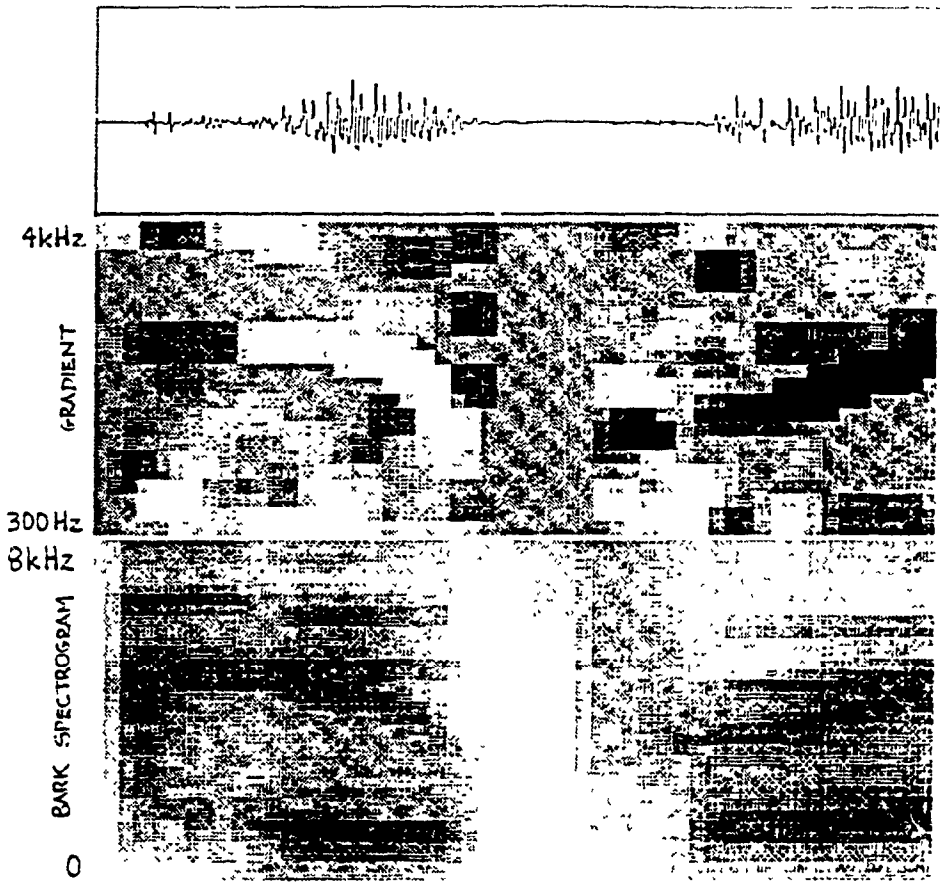



**Figure 3.3:** Top: a signal from the TIMIT continuous speech database. Sampling rate: 16 kHz. Spoken word: recuperate. Middle: second order time/frequency derivative, or gradient (24 values). When a formant is rising (/p/), the gradient is positive (darker) and when a formant is falling the gradient is negative (lighter) (/k/). Bottom: Bark scaled spectrogram (32 filters).

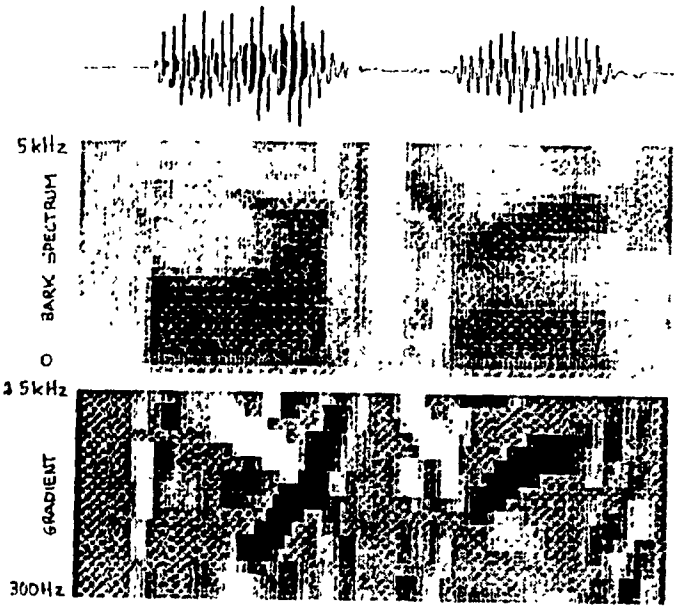



Figure 3.4: Top: a signal from the TI connected digit database. Sampling rate: 10 kHz. Spoken words: four three. Middle: second order time/frequency derivative, or gradient (18 values). Bottom: Bark scaled spectrogram (24 filters).

$f-1$ to filter $f+1$, then the gradient $G(f,t)$ is positive. If there is a falling frequency shift, the gradient is negative. In order to track the rapid spectral changes occuring between a closure and a vowel, we have set the frequency interval to 2 filters and the time interval to 5 frames. These intervals represent about 1.5 Bark on the frequency scale, and 20 msec on the time scale.

Finally, we review a spectral dissimilarity measure, inspired by [Fant 73] that should track spectral discontinuities in the speech signal. Given the smoothed spectrum $X(i)$ and $X(j)$ at frames $i$ and $j$, a distance can be defined from the dot product of this two spectral frames:

$$d(i,j) = 1 - \frac{\sum_f X_f(i)X_f(j)}{(\sum_f X_f^2(i) \sum_f X_f^2(j))^{1/2}} \tag{3.8}$$

This distance is 0 for spectra that are identical and is close to 1 for spectra that are very dissimilar. In this thesis we use the following symmetric measure of dissimilarity, that spans over an interval of 60 msec:

$$D(i) = d(i+3, i-3) + d(i+6, i-6) \tag{3.9}$$

In general, the detector $D$ shows a broad peak during the release of a unvoiced plosive and a smaller amplitude variation for nasal murmurs and voiced plosives. Figure 3.5 illustrates this parameter.

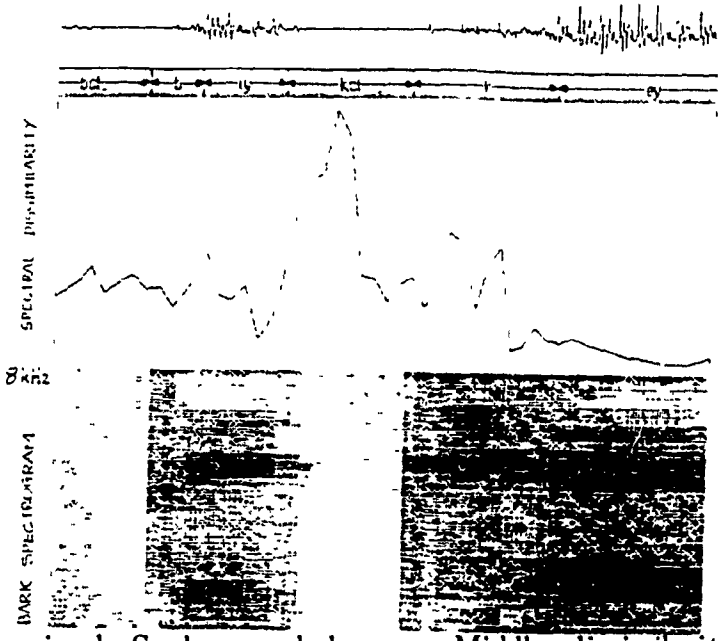


Figure 3.5: Top: signal. Spoken word: became. Middle: dissimilarity measure between neighbour spectral frames. Bottom: Bark scaled spectrogram.

## 3.4 Waveform based parameters

Relevant phonetic information can be extracted directly from the waveform of the speech signal at a low computational cost (see [OShaughn 87] for an overview). For example, energy and zero-crossing measurements can contribute to the detection of short stop and nasal sounds, and to the discrimination between voiced and unvoiced speech samples. In the following we review a set of parameters that should emphasize the changes in the energy of the waveform from a consonant to the neighbouring vowel.

Consider a sinusoidal signal at frequency $F_z$. The zero-crossing rate $ZCR(t)$ of that signal (defined as the number of zero crossings per sample estimated from a time window long enough to include a few periods) is related to the fundamental frequency $F_z$, since a sinusoid has two zero crossings per period. In particular, if $F_s$ is the sampling rate, then:

$$F_z(t) = (ZCR(t) * F_s)/2 \tag{3.10}$$

Speech is not a sinusoidal signal, and $F_z$ roughly correlates with a frequency location of major energy concentration, provided that there is no noise added to the speech signal and that the speech signal has zero mean. For voiced speech samples, $F_z$ correlates with a multiple of the glottal pulse frequency, and sometimes with the first formant resonance of the vocal tract, and for unvoiced speech samples in general $F_z$ will be at a higher frequency. For example, during the pronounciation of an unvoiced plosive followed by a

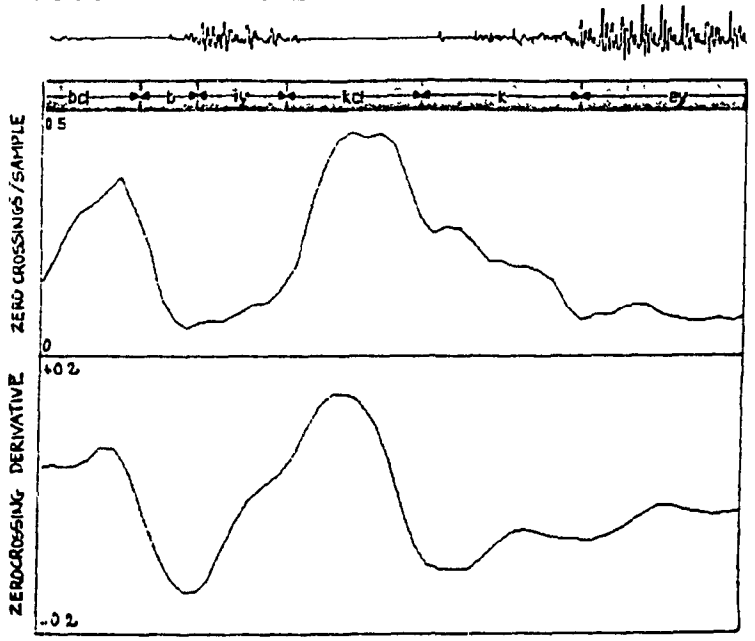



Figure 3.6: Top: signal. Spoken word: became. Middle: zerocrossing rate, Bottom : time derivative of the zero crossing rate.

vowel, the zero-crossing rate varies from a higher value during the closure to a lower value at the vowel onset. Since the absolute values of $F_z$ may vary depending on the phonetic context, it is appropriate to consider also its time derivative computed as follows:

$$\frac{\Delta F_z(t)}{\Delta t} = R[F_z(t - \Delta t), F_z(t + \Delta t)] \tag{3.11}$$

setting $\Delta t = 4$ frames, if we consider a rather long time interval of $2\Delta t + 1 = 9$ frames. Voiced closures and nasal murmurs usually do not show a high $F_z$. Therefore, when there is no noise added to the speech signal, we consider both parameters $F_z$ and $\Delta F_z$ as robust correlates for voicing discrimination and for detecting voiced/unvoiced transitions. Figure 3.6 illustrates these two parameters.

In order to track rapid releases of energy, we use the energy of the pre-emphasized and windowed signal $s(i)$ centered at frame $t$ and updated every 5 msec:

$$E(t) = 10 \log(\sum_i s(i)^2) \tag{3.12}$$

and its time derivative approximated by the linear regression of 9 successive time samples:

$$\frac{\Delta E(t)}{\Delta t} = R[E(t - \Delta t), E(t + \Delta t)] \tag{3.13}$$

This time interval of $2\Delta t + 1 = 9$ frames represents 45 msec. The time derivative of the energy spanning approximately 50 msec has been found to be relevant in the distinction between plosives and fricatives [Weigelt 90]. Plosive bursts show a distinct peak in $\Delta E(t)$,

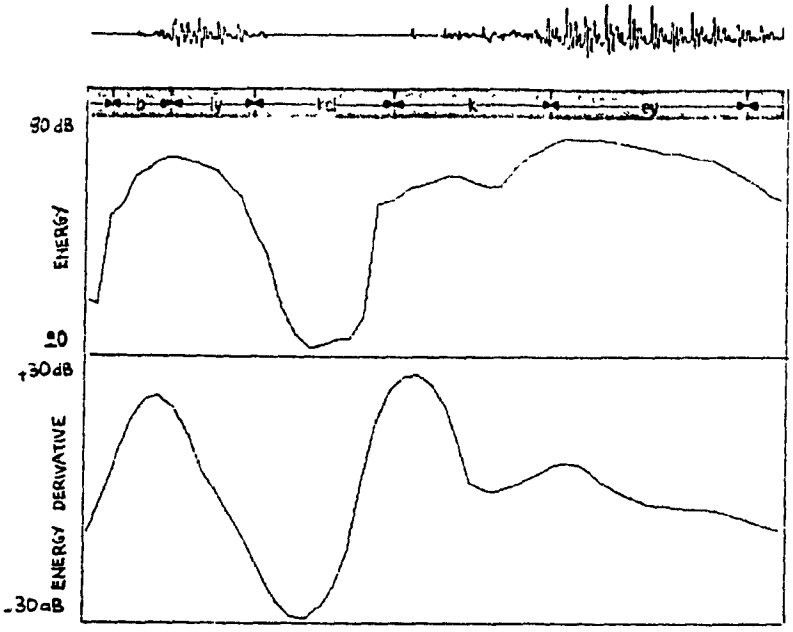



Figure 3.7: Top: signal. Spoken word: became. Middle: Energy. Bottom : time derivative of the energy.

while fricatives do not. Considering the application to the recognition of plosive and nasal sounds, it is possible to discriminate, at least visually, unvoiced velar stops and nasals. Indeed, unvoiced velar stops show a slow change in the energy function, while nasals are represented by long valleys that are visible both in the energy function and in its derivative. Figure 3.7 illustrates these two parameters. Another useful parameter is the voicing energy $V(t)$, derived from the energy of the input signal limited in the 60-500 Hz band, and its time-derivative $\Delta V(t)/\Delta t$ defined as in the above equation for $\Delta E(t)/\Delta t$. In this thesis we measure $V(t)$ from the speech signal filtered in the time domain by a fast IIR (infinite impulse response) band-pass Butterworth filter. We expect these last two parameters to help in the discrimination and the segmentation of unvoiced plosives in vocalic context. Indeed, $V(t)$ should vary from low to high values and $\Delta V(t)$ should show a broad peak at the vowel onset.

## 3.5 Summary

We have reviewed a collection of acoustic parameters that describe the speech signal in terms of its phonetic relevance. Some parameters are expected to contribute to the discrimination of the place of articulation and voicing manner of stop consonants, others are expected to be useful for the task of segmenting the speech signal, and discriminating stops from other consonants. The properties of all of the proposed parameters are

| *parameter* | *size* | *definition* | *is related to ...* |
|---|---|---|---|
| $X(f,t)$ | 32 | FFT-Bark spectrogram | Auditory Spectrogram |
| $\Delta X/\Delta f$ | 7 | frequency regression | global spectral shape |
| $\delta^2 X/\delta f \delta t$ | 24 | time/frequency derivative | peak trajectories |
| $D(t)$ | 1 | spectral dissimilarity | rapid changes in the spectrum |
| $F_z(t)$ | 1 | Frequency of the zero-crossing | voiced speech |
| $\Delta F_z(t)$ | 1 | time regression of $F_z$ | voiced/unvoiced transitions |
| $E(t)$ | 1 | signal energy | syllable onset and offset |
| $\Delta E(t)$ | 1 | time regression of $E$ | energy peaks and valleys |
| $V(t)$ | 1 | energy in 60-500 Hz band | voiced speech |
| $\Delta V(t)$ | 1 | time regression of $V$ | voiced/unvoiced transitions |

Table 3.1: Summary of the acoustic parameters

summarized in Table 1. These parameters are obviously highly correlated, and most of them are computed from the smoothed spectrogram $X$. The temporal evolution of these parameters is expected to represent sufficient information for the discrimination of stop and nasal sounds in continous speech, independetly from the speaker. In Chapter 6 we will report comparative experiments using the spectrogram in combination with the other parameters.

# Chapter 4

# Linear and Logistic Models

The previous chapters introduced the problem of classifying stop and nasal sounds from the point of view of experimental phonetics and acoustic analysis. In the next two chapters, we present the algorithms that will be used for solving the problem, from the point of view of statistical pattern recognition. Consider a population $\mathcal{X}^{train}$ of $N$ samples of a p-dimensional vector of real values. Each sample is a vector of observations or measurements that has been previously labeled as belonging to one out of M classes $C_i$. The general pattern recognition problem is to design a *classification algorithm* that is able to label a new population $\mathcal{X}^{test}$ with enough accuracy, i.e. with a minimum number of classification errors. The algorithms presented in the next two chapters model the observation vectors by several parameters. The parameters of the classifier will be optimized based on the labeled population $\mathcal{X}^{train}$. In our case, we extract the population $\mathcal{X} = \mathcal{X}^{train} \cup \mathcal{X}^{test}$ from the TIMIT database, the observation vectors are the acoustic parameters discussed in the preceding chapter, and the $M$ classes to be discriminated are the 10 stop and nasal sounds.

In this chapter we will review some popular *linear* and *loglinear* models used in pattern recognition. This term refer to algorithms that make some important assumptions about the distribution of the input data in each class. These assumptions allow the design of simple classifiers with a few free parameters to be estimated. However, If the input data violate the assumptions, these classifiers will not minimize the errors.

In speech recognition tasks, and in particular for the acoustic parameters that we described in the preceding chapter, the class distributions are rather complex. Almost certainly, they will violate the assumptions made by linear and loglinear classifiers. Also, it is required to classify sequences of statistically correlated observations. Then, there is the need for *non linear* classifiers that are fed by more than one observation vector at the time. Therefore, A particular class of non linear classification technique, multilayer perceptrons (MLP) will be preferred. That technique, described in the next chapter, can

be considered as a very powerful extension to the loglinear regression model explained at the end of this chapter. Of course, the number of parameters to be estimated in the non linear case will rise considerably.

Singular Value Decomposition, Linear Discriminant Analysis and Logistic Regression will be discussed in some detail in the next sections. The ideas and the algorithms presented in this chapter constitute the necessary background for understanding how our phonetic classifier based on MLP works, and what are the advantages and the limitations of using linear and loglinear models of the observation parameters, with respect to non linear models such as MLP. We begin by discussing the Singular Value Decomposition, an important tool that will be used for solving the eigenvalue equation involved by Linear Discriminant Analysis.

## 4.1 Singular Value Decomposition

We begin by describing the Singular Value Decomposition (SVD). This well known method provides a compact description of the underlying structure of any data matrix. The properties of the SVD method are discussed in many matrix computations textbooks and tutorials (among others, [Stewart 73, Klema 80]).[1] Remind the SVD theorem:

**SVD Theorem** Given any $(n \times p)$ matrix $X$, it is possible to write

$$\begin{aligned} X &= U\ D\ V^T \\ (n \times p) &\quad (n \times r)(r \times r)(r \times p) \\ D &= diag(w_1, w_2, \ldots, w_r) \\ & w_i \geq w_{i+1} \geq 0 \end{aligned} \tag{4.1}$$

where $r \leq p$ is the rank of matrix $X$, i.e. the effective number of its linear independent columns, D is a diagonal matrix filled with $r$ positive singular values $w_i$, and the $r$ columns of $V$ (the rows of $V^T$) are called the (right) singular vectors of X, and $U$ is another orthogonal matrix of (left) singular vectors. Each of the $r$ columns (rows) of $V$ ($U$) has unit length.

Suppose the matrix $X$ is filled with $n$ samples of $p-dimensional$ data. The SVD theorem tells us that each sample (or row) $x_i^T$ of $X$ can be expressed as the linear combination of $r$ orthogonal vectors $v_j$. These $r$ vectors can be considered as an alternative set of orthogonal coordinate axes, that statisticians call principal components. The covariance matrix of the data in this new space is D, that is diagonal. This means that in this new

[1] The C listing of a general SVD program can be found in [Press 88] Programs for SVD-based applications are available on electronic mail, through the address *netlib@research att com*

coordinate space two different variables are not statistically correlated. Moreover, since in the new coordinate system the variance of each variable $j$ is $w_j$, the spread of the data $X$ projected on each singular vector $v_j$ is proportional to its associated singular value.

Let's formalize how we can obtain the singular vectors from the estimated covariance matrix $W$ of the data sample $X$.

$$W_{ij} = \frac{1}{n}\sum_{k=1}^{n}(x_{kj} - < x_j >)(x_{ki} - < x_i >) \tag{4.2}$$

Assume that $X$ has been normalized such that each (estimated) mean value $< x_i >$ is zero. The variance of the projection of the sample $X$ onto any vector $v$ is $v^T W v$. We are looking for the unit length vector(s) that maximize $w^2$:

$$w^2 = v^T W v \tag{4.3}$$

We take the partial derivative of the right part of the above equation with respect to $v$ and we set it to zero. We find the eigenvalue equation:

$$Wv = w^2 v \tag{4.4}$$

Since $W = X^T X$, the right singular vectors of $X$ are the eigenvectors of $W$ and the eigenvalues of $W$ are the singular values of $X$ squared. Therefore the singular vector with the largest singular value accounts for the greatest variance of the data, and the vectors with the smallest singular values account for the smallest variance. In Figure 4.1 the two singular vectors are plotted for a simple two dimensional data distribution. In practice, the rank $r$ of the data matrix $X$ is unknown, and the SVD algorithm will return $p$ vectors as well as $p$ singular values. It is the care of the user to choose how many vectors to retain, based for example on the relative magnitude of the singular values or on other criteria.

When the number $n$ of rows in $X$ is small, the SVD algorithm can be applied directly to the matrix $X$. When $n$ is too large with respect to the computing resources, the SVD algorithm can be applied to the estimated covariance matrix $W$, since the singular vectors of a symmetric positive definite matrix are also its eigenvectors. Moreover, the symmetric nature of $W$ reduces the complexity of the algorithm.

Since statistical analysis algorithms are often based on the evaluation of an inverse matrix computed from $W$, it is desirable to evaluate how close is $W$ to be singular. A criterion to evaluate the condition of $W$ with respect to inversion is the ratio of the 2-norms of $W$ and $W^{-1}$. If $W$ is ill-conditioned its determinant is small and the 2-norm of $W^{-1}$ is large with respect to the Euclidean norm of $W$. Applying SVD to both matrices allow us to evaluate the condition number of the matrix $W$:

$$\frac{\|W\|_2}{\|W^{-1}\|_2} = \frac{w_1}{w_p} \tag{4.5}$$

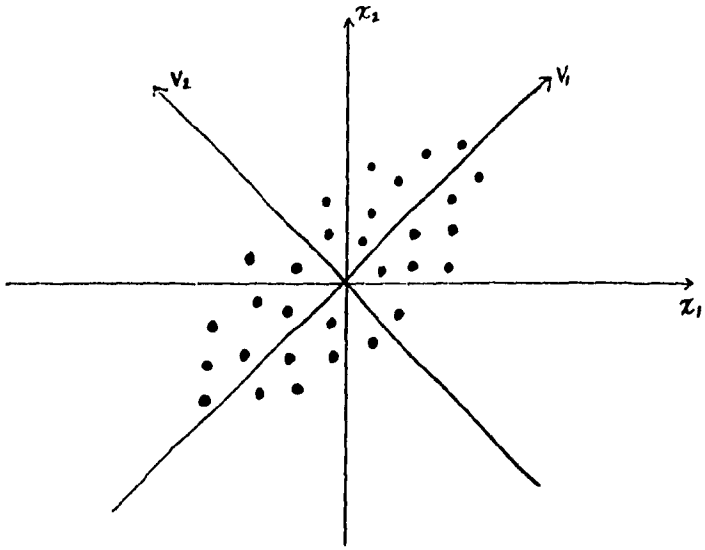



Figure 4.1: Plot of a two dimensional cloud of data $X$ and of the singular vectors of $X^TX$, or principal components.

---

Therefore SVD provides a very useful tool for evaluating the condition of a matrix with respect to inversion. It should be noted that SVD can be applied to any matrix, either well or ill conditioned, since it is a numerically stable algorithm that involves only matrix rotations and no matrix inversions.

SVD and principal component analysis (PCA) has already been applied to speech processing. In speech enhancement from noise [Bakamidis 90] suggests that it is possible to discriminate speech from noise by SVD because the speech signal is responsible for the singular vectors with the largest singular values (largest variance), while the noise is responsible for the singular vectors with the smaller singular values (smaller variance). In speech coding, [Atal 89] shows how the excitation of the linear prediction filter can be expressed as the linear combination of singular vectors of the autocorrelation matrix of the filter impulse response. The number of singular components of the exitation to retain, and the precision of their coding is a compromise between a lower transmission rate and the perceived speech quality. In automatic speech recognition, the transformation of many correlated acoustic features (such as the spectrum and the energy of few successive frames) into fewer uncorrelated and normalized features has been proved to be useful either for a dynamic programming approach [Bocchieri 86] or for a continuous densities hidden Markov model methodology [Brown 87].

In this thesis, SVD will not be applied directly to the data. Instead, it will be applied to the output vector of a non linear classifier. The principal components will be used to

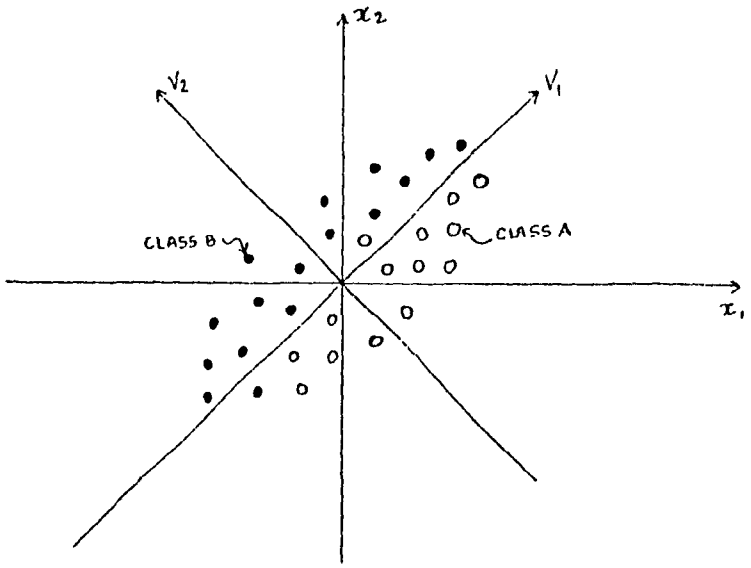



Figure 4.2: A case in which the classes overlap when they are projected on the first principal component. Adapted from Brown 87.

represent the output distribution of the training set with a compact set of uncorrelated parameters. This will be useful when the output of the classifier will be processed by another statistical algorithm, i.e. a hidden Markov model.

The reduction of $p$ possibly correlated features into $r$ uncorrelated ones is advantageous when we have to estimate many statistical parameters from a finite size training set, and we want to reduce the number of features without loosing relevant information, but it is not clear if the use of principal components will be of any advantage in a classification task. It is possible that the directions of greater variance (the principal components) are also the directions of maximum overlapping between the classes. Figure 4.2 illustrates this unfortunate case.

## 4.2 Linear Discriminant Analysis

In his PhD thesis [Brown 87] suggests the use of linear discriminant vectors as a promising alternative to principal components, when the task is not data compression but pattern recognition by statistical methods. In the next section, we will report on the use of linear discriminant analysis, based on the thesis by Brown and on the textbook by [Dillon 84].

The goal of PCA is to account for the greatest variability of the whole data sample with a smaller set of uncorrelated features. Linear discriminant analysis (LDA) looks for the directions in the feature space that account for the greatest discrimination between

the classes. Ideally, when we project the data sample on a discriminant direction, we would like to see the elements of one class clustered around a certain average value with a small variance, and the elements of all the other classes scattered far away from the average of that particular class. In other words, when projected onto the discriminant vector, one class should have a small variance, while the entire sample (i.e. all the classes $i$) should have a large variance. Define $S$ as the average within-class covariance matrix, that can be estimated from:

$$S = \frac{1}{n}\sum_i n_i W^i \tag{4.6}$$

We try to maximize the ratio of the total projected variance to the average within-class projected variance, defined as:

$$\lambda = \frac{v^T W v}{v^T S v} \tag{4.7}$$

Setting the gradient of $\lambda$ with respect to $v$ to zero yields this time the *generalized* eigenvalue equation:

$$Wv = \lambda S v \tag{4.8}$$

The SVD method can be applied for the solution to this eigen problem. Since $S$ and $W$ are (estimated) covariance matrices, they are symmetric and (almost always) positive definite and can be decomposed into the product of a lower triangular and an upper triangular matrix, via the Cholesky decomposition algorithm [Stewart 73]. The eigenvalue equation becomes:

$$L_1 L_1^T v = \lambda L_2 L_2^T v \tag{4.9}$$

Introduce the vector $z = L_2^T v$. In terms of $z$, the equation is:

$$L_2^{-1} L_1 L_1^T L_2^{-T} z = \lambda z \tag{4.10}$$

Define $A = (L_2^{-1} L_1)^T$. The transformed equation is now a standard eigenvalue equation:

$$A^T A z = \lambda z \tag{4.11}$$

and the eigenvectors of $A^T A$ are the right singular vectors of $A$. In summary, provided that $L_2$ is non singular, that is $S$ is full rank, the following algorithm will find the linear discriminant vectors:

1. Decompose $W$ into $L_1 L_1^T$ and $S$ into $L_2 L_2^T$ via the Cholesky decomposition.
2. Invert $L_2$ by columns, solving the linear system $L_2 L_2^{-1} = I$.
3. Apply the SVD algorithm to $(L_2^{-1} L_1)^T$.
4. Transform the right singular vectors $z$ into $v = L_2^{-T} z$.

In practise, we would like to avoid computing the Cholesky decomposition of an ill-conditioned matrix $S$, and we can take advantage from the fact that an analogous algorithm can be applied by defining $z = L_1^T v$ and inverting from the Cholesky decomposition of $W$. Before applying the algorithm, it is indeed appropriate to evaluate the condition number of both $W$ and $S$ by SVD as in the equation 4.5, and to apply the decomposition to the covariance matrix with the smallest condition number.

The linear discriminant vectors are expected to account for the the greatest discrimination between the classes. However, there are some important conditions under which LDA is an optimal procedure for producing the smallest classification error rate of the data sample $X$. In particular, we must assume:

- The distribution of the $p$ initial features in each class is a unimodal Gaussian Multivariate.
- Each one of the considered classes has the same expected covariance matrix.

To clarify this point, we consider how classification is performed by LDA. When an unknown test pattern is presented to the classifier, it is projected onto the space described by the discriminant directions with the largest eigenvalues, and the Euclidean distance from each projected class average is computed. The pattern is labeled with the closest average class label. If the classes are not Gaussian unimodal, projecting the data onto linear discriminant vectors is theoretically unjustified, since the data distributions cannot be modeled faithfully by the mean vectors and the covariance matrices used in the generalised eigenvalue equation.

If the classes are Gaussian multivariate, the optimal classifier is the one which computes the Mahalanobis distances between a pattern and each class average, and then pick the class with the minimum distance [Duda 73](pp. 22-31). The Mahalanobis distance between two classes, or between one pattern and one class is a quadratic distance weighted by the inverse of the within-class covariance matrix, therefore it takes into account the spreading of the data in the original feature space. At a given Euclidean distance between two class averages, if the spread of each class is large the two classes tend to overlap and the diagonal terms of the covariance matrix are large. This will be reflected by a small Mahalanobis distance. Formally, this distance is directly derived by modeling each class distribution with a Gaussian multivariate. More precisely, it is twice the exponent of the Gaussian:

$$D_M(X, C_i) = (X - \mu_i)^T W'^{-1} (X - \mu_i) = X^T X + X^T W'^{-1} X - 2\mu_i^T W'^{-1} X + \mu_i^T W'^{-1} \mu_i \tag{4.12}$$

It can be proven [Duda 73] (pp. 152-153) that computing the Euclidean distance onto the discriminant space is equivalent to computing the Mahalanobis distance, *only if* we assume the same covariance matrix $W' = W$ for each class and different class averages

$\mu_i$. Indeed, the quadratic term $X^T W^{-1} X$ that appears in the Mahalanobis distance can be dropped when we compare two such distances if it is the same for each class, and the linear factors can be expressed in terms of the linear discriminant vectors [Duda 73] (pp. 152-153).

When the two assumptions are not satisfied, the procedure will not be optimal, and we are not expected to minimize the misclassification error rate of the sample $X$ by projecting the data on the linear discriminant vectors.

Coming back to the comparison between LDA and PCA with respect to a classification problem, it is possible that in practise the two methods may perform similarly. This may happen when the directions of greater variance of the data sample are close to the directions of maximum discrimination, or when the average within-class covariance matrix is almost diagonal, or when the assumptions underlying LDA are not matched by the training data, or when both estimated covariance matrices are ill-conditioned, giving poor results in the estimation of the LDA eigenvectors.

## 4.3 Logistic Regression

We conclude this chapter by presenting a variation to Linear Discriminant Analysis that is closely related to Artificial Neural Networks. Consider the two class problem, in which we are asked to assign a vector $X$ to one of two classes $C_i$, with $i = 1, 2$. According to Bayes' theorem, the posterior probability of $X$ being a member of class $C_i$, depends on the conditional joint probabilities $Pr(X|C_i)$.

$$Pr(C_1|X) = \frac{Pr(X|C_1)Pr(C_1)}{Pr(X|C_1)Pr(C_1) + Pr(X|C_2)Pr(C_2)} \tag{4.13}$$

Assuming equal prior probabilities $Pr(C_i)$, and dividing numerator and denominator of the right-hand side of the above equation by $Pr(X|C_1)$ we obtain:

$$Pr(C_1|X) = \frac{1}{1 + \frac{Pr(X|C_2)}{Pr(X|C_1)}} \tag{4.14}$$

If the class-conditional probabilities are Gaussian multivariate with different means $\mu_i$ and common covariance matrix $W$, the above equation becomes:

$$Pr(C_1|X) = f(X, \theta) = \frac{1}{1 + \exp -(V_0 + V^T X)} \tag{4.15}$$

where the parameter vector $\theta = (V_0, V^T)$ can be obtained by Linear Discriminant Analysis. Applying the algorithm outlined in the previous section for the 2 class problem, we find

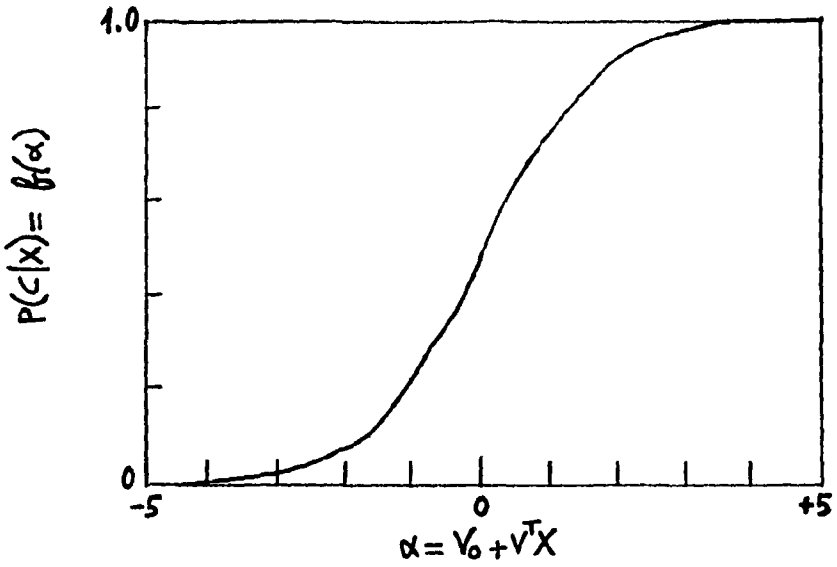



Figure 4.3: The sigmoid function is an estimate of a posterior class probability in the loglinear regression model.

---

one discriminant direction corresponding to the greatest eigenvalue and one separation point:

$$\begin{aligned} V &= W^{-1}(\mu_1 - \mu_2) \\ V_0 &= -\frac{1}{2}(\mu_1 + \mu_2)^T W^{-1}(\mu_1 - \mu_2) \end{aligned} \tag{4.16}$$

The function $f(X, \theta)$ is called a logistic or sigmoid function. It is S-shaped and has an inflection occurring at $\frac{1}{2}$. Changes of amplitude in the threshold or separation point $V_0$ shift the surface laterally, while changes in the vector $V$ affect its dispersion (see figure 4.3).

This approach can be extended to the problem of classifying a pattern into one of $M > 2$ classes, in which case for each class $C_i$ the factor in the exponential can be expressed in terms of the Mahalanobis distance between the pattern $X$ and the class average $\mu_i$, disregarding the quadratic term $X^T W^{-1} X$ that appears in all of the $M$ distances.

$$\begin{aligned} V^i &= W^{-1}\mu_i \\ V_0^i &= -\frac{1}{2}\mu_i^T W^{-1}\mu_i \end{aligned} \tag{4.17}$$

Therefore, the logistic regression model is formally equivalent to Linear Discriminant Analysis if the classes are distributed as Gaussian multivariates with different mean vectors and the same within-class covariance matrix. However, there are strong theoretical

arguments in favor of logistic regression over Linear Discriminant Analysis. [Cox 70, Anderson 72, Press 78]. Indeed, logistic regression can model more families of class conditional distributions than LDA. In particular, it can be proven that this model works if the distributions of the data $X$ are [Anderson 72]:

1. Gaussian multivariates with equal covariance matrices;
2. independent Binary multivariate;
3. Binary multivariate following the logistic model with equal quadratic and higher order terms for each class;
4. a combination of (1) and (3).

In fact, logistic regression has been initially formulated for the analysis of binary data [Cox 70]. Another advantage of logistic regression is its relative robustness with respect to data that do not fit the assumptions [Press 78]. Since logistic regression can model more families of probability distributions than linear discriminant analysis, it can be suitable for many more classification problems.

It should be noted that there are some problems and some distributions that neither linear discriminant analysis nor logistic regression can model accurately. In particular, both cannot model multi-modal distributions, in which each class is represented by more than one non connected and possibly non convex cloud of points in the original feature space. A classical example of such a problem is the classification of binary data according to the exclusive-or (XOR) rule.

If the class conditional distribution do not satisfy the LDA assumptions we should not estimate the logistic regression parameters by LDA. Consider the $M$ class problem in a feature space of $p$ dimensions. We wish to model the probabilities $Pr(C_i|X)$ with sigmoid functions that depend on the unknown matrix of $(p+1)M$ parameters $\theta = (\theta_1, \ldots, \theta_M)$. Instead of using Linear Discriminant Analysis which require the estimates of $M$ class means and of the within-class and sample covariance matrices, we can try to estimate directly the parameters $\theta$ from the available sample $\mathcal{X}^{train}$ of $n$ labeled training data.

$$\mathcal{X}^{train} = (X_1 \in C_i, X_2 \in C_j, \ldots, X_n \in C_k) \;\; i, j, k = 1, \ldots M \tag{4.18}$$

Two approaches can then be applied: *maximum likelihood* estimation or *Least Mean Square* estimation. All approaches lead to an iterative algorithm, in which we define a differentiable analytical criterion $\mathcal{E}(\theta)$ that has to be optimized by the estimated value of $\theta$. We start from an initial, possibly random set of parameters $\theta^0$ and then we iteratively correct the estimates $\theta^{t+1}$ from the precedent values $\theta^t$ until the criterion has been met. Since the same estimation problem has to be solved for the class of Artificial Neural Networks that we take in consideration for this thesis [Gish 90], we postpone the discussion to the next chapter.

# Chapter 5

# Non Linear Models

Multilayer perceptrons (MLP) are distributed networks of many elementary units called artificial neurons. Each unit performs simple computations on its input vector, but the sytem has a complex overall behavior. In the past five years, these networks have found many applications, including adaptive equalization, signal modeling, control systems, pattern recognition and machine learning.

Considering a pattern recognition problem, the parameters governing the computations for each unit can be optimized by an iterative algorithm in order to classify patterns from examples, like a logistic regression machine. Unlike a logistic regression machine, these units can be activated either by the observation vector or by other such units. In a linear or logistic model any unit computes a function of the input parameters only. One advantage of using MLP instead of linear discriminant analysis or logistic regression is that sufficiently complex MLP do not make any restrictive assumption about the underlying distribution of the input data, and therefore they can model more families of statistical distributions. Other advantages are that one can build complex classifiers that look at *sequences* of heterogenous real-valued and binary inputs, and that the classification algorithm can be readily implemented on parallel hardware.

This chapter reviews the basic properties of MLP and the associated optimization algorithm. Some links between linear discriminant analysis, logistic regression and MLP are presented. Finally, the issues related to our particular classification problem are discussed. The presentation stresses links between MLP and other statistical and optimization algorithms. The presentation is based on the textbook by [Duda 73], the fundamental paper by [Rumelhart 86] and on three tutorial papers [Hinton 87, Lippmann 87, Lippmann 89]. A complete report on this field goes beyond the scope of this thesis, and the interested reader should refer to the books by Duda and Hart [Duda 73], the one by the Parallel Distributed Processing Research Group [Rumelhart 86] and the up-to-date book by Hertz, Krogh and Palmer [Hertz 91].

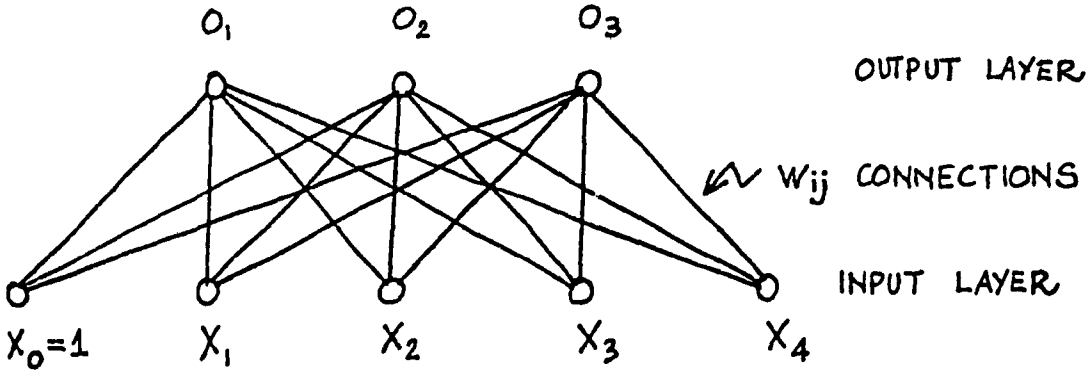



Figure 5.1: A single layer perceptron.

## 5.1 Multilayer Perceptrons

**A Simple Loglinear Perceptron** We begin our review by describing a simple network, the perceptron, that implements a logistic classifier. A perceptron is a layer of $K$ nodes. Each node is connected to the input vector via a set of adjustable weights. The weighted sum of the inputs is passed through a continous non-linear function in order to produce the output of the node. A perceptron is illustrated in figure 5.1. For example, the output node $j$ performs the following operation on the input vector $X^T = (x_0 = 1, x_1, x_2, \ldots, x_p)$ (the weight $w_{0j}$ being an adjustable threshold):

$$o_j = f(\sum_{i=0}^{p} w_{ij} x_i) \tag{5.1}$$

$$f(x)_{LOG} = \frac{1}{1 + \exp -x} \tag{5.2}$$

$$f(x)_{SYMM} = tanh(x) = 2f(x)_{LOG} - 1 \tag{5.3}$$

For the symmetric case, the output spans the range $-1.0$ to $+1.0$, while for the logistic case, the output spans the range 0.0 to $+1.0$. In both cases, the function $f$ is differentiable with respect to each of the $p + 1$ connecting weigths $w_{ij}$. Following [Rumelhart 86] we introduce the variable $net_j = \sum_i w_{ij} x_i$ that represents the network activation sent to node

$j$, and we can write:

$$\frac{\partial f(net_j)}{\partial w_{ij}}_{LOG} = \frac{1}{2}\frac{\partial f(net_j)}{\partial w_{ij}}_{SYMM} = f(net_j)_{LOG}(1.0 - f(net_j)_{LOG})x_i \qquad (5.4)$$

The output function $f(x)$ has opposite values with respect to its mid-range, whether the scalar product between the input vector and the weight vector is greater or lower than the threshold $w_{0j}$. Close to the mid-range value, $f(x)$ is almost a linear function and its derivative is maximum. This function can be interpreted as an estimate of the a posteriori class probability $Pr(C_j|X)$. If we want to separate two classes with one of these units, we may select the weights $w_{ij}$ by LDA with the matrix computation algorithm described in the previous chapter. This technique will be optimal only if the two classes are distributed as Gaussian multivariates with different average values and the same covariance matrix. If we do not estimate the connecting weights by LDA the perceptron is equivalent to a logistic regression machine and more types of distributions, like mixtures of Gaussian multivariates and dichotomous variables, can be optimally discriminated this way. In general, it has been proven that

> *If the classes can be separated by a linear combination of the input variables, then a single-layer loglinear perceptron can be designed to insure the minimum number of classification errors.* [Hertz 91] (pp. 102-108)

We shall present the algorithm for selecting the weights of a perceptron in the more general case of networks with hidden nodes.

**Multilayer Perceptrons** Consider the case of multi-modal distributions, when the classes are represented by non-connected convex regions. If the modes of the distributions are known it is possible to design a complex perceptron topology to solve that problem. In general, the topology is such that there exists *hidden* nodes connected to the input vector, and the output nodes may be connected to the input feature vector *or* to some of the hidden nodes. By distributing the information on the correlated activation of several hidden nodes, multi-layer perceptrons can out-perform a simple loglinear perceptron by computing complex non-linear functions of the input vectors. Such is the case for the XOR problem, for which it has been shown that one hidden node is sufficient to model the two modes of the observable distribution [Rumelhart 86]. A simple perceptron with one hidden layer is illustrated in figure 5.2.

The problem is how to optimize all the weights of complex networks with many hidden nodes. An optimization algorithm exists for the class of MLP in which the nodes are divided into three or more layers: one input layer (the observation vector) at the bottom level, one or more hidden layers, and one output layer at the top level. A node in any layer is activated by other nodes belonging to layers underneath, but never by nodes belonging

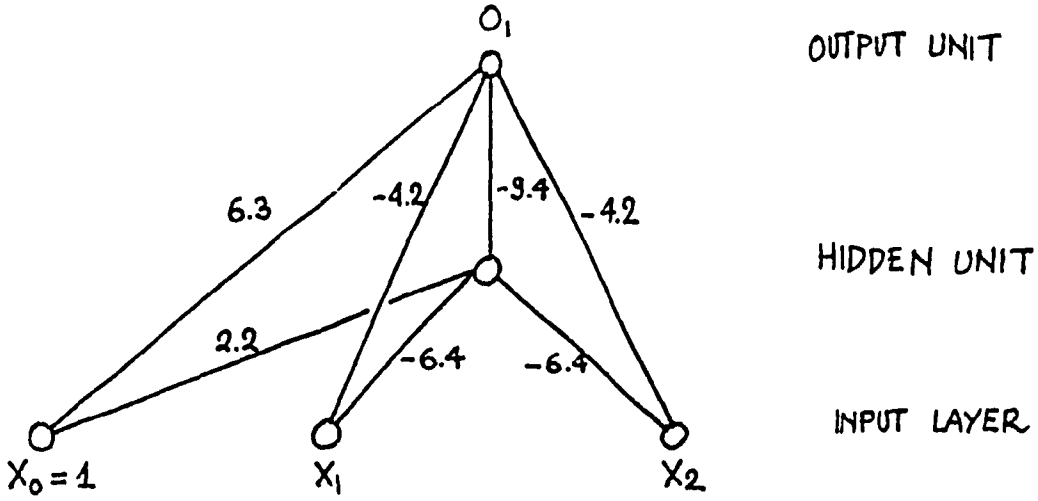



Figure 5.2: A multi-layer perceptron that solves the XOR classification problem. Adapted from Rumelhart 86.

to layers above it. The algorithm is called *generalized delta rule* or *back-propagation* [Rumelhart 86] and is a very powerful extension to the classical gradient descent algorithm existing for a class of adaptive filters [Widrow 60]. The algorithm is outlined in the next section.

## 5.2 Optimization of the network parameters

In general, we have to set beforehand a different *desired* configuration for the output vector of $K$ nodes, depending on each class. The common practice is to set $K = M$ output nodes, and to set a desired output $d_i = 1.0 - \upsilon$ when the input vector belongs to class $C_i$, and $d_i = -1.0 + \upsilon$ or $d_i = 0.0 + \upsilon$ otherwise[1]. Then, we define an error criterion $\mathcal{E}(\theta)$ for evaluating the performance of a particular set $\theta$ of $Q$ weights. For the case of a single-layer perceptron $Q = K(p+1) = M(p+1)$. For the case of a fully connected three-layer perceptron with $H$ hidden nodes $Q = M(H+1) + H(p+1)$. It is necessary for the error criterion to be a differentiable function with respect to all the network outputs $o_j$. In general, the error criterion is a monotonic function of difference terms $(d_j - o_j)$. Since each output function depends on the parameters $w_{ij}$, the error criterion is ultimately a

[1] $\upsilon$ is a small positive constant that is set to prevent the network operating in the saturating region of the output function.

function of $w_{ij}$, or $\theta$.

$$\mathcal{E}(\theta) = \mathcal{E}(\ldots, d_j - o_j, \ldots) = \mathcal{E}(\ldots, w_{ij}, \ldots) \tag{5.5}$$

The following algorithm will optimize the weights with respect to the given error criterion. Starting from an initial set of random weights $\theta^0$ we can iteratively make small adjustments to all the weights until the criterion is optimized. Let $\theta$ be a point in the $Q$-dimensional real space $\Re^Q$. If we make a small adjustement to all the weights we move from the point $\theta$ to the point $\theta + \Delta\theta$, and we can express the changes in the criterion function as a Taylor expansion around the initial point $\theta$.

$$\mathcal{E}(\theta + \Delta\theta) = \mathcal{E}(\theta) + \nabla\mathcal{E}(\theta)^T\Delta\theta + \frac{1}{2}\Delta\theta^T\nabla^2\mathcal{E}(\theta)\Delta\theta + \ldots \tag{5.6}$$

If we stop the Taylor expansion at the first term, we have to estimate the *gradient vector* $\nabla\mathcal{E}(\theta)$ of $Q$ first-order derivatives. If we want a more accurate estimate of the changes in the error criterion, we have take into account the second term and we have to estimate some or all the terms of the *Hessian matrix* $\nabla^2\mathcal{E}$ of $Q^2$ second-order derivatives.

$$\nabla\mathcal{E}_n = \frac{\partial\mathcal{E}}{\partial w_{ij}} \tag{5.7}$$

$$\nabla^2\mathcal{E}_{nm} = \frac{\partial^2\mathcal{E}}{\partial w_{ij}\partial w_{kl}} \tag{5.8}$$

The first-order approximation leads to the *back-propagation* algorithm, while the second order approximation leads to the class of *conjugate gradient* and approximate *Newton* algorithms [Becker 89, McDonald 90]. We consider here the first-order approximation. In order to minimize the *approximate* error criterion, at each iteration $t$ the adjustments are proportional to the negative of the gradient of $\mathcal{E}$ with respect to the weights ($\eta$ being a small proportionality constant called the learning rate).

$$\Delta\theta = -\eta\nabla\mathcal{E}(\theta) \tag{5.9}$$

This adaptation rule can be computed for each one of the $Q$ weights $w_{ij}$ as a product of two independent terms, as follows:

$$w_{ij}^{t+1} - w_{ij}^t = -\eta\frac{\partial\mathcal{E}^t}{\partial w_{ij}^t} = -\eta\frac{\partial\mathcal{E}^t}{\partial o_j^t}\frac{\partial o_j^t}{\partial w_{ij}^t} \tag{5.10}$$

We assume that the *real* error criterion behaves like the approximate one, and that it should decrease by a small amount after each adjustment of the weights, until it reaches a minimum value. Indeed, when we substitute the computed value of $\Delta\theta$ into the Taylor expansion of $\mathcal{E}(\theta)$ we obtain:

$$\mathcal{E}(\theta + \Delta\theta) \approx \mathcal{E}(\theta) - \frac{1}{2}\eta^2\|\nabla\mathcal{E}(\theta)\|^2 \tag{5.11}$$

From the above equation, we notice two important aspects of the algorithm. First, the learning rate $\eta$ plays a key role for the convergence of the adaptation algorithm. If $\eta$ is too small the convergence rate can be very slow, and if it is too big, the weight adjustments computed by the first-order approximations are too big to follow the real shape of the error criterion surface. Second, the convergence is slow if the amplitude of the gradient of the error criterion is small.

In general, there are several important issues to be concerned with such a gradient descent adaptation rule. If the network does not have hidden nodes and the classes are linearly separable, the error criterion has only one minimum with respect to the weight space, and the first-order approximation will be adequate [Sontag 91]. This adaptation rule will find the minimum of the error surface eventually (in a finite number of iterations), depending on the initial weights $\theta^0$ and on the learning rate $\eta$. If the network has hidden nodes the error criterion surface is no longer convex*whether or not the classes are linearly separable*, and the adaptation algorithm is not guaranteed to converge to the global minimum and may converge instead to one of the many local minima [Sontag 91]. (see for example, the experiments reported by [Kolen 90] on the XOR problem).

Depending on the relationship between the difficulty of the problem and the network topology, both the convergence rate and the final value of the error criterion may vary considerably. In practice, the application of the back-propagation algorithm to perceptrons with hidden units has been shown to solve many non linear pattern recognition problems that are not solved optimally by standard linear and logistic models, and the most important issue remains the number of iterations required for the convergence of the algorithm, and not the presence of local minima [Hinton 87]. In the following, we derive the adaptation rule for a particular type of criterion: *Least Mean Square* (LMS).

**Least Mean Square Error Criterion** We define the following criterion for evaluating the performance of a particular set of weights:

$$\mathcal{E}(\theta) = \sum_{\mathcal{X}} (\sum_{j=1}^{K} (d_j - o_j)^D) \tag{5.12}$$

The expression $(d_j - o_j)^D$, where $D$ is an integer, defines a distance measure between the desired outputs and the network outputs. $\mathcal{X}$ is either the complete set or a subset of training patterns. Different values of $D$ imply different metrics and different assumptions about the distribution of the output values $o_j$ [Burrascano 91]. In particular, $D = 1$ is the $L_1$ City-Block metric, that assumes that the output distribution decay exponetially from the average. $D = 2$ is the $L_2$ Euclidean metric, that assumes that the distributions are Gaussian. $D = \infty$ is the $L_\infty$ Chebyshev metric[2] that assumes that the distribution

[2]The most popular metric is the Euclidean one, although there are some theoretical justifications for choosing the Chebyshev metric, at least at the beginning of the adaptation and for small learning rates

is box-car shaped (uniform). We derive here the adaptation rule for the case $D = 2$. The partial derivatives of the error criterion can be recursively computed from the top to the bottom layer in an efficient manner. If the node $j$ is an output node and we use the logistic function, the derivatives are [3]:

$$\frac{\partial \mathcal{E}}{\partial o_j} = -(d_j - o_j) \tag{5.13}$$

$$\frac{\partial o_j}{\partial w_{ij}} = (1.0 - o_j)o_j o_i \tag{5.14}$$

$$\frac{\partial \mathcal{E}}{\partial w_{ij}} = -(d_j - o_j)(1.0 - o_j)o_j o_i \tag{5.15}$$

If the node is not an output node the derivatives can be computed by summing the derivatives already computed for all the nodes $k$ in the layers above $j$ to which the node sends its output $o_j$.

$$\frac{\partial \mathcal{E}}{\partial o_j} = \sum_k w_{jk} \frac{\partial \mathcal{E}}{\partial o_k}(1.0 - o_k)o_k \tag{5.16}$$

$$\frac{\partial \mathcal{E}}{\partial w_{ij}} = (1.0 - o_j)o_j o_i \sum_k w_{jk} \frac{\partial \mathcal{E}}{\partial o_k}(1.0 - o_k)o_k \tag{5.17}$$

It has been proven that if we use $K = M$ output nodes, we set the desired outputs 0.0 and 1.0 and we adjust the parameters of the network according to this criterion, we are minimizing the mean square error between the network output and the a posteriori class probability $Pr(C_j|X)$ [Gish 90, Shoemaker 91].

$$\mathcal{E}_{LMS} \Longleftrightarrow \sum_{j=1}^{K} (Pr(C_j|X) - o_j)^2 \tag{5.18}$$

How small this mean square error can become depends on how complex the network is with respect to the real input data distributions. If we allow the network to be arbitrarily complex, the mean square error should theoretically converge to zero and the network output will approach to the a posteriori class probability [Bourlard 88]. In practice, the convergence of the network will depend on the behavior of the adaptation algorithm.

An interesting point is that we are not limited to encode the output layer with nodes that are meant to approximate a posteriori class probabilities. Instead, we can use this criterion for any particular desired output encoding. In particular, it can be proven that

Indeed, at the beginning of the adaptation algorithm, the distribution is likely to be uniform, and as we get closer to convergence, the distribution is likely to be Gaussian. In the paper by Burrascano, the adaptation rule for $D = \infty$ is presented and discussed

[3] For the sake of simplicity, we do not indicate in the equations the summation over all the training pattern patterns $\sum_X$.

a perceptron with enough hidden nodes can approximate in a Mean Square sense an arbitrary bounded and non-constant function of the input parameters [Hornik 91].

In the experiments reported for this thesis, we try to adapt the network in order to compute a non-linear transformation from an input feature space of dimension $p$ to an output feature space in $K$ dimensions ($K$ does not have to be equal to the number of classes). If the classes are not linearly separable in the input space $p$, we hope that the transformation results in linearly separable classes, in which case we can apply a *linear* method to classify the output sample, assuming that the output distributions after convergence of the adaptation algorithm are, for example, Gaussian.

## 5.3 Recurrent MLPs

So far, we have discussed *feedforward* networks. In a *recurrent* network, a unit $i$ at time $t - d$ can be connected to a unit $j$ at time $t$ ($d$ is a delay, and $j$ might be $i$). Recurrent connections allow the network to capture important information from the temporal variations in the input parameters [Rumelhart 86, Bourlard 89].

Given a network with recurrent connections, the propagation algorithm can be generalized to incorporate time [Rumelhart 86] (pp. 354-361). In general, The LMS error criterion for a training sequence $p$ of $T$ patterns becomes:

$$\mathcal{E}^p_{LMS}(\theta) = \sum_{t=1}^{T}(\sum_{j=1}^{K}(d_j - o_j)^2) \tag{5.19}$$

The adaptation of the network parameters is as follows (see figure 5.3). In a *forward* pass the outputs of all the nodes in the network and the error criterion for the output nodes are computed and saved for each time frame of the sequence, starting from the first frame to the last frame. Then, in a *backward* pass, the gradient for each connection is accumulated from the output layer to the input layer starting from the *last* frame to the *first* frame. In other words, equations 5.15 and 5.17 are extended to the entire sequence of frames starting from the last frame. This way, the changes in each of the weights $w_{ij}$ take into account the fact that the activity of the network at time $t$ might affect its activity at any successive time. When the backward pass reaches the first frame, the changes in the weights can be either applied directly (*on-line* learning) or saved in order to be applied after the presentation of the complete training set (*off-line* or *batch* learning). If the training data is large enough, on-line learning will assure faster convergence [LeCun 89]. The major problem with this procedure is the memory requirements [Rumelhart 86]. Indeed, as we unfold the network in time for the forward phase, we need to save the successive activities for all the nodes, and in the backward phase we have to save all the partial gradient computations. The benefit is that there are no constraints on the network connections.

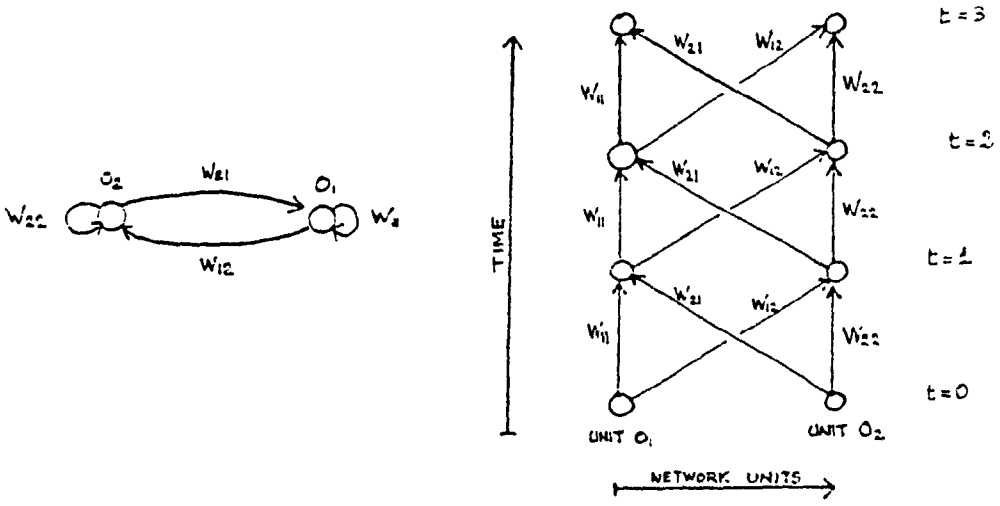



Figure 5.3: Back-propagation in time. (a) A recurrent network. (b) The same network is unfolded in time. Adapted from Rumelhart 86.

## 5.4 Design of Recurrent MLPs for Speech Recognition

There are some important factors that have to be carefully studied for the successful application of recurrent networks to speech recognition. One of them is the design of topologies that integrate information over time and frequency. Several researchers are studying this problem [Bourlard 88, Waibel 89, Jordan 89, Watrous 90, Bengio 90] . Other factors are the selection of the input parameters and the choice of the desired output encoding. In the following we review some issues that have been addressed in the experiments reported in the next chapter.

**Tapped Delay Lines** [Waibel 89, Bengio 90, Hertz 91] One assume that the output of the newtwork at time $t$ depends on the input sequence at time $t + \Delta t \ldots t - \Delta t$. The simplest idea is to consider a sequence of delayed vectors as the input to the network[4] as in figure 5.4. In particular, in this thesis we want to classify stop and nasal phonemes in a /CV/ context. Following the phonetic review of Chapter 2, it seems reasonable to take a decision based on a sequence of input parameters centered on the consonant, including the preceding closure and the following vowel onset. During the closure, the

[4]Interestingly, this structure is analogous to an adaptive equalizer.

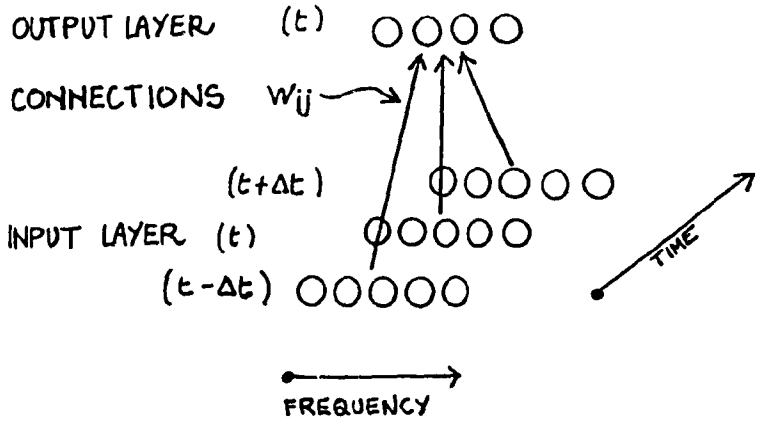



Figure 5.4: A simple time-delay neural network is similar in structure to an adaptive equalizer. Adapted from Hertz 91.

signal contains information about the presence of voicing. At vowel onset the spectrogram and the time/frequency gradient determine the direction of the second formant trajectory.

In general, tapped delays can be inserted between any two layers. This is a simple and very effective idea that solves the problem of integrating *contextual* acoustic information in the classification process. However, it does not solve the problem of *time warping.* When we deal with many speakers and many different speaking rates, the duration of acoustic events, like formant transitions, is extremely variable. The introduction of many tapped delay lines does not allow modelling this variability adequately.

The solution to this problem might be not to expect the MLP to perform time warping. Instead, the MLP can be integrated with a Dynamic Programming (DP) module [Silverman 90], and in particular with a hidden Markov model (HMM) based algorithm [Picone 90]. This methodology is under investigation by many researchers [Robinson 90a, Bridle 90, Bourlard 88, Franzini 90, Bengio 91a]. The dynamic programming module allows for explicit non-linear time warps of the input sequence. At the end of the next chapter we will describe a preliminary experiment that has been run for coupling MLP classifiers with hidden Markov models.

**Recurrent Connections** [Jordan 89, Bengio 90, Watrous 90, Hertz 91] Feedforward networks with tapped delay lines can be enriched with some feedback connections. These connections allow the network to remember its state at preceding instant in time. There-

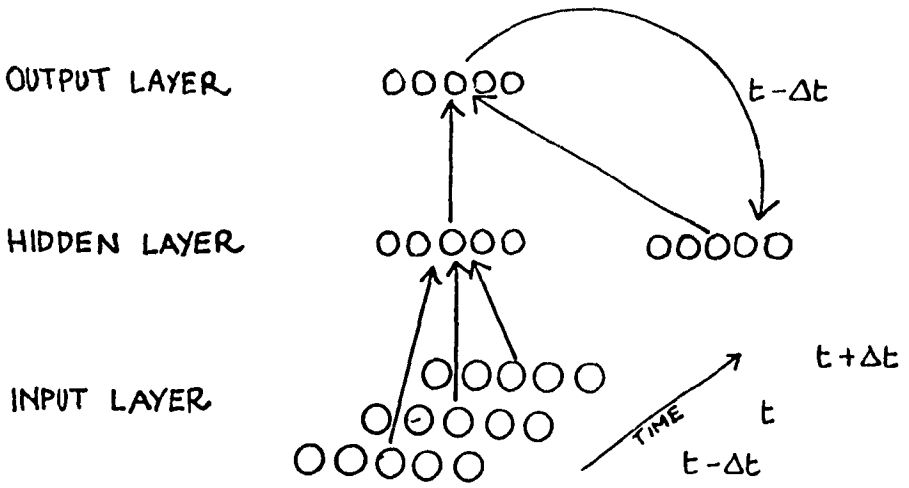



Figure 5.5: A simple recurrent architecture that includes some time delays.

fore, the output of the network at time $t$ does not depend only on the input sequence, but *also* on the past state(s) of the network[5]. In this thesis, we use some specialized hidden units that, at any time $t$, are activated by the output units and/or the other hidden units at time $t-1$ [Jordan 89], as in figure 5.5. Considering the problem of consonant classification, it is possible that relevant acoustic cues appear in the speech signal at different times. Suppose, for example, that the very first frame of the burst of a /k/ is clearly compact around a high frequency peak, and that during the following aspiration the spectrogram is fuzzy. If the network outputs are firing correctly on that first frame, then feedback units corresponding to the feature *velar* will be activated, and they will contribute to sustain the state of the network, until some other acoustic cues, like the fall of the 2nd formant, will reinforce the firing of all the nodes representing phoneme /k/.

**Localized Connectivity Constraints** [Leung 88, LeCun 89, Watrous 90] Fully connected networks with time delays and recurrent connections might be very hard to optimize, due to the very high number of free parameters, especially if the amount of training data is not sufficient. On the other hand, it is possible to design the structure of the network as a function of the features to be recognized, and at the same time, limit the number of free parameters. For the case of consonant sounds, suppose that the input parameter vector is the spectrogram. The first hidden layer can be designed to capture different features from different frequency bands. In this thesis, we divide the auditory

[5]This behavior is similar to systems used in control theory.

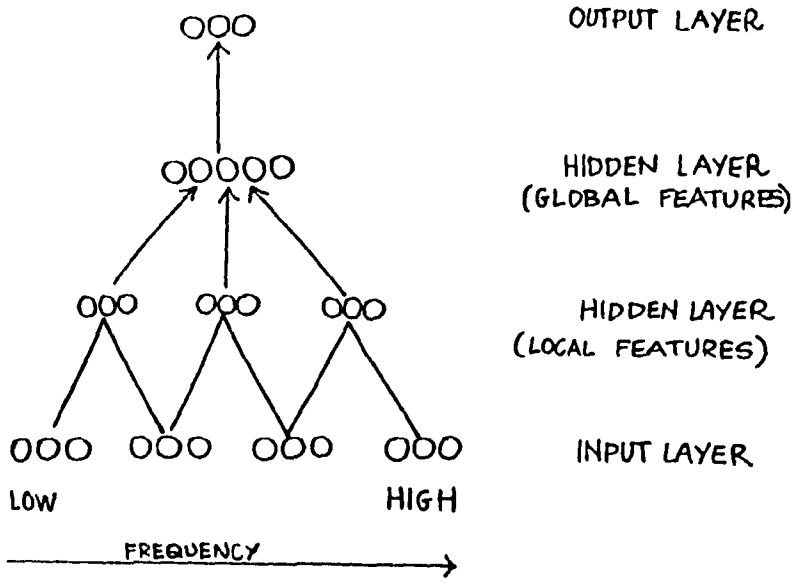



Figure 5.6: A simple network structure with localized connectivity constraints.

scale from 100 to 7900 Hz into 4 regions. Three small groups of hidden nodes are connected to two overlapping regions of the spectrogram each, with time delays (see figure 5.6). Therefore, the first group of hidden nodes computes features related to the low frequencies, the second group computes features related to the mid range, and the third group computes features related to the high frequencies of the spectrogram. If the input vector contains other parameters, such as the time/frequency gradient and other time domain parameters, then other groups of hidden nodes can be allocated to process these new parameters. A second hidden layer will have the role of integrating the information coming from different frequency bands and from different types of input parameters.

**Choice of the Input Acoustic Parameters** Another important aspect to be studied is the choice of the input parameters [Leung 88, Leung 90, Robinson 90b, Bengio 90, Meng 91]. Usually, the input to an MLP used for speech recognition is the auditory spectrogram. An adequate size of the input window should contain enough information regarding the acoustic cues of the features to be recognized. However, as long as the number of connecting weights do not get out of hand, other parameters, such as the one described in Chapter 3, can be *added* to the spectrogram. The parameters that are computed from the spectrogram are correlated to it, and could theoretically be computed by the nodes of the first hidden layer. If we provide more inputs to the network, it might be that the convergence rate of the adaptation algorithm is faster, or that the classification performance improves.

**The Search For an Optimal Desired Output Encoding** [Bengio 90, Bimbot 90, Meng 91] Usually, each class is represented by one node of the output layer. We have seen in Chapter 2 that stop and nasal sounds share several articulatory features, and therefore they share the acoustic cues that are related to theses articulatory features, for example, second formant transitions. Setting one output node per class implies that we teach the network to discriminate all the classes, in spite of the fact that we know that some of the classes share some acoustic cues, and are potentially more confusable. Then, it might be advantageous to encode the desired outputs with binary articulatory features. Moreover, the desired output encoding can be extended to represent phonetic context, simply by multiplying the number of output nodes by the number of relevant contexts. This way, the network can specialize to learn the relationship between different acoustic cues and different articulatory features. Also, the structure of the network can be guided by the choice of the desired output encoding. For example, the second hidden layer can be divided into two groups. One group sends its outputs to the nodes describing the place of articulation, and another group sends its outputs to the nodes describing the manner of articulation and the degree of voicing.

## 5.5 Summary

MLP are non linear networks that seem well suited to perform difficult pattern classification tasks, such as speaker independent consonant classification in continuous speech. They represent a theoretical improvement with respect to linear statistical models, because networks with hidden nodes can be adapted to classify data that do not fit the assumptions of either LDA or Logistic Regression. We have reviewed the LMS error criterion and the back-propagation algorithm. They allow optimizing the parameters of all MLP, as well as of a Logistic Regression machine. For a network with hidden nodes, this adaptive algorithm is not guaranteed to converge to a global optimal value of the parameters, but it is very flexible. It allows the design of a MLP with unconstrained time delays and recurrent links. We have addressed some of the problems that have to be solved when applying MLPs to a speech recognition task. The next chapter reports the experiments that have beeen run on the TIMIT database in order to clarify three key factors in the design of MLP based phonetic classifiers: the choice of the *desired output encoding*, the selection of the *network topology* and of the *input parameters*.

# Chapter 6

# Experiments

After having reviewed the techniques that we used in this thesis, we proceed in describing in detail the experiments that have been run on the TIMIT database. In the last section of this chapter, we will also report some preliminary experiments that have been carried out at our laboratory in order to recognize phonemes in continuous speech.

For the task of classifying the 10 stop and nasal sounds, we have tried to answer the following questions: which *desired output encoding*, which input *acoustic parameter* and which *network topology* give the best performance? Let's discuss in more details the problems that have been addressed:

**Varying the Desired Output Encoding** Usually, the output layer of a MLP has one node per class. As discussed in the preceding chapter, it is possible to encode the output layer so that each node represents a relevant feature, and each phoneme is represented by the activation of several nodes. This distributed representation can then be processed at a higher level of a phonetic decoder. Will this distributed encoding improve the classification capabilities of the MLP?

**Using Different Input Parameters** Is the spectrogram alone sufficient to solve our recognition problem? Will any of the spectrogram based parameters and waveform based parameters presented in chapter 3 be of any help in our classification task?

**Selecting the Network Topology** What is the appropriate topology of the MLP, taking into account the way input parameters and output encoding relate to each other, and the fact that we wish to avoid the problem of optimizing a large number of connecting weights?

| Speakers | Male | Female | Total |
|---|---|---|---|
| Train | 240 | 103 | 343 |
| Test | 50 | 27 | 77 |

Table 6.1: Speaker composition for the 1988 version of the TIMIT database.The speakers whose name starts with a letter between 'a' and 'r' are in the training set

| Tokens | /p/ | /t/ | /k/ | /b/ | /d/ | /g/ | /dx/ |
|---|---|---|---|---|---|---|---|
| Train | 1182 | 1926 | 1399 | 1212 | 1216 | 517 | 1247 |
| Test | 273 | 413 | 347 | 294 | 285 | 122 | 295 |

| Tokens | /m/ | /n/ | /ng/ | /nx/ | /em/ | /en/ | /eng/ | Total |
|---|---|---|---|---|---|---|---|---|
| Train | 1608 | 1359 | 118 | 476 | 29 | 80 | 2 | 12371 |
| Test | 383 | 326 | 29 | 123 | 2 | 14 | 1 | 2907 |

Table 6.2: Frequency of occurrence for each phone considered in the database. Some nasal allophones are very rare.

**Interpretation of the Network Outputs** If there is only one desired output node per class, the classification rule is straightforward. If each class is represented by the activation of several nodes that are meant to be features rather than class probabilities, we can perform classification by interpreting the distribution of the output activations. What are the advantages of using Linear Discriminant Analysis (LDA) for this purpose, compared to a simple rule based on the minimum Euclidean distance between the net output vector and the desired output vector for each class? What happens in practice if we base our classification decision by projecting the net output vectors on the principal components (PCA) instead of the linear discriminant directions?

## 6.1 Experimental Setup

**The TIMIT Database** The database used for all the experiments is extracted from the 1988 version of the TIMIT multi-speaker continuous speech database [Seneff 88b, Zue 90]. For each of the 8 *si* and *sx* type sentence read aloud by 420 different speakers we have considered all the occurrences of the 10 stop and nasal sounds followed by any of the 18

| *parameter* | *definition* | min value | max value |
|---|---|---|---|
| $X^*(f,t)$ | auditory spectrogram | 20 dB | 80 dB |
| $\Delta X/\Delta f$ | frequency regression | -5.0 | +5.0 |
| $\Delta X/\Delta t$ | time regression | -3.5 | 4.5 |
| $\delta^2 X/\delta f \delta t$ | time/frequency derivative | -1.0 | +1.0 |
| $D(t)$ | spectral dissimilarity | 0.002 | 0.300 |
| $F_z(t)$ | Frequency of the zero-crossing | 1000 Hz | 7500 Hz |
| $\Delta F_z(t)$ | time regression of $F_z$ | -350 | +450 |
| $E(t)$ | signal energy | 20 dB | 50 dB |
| $\Delta E(t)$ | time regression of $E$ | -3.5 | +4.5 |
| $V(t)$ | energy in 60-500 Hz band | 20 dB | 50 dB |
| $\Delta V(t)$ | time regression of $V$ | -3.5 | +4.5 |

Table 6.3: Range for all the normalized input acoustic parameters.

vocalic labels (including vowels and diphtongs)[1] Since the 1988 release of the database contains speakers that should be used for training purposes, we had to split them according to a non standard rule. We have decided to put all the speakers whose name started with a letter between 'a' and 'r' in the training set, and all the remaining speakers in the test set. The composition of the training set and of the test set is summarized in table 6.1. The test set consisted of 2,907 tokens with stop and nasal phonemes extracted from 612 sentences spoken by 77 speakers, and represents 23.5% of the data with respect to the training set. The frequency of occurrence of each phoneme is consistent between the training and the test set. Some phonemic labels are very rare.

**Acoustic Analysis** For all the experiments, each token consisted of a sequence of feature vectors computed every 5 msec starting 30 msec before the target phonetic label and ending 30 msec after the label. In other words, the final part of closure and sometimes some other consonant (may be another stop or nasal) or vowel preceding the phoneme and the initial part of the following vowel were included in the analysis of the speech signal. Each computed feature was normalized in order to span the range between $-0.5$ and $+0.5$. The appropriate lower and upper bounds were set after analyzing the histograms and the graphic display for each feature computed for a dozen sentences randomly extracted from the training set. Values outside the chosen range were clipped. The selected range for

[1]Concerning the syllabic and the velar nasals, they *follow* vocalic segments, when they are pronounced in isolated words However, in continuous speech, we found some syllabic and velar nasals that *are followed* by vowels In general, this happens at the boundary between two words Therefore, we included these few tokens in the experimental database

| *localized groups* | *spectrum* | *time domain* | *gradient* |
|---|---|---|---|
| input (63) | 4 by 8 | 7 | 4 by 6 |
| hidden 1 (86) | 3 by 18 | 7 | 3 by 6 |
| *connections* | *frames* | *msec* | *type* |
| input (63) to hidden 1(86) | $t-3, t, t+3$ | 30 | small overlapping groups |
| hidden 1 to hidden 2a(20) | $t$ | 0 | fully connected |
| hidden 1 to hidden 2b(30) | $t$ | 0 | fully connected |
| hidden 2a to output pl(6) | $t, t-1$ | 5 | fully connected |
| hidden 2b to output ma(4) | $t, t-1$ | 5 | fully connected |
| output pl to hidden recpl(6) | $t-1$ | 5 | fully connected |
| output ma to hidden recma(4) | $t-1$ | 5 | fully connected |
| hidden recpl to output pl | $t$ | 0 | fully connected |
| hidden recma to output ma | $t$ | 0 | fully connected |

Table 6.4: Number of nodes in each group, time delays and type of connections between each layer in the default topology.

each one of the computed feature is given in table 6.3.

**Default Network Topology** We describe here the network topology that gave the best performance among the one that we tried. This topology has been used for most of the experiments. Other types of topologies will be discussed in a later section. Figure 6.1 illustrates this topology and table 6.4 summarizes the time delays between each layer. The input to the network at frame $t$ is the sequence of input feature frames $t-3$, $t$ and $t+3$. Therefore, at time $t$ the network looks at a time interval of 30 msec centered around t. The desired output encoding is based on the TIMIT phonetic label of frame $t$. The default topology consists of two hidden layers and an output layer[2].

The first hidden layer is connected to the 3 input feature frames, and is organized into localized groups of a dozen nodes. Each group is connected to a small portion of the input features. For example, the spectrum is divided into 4 regions of 8 nodes, and the 3 groups of the first hidden layer are connected to 2 regions each. The first hidden layer nodes compute discriminant features that are localized in the input feature space. The second hidden layer is connected to all the nodes of the first hidden layer and is organized into two groups. The first group sends its outputs to all the output nodes describing the *place*

[2]The performance of different networks with either one or no hidden layers is presented and discussed in a later section

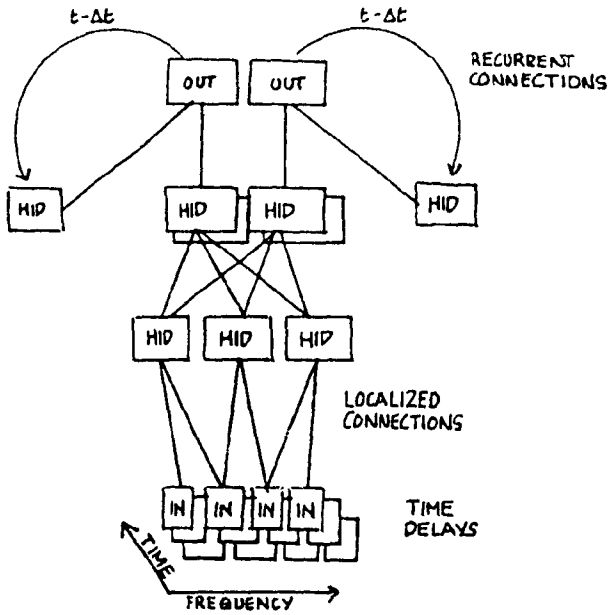



**Figure 6.1: The default network topology. It is a recurrent network with time delays and localized connections.**

*of articulation* at frames $t$ and $t+1$. The second group is connected to all the output nodes describing the *manner of articulation* and *voicing* at frame $t$ and $t+1$. A third group of hidden nodes provides a recurrent link output layer at frame $t$[3]. by receiving as input the output layer at frame $t-1$ and sending back its outputs to the output layer at frame $t$.

For all the experiments involving networks with hidden nodes we tried to keep the total number of connections of the same order of magnitude. All the networks with hidden nodes had about 7500 connections. This complexity was justified because a large amount of training data coming from hundreds of different speakers was available. Networks with less connections performed poorly on the test set, and networks with more than 7500 connections were too slow to train. All the nodes used the symmetric output function, except for the output layer nodes which used the logistic function.

**Adaptation Parameters** For this thesis, we did not search for an optimal adaptation algorithm. Instead, we were interested in setting similar experimental conditions for each network. The initial weights were assigned randomly in the range $-0.20$ to $+0.20$ with a uniform distribution, and then iteratively adapted at the end of the presentation of each token (*on line* updating) One epoch consisted in the presentation of all the 12,731 different training tokens in random order (about 200,000 frames). A different order was

[3]Experiments with and without this recurrent connection will also be described

selected for each epoch For all networks, the learning rate was set to 0 07 for the first epoch ($e = 1$) and then smoothly decreased to 0.00001 according to the following heuristic rule.

$$\eta(e) = 0\ 07\frac{1 + \exp 0.25}{1 + \exp(0.25 \times (e/3))} + 0\ 00001 \quad (6\ 1)$$

A problem arose because of the uneven frame-by-frame distribution of the phonetic classes in the training set. Longer and more frequent phonemes (mostly /p,t,k,m,n/) contributed in a larger part to the error criterion than shorter and less frequent ones The gradient descent adaptation rule tends to minimize the error criterion only for the classes that accounted for the majority of the frames. Therefore, the learning rate was set to a different value for each class $j$, depending on the relative number of frames $f_j$ in the training set[4].

$$\eta_j(e) = \eta(e) \times (\frac{\min_i f_i}{f_j} + \epsilon_j) \quad (6\ 2)$$

The small constants $\epsilon_j$ are introduced for the most frequent classes in order to avoid setting too small learning rates when there is a large difference in class frequency.

All classification tables reported in the next sections refer to a certain epoch This epoch is the epoch for which the network produces the best performance on the *test* set It is a measure of the convergence rate of the adaptation algorithm It tells how many epochs were necessary for the network to generalize adequately Strictly speaking, the correct pattern recognition term for the test set used in this way is *evaluation set* It should be noted that if the adaptation continues after that epoch, usually the error on the training set will decrease (slowly), but the error on the test set is stable, or increases

Most of the experiments were run on a MIPS/RISComputer. One training epoch (i e. 12,731 forward passes and backward passes over an average of 16 5 frames per token) took approximately 120 minutes.

**Classification Rules and Errors** Classification performances were evaluated only for the stop and nasal phonemes, and not for the context phonemes The first classification rule is based on the Euclidean distance between the target outputs for each class and the actual outputs of the MLP. If the highest output is less than a heuristic threshold set a priori to 0.3, all the outputs are assumed to be low, and the frame (or the phoneme) is put in the rejection class[5] The rejection class is represented by all the non stop and non nasal TIMIT labels that precede or follow a target phoneme in a token For phoneme classification each output was averaged for the duration of the TIMIT label

[4]Ralf Kompe, personal communication

[5]Slightly different values of the threshold (between 0 2 and 0 3) did not affect the classification performance

More precisely, the classification rule is the following. Chose the class $j^*$ such that:

$$j^* = \begin{cases} \arg\min_j \sum_k (d_k^j - o_k)^2 & \text{if } \max_k o_k > 0.3 \\ reject & \text{otherwise} \end{cases} \quad (6.3)$$

Most of the experiments use the first classification rule. The second classification rule is based either on PCA or LDA of the network outputs considered as a feature vector. More precisely, the class averages, the sample covariance matrix $W$ and the within-class covariance matrix $S$ are estimated from the network outputs of the training set. Then, the SVD based algorithm described in chapter 4 finds either the PC or the LD directions, and the class averages projected on these directions are computed from the training set. For each test token, the outputs are projected on the estimated PC or LD directions and the classification rule is the following. Choose the class $j^*$ such that:

$$j^* = \begin{cases} \arg\min_j (\sum_i f_i - f_j) \times (\sum_k (v_k^j - u_k)^2) & \text{if } \max_k o_k > 0.3 \\ reject & \text{otherwise} \end{cases} \quad (6.4)$$

In the formula, $f_j$ is the frame frequency of class $j$ in the training set, $v_k^j$ is the projection of the $j$ class average on the $k$th PC or LD direction, and $u_k$ is the projection of the test token outputs. The factor that depends on the class frequencies is introduced for balancing the uneven class distribution that might have affected the computation of the eigenvectors [Dillon 84].

Finally, errors are counted for each stop and nasal frame (and phoneme) that has been incorrectly classified. In some experiments, errors are reported separately for the preceding closure and the following vowel In general, all the networks were "well behaved" concerning their temporal evolution. This means that the output nodes representing stop and nasal sounds were low during the preceding closure and shortly after the beginning of the following vowel. It should be noted that the effective learning rate was set to a very low value before and after the target phoneme, according to the frequency dependent rule given above, so that the networks were trained to discriminate the 10 stop and nasal phonemes, rather than discriminate between target and context.

## 6.2 Comparative Experiments

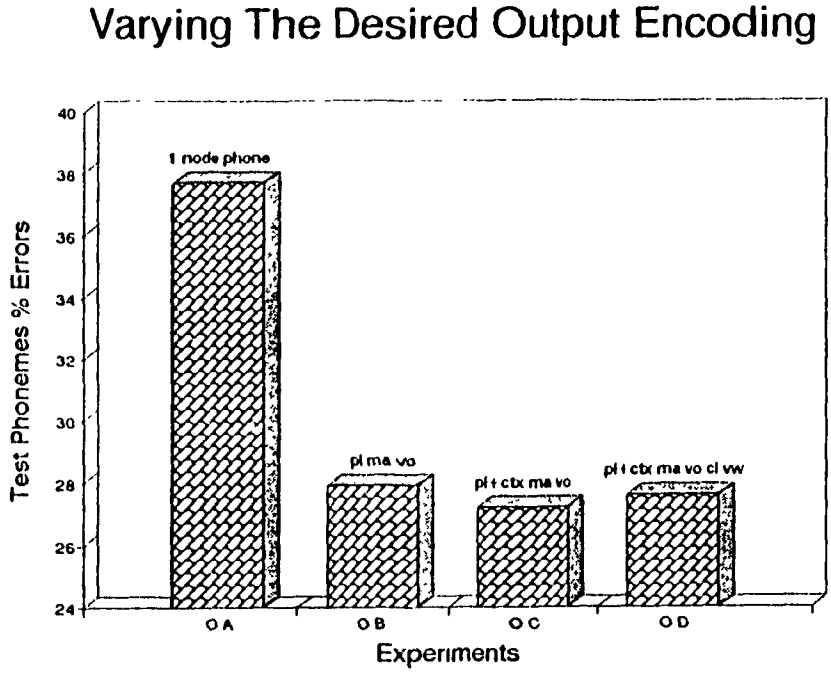
### 6.2.1 Varying The Desired Output Encoding

All the experiments refer to the default network topology. The input to the network consisted of the Bark scaled spectrogram (32 features). Four experiments are reported (Table 6.5)

**O.A One Node Per Phoneme** The output layer represents the 10 phonetic classes /p,t,k, b,d,g,dx, m,n,ng/ with one node per phoneme. The desired output is set to 0.95 when the TIMIT label corresponds to the node label and to 0.05 otherwise, in particular before and after the target phoneme. It took 46 epochs to the network to reach 36.5 % phoneme errors on the training set and 37.7 % phoneme errors on the test set. The performance on the nasal phonemes was particularly deceiving. The training phonemes that were incorrectly classified were 44 % for /m/, 62 % for /n/, and 51 % for /ng/ (albeit less frequent). Also, alveolar phonemes were incorrectly classified Phoneme errors were 33 % for /t/ and 42 % for /d/ on the training set. The same type of confusions occurred for the test set.

**O.B Place Manner and Voicing** The output layer is divided into two groups. One group of 4 nodes represents the place of articulation and the flapped allophone and another group of 3 nodes represents manner and voicing. The 7 nodes are labeled respectively *labial, alveolar, velar, flap, voiced, stop*, and *nasal*. The *voiced* output node is set to high for the voiced plosives, the nasals, and for all the voiced TIMIT labels to the left and to the right of the target phoneme , including voiced closures. The flapped allophones /dx,nx/ are represented by two manner nodes: *alveolar* and *flap*. The output nodes *alveolar, velar, labial* are set to high values for all the consonants that share the same place of articulation, including plosives, nasals, fricatives and liquids (that sometimes are included in the right context of a target phoneme). Specialized nodes in the second hidden layer send their activation either to the *place* nodes or to the *manner and voicing* nodes This way, the hidden nodes can be specialized to discriminate between a few binary and ternary features, rather than discriminate between the ten phonemes After 46 epochs, the network reached 26.0 % phoneme errors on the training set and 27.9 % phoneme errors on the test set. The breakdown of the performance for the training set showed that there was a significant improvement in the classification of nasal and alveolar phonemes For /m/, phoneme errors decreased from 44 to 30 %, for /n/ from 62 to 40.5 % , for /ng/ from 51 to 24 %, for /t/ from 33 to 23.8 % and for /d/ from 42 to 27.6 %

**O.C Place in Context, Manner and Voicing** Encouraged by the performance of network **O.B**, we extended the idea of feature nodes to model the context in which each phoneme appeared. This extension is motivated by the fact that the place of articulation of the following vowel influences the trajectory of the second formant in the transition between the consonant and the vowel (see Chapter 2). The output layer is similar to experiment **O.B**, except that now each place of articulation is represented by two nodes instead of one, depending on the left vocalic context For example, if the token is a /p/ followed by any of the front vocalic segments /ae,eh,ey, ih,ix,iy/ then the node *labial + front* is set to high and the node *labial + non front* is set to an intermediate value (0.55) The total number of output nodes is 10, with 7 nodes describing the place of articulation




| *Output Representation* | *Output Nodes* | *Epoch* | *Train % Errors* | | *Test % Errors* | |
|---|---|---|---|---|---|---|
| | | | *Frames* | *Phones* | *Frames* | *Phones* |
| O.A *1 node/phone* | 10 | 36 | 35.0 | 36.5 | 36.4 | 37.7 |
| O.B *pl ma vo* | 7 | 36 | 26.1 | 26.0 | 28.7 | 27.9 |
| O.C *pl+ctx ma vo* | 10 | 28 | 23.8 | 24.5 | 27.0 | 27.2 |
| O.D *pl+ctx ma vo cl vw* | 14 | 28 | 25.7 | 25.6 | 27.4 | 27.6 |

Table 6.5: Comparing the same classifier with different desired output encoding. *pl*: place, *ma*: manner, *vo*: voicing, *ctx*: front/non front context, *cl*: voiced/unvoiced closure, *vw*: front/non front vowel.

(remind that the flap is represented by one node). Confusions between contexts were not counted. After 28 epochs, the network settled to 24.5 % phoneme errors on the training set and 27.25 % phoneme errors on the test set

**O.D Modeling the Context with More Output Nodes** When we looked at the frame-by-frame and phoneme confusion matrices for all the preceding networks (the tables were similar to Tables 6.9 and 6.10, with more errors), we noticed some persistent confusions. (1) Short unvoiced plosives fell into their voiced cognates and long voiced plosives fell into the unvoiced ones. (2) The place of articulation for nasals was difficult to recognize, even though it improved significantly by using explicit *place* output nodes. Frankly, we did not know how to improve the recognition on the nasals, and we concentrated on the voiced/unvoiced discrimination and on the context. For this new experiment, the output layer is similar to experiment **O.C**, except that more output nodes model explicitly voiced/unvoiced discrimination and the context of the phoneme to be recognized. In particular, there are 7 contextual nodes for the *place* of articulation, and 7 other nodes labeled respectively *nasal, voiced stop, unvoiced stop, silence, voiced closure, front vocalic, non front vocalic.* There are 14 output nodes in total After 26 epochs, the network settled to 25.6 % errors on the training phonemes and 27.6 % errors on the test phonemes. The network was able to discriminate *voiced* vs *unvoiced* closures and *front* vs *non front* vowels roughly 70 % of the time[6] both in the training and in the test set. However, in spite of the recurrent connections and the time delays that provided some contextual information to the network, the performance on the 10 target consonants decreased slightly.

**Summary** The performance of the MLP classifier strongly depended from the choice of the desired output encoding. With respect to the *1 node per phoneme* encoding the errors on the test phonemes decreased of 26 % when different nodes are used for *place*, *manner* and *voicing* Taking into account the right context for each different place of articulation decreased the errors of 28 %. On the other hand, no improvement was found on the classification of stop and nasal sounds if the output layer modeled *directly* the left and right phonetic context with additional nodes, using the same network structure We conclude that if the structure of the network remain the same, the classifier do not take advantage of the addition of different output nodes for the left and right context

[6] In a few cases, we noticed that the TIMIT labeling of the closures was not what we expected The voicing label was probably set depending on the preceding context and a few times did not correspond to what we saw on the waveform Sometimes, a voiced closure was labeled where we could not see any periodicity in the signal, and other times a small but evident periodicity was classified as an unvoiced closure We conjecture that these few errors did not affect the training of the network, although they might have slightly affected the results on the test set

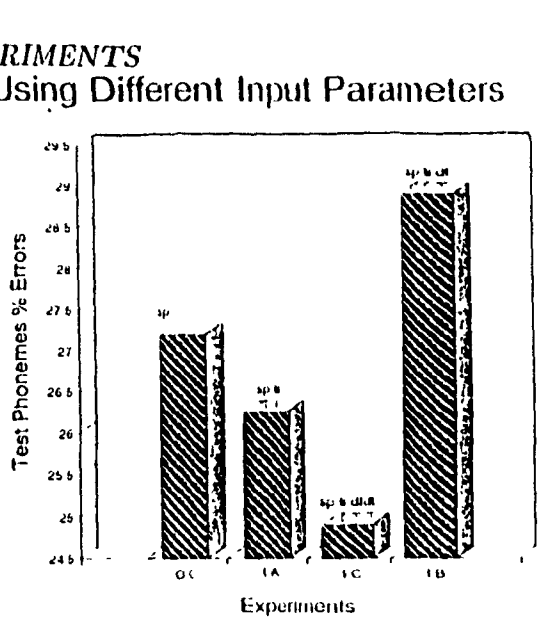
### 6.2.2 Using Different Input Parameters

All the experiments reported in this section refer to the default topology and to the output encoding described in the experiment **O.C**, that is 10 nodes describing distinctive phonetic features, with two different nodes for each place of articulation, depending on the place of articulation of the following vowel. Four experiments are reported (Table 6.6). They are different only with respect to the input parameters. The baseline performance that we tried to improve was represented by network **O.C** that used as input the Bark scaled spectrogram. When we added input features, we tried to keep the complexity of the network around 7500 connections, by diverting some of the nodes of the first hidden layer to look at the added features rather than at the spectrogram, using localized connectivity constraints.

**I.A Adding Temporal Features** For this experiment, we added the following seven parameters to the spectrogram: the energy, the energy in the 60-500 Hz band, the zero crossing rate, their time derivatives, and the spectral dissimilarity function. Each input frame had 39 parameters. Eighteen nodes of the first hidden layer were diverted to look at the added parameters. After 25 cycles the phoneme error rate was 26.25 % on the test set while the errors on the training set were 24.0 %. This improvement was contributed by a small increase in the discrimination of voiced vs. unvoiced plosives, in particular /d,g/ vs /t,k/. We did not analyze the single contribution of each one of the added features, and we conjecture that the main contribution comes from the energy and its time derivative, as well as from the spectral dissimilarity function. These three parameters have a different behavior whether the changes in the structure of the speech signal are fast or slow (see Chapter 3).

**I.B Temporal Features and Frequency Slopes** We added 7 other parameters to the spectrogram and the 7 temporal parameters described above. The new parameters were the spectral slopes, computed on 7 equally spaced frequencies of the spectrum. Each input frame had 46 parameters. We were expecting these new parameters to help in the discrimination between the three places of articulation. Instead, the performance degraded substantially. Indeed, the overall error rate on the test phonemes was almost 29 % after 25 epochs.

**I.C Temporal Features And Time/Frequency Gradient** We dropped the 7 slope coefficients, and we added 21 gradient detectors to the 32 Bark scaled filters and the 7 time parameters. This way, each input feature had 63 acoustic parameters. After 20 epochs, the error rate on the test phonemes decreased to 24.9 %. With respect to using the spectrogram and the time parameters, the performance improved significantly on the phonemes /p,t,m/ that were very frequent, and degraded slightly for /b,d,g,k/. On the




| *Input Parameters* | *Input Nodes* | *Epoch* | *Train % Errors* | | *Test % Errors* | |
|---|---|---|---|---|---|---|
| | | | *Frames* | *Phones* | *Frames* | *Phones* |
| O.C *sp* | 32 | 28 | 23.8 | 24.5 | 27.0 | 27.2 |
| I.A *sp ti* | 39 | 25 | 24.6 | 24.0 | 27.0 | 26.2 |
| I.B *sp ti df* | 46 | 25 | 26.3 | 25.5 | 29.4 | 28.9 |
| I.C *sp ti dfdt* | 63 | 20 | 24.3 | 23.3 | 25.4 | 24.9 |

Table 6.6: Comparing the same classifier with different input parameters *sp*: Bark spectrogram, *ti*: temporal parameters, *df*: frequency derivatives (slopes), *dfdt*: time/frequency derivatives (gradient).

other hand, we obtained the fastest convergence rate of all the experiments that we tried. For comparison, the network with the spectrogram and the time parameters had an error rate of 27.4 % on the test set after 20 epochs.

**Summary** The performance of the MLP classifier depends also from the type of input acoustic parameters. With respect to using the FFT based Bark scaled spectrogram alone, the errors on the test phonemes decreased of 3.6 % when we added some temporal features, and of 8.4 % when we added some temporal features *and* the time/frequency gradient that measures formant transitions. Adding the temporal features *and* the 8 frequency slopes did not improve the performance. Further work is needed to refine the computation of the slope parameters. For example, the values can be normalized with respect to an average slope computed from a very long term spectrum of the speech signal. It is possible that speaker-dependent tilts of the spectrum could be eliminated this way.

### 6.2.3 Selecting The Network Topology

The experiments reported so far refer to the default network topology, which includes two hidden layers and recurrent connections at the level of the output layer. Before choosing this one as the default architecture we have investigated a number of other topologies, and we report here the most interesting results (Table 6.7 and Figure 6.2). For all the experiments reported in this section, the input to the network was the 43 dimension vector with the Bark scaled spectrogram and the 7 energies and time domain parameters, and the output encoding was the same as for the experiments with the input parameters, i.e. it represented context dependent articulatory features (experiments **O.C** and **I.A** to **I.B**).

**T.A Perceptron** This baseline experiment is a perceptron without any hidden layers or recurrent connections. In other words, this is a regression machine that computes semi-linear or logistic discriminant functions, as discussed at the end of Chapter 4. The input layer at frame $t$ are the 3 input frames at time $t-3, t, t+3$ (spanning 30 msec), directly connected to the output layer. Since there are no hidden nodes, the number of connections is reduced to 1180. All the adaptation parameters are the same as for the other experiments. The network is optimized with the usual LMS criterion, and each training epoch requires 1/8 th of the time required for the most complex networks with two hidden layers. After 20 cycles, the perceptron stabilizes to 45 % phoneme errors on the test set. This poor performance is due to fact that there are too many different training and test tokens with respect to the complexity of the network. Most likely, the input distributions are far from being unimodal Gaussian multivariate and/or Binary, that are the only families of distributions that can be optimally discriminated by the perceptron. However, we did not perform any statistical test to confirm the non normality of the input data within the different classes.

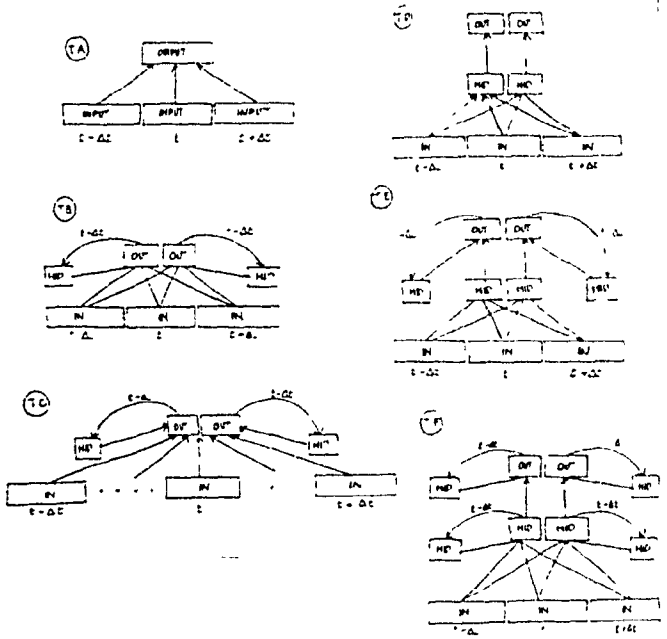


**Figure 6.2: Schematic description of the different network topologies that are reported in the experiments T.A to T.F**

**T.B Perceptron with Recurrence** The perceptron **T.A** is added with a layer of 10 hidden nodes that provide a recurrent connection with respect to the output layer. The hidden nodes at frame $t$ are fed by the output layer at frame $t-1$. The complexity of this simple recurrent network is 1290 connections. The performance on the test phonemes is 43 % errors after 24 epochs. It is interesting to note that when we added the recurrent connections, the frame errors on the test set decreased from 51 % to 47.6 %

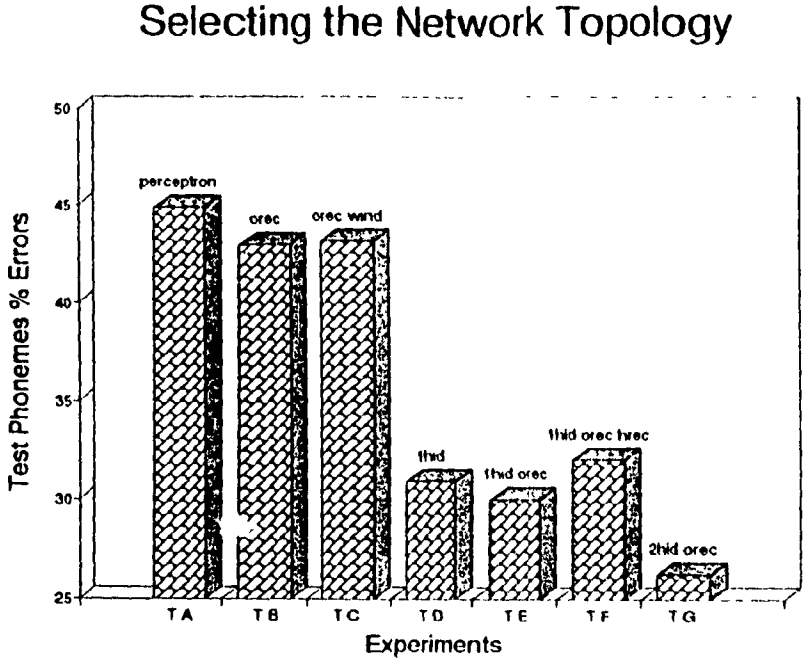
**T.C Larger Input Window** We assumed that at least 3 input frames spanning 30 msec. were necessary to classify the 10 stop and nasal sounds, in order to take into account some contextual information coming from the closure and the vowel. We decided to investigate if more context was useful to perform classification. We took the simple network topology **T.B**, and we enlarged the input window. This way, the input to the recurrent network at frame $t$ is augmented to accommodate 9 frames (45 msec.) centered around frame $t$. The complexity of this network was 3264 weights. After 36 epochs, the performance on the test phonemes is 43.2 % errors. We concluded that the use of a large input window was not useful for the task of classifying stop and nasal sounds with this type of perceptron. We suspect that the larger the window, the more variable the input parameters are, because different speakers talk with different rates. The perceptron does not solve this *time warping* problem.

**T.D One Hidden Layer** We proceeded by adding more and more hidden nodes to the simple perceptron **T.A** in order to model more accurately the unknown complexity of the class distributions. First, we added one hidden layer between the input and the output. This layer is fully connected to the input layer, an is divided into two groups. The first group (24 nodes) sends its activity to the 6 *place of articulation* output nodes. The second group (34 nodes) sends its activity to the 4 *manner of articulation* output nodes. This network had 7414 connections. After 36 epochs, the errors on the test phonemes were 31 %.

**T.E One Hidden Layer and Recurrence** The network **T.D** is added with a group of 10 hidden nodes that provide the recurrent connections to the output layer, just like for the perceptron **T.B**. After 46 epochs, the performance on the test phonemes is 30 % errors. We noticed again that adding the few hundred recurrent connections improved the frame-by-frame performance, from 35 % to 30.7 %. We conjecture that the recurrent connections helped in sustaining a firing node for the duration of a phoneme, in the case that the input is very informative at the phoneme boundary and then becomes less informative, as it happens for a long aspiration or for a nasal.

**T.F One Hidden Layer and More Recurrence** Encouraged by the results obtained by the use of recurrent connections, we added more recurrent nodes. The network **T.E** is added with two groups of hidden nodes that provide the recurrent connections to the two groups of the first (and only) hidden layer. After 46 epochs, this topology yielded 32 % errors on the test set. This performance is worse than using no recurrence at all. We may explain this behavior by the following reflection. After just a few iterations, say 3 or 4, the 10 outputs for a network with at least one hidden layer is correct for more than 50 % of the training frames. Therefore, the hidden nodes that are connected to the output layer receive an *informative* input from just 10 input nodes. On the other hand, the hidden nodes that are connected to the *other* hidden nodes, receive a more complex information from a higher dimensional input vector. We conjecture that the adaptation of these recurrent connections is more difficult with such a topology

**T.G Two Hidden Layers and Recurrence** This is the default network topology. It has two hidden layers, local connectivity constraints for the first layer, and recurrent connections between the output layer and some of the hidden nodes (Figure 6.1) It is the network used in experiment **I.A** that we repeat here for providing a comparison. It should be noted that the complexity of the network is 7114 weights. After 26 cycles it yielded 26.25 % phoneme errors on the test set.




| *Network Topology* | *Weights* ×1000 | *Epoch* | *Train % Errors* | | *Test % Errors* | |
|---|---|---|---|---|---|---|
| | | | *Frames* | *Phones* | *Frames* | *Phones* |
| T.A *perceptron* | 1.2 | 20 | 50.5 | 43.5 | 51.0 | 44.9 |
| T.B + *orec* | 1.2 | 24 | 47.0 | 41.7 | 47.6 | 43.0 |
| T.C + *orec wind* | 3.6 | 30 | 46 5 | 40.9 | 48.1 | 43.2 |
| T.D + *1hid* | 7.5 | 36 | 33.7 | 28.3 | 34.9 | 30.9 |
| T.E + *1hid orec* | 7.5 | 46 | 28.6 | 27.5 | 30 7 | 30 0 |
| T.F + *1hid orec hrec* | 7.5 | 46 | 30.6 | 30.3 | 32.6 | 32.0 |
| T.G + *2hid orec* | 7.1 | 25 | 24.6 | 24.0 | 27.0 | 26.2 |

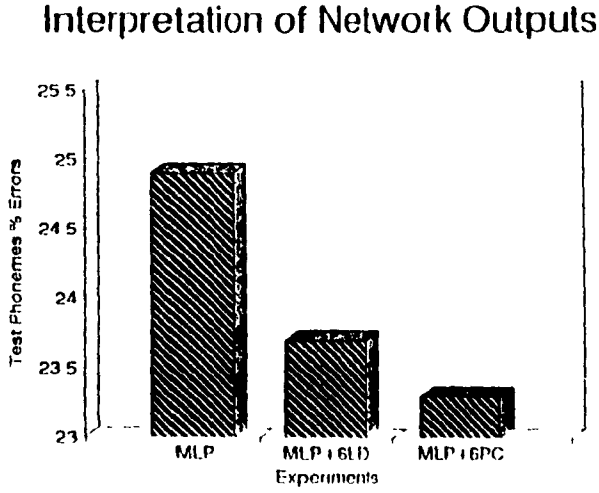
Table 6.7: Comparing different topologies for the same input and the same desired output of the classifier. All networks have time delays. *hrec*: recurrence at the hidden level, *orec*: recurrence at output level, *wind*: larger input window, *1hid*: one hidden layer (fully connected to the input), *2hid*: two hidden layers (local connectivity)

**Summary** The classification performance of the MLP classifier strongly depend on the topology of the network. With respect to a perceptron, by simply adding a layer of hidden nodes the errors on the test phonemes decreased of 31.3 %, and adding some other hidden nodes that provided recurrent connections to the output layer decreased the errors of 33.3 %. We noticed that adding the recurrent connection at the level of the output layer always decreased the frame-by-frame error rate. With respect to using one hidden layer, the use of two hidden layers (keeping a fixed total number of weights) with localized connectivity constraints decreased the errors on the test phonemes of 12.5 %.

After all these experiments, we concluded that the main factor that improved the performance both in terms of phonetic classification and convergence rate of the estimation algorithm was the design of a multi-layered network with localized connections, as suggested by [LeCun 89, Watrous 90]. On the other hand, this *trial and error* approach to network design is extremely time consuming and completely experimental. It was impossible to figure out *a priori* or with less effort which topology would yield the best performance. Also, these findings should not be applied blindly to other phonetic recognition problems. For example, the type of local connectivity constraints as well as the size of the input window and the number of time delays are all parameters that are well suited for the classification of stop and nasal sounds, and we conjecture that they are also adequate for other consonants. Other topologies might be tried for other sounds. It should be noted that other topologies could have been tried also for our problem. For example, topologies with local connectivity constraints with a smaller resolution, or other types of recurrent connections. We just did not have enough computing resources to explore the too many possibilities, and we decided to stop experimenting at a certain point.

### 6.2.4 Interpretation of the Network Outputs

In this section we report about different classification rules (Table 6.8). We considered the network which yielded the best performance (experiment **I.C**), and we classified the test set in three different ways. First, we compared each output vector with the *a priori* targets for each class using the Euclidean distance metric. This method yielded 24.9 % phoneme errors on the test set. Second, we estimated the first 6 linear discriminant vectors and the 10 class targets from the training set statistics, and we compared the outputs with the targets projecting them on the 6 first linear discriminant vectors. This method yielded 23.7 % phoneme errors. Third, we estimated the first 6 principal components from the training set statistics, and this time the phoneme errors were 23.3 %. Using either 5 or 7 vectors yielded a slightly worse result for both methods. We see that both methods yielded a sightly better performance than comparing directly the net outputs to the desired outputs. In particular, the phoneme errors on the test set decreased by 1.8 % using LDA and by 6.1 % using PCA. We conjecture the following simple explanation. When we computed the linear discriminants and the principal components, we adjusted




| Classification rule | MLP | MLP+PC | MLP+LD |
|---|---|---|---|
| Test frame errors | 25.4 | 24.2 | 24.7 |
| Test phoneme errors | 24.9 | 23.3 | 23.7 |

Table 6.8: Performance rates of the same network with different classification rules. **MLP**: Euclidean distance between targets and 10 output nodes. **MLP+PC**: 6 principal components of the net outputs. **MLP+LD**: 6 linear discriminant directions of the net outputs.

| | p | t | k | b | d | g | dx | m | n | ng |
|---|---|---|---|---|---|---|---|---|---|---|
| p | 81.84 | 3.94 | 1.79 | 10.33 | 0.67 | | | | | |
| t | 1.61 | 78.44 | 7.48 | | 10.25 | | | | | |
| k | 2.06 | 5.24 | 81.91 | | 1.25 | 8.81 | | | | |
| b | 6.25 | | | 85.42 | 4.96 | | | 1.49 | | |
| d | | 7.17 | | 5.49 | 73.42 | 3.88 | 3.71 | 1.43 | 2.11 | |
| g | | | 8.51 | 1.55 | 8.82 | 77.09 | | | | 2.32 |
| dx | | | | | 5.10 | | 90.16 | | 3.85 | |
| m | | | | 1.63 | | | | 67.62 | 23.94 | 5.55 |
| n | | | | | 1.51 | | 4.80 | 16.11 | 68.13 | 8.37 |
| ng | | | | | 1.52 | | 1.22 | 2.13 | 34.15 | 60.37 |

Table 6.9: Frame-by-frame Confusion matrix for the best performing network. Classification is based on the 6 principal components of the net outputs. rows: spoken, columns: recognized. Errors that are less then 1.0 % are not indicated.Average error: 24.2 %.

the target outputs in order to match the average behavior of the network on the training set, rather than using some a priori idealistic values. These more realistic targets were also closer to the average outputs for the test set. Apart from slightly increasing the classification performance, using LDA or PCA is convenient because the output of the classifier is more compact, and produces statistically uncorrelated features. This way, we can integrate many network outputs and provide a compact set of uncorrelated features to a statistical phonetic decoder. The use of PCA proved to be more accurate than LDA. The reason can be that the linear discriminant directions are close to the principal components, and the algorithm for finding the principal components was more precise because it did not require the computation of an inverse from the input covariance matrices.

## 6.3 The Best Performing Network

After the description of all the experiments, we summarize the successive steps that took us to the design of the best performing network (experiment **I.C**, classification performed by PCA). The frame-by-frame and phoneme confusion matrices are given in Tables 6.9 and 6.10. Table 6.11 lists the successive refinements that decreased the error rate on the test set.

**Input Parameters** In addition to the Bark scaled spectrogram computed from the FFT every 5 msec (32 triangular filters), we used 7 acoustic parameters related to global spectral and temporal changes of the speech signal (such as the energy and its time

| | p | t | k | b | d | g | dx | m | n | ng |
|---|---|---|---|---|---|---|---|---|---|---|
| p | 73.91 | 1.81 | 1.09 | 20.29 | | 1.09 | | | | |
| t | | 72.97 | 5.98 | | 17.22 | 1.44 | | | | |
| k | 1.68 | 4.75 | 73.18 | | 1.68 | 15.64 | | | 1.40 | |
| b | 4.38 | | | 89.90 | 2.69 | | 1.01 | | | |
| d | 1.05 | 4.18 | | 5.23 | 80.14 | 3 48 | 2 44 | 1 05 | 1.39 | |
| g | | | 4.00 | | 8.80 | 82.40 | | | | 2.40 |
| dx | | | | | 2.03 | | 94.93 | | 3.04 | |
| m | | | | 2.02 | | | | 69.44 | 22.73 | 4 80 |
| n | | | | | | | 7.59 | 13.20 | 69.80 | 7 23 |
| ng | | | | | | | | 4 55 | 38 64 | 56 82 |

Table 6.10: Phoneme confusion matrix for the best performing network. Phoneme classification is based on the 6 principal components of the net outputs. rows: spoken, columns: recognized. Errors that are less then 1.0 % are not indicated.Average error. 23.3 %.

derivative and the spectral dissimilarity function) and 24 other parameters that measure formant peak transitions between 300 and 4000 Hz (the time/frequency gradient). With the same network complexity, adding these two types of parameters decreased the errors by 8.4 %.

**Network Topology** The topology that yielded the best performance among the many that we tried is essentially a network with two layers of hidden nodes, local connectivity constraints, time-delays and recurrent connections at the level of the output layer.

**Desired Output Encoding** Rather than having 1 node per phoneme, the output layer represent distinctive phonetic features. Each different place of articulation is represented by two nodes, depending on whether the following phoneme has a forward or backward place of articulation. This encoding decreased the errors on the stop and nasal phonemes by 28 %.

**Classification** Rather than comparing directly the net outputs with the desired outputs, the output vector is first projected on the first few principal component directions, and then compared to the class averages estimated from the training data

This configuration yielded 23.3 % phoneme errors and 24.2 % frame errors on a test set of 2907 stop and nasal phonemes pronounced in continuous speech by 77 different

| *Type of Network* | *Test % Errors* | |
|---|---|---|
| | *Frames* | *Phones* |
| *output: 1 node/phone* | 36.4 | 37.7 |
| *output: pl ma vo* | 28.7 | 27.9 |
| *output: pl+ctx ma vo* | 27.0 | 27.2 |
| *input: sp ti* | 27.0 | 26.2 |
| *input: sp ti dfdt* | 25.4 | 24.9 |
| *class: 6 pc* | 24.2 | 23.3 |

Table 6.11: Comparing different types of networks for classifying stop and nasal phonemes. The table refers to the test set of 77 different speakers and 2907 tokens. *output*: different outputs, but the same input (the spectrogram). *pl*: place, *ma*: manner, *vo*: voicing, *ctx*: front/non front context. *input*: same outputs (*pl+ctx ma vo*), but different inputs. *sp*: spectrogram, *ti*: temporal features, *dfdt*: time/frequency gradient. *class: 6 pc*: classification of the best net with PCA.

speakers The training data was 12.731 tokens pronounced by 343 speakers. The tokens were extracted from /CV/ segments.

## 6.4 Error Analysis

In this section, we summarize and discuss some general trends that appeared in the behavior of the MLP based classifiers. The figures that are reported in this section refer to the network I.C, when classification is performed by projecting the output vector on the principal component space. Although this is the configuration that gave the best performance, the behavior of the other networks with two hidden layers was qualitatively very similar, but with more errors. The frame-by-frame and phoneme confusion matrices are given in Tables 6.9 and 6.10, while figure 6.3 gives an idea about the performance for each phones

### 6.4.1 Plosive Classification

The MLP classifiers performed better in classifying plosives than in classifying nasals To quantify this statement and to have an idea about the errors on the plosives, we tested the best performing network on the plosive tokens only The test set was composed of 2057 phonemes spoken by the 77 different speakers, with an average of 290 phonemes per class. The frame by-frame errors were 18 % and the phoneme errors were 19 %. Comparing the

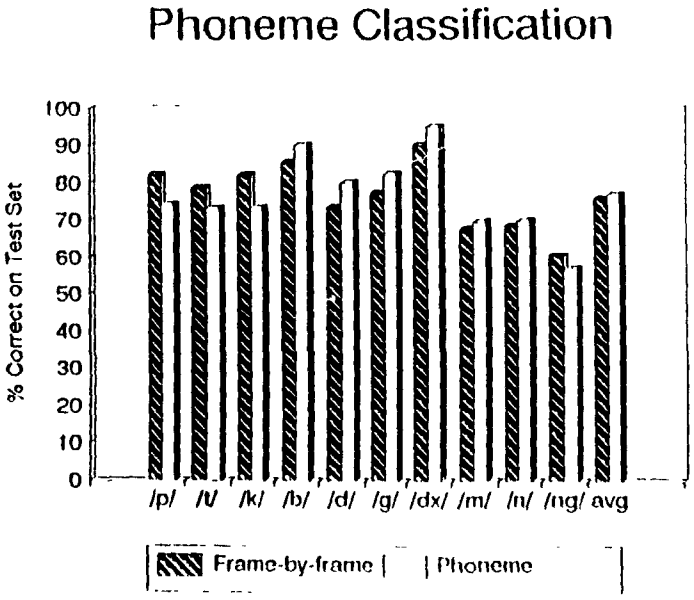



Figure 6.3: Breakdown of the classification performance for each phone. This diagram refers to the best network applied to the test set of 77 speakers. Compare the frame-by-frame performance to the average phonetic performance for /ptk/ vs. /bdg/.

confusion tables for frames and phonemes, we noticed that short unvoiced plosives were classified into their voiced cognates, and long voiced plosives were classified as unvoiced (see also the diagram in figure 6.3). For example, an average of 10 % of the unvoiced frames was classified as voiced, but this figure corresponded to 20.2, 17.2 and 15.6 % of the phonemes /p,t,k/ respectively. We can explain this behavior in the following way. It is possible that the recurrent connections and the time delays contributed to classify short bursts into voiced plosives. This result is encouraging, because it means that the MLP was able to extract a temporal acoustic cue that is correlated to the voiced/unvoiced discrimination. On the other hand, in continuous speech phonemes are pronounced by different speakers at different rates, depending also on the phonetic context (Remind the example of the longer /p/ in *pin* vs. the shorter /p/ in *spin*) When the burst is short, the acoustic analysis using a fixed analysis window will merge spectral information coming from the burst and the vowel. The use of specialized output nodes for *voiced closure* and *silence* in combination with the input parameters that we tried did not improve the performance. Therefore, either we use a spectral analysis method with a higher resolution, or we try to capture more information from acoustic cues of periodicity in the preceding closure. A promising approach is detecting periodicity from the normalized correlation of neighbor segments of the speech signal[7].

[7]Maurizio Omologo, personal communication

| Frames | labial | alveolar | velar | Phones | labial | alveolar | velar |
|---|---|---|---|---|---|---|---|
| labial | 90 8 | 6.8 | 2.0 | labial | 93.4 | 4.4 | 1.6 |
| alveolar | 2 2 | 91.8 | 5.8 | alveolar | 2.3 | 92.7 | 4.8 |
| velar | 1 9 | 7.9 | 89.9 | velar | 1.5 | 8.3 | 89.7 |

Table 6.12 Confusion matrices for the best network used for classyfing the place of articulation of stop phonemes. Classification is performed by selecting 2 linear discriminant directions from the 10 output nodes. Rows: spoken, Columns: recognized. The average % error is 9% frame by frame, and 7.8% per phoneme Results for the test set of 77 speakers.

The 3 way classification of the places of articulation for plosives was very satisfying. On the 77 speaker test set of 573 labials /p,b/, 1000 velars /t,d,dx/ and 484 alveolars /k,g/ the best MLP yielded 9 % frame errors and 7.8 % phoneme errors (Table 6.12). The biggest error was the confusion between the alveolar and the velar place. This good performance can be explained by the fact that on one hand there were thousands of training tokens available for these features, and on the other hand the input acoustic parameters and the MLP input window size were appropriate for capturing the acoustic cues necessary for discriminating the place of articulation.

We were able to compare the performance of this 3-way classifier with the performance of *two* separate classifiers, one used for discriminating the unvoiced plosives only, and another one used for discriminating the voiced plosives only. These MLP classifiers had the same structure as the best performing network. On a limited test set of 48 speakers and 659 phonemes, the /p,t,k/ classifier yielded 11.4 % phoneme errors Using the same 18 speaker and 525 test phonemes, the /b,d,g/ classifier yielded 14.3 % phoneme errors. We conclude that it is advantageous to use the same network for phonemes that share the same place of articulation.

Examining some typical errors committed by the best performing network (table 6.13), we noticed the following trends First, very short plosives were very frequent in the error list Second, the right context that appeared with some regularity were the variations of the central vowel /ix,ax,axr/. Concerning the effect of phoneme duration, we believe that to solve this problem it is necessary to use an acoustic analysis with a higher temporal resolution. Concerning the effect of context, it is possible that the onset spectrum of the central vowel is extremely variable between many different speakers. This fact might have caused a poor discrimination of the preceding consonant, because the classifier was always using contextual information, via time delays and the time/frequency gradient .

| Left | Spoken | Right | Recog. | Dur. | Example File |
|---|---|---|---|---|---|
| vcl | b | iy | p | 6 | dr8/mtcs0/sx352 |
| vcl | d | ix | t | 9 | dr6/mtju0/sx40 |
| sil | d | ix | b | 3 | dr2/mtat1/si779 |
| sil | k | ax | g | 6 | dr5/fskp0/si1098 |
| sil | k | ix | g | 6 | dr3/msfv0/si632 |
| ax | m | ix | n | 9 | dr7/fvkb0/sx79 |
| sil | m | ah | n | 16 | dr2/fscn0/sx266 |
| iy | n | axr | ng | 9 | dr8/mslb0/sx293 |
| sil | n | ow | m | 8 | dr5/fsdc0/si2234 |
| sil | p | axr | b | 3 | dr1/msjs1/sx279 |
| sil | p | ix | b | 3 | dr1/mtrr0/si918 |
| sil | t | ux | d | 6 | dr5/fsjg0/sx40 |
| sil | t | axr | k | 10 | dr4/mteb0/sx413 |
| sil | t | iy | d | 6 | dr3/mtpg0/si2013 |
| ix | nx | ix | dx | 5 | dr5/msas0/si1376 |

Table 6.13: Typical errors of any classifier. For each error, we indicate the left and right context, as well as the phoneme duration in 5 msec frames *sil* means silence or unvoiced closure. *vcl* means voiced closure.

| Frames | labial | alveolar | Phones | labial | alveolar |
|---|---|---|---|---|---|
| labial | 70.5 | 29.5 | labial | 72.0 | 28.0 |
| alveolar | 16.0 | 84.0 | alveolar | 12.0 | 88.0 |

Table 6.14: Confusion matrices for the best network used for classifying the place of articulation of nasal phonemes /m,n/ only. The velar phoneme /ng/ is ignored because it was too rare to be statistically significant. Classification is performed by selecting 1 linear discriminant direction from the 14 output nodes. Rows: spoken, Columns: recognized. The average % error is 23% frame by frame, and 18.7% per phoneme. Results for the test set of 77 speakers.

## 6.4.2 Nasal Classification

We do not examine here the errors on /ng/ because this phoneme was too rare with respect to the other phonemes to be statistically significant. The performance on the nasal phonemes was not as good as for the plosives (see table 6.14). In particular, all the networks had some problems in recognizing the place of articulation for /m/ and /n/, regardless of the fact that a large amount of training data was available. The best network performed almost 70 % recognition for /m/ and /n/. This figure should be compared with about 80 % classification rates for the 7 plosives. It should be noted that using the *same* output nodes for the stop and the nasal phonemes with the same place of articulation significantly improved the performance on the *nasal* phonemes. Evidently, at the boundaries between the consonant and its context, the acoustic cues for the place of articulation are similar between the plosive and the nasals. This is not surprising, since the vocal tract configuration is the same for plosives and nasals with the same place of articulation, and therefore the resonant frequencies contributed by the vocal tract must be the same. Therefore we doubt that a separate *specialized* network for the nasal phonemes would perform better than our 10 phoneme classifier.

When we classified the net outputs taking into account *only* the 396 /m/ and 553 /n/ test phonemes, we obtained 18.65 % confusions between the two places of articulation, (vs. 7.8 % for the 3 plosive places) and 22.9 % frame errors (vs. 9 % for the plosives) Results are reported in Table 6.14. These errors did not seem to depend on the left or right phonetic context (see Table 6.4.1). This performance can be explained by the fact that the acoustic cues for nasality are strongly evident on the spectrogram, and they tend to *hide* the acoustic cues for the place of articulation This problem is addressed in [Zue 79].

## 6.5 Integrating MLP Classifiers in a Phonetic Decoder

In this last section, we report some preliminary experiments that have been carried out at our laboratory in order to integrate this type of MLP classifiers in an acoustic-phonetic decoder for continuous speech recognition. The experiments reported in this section are the result of a team effort. They have been carried out by Yoshua Bengio, Ralf Kompe and myself.

### 6.5.1 Methodology

At the same time that we were experimenting different MLP classifiers for stop and nasal sounds, a preliminary experiment has been performed using a prototype system based on the integration of MLP classifiers with HMMs.

The methodology that we applied is the following. State of the art acoustic-phonetic decoders for speaker independent continuous speech recognition are based on Dynamic Programming [Silverman 90], and in particular on a statistical model of the speech signal. For a first order hidden Markov model (HMM), each consecutive acoustic frame is considered as the independent outcome of an unobservable probabilistic process (see, for example [Picone 90]). To limit the number of statistical parameters to be estimated, such systems require a constrained set of input parameters. As a consequence important phonetic information may be lost in the acoustic front-end of a recognition system. Recently, there have been several approaches for integrating MLP classifiers with HMMs (among others, we refer to [Austin 91, Bridle 90, Bourlard 88, Franzini 90])

We advocate here the use of a *hybrid* acoustic-phonetic decoder, in which one or many MLPs classify the incoming speech signal in terms of relevant articulatory features describing the place and manner of articulation and a degree of voicing. The combined MLP outputs provide a sequence of observation vectors for a phonetic decoder based on a continuous densities hidden Markov model [Picone 90].

### 6.5.2 Experimental Setup

To limit the computational complexity of the experiments to a reasonable figure the following 8 class problem has been considered: */p,t,k, b,d,q,dx, all other phonemes/* For this problem *si* and *sx* sentences from regions 2, 3 and 6 of the 1988 version of the TIMIT database were used, with 1080 training sentences and 224 test sentences, 135 training speakers and 28 test speakers respectively. The train/test splitting rule was the same as for the other experiments.

### 6.5.3 MLP Perform Feature Extraction

The experimental system was the following. Rather than having a single MLP that computes the vector of acoustic parameters, we have two networks, MLP1 and MLP2. They are initially trained to perform broad classification (MLP1) and plosive classification (MLP2) respectively The input acoustic parameters, the topology, and the desired output encoding of these networks are similar to the MLP for stop and nasal classification, and their outputs describe articulatory features such as the place and manner of articulation and a degree of voicing.

The broad classification network (MLP1) has been developed by Ralf Kompe and has 5 outputs corresponding to five broad categories: non-nasal sonorant, nasal, plosive, fricative, and silence. Details about this classifier can be found in [Bengio 91b]

The plosive recognition network (MLP2) was developed by myself and had an output layer with 16 nodes describing the place and manner of articulation of plosives with two instantiations of each place nodes depending on whether the following phoneme has a forward or backward place of articulation. The desired output encoding was similar to the one described for the stop and nasal experiment **O.D**. In particular, the output nodes were labeled as follows. For this experiment, we considered four different places of articulation (labial, alveolar, velar, and flapped alveolar) with two different nodes for each place, The remaining eight nodes were labeled: unvoiced plosive, voiced plosive, vocalic front, vocalic non-front, liquid, fricative, nasal, silence. The network topology was similar to the default topology for the stop and nasal classifiers, with two hidden layers, localized connectivity, and recurrent connections at the output layer. The input parameters for each frame were the Bark scaled spectrogram, the 7 temporal parameters, the gradient detectors and the slope coefficients. At that time, this was assumed to be the best performing configuration.

PCA was applied to the outputs of the combined MLP. This transformation was performed by multiplying the combined MLP output vector by a rectangular matrix Each column of the matrix was one of the principal components. This matrix multiplication has been implemented as a single-layer linear perceptron, called SLP. The SLP outputs a vector of 8 parameters (the MLP outputs projected on the first 8 principal components). The overall structure MLP1+MLP2+SLP is equivalent to a complex time delay multi-layer perceptron with three hidden layers of loglinear units and linear output units.

### 6.5.4 HMM perform phonetic decoding

For each sentence to be decoded, the 8 parameter vector is the sequence of observation vectors for a Continuous Densities HMM, with 11 left-to-right models. In order to improve the modeling of the rejection class four different models were considered: nasals, fricatives, non-nasal sonorants, and silence The recognition results are obtained by merg-

ing these four subclasses, such that the total number of classes to recognize is 8. Each phonetic model had 14 states, 28 transitions, 3 self loops, without explicitly modeling the state duration, and tied output probability distributions with 3 basic different distributions characterizing the beginning, middle and final part of each segment. Each of these distributions was modeled by a Gaussian mixture with 5 densities. The Gaussian covariance matrices were assumed to be diagonal since the parameters were initially principal components.

### 6.5.5 Training of the hybrid MLP-HMM system

First, the parameters of the two MLP classifiers were optimized *separately*, using sentence tokens for MLP1 and phoneme tokens for MLP2. Then, two iterations of the *Forward-Backward* optimization algorithm [Picone 90] were run in order to estimate the state transition probabilities and the parameters of the tied output distributions for each state, using all the training sentences. This is a *maximum likelihood estimation* procedure, in which one tries to estimate the HMM parameters in order to maximize the likelihood of the observed sequence of vectors given the constraints of the model. Last, two iterations of a *global* optimization procedure were run using the training sentences one by one. This procedure has been developed by Yoshua Bengio and is described in details in [Bengio 91a, Bengio 91c]. Very briefly, it allows a joint optimization of the parameters of the continuous densities HMM *and* of all the connecting weights of the combined structure MLP1+MLP2+SLP. For each one of the SLP outputs and each one of the training sentences, it is possible to compute a gradient derived from the likelihood of the correct sequence of phonetic models. This gradient replaces the derivative of the LMS error criterion with respect to each SLP output unit. Then, the gradient can be transmitted to all the connecting weights using the back-propagation algorithm. Figure 6.4 illustrates the outputs of the combined MLP and the output phoneme string obtained by Viterbi decoding of the HMM for a short segment of a test sentence.

### 6.5.6 Preliminary Evaluation

The performance of the hybrid system was compared with that of a rough post processor applied to the outputs of the MLPs. A simple algorithm assigned a symbol to each output frame of the MLPs by comparing frame by frame the target output vector with actual output vector. It then smoothed the resulting string to remove very short segments and merged consecutive segments that had the same symbol. In order to evaluate the advantages of using MLPs as sequences of observations for the HMM, the same HMM models were used to perform recognition, but using a standard set of acoustic parameters: 8 cepstral coefficients computed from the Bark scaled spectrogram, 8 cepstral time derivatives, the signal energy and its time derivative (18 inputs).

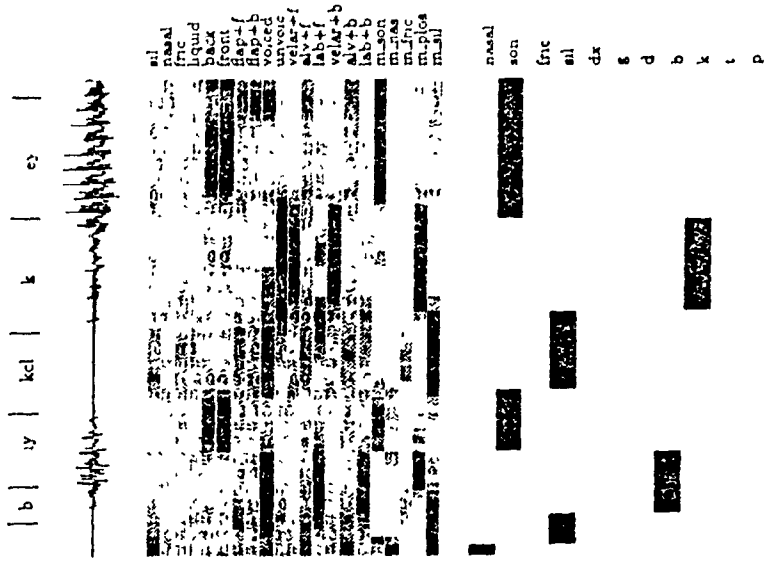



Figure 6.4: Top: signal. Word spoken: "became". Middle: MLP outputs related to distinctive phonetic features. Bottom: Output string decoded by the Viterbi algorithm from the trellis of HMMs.

---

The comparative results for the three systems are summarized in Table 6.15. Performance rates were evaluated for the 8 classes, *including* plosives and the rejection class. It should be noted that for the majority of the frames of any sentence the correct phoneme belonged to the rejection class. For this reason, rather than looking at the absolute values, it is interesting to compare between different configurations of the hybrid systems.

The overall recognition rate (100% - %deletions - %substitutions) for the 8 classes with the hybrid system after two training iterations is 90% on a total of 7214 phonemes, and its accuracy (100% - %deletions - %substitutions - %insertions) is 86% Note that this is an improvement over the performance obtained with a HMM trained without global optimization (86% recognition and 81% accuracy), The MLPs alone yielded 85% recognition but only 53% accuracy, because of the high number of insertions of plosive segments (32%). The HMM eliminates most of these insertions because it optimizes its parameters over entire sentences, rather than over short segments of speech. In addition the HMM provides more appropriate target values for the outputs of the MLP. It shoul be noted that the use of a hybrid system decreased the performance on the plosives, that were less frequent than the other classes

This preliminary experiment was very encouraging. It demonstrated a very promising way to integrate one or many MLP classifiers into a statistical phonetic decoder,

| *Phonetic Decoder* | *% rec* | *% ins* | *% del* | *% subs* | *% acc* |
|---|---|---|---|---|---|
| *MLPs alone* | 85 | 32 | 0.04 | 15 | 53 |
| *HMMs alone* | 76 | 6.3 | 2.2 | 22.3 | 69 |
| *MLPs+HMM* | 87 | 6.8 | 0.9 | 12 | 81 |
| *MLPs+HMM+global opt.* | 90 | 3.5 | 1.4 | 9.0 | 86 |

Table 6.15: Performance evaluation: MLPs alone, HMMs with standard cepstrum, delta cepstrum, energy and delta energy input features, MLPs with HMMs, and with global optimization. The task was to recognize 8 classes */p,t,k, b,d,g, dx, other phonemes/* in continuous speech. The table refers to the test set of 28 speakers and 224 sentences.

---

based on specific knowledge in experimental phonetics. For a complete evaluation of this methodology, one has to complete an experiment involving the full set of American-English phonemes. Future work will be devoted to this problem.

# Chapter 7

# Conclusion and Open Problems

A main advantage of MLPs is the possibility to classify sequences of many input parameters with much flexibility. No statistical assumption is made about the nature of the inputs, and sufficiently complex MLPs perform well on difficult pattern recognition problems such as speaker independent phoneme classification in continuous speech. Specialized ANNs can be integrated into a statistical phonetic decoder which model the temporal structures of the speech signal. As a first step towards the design of such a hybrid phonetic decoder. we studied the problem of classifying stop and nasal sounds. The comparative experiments reported in this thesis showed that key factors for improving the performance of such classifiers are the proper choice of the *input parameters*, of the *internal topology* and of the *desired output* representation. These parameters strongly depend on the acoustic correlates of the phonemes to be recognized and are inspired by experimental studies in Phonetics and by signal processing strategies. In general, different parameters will be applied to different classes of phonemes. From the results of the many experiments reported in this thesis, we can draw a number of concluding remarks that should be useful for future research.

**Varying the input parameters** MLP are able to cope with several inputs per frame, and no assumption needs to be made about their statistical distribution, therefore one is relatively free to choose acoustic parameters, based on phonetic and signal processing knowledge. For the problem of classifying stop and nasal sounds, with respect to using the spectrogram alone, the addition of some global temporal and spectral parameters and of a gradient detector that measures formant transitions decreased the error rate by 8.4 % on the test set, with respect to using the spectrogram alone.

Concerning the Bark-scale spectrogram, we have used a computationally inexpensive and straightforward method based on the Fast Fourier Transform (FFT). We believe

there are intrinsic limitations which such a method, due to the fixed window of the speech signal. At the expenses of a somewhat heavier computational load, one can devise the following simple modifications to the used algorithm. First, a more accurate frame-by-frame analysis can be performed by averaging the FFT computed for some (say 3) slightly shifted windows. This way, one can limit the effects of moving the analysis window asynchronously from the pitch period. Second two (or more) different analysis window length can be used, a shorter (about 10 msec) and a longer one (from 25 to 35 msec). The resulting spectrogram can integrate both analysis lengths [Cheung 91]. This way, the analysis would be more informative regarding both short (wide band) and long (narrow band) acoustic events, like high frequency bursts and formant transitions. Both these modifications would be easy to implement and should be investigated, since they would not require specific hardware configurations. Alternatively, one can use either a pitch-synchronous analysis step and window length, or a bank of nonlinear time domain passband filters, provided that the more complex computations are carried out in an efficient manner.

Also, other time domain parameters can be added to the spectrogram, like the normalized correlation between neighboring segments of speech [Medan 91] in order to capture some acoustic details that cannot be found in the Bark scaled spectrogram. Such parameters will certainly help in the voiced/unvoiced discrimination.

**The search for an adequate topology** Some experiments have been run comparing different network topologies, with or without recurrence and with or without hidden layers. Two hidden layers and some recurrent connections between the hidden layer and the output layer were found to be necessary for improving the classification performance on the stop and nasal sounds. In order to represent acoustic phonetic details it was necessary to consider some context in the input, but a large context was found to be impractical for the type of network that we used. The MLP that we used were not able to deal with the greater variability in the input that was introduced by a larger window. In other words, it was not possible for a MLP with time delays and a large input window to perform time warping and generalize adequately[1] on a multi-speaker task. To solve this problem, we advocate the use of a hybrid phonetic decoder in which the time warps are managed by a statistical algorithm.

The use of a *divide and conquer* or *modular* approach is believed to be advantageous for the classification performance and for the reduction of the number of free parameters of the classifier. This approach can be applied in terms of localized connectivity between the input layer and the hidden layer, and in terms of subdividing a recognition task into several, hopefully easier subproblems. In spite of the simplifications introduced by the *modular* approach, the complexity of the considered MLP is still impressive. In general

[1] Yoshua Bengio, personal communication

about 7100 connections were necessary to discrimmate adequately the 10 stop and nasal sounds on a test set of 77 speakers.

The search for a more adequate and more compact topology for speaker independent phoneme recognition in continuous speech is still an open problem. Research should be directed towards non linear networks that model the temporal organization of the speech communication process with specialized architectures. These architectures will capture more information than a few contextual input frames.

**What Are The Best Target Outputs ?** The classification performance on stop and nasal sounds improves significantly when the output of the ANN are distinctive phonetic features, rather than phonetic labels (28 % decrease of the error rate). This output representation has been extended to model the effects of coarticulation by simply multiplying the number of output nodes for the place of articulation, depending on the right context.

In addition, Principal Component Analysis of the output vector provides a small number of statistically uncorrelated output features, as well as a set of target values that match the behavior of the network for the training data more realistically than some *a priori* desired values This small set of uncorrelated features can be processed by a statistical phonetic decoder (i.e. continuous densities hidden Markov models).

In general, MLP can effectively compute some non linear features from the speech signal. In the input feature space, the classes are not linearly separable, since a simple perceptron (i.e. a logistic regression machine) is able to classify correctly only 55% of the test samples. Thus, we can look at the MLP as a non linear transformation of the input feature space that yields another feature space in which the classes are more likely to be linearly separable To achieve linear separability, it would not be necessary to set the network to the same desired output for *all* the training tokens belonging to the same class. Rather, one could try to adapt the targets to the training data.

Along this approach, the reported plosive recognition experiment coupled the MLP outputs to the observation sequence of a continuous densities hidden Markov model. The global optimization of all the parameters of the system permitted to adjust the targets of the MLP in order to better represent the network output distributions for the training data. A different approach can be to search for a new analytical form of the error criterion, based on information theory [Gish 90], but without constraining the outputs to estimate a posteriori class probabilities.

# Bibliography


[Anderson 72] Anderson J.A. (1972) Separate sample logisitic discrimination *Biometrika.* Vol. 59 No. 1. pp. 19-35.

[Atal 89] Atal B.S. (1989) A model of LPC excitation in terms of eigenvectors of the autocorrelation matrix of the impulse response of the LPC filter, *Proc Int. Conf. on Acoust. Speech and Signal Proc. (ICASSP-89)* Glasgow UK. pp. 45-48.

[Austin 91] Austin S., Makhoul J., Schwartz R , and Zavaliagkos G (1991) Continuous speech recognition using segmental neural nets. *Proc DARPA Speech and Natural Language Workshop.* Morgan Kaufman

[Bakamidis 90] Bakamidis S.,Dendrinos M. and Carayannis G. (1990) SVD Analysis by synthesis of harmonic signals, *I.E.E.E. Trans. on Signal Processing*, Vol 39, No. 2, February, pp. 472-476.

[Bartkova 87] Bartkova K. and Jouvet D. (1987) Speaker Independent Speech recognition Using Allophones. *Proc. Int. Conf. of Phonetic Sciences* Tallin Ussr. Vol. 5. pp. 244-247.

[Becker 89] Becker S. and LeCun Y. (1989) Improving the convergence of back-propagation learning with second order methods. in *proc of the 1988 Connectionist Models Summer School.* Pittsburgh 1988. Touretzky, Hinton and Sejnowski ed. Morgan Kaufmann, San Mateo, CA pp. 29 37

[Bengio 90] Bengio Y., Cardin R and De Mori R. (1990) Speaker independent speech recognition with neural networks and speech knowledge, in *Advances in Neural Information Processing Systems II*, Denver 1989, Touretzky D.S ed, Morgan Kauffman, san Mateo, Ca, pp 218-225.

[Bengio 91a] Bengio Y., DeMori R., Flammia G., and Kompe R (1991) Global optimization of a neural network - hidden markov model hybrid To appear in *Proc. Int Joint Conf. on Neural Networks (IJCNN-91)* Seattle, USA, July 1991

[Bengio 91b] Bengio Y., De Mori R., Flammia G., and Kompe R (1991) Phonetically motivated acoustic parameters for continuous speech recognition using artificial neural networks To appear in *Proc Eurospeech-91*, Genova, Italy

[Bengio 91c] Bengio Y. (1991) *Artficial Neural Networks and Applications to Sequence Recognition* PhD Thesis (to appear) School of Computer Science. McGill University Montreal Canada

[Bimbot 90] Bimbot F. Chollet G. Tubach J.P (1990) Phonetic features exraction using time-delay neural networks. *Proc Int. Conf. on Spoken Language Proc.* Kobe. Japan. pp. 665-668.

[Blumstein 79] Blumstein S.E. and Stevens K.N. (1979), Acoustic invariance in speech production: Evidence from measurements of the spectral characteristics of stop consonants. *Jour. of Acoust. Soc. Amer. (J.A.S.A.)* Vol. 66, No. 4, pp 1001-1018.

[Blumstein 80] Blumstein S.E. and Stevens K.N. (1980), Perceptual invariance and onset spectra for stop consonants in different vowel environments. *J.A.S.A.* Vol. 67, No. 2, pp. 648-661.

[Bocchieri 86] Bocchieri E.L. and Doddington G.R. (1986) Frame-specific statistical features for speaker independent speech recognition. *IEEE Trans. on Acoust. Speech and Signal Proc. (ASSP*, Vol. 34, No. 4, pp. 755-764.

[Borden 84] Borden J.B. and Harris K.S. (1984) *Speech Science Primer: Physiology, Acoustics and Perception of Speech*, 2nd edition, Williams and Wilkins publishers, Baltimore, MD.

[Bourlard 88] Bourlard H. and Wellekens C.J. (1988) Links between markov models and multi-layer perceptrons. In S. Touretzky ed. *Advances in Neural Information Processing Systems 1*, Morgan Kauffman. pp. 502-510.

[Bourlard 89] Bourlard H. and Wellekens C.J. (1989): Speech dynamics and recurrent neural networks. *Proc. ICASSP-89* Glasgow. UK. pp. 33-36.

[Bridle 90] Bridle J.S. (1990) Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In Touretzky ed. *Advances in Neural Information Processing Systems 2*, Morgan Kauffman. pp 211-217.

[Brown 87] Brown P.F (1987) *The Acoustic-Modeling problem in Automatic speech recognition*, PhD Thesis, Department of Computer Science, Carnagie-Mellon University, CMU-CS-87-125.

[Burrascano 91] Burrascano P. (1991) A Norm selection criterion for the generalized delta rule, *IEEE Trans. on Neural Networks*, Vol. 2, No. 1, January, pp. 125-130.

[Bush 83] Bush M A., Kopec G.E., and Zue V.W (1983) Selecting acoustic features for stop-consonant identification, *Proc ICASSP-83.* Boston pp. 712-745.

[Cheung 91] Cheung S. and Lim J.S. (1991) Combined Multi-Resolution (Wideband / Narrowband) Spectrogram. *proc ICASSP-91.* Toronto. CA pp. 457-460.

[Chistovich 79] Chistovich L.A., Sheikin R. and Lublinskaja V.V. (1979) Centres of gravity and spectral peaks as the determinants of vowel quality. in *Frontiers of Speech Communication Research*, Lindblom and Ohman eds Academic Press. London. pp. 143-157.

[Cole 86] Cole R., Stern R., and Lasry M (1986) Performing fine phonetic distinctions: templates vs. features. in *Invariance and Variability in Speech Processes*. Perkell and Klatt eds. Erlbaum. Hillsdale NJ. pp. 325-341.

[Cox 70] Cox D.R. (1970) *The Analysis of Binaty Data*. Methuen pub. London.

[DARPA] *Speech and Natural Language*. Proc. *DARPA Workshops* Published in 1990 and 1991 by Morgan Kaufman, San Mateo, CA.

[Davis 80] Davis S. and Mermelstein P. (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on ASSP*. Vol. ASSP-28. pp. 357-366.

[De Mori 87] De Mori R., Lam L. and Gilloux (1987) Learning and plan refinement in a knowledge-based system for automatic speech recognition, *IEEE trans. on Pattern Anal. and Mach. Intell. (PAMI)*, Vol. PAMI-9, No. 2 (march), pp. 289-305.

[Deng 90] Deng L , Lennig M., and Mermelstein P. (1990) Modeling microsegments of stop consonants in a hidden Markov model based word recognizer. *J.A.S.A* Vol. 87, No. 6, pp. 2738-2747.

[Deng 91] Deng L and Erler K. (1991) Microstructural speech units and their HMM representation for speech recognition. *Proc. ICASSP-91*. Toronto. CA. paper 56.S3.11.

[Dillon 84] Dillon W.R. and Goldstein M. (1984) *Multivariate Analysis. Methods and Applications* , John Wiley & Sons. New York

[Duda 73] Duda R.O. and Hart P.E. (1973) *Pattern Classification and Scene Analysis* Wiley and Sons, New York

[ElJaroudi 91] El-Jaroudi A. and Makhoul J. (1991) Discrete all-pole modeling *IEEE Trans. on Signal Processing*. Vol. 39. No. 2 pp 411-423

[Fant 70] Fant G. (1970) *Acoustic Theory of Speech Production*, Mouton & Co, The Hague, 2nd edition.

[Fant 73] Fant G. (1973) *Speech Sounds and Features*, MIT press, Cambridge, Mass

[Franzini 90] Franzini, M., Lee K-F. and Waibel A (1990) Connectionist Viterbi training: a new hybrid method for continuous speech recognition in Proc *ICASSP-90*, Albuquerque, NM pp. 425-428

[Fujimura 62] Fujimura O. (1962) Analysis of nasal consonants. *J A S A* Vol 34 pp 1865-1875

[Jakobson 61] Jakobson R., Fant G., and Halle M (1961) *Preliminaries to Speech Analysis: The Distinctive Features and Their Correlates* MIT Press, Cambridge Mass.

[Jordan 89] Jordan M.I. (1989) Serial Order. A Parallel, Distributed processing Approach. in *Advances in Connectionist Theory. Speech* Elman and Rumelhart eds Erlbaum pub. Hillsdale

[Greenberg 88] Greenberg S. ed. (1988) Special issue of the *Journal of Phonetics* on "Representation of Speech in the Auditory Periphery". Vol. 16.

[Gish 90] Gish H. (1990) A probabilistic approach to the understanding and training of neural networks classifiers.*Proc ICASSP-90* Albuquerque NM. pp. 1361-1364.

[Glass 86] Glass J.R. and Zue V.W. (1986). Detection and Recognition of Nasal Consonants in American English. *proc. ICASSP-86.* Tokyo Paper 51 5.1. pp. 2767-2770.

[Gupta 87] Gupta V.N. Lennig M. Mermelstein P. (1987) Integration of Acoustic Information In a Large Vocabulary Word Recognition *proc ICASSP-87* Dallas, TX. pp. 697-700.

[Handel 89] Handel M. (1989) *Listening: The Perception of Auditory Events.* MIT Press, Cambridge, Massachussetts.

[Hermansky 90] Hermansky H. (1990) Perceptual linear predictive (PLP) analysis of speech. *J.A.S.A.* Vol. 87. No. 4. pp. 1738-1752.

[Hertz 91] Hertz J. Krogh A. and Palmer R.G. (1991) *Introduction to the Theory of Neural Computation* Lecture Notes Volume 1, Santa Fe Institue, Addison-Wesley Publishing Company.

[Hinton 87] Hinton G.E. (1987) *Connectionist Learning Procedures* Technical Report CMU-CS-87-115. Carnagie Mellon University Computer Science Departement.

[Hornik 91] Hornik K (1991) Approximation Capabilities of Multilayer Feedforward Networks *Neural Networks* Vol. 4. pp 251-257.

[KewleyP 82] Kewley-Port D (1982) Measurement of formant transitions in naturally produced stop consonant-vowel syllables *J.A.S.A.* Vol 72, No 2, August 1982, pp. 379- 389

[KewleyP 83] Kewley-Port D (1983), Time-varying features as correlates of place of articulation in stop consonants, *J.A.S.A* Vol. 73, No 1, pp 322 335

[Klema 80] Klema V C and Laub A J (1980) The singular value decomposition its computation and some applications, *IEEE trans on Automatic Control,* Vol. AC-25, No 2, pp 164-176

[Kolen 90] Kolen J.F and Pollack J.B. (1990), Backpropagation is Sensible to Initial Conditions, *Complex Systems*, Vol. 4, pp. 269-280.

[Kuroski 84] Kuroski K. and Blumstein S. (1984) Perceptual integration of the murmur and formant transitions for place of articulation in nasal consonants. *J.A.S.A.* Vol. 76. No. 2. August 1984. pp. 383-390.

[LeCun 89] Le Cun Y. (1989) Generalization and network design strategies. in *Connectionism in Perspective.* Pfeifer, Schreter, Fogelman, and Steels eds. North Holland. pp. 143-155.

[Lee 89] Lee K.F. (1989) *Automatic Speech Recognition: the Developement of the SPHINX System.* Kluwer Academic Publishers, Norwell, Mass.

[Leung 88] Leung H.C. and Zue V.W. (1988) Some Phonetic Recognition Experiments using Artificial Neural Networks. *Proc. ICASSP-88.* New York. pp. 422-425.

[Leung 90] Leung H.C. and Zue V.W.(1990) Phonetic Classification Using Multi-Layer Perceptrons, *Proc. ICASSP-90*, Albuquerque, NM. Sec. 10.10, pp. 525-528.

[Lindblom 83] Björm Lindblom (1983) Economy of Speech Gestures. in *the Production of Speech* MacNeilage ed. Springer-Verlag, New York. pp. 218-245.

[Lippmann 87] Lippmann R.P. (1987) An introduction to computing with neural nets. *IEEE ASSP magazine.* April 1987. pp. 4-22.

[Lippmann 89] Lippmann R.P. (1989) Review of neural networks for speech recognition. *Neural Computation* Vol. 1, pp. 1-38.

[Makhoul 75] Makhoul J. (1976) Linear prediction: a tutorial review. *Proc. IEEE* Vol. 63 pp. 561-580.

[Makhoul 90] Makhoul J., Jelinek F., Rabiner L., Weinstein C. and Zue V. (1990) Spoken Language Systems. *Annual Review of Computer Science*, Vol. 4, pp. 481-501

[McDonald 90] McDonald R O'N (1990) *A Neural Network Approach to Phoneme Recognition* S M. Thesis, Center for Supercomputing Research and Development University of Illinois. Urbana. IL

[Medan 91] Medan Y., Yair E., Chazan D. (1991) Super Resolution Pitch Determination of Speech Signals *IEEE Trans. on Signal Processing* Vol. 39 No. 1 pp 40-48

[Meng 91] Meng H M and Zue V W (1991) Signal Representation Comparison for Phonetic Classification. *Proc ICASSP-91* Toronto CA pp 285-288.

[Mermelstein 77] Mermelstein P. (1977) On detecting nasals in continuous speech *J A S A.* Vol 61. pp. 581-587

[Nathan 90] Nathan K.S. and Silverman H.F. (1990) High-Resolution Characterization of Formants in Vowel-Consonant Transitions *Proc ICASSP-90* Albuquerque. NM. pp 353- 355

[Nathan 91] Nathan K.S. and Silverman H.F (1991) Classification of Unvoiced Stops Based on Formant Transitions Prior to Release. *proc ICASSP-91* Toronto, CA. pp. 445-448.

[OShaughn 87] O'Shaughnessy D. (1987) *Speech Communcation, Human and Machine* Addison Wesley Pub.

[Picone 90] Picone J. (1990) Continuous Speech Recognition Using Hidden Markov Models. *IEEE ASSP magazine* Vol. 7, No. 3. July 1990 pp 26 41

[Press 78] Press S.J. and Wilson S. (1978) Choosing between logistic regression and discriminant analysis. *Jour. of Amer. Stat. Ass.* Vol. 73. No 364. pp 699-705.

[Press 88] Press W.H. et al. (1988) *Numerical recipes in C*, Cambridge University Press, pp. 60-71.

[Rabiner 78] Rabiner L. and Shafer R. (1978) *Digital Processing of Speech Signals* Prentice-Hall, Englewood Cliffs, N.J.

[Robinson 90a] Robinson T. and Fallside F. (1990) *Phoneme recognition from the TIMIT database using Recurrent Error Propagation Networks.* Technical Report CUED/F-INFENG/TR 42, Cambridge University Engineering Department, Cambridge, England.

[Robinson 90b] Robinson T. and Fallside F. (1990) A Comparison of preprocessors for the Cambridge Recurrent Error Propagation Network Speech Recognition System. *Proc. Int. Conf Spoken Lang. Proc. (ICSLP-90).* Kobe, Japan pp. 1033-1036

[Rumelhart 86] Rumelhart D.E. Hinton G.E. and Williams R.J. (1986) Learning internal representation by error propagation in *Parallel Distributed Processing* Rumelhart et al. eds MIT Press, Vol 1, pp 318 362

[Seitz 90] Seitz P.F. McCormick M.M. Watson I.M and Bladon R.A (1990) Relational spectral features for place of articulation in nasal consonants *J.A.S.A.* Vol 87 No 1. January. pp. 351-358.

[Seneff 88a] Seneff S. (1988) A Joint Synchrony/Mean-Rate model of auditory Speech processing. *Journal of Phonetics* Vol 16, pp. 55-76

[Seneff 88b] Seneff S and Zue V. (1988) Transcription and alignement of the TIMIT database, in J S Garofolo ed *Getting started with the DARPA TIMIT CD-ROM An acoustic phonetic continous speech database* Nat Inst of Stand and Tech. Gaithersburgh, MD.

[Shoemaker 91] Shoemaker P.A. (1991) A Note on Least-Squares Procedures and Classification by Neural Network Models, *IEEE Trans. on Neural Networks* Vol. 2, No.1, January, pp. 158-160.

[Shwartz 85] Schwartz R.M., Chow Y.L., Kimball O.A. Roucos S., Kimball M., and Makhoul J. (1985) Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continous Speech. *Proc. ICASSP-85.* Tampa. FL. pp. 1205-1208.

[Silverman 90] Silverman H.F. and Morgan D.P. (1990) The Application of Dynamic Programming to Connected Speech Recognition. *IEEE ASSP magazine*, Vol. 7, No. 3, July 1990, pp 6-25

[Sontag 91] Sontag E.D. and Sussmann H.J. (1991) Back-Propagation Separates where Perceptrons Do. *Neural Networks.* Vol. 4. pp. 243-249.

[Stevens 74] Stevens K.N. and Klatt D. (1974) Role of formant transitions in the voiced-voiceless distinction for stops *J.A S.A.* Vol. 55, No. 3. March 1974. pp. 653-659.

[Stevens 75] Stevens K.N. (1975) The potential role of properties detectors in the perception of consonants, in *Auditory Analysis and Perception of Speech* G. Fant and M.A. Tatham ed., Academic Press, pp. 303-330.

[Stevens 80] Stevens K.N. (1980) Acoustic correlates of some phonetic categories. *J.A.S.A* Vol. 68. No. 3 September 1980 pp. 836-842.

[Stevens 81] Stevens K.N. and Blumstein S E. (1981) The search for invariant acoustic correlates of phonetic features, in *Perspectives on the study of speech* P.D. Eimas and J.L. Miller ed., Lawrence Erlbaum ass pp. 1-38.

[Stevens 83] Stevens K N. (1983) Design Features of Speech Sound Systems, in *The Production of Speech*, MacNeilage P.F ed , Springer-Verlag, New York pp 248-261.

[Stewart 73] Stewart G.W (1973), *Introduction to Matrix Computations.* John Hopkins University Press

[Suomi 85] Suomi K (1985) The vowel-dependence of gross spectral cues to the place of articulation of stop consonants in CV syllables, *Journal of Phonetics* Vol 13, pp 267-285.

[Vernooij 89] Vernooij G J , Bloothooft G , and Van Holsteijn Y (1989) A Simulation Study on the Usefulness of Broad Phonetic Classification in Automatic speech Recognition, *Proc ICASSP-89* Glasgow. UK pp 85-88

[Waibel 89] Waibel A., Hanazawa T , Hinton G , Shikano K , and Lang K (1989) Phoneme recognition using Time-Delay Neural Networks,*IEEE Trans on ASSP*, Vol 37, pp 328-339

[Watrous 90] Watrous R L. (1990) Phoneme Discrimination using Connectionist networks. *J.A.S.A.* Vol. 87. No. 4 pp. 1753-1772.

[Weigelt 90] Weigelt L.F., Sadoff S.J. , and Miller J.D. (1990) Plosive/fricative distinction: The voiceless case, *J.A S.A* , Vol 87, No. 6, pp. 2729-2737

[White 90] White G.M. (1990) Natural language understanding and speech recognition. *Communications of the A C.M.* Vol. 33 No 8. August 1990. pp 72-82.

[Widrow 60] Widrow B. and Hoff M.E. (1960) Adaptive Switching Circuits In *1960 IRE WESCON Convention Record*, part 4, pp 96-104, New York, Reprinted in Anderson and Rosenfeld ed. (1988) *Neurocomputing Foundations of Research* MIT Press, Cambridge, Mass.

[Young 90] Young S.R. (1990) Use of dialogue, pragmatics and semantics to enhance speech recognition, *Speech Communication*, Vol 9, pp. 551-564.

[Zue 79] Zue V. and Laferriere M. (1979) Acoustic study of medial /t,d/ in American English, *J.A.S.A.* Vol. 66, No. 4, pp. 1039-1050.

[Zue 85] Zue V (1985) The Use of Speech Knowledge in Automatic Speech Recognition. *Proc. of the IEEE*, Vol. 73, No 11. pp. 1602-1615

[Zue 90] Zue V., Seneff S., and Glass J. (1990) Speech Database Development. TIMIT and Beyond. *Speech Communication*, Vol. 9, pp. 351-356.

[Zwicker 80] Zwicker E. and Terhardt E. (1980) Analytical expressions for critical band rate and critical bandwiths as a function of frequency. *J.A.S.A.*, Vol 68, No. 5, pp. 1523-1525.