Learning Discriminant Narrow-Band Temporal Patterns for Automatic Recognition of Conversational Telephone Speech

by

Barry Yue Chen

B.S. (University of Maryland at College Park) 1997

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Nelson Morgan, Chair
Professor Michael Jordan
Professor John Ohala

Spring 2005

The dissertation of Barry Yue Chen is approved:

Chair Date

Date

Date

University of California, Berkeley

Spring 2005

Learning Discriminant Narrow-Band Temporal Patterns for Automatic Recognition of Conversational Telephone Speech

Abstract

Learning Discriminant Narrow-Band Temporal Patterns for Automatic Recognition of Conversational Telephone Speech

by

Barry Yue Chen

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Nelson Morgan, Chair

Typical automatic speech recognition (ASR) systems extract features from the full spectrum of speech over relatively short time spans (from about 25 milliseconds to approximately 100 milliseconds). They rely on the short-term spectral envelope of speech for modeling speech sounds. This dependence on the short-term spectral envelope of speech may account for the fact that ASR systems still fall short of human recognition ability. Variabilities in the speech signal come from environmental sources (such as noise and reverberation) as well as from the speaker herself/himself (such as accent and speaking style). These variabilities create difficult problems for typical ASR systems relying on the short-term spectral envelope of speech. This thesis further explores the extraction of discriminant speech information from long-term narrow-frequency energy trajectories of speech. These long-term narrow-frequency energy trajectories stretch over 500 milliseconds of speech and span critical-bandwidths. Previous work on extracting information from these long-term trajectories led to the development of a neural network architecture called Neural TRAP [52, 112]. Neural TRAP consists of two stages of multi-layer perceptrons (MLPs), each of which is a single hidden layer fully-connected MLP. The first stage is trained to estimate the phone posterior probabilities within each critical-band, while the second stage uses the critical-band level phone probabilities to come up with an overall estimate of the full spectrum phone posterior probabilities. This system was competitive to conventional ASR systems, but in combination with conventional systems, Neural TRAP

significantly improved ASR performance. We extend the Neural TRAP work along two major directions in this thesis. First, we develop two new Neural TRAP-like architectures that extract different critical-band level information. The first new architecture, Hidden Activation TRAP (HAT), is like Neural TRAP except that instead of using the outputs of the critical-band MLPs, which estimate critical-band level phone probabilities, it uses the outputs of the critical-band hidden units, which represent probabilities of certain discriminant energy trajectories. The second new architecture, Tonotopic Multi-Layer Perceptron (TMLP), has the same network topology as HAT, but the critical-band hidden unit parameters and the discriminant energy trajectories that they model are not constrained to learn critical-band level phone posteriors, rather they are free to learn useful critical-band discriminant patterns for the estimation of the full-band phone posteriors. The second major extension in this thesis is the integration of the long-term narrow-band systems with a conventional ASR system for the recognition of conversational telephone speech (CTS). By augmenting conventional short-term features with features derived from a combination of phone posteriors estimated by the long-term systems and by more conventional intermediate-term systems, we achieve word error rate reductions of about 9% relative on CTS, which is considered impressive for this task.

Professor Nelson Morgan
Dissertation Committee Chair

For my wife Joan
my mother Ping
my father Chaing
and my sister Anne

Contents

List of Figures

List of Tables

Acknowledgments

I'm blessed to be a long-time member of the speech group at the International Computer Science Institute where I've had many helpful interactions with a group of talented and supportive colleagues. I would like to first thank Jeff Bilmes, Dan Ellis, Eric Fosler-Lussier, Dan Gildea, Brian Kingsbury, Nikki Mirghafori, Warner Warren, and Su-Lin Wu for their early encouragement and guidance to a young pup in the group. A special thanks to my "Jedi Master" and friend Mike Shire who taught me everything I needed to know to be a productive member of the speech group. Thanks to all the old Aurora Team members from whom I learned a lot and shared food and laughs with: Stephane DuPont, Carmen Benitez, the wonderfully kind and generous folk from Hynek's OGI lab: Pratibha Jain, and Sunil Sivadas. Thanks to all my coworkers, administrative staff, computer gurus, and visitors who have helped and continue to make ICSI the greatest place to do research: Jitendra Ajmera, Jeremy Ang, Xavier Anguera, Jennifer Aube, Don Baron, Sven Behnke, Sonali Bhagat, Kofi Boakye, Hannah Carvey, Ozgur Cetin, Patrick Chew, Jaclyn Considine, Rajdip Dhillon, Michael Ellis, Lila Finhill, Arlo Faria, Marc Ferras, James Fung, David Gelbart, Dan Gillick, Steve Greenberg, Frantisek Grezl, Andy Hatch, Micha Hersch, Julie Higashi, Theresa Hilaire, Leah Hitchcock, Yan Huang, David Johnson, Darcel Jones, Michael Kleinschmidt, Konsta Koppinen, Yang Liu, Javier Macias, Scott McCommas, Jenny Nguyen, Chris Oei, Carmen Palaez, Albert Park, Mary Penilla, Barbara Peskin, Tuomo Pirinen, Madeline Plauche, Maria Quintana, Carrie Schwalbe, Liz Shriberg, Stephen Stafford, Diane Starr, Panu Sumervuo, Litonya Walker, and Britta Wrede. Thanks also to Shawn Chang who helped enormously by modifying Quicknet for the TMLP as well offering lots of helpful advice.

This work benefited greatly from all my discussions with Qifeng Zhu, whom I recommend to anyone looking for a wonderful teammate. Thanks to my super cool proofreaders, Chuck Wooters and Adam Janin, for many useful comments and suggestions. Chuck, thanks for encouraging my writing progress as well as for the many corny jokes. Thanks also to Andreas Stolcke for all his kind help with the SRI recognizer.

I would like to thank my committee members for their time and guidance. Thanks to Michael Jordan for his inspirational teaching in my favorite class at Berkeley, Graphical Models. Thanks also to John Ohala for graciously being my outside reader. For all his encouragement, advice, and ideas, I'd like to thank Hynek Hermansky. To my fearless

leader, benefactor, and adviser, Nelson Morgan, I give many thanks. Thanks for making ICSI such a great place to learn and grow, and for your support and guidance over the years.

Thank you, Mommy, Daddy, and Anne for not asking too many "When are you graduating?" questions and for all of your support and love. Thank you, my dear sweetie Joan for all the hugs and kisses that washed all the graduate school worries away and gave me the energy to persevere.

This work would not have happened without the generous support from DARPA which funded me through the DARPA EARS Novel Approaches Grant: No. MDA972-02-1-0024.

Chapter 1

Introduction

One of the funnier moments in a Star Trek movie happened when the crew of the starship Enterprise attempt to save the Earth by traveling back hundreds of years to the late 1980's in search of whales. To fulfill their quest, these futuristic travelers must deal with "primitive" technologies. They were used to teleporting from one side of the planet to another, and now they had to ride the buses across town. In one scene, the chief engineer of the Enterprise sits in front of a computer, picks up the mouse and uses it as a microphone to talk with the computer. To his dismay, the computer does not even respond with a beep or a boop. In his time, automatic speech recognition (ASR) had been long solved, and people could interact with computers by simply talking. In our time, ASR, the process by which a computer takes what a user says and translates it into text, remains a challenging area of research.

1.1 ASR: Not a Solved Problem

You wouldn't think that ASR still poses a challenge considering that today there are powerful ASR products in the market capable of performing a variety of tasks including dictation, command and control, and automated telephone call center routing. These products recognize speech "pretty well" under ideal conditions, where an ideal condition is one in which the recognizer was trained to deal with. However, when compared with humans, ASR systems still perform much more poorly. Furthermore, under non-ideal conditions, performance of current state-of-the-art speech recognizers degrades sharply.

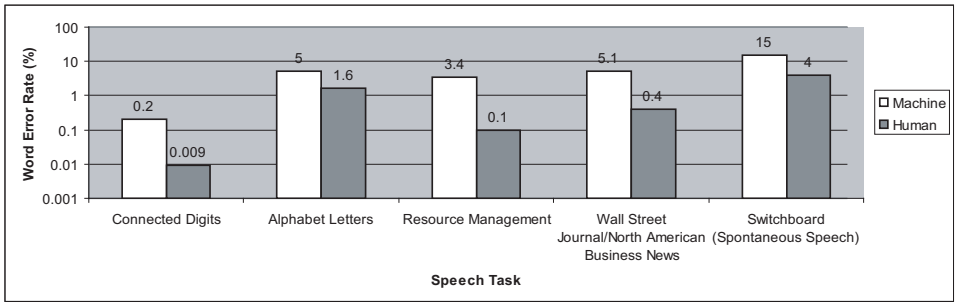

Figure 1.1: A comparison of word error rates for machines and humans from [84]. When possible, machine word error rates are updated from a variety of sources [76], [36], [104], and [133].

In 1997 Richard Lippmann surveyed the state-of-the-art performance of ASR compared with human performance on various speech recognition tasks [84]. Figure 1.1 compares the word error rates[1] of machines versus that of humans on these tasks. Where possible, I have updated the machine word error rates to reflect some of the progress that has been made since 1997. Speech recognition performance by machines is still much worse than that by humans.

Another way to evaluate the quality of current ASR performance is to compare using ASR as an input method against other conventional input methods such as typing. Speech researcher Roger K. Moore has measured the number of *correct* words per minute[2] from typing and from a speaker dependent large vocabulary continuous speech recognition (SD LVCSR) system like the ones you can buy from ScanSoft or IBM for home use. He found that an expert QWERTY typist can type up to 70 correct words per minute, while the SD LVCSR system can only output about 30 correct words per minute [95]. It is interesting to note that while the number of words per minute from the SD LVCSR system is about 107, the number of correct words per minute drops down to 30. The explanation for this drop is that the ASR system makes mistakes which takes time for the user to correct, thus greatly reducing the number of correct words per minute. Consumers expecting ASR dictation products to be as good as a secretary may be sorely disappointed. The word error rates of ASR systems are still too high.

The errors made by ASR systems come from two major sources of variability:

[1] Word error rate is a typical performance measure for ASR systems and is defined to be the total number of errors (word substitutions, insertions, and deletions) divided by the total number of words.

[2] This is a measurement of how many of the desired input words can be inputted per minute.

environmental variations, and speaker variations. Environmental variations can consist of sounds picked up by the microphone that happen in the background, e.g., a barking dog, a noisy computer fan, or even other people gossiping and laughing. We refer to this kind of environmental variation as "background noise". Another kind of variation caused by the environment is reverberation or the echo effect. Sound waves coming from a speaker, not only travel to the microphone directly from the speaker's mouth, but also indirectly from reflections off walls and other objects. These sound reflections cause significant performance degradations in ASR systems. Speaker variation can happen both within a specific speaker (at different times) or across different speakers (i.e., from one speaker to another speaker). An example of within speaker differences occurs when a person speaks at different rates, possibly because of time pressures or varying levels of excitement. People also tend to talk differently depending on the audience. For example, when speaking formally to a boss or a superior, one may want to enunciate and use a more sophisticated vocabulary. In contrast, when speaking to a friend, a person is more likely to use slang and talk casually. A person's speech may also sound differently when he/she is sick or has just woken up. The previous examples highlight variations caused by vocabulary change as well as variations in the quality of the speech signal. Cross speaker variability may occur in the pitch of their voices, the accents in their speech, the rhythm and pace of their delivery, and all the same variations that can happen within the same speaker. All these sources of speaker variability, as well as the environmental variability mentioned above, contribute to making speaker independent large vocabulary continuous speech recognition such a challenging task. Conversational telephone speech (CTS), consisting of recordings of people talking over the phone about everyday topics, represents one of the biggest challenges facing ASR today. One of the goals of this thesis is to address this challenge and improve performance on CTS. Before we outline other goals of this thesis, let us first discuss the motivation for our approach starting with a brief explanation of conventional ASR systems.

1.2 Typical ASR Systems

A typical state-of-the-art ASR system tries to find the best sequence of words given a set of acoustic observations and modeling parameters (e.g., grammar, pronunciation, and phonotactics). Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denote a sequence of N acoustic

observation vectors or "feature" vectors, and let $\mathbf{W} = \{word_1, word_2, \ldots, word_M\}$ denote a sequence of M words. The ASR system outputs the word sequence, $\mathbf{W}*$, that maximizes the following equation:

$$\mathbf{W}* = \underset{\mathbf{W}}{\operatorname{argmax}}\, P(\mathbf{W}|\mathbf{X}, \theta) \tag{1.1}$$

where θ represents all the trained model parameters. Instead of building an all-encompassing model of $P(\mathbf{W}|\mathbf{X}, \theta)$, we can factor this probability into several smaller models. First, let us consider words as a sequence of sub-word units or states. The most common choice for these sub-word states are sub-phones which are portions of phones[3]. Without loss of generality, we will denote this sequence of sub-word states by a sequence of phones: $\mathbf{Q} = phone_1, phone_2, \ldots, phone_K$. Equation 1.1 can be rewritten as Equation 1.2 by summing over all the possible phone sequences, $\mathbf{Q}$, that together make up the word sequence, $\mathbf{W}$.

$$\begin{aligned}
\underset{\mathbf{W}}{\operatorname{argmax}}\, P(\mathbf{W}|\mathbf{X}, \theta) &= \underset{\mathbf{W}}{\operatorname{argmax}} \sum_{\mathbf{Q}} P(\mathbf{W}, \mathbf{Q}|\mathbf{X}, \theta) && (1.2)\\
&= \underset{\mathbf{W}}{\operatorname{argmax}} \sum_{\mathbf{Q}} \frac{P(\mathbf{X}|\mathbf{W}, \mathbf{Q}, \theta) P(\mathbf{W}, \mathbf{Q}|\theta)}{P(\mathbf{X}|\theta)} && (1.3)\\
&= \underset{\mathbf{W}}{\operatorname{argmax}} \sum_{\mathbf{Q}} P(\mathbf{X}|\mathbf{W}, \mathbf{Q}, \theta) P(\mathbf{Q}|\mathbf{W}, \theta) P(\mathbf{W}|\theta) && (1.4)\\
&= \underset{\mathbf{W}}{\operatorname{argmax}} \sum_{\mathbf{Q}} P(\mathbf{X}|\mathbf{Q}, \theta_{\mathbf{AM}}) P(\mathbf{Q}|\mathbf{W}, \theta_{\mathbf{PM}}) P(\mathbf{W}|\theta_{LM}) && (1.5)
\end{aligned}$$

Invoking Bayes' rule we arrive at Equation 1.3. Notice that $P(\mathbf{X}|\theta)$ in the denominator of Equation 1.3 is constant over all word sequences, so we can drop this term in the argmax. Equation 1.4 results from this and the factoring the joint probability $P(\mathbf{W}, \mathbf{Q}|\theta)$. We then apply the conditional independence assumption that the sequence of features $\mathbf{X}$ is conditionally independent of the word sequence $\mathbf{W}$ given the phone sequence $\mathbf{Q}$ which gives us Equation 1.5. Equation 1.5 consists of three probability models which also happen to define three major subdivisions in ASR research. They are:

- $P(\mathbf{X}|\mathbf{Q}, \theta_{\mathbf{AM}})$: The **acoustic model** models how probable a sequence of features are given a sequence of phones.
- $P(\mathbf{Q}|\mathbf{W}, \theta_{\mathbf{PM}})$: The **pronunciation model** models how probable a sequence of

[3] A phone is defined as any single speech sound considered as a physical event without regard to its place in the sound system of a language [47].

phones are given a sequence of words, essentially providing a pronunciation dictionary that shows how to pronounce words using their constituent phones.

- $P(\mathbf{W}, \theta_{\mathbf{LM}})$: The **language model** models how probable a given word sequence is. This is where grammatical and semantic constraints are modeled.

Many researchers actively pursue improvements in pronunciation as well as language modeling, but this thesis primarily focuses on innovations in the acoustic model. One important component in the acoustic model is the set of acoustic observations used to represent speech, i.e., the front-end features. Nearly every state-of-the-art ASR system uses features that represent some form of the spectral envelope of speech. Figure 1.2 shows some of the typical processing steps. First, we window the speech waveform by applying a 25-millisecond Hamming window every 10 milliseconds. Next, we transform the time domain speech signal into the frequency domain by computing a 256-point fast Fourier transformation (FFT) on each of the windows every 10 milliseconds. Inspired by how the human peripheral auditory system works [48], the next two steps smooth in frequency and compress the magnitude. The squared magnitudes of groups of FFT output bins are averaged together to simulate an auditory-scaled filter bank. The output of the filter bank is compressed by applying the log. For every 10 milliseconds of speech, the result is a smoothed and compressed representation of the spectral envelope of speech. When computing either of the typical features, Mel-Frequency Cepstral Coefficients (MFCC) [87] or Perceptual Linear Predictive (PLP) features [48], additional transformations are applied which further smooth out the spectral envelope. While these features are computed over the entire frequency range, the temporal context of the features is quite limited, coming from the original analysis window of 25 milliseconds. Most state-of-the-art ASR systems use front-end features that have some form of velocity (delta) and acceleration (double delta) components or have been transformed by linear projections computed over several consecutive features. The result of such operations effectively widens the temporal context to about 90 milliseconds. Pictorially, the conventional feature extraction processes speech within narrow vertical rectangles like the one shown in Figure 1.2.

To represent $P(\mathbf{X}|\mathbf{Q}, \theta_{\mathbf{AM}})$, state-of-the-art acoustic models use Hidden Markov Models (HMMs). HMMs are probabilistic finite state machines. There are states in an

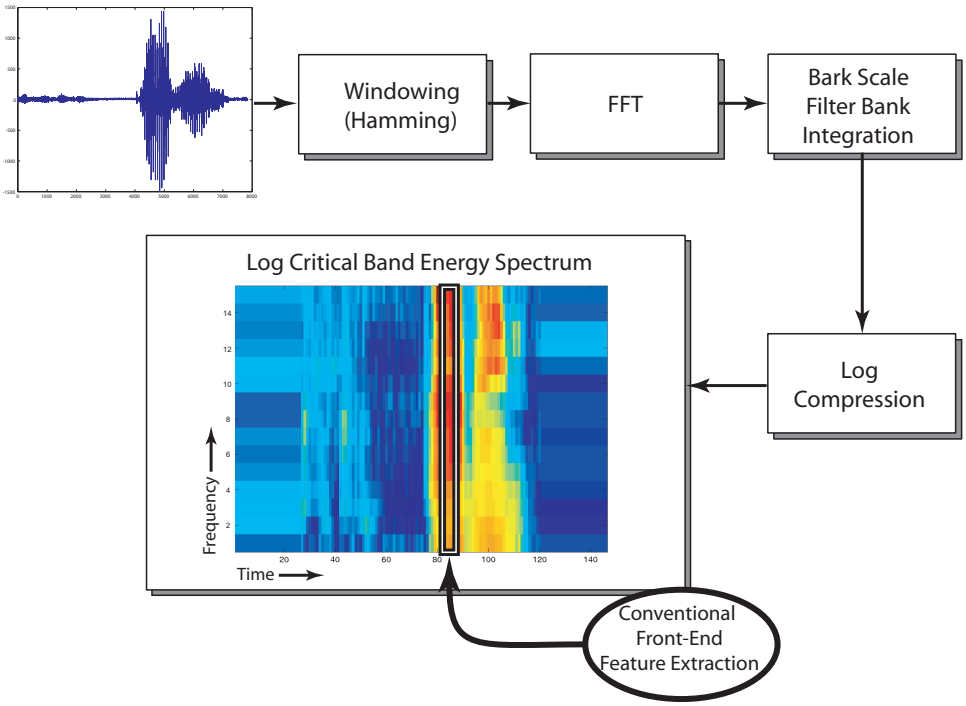

Figure 1.2: Typical front-end feature calculation block diagram.

HMM which represent portions of phones, triphones[4], or some other sub-word unit. Within a state there are probabilities for transitioning to another state or to remain in the current state. Also, within a state there is a probability associated with emitting a certain acoustic feature vector. Figure 1.3 shows an HMM for the word "cat" which is depicted as a sequence of phones (/k/, /ae/, and /t/). Starting in state /k/, there is a probability of staying in state /k/ given by $P(q_t = /k/|q_{t-1} = /k/)$ and a probability of transitioning to state /ae/ represented by $P(q_t = /ae/|q_{t-1} = /k/)$. State /k/ also has a probability of emitting a certain feature vector $\mathbf{x}_t$ given by $P(\mathbf{x}_t|q_t = /k/)$. In general, the overall probability of a sequence of feature vectors given a sequence of phone states from an HMM is given by Equation 1.6, where θ_{AM} is omitted for simplicity but assumed as a conditioning variable in all of the probabilities.

$$P(\mathbf{X}|\mathbf{Q}) = P(x_1|q_1)P(q_1)\prod_{t=2}^{N} P(x_t|q_t)P(q_t|q_{t-1}) \tag{1.6}$$

HMMs make two key modeling assumptions:

[4]Triphones are contextual phones defined by the current phone and the previous and subsequent phone.

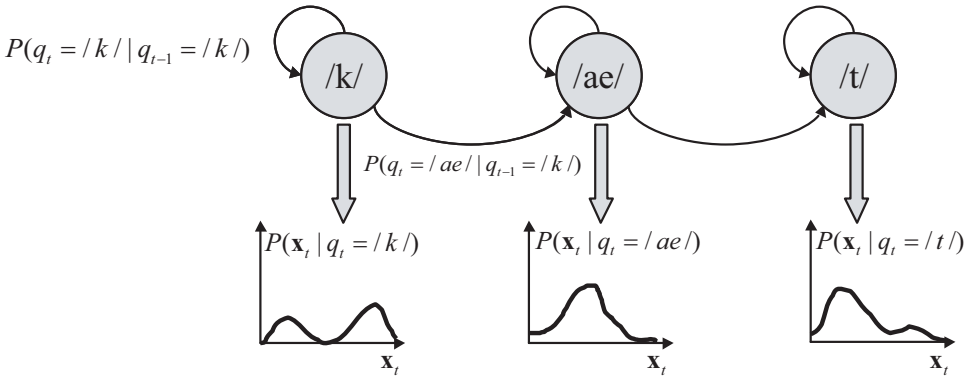

Figure 1.3: An example of a Hidden Markov Model for the word "cat".

- A feature vector at time t is conditionally independent of everything else given the state at time t. In other words, the emission probability distribution doesn't change from time to time within the same state.

- The next state is conditionally independent of all previous states and feature vectors given the current state. Essentially, the next state depends only on what the current state is. This assumption is often referred to as the first-order Markov assumption.

While the second modeling assumption provides a means of modeling the time evolution of feature vectors when states transition, the first modeling assumption implies that the time evolution of feature vectors is not modeled within a single state since the emission probability distribution doesn't change. This means that the probability of being in a phone state is derived from a distribution on a front-end feature that is computed over a very small amount of time context.

1.3 Motivation

Conventional ASR acoustic models are based on capturing the representative spectral profiles of speech sounds. While these spectral profiles or spectral envelopes span the entire frequency bandwidth in the speech, they have very short temporal extents (25 milliseconds - 90 milliseconds). As evidenced by current ASR performance, this short-term approach seems to capture some information about the underlying speech; however, current ASR systems are particularly sensitive to the aforementioned variations in the

speech signal that have deleterious effects on the spectral envelope of speech [113]. The temporal information that is captured by current ASR systems is incorporated in a limited way via the first-order Markov modeling in the HMMs.

1.3.1 Narrow-Band Temporal Patterns Approach to ASR

It is this weakness of relying on the short-term spectral envelope of speech that the work in this thesis addresses. The main goal of this thesis is to capture long-term temporal information in speech and apply it on the recognition of conversational telephone speech (CTS). In particular, this thesis explores and discusses the learning of discriminant temporal patterns (or temporal profiles as opposed to spectral profiles) within narrow-frequency bands spanning long periods of time (about 500 milliseconds). This work extends ground breaking research in TempoRAl Patterns (TRAPs) conducted by Sangita Sharma, Hynek Hermansky, and Pratibha Jain [52, 53, 54, 112, 62] which will be discussed in detail in the next chapter. Our approach to improving the state-of-the-art performance is to develop data-driven front-end features that extract information from speech energy in narrow-frequency bands over long periods of time using neural networks. Instead of developing spectral features from narrow vertical rectangles in the time/frequency plane, we will extract temporal features from long horizontal rectangles as in Figure 1.4. In addition to developing these temporal features, the work in this thesis also combines these features with the conventional spectral features. In this way, we use the information captured by the temporal features to complement the information already provided by the conventional spectral features for the purpose of improving ASR on CTS which contains large amounts of speaker variation.

1.3.2 Why Narrow-Frequency Bands?

We draw our motivation for learning in narrow-frequency bands from a series of human listening experiments. Harvey Fletcher's human listening experiments [37] and Jont Allen's summary of his work [2] provide the main impetus for working on narrow-frequency bands. Fletcher's hypothesis is that independent, narrow-frequency detectors, working in parallel, account for the robustness of human auditory processing. Fletcher introduced the

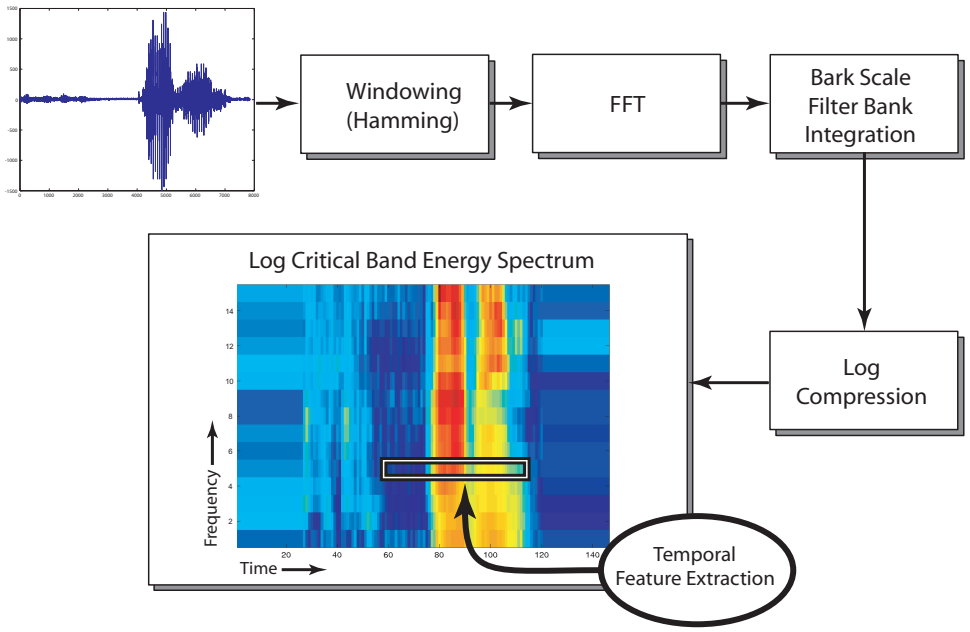

Figure 1.4: Proposed temporal front-end feature calculation block diagram.

Articulation Index (AI) model for predicting speech articulation[5], which states that the total articulation error is equal to the product of independent sub-band articulation errors. Other listening experiments have also shown how humans seem to be able to discriminate between speech sounds given only narrow-frequency speech. Greenberg, et. al. [44, 118] and Warren, et. al. [128, 127] show independently how words can still be recognized despite filtering out all frequencies of speech except for several narrow frequencies. Lippmann [83] also shows that recognizing consonants in nonsense CVC syllables can still be done effectively by human listeners even when the speech is missing middle frequencies from 800 Hz to 3,150 Hz. His listeners could correctly identify 91.6% of the consonants even when missing these frequencies.

Another set of listening experiments shows evidence that humans can robustly detect some set of fundamental categories or speech attributes within narrow-band signals. Miller and Nicely performed an analysis of consonant identification experiments where listeners were given speech that had been filtered by a series of high, low, and band-pass filters [88]. They found that the patterns of errors were not random. Instead, errors seem

[5]Articulation refers to the recognition of nonsense speech sounds, while intelligibility refers to the recognition of meaningful speech.

to be grouped along several speech attributes like voicing, nasality, affrication, place of articulation, and an attribute they call duration to distinguish between /s/, /sh/, /z/, and /zh/. Confusions between consonants sharing an attribute (e.g., voiced consonants) are more often confused with each other, but not often confused with consonants not sharing the attribute (e.g., unvoiced consonants). Also, they measured the mutual information of spoken and perceived consonants in noisy band-limited speech and found that the information transmitted for the voicing attribute is the most robust, followed by nasality, while place is the attribute that is least robust to noise. These results show that certain speech attributes are robustly detected within narrow-frequency bands, and these attributes are detected more robustly than larger units of speech like consonants.

In this work we primarily focus on overlapping narrow-frequency bands spanning a "critical bandwidth". The critical bandwidth comes from early hearing experiments performed by Harvey Fletcher which showed that the threshold of hearing a pure sinusoidal tone with a noise signal centered at the tone increases as the noise signal's bandwidth is widened up to a certain bandwidth. After exceeding this bandwidth, which he referred to as a critical bandwidth, there is no change in the hearing threshold for the sinusoidal tone. In other words, only noise falling within the critical bandwidth of a narrow-band signal can contribute to the masking, and in this way one can consider critical-bands as a series of frequency selective filters. Motivation for using critical-bands also comes from some work on deriving discriminant functions in frequency for ASR. Malayath and Hermansky used linear discriminant analysis (LDA) to derive filters in the frequency domain [85] and found that these filters very much resemble the bank of critical-band filters used in traditional front-end processing techniques like PLP and MFCC.

1.3.3 Why Long-Term?

Human recognition of phones in nonsense syllables has an error rate of about 1.5% according to Allen's analysis of Fletcher's early listening experiments [2]. In contrast, the ASR error rates on phone recognition tasks are still an order of magnitude worse [78, 27, 106, 3]. One reason for the discrepancy in performance between humans and machines is that humans use longer-term information about the phone which is not captured by the current emphasis on the short-term spectral envelope in most ASR systems. Note, this longer-term information does not simply come from semantic context since Fletcher's study

used nonsense syllables. There must be some important long-term characteristics within the acoustics that are cues to the phonetic identity. Researchers have also shown, using information theoretic analysis, that there is significant discriminant information about the identity of the current phone at times up to several hundred milliseconds away [130, 14].

1.3.4 Complementarity to Conventional Features

Finally, by looking for discriminant information in very long time contexts within critical-bands, finding information that is complementary to the information in the short-term conventional analysis is highly likely. The temporal analysis in this thesis helps more on some speech sounds than the short-term conventional features and vice versa. Over the years, many other ASR systems have greatly benefited by using multiple experts or streams of information. Here is but a sampling of successful combination approaches for ASR: [92, 71, 52, 115, 65, 86, 31, 12, 1, 73, 94]. Performance in clean conditions as well as robustness to noisy conditions improves greatly when combining multiple streams of information. This work is yet another example of the benefits obtained by combining different streams of information.

1.4 Thesis Overview

1.4.1 Thesis Goals

In the past few years, several systems that utilize speech information from narrow-frequency channels over long periods of time have demonstrated promising recognition performance improvements. The main thrust of this thesis is to further improve these long-term narrow-band ASR systems. More specifically, this thesis has three main goals:

1. To design and implement new neural network architectures for the learning of phonetically discriminant patterns within critical-bands over long periods of time.

2. To integrate these new architectures with a state-of-the-art ASR system by using the outputs of the neural networks as a data-driven feature vector for the purpose of improving recognition performance on challenging ASR tasks, such as conversational telephone speech.

3. To learn the strengths and weaknesses of these new neural network architectures by comparing them to several existing methods for extracting information within critical-bands over long periods of time.

1.4.2 Thesis Outline

This thesis proceeds as follows. Chapter 2 gives background information useful for understanding the thesis work. This includes a survey of previous work to help the reader frame this work within the state-of-the-art in ASR research. Chapter 3 presents two new neural network architectures for extracting information within critical-bands over long periods of time: Hidden Activation TRAP (HAT) and Tonotopic Multi-Layer Perceptron (TMLP). Performance on a phone recognition task for HAT, TMLP, and other temporal systems is also presented in this chapter as well as their performance in artificial noise and reverberant conditions. In Chapter 4 we explain the approach of using functions of posterior probabilities approximated by neural nets as features for a state-of-the-art ASR system and describe the series of experiments that lead to our best system configuration for the conversational telephone speech recognition task. A comparison of the various temporal systems on a full conversational telephone speech recognition task is presented in Chapter 5. We show that HAT and TMLP significantly outperform some other narrow-band temporal systems, and we analyze where these improvements come from. In Chapter 6 we present an empirical study examining the optimal configuration for TMLP given constraints in total parameters as well as training data. A section on sharing critical-band hidden units in the TMLP is also presented. By sharing these parameters, we are able to show which discriminant temporal patterns are common among different critical-bands. In Chapter 7, we summarize the major themes and points in this thesis and speculate on future directions. Appendices C, D, and E contains a gallery of discriminant temporal patterns learned in HAT and TMLP.

Chapter 2

Background

Having motivated the general approach of extracting speech information within narrow-frequency bands over a relatively long amount of time, we survey the research landscape in this background chapter. Specifically, we are interested in showing how the work in this thesis "stands on the shoulders of giants"[1] by reviewing relevant previous work.

2.1 Related Work

2.1.1 Multi-Layer Perceptrons

Multi-Layer Perceptrons (MLPs) are artificial neural networks that have been successfully used in many ASR systems over the past 15 years. They are one of the central tools used in this thesis, and so we provide a brief description of them. MLPs can be thought of as universal function approximators and are commonly used in ASR as phonetic posterior probability estimators. Given a set of input features, the MLPs are trained to learn the mapping to phonetic probabilities posterior on the input features.

Since it has been shown theoretically that fully-connected 3-layer neural networks with a single, sufficiently large hidden layer of units can approximate any function [74, 75], 3-layer MLPs are typically used. A 3-layer MLP, similar to ones used in this thesis, is

[1] This quote is often attributed to Isaac Newton who wrote "If I have seen further it is by standing on ye shoulders of Giants".

pictured in Figure 2.1. The inputs to the neural net are copied into nodes of the first layer, which is referred to as the input layer. The input layer is fully-connected to the next layer, called the hidden layer, which means that the output of every hidden unit is a function of every input node. The value at the output of the jth hidden unit, H_j, is a weighted sum of all the inputs passed through a sigmoid nonlinearity:

$$H_j \stackrel{\text{def}}{=} sig\left(\sum_{i \in inputs} in_i W_{i,j} + B_j\right) \tag{2.1}$$

where in_i is the ith input, $W_{i,j}$ is the trainable weight parameter between the ith input and the jth hidden unit, and B_j is the trainable bias for the jth hidden unit. The sigmoid function is given by:

$$sig(x) \stackrel{\text{def}}{=} \frac{1}{1 + \exp(-x)} \tag{2.2}$$

These hidden units are fully-connected to the last layer which is called the output layer. The output of the kth output unit is given by the softmax function:

$$Out_k \stackrel{\text{def}}{=} \frac{\exp(Z_k)}{\sum_{K \in outputs} \exp(Z_K)} \tag{2.3}$$

where Z_k is given by equation (2.4):

$$Z_k \stackrel{\text{def}}{=} \sum_{j \in hiddenunits} H_j W_{j,k} + B_k \tag{2.4}$$

$W_{j,k}$ and B_k are the trainable weights and bias for the kth output unit. For every category that we wish to classify, there is an output unit whose value approximates the posterior probability of the corresponding category after training.

The training procedure that we use for these MLPs is the gradient descent-based error back-propagation algorithm [108]. We use the cross-entropy error criterion [15] with the training targets in a "1-of-c coding". This means that there are c output classes, and the target vector consists of "0.0"s except for the single dimension corresponding to the labeled category which gets a value of "1.0". The learning schedule is a form of early stopping with cross-validation. This means that each epoch's learning rate is determined by how well the MLP is performing on a separate cross-validation data set. Initially the learning rate is high, and as improvements in accuracy on the cross-validation data become smaller, the learning rate is exponentially reduced. Finally, when no more accuracy improvements happen, the training is stopped to prevent overfitting. If the MLP has a

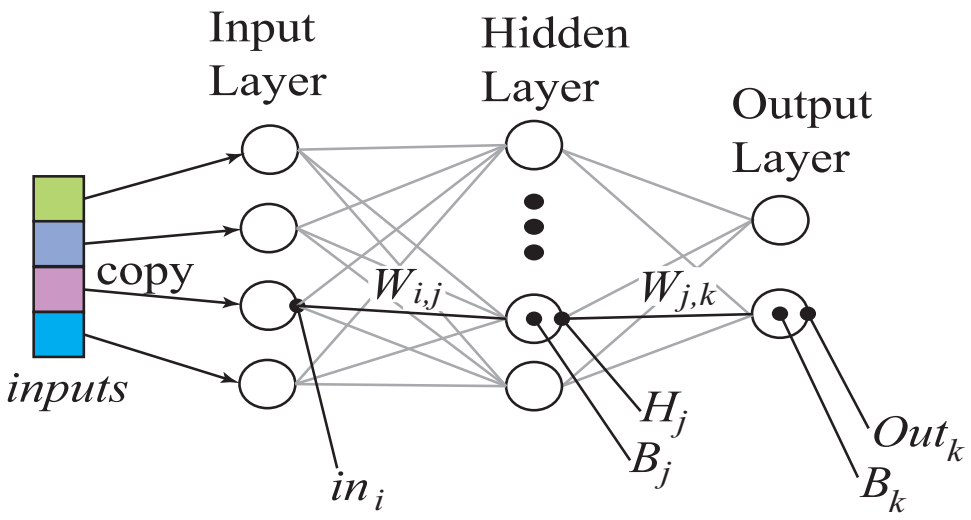

Figure 2.1: A 3-Layer Multi-Layer Perceptron

sufficiently large number of hidden units to approximate the mapping function between the inputs and output classes, then the outputs of an MLP trained in this way can be considered probabilities of the training categories posterior on the inputs. For a detailed proof of this, as well as further description of the learning approach, refer to [96].

2.1.2 The Hybrid HMM/ANN and Tandem ASR Architectures

The work presented in this thesis contains many experimental results on the automatic recognition of words in various standard speech databases, and it uses two distinct ASR architectures: the hybrid HMM/ANN [18] and the Tandem [54, 49, 34, 32] architectures. Both of these architectures use feed-forward neural nets like the 3-layer MLPs described above to derive estimates of phone posterior probabilities. In the hybrid HMM/ANN architecture, these phone probabilities are used directly in a dynamic-programming-based Viterbi search [107, 105, 57], which approximates the forward algorithm for HMMs [11], to recognize the best sequence of words. In the Tandem architecture the MLP serves as a data-derived feature extractor. The estimated phone posteriors from the MLP are transformed and then used as front-end features to a standard Gaussian mixtures-based HMM system. A block diagram for a typical hybrid HMM/ANN system is depicted in Figure 2.2, while that for a Tandem ASR system is pictured in Figure 2.3.

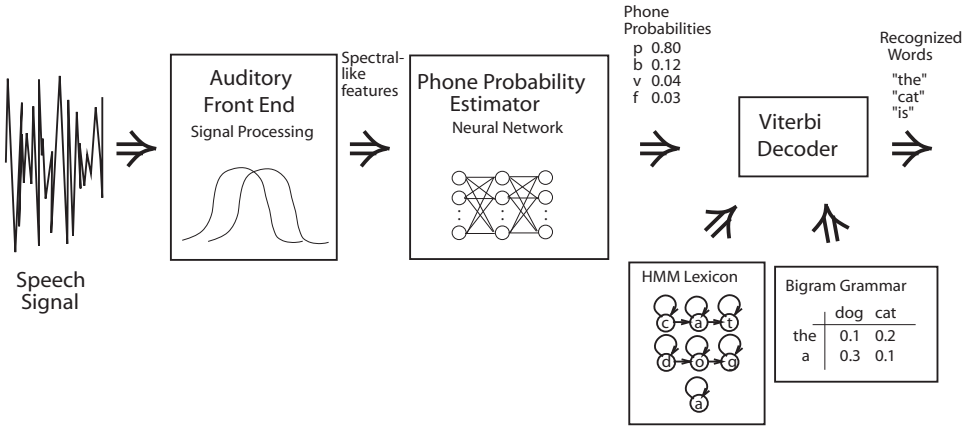

Figure 2.2: Hybrid HMM/ANN ASR system. Speech is transformed into spectral-like features, which are sent to a neural net that estimates phone posterior probabilities used for decoding (typically after division by priors to yield scaled likelihoods) by a Viterbi decoder under grammar and pronunciation constraints.

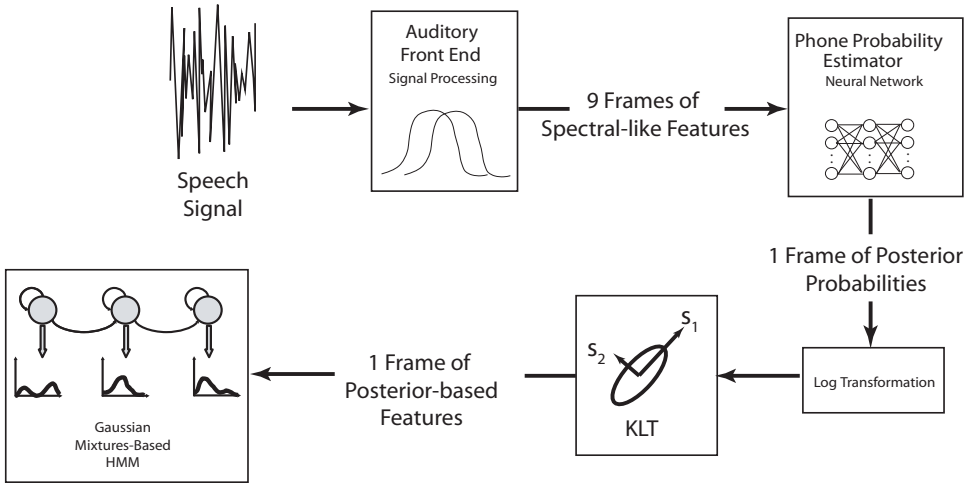

Figure 2.3: A typical Tandem ASR system. Speech is transformed into spectral-like features, which are sent to a neural net that estimates phone posteriors. These are then transformed by the log and Karhunen Lòeve Transform (including dimensionality reduction) and used as posterior features for a standard Gaussian mixtures-based HMM.

Much of the new work in the ASR research community has focused on a cousin of the hybrid approach, which uses Gaussian mixtures for modeling the acoustic emission probabilities in HMMs. Many powerful techniques, like adaptation based on Maximum Likelihood Linear Regression (MLLR) [39], speaker-adaptive feature transformation (SAT) [80], tied context dependent triphones [131], etc. were developed for these Gaussian mixtures-based HMMs and led to major reductions in word error rates. These techniques were harder to integrate within the hybrid system, and so they were either not tried or were only moderately effective. As a result, the perfomance of many hybrid systems lagged that of the Gaussian mixtures-based HMMs. With the advent of the Tandem system, the advantage of discriminative training of the neural nets could be combined with all the powerful adaptation techniques developed for the Gaussian mixtures-based HMMs. In [12], Benitez et. al. improved the original Tandem setup by using the outputs of the MLPs to augment the traditional PLP features instead of replacing them. This led to great improvements in the performance of the recognizer compared to the baseline system that simply used the PLP features. There are several issues that arise when using the Tandem approach. First, the development time is greater because of the additional training time needed for the neural net. Also, there are issues involving the transformation of the MLP outputs (posterior probabilities) to features that are better suited for the modeling assumptions implied by the Gaussian mixtures-based HMM. This involves choosing suitable transformations and also determining what amount of dimensionality reduction is optimal.

2.1.3 TempoRAl Patterns - TRAPs

The work in this thesis is most closely related to the study of temporal patterns or TRAPs. For decades, conventional ASR systems have based the feature extraction process on the premise that each of the various speech sounds or phones have distinctive patterns in frequency. For example, the spectral envelope of a typical /i/ sound, as in "beet", has magnitude peaks near 280, 2250, and 2900 Hz, while a typical /U/ sound, as in "book", has peaks near 450, 1030, and 2380 Hz. In a similar way, one can look for distinctive patterns along time within narrow-frequency bands. This is exactly what Hynek Hermansky and Sangita Sharma did in their TRAPs work [52, 53, 112]. Using speech data that was phonetically hand-transcribed, they first computed frames of log critical-band energies for every 10 milliseconds of speech. Each of these frames was given a

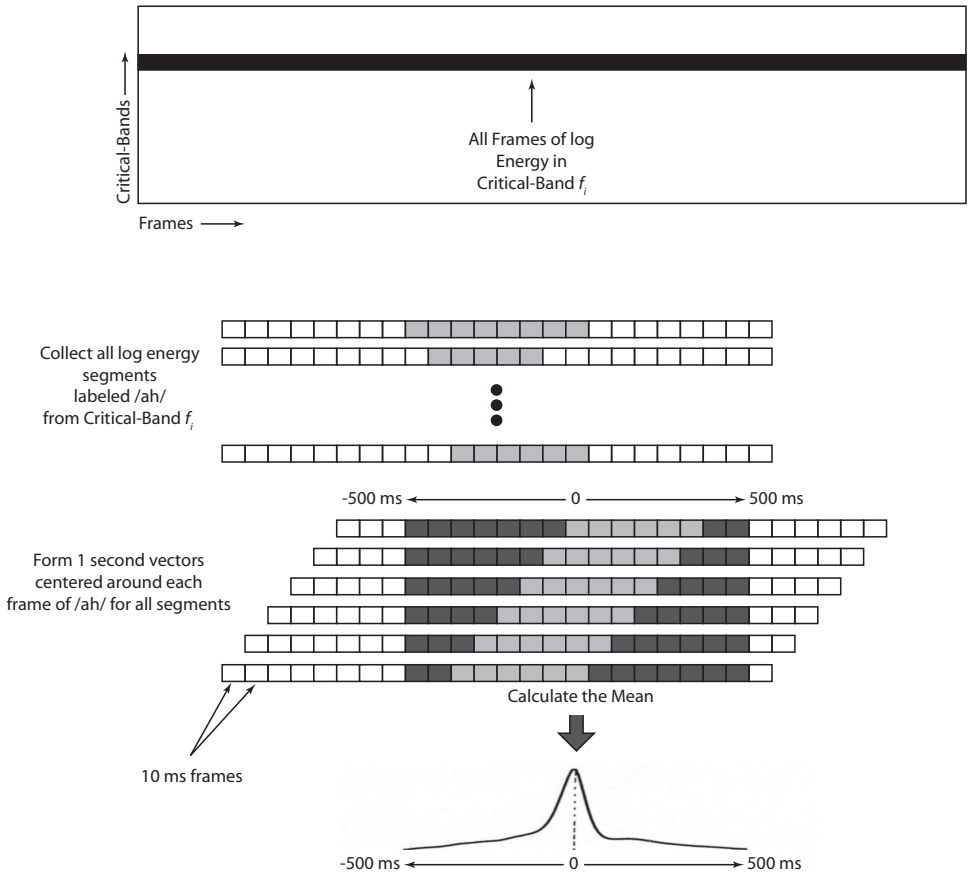

Figure 2.4: Computation of the temporal evolution of phoneme /ah/ for critical-band f_i from a labeled database. Adapted from [112].

phone label corresponding to the phonetic transcription occurring at the time in the speech waveform that the frame was calculated. For each frame within a single critical-band, they concatenated 50 consecutive frames before and after the frame to form a 101-frame critical-band energy trajectory or temporal pattern[2]. By taking all the energy trajectories whose center frame was labeled with the same phone and averaging these energy trajectories, they were able to produce representative temporal patterns for each of the 45 hand-labeled phones in their speech data. Figure 2.4 shows this process of producing these representative temporal patterns, which they call "Mean TRAPs".

Having calculated mean temporal patterns or Mean TRAPs for each critical-band and every phone, Sharma plotted these Mean TRAPs. Figure 2.5 displays the 45 Mean

[2]This trajectory is about 1 second wide

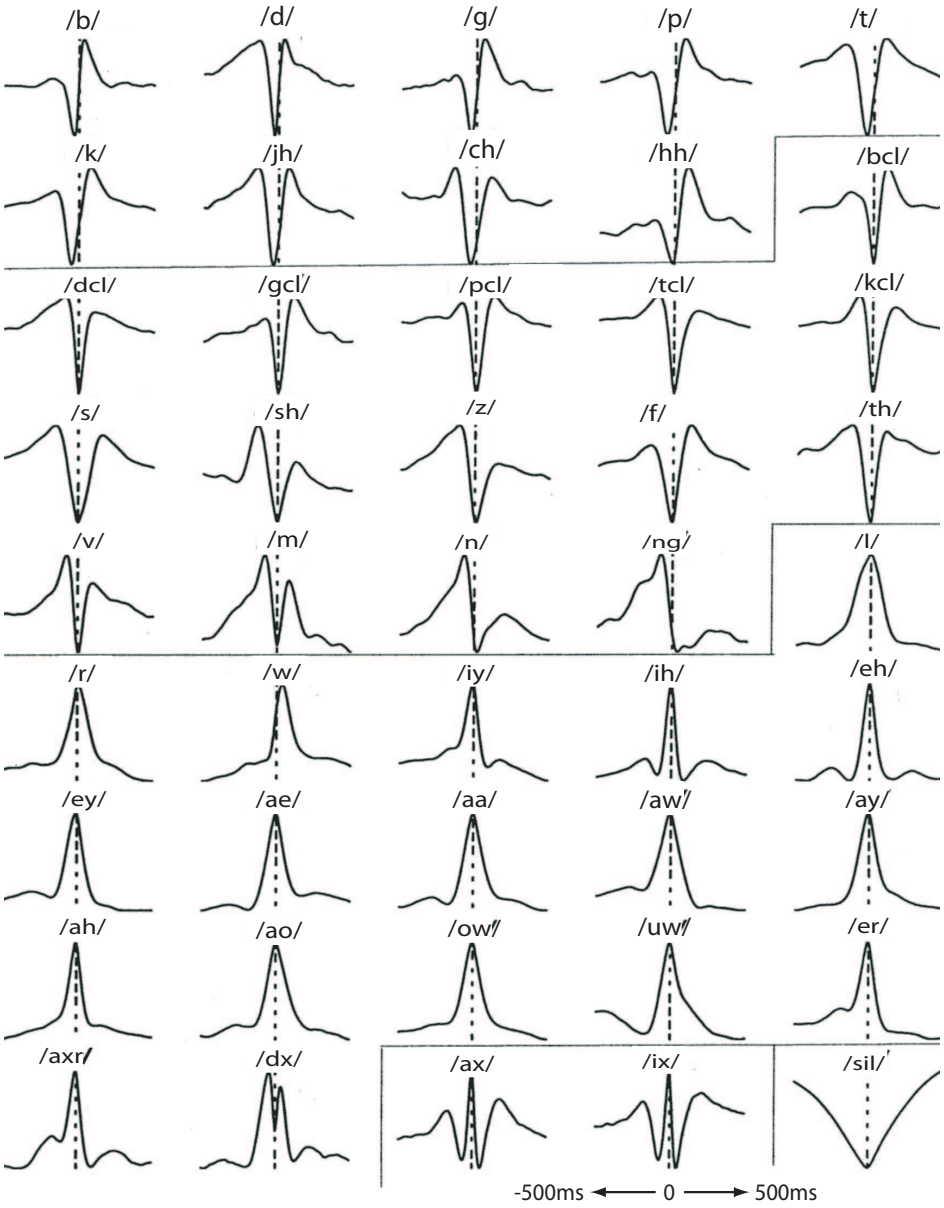

Figure 2.5: Mean TRAPs for 45 phonemes for critical-band 5 (446-637 Hz). The dotted line for each of the TRAPs represents the center frame, or time=0 milliseconds. The patterns separated by solid lines represent sounds with similar temporal patters. The Y-axis corresponds to the energy magnitude. Adapted from [112].

TRAPs calculated for critical-band 5 (446-637 Hz) adapted from [112]. From this figure, you can see how every phone has its unique temporal pattern. Some of the temporal patterns look similar to each other. Temporal patterns coming from vowels (/iy/, /ih/, /eh/, /ey/, /ae/, /aa/, /aw/, /ay/, /ah/, /ao/, /ow/, /uw/, /er/, and /axr/) look pretty similar in that they all have high energy at the center frame. Stop consonants (/b/, /d/, /g/, /p/, /t/, and /k/) also look alike; each has a low energy valley preceding the center frame corresponding to the complete closure in the vocal tract.

Based on these observations, Hermansky and Sharma surmised that they could use similarity measures to the Mean TRAPs in each critical-band as features for a neural net classifier. They created 101-frame energy trajectories centered at every frame in the same way as was done to create the Mean TRAPs. Then they calculated the similarity score (given by Equation 2.5) to each of the Mean TRAPs in every critical-band.

$$d(x, y) = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} \tag{2.5}$$

$d(x, y)$ is the distance between trajectory x and trajectory y as of function of the covariance between x and y (σ_{xy}^2) and the standard deviations of x and y (σ_x and σ_y). This resulted in a set of numbers (15 critical-bands by 29 phones) that were used as inputs to a merger MLP trained on corresponding phone targets. Using this MLP in the hybrid HMM/ANN recognition setup, they achieved a word error rate (WER) of 11.5% on the OGI Numbers corpus. State-of-the-art performance at that time hovered around 6% for this corpus, but 11.5% was not a terrible result for such a radically new approach.

Since many of the Mean TRAPs looked similar, Hermansky and Sharma also clustered them agglomeratively using the same distance metric in Equation 2.5. They called these cluster centroids "Broad TRAPs" because the Mean TRAPs automatically grouped into five broad phonetic categories: vowels, stop-consonants, fricatives, schwas, and silence. A picture of the Broad TRAPs for critical-band 5 as adapted from [112] is shown in Figure 2.6. Using these Broad TRAPs as the templates for recognition, they again computed similarity measures of test speech to these Broad TRAPs and used these measures as input to an MLP trained to learn phone probabilities. Within the hybrid HMM/ANN recognition setup, this Broad TRAP system gave a 12.8% WER on the OGI Numbers corpus.

To improve upon these initial TRAP-based systems, they developed the Neural

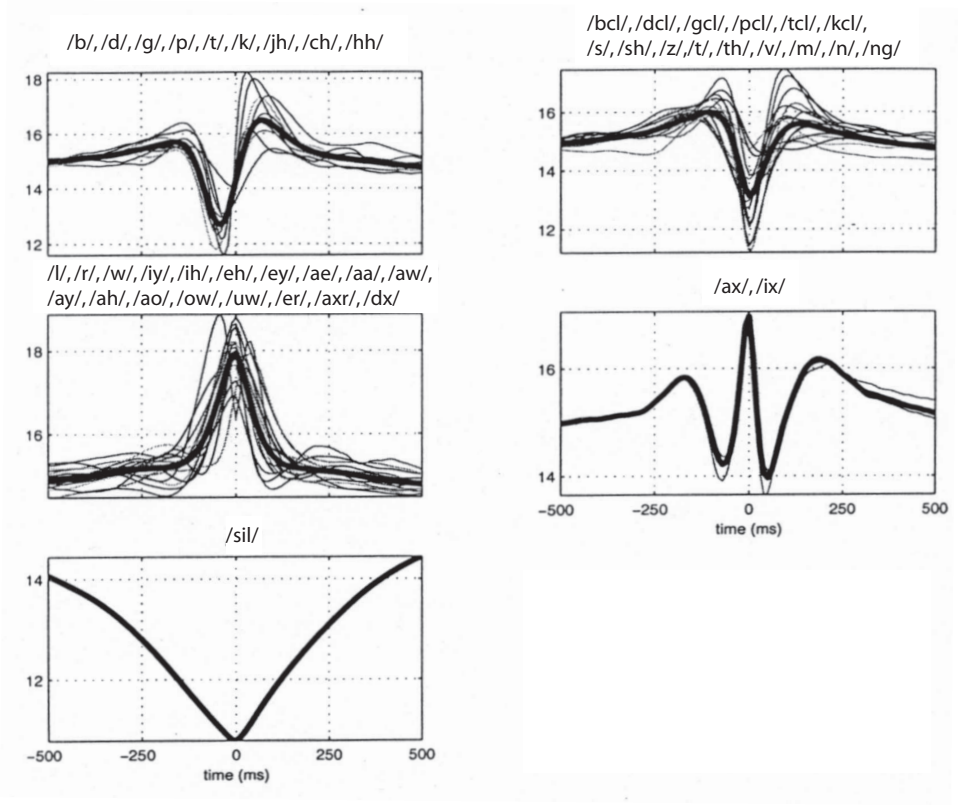

Figure 2.6: Broad TRAP clusters of the fifth critical-band (438 Hz - 629 Hz) time trajectory. The thinner lines in each plot represent the individual Mean TRAP of the phonemes clustered in one category. The thicker line is the Broad TRAP and represents the weighted mean of the constituent Mean TRAPs. Adapted from [112].

System	WER
Baseline	6.5%
Neural TRAP	7.6%
Mean TRAP	11.5%
Broad TRAP	12.8%
Combined: Baseline+Mean TRAP	6.0%
Combined: Baseline+Neural TRAP	5.5%

Table 2.1: Word error rate results on various systems on OGI Numbers corpus.

TRAP system. The Neural TRAP system consists of two stages of MLPs. The first stage MLPs are a set of critical-band MLPs (one for each critical-band) that estimate critical-band level phoneme probabilities from 101-frame energy trajectories. These critical-band MLPs replace the simple similarity metrics with a powerful universal function approximator that is discriminant in nature. The second stage of the Neural TRAP system consists of a single MLP that combines the output of the each of the critical-band MLPs to form a single estimate for the phone posterior probability. This Neural TRAP system outperformed their previous Mean TRAP system by achieving a 7.6% word error rate on the same OGI Numbers corpus.

Table 2.1 summarizes the performance of the various hybrid HMM/ANN systems tested by Sharma on the OGI Numbers corpus [112, 52]. The baseline system is a standard HMM/ANN setup where 9 frames of 8th order PLP cepstral coefficients along with 9 deltas and 9 acceleration coefficients are used as inputs to an MLP outputting phone posteriors. Note that the temporal context of this baseline system is 9 frames (about 100 milliseconds). By themselves, the TRAP-based systems do not outperform the baseline system. Neural TRAP is competitive to the baseline (7.6% vs. 6.5%), while the Mean TRAP and Broad TRAP systems are much worse. However, when combining the outputs of the TRAP-based MLPs (which look at temporal extents of about 1 second) to that of the baseline MLP by simply averaging them in the log domain, recognition performance beats that of the baseline system (6.0% for combination with Mean TRAP and 5.5% for combination with Neural TRAP). In general TRAP-based systems are typically competitive with conventional systems that rely on the spectral envelope of speech, but when combined with these conventional systems, performance improves over that of ei-

ther system alone. This suggests that the method of extracting information from speech within long-term and narrow-frequency bands is providing complementary information to the conventional methods. Other TRAP-based studies have also shown results consistent with this generalization [54, 64, 24].

Because the work in this thesis and much of the other related work on TRAPs requires deeper understanding of the Neural TRAP system, we will now go into greater detail about the Neural TRAP system. Figure 2.7 shows a block diagram explaining the processing steps for the Neural TRAP setup. The inputs to the Neural TRAP setup are 19 101-frame log critical-band energy trajectories[3]. Each energy trajectory is fed into the corresponding first stage critical-band MLP whose outputs are then either taken before the final softmax or transformed by log and fed into the second stage merger MLP. To train a complete Neural TRAP system, the first step is to train the critical-band MLPs using the standard error back-propagation algorithm. Hermansky and Sharma used the overall phone labels as targets for each of the critical-band MLPs, so that each critical-band MLP would learn to gather all the evidence within critical-band energy trajectories for phone discrimination. Once these MLPs were trained, the training data was forward passed through them to create input training data for the merger MLP. The training pairs for the merger MLP are either the outputs before the final softmax or the log outputs from the first stage MLPs and the same phone labels used in the first stage training. The second stage merger MLP is also trained with the error back-propagation algorithm, and its outputs approximate phone posterior probabilities.

It is interesting to discuss the nature of the narrow-frequency long term energy trajectory that is learned by these various TRAP-based ASR systems. In the Mean TRAP ASR system, an underlying representation of the temporal patterns for each phone in every critical-band is captured by averaging together all such examples in the training data. The Broad TRAP ASR system further collapses these Mean TRAPs into 5 cluster centroids. Both of these systems learn basic canonical trajectory patterns that are then used as a template for matching during testing. In contrast, the Neural TRAP system learns a discriminant mapping from the critical-band trajectories to critical-band level phone probabilities. Such mappings produce critical-band level evidence for the presence

[3]There are 19 critical-bands when the sampling rate is 16000 Hz and 15 when the sampling rate is 8000 Hz.

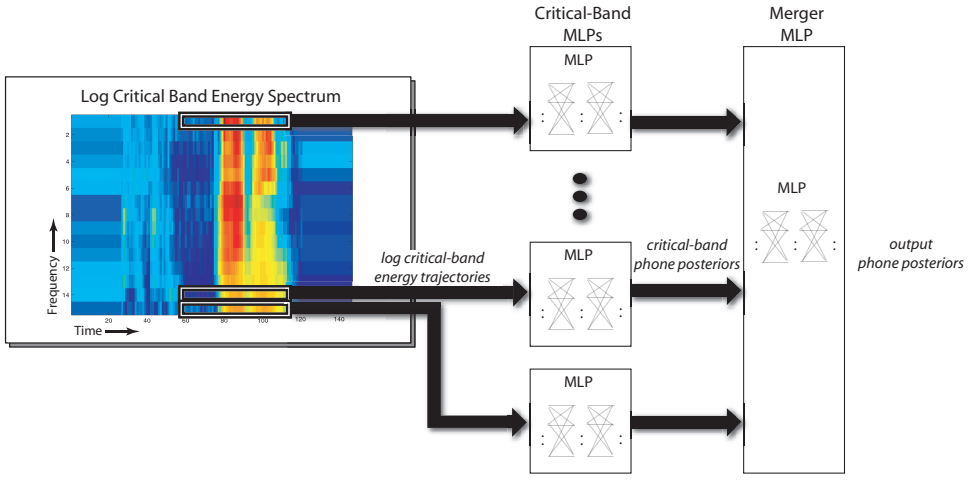

Figure 2.7: The Neural TRAP architecture consists of two stages of MLPs. The first stage is a set of critical-band MLPs estimating the critical-band level phone posteriors. The second stage is a merger MLP that combines the critical-band level phone posteriors to get an overall estimate of the phone posterior probabilities.

or absence of each phone. Neural TRAP works much better than either the similarity-based Mean TRAP or Broad TRAP systems, which suggest that the discriminant mapping produced by the critical-band MLP is better able to capture subtle differences between different phones not captured by the similarity measure to Mean TRAPs or Broad TRAPs.

It is also interesting to note that the performance difference between the Mean TRAP and Broad TRAP systems is not large, which led Sharma to write that "full phoneme classification on each sub-band temporal energy pattern may not be necessary". Additionally, the mapping from critical-band energy trajectories to phone probabilities learned by the critical-band MLPs in the Neural TRAP system is not perfect. The reported frame error on the OGI Numbers corpus from [112] ranges from a low of 65% to a high of 69%. One may have expected this since it is really hard to distinguish one phone from another simply given a critical-band speech signal. Still, this raises an important question: are phone probabilities at the critical-band level the best information to extract for better ASR performance? If not, then what other kind of evidence within critical-band trajectories should be collected? The two new neural net architectures presented later in this thesis address these questions. The first new architecture, Hidden Activation TRAP (HAT), shows that mapping all the way to phones at the critical-band level is not necessary and actually hurts performance. The second architecture, Tonotopic Multi-Layer Perceptron (TMLP),

automatically learns what is important at the critical-band level to improve the overall phone classification rate.

Other work that has built upon the foundation of Sharma's work can be grouped into 3 categories: improvements to the features presented to Neural TRAP; applications to other ASR tasks or other speech related problems; and explorations of different techniques to learn important critical-band level information.

The first category consists of research devoted to improving the input features to Neural TRAP. One line of work within this first category is to replace the adjacent frames of log critical-band energies with more elegant approaches that avoid artifacts arising from windowing speech and applying the short-term FFT. In [99], Motlíček et al. derived inputs to Neural TRAP directly from the time domain signal using a bank of band-pass Gammatone filters. In [7], Athineos et al. created inputs for Neural TRAP by applying Frequency Domain Linear Prediction to the speech signal which essentially fitted an all-pole model to the speech signal's squared Hilbert envelope. Motlíček's approach did not significantly improve over the original Neural TRAP's inputs, while Athineos showed about a 10% relative improvement on the OGI Numbers task.

Another line of research for improving the inputs to Neural TRAP is the preprocessing of the original frames of log critical-band energies with various filters. In [46, 69], Grezl and Karafiat applied 1-dimensional and 2-dimensional filters to the log critical-band spectrum which in essence either averaged or differentiated the energy across adjacent frequency bands and adjacent frames. After these modifications, they concatenated adjacent frames within each critical-band for input to the Neural TRAP classifier. They found that in combination with the original Neural TRAP, this new Neural TRAP based on modified features gave some amount of complementary information and improved performance over uncombined systems. Jain found in her thesis that transforming the original critical-band energy trajectory by performing principal components analysis (PCA) or a discrete cosine transform (DCT) and then keeping only half of the original features also improved the performance of Neural TRAP [62]. Finally, a push to using three adjacent critical-band energy trajectories instead of one as inputs to each first stage MLPs in Neural TRAP has also led to better performance than the original Neural TRAP. Using 3 bands, Jain and Hermansky were able to beat the performance of the conventional HMM/ANN system that used PLP features as input to the MLP on the OGI Numbers task [63].

The second category of extensions to the original Neural TRAP system is the application of Neural TRAP to different tasks, whether they be different speech recognition test sets or other non-ASR tasks. In [64], Neural TRAP was used to derive front-end features for a distributed speech recognition system applied to noisy digit recognition. Schwarz et al. [111] used Neural TRAP to perform TIMIT phoneme recognition, and Kingsbury et al. used the Neural TRAP architecture applied on the task of robust voice activity detection [70].

The final category of extensions explore alternative methods to the learning of critical-band level information. More specifically, what categories of targets are appropriate to learn at the critical-band level. As discussed before, the original Neural TRAP learns a nonlinear discriminant mapping from the critical-band energy trajectories to critical-band level phone probabilities, and these mappings are not very accurate. Jain and colleagues developed a modified Neural TRAP that learned discriminant temporal patterns for classifying six broad categories based on manner of articulation [64], both at the critical-band level and full-band level. Using the outputs of their new system as features to augment conventional MFCC features, they were able to show consistent improvements on the Aurora-2 noisy continuous digits data. Hermansky and Jain also explored a new method based on Neural TRAP that was designed to learn temporal patterns that are shared by speech sounds within the same critical-band and across different critical-bands [50]. Motivated by the clustering of Mean TRAPs into Broad TRAPs, they developed Universal TRAP (UTRAP), which basically used data-derived class labels for the training of a single critical-band MLP that replaced all the first stage critical-band MLPs in the Neural TRAP setup. While the second stage merger MLP was still trained using phone targets, the first stage critical-band MLPs were trained using targets that were derived as follows: starting from the set of Mean TRAPs calculated for every phone in every critical-band, they performed an agglomerative clustering (the similarity metric was given by Equation 2.5) of all these Mean TRAPs to come up with a set of 9 centroids. These 9 centroids represented distinct speech events that commonly occured in all critical-bands. Next, they relabeled each frame of speech in every critical-band with a label corresponding to the centroid that was most similar (as measured by Equation 2.5) to the temporal trajectory centered at that particular frame. They reported that the UTRAP system performed comparably to a Neural TRAP system where the critical-band targets were the Broad TRAP categories,

while using many fewer parameters [50].

2.1.4 Multi-Band

The Neural TRAP system described above is an example of a multi-band speech recognition system. In multi-band speech recognition, evidence of phonetic events are first analyzed in independent sub-frequency bands that are later merged for classification of speech sounds. The main difference between Neural TRAP and more conventional multi-band systems is that the sub-frequency bands in Neural TRAP are typically much narrower, and the temporal context for Neural TRAP is from 500 milliseconds to 1 second compared with conventional multi-band systems that take evidence spanning no more than 100 milliseconds.

The collaboration between Bourlard, Dupont, Hermansky, Tibrewala, Morgan, and Mirghafori created complete multi-band ASR systems for recognizing continuous speech within the hybrid HMM/ANN framework [16, 17, 55, 123, 92, 91]. These systems consisted of MLPs estimating phone posteriors within sub-bands (comprised of 2 or more adjacent critical-bands), a fusion step to merge sub-band phone posteriors to create an overall phone posterior (usually a simple frame-wise average or product), and then the HMM Viterbi decoder. They tested their systems on various speech databases ranging from a simple digits and continuous numbers corpus to the large vocabulary conversational Switchboard task. They also tested the noise robustness of multi-band systems by artificially corrupting their speech data. Generally, the performance of multi-band systems were as good (and in some cases better) than full-band systems in clean conditions; however, in band-limited noisy conditions, multi-band systems significantly outperformed full-band systems. Moreover, in combination with full-band systems, multi-band systems further improved ASR performance over the baseline full-band systems. Other researchers have also corroborated these general findings in their own multi-band systems that were not necessarily based upon the hybrid HMM/ANN framework [22, 23, 102, 103, 25, 98].

One issue that occurs in the design of multi-band systems is the choice of categories to classify at the sub-band level that would lead to the best performance improvements for ASR at the full-band level. In the typical multi-band systems built within the hybrid HMM/ANN framework, sub-band MLPs are trained on the full-band phone targets

in the same way the critical-band MLPs in the Neural TRAP system are trained. This may not be the best kind of target because sub-frequency bands may not contain all the information necessary to do full phone classification. For example, consider the two fricatives /f/ and /s/. At lower sub-bands, they are almost indistinguishable. Only at the high frequency sub-bands can one easily distinguish these two fricatives. Mirghafori in [91] observed that the sub-band MLPs do confuse certain phones quite often. Her hypothesis was that by combining the most confusable sub-band phone classes, the sub-band MLPs could devote more trainable parameters to better model those phones for which the particular sub-band contained the most information for classification. Once these sub-band phones were merged, she retrained MLPs on these new merged sub-band phone categories. She found performance improvements at the frame accuracy level, which did not translate to improvements at the word level.

Others have approached this issue from a global optimization perspective. Instead of deriving merged sub-band phone categories as Mirghafori does or deriving clustered Mean TRAP targets as Jain does in UTRAP, researchers have automatically learned what is important at the sub-band levels via optimization procedures. Cerisara et al. [21] used the discriminant minimum classification error criterion (MCE) [66] to guide the training of each sub-band classifier. Daoudi et al. in [25] and Saul et al. in [109, 110] treated sub-band categories as hidden variables within probabilistic graphical models and used the expectation maximization algorithm [26] to automatically learn the model parameters to maximize the likelihood on the training data. In Saul's work, various sub-band detectors for evidence of voicing or sonorance were automatically learned without the need for sub-band labeling of the evidence. One of the new neural net architectures presented in this thesis, the Tonotopic Multi-Layer Perceptron, automatically learns what critical-band level categories are useful for phonetic classification using the error-back propagation algorithm.

2.1.5 Temporal Filtering

There has been a considerable amount of work devoted to the temporal filtering of front-end features to improve ASR performance. Temporal filtering in this context refers to the processing of speech features (or spectral energies of speech) over time. All of the TRAP-based systems, including the systems developed later in this thesis, are examples of data-derived temporal filters. One of the earliest successful approaches to the tempo-

ral filtering of features is Furui's velocity and acceleration coefficients [38]. By appending the calculated velocities and accelerations of each of the original front-end features, ASR performance improves so consistently that the use of velocity and acceleration coefficients today is ubiquitous. Cepstral Mean Subtraction (CMS) [6] is another effective temporal filtering technique that subtracts out the mean of each of the cepstral coefficients calculated over long periods of time (whole utterances, whole conversations, or all examples). CMS is often used to make ASR systems more robust to changes in the channel like the ones caused by microphone differences. RASTA-PLP is also another technique that improves robustness to channel effects [51] by suppressing constant factors in each spectral component of the speech signal.

All of these earlier instances of temporal filtering, which led to increased ASR accuracies, can be studied from the point of view of modulation frequencies. Modulation frequencies [59] are the rates at which the spectral amplitudes of speech change. Just as the conventional speech spectrum measures the energy content at various frequencies or rates of changes in the time domain speech signal, a modulation spectrum measures the energy content at various modulation frequencies or rates of changes of the spectral energy over time [8]. We can view all temporal filtering techniques as processes that either emphasize or deemphasize certain modulation frequencies. CMS, which removes unchanging components in the cepstrum, filters out 0 Hz modulations; RASTA-PLP passes components of the modulation spectrum between about 1 Hz and 12 Hz; and the velocity and acceleration features emphasize modulations at 10 Hz [51].

For human speech perception, intelligibility of spoken words is directly related to how well slow changes in the speech spectrum (modulation frequencies less than 16 Hz) are preserved [58, 59]. Others have also filtered the spectrum (or cepstrum) of speech over time to demonstrate in human perceptual experiments which modulation frequencies are required for high intelligibility. Drullman et al. showed that modulation frequencies above 16 Hz are not required for good intelligibility, and that significant intelligibility remains when only rates less than 6 Hz are preserved [29]. Arai et al. [5] extended Drullman's results to the logarithmic domain and applied various kinds of filters (high-pass, low-pass, and band-pass) to show that modulation frequencies between 1 and 16 Hz are necessary to preserve speech intelligibility. Kanedera et al. measured the effect of modulation filtering to ASR performance and also showed the importance of modulation frequencies between

1 and 16 Hz [68].

Newer temporal filtering techniques can be roughly classified in one of two categories: knowledge-driven or data-driven. In knowledge-driven techniques, the filters are mostly designed based on expert knowledge, i.e., which modulation frequencies are important for ASR. In data-driven techniques, some part of the filter design is guided by the minimization/maximization of an error/goodness score on training data. We will briefly summarize newer temporal filtering techniques for ASR according to these rough classifications.

Knowledge-Driven Temporal Filters

Motivated by the perceptual studies on the relationship between intelligibility and preservation of low modulation frequencies, Kingsbury et al. developed Modulation Filtered SpectroGram (MFSG) features [71]. MFSG processing steps were designed so that modulation frequencies outside of the range between 0 and 8 Hz were filtered out, while modulations at 4 Hz were emphasized. As reported in [71], MFSG outperformed regular PLP features in noisy and reverberant conditions, but did not outperform RASTA-PLP, another temporal filtering technique that is also sensitive to slow modulations in a different way. Combining systems trained on RASTA-PLP features with that of MFSG features yielded significant performance improvements.

Nadeu et al. also developed temporal filters that not only emphasize certain regions in the modulation spectrum but also flatten out the modulation spectrum within these regions [101]. According to [101], this equalization of the modulation spectrum makes the filtered features a better match for the modeling assumption of typical HMMs which model the emission of features from a single HMM state as being independent and identically distributed. His filters emphasized modulation frequencies at 3 Hz, which happens to be a common syllable rate of speech. Nadeu extended his approach to time and frequency filtering in [100]. These techniques also led to significant ASR performance improvements in both clean and noisy conditions.

Measurements of the average magnitude of modulation frequencies at different auditory frequencies of Mandarin syllables motivated Shen et al. to develop a bank of RASTA-like temporal filters [82]. The parameters of these filters were set to emphasize

the important modulation frequencies of their speech data. They measured the difference between the magnitudes of noise and speech with respect to modulation frequency to determine which modulation frequencies were important. The lower this difference was at a particular modulation frequency, the more important this frequency was for speech intelligibility. They found that for Mandarin syllables in noisy and mismatched conditions (additive white noise and microphone mismatch) the regions of importance were between 4-8 Hz and between 8-12 Hz.

Ben Milner interpreted temporal filtering techniques as simply a matrix multiplication between a temporal filtering matrix and a "stacked" matrix of features formed by concatenating successive feature vectors [89, 90]. If the temporal filtering matrix consisted of a set of Discrete Cosine Transform (DCT) basis functions and the stacked matrix consisted of cepstral vectors, Milner called their product Cepstral-Time Matrices (CTMs). A subset of the elements in CTMs can be used as front-end features for ASR. Keeping a particular element in a CTM corresponded to choosing which modulation frequencies at which quefrencies to preserve. He empirically optimized the choices of elements in CTMs on different tasks and showed that 3.9-11.7 Hz in modulation frequency is best for isolated digits, 2.84-8.5 Hz is best for connected digits, and 3.9-15.6 Hz is best for a sub-word town names task [90].

Finally, Yuo et al. developed a robust feature for ASR by temporal filtering of the autocorrelation trajectories in speech [132]. They reasoned that if noise is uncorrelated with speech and if the noise is stationary[4], then the rate of change of the autocorrelation of noisy speech is equal to the rate of change of the autocorrelation of clean speech; therefore, this rate of change in the autocorrelation of noisy speech is a good feature to calculate if you want to get a noise-free representation of the clean speech. Calculating the rate of change in the autocorrelation sequences is analogous to applying a difference filter to the autocorrelation trajectories. These authors showed, unsurprisingly, that on artificially added noisy speech, their temporal filtering technique gave great ASR performance improvements.

[4] A big assumption because most noises are nonstationary.

Data-Driven Temporal Filters

All Neural TRAP-like systems, including the extensions to Neural TRAP presented in this thesis, are examples of data-derived temporal filters. The hidden units of the critical-band MLPs learn hyperplane separations in the long-term log energy trajectory feature space. These hyperplane separations are in fact discriminant temporal filters that help separate various phonetic sounds within the long-term log energy trajectories. These discriminant temporal filters are derived from the data because they are learned as a result of the error-back propagation algorithm with speech training data.

Others have tried to derive temporal filters from data in much the same way as the TRAP-like systems. Generally, the steps are as follows: first, form either spectral energy trajectories (spectral energy measurements over a sequence of frames) or feature component trajectories (like a particular PLP coefficient over a sequence of frames). Next, learn a linear projection in the space of these spectral energy/feature component trajectories to maximize (minimize) some goodness (error) function on training data. Finally, use these linear projections as temporal filters by applying them to incoming trajectories.

Here are a sampling of common linear transformation techniques that researchers have tried. Principal component analysis (PCA), also known as Karhunen Lòeve Transform (KLT), finds the linear transformation that projects the data onto axes in the directions of the maximal variation within the data [30]. Linear Discriminant Analysis (LDA) finds a linear projection that best maximizes the ratio of the between-class scatter to the within-class scatter of the projected data [30]. When applied to ASR, people generally use sub-word units like phones or HMM states as class labels for LDA. Independent Component Analysis (ICA) projects the data into dimensions that are as statistically independent from each other as possible [79]. Minimum Classification Error (MCE) [66] can be used to find the linear projection that minimizes the classification error function which is the likelihood ratio of the correct class models to the incorrect class models, where the classes are sub-word units like phones.

All of the above linear transformation techniques have been applied in the context of deriving temporal filters for ASR [10, 124, 81, 115, 117, 116, 61, 60]. These temporal filters can be applied on individual trajectories of MFCCs as in [81, 61, 60], or on individual log critical-band energies as in [10, 124, 115, 117, 116]. These temporal filters can

also be designed to have built-in robustness to certain environmental conditions by using training data corrupted by these environmental conditions as in [117]. In general, these data-driven linear transformation techniques for deriving temporal filters improved ASR performance more than other knowledge-driven temporal filtering techniques like velocity and acceleration features or RASTA-PLP. Unlike Neural TRAP and other TRAP-like extensions, these techniques only involve a linear transformation, while the temporal filters learned by Neural TRAP are nonlinear transformations capable of capturing more complex separations in temporal trajectories.

While there has been a lot of activity on deriving temporal filters within each MFCC coefficient or each critical-band energy trajectory, there is a body of work that allows for the learning of filters that span regions in the spectro-temporal plane. A simple example of such systems are the conventional hybrid HMM/ANN MLPs that take 9 frames of PLP feature vectores as input features and outputs phone probabilities [18]. The hidden units learn spectro-temporal filters or hyperplane separations in time and frequency. The neural nets developed by Antoniou et al. also uses more frames as inputs to learn discriminative spectro-temporal information [4, 3]. Recurrent Neural Nets are similar to these feed-forward MLPs, except that they allow for feed-back connections that can be used to learn temporal relations between successive feature vectors [106]. Time-Delay Neural Networks (TDNNs) [125] are similar to the standard hybrid HMM/ANN MLP in that they both can learn temporal relations in the input space of MFCC or PLP features.

Other activity on learning spectro-temporal filters include: Kajarekar's application of LDA jointly in both time and frequency in [67], Somervuo's experiments with other types of time-frequency transformations [119, 120], and work by Kleinschmidt et al. [73, 72] in deriving a set of Gabor shaped filters in time and frequency motivated by the existence of spectro-temporal receptive fields of neurons in the primary auditory cortex. Kleinschmidt et al. started from a pool of Gabor filter functions, each of which is defined by a product of a 2-dimensional Gaussian envelope and a complex exponential function which gives the the Gabor filter a ripple. From this pool, he picked a subset of these Gabor filters that gave the best performance on development data. In [73], Kleinschmidt and Gelbart reported a 7% relative improvement on the Aurora2 noisy digits task using this approach.

Chapter 3

Development of Novel TRAP-Like Classifiers

Chapter 1 motivated the approach of learning useful information within long spans of narrow-frequency channels in speech for ASR, and Chapter 2 reviewed previous related approaches. In this chapter, we introduce two new neural net architectures for the learning of phonetically discriminant critical-band temporal patterns. The first is called Hidden Activation TRAP (HAT) and the second is called Tonotopic Multi-Layer Perceptron (TMLP). We will describe both of these neural net architectures and the motivation leading to their design. This chapter also contains a set of initial experiments on a widely used continuous phone recognition task: TIMIT. We will show how HAT and TMLP reduce phone error rates on TIMIT while using 84% fewer parameters than a comparable Neural TRAP system.

3.1 Improving the Original Neural TRAP

As described in Chapter 2, Neural TRAP [52][1] takes a radically alternative approach to extracting phonetically discriminant information from speech. Instead of extracting phonetic information from spectral slices of short amounts of time (about 25 milliseconds), as conventional ASR systems do, Neural TRAP extracts phonetic information from separate frequency channels (critical-bands) spanning the full spectrum over a large

[1] Recall from Chapter 2 that TRAP is a mnemonic for TempoRAl Pattern.

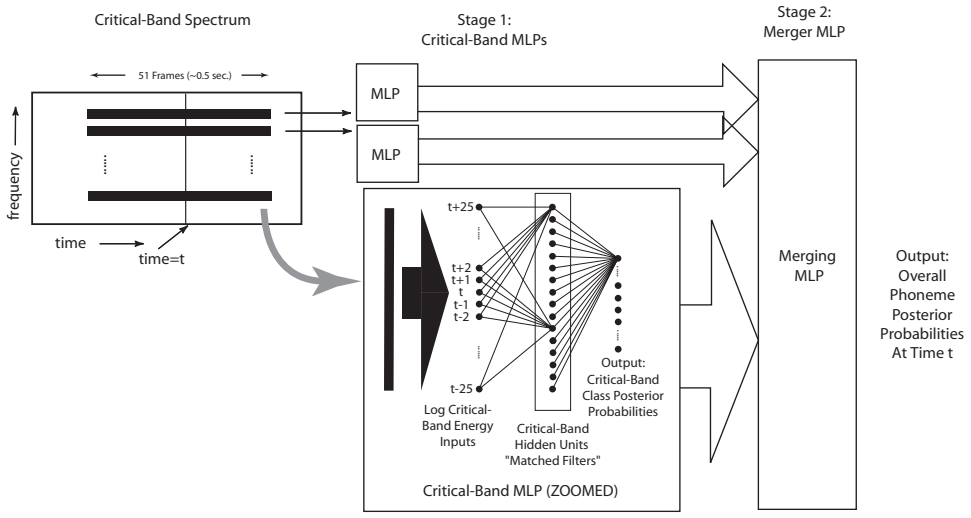

Figure 3.1: The Neural TRAP acoustic model with zoomed in view of a critical-band MLP.

amount of time (0.5 second to 1 second). In other words, Neural TRAP learns phonetically discriminant temporal information within narrow-frequency bands. It is capable of achieving comparable performance to conventional ASR systems, but using it in combination with conventional features, researchers have shown significant performance improvements in many conditions, especially in high noise conditions [53, 64].

Before developing the two new neural net extensions to Neural TRAP, we briefly review how the Neural TRAP system works. A Neural TRAP acoustic model as shown in Figure 3.1 consists of two stages of 3-layer fully-connected Multi-Layer Perceptrons (MLPs). The first stage is a nonlinear mapping from log critical-band energy time trajectories[2] to critical-band level phonetic probabilities, and the second stage consists of another MLP that combines these critical-band phonetic probabilities (one set per critical-band) to obtain the overall phonetic probabilities.

Let us focus our attention on the first stage of the Neural TRAP acoustic model. For each critical-band, there is an MLP trained using the standard error back-propagation algorithm [108] to learn phone posteriors by minimizing the cross-entropy [15] between the network output and target vectors. Each net takes, as input, a half second (or a 1 second as in [53, 112]) long log critical-band energy temporal trajectory consisting of 51 consecutive frames (one frame per 10 milliseconds calculated using a short-term FFT over

[2]A log critical-band energy time trajectory refers to a time sequence of log critical-band energy values.

25 milliseconds), and the training target is the phone label for the current frame. After training converges to a minimum, we can interpret the transformations happening in each of the layers.

Webb and Lowe in [129] derived a general result for nonlinear adaptive feed-forward layered networks, of which these critical-band MLPs are an example. Their central claim was that "minimising the error at the output of the network is equivalent to maximising a particular norm, the network discriminant function, at the outputs of the hidden units. The first part of the network is explicitly performing a nonlinear transformation of the data into a space in which the classes may be more easily separated. The specific nature of this transformation is constrained to maximise the network discriminant function." Although their result was derived with linear output units and sum of squares error function, a similar result can be derived for softmax output units and cross-entropy error criterion. According to Webb, the hidden units transform the input into a space that makes the classes more separate, while the output units map from this hidden space to the output class (or class probabilities in our case). Applying this interpretation to Neural TRAP, the hidden units of the first stage critical-band MLPs learn hyperplane separations in the input space of the 0.5 second long log critical-band energy trajectories. Another way to look at it is that they learn matched temporal filters on the temporal evolution of log critical-band energies useful for separating phonetic classes on the temporal evolution of the log critical-band energy, while the output units map the outputs of the matched temporal filters to phone probabilities.

In the original Neural TRAP system [52, 112] these critical-band MLPs learn 300 such matched filters for each critical-band. The hidden-to-output layer of these critical-band MLPs combine the outputs of the matched filters to form phone probabilities. The actual performance of these critical-band MLPs on phone classification is actually quite low. One way to see this is by measuring the frame classification accuracy. To compute the frame classification accuracy (or conversely, the frame classification error rate), we count how many times the maximum output (i.e., the class with the greatest posterior probability) of the MLP corresponds to the correct or labeled phone over all the frames in a test set. The accuracy is the ratio of this count divided by the total number of frames[3].

[3]Classification error is the ratio of the total number of frames minus this count divided by the total number of frames.

Critical-Band	Frequency Range (Hz)	MLP Frame Accuracy (%)
1	18-163	30.99
2	118-267	28.39
3	220-379	29.97
4	329-502	31.69
5	446-637	33.42
6	575-790	33.68
7	720-965	33.07
8	885-1165	32.93
9	1073-1397	31.72
10	1290-1667	30.73
11	1542-1982	29.58
12	1836-2350	30.48
13	2180-2782	28.80
14	2582-3289	27.82
15	3055-3885	27.93
16	3609-4587	28.67
17	4262-5412	29.54
18	5030-6383	30.33
19	5933-7527	29.37
1-19	18-7527	61.85

Table 3.1: Frame classification accuracy for first stage Neural TRAP critical-band MLP classifiers on the TIMIT cross-validation set. The half power cut-off points of each critical-band are also displayed. The MLPs are trained to classify 1 of 61 phones and each net has 300 hidden units. Chance performance is 12.13%. The last line in the table is the frame accuracy for the second stage Neural TRAP merger MLP.

Table 3.1 shows the frame classification accuracies of first stage Neural TRAP critical-band MLPs on the cross-validation data from TIMIT. For comparison sake, the frame classification accuracy from the Neural TRAP merger MLP is also shown. The frame accuracies for the critical-band MLPs range from 27.82% to 33.68% which is significantly greater than the chance performance of 12.13%[4]. Although the frame accuracies for the critical-band MLPs are much better than chance, they are much lower than the frame accuracy for the merger MLP which integrates information from the entire frequency range of the speech data. It seems that there is not enough information within a 0.5 second long log critical-band trajectory to accurately classify all phones, which is not surprising considering that different phones may look quite similar within a single narrow-frequency band.

To improve the Neural TRAP system, we think it is important to further examine and redesign the critical-band level classifiers. More specifically, we believe that mapping to phone probabilities at the critical-band level may not be optimal. This leads us to ask two questions:

1. Can we skip the mapping from the outputs of the matched filters to critical-band phone posteriors?

2. Is there a better way to train critical-band matched filters?

3.1.1 Can we skip the mapping from the outputs of the matched filters to critical-band phone posteriors?

We have noted how the low frame classification accuracies suggest that we cannot make all phone distinctions given only a single critical-band temporal energy trajectory. We hypothesize that whatever important phonetic information that can be gleaned from the critical-band trajectory is already captured by the matched filters (critical-band MLP hidden units). The additional mapping from the matched filters to phone posteriors may be an extraneous and inaccurate mapping. Why not skip this intermediate mapping and instead use the outputs from the matched filters from every critical-band as inputs for

[4]Chance performance assumes a classifier that always chooses the class with the highest prior probability in the training set. In the TIMIT training data, the silence phone is the class with the highest prior probability, and it makes up 12.13% of the cross-validation set.

the second stage merger? In this way, we hope to find a more accurate and parsimonious model.

3.1.2 Is there a better way to train critical-band matched filters?

Because training MLPs to learn phone posteriors from log critical-band temporal trajectories is too difficult a task, what categories, instead of phones, should we train the first stage Neural TRAP MLPs to learn? In [64] the critical-band classifiers are trained to learn six broad categories based on manner of articulation. One can also imagine training the critical-band classifiers to other linguistic feature-like classes that can be better distinguished at the critical-band level; however, it would be better to learn what categories are important from data. Furthermore, any training labels that we can specify at the sub-band level based on full-band phonetic labels may be inaccurate because of potential asynchrony among the sub-bands [93]. We experiment with a new model for Neural TRAP which consists of a single 4-layer neural network whose architecture resembles Neural TRAP and whose training procedure obviates the need to specify critical-band categorical targets - the log critical-band matched filters are learned automatically from data without specifying critical-band level labels.

3.2 Hidden Activation TRAP (HAT)

To answer the first question above, we have developed a variant of the Neural TRAP acoustic model that we call Hidden Activation TRAP (HAT). The HAT architecture is very similar to the Neural TRAP architecture, but it differs in one crucial aspect: the mappings from the critical-band hidden units to critical-band level phone posterior probabilities are discarded. More specifically, we train a bank of critical-band MLPs whose inputs are 51 frames of log critical-band energies and target labels are the labeled phone for the center frame. This training procedure is identical to the first stage training for the critical-band MLPs of Neural TRAP; however, the choice of how many hidden units is determined from frame accuracy curves (more about this below). Once the critical-band MLPs are trained, we "chop off" the hidden-to-output layer of every critical-band MLP, leaving only the outputs ("activations") of the hidden layer (hence, Hidden Activation TRAP). After error back-propagation training, one can interpret these hidden layer acti-

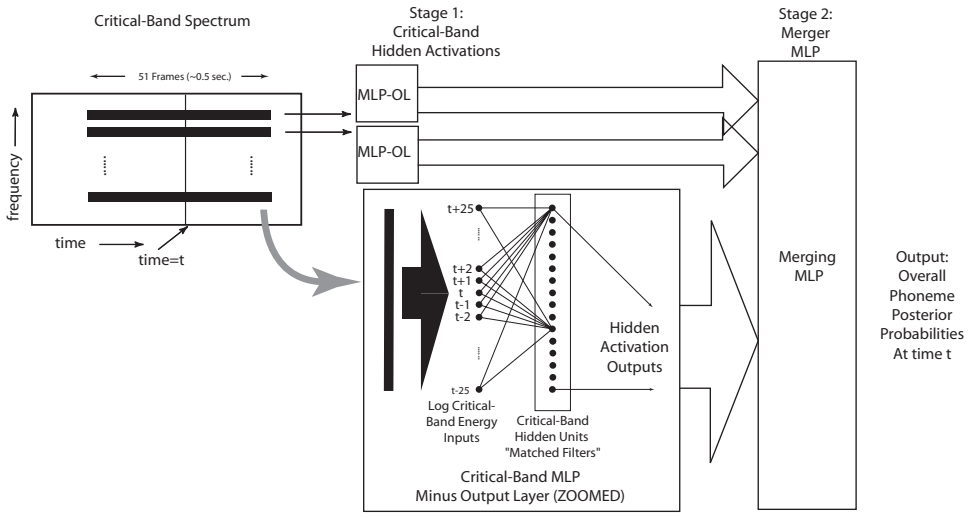

Figure 3.2: Hidden Activation TRAP (Note: MLP-OL stands for MLP minus the output layer).

vations as the outputs of discriminatively trained critical-band matched filters. The second stage of HAT is just like Neural TRAP: a merger MLP (trained using the same training set and cross-validation set as in the first stage critical-band training) takes the hidden activations from all the critical-band MLPs and learns the mapping to phone posteriors. The HAT setup is shown in Figure 3.2.

It may seem that we don't gain much from this HAT approach except reducing the number of parameters via the chopping off procedure, but we can further reduce the number of parameters significantly by reducing the number of matched filters required per critical-band. In conventional Neural TRAP this number was set to 300 per critical-band. There are two ways to determine an optimal number of hidden units (or matched filters) per critical-band. One way is to train a series of critical-band MLPs with an increasing number of hidden units for every critical-band and then examine where the "knees" in the frame accuracy curves occur. For every critical-band, we have plotted the MLP frame accuracy on a cross-validation set versus the number of hidden units in Figure 3.3. From this figure, we notice that most of the steepest increases in accuracy have already occurred when the number of hidden units has been increased to 20 or 25.

Another way to determine an optimal number of matched filters for HAT, is to train several complete HAT models that differ only in the number of matched filters per critical-band. For fair comparison, we kept the total number of parameters constant (about

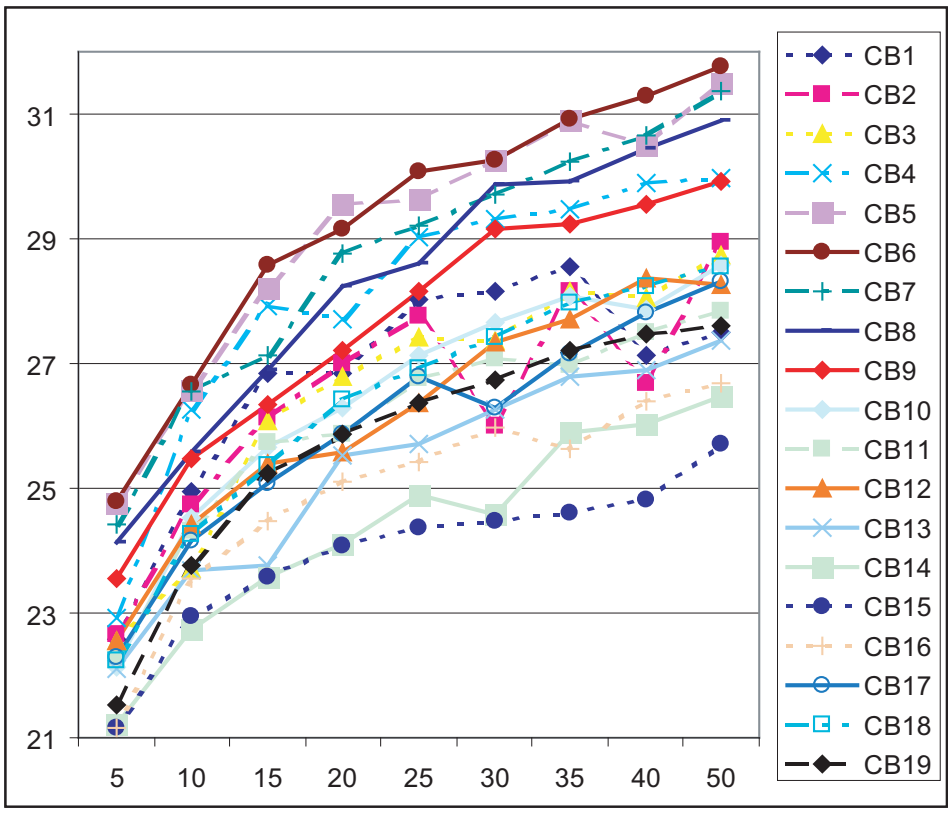

Figure 3.3: Frame accuracies of 19 critical-band MLPs on the TIMIT cross-validation data as a function of number of hidden units per critical-band.

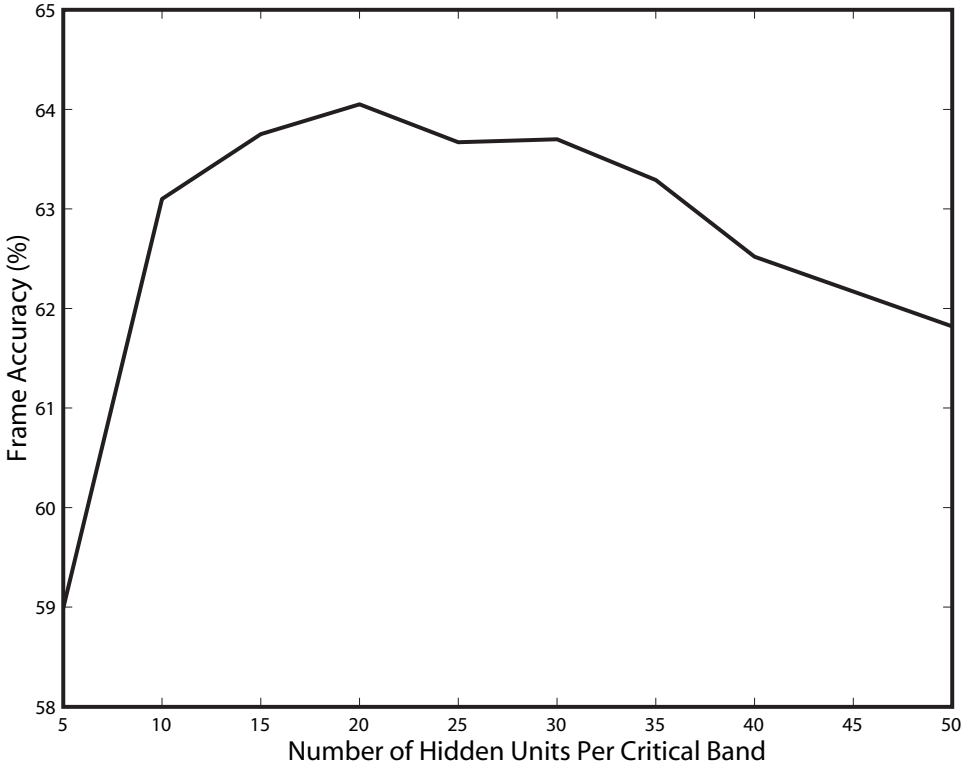

Figure 3.4: HAT frame accuracy on the TIMIT cross-validation data as a function of number of hidden units per critical-band.

160,000 total neural net weights and biases with 1.12 million frames of training data). The frame classification accuracy of these HAT models on the TIMIT cross-validation data set has an optimal performance peak at 20 matched filters per critical-band as seen in Figure 3.4.

3.3 One Stage Training: Tonotopic Multi-Layer Perceptron(TMLP)

To examine question 2 from above, we have also created a 4-layer MLP that trains the critical-band matched filter without the need for specifying critical-band level targets. We call this MLP the Tonotopic Multi-Layer Perceptron (TMLP), which is inspired by the tonotopic organization of the human peripheral auditory system, where different positions in the cochlea are sensitive to different frequencies. The first hidden layer of the TMLP is

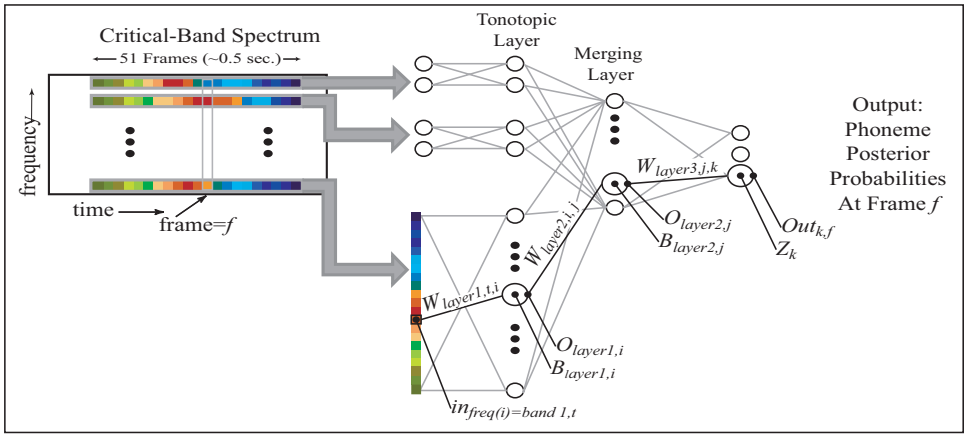

Figure 3.5: Tonotopic Multi-Layer Perceptron.

tonotopically organized into several sets of hidden units. Each of these sets is constrained to see inputs coming only from a single critical-band, and together, all of the sets span the frequency range of speech. The second hidden layer, as well as the output layer are fully-connected with their previous layers. Figure 3.5 shows the structure of a TMLP. We also refer to the first layer hidden units as critical-band hidden units.

As in HAT and Neural TRAP we use log critical-band energies as inputs to the TMLP. After computing the log critical-band energies of speech every 10 milliseconds and normalizing these energies by subtracting/dividing the mean/standard deviation calculated over each utterance, we take 51 consecutive frames (about 500 milliseconds) of these normalized energies as the input layer of the TMLP. The output of the ith first layer hidden unit for frame f is given by Equation 3.1:

$$O_{layer1,i} \stackrel{\text{def}}{=} sig\left(\sum_{t=f-25}^{f+25} in_{freq(i),t} W_{layer1,t,i} + B_{layer1,i}\right) \tag{3.1}$$

where $sig(x)$ is the logistic sigmoid function given in Equation 3.2. $in_{freq(i),t}$ is the tth frame of energy in the one and only one frequency band that the ith first layer hidden unit is constrained to see. $W_{layer1,t,i}$ and $B_{layer1,i}$ are the trainable weights and bias respectively for the ith unit.

$$sig(x) \stackrel{\text{def}}{=} \frac{1}{1+\exp(-x)} \tag{3.2}$$

The second layer of hidden units takes the outputs of all first layer hidden units

as inputs. The output of the jth second layer hidden units is given by Equation 3.3:

$$O_{layer2,j} \stackrel{\text{def}}{=} sig\left(\sum_I O_{layer1,i} W_{layer2,i,j} + B_{layer2,j}\right) \tag{3.3}$$

$W_{layer2,i,j}$ and $B_{layer2,j}$ are the trainable weights and bias respectively for the jth second layer hidden unit. Finally, the outputs of the TMLP are given by Equation 3.4:

$$Out_{k,f} \stackrel{\text{def}}{=} \frac{\exp(Z_k)}{\sum_K \exp(Z_k)} \tag{3.4}$$

where Z_k is given by Equation 3.5:

$$Z_k \stackrel{\text{def}}{=} \sum_J O_{layer2,j} W_{layer3,j,k} + B_{layer3,k} \tag{3.5}$$

$W_{layer3,j,k}$ and $B_{layer3,k}$ are the trainable weights and bias for the kth output unit.

Just like the HAT and Neural TRAP merger training, the TMLP is trained with output targets that are "1.0" corresponding to the phone labeled in the current frame, and "0" for all others. The TMLP is also trained to minimize cross-entropy error by using the error back propagation algorithm. Unlike HAT and Neural TRAP, the critical-band level categories of the TMLP corresponding to the critical-band hidden units are learned as a part of the overall error back-propagation. This obviates the need to specify any kind of critical-band training targets because the one stage training learns what is important for phone discrimination.

3.4 Discussion: Learning in HAT and TMLP

Having described the two new architectures for learning discriminant temporal information, it is instructive to discuss the nature of the speech information that these two models can extract. Just as in the Neural TRAP case, both HAT and TMLP are designed to learn phonetically discriminant information within long spanning (around 500 milliseconds) narrow-frequency channel (critical-bands) energy trajectories. All of these models first constrain the learning within critical-bands, and then integrate the discriminant information from all critical-bands. Another way to say this is that Neural TRAP, HAT, and TMLP impose a constraint upon the learning of temporal information from the

time-frequency plane: correlations among individual frames of energies from different frequency bands are not directly modeled. Instead, they model correlation between long-term energy trajectories from different frequency bands.

It is also interesting to note that TMLP places less constraints on the learning of discriminant temporal trajectory information than HAT and Neural TRAP. Because TMLP is a single neural network whose parameters are learned via the gradient descent error back-propagation algorithm, the critical-band hyperplane separators in TMLP are not constrained to learn discriminants that are optimal for separating phone targets at the critical-band level. They can learn whatever is best for the next hidden layer to do its job. HAT and Neural TRAP learn the critical-band hyperplane separators that are best for separating the phones based on the critical-band level labels that we provide. As described in Sub-section 3.1.2, our critical-band level phone labels may not be the right classes to learn, and they may also be inaccurate. Because HAT is a more constrained model than TMLP, the family of distributions that TMLP can learn is larger, and because HAT has the same connections as TMLP, the family of distributions that HAT can learn is a subset of that for TMLP. Figure 3.6 shows a cartoon picture of the family of distributions learned by these two new Neural TRAP extensions.

While it is true that TMLP can potentially model a richer family of distributions compared with HAT, sometimes constraints can be helpful. In cases when training data is sparse, constraints can help the classifier focus on learning the important details. Also, in cases when there is noise in the data, constraints can help the classifier ignore irrelevant and misleading information.

3.5 Experimental Setup

In subsequent sections, we will present experiments demonstrating the performance of HAT and TMLP on a small phone recognition speech task, so in this section, we describe our experimental setup. We use the TIMIT database [40] for the experimental work in this chapter. The TIMIT speech database, recorded at TI and transcribed at MIT (hence TIMIT), consists of about 4.27 hours of speech spoken by 630 different speakers from the 8 major dialect regions in the United States. It was recorded at 16,000 Hz with a close talking microphone in the studio. The prompts spoken by the speakers were designed

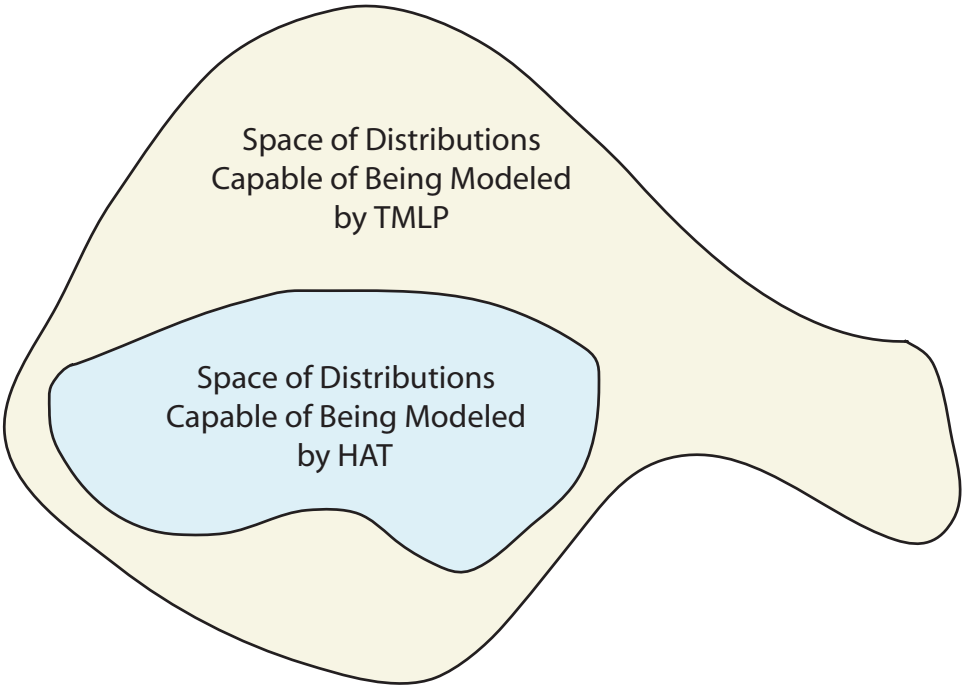

Figure 3.6: Cartoon of the family of distributions modeled by TMLP and HAT.

to provide a good coverage of pairs of phones and to be diverse in sentence types and phonetic contexts. See timit.readme file found in [40] for more details. There are a total of 2,342 unique prompts found in TIMIT, and they do not sound completely like sentences people would naturally utter. For example, the most famous TIMIT prompt is: "She had your dark suit in greasy wash water all year". Because of the odd nature of the prompts and because there are so few of them, speech recognition researchers have tended to use TIMIT for phone recognition experiments only.

Using the recommended training set consisting of 3,696 utterances, we set aside 10% of these utterances (370 utterances, 111,446 frames, .31 hours) for a separate cross-validation set, and keep the remaining 90% (3,326 utterances, 1,124,823 frames, 2.81 hours) for training our various MLPs. The cross-validation set is used for adjusting the learning rate during MLP training and also for determining the early stopping point to prevent overfitting. For all of our test results we use the complete TIMIT test set consisting of 1,344 utterances (410,920 frames, 1.14 hours) and 51,664 total phone tokens.

In the experiments of this chapter, we use the hybrid ANN/HMM speech recog-

nition framework [18] described in Chapter 2. The artificial neural nets estimate phone posteriors. These posteriors are then scaled by the phone priors to produce the scaled likelihoods needed for the HMM back-end Viterbi decoder. We use the Chronos decoder [107] as well a standard phone bigram language model during decoding.

Each of the various neural nets is trained to learn the original 61 TIMIT phones shown in Table 3.2. The best phone sequence decoded by Chronos is at first a sequence of these 61 original phones. In many previous studies using TIMIT, researchers map these 61 phones into a smaller set of 39 phones [77] and report their results using this smaller phone set. Table 3.2 also shows the mapping from the original 61 phones to this smaller 39 phone set. We perform the same mapping on the best phone sequence decoded. To obtain our final phone error rate which is the sum of %substitutions+%deletions+%insertions, we perform a standard dynamic programming string alignment to the TIMIT test set's reference phone sequences which are also mapped to the 39 phone set[5].

In the following sections we present results in clean condition as well as in noisy and reverberant conditions. Please note, however, that all training was done using clean speech, so that we can test the robustness of each of the systems to unseen conditions. We have experimented with two noisy conditions: Mercedes Benz noise (recorded inside the car) and exhibition hall noise (containing mainly speech babble, e.g., people talking in the background). The noise files come from the Wall Street Journal Task for the AURORA2 evaluations [56]. We add these noises to the clean files at different signal to noise ratios. We also convolve the clean signals with a room impulse response to give a moderately reverberant testing condition. This room impulse response has a 60 dB reverberation time of 0.8 seconds which means that it takes 0.8 seconds for the echoes to become 60 dB less powerful than the original speech signal. We are grateful to Jim West, Gary Elko, and Carlos Avendaño who collected this impulse response in the Bell Labs Varechoic chamber and made it available to our research group [126, 9].

The features fed to our various TRAP-like acoustic models are calculated from the clean, noisy and reverberant speech waveforms. These features are log critical-band

[5]The phone error rates that we obtain using this simple 61 to 39 mapping are actually underestimates of our potential performance. Lower phone error rates can be obtained by performing the mapping at an earlier stage. By summing the posterior probabilities corresponding to phones from the 61 phone set that are mapped to a single phone from the 39 phone set, posteriors for the 39 phone set can be obtained. Using these 39 phone set posteriors for decoding leads to lower phone error rates, but for simplicity chose to perform the mapping after the decoding step.

ASR Phoneme Symbols					
TIMIT 61	Example	TIMIT 39	TIMIT 61	Example	TIMIT 39
b	**b**ee	b	l	**l**ike	l
d	**d**ay	d	el	bott**le**	l
g	**g**ay	g	r	**r**ight	r
p	**p**ea	p	w	**w**ire	w
t	**t**ea	t	y	**y**es	y
k	**k**ey	k	hh	**h**ay	hh
dx	dir**t**y	dx	hv	a**h**ead	hh
bcl	(b closure)	h#	iy	b**ee**t	iy
dcl	(d closure)	h#	ih	b**i**t	ix
gcl	(g closure)	h#	eh	b**e**t	eh
pcl	(p closure)	h#	ey	b**ai**t	ey
tcl	(t closure)	h#	ae	b**a**t	ae
kcl	(k closure)	h#	aa	f**a**ther	aa
jh	**j**oke	jh	aw	ab**ou**t	aw
ch	**ch**oke	ch	ay	b**i**te	ay
s	**s**ound	s	ah	b**u**t	ax
sh	**sh**out	zh	ao	b**ou**ght	aa
z	**z**oo	z	oy	b**oy**	oy
zh	a**z**ure	zh	ow	b**oa**t	ow
f	**f**ish	f	uh	b**oo**k	uh
th	**th**in	th	uw	b**oo**t	uw
v	**v**ote	v	ux	t**oo**t	uw
dh	**th**en	dh	er	b**ir**d	er
m	**m**oon	m	axr	butt**er**	er
em	botto**m**	m	ax	**a**bout	ax
n	**n**oon	n	ax-h	s**u**spect	ax
nx	wi**nn**er	n	ix	deb**i**t	ix
ng	si**ng**	ng	h#	(non-speech events)	h#
eng	washi**ng**ton	ng	pau	(pause)	h#
en	butto**n**	n	epi	(epenthetic silence)	h#
q	(glottal stop)	h#			

Table 3.2: The 61 original TIMIT phones, their 39 phone equivalents, and an example of the phone.

energies calculated for every critical-band and for each frame every 10 milliseconds. The mean and standard deviation of the energies from each critical-band are calculated and subtracted (divided in the case of standard deviation) on a per utterance basis. 51 consecutive frames of the log energies from each critical-band form the input features for our systems at the time corresponding to the 26th frame. These 51 consecutive log energy values form critical-band energy trajectories spanning a time context of half a second which is twice as long as the average syllable duration of 250 milliseconds.

3.6 Clean Results

In order to demonstrate the performance of our two new temporal ASR systems in clean conditions, we trained and tested four systems according to the experimental setup described in section 3.5. This section presents results of experiments in clean conditions, where "clean" refers to the fact that we did not artificially contaminate either the training or test sets with noise nor reverberation. Speaker and speaking variations, however, are still present within the recordings.

We trained a Neural TRAP baseline, a HAT, a TMLP, and a conventional hybrid ANN/HMM ASR system that uses 9 frames of PLP features. The baseline Neural TRAP system is similar to the one presented in [53]. This Neural TRAP system has 300 hidden units per critical-band MLP and a merger MLP with 317 hidden units for a total of 1,032,377 trainable parameters. The HAT system has 20 hidden units per critical-band and also 317 hidden units for the merger. The total number of parameters for the HAT system is 159,935. The TMLP system also contains 20 hidden units per critical-band, 317 hidden units for the merger and has the same number of parameters as the HAT system. Finally, for comparison with a conventional ANN/HMM system, we made a PLP system that uses 12th order PLP [48] plus energy and first and second derivatives as input features. These features undergo a per-utterance mean and variance normalization and are then fed to an MLP with 9 frames of input context which estimates the phone posteriors and contains roughly 160,000 parameters also. The results on the uncorrupted TIMIT test set are shown in Table 3.3 where PLP denotes the conventional ANN/HMM system. We have also added a column for relative improvements of the new temporal systems compared with the baseline Neural TRAP. Because PLP differs significantly from the temporal systems

System Description	Phone Error Rate (%)	Relative Improvement (%)
Baseline: Neural TRAP	32.7	-
HAT	29.8	8.9
TMLP	31.0	5.2
PLP	29.7	N/A

Table 3.3: Phone error rates of 3 different temporal ASR systems and a typical ASR system on the full TIMIT test set mapped to 39 phones under clean conditions.

System Description	Phone Error Rate (%)	Relative Improvement (%)
Baseline: PLP	29.7	-
PLP+Neural TRAP	27.2	8.4
PLP+HAT	26.5	10.8
PLP+TMLP	26.8	9.8

Table 3.4: Phone error rates of the frame-wise product of posterior combination of 3 temporal MLPs and a PLP MLP on the full TIMIT test set under clean conditions.

which focus on learning long narrow-frequency patterns rather than short spectral slices, the relative improvement comparison to Neural TRAP is not appropriate.

In addition to these stand-alone results, we have also tried combining all temporal systems with the conventional PLP system. Because the PLP system is extracting information from spectral slices and not from critical-band energy trajectories, we expect to see great improvements when the temporal systems are combined with PLP. Although there are more elaborate ways to combine posterior probabilities, we simply multiplied the posterior probabilities from the two different systems and scaled them by the square of the priors for each phone. This implies the that the two probability streams are conditionally independent given the underlying phone. This combination technique has worked well in prior combination studies [65]. The phone error rates of the combination systems on the TIMIT test set are shown in Table 3.4.

3.7 Clean Discussion

HAT outperforms Neural TRAP by 2.9% absolute on the TIMIT test set consisting of 51,664 phone tokens. This result is significant at the 0.05 level using a "difference of proportions" significance test. This particular significance test assumes that the two error rates are samples from a binomial distribution, and then tests the two binomials for being significantly different using a Z-score. TMLP also outperforms Neural TRAP, but this time by only 1.7% absolute. This too is statistically significant at the 0.05 level. The difference in performance between HAT and TMLP is also statistically significant at the 0.05 level; however, the difference between HAT and the conventional PLP system is not statistically significant. From this, we see that both of the two new temporal systems outperform Neural TRAP, and HAT is comparable in phone recognition performance to the conventional PLP system.

With only 20 discriminative patterns per critical-band in the HAT and TMLP systems, we can achieve better phone recognition performance in clean conditions than Neural TRAP which uses 300 discriminative patterns per critical-band. Additionally, the HAT and TMLP systems have 84% fewer parameters than Neural TRAP. Because HAT outperforms Neural TRAP, we can begin to answer question 1 from above; dropping the additional mapping from hidden unit activations to critical-band phone posteriors helps. Unfortunately, in clean conditions, it does not yet seem helpful to unconstrain the critical-band learning targets because TMLP does not outperform HAT. Constraints are often useful when there is not enough data, suggesting perhaps that TMLP might not be getting enough data for training.

The combination of all of these temporal systems with the conventional PLP system all give wonderful performance improvements over the PLP system alone (8.4% - 10.8% relative improvements). The difference in performance between the combination of HAT with PLP and the combination of Neural TRAP with PLP is statistically significant at the 0.05 level; however, the difference between the Neural TRAP combination and the TMLP combination is only significant at the 0.01 level. HAT gives the most improvement of all the temporal systems in combination with PLP.

Test Condition	System Description			
	Neural TRAP	HAT	TMLP	PLP
Reverberant	56.3%	**54.2%**	58.0%	59.2%
Benz Noise				
20 dB	35.9%	**33.8%**	35.5%	36.5%
10 dB	42.7%	**42.2%**	42.8%	**42.2%**
0 dB	55.0%	56.7%	54.2%	**50.5%**
Exhib. Noise				
20 dB	41.6%	**39.9%**	41.8%	40.4%
10 dB	61.4%	63.4%	62.0%	**60.0%**
0 dB	102.2%	95.7%	**86.5%**	95.9%

Table 3.5: Phone error rates of the four systems on the TIMIT test set mapped to 39 phones under various noise and reverberant conditions. The noises are added at 3 different signal-to-noise ratios (20 dB, 10 dB, and 0 dB), and the best system performances are in bold.

3.8 Noisy and Reverberation Results

We have also tested our new temporal systems in noisy and reverberant conditions. For noisy test conditions, we artificially added two types of noises at different signal-to-noise ratios. For the reverberant conditions, we convolved a room impulse response to the test sets as described in Section 3.5. Table 3.5 shows the stand-alone phone error rates of Neural TRAP, HAT, TMLP, and PLP, while Table 3.6 shows the phone error rates for the three temporal systems in combination with PLP.

3.9 Noisy and Reverberation Discussion

It has been shown in [114], that in reverberant conditions, systems that use discriminant temporal filters are more effective than conventional features. Here, we see that all of the other temporal systems significantly outperform PLP in moderate reverberation. HAT performs the best at 54.2%, and Neural TRAP is better than TMLP. Again, in combination with PLP, these temporal systems add additional improvements to PLP alone. The combination of HAT and PLP is the best followed by Neural TRAP and PLP, and TMLP and PLP. From this, we conclude that the long-term temporal processing techniques are more effective in dealing with reverberation than the shorter-term spectral processing of

Test Condition	Combination System		
	PLP+ Neural TRAP	PLP+HAT	PLP+TMLP
Reverberant	52.9%	**52.4%**	54.1%
Benz Noise			
20 dB	30.9%	**30.7%**	30.9%
10 dB	**35.9%**	36.2%	36.3%
0 dB	**44.9%**	45.8%	**44.9%**
Exhib. Noise			
20 dB	36.2%	**35.8%**	36.5%
10 dB	**54.4%**	55.7%	55.6%
0 dB	79.9%	**65.8%**	81.3%

Table 3.6: Phone error rates of the combined systems on the TIMIT test set mapped to 39 phones under noise and reverberant conditions. The noises are added at 3 different signal-to-noise ratios (20 dB, 10 dB, and 0 dB), and the best system performances are in bold.

PLP. Constraining the critical-band hidden units to learn discriminants useful for critical-band phone targets as we do with HAT and Neural TRAP is more effective than TMLP's global optimization in reverberant conditions.

When corrupting the test set with Mercedes Benz noise which predominantly has spectral energy in the low frequencies, we see that the performance depends on the signal-to-noise ratio (SNR). In 20 dB and 10 dB SNRs, HAT outperforms both Neural TRAP and TMLP, but in 0 dB SNR, both Neural TRAP and TMLP outperform HAT. Compared with PLP, the temporal systems only show better results in 20 dB SNR. The combination results are quite comparable with none of the temporal systems showing significant advantages over one another except both the Neural TRAP and TMLP combinations with PLP outperform the HAT combination at 0 dB SNR. As in all previous conditions including the clean condition, the combination of the temporal systems with PLP greatly reduces the phone error rates compared to non-combined systems.

Exhibition hall noise is the toughest noise condition because the noise is speech. Among the temporal systems, there is no clear winner because HAT does best at the 20 dB SNR level, and Neural TRAP is the winner at the 10 dB SNR level, and TMLP wins at the 0 dB SNR level. PLP also performs at roughly the same levels as the temporal systems. Finally, the combination of PLP with the temporal systems provide a huge boost in performance, lowering the phone error rates greatly.

3.10 Narrow-Band Discriminant Temporal Patterns

As described in Section 3.1, one can consider that the critical-band hidden units of HAT and TMLP learn matched temporal filters useful for phonetic classification on the temporal evolution of the log critical-band energy. These matched filters detect certain narrow-band discriminant temporal patterns for phonetic classification; when these patterns are present in the speech, critical-band hidden units tuned to detect these patterns output high activation values. As with any filter, these matched filters also have a frequency response. Since these matched filters operate on the time evolution of energy within a frequency band of speech, their frequency response shows which rates of change in the energy trajectory a matched filter is sensitive to. These rates are called modulation frequencies and are described further in Section 2.1.5. It has been shown by many researchers that modulation frequencies between 0 and about 16 Hz are important for speech recognition [58, 59, 29, 5, 68].

It is interesting to see what are the narrow-band discriminant temporal patterns that HAT and TMLP have learned after training them to perform phonetic classification on TIMIT. In Appendix C we plot cluster centroids of these patterns for both HAT and TMLP. More specifically, we take the input-to-hidden unit weights of each critical-band hidden units (these are the matched filters), and then cluster them agglomeratively since there are too many of them to display and since many of them resemble each other. We then plot these patterns and their corresponding modulation frequency responses. Figures 3.7, 3.8 show examples of discriminant temporal patterns and corresponding modulation frequency responses learned by the HAT trained on TIMIT, and Figures 3.9, 3.10 show examples for the TMLP trained on TIMIT. The discriminant temporal patterns are centered at frame 0 (x-axis) and range from 25 frames in the past (-25) to 25 frames in the future (25). There are 51 total frames which spans about 500 milliseconds of context. The y-axis for the patterns is the magnitude. The x-axis for the corresponding modulation frequency response is the modulation frequency, while the y-axis is the filter gain in decibels.

From these examples and others like them in Appendix C, we observe that all of the patterns are sensitive to modulation frequencies between 0 and about 20 Hz. This is nearly consistent with previous findings about the importance of low modulation frequencies for speech recognition. Another interesting observation is that some of these

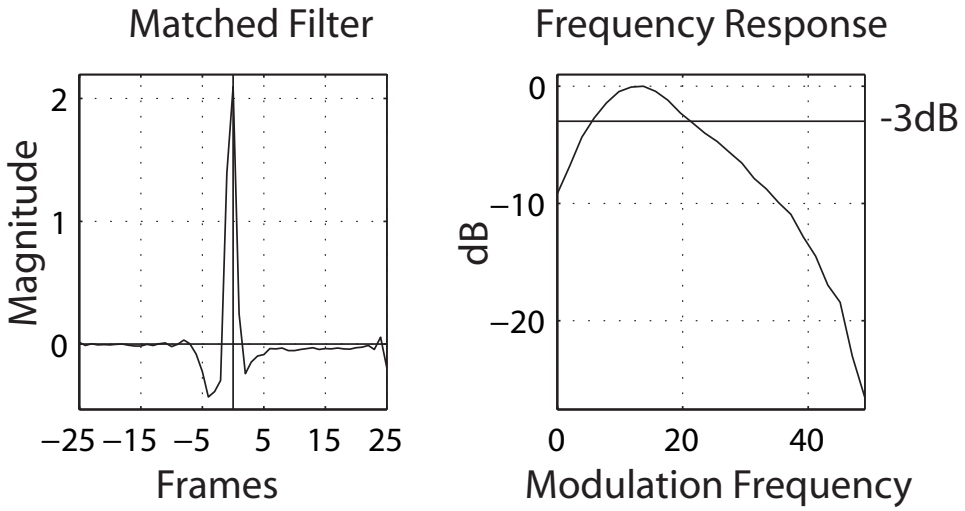

Figure 3.7: An example input to critical-band hidden unit weight pattern (matched filter) for the HAT trained on TIMIT and its corresponding frequency response.

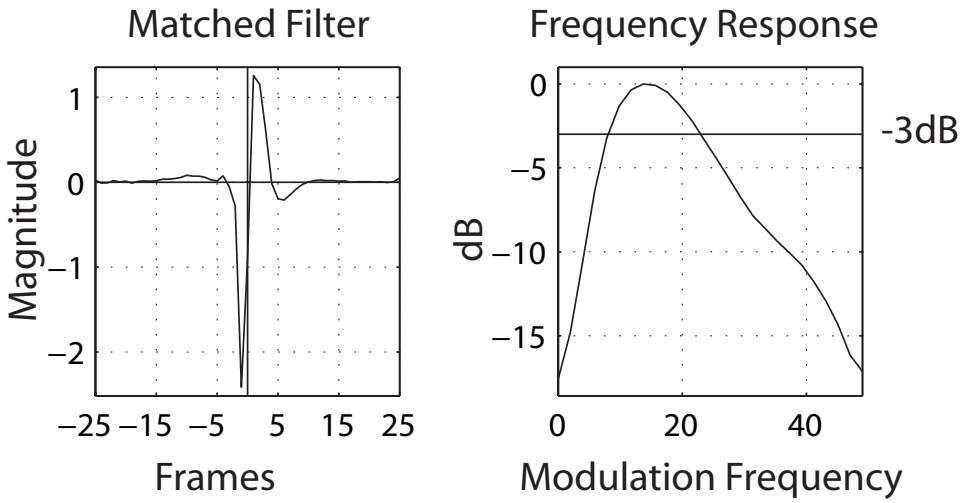

Figure 3.8: An example input to critical-band hidden unit weight pattern (matched filter) for the HAT trained on TIMIT and its corresponding frequency response.

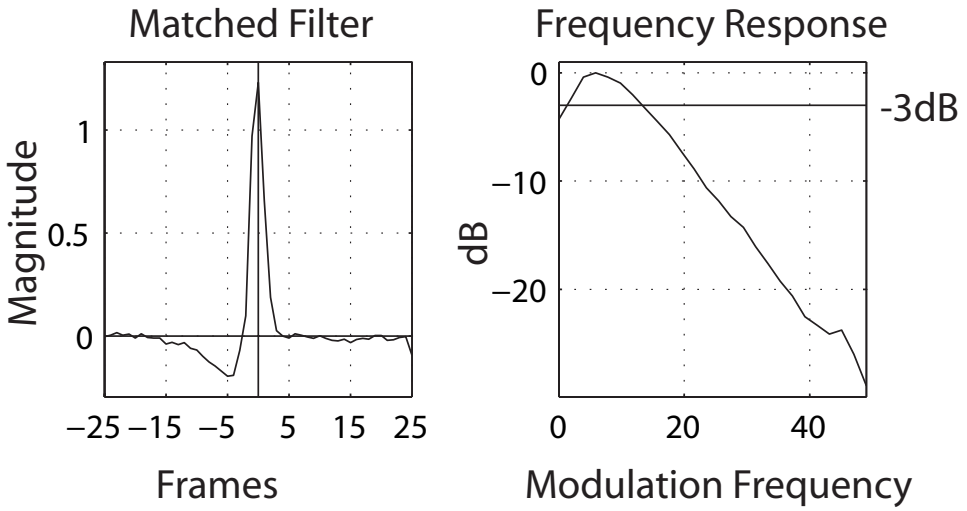

Figure 3.9: An example input to critical-band hidden unit weight pattern (matched filter) for the TMLP trained on TIMIT and its corresponding frequency response.

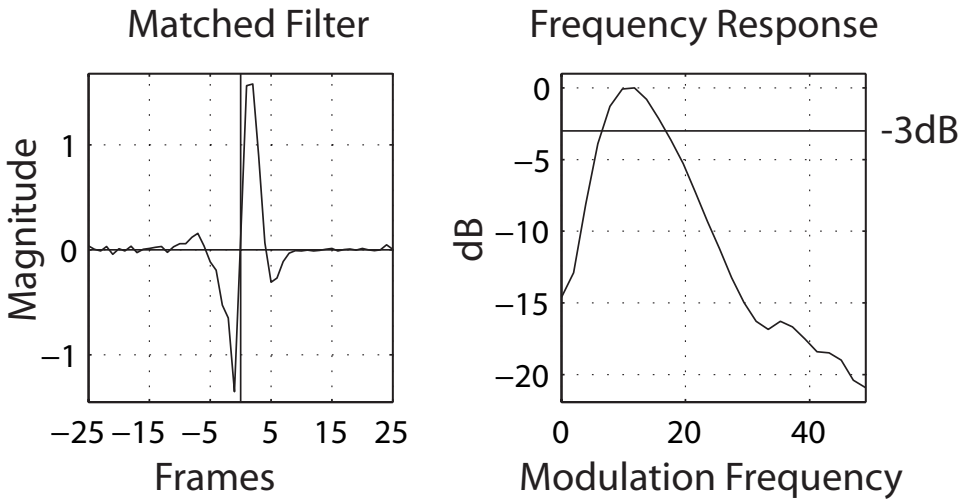

Figure 3.10: An example input to critical-band hidden unit weight pattern (matched filter) for the TMLP trained on TIMIT and its corresponding frequency response.

patterns resemble temporal patterns used by other researchers for the temporal filtering of speech. Figures 3.7 and 3.9 resemble the "Mexican hat" filters which detect high energy, and Figures 3.8 and 3.10 resemble the "derivative" filters which detect onsets of energy. Both of these patterns were learned when applying Linear Discriminant Analysis (LDA) to temporal energy trajectories in [10, 124, 115, 67]; moreover, some of the patterns in Appendix C look similar to the Mean TRAPs found in [112]. From a reusability standpoint, the similarity of these patterns in future applications important. Since certain temporal patterns seem to appear over and over again as a result of training using different approaches on different training databases, it would be reasonable to fix them and reuse them in future ASR applications on different tasks as a preprocessing step in feature extraction. In Chapter 5 we look at the patterns learned by HAT and TMLP trained on conversational telephone speech and also compare them to patterns learned using both Principal Components Analysis (PCA) and LDA.

3.11 Conclusions

In this chapter we have developed two new variants to Neural TRAP: Hidden Activation TRAP (HAT) and Tonotopic Multi-Layer Perceptron (TMLP). Both have been shown to drastically reduce the number of parameters required while improving the phone recognition performance under clean condition compared to Neural TRAP. We have found that approximately 20 discriminative temporal filters per critical-band is sufficient to perform TIMIT phone recognition. By showing how HAT outperforms Neural TRAP, we have shown that skipping the mapping from the outputs of the discriminant matched filters to critical-band phone posteriors is helpful. So far, we have not noticed any significant advantages to allowing the critical-band filters to be globally optimized (as in TMLP) and not constrained to learn separators for critical-band level phone targets (as in HAT).

In noisy and reverberant conditions, these temporal systems (Neural TRAP, HAT, and TMLP) show varying degrees of improvements. Under additive noise conditions all temporal systems are comparable to the PLP system. In a moderately reverberant condition, all temporal systems outperform the traditional PLP system.

We have also seen how effective it is to combine the temporal systems which learn discriminant long-term narrow-frequency patterns with conventional systems which learn

discriminants in spectral slices. All combination results in every condition tested outperform all uncombined results. Our clean combination results are close to the best published TIMIT phone recognition error rate that we are aware of. The PLP+HAT combination error rate on clean, 26.5%, is slightly greater than the best published TIMIT phone recognition error rate of 24.2% in [3]. Finally, the narrow-band discriminant temporal patterns learned by both HAT and TMLP in this chapter preserve the low modulation frequencies of speech which are important for recognition.

Chapter 4

Temporal Systems for CTS

In the previous chapter, we introduced two new temporal ASR systems, HAT and TMLP, and showed promising improvements over Neural TRAP on a small phone recognition task. In this chapter, we explore the integration of Neural TRAP into a state-of-the-art Gaussian mixtures-based HMM recognizer. Our goal is to develop a baseline temporal ASR system setup that is capable of competitive performance on the challenging task of conversational telephone speech (CTS). Once this baseline setup is developed, we will be able to compare various temporal feature extraction techniques like HAT and TMLP to Neural TRAP on CTS in subsequent chapters.

Our basic approach to the baseline setup uses the phone posteriors estimated by Neural TRAP to augment conventional front-end features. This chapter presents a series of experiments on progressively more difficult speech recognition tasks that we use to guide the design of our baseline setup and to show how our approach can improve ASR performance over a wide range of speech data.

4.1 Posterior Probabilities as Features

For decades, the feature extraction component of speech recognition engines has consisted of some form of local spectral envelope estimation, typically with some simple transformation. Current typical front-ends consist largely of the Mel cepstrum [87] or PLP [48] computed from an analysis window of roughly 25 or 30 ms surrounding a central signal point, stepped along every 10 ms. These features are often augmented by delta

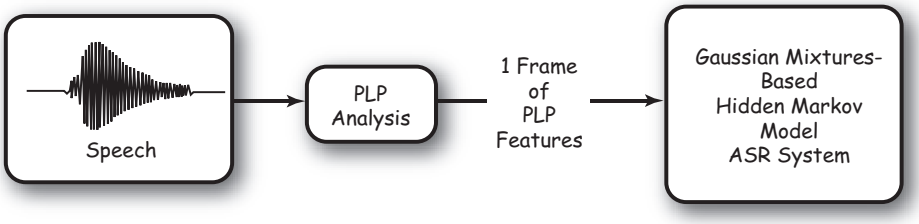

Figure 4.1: Block diagram of a conventional ASR system using PLP front-end features for a standard Gaussian mixtures-based HMM system.

features [38] and transformed by various linear transformations (e.g., linear discriminant analysis and heteroskedastic discriminant analysis) which makes the effective temporal context of these features around 90 milliseconds. A picture of the conventional ASR system using PLP features is shown in Figure 4.1.

These standard front-end features were designed based on expert knowledge. In recent years, there has been a push for more data-driven approaches for deriving front-end features. One such approach, Tandem acoustic modeling [49, 34, 32] as described in Chapter 2, uses an MLP to learn posterior probabilities of phonetic units. These posterior probabilities are then transformed and used as features for a standard Gaussian mixtures-based hidden Markov model (GMHMM). The transformations applied to the posteriors are designed to make the resulting features more Gaussian and decorrelated which tend to help the Gaussian mixture models with diagonal covariance matrices better model these features. The transformations are the logarithm followed by principal components analysis (PCA). Figure 4.2 shows a typical Tandem ASR system.

MLPs learning posterior probabilities of sub-word units are excellent feature extractors. The ideal feature for ASR is one in which variabilities such as speaker differences are suppressed, while variabilities in phonetic units are enhanced. Phonetic posterior probabilities possess these qualities. In [134], Zhu et al. calculated the variance of speaker adaptive transform (SAT) matrices across all speakers in a CTS test set for standard PLP features as well as for MLP-based features. The variance of each component in the SAT matrix is directly proportional to the amount of speaker variability present in the corresponding feature component. The components in the PLP features showed much higher speaker variability than the components in the MLP-based features.

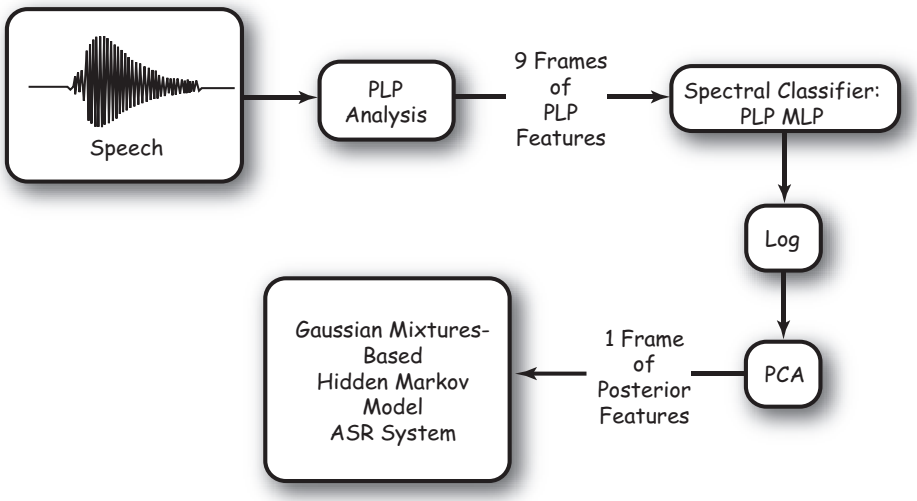

Figure 4.2: Block diagram of the Tandem ASR system. It uses transformed posterior probabilities estimated by an MLP as data-derived front-end features for a standard Gaussian mixtures-based HMM system.

Another benefit of the Tandem setup is how it is readily amenable to classifier combination. Multiple MLPs can be trained to extract discriminant speech information in vastly different ways and then combined to give much better estimates of phonetic posteriors. In Chapter 3, we found that combining a standard spectral MLP classifier with each of the temporal MLP classifiers (Neural TRAP, HAT, and TMLP) gave significant performance improvements. Using simple combination techniques within the Tandem setup is straightforward and can lead to significant reductions in word error rates. Figure 4.3 shows the combination of a spectral MLP classifier and a temporal MLP classifier in the Tandem setup.

One weakness of the Tandem setup which was observed when researchers at the International Computer Science Institute tried to use the Tandem setup for recognizing digits in noisy test conditions was that the feature extracting MLPs trained in clean conditions did not always estimate the phone posteriors very well in noisy test conditions. As with many discriminative training techniques, the resulting classifiers can be susceptible to mismatch between training and testing conditions. To alleviate some of the effects of mismatch, Stephane Dupont proposed to use the MLP-based features to augment the existing conventional features instead of replacing them [12]. When the MLP-based features

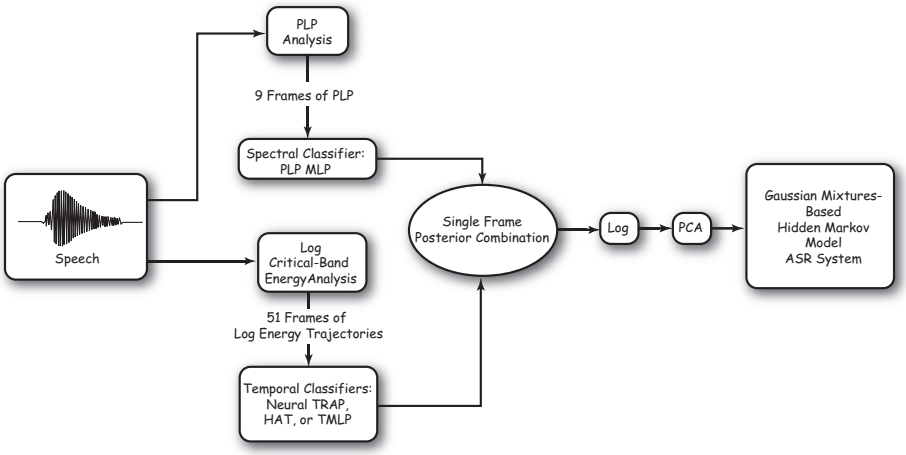

Figure 4.3: Block diagram of a multi-stream Tandem ASR system. Two MLPs extracting discriminant speech information in different yet complementary ways are used to derive posterior probability-based front-end features. The outputs of these MLPs are combined, transformed, and then used as front-end features for a standard Gaussian mixtures-based HMM system.

give poor estimates of phone posteriors, the original PLP features might help the HMM back-end to still come up with the correct classification. Figure 4.4 shows the augmentation of standard PLP features with MLP-based features coming from a combination of two different MLPs.

By combining MLP classifiers that extract information differently than conventional features, the resulting augmented Tandem ASR system can capture speech information from three (or more when combining more than two MLP classifiers) different snapshots. The first snapshot comes from the conventional features which allows the recognizer to capture information from narrow spectral slices. The second and third snapshots come from the different MLP approaches. We will test the effectiveness of the combination of an MLP recieving 9 frames of conventional PLP features with Neural TRAP in the following sections. The 9 frame PLP/MLP can capture speech information from intermediate width spectral slices (100 ms), while Neural TRAP extracts information from long-term narrow-frequency log energy trajectories.

This Tandem augmentation approach proved to be very effective in reducing word error rates on small digit recognition tasks [12]; however, success in small recognition tasks does not necessarily scale to more difficult tasks where the vocabulary is much larger and

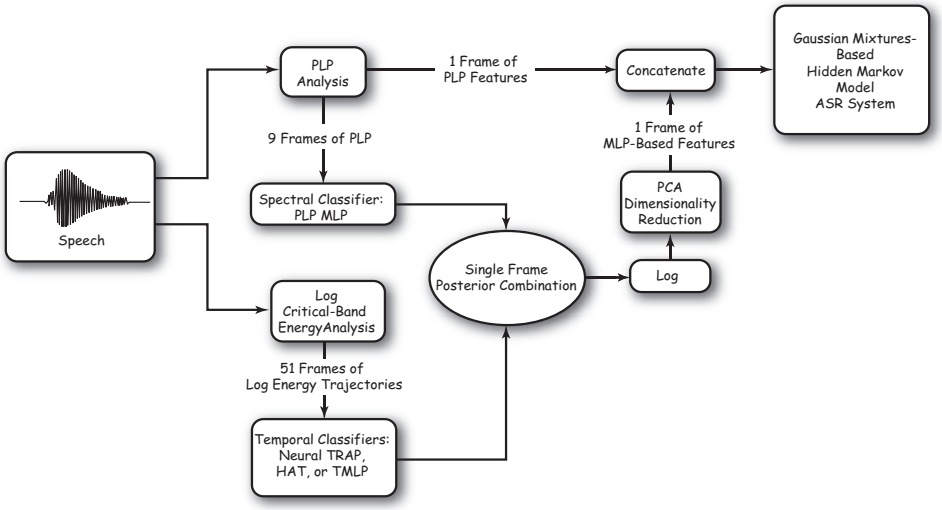

Figure 4.4: Block diagram of a multi-stream augmented Tandem ASR system. Two MLPs extracting discriminant speech information in different yet complementary ways are used to derive posterior probability-based front-end features. The outputs of these MLPs are combined, transformed, dimensionality reduced, and then concatenated to conventional front-end features. The resulting augmented front-end feature is input to a standard Gaussian mixtures-based HMM system.

the speaker variabilities are much greater and the systems used tend to model much more detail and use more elaborate techniques that can be incorporated given large amounts of training data. Our goal in the rest of this chapter is to test the effectiveness of this Tandem augmentation approach on a series of more difficult speech recognition tasks as well as to determine a good operating configuration for the setup. When using this Tandem augmentation approach there are some system issues that must be addressed for optimal performance. The first one is what type of combination technique should be used for the MLP classifiers. The second one is to determine the best number of MLP-based feature dimensions to keep after PCA. Keeping too many features may require much more training data and parameters for the GMHMM, while too few may mean a loss in useful information for recognition.

4.2 Combination Techniques and Dimensionality Reduction

We are interested in testing out several simple frame-wise posterior combination techniques that have performed comparably to more complicated combination techniques [65, 94]. All of these frame-wise posterior combination techniques can be represented by a weighted sum of posteriors or log posteriors. Generally, the combined posterior probability of the kth phone, q_k, given the features $\mathbf{X}$ can be written as Equation 4.1:

$$P(q_k|\mathbf{X}) = \omega_1 P(q_k|\mathbf{X_1}) + \omega_2 P(q_k|\mathbf{X_2}) \tag{4.1}$$

where $P(q_k|\mathbf{X_1})$ and $P(q_k|\mathbf{X_2})$ are the posterior probabilities (or log posterior probabilities) of the phone class q_k given evidence from stream 1 ($\mathbf{X_1}$) and stream 2 ($\mathbf{X_2}$) respectively for a single frame of speech. ω_1 and ω_2 are the stream weights which depend on the combination technique used.

We have tested three frame-wise posterior combination methods: the average of the posteriors combination (AVG); the average of log posteriors combination (AVGLog), and finally, the inverse entropy weighted combination (INVENT) [94]. The first two combination methods essentially assume that each MLP feature stream is equally important, while the entropy-based combination assumes that the MLP feature with lower entropy is more important than an MLP feature with high entropy. This is intuitively correct, since a low entropy posterior distribution (such as would occur with a high single peak) implies strong confidence in class identity.

For both the average combination and the average of the log combinations , ω_1=ω_2=0.5, but in the average of the log combinations, we first apply log to the posteriors before the weighted sum in Equation 4.1. In the inverse entropy-based posterior combination, ω_i is the inverse entropy computed over one frame for the MLP output from stream i. Then all of the 'ω's are normalized so that they sum to one. A threshold of 1.0 is applied for all entropy calculations. If the entropy for a frame from an MLP is greater than 1.0, it is set to a large value (e.g., 10,000) so that the weight is a very small number. Note that the inverse entropy combination technique dynamically weights each stream. The calculated entropies change from frame to frame, but in both average combinations the weights remain fixed at 0.5.

The other main issue for the augmented Tandem setup is the optimal dimensionality of the MLP-based features. Our neural nets are trained to learn posteriors of

46 monophones, so without truncating the number of features after PCA the total augmented feature vector will have a size of 85 (39 original PLP features + 46 posterior-based features = 85 augmented features). Increasing the number of features can potentially increase separability of the classes, but adding too many features may lead to the curse of dimensionality: the number of training examples and parameters in the model required for high performance grows exponentially with respect to the number of feature dimensions. Keeping all 46 posterior-based features may also not be necessary because some features contain more information than others.

Another technical detail that we encountered when implementing our augmented Tandem setup is the effect of a tuning parameter called the Gaussian weight[1]. In the SRI recognition system, this weight controls the relative contribution of each of the Gaussian components in the Gaussian mixture model to the overall likelihood score. The likelihood of a particular frame of features $\mathbf{X}$ is given by Equation 4.2.

$$P(\mathbf{X}|q) = \sum m_i P_i(\mathbf{X}|q)^\gamma \tag{4.2}$$

where m_i is the ith mixture weight, $P_i(\mathbf{X}|q)$ is the ith Gaussian, and γ is the Gaussian weight parameter. There are other tuning parameters like the Gaussian weight that are important in practice for good recognition performance. Some of these include the language model weight and word transition weight which balance the relative influence of the language model scores and word transitions respectively on the scores of possible sentence hypotheses. We investigate the effect of tuning the Gaussian weight, while varying the number of dimensions of the MLP-based features on recognition performance.

4.3 Experimental Setup

In all of the experiments we perform in this chapter, our baseline feature vector consists of 12th order PLP coefficients plus energy computed over a 25 ms frame window every 10 ms. 1st and 2nd order deltas are calculated and appended together to yield a 39 dimensional baseline feature. We also normalize the PLP features by subtracting the mean and dividing by the standard deviation calculated over an entire conversation side.

[1] This is a tuning parameter that is found in the SRI recognition system and may not exist in other large vocabulary recognition systems.

For contrast, we augment the baseline PLP features with a combination of two probability-based feature streams: PLP/MLP features and Neural TRAP features. For the PLP/MLP stream, we train an MLP using 9 consecutive frames of the baseline PLP features as inputs and 46 phone targets generated from forced alignments using SRI International's state-of-the-art Gaussian mixtures-based HMM ASR system. For the Neural TRAP stream, the first stage MLPs take PCA transformed log critical-band energy trajectories formed by taking 51 consecutive frames of log critical-band energies every 10ms. These critical-band MLPs are trained with the same phone targets as used for training the PLP/MLP stream. A merger MLP (trained with these same phoneme targets) combines the critical-band MLPs' outputs to give one estimate of phone posteriors every 10 ms.

We combine the outputs of the Neural TRAP classifier and the PLP/MLP using one of the frame-wise posterior probability combination techniques described above. After combination, we take the log of the posterior vector to reduce its skew (in practice this makes the posterior vector more Gaussian), and then orthogonalize and reduce the dimensionality of the posterior vector using PCA. The resulting variables are then appended to the original PLP cepstra to form the augmented feature vector. It is important to note that this combined-augmented feature integrates information about speech from three different time scales. The original PLP features capture short-term information (about 25 milliseconds), the PLP/MLP stream captures intermediate-term information (approximately 100 milliseconds), and the Neural TRAP stream captures long-term information (around 500 milliseconds). Refer to Figure 4.4 for a block diagram of this process.

In what follows, we refer to these augmented features as PLP+*combomethod*(*Streams*) features, where *combomethod* can be one of the three frame-wise posterior combination methods: the average of the posteriors combination (AVG); the average of log posteriors combination (AVGLog), and finally, the inverse entropy weighted combination (INVENT). *Streams* refers to the PLP/MLP feature stream and the Neural TRAP feature stream. These features serve as the front-end features for our recognition experiments. We use a stripped-down version of SRI's state-of-the-art Hub-5 conversational speech transcription system for our HMM back-end. In particular, the back-end that we used was similar to the first pass of the system described in [122], using a bigram language model and within-word triphone acoustic models. For fairness of comparison, all HMMs have roughly the same number of trainable parameters. The

HMMs also share the same training set with all of the neural net systems.

4.4 Stage 1: The Numbers Task

As noted previously, all the basic techniques employed here were originally developed using quite small tasks. In particular, prior to the experiments reported here, the MLP-based feature transformations, the temporal features (Neural TRAP), and the methods used to combine and use them within the augmented Tandem approach were all trained and tested on a number of smaller tasks including the OGI Numbers task [20] (the Numbers95 corpus). In these earlier Numbers experiments, Numbers data was used for both training and testing. As explained earlier augmenting the baseline features with a combination of PLP/MLP and Neural TRAP-based features improves ASR performance. Whether this result scales to larger tasks is an open question.

In the remaining sections of this chapter, we want to apply this augmented Tandem approach to a series of larger and more difficult ASR tasks. Our final goal is to create an augmented Tandem system for the difficult task of conversational telephone speech (CTS). Simply taking the features and applying them to the CTS task risked failure without obvious diagnostic potential. Consequently, we designed a three-stage approach to the development process. Our initial step was to train on a combination of CTS data and read speech, and then test on OGI Numbers.

4.4.1 The Numbers Task Description

The training set for this stage was an 18.7 hour subset of the old "short" SRI Hub5 training set for CTS. 48% of the training data was male and 52% female. 4.4 hours of this training set comes from English CallHome [19], 2.7 hours from Hand Transcribed Switchboard [45], 2.0 hours from Switchboard Credit Card Corpus [42], and 9.6 hours from Macrophone [13] (read speech which contains many examples of continuous numbers). All of these training sources are large vocabulary corpora (consisting of more than 25,000 different words). In contrast, the OGI Numbers corpus which we use as the test data consists of only 32 words.

We divided the entire OGI Numbers corpus into three sets. One was used for

system parameter tuning, one for development testing, and another for final testing. We used the official dev set (0.6 hours) of the Numbers95 corpus to tune the language model weight and word transition weight. We report our results on the final test set which contains 1.3 hours of speech (2,519 utterances and 9,699 word tokens).

After training MLPs for posterior estimation, we calculated the classification accuracy on the development set. For PLP/MLP, this accuracy was 67% computed over 415,985 frames, and for Neural TRAP it was 68%. Combining the two using inverse entropy weighting or simply averaging the posterior gave roughly the same frame classification accuracy of about 70.9%. Thus the two MLPs can be simply combined to significantly improve frame accuracy, which suggests that they provide information that is complementary.

4.4.2 Results on the Numbers Task

Using the training set defined above, we trained triphone gender-independent HMMs using the SRI speech recognition system. Although the recognition task was numbers, the HMMs were trained for broader vocabulary and speaker coverage. Thus we hoped that the conclusions reached with this training data might generalize better to other tasks. The testing dictionary contained thirty words for numbers and two words for hesitation, and we used a simple bigram language model trained on our Numbers tuning set.

System	Numbers Test Set WER	Relative Reduction WER
PLP Baseline	4.0%	-
PLP+AVG(*Streams*)	3.3%	17.5%
PLP+AVGLog(*Streams*)	3.2%	20.0%
PLP+INVENT(*Streams*)	3.3%	17.5%

Table 4.1: Word error rate (WER) and relative reduction of WER on Numbers using different combination approaches. *Streams* denotes the PLP/MLP feature stream and the Neural TRAP feature stream.

We incorporated PLP/MLP and Neural TRAP features by frame-wise posterior combination. The combined features were then reduced in dimension to 17 using PCA and concatenated to the baseline PLP features to create an augmented feature vector of dimension 56. As noted previously, we used several frame-wise poste-

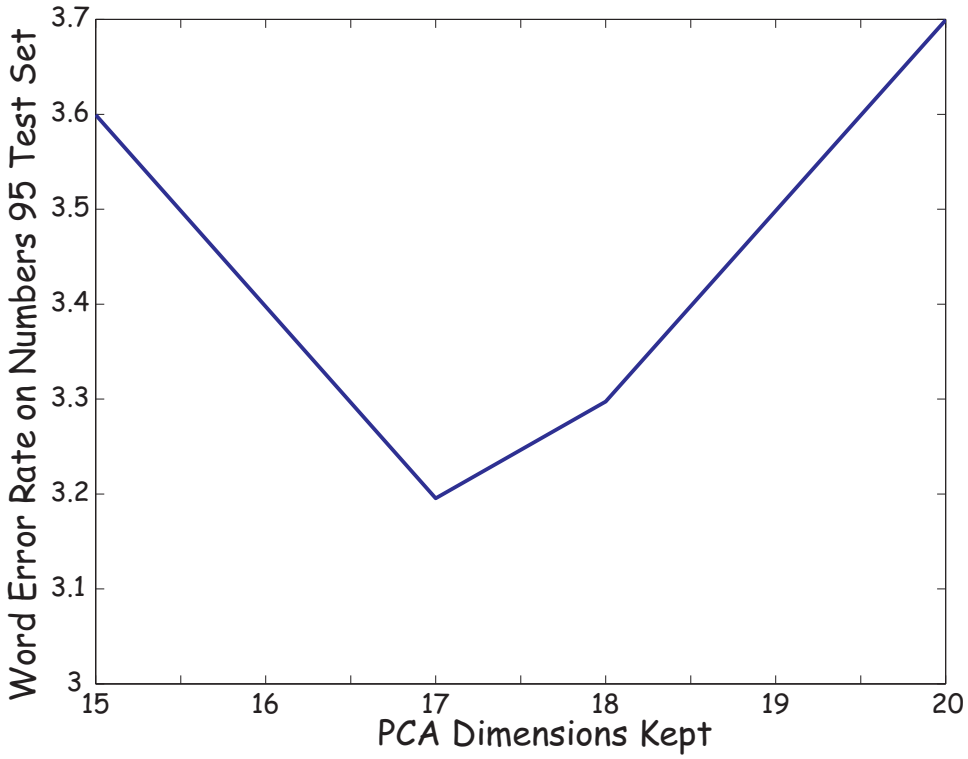

Figure 4.5: Word error rate on the Numbers 95 test set as a function of the number of PCA dimensions kept in the PLP+INVENT(*Streams*) system without tuning the Gaussian weight.

rior combination methods: the average of posteriors PLP+AVG(*Streams*), the average of log posteriors PLP+AVGLog(*Streams*), and the inverse entropy weighted combination PLP+INVENT(*Streams*) (see Table 4.1). All three performed roughly the same, achieving a 17.5-20% relative reduction in word error rate.

Note, before we tried tuning the Gaussian weight, truncation of the PCA output (that is, eliminating some low-variance components) was critical to performance. Keeping the top 17 dimensions was the optimal length on all of our tuning data without changing the Gaussian weight. Figure 4.5 shows the effect of the PCA dimensions kept on word error rate on the Numbers95 test set without changing the Gaussian weight. The performance curve is from the augmented Tandem system using inverse entropy combination (PLP+INVENT(*Streams*)). Notice that the WER is quite sensitive to the number of dimensions. Just changing the number of dimensions by two can cause degradations of .4% absolute.

These experiments showed that the combination of the three features (baseline PLP, PLP/MLP, and Neural TRAP) can improve the recognition performance over using the baseline PLP features alone. On the other hand, all the approaches to posterior combination were roughly equivalent in this case. These preliminary conclusions would later be tested on tasks of increasing complexity.

4.5 Stage 2: The Top-500 Word CTS Task

Our methods continued to work well on the small vocabulary continuous numbers task even when we did not train explicitly only on continuous numbers. Before applying our approaches to the full vocabulary Switchboard task, we considered a second stage task, that of recognizing the 500 most common words[2] in Switchboard I [41]. There were several advantages to using this intermediate task. First, since the recognition vocabulary consisted of common words from Switchboard, it was likely that error rate reduction would apply to the larger task as well. Second, there were many examples of these 500 words in the training data, so less training data was required than would be needed for the full task. This in turn reduced training time accordingly. Lastly, recognition complexity in this task was smaller, which also reduced experimental turn-around time.

4.5.1 Top-500 Words Task Description

For training, we created a subset of the "short" training set used at SRI for CTS system development, which we referred to as the Random Utterances of Short Hub or the RUSH set. This RUSH set consists of utterances from 217 female and 205 male speakers, which was the same number of speakers as the short CTS training set, but contains one third of the total number of utterances. The female speech consists of 0.92 hours from English CallHome, 10.63 hours from Switchboard I [41] with transcriptions from Mississippi State [28], and 0.69 hours from the Switchboard Cellular Database [43]. The male speech consists of 0.19 hours from English CallHome, 10.08 hours from Switchboard I, 0.59 hours from Switchboard Cellular, and 0.06 hours from the Switchboard Credit Card Corpus.

The top-500 word test set was a subset of the 2001 Hub-5 evaluation data

[2]This task was proposed by our colleague George Doddington.

(Eval2001). Given the 500 most common words in Switchboard I, we chose utterances[3] from the Eval2001 data in which 90% or more of the words in the utterance were on the word list. In other words, we allowed at most 10% of the words in an utterance to be out of vocabulary (OOV) words. 49.6% of the utterances in the Eval2001 data met this requirement, and the total OOV rate was 3.2%. We partitioned this set into a tuning set (0.97 hours, 8242 total word tokens) and a test set (1.42 hours, 11845 total word tokens). We used the tuning set to tune system parameters like word transition weight and language model weight, and we determined word error rates on the test set. The language model used in both the 500 word task as well as the full vocabulary task was the first-pass bigram language model used by SRI for the large vocabulary evaluations in 2000.

4.5.2 Results on Top-500 Words Task

Using the baseline PLP features, we trained gender dependent triphone HMMs on the 23 hour RUSH training set, and then tested this system on the 500 word test set achieving a 43.8% word error rate (see Table 4.2, which shows the word error rates of our various systems on the top-500 word test set). As seen in the table, the word error rate was reduced about 10% relative by augmenting the baseline features with the combined PLP/MLP and Neural TRAP features. In this case, we trained gender dependent PLP/MLP feature nets and Neural TRAP systems.

System	500 Word Test Set WER	Relative Reduction WER
PLP Baseline	43.8%	-
PLP+AVG(*Streams*)	39.4%	10.0%
PLP+AVGLog(*Streams*)	39.5%	9.8%
PLP+INVENT(*Streams*)	39.2%	10.5%

Table 4.2: Word error rate (WER) and relative reduction of WER on the top-500 word test set of systems trained on the RUSH set using different combination approaches. *Streams* denotes the PLP/MLP feature stream and the Neural TRAP feature stream.

All three combination methods performed roughly the same. Even though the more complicated inverse entropy combination technique performed only slightly better

[3] An utterance is defined to be a string of words separated by less than 0.3 seconds, and greater than 0.3 seconds of separation at the beginning and end.

than the simple average combination methods, both styles have their appeal. The averaging methods are certainly simple and don't rely on any estimation method. On the other hand, the inverse entropy combination technique is potentially robust to poor classifier streams. We experienced this property for one of our later (CTS) experiments. Due to a bug in our procedures, we unintentionally combined a badly degraded Neural TRAP stream with the other features using both methods. When probabilities were combined using the AVG and AVGLog methods, the degraded stream hurt performance badly. On the other hand, the inverse entropy-weighting reduced the importance of the poor stream so that the overall performance essentially matched what we had for a feature that consisted of the baseline PLP features concatenated with the PLP/MLP feature alone. Thus, the entropy-based approach to combination appears to be more robust to unexpectedly poor streams. We expect that this property might be particularly useful for future efforts in which we might combine a larger number of streams where some streams may sometimes provide less useful information.

As in the numbers task stage, we plot the WER curve showing the effect of the number of dimensions after PCA for the PLP+INVENT(*Streams*) system in Figure 4.6. Without tuning the Gaussian weight, we again see that the best choice of number of dimensions is still at 17, and the WER is quite sensitive to this choice (especially to overestimates of the dimension). When changing the number of dimensions kept from 17 to 19 the WER jumps from 39.6% to 40.4% on the top-500 word tuning set.

4.6 Stage 3: Full CTS Vocabulary

Having seen how our approaches scaled with increasing test set complexity, we applied these approaches to the third and last stage: full vocabulary CTS task.

4.6.1 The Full CTS Task Description

We tried using our previously defined RUSH training set for this task and found it inadequate for training given the increase in vocabulary. Error rates on Switchboard test sets were unacceptably high for the RUSH training set. Instead, we used SRI's entire "Short" CTS training set from which RUSH was derived. This set contained a total of 68.95 hours of CTS. 2.75 hours of English CallHome, 31.30 hours from Mississippi State

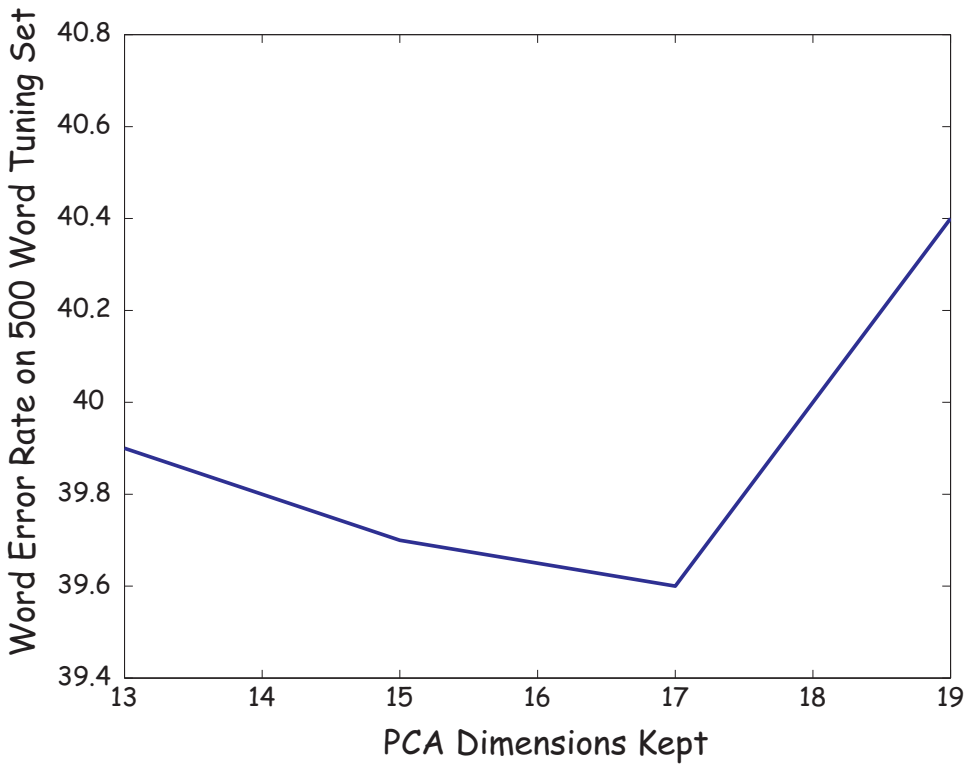

Figure 4.6: Word error rate on the top-500 word tuning set as a function of the number of PCA dimensions kept in the PLP+INVENT(*Streams*) system without tuning the Gaussian weight.

transcribed Switchboard I, and 2.03 hours of Switchboard Cellular form the data from female speakers. The male speaker data came from 0.56 hours of English CallHome, 30.28 hours from Switchboard I, 1.83 hours from Switchboard Cellular, and 0.20 hours of Switchboard Credit Card Corpus. As in the 500 word task, we trained triphone gender dependent HMMs as well as gender dependent PLP/MLP and Neural TRAP systems.

For testing, we used the 2001 Hub-5 Switchboard evaluation set (Eval2001) from which our top-500 word test set was derived. This evaluation set contains a total of 6.33 hours of speech, 62,890 total word tokens. For tuning our system parameters, we used a subset of the 2001 Hub-5 development set.

4.6.2 Results on the Full CTS Task

The baseline system achieved a 43.8% word error rate on the Eval2001 set (see Table 4.3, which shows the word error rates of our various systems on the Eval2001 set). The augmented features reduced the error rate by about 7% relative. For this task, there was a small penalty for the AVGLog combination method in comparison to the other approaches.

System	Hub-5 EVAL2001 WER	Relative Reduction WER
PLP Baseline	43.8%	-
PLP+AVG(*Streams*)	40.5%	7.5%
PLP+AVGLog(*Streams*)	41.0%	6.4%
PLP+INVENT(*Streams*)	40.6%	7.3%

Table 4.3: Word error rate (WER) and relative reduction of WER on the 2001 Hub-5 evaluation set of systems trained on SRI's "Short" CTS training set using different combination approaches. *Streams* denotes the PLP/MLP feature stream and the Neural TRAP feature stream.

4.7 Dimensionality Tuning

In the previous sections we showed how all the various frame-wise posterior combination techniques yielded similar results. Now, we want to further investigate the effect

Description	Dimensions Retained				
	15	17	19	25	35
RUSH Training Top-500 Test WER (%)	38.4	38.5	39.0	39.2	39.4
NSH5 Training Eval 2001 WER (%)	39.7	39.6	39.4	39.1	39.3

Table 4.4: The effect on word error rates from the PLP+INVENT(*Streams*) features while varying the number of dimensions retained after PCA and tuning the Gaussian weight.

on performance when modifying both the number of dimensions kept after PCA and the value of the Gaussian weight in the SRI recognition system. In previous experiments, we found that performance was optimal when keeping only the top 17 dimensions after PCA and that small changes in the dimensionality led to large changes in performance. In the experiments below, we find that this effect can be lessened by tuning the Gaussian weight. To tune this Gaussian weight parameter, we simply set the Gaussian weight to various values and ran the recognizer on our tuning data. Then we picked the Gaussian weight value that gave the smallest WER and used this value for recognition of the test set. Table 4.4 shows the effects on WER when tuning both the number of dimensions after PCA and the Gaussian weight. The features used are the baseline 39-dimensional PLP features augmented with the inverse entropy combination of PLP/MLP and Neural TRAP.

The differences in WER for different dimensions range from 1.0% absolute in the top-500 word test to 0.6% absolute on the Eval 2001 test set. These differences are statistically significant which means that the number of dimensions kept after PCA is still vital for good performance; however, the absolute differences in WER when the number of dimensions is close to the minimum is quite small and statistically insignificant. For the top-500 word test the minimum WER is achieved with 15 dimensions, while in the case of Eval 2001 the best number of dimensions is 25. Compare, for example, the WER for the top-500 word test at 15 and 17 dimensions. These only differ by .1% absolute (38.4% vs. 38.5%). Also compare the WER on Eval 2001 at 25, 19, and 35 dimensions. The absolute differences are only .2%-.3% which is quite small considering the large jump in number of dimensions. When tuning the Gaussian weight and the number of dimensions concurrently, performance still depends to a large degree on the number of dimensions kept, but once the number of dimensions is near the optimal, the WER differences are not significant.

4.8 Conclusion

We applied the PLP/MLP and the Neural TRAP features, developed for a very small task, to a series of successively larger problems. We found that:

1. Word error rate was significantly reduced for the small tasks as well as the larger tasks,

2. The combination methods, which gave equivalent performance for the smaller task, were also comparable on the larger tasks,

3. And tuning the Gaussian weight concurrently with the number of dimensions was an important step to achieve optimal performance.

Regarding the first point, the approach of using a combination of PLP/MLP and Neural TRAP features to augment the baseline PLP features consistently improves ASR performance on a variety of training/testing sets. An absolute error rate reduction of over 3% on Switchboard is quite significant. However, the typical relative reduction in error is somewhat smaller for the larger tasks (ranging from 20% on the Numbers task to 7% on the full CTS task). Thus, having statistically significant error rate reduction may scale, but the degree of improvement may not without further work using the CTS task. Nonetheless, even a 7% relative improvement is often of significant interest for larger tasks like CTS. For such tasks, sizable improvements are typically only obtained by a combination of many small innovations.

The second observation seems to be unequivocally confirmed in these three stages of experiments - we observed no consistent (scalable) advantage to using any of the three chosen methods for combining posteriors as part of the process of generating probability-based front-end features. On the other hand, as noted earlier, the inverse entropy method appears to be quite robust to catastrophic degradations of feature streams. We also should emphasize the limitation of this experiment, in which we were only combining two streams, both of which were fairly effective for phonetic discrimination. If we begin to use a significantly larger number of streams, some streams will be more likely to be ineffective at least some of the time, and a dynamic weighting method like the inverse entropy approach may show a clearer advantage. This view seems to be supported by earlier work at IDIAP [94].

The third observation is a practical matter of tweaking the system to achieve the best possible results. While we cannot make any generalization about the exact number of dimensions to keep after PCA for any other speech recognition task, we can say that the number of dimensions to keep should be tuned. Furthermore, tuning the Gaussian weight in conjunction with the dimensionality can lessen the importance of getting the exact optimal dimensionality.

Finally, we have achieved our goal of setting up a competitive baseline recognizer for CTS. The word error rates reported in this chapter are around 40% on Eval 2001 which is similar to the performance of a typical state-of-the-art recognizer performing only a first-pass decode (i.e., a simple Viterbi decode using a bigram language model without later adaptation, 4-gram language model, system combination, etc.) on similar CTS test data [35].

Chapter 5

Comparison of Temporal Systems for CTS

In Chapter 2 we introduced several new temporal systems based on the Neural TRAP idea: Hidden Activation TRAP (HAT) and Tonotopic Multi-Layer Perceptron (TMLP). Each of these temporal systems learn discriminant phonetic information within long-spanning narrow-frequency channels. We developed ASR system configurations that utilize Neural TRAP for improving performance on conversational telephone speech (CTS) in Chapter 4. Now we are poised to undertake a comparative study between various approaches incorporating information from long time spans (about 500 milliseconds) using the ASR system configurations introduced in Chapter 4 for the improvement of performance on CTS. Specifically, we are interested in comparing:

1. The narrow-band constraint of learning long-term information versus unconstrained versions,

2. The nonlinear approach to learning phonetically discriminant critical-band information versus various linear approaches,

3. And various nonlinear MLP-based approaches with each other.

We also corroborate one of the key findings about temporal systems in the previous chapters as well as in previous work: temporal systems offer complementary phonetic information in combination with conventional systems that extract phonetic information from shorter

time spans over the entire spectrum. We find that the combination of a conventional front-end feature (spanning approximately 25 milliseconds), a conventional MLP-based feature (spanning about 100 milliseconds), and a temporal system-based feature (spanning around 500 milliseconds) achieves impressive performance improvements on CTS.

5.1 Various Temporal Systems

In this section we describe all of the different approaches to learning long-term speech information for phonetic classification. Because these approaches extract information in time, we refer to these approaches as temporal systems. Typical ASR front-end features extract information from short-term spectral slices of about 25 milliseconds, while traditional hybrid ANN/HMMs model medium-term spectral chunks spanning about 100 milliseconds by learning transformations over 9 consecutive frames of features. All the temporal systems below extract information from long-term speech energies spanning approximately 500 milliseconds. Each of the temporal systems can be grouped into one of three categories based on whether there is a narrow-frequency band constraint and whether the initial transformation on the spectral energies is linear or nonlinear.

The starting point for all of these temporal systems is the log critical-band energy spectrum of speech. Every 10 milliseconds in the speech signal, we apply a centered 25-millisecond Hamming window and then calculate the squared magnitude of a 256-point FFT. 15 Critical-band energies are calculated from these squared magnitudes by averaging adjacent magnitudes within each of the 15 critical-band filters. We then apply the log and normalize by subtracting the mean and dividing by the standard deviation calculated over all frames[1] within a single utterance. See Figure 1.2 in Chapter 1 for an illustration of this process.

5.1.1 Unconstrained Approaches

In the unconstrained approaches, we allow the MLP classifiers to learn any information contained within the 15 critical-bands x 51 frames of log critical-band energy input. Essentially, we simply feed 51 consecutive frames (about 500 milliseconds) of log critical-

[1] A frame corresponds to the speech measurements calculated every 10 milliseconds.

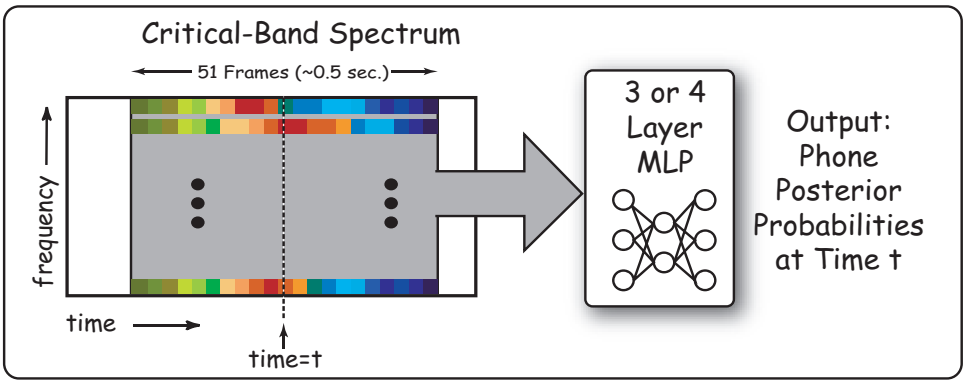

Figure 5.1: Architecture for unconstrained approach.

band energies from all 15 critical-bands to the MLP classifier and let it learn what it needs to estimate the phone posteriors. We have experimented with two different fully-connected MLPs: a 3-layer MLP consisting of a single hidden layer, and a 4-layer MLP consisting of two hidden layers. Figure 5.1 illustrates the unconstrained approach for building a temporal system.

It is important to note that these unconstrained temporal systems can learn any kind of relationship among all of the 15x51=765 log energy values. For example, these unconstrained temporal systems can directly model events such as high energy at low frequencies 20 frames before the current frame concurrently with high energy at high frequencies 23 frames after the current frame. The main difference between the 3-layer and 4-layer MLP is an extra hidden layer in the 4-layer MLP which may simplify the job of learning phone posteriors by breaking the intermediate modeling into two stages. The number of first and second layer hidden units in the 4-layer MLP was determined by optimizing the frame classification accuracy under the constraint of keeping the total number of weights and biases the same as the total for the 3-layer MLP (516,000 weights and biases). The 3-layer MLP has 765 input units, 636 hidden units, and 46 output units, while the 4-layer MLP has 765 input units, 318 first hidden layer units, 750 second hidden layer units, and 46 output units. In what follows we refer to the 3-layer MLP system by "*15 x 51 MLP3*" and the 4-layer MLP system by "*15 x 51 MLP4*".

All MLPs in this chapter are trained on 46 phone targets derived from forced aligned phone labels provided by SRI's state-of-the-art ASR system [121]. The training procedure proceeds as explained in Chapter 2 where the weights and biases are modified

to reduce an error measurement between the training targets and MLP outputs. After training, the outputs approximate posterior probabilities of the target classes which are phones in our case. For fairness of comparison, all temporal systems have the same number of trainable parameters (516,000 trainable parameters on about 30 hours of speech per gender, corresponding to approximately 12,000,000 frames, for frames-to-parameters ratio of about 23.). Also, for all MLPs, the hidden units have a sigmoid nonlinearity and the output units have a softmax nonlinearity.

5.1.2 Constrained Linear Approaches

In contrast to the unconstrained approaches, the constrained approaches first restrict the classifiers to learn information within critical-band energy trajectories spanning half a second. These constrained architectures are forced to represent temporal structure. We investigate several architectures that partition the learning into two constrained stages. The first stage learns what is important for phonetic classification given individual critical-band energy trajectories of 51 frames (about 500 milliseconds), and the second stage combines what was learned at each critical-band to learn overall phone posteriors. This "divide and conquer" approach to learning splits the task into two smaller and possibly simpler sub-learning tasks.

In this subsection we describe linear approaches to learning narrow-frequency temporal information. The first of these two-stage architectures calculates principal component analysis (PCA) transforms on successive 51-frame log critical-band energy trajectories for each of the 15 bands. We use the resulting transform matrices to orthogonalize the temporal trajectory in each band, retaining only the top 40 dimensions per critical-band. PCA projects the original 51 dimensional energy trajectory in directions of maximal variance. Figure 5.2 shows how we then use these transformed (and dimensionally reduced) critical-band "features" as input to an MLP that estimates phone posteriors. This merger MLP has 750 hidden units.

In a related approach, we replace PCA with linear discriminant analysis (LDA) "trained" on the same phone targets used for MLP training. This transform projects the log critical-band energy trajectories of a single band onto vectors that maximize the between-class variance and minimize the within-class variance for phone classes. We also keep the

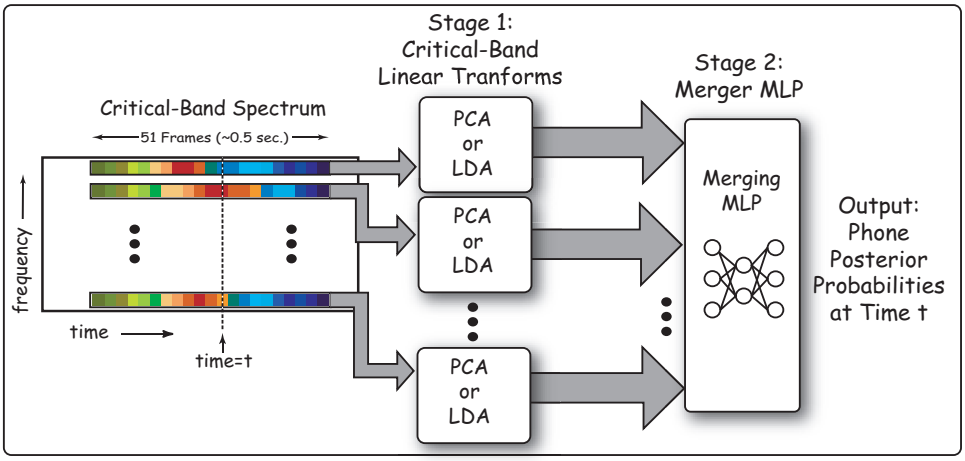

Figure 5.2: Architecture for constrained linear approaches.

top 40 dimensions after the LDA projection and send them into a merger MLP with 750 hidden units. We henceforth denote the two two-stage linear approaches as "*PCA40*" and "*LDA40*" respectively.

5.1.3 Constrained Nonlinear Approaches

We also experiment with five constrained nonlinear approaches. The first four of these approaches are based on the Neural TRAP architecture where critical-band MLPs are trained to learn phone probabilities separately on each of the 15 bands of 51-frame log critical-band energy trajectories. Once trained, we use the outputs at different points in these critical-band MLPs as inputs for a second stage merger MLP that combines and transforms this critical-band information into estimates of phone posteriors. The goal of comparing these first four approaches is to discover what form of critical-band information is most suitable for the second stage merger MLP. Are the hidden activations the most suitable, are the critical-band level phone probabilities the best for classification performance, or something else? The fifth constrained nonlinear approach is the Tonotopic Multi-Layer Perceptron (TMLP) which learns all of the discriminant critical-band hidden unit parameters as a result of a single global error back-propagation algorithm.

Figure 5.3 shows the first four nonlinear two-stage architectures. In the first of these architectures, the input to the second stage is the dot product of the log critical-

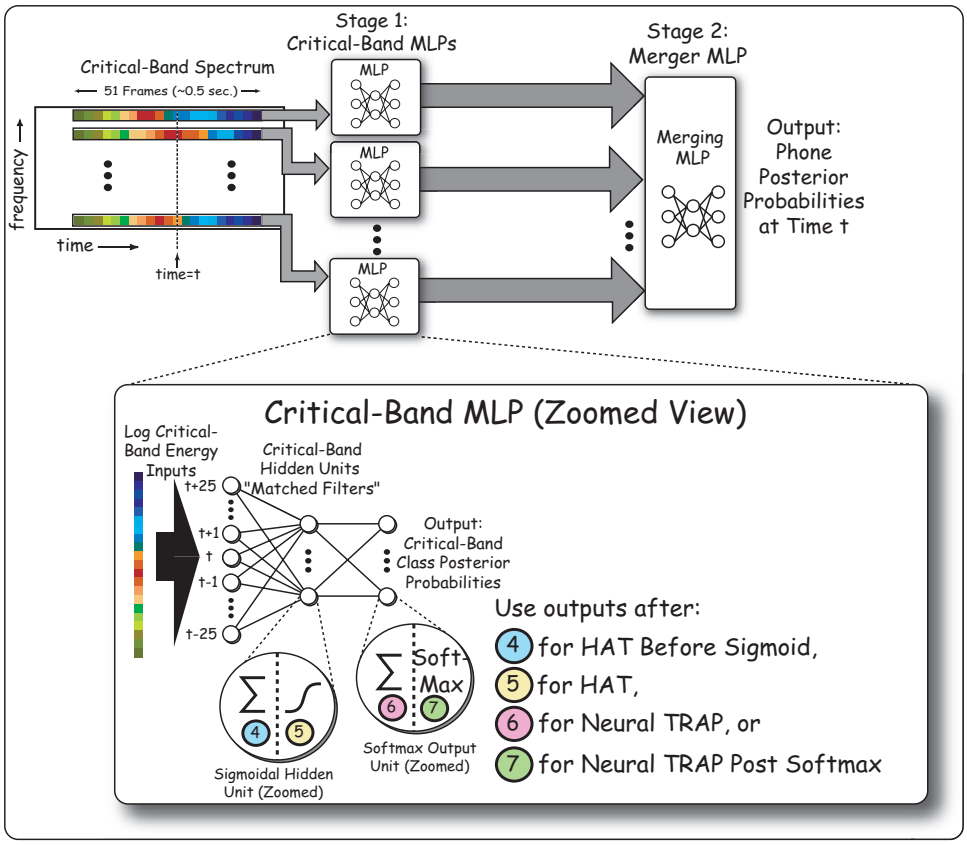

Figure 5.3: Architecture for constrained nonlinear approaches.

band energy inputs with the input to hidden unit weights of the corresponding critical-band MLP. Another way to say this is that the values before the sigmoid in each critical-band hidden unit are used as the inputs to the second stage merger MLP. We refer to this architecture as "*HAT Before Sigmoid*" because it uses the hidden activations before the sigmoid nonlinearity as inputs to the merger. While this first approach consists of a linear matrix multiply, we categorize it in this subsection because the matrix is learned as part of a structure that includes nonlinear sigmoid functions, which have a significant effect on the values learned.

The second approach, Hidden Activation TRAP or "*HAT*", takes the outputs of each hidden unit as the input to the merger MLP. The third approach takes the values after the hidden-to-output weight matrix multiplication, but just before the final softmax nonlinearity of the critical-band MLPs. This approach is equivalent to the Neural TRAP architecture, so it is denoted as "*Neural TRAP*". The fourth approach uses the regular activations from the critical-band MLPs that are phoneme posterior probabilities conditioned on the log critical-band energy inputs. This nonlinear approach is denoted as "*Neural TRAP Post Softmax*".

As discussed in more detail in Chapter 3, the critical-band MLPs trained to approximate critical-band phone posterior probabilities do not achieve high classification accuracy suggesting that phone classification at the critical-band level is very difficult. We developed HAT to show that whatever useful information within the critical-band is already captured in the critical-band hidden unit representations, and that further mapping from this hidden representation to critical-band phone probabilities is unnecessary and leads to poorer overall classification accuracy. The comparisons of these Neural TRAP-based temporal systems corroborate these earlier findings in the context of ASR on CTS.

The last of the nonlinear approaches to learning temporal information is the *TMLP* which is fully described in Chapter 3. Figure 5.4 shows the *TMLP* setup. *TMLP* has the same connections as *HAT* except that the critical-band hidden units are learned via a global error back-propagation algorithm. This allows the *TMLP* to learn a richer class of distributions because the critical-band hidden units are not constrained to minimize classification error of critical-band level phone targets.

The choice of number of critical-band level hidden units as well as the number of merger hidden units for the five systems described above is optimized for *HAT* while

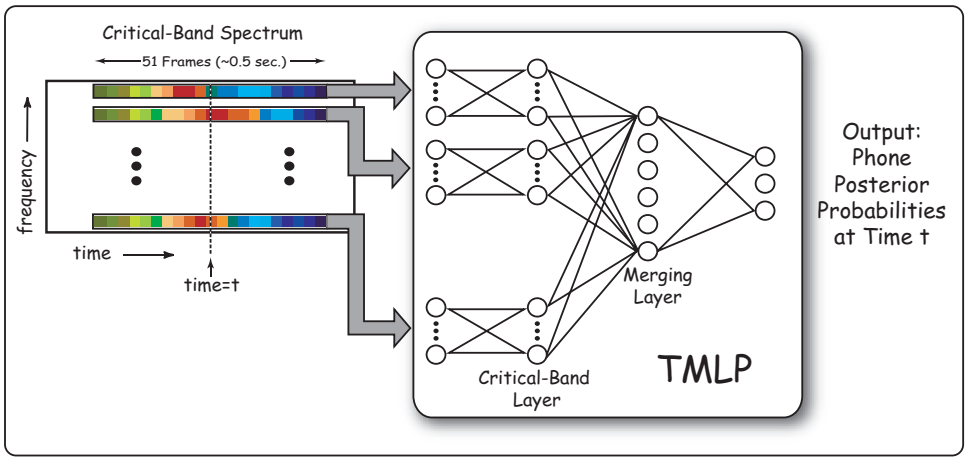

Figure 5.4: Architecture for *TMLP*.

fixing the total number of trainable parameter to about 516,000. Using 40 hidden units per critical-band and 750 hidden units in the merger achieves the best frame accuracy result for *HAT* on the 2001 Hub-5 evaluation data (Eval2001). This choice may not be optimal for *Neural TRAP* systems, but we will explore the effect of having a larger number of critical-band units for *Neural TRAP* in Section 5.4.8.

5.2 Two Conventional Features

The typical choice for front-end features in state-of-the-art ASR systems is either Mel-Frequency Cepstral Coefficients (MFCC) or Perceptual Linear Predictive (PLP) features. Both derive features from very short time spans (about 25 milliseconds). In the Tandem ASR system as described in [49], MLP-based features are derived from 9 consecutive frames of PLP features which span an intermediate time context (about 100 ms). In the experiments that follow, we compare each of the various temporal systems in configurations that augment the conventional short time span features with and without frame-wise combination with the intermediate time MLP-based features.

We use the SRI 2003 evaluation system's conventional front-end feature for the short time span features. These features come from 12th order PLP features plus energy with the first three derivatives. This 52 dimensional feature is transformed and reduced in dimension to 39 via a heteroskedastic linear discriminant analysis (HLDA) transforma-

tion trained on Gaussian state targets. We denote this short-term conventional feature as "*HLDA(PLP+3d)*". Our intermediate time MLP-based features come from a fully-connected 3-layer MLP trained on the same phone targets used to train the temporal systems. This MLP takes 9 frames of 12th order PLP features plus energy with the first two derivatives as input, and we refer to this as "*9 Frame PLP MLP*". The PLP features are mean and variance normalized over an entire conversation side before being used as inputs to the MLP. This MLP has about 516,000 parameters for fair comparisons with the temporal systems.

5.3 ASR System Configurations

5.3.1 Experimental Setup

For all of the experiments reported in this chapter, we show test results on the 2001 Hub-5 evaluation data (Eval2001), a large vocabulary conversational telephone speech test set consisting of a total of 2,255,609 frames (6.27 hours) and 62,890 words. We use the 2001 Hub-5 development data (Devel2001) to tune the language model weight, word transition weight, and the Gaussian weight. We optimize these weights to maximize performance on Devel2001, and then use the optimal values for recognition on Eval2001.

The training set that we use for both MLP and HMM training consists of about 68 hours of conversational telephone speech data from four sources: English CallHome, Switchboard I with transcriptions from Mississippi State, and Switchboard Cellular. This training set corresponds to the one used in [97] without Switchboard Credit Card data. Training for both MLPs and HMMs was done separately for each gender, and the test results presented later reflect the overall performance on both genders. We hold out 10% of the training data as a cross-validation set in MLP training. For fairness in comparison, all MLP-based feature extractors have roughly the same number trainable parameters (about 516,000 on about 30 hours of speech per gender, corresponding to approximately 12,000,000 frames, for a frames-to-parameters ratio of about 23.).

Once the MLPs are trained, we use them to generate various front-end features for the back-end SRI recognizer in a similar manner as was done in [34]. More specifically, we use these MLP-based features in one of three system configurations:

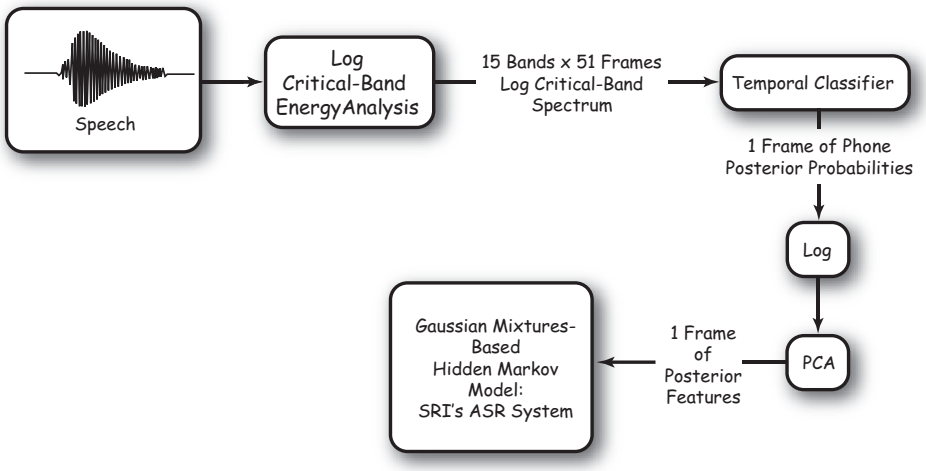

Figure 5.5: In the stand-alone Tandem ASR system configuration, the phone posterior probabilities of an MLP classifier are transformed and used as front-end features for the SRI Gaussian mixtures-based HMM recognizer.

1. stand-alone Tandem features,

2. augmenting standard short-term *HLDA(PLP+3d)* features,

3. and in combination with the intermediate-term *9 Frame PLP MLP* features and augmenting standard short-term *HLDA(PLP+3d)* features.

The back-end SRI recognizer that we use is similar to the first pass of the system described in [122] with a bigram language model and within-word triphone acoustic models.

5.3.2 Stand-Alone Tandem

The first ASR configuration that we use for our comparison tests is the stand-alone Tandem feature setup. This setup allows us to test how well a particular MLP is at extracting useful phonetic information by itself. The MLP's phone posteriors are transformed and used as the front-end feature for the back-end Gaussian mixtures-based HMM recognizer. This stand-alone setup is pictured in 5.5. The box labeled "Temporal Classifier" is one of the various temporal systems described earlier.

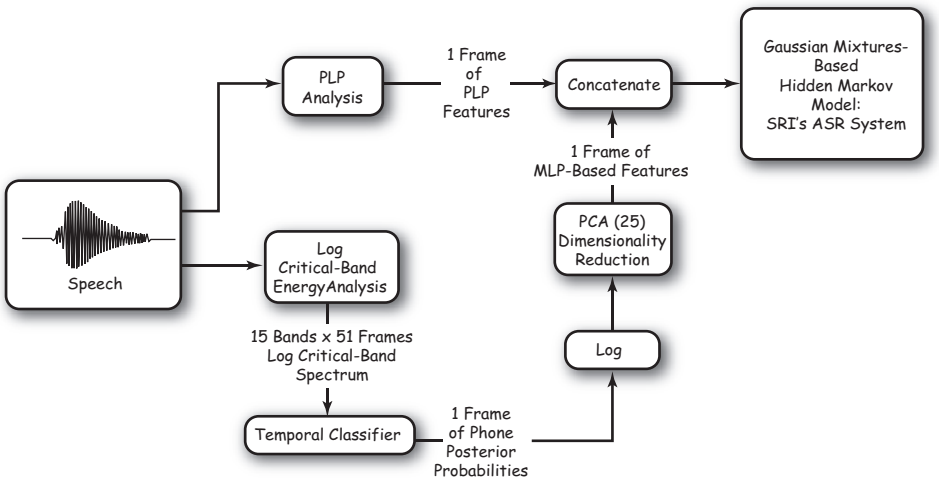

Figure 5.6: In the augmented feature ASR system configuration, the phone posterior probabilities of an MLP classifier are transformed, dimensionality reduced, concatenated with the short-term *HLDA(PLP+3d)* features, and used as front-end features for the SRI Gaussian mixtures-based HMM recognizer.

5.3.3 Augmented Feature

In the augmented feature configuration, the MLP-based feature extractor is used to augment the standard short-term *HLDA(PLP+3d)* features. Specifically, we take the phone posterior outputs, apply the log, and perform PCA. In Chapter 4 we found that keeping the top 25 dimensions from the MLP feature stream gives the best recognition performance. In this chapter, we continue to keep the top 25 dimensions after PCA, and then concatenate these MLP-based features to the standard short-term *HLDA(PLP+3d)* features. The resulting 64 dimensional feature is used as the input features for the SRI Gaussian mixtures-based HMM recognizer. This configuration allows us to see how much improvement can be achieved by augmenting a short-term information stream with a long-term information stream. Figure 5.6 shows a block diagram of this augmented feature configuration.

5.3.4 Combined-Augmented Feature

The combined-augmented feature configuration utilizes all three temporal contexts: short (around 25 milliseconds), intermediate (about 100 milliseconds), and long (approximately 500 ms). First, the intermediate-term *9 Frame PLP MLP* phone poste-

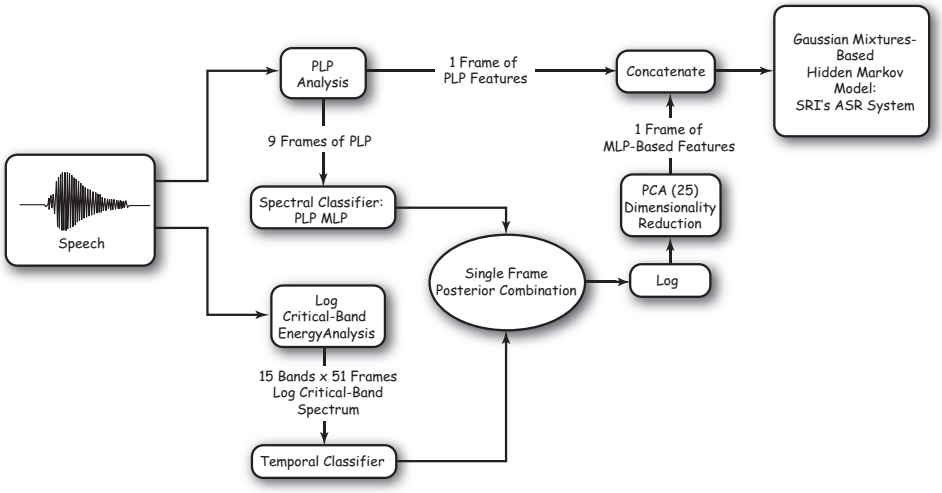

Figure 5.7: In the combined-augmented feature ASR system configuration, the phone posterior probabilities of a long-term MLP classifier are combined with the posteriors of an intermediate-term MLP classifier, transformed, dimensionality reduced, concatenated with the short-term *HLDA(PLP+3d)* features, and used as front-end features for the SRI Gaussian mixtures-based HMM recognizer.

rior probabilities are combined with the long-term temporal system posteriors using the inverse entropy combination method [94] described in Chapter 4. We then apply the log and reduce the dimensionality to 25 via PCA. Finally, we concatenate this 25 dimensional MLP-based feature with the short-term *HLDA(PLP+3d)* features and use the resulting 64 dimensional feature as inputs to the SRI Gaussian mixtures-based HMM recognizer. This configuration makes it possible to test the additional performance improvements we get when combining all three temporal contexts. This configuration is depicted in Figure 5.7.

5.4 Results

In what follows, we report the word error rate results on Eval2001 for all of the various feature configurations. We use a difference of proportions significance test with 0.05 as the default level to determine statistical significance in our comparisons. For example, anytime we say that system A is significantly better than system B, we mean that the difference in performance between system A and B is statistically significant under this significance test at the 0.05 level. For all of the MLP-based features, we include the frame accuracy which is a measure of how well an MLP classifies the phone classes at the frame

System Description	WER on Eval2001 (%)
Non-Augmented *HLDA(PLP+3d)*	37.2

Table 5.1: Word error rate performance on Eval2001 of a system using conventional feature extraction based on modeling spectral slices.

level. Also note that for fairness of comparisons, the back-end Gaussian mixtures-based HMMs all have roughly the same number of trainable parameters.

5.4.1 Conventional Features

When using the conventional short-term *HLDA(PLP+3d)* features, a simple forward decoding of Eval2001 by the SRI recognizer achieves a 37.2% word error rate (WER) as shown in Table 5.1. For a simple forward decoding pass without adaptation and system combination, 37.2% on Eval2001 is respectable. Indeed, this was state-of-the-art performance a few years ago.

Table 5.2 summarizes the results when using the intermediate-term (about 100 milliseconds) *9 Frame PLP MLP* feature. It has a frame accuracy of 67.57%, which is pretty good for MLP classifiers on Eval2001. When we use the transformed posteriors from the *9 Frame PLP MLP* as features, the SRI recognizer scores a 41.2% WER on Eval2001. This is much worse than the short-term feature alone (41.2% vs. 37.2%), but when we concatenate the dimensionality reduced *9 Frame PLP MLP* feature with *HLDA(PLP+3d)*, the system reduces the WER to 35.6%. This is a 4.3% relative reduction in WER from the system that uses the *HLDA(PLP+3d)* features alone. Relative reductions of 3% or more are typically considered successes when trying to improve system performance on the challenging CTS tasks.

5.4.2 Unconstrained Approaches

The unconstrained approaches for temporal systems ideally could learn any classification function within the 15 critical-bands x 51 frames of log energies. The 3-layer fully-connected *15 x 51 MLP3* classifies 64.73% of the frames correctly, achieves 48.0%

System Description	Frames Correct (%)	Stand-Alone WER (%)	Augment WER (%)
9 Frame PLP MLP	67.57	41.2	35.6

Table 5.2: Conventional *9 Frame PLP MLP* system performances on Eval2001.

System Description	Frames Correct (%)	Stand-Alone WER (%)	Augment WER (%)	Combined-Augment WER (%)
15 x 51 MLP3	64.73	48.0	36.6	34.8
15 x 51 MLP4	67.88	44.3	35.6	34.3

Table 5.3: Unconstrained temporal system performances on Eval2001.

WER in stand-alone feature configuration, performs at 36.6% WER when augmenting *HLDA(PLP+3d)*, and reduces WER to 34.8% in combination with *9 Frame PLP MLP* and augmenting *HLDA(PLP+3d)*. In contrast, the 4-layer fully-connected *15 x 51 MLP4* classifies 67.88% of the frames correctly, achieves 44.3% WER in stand-alone feature configuration, performs at 35.6% WER when augmenting *HLDA(PLP+3d)*, and reduces WER to 34.3% in combination with *9 Frame PLP MLP* and augmenting *HLDA(PLP+3d)*. Table 5.3 lists the results for these unconstrained approaches.

15 x 51 MLP4 significantly outperforms *15 x 51 MLP3* in all feature configurations as well as in frame classification. Although, both have the same total number of parameters, the 4-layer *15 x 51 MLP4* is better able to leverage these parameters for the learning of phonetically discriminant information. Theoretically a 3-layer MLP can learn any mapping function given a sufficient amount of hidden units; however, in practice when there may be constraints in the total number of parameters allowable, a 4-layer MLP can outperform the 3-layer MLP because the extra hidden layer can make the modeling job of later layers easier[2].

[2]We also tried 5-layer MLPs but were unable to achieve comparable performance.

System Description	Frames Correct (%)	Stand-Alone WER (%)	Augment WER (%)	Combined-Augment WER (%)
PCA40	65.50	45.3	36.2	34.6
LDA40	65.52	46.5	36.4	34.5

Table 5.4: Constrained linear temporal system performances on Eval2001.

5.4.3 Constrained Linear Approaches

Table 5.4 shows the performance results for the constrained linear approaches for temporal system design. Both *PCA40* and *LDA40* perform at roughly the same levels except for the stand-alone feature configuration where *PCA40* significantly outperforms *LDA40* (45.3% vs 46.5%). As previously discussed, PCA transforms data in directions of maximal spread, while LDA transforms data in directions of maximal class separability. From these results, performance does not improve by transforming the log critical-band energy trajectories in directions of maximal class separability compared to simply projecting the trajectories along directions of maximal spread.

5.4.4 Constrained Nonlinear Approaches

The first four constrained nonlinear approaches are based on the two-stage Neural TRAP architecture and differ only in the point at which to take the inputs for the second stage merger MLP. All of these two-stage Neural TRAP-based systems learn discriminant information at the critical-band level useful for classifying critical-band level phone targets. On the other hand the *TMLP* learns critical-band level information useful for classifying full-band phone targets. The results of these five approaches are summarized in Table 5.5.

Looking at the Table 5.5, we notice that two systems perform at noticeably higher levels than the other three systems in all feature configurations and frame accuracy. *HAT* and *TMLP* both outperform *HAT Before Sigmoid*, *Neural TRAP*, and *Neural TRAP Post Softmax*. In terms of frame accuracy, *HAT* and *TMLP* perform similarly (66.91% for *HAT* and 67.12% for *TMLP*). The closest competitor is *Neural TRAP* which performs at 65.85% accuracy. When using these five systems in stand-alone feature configuration, *HAT* and *TMLP* have a 44.5% and 44.9% WER respectively. The closest any other system gets to

this performance level is 45.9% WER achieved by both *Neural TRAP* and *HAT Before Sigmoid* which is statistically significantly worse.

The story is consistent when augmenting the *HLDA(PLP+3d)* features with the constrained nonlinear temporal features. Using *HAT* and *TMLP* to augment *HLDA(PLP+3d)*, the WER is 35.6% and 35.5% respectively. The others temporal systems get as close as 36.3% WER achieved by the *HAT Before Sigmoid* system which is still statistically significantly worse. Finally, in the combined and augmented feature configuration, *HAT* and *TMLP* achieve a 34.1% and 33.9% WER on Eval2001. The other systems still underperform *HAT* and *TMLP*, but this time only *TMLP* is significantly better than the others at the 0.05 level. Another .1% absolute difference would make *HAT*'s performance improvement significant.

One of the main findings in Chapter 4 is that *HAT* and *TMLP* perform better than *Neural TRAP* in clean conditions on the TIMIT phone recognition task. The above results on CTS also corroborate these finding; *HAT* and *TMLP* outperform all other Neural TRAP-based systems in clean conditions.

In this chapter we can also make some comments about which critical-band measurements to use as inputs to a merger MLP. Comparing *HAT Before Sigmoid*, *HAT*, *Neural TRAP*, and *Neural TRAP Post Softmax*, we have already commented that *HAT* significantly outperforms all the others. The only difference between *HAT* and *HAT Before Sigmoid* is the sigmoid nonlinearity. Both learn critical-band energy trajectory patterns, but *HAT* uses the sigmoid to transform the inner product of the learned energy trajectory patterns and the input energy trajectories into "probabilities" of these learned energy trajectory patterns. *Neural TRAP* differs from *HAT* by adding an additional mapping from the critical-band hidden unit output space to critical-band level phones. This extra mapping to phones reduces performance, suggesting that phone categories are not the best targets at the critical-band level. *Neural TRAP Post Softmax* normalizes the *Neural TRAP* inputs to the merger MLP to sum to one in each critical-band. The merger MLP in *Neural TRAP Post Softmax* uses critical-band phone posteriors as input features. This also, does not work very well and compared to *Neural TRAP* performance suffers when performing this normalization.

Comparing *TMLP* with *HAT*, we do see a slight improvement from *TMLP* augmenting the short-term *HLDA(PLP+3d)* features as well as in combination with the

System Description	Frames Correct (%)	Stand-Alone WER (%)	Augment WER (%)	Combined-Augment WER (%)
HAT Before Sigmoid	65.80	45.9	36.3	34.9
HAT	66.91	44.5	35.6	34.1
Neural TRAP	65.85	45.9	36.5	34.5
Neural TRAP Post Softmax	63.96	48.2	36.8	34.5
TMLP	67.12	44.9	35.5	33.9

Table 5.5: Nonlinear temporal system performances on Eval2001.

intermediate-term *9 Frame PLP MLP* and augmenting *HLDA(PLP+3d)* features. However, as a stand-alone feature *TMLP* performs worse than *HAT*. From this, it seems that the more unconstrained *TMLP* learns information that is marginally more complementary to the conventional features than *HAT*.

5.4.5 Augmenting Conventional Features

In this subsection we take a closer look at the improvements that each of the MLP-based systems bring when augmenting the short-term *HLDA(PLP+3d)* features. Table 5.6 summarizes the WER results and relative improvements over using *HLDA(PLP+3d)* features alone for the various MLP-based features augmenting the *HLDA(PLP+3d)* features. *9 Frame PLP MLP*, *15 x 51 MLP4*, *HAT*, and *TMLP* all outperform the other MLP-based features obtaining a 35.6%, 35.6%, 35.6%, and 35.5% WER respectively on Eval2001. The rest of the systems perform considerably worse at 36.2% and higher. As commented before, a 3% relative reduction or more in WER is considered impressive for such a difficult task as CTS. All long-term systems improve WER compared to *HLDA(PLP+3d)* alone. The intermediate-term *9 Frame PLP MLP* also improves performance significantly, and it does so to the same extent as the long-term systems of *15 x 51 MLP4*, *HAT*, and *TMLP*.

5.4.6 Combined-Augmented Features

Table 5.7 displays the WER results for all of the temporal systems-based features in combination with the *9 Frame PLP MLP* features which are then used to augment the *HLDA(PLP+3d)* features. From these results, we can see how much more improvement

System Description	Eval2001 WER (%)	Relative Improvement (%)
Baseline: Non-Augmented *HLDA(PLP+3d)*	37.2	-
9 Frame PLP MLP	35.6	4.3
15 x 51 MLP3	36.6	1.6
15 x 51 MLP4	35.6	4.3
PCA40	36.2	2.7
LDA40	36.4	2.2
HAT Before Sigmoid	36.3	2.4
HAT	35.6	4.3
Neural TRAP	36.5	1.9
Neural TRAP Post Softmax	36.8	1.1
TMLP	35.5	4.6

Table 5.6: Comparison of all MLP-based features used to augment the short-term *HLDA(PLP+3d)* features. WER results as well as relative improvement over the *HLDA(PLP+3d)* features alone reported for Eval2001.

we can obtain by combining the long-term information to the medium and short-term information streams. The baseline performance comes from the augmenting the short-term features with the intermediate-term features of *9 Frame PLP MLP*. This baseline system gets a 35.6% WER on Eval2001.

Combining the any of the long-term information streams to the short and intermediate-term streams improves performance. The best long-term information stream comes from *TMLP* followed closely by *HAT*. The unconstrained *15 x 51 MLP4* is slightly worse than *TMLP* and *HAT*, but slightly better than all of the other temporal systems. We surmise that the narrow-band frequency constraints imposed by *HAT* and *TMLP* help it to learn more complementary information to the *9 Frame PLP MLP* system than that learned by the unconstrained *15 x 51 MLP4* system. From this, we conclude that the narrow-band frequency constraint in the long-term systems is useful in combination with the conventional *9 Frame PLP MLP* system, but it must be implemented appropriately (for example, in the form of *HAT* or *TMLP* or perhaps other improved Neural TRAP-based extensions that we did not test here).

System Description	Eval2001 WER (%)	Relative Improvement (%)
Baseline: *9 Frame PLP MLP*	35.6	-
15 x 51 MLP3	34.8	2.2
15 x 51 MLP4	34.3	3.7
PCA40	34.6	2.8
LDA40	34.5	3.1
HAT Before Sigmoid	34.9	2.0
HAT	34.1	4.2
Neural TRAP	34.5	3.1
Neural TRAP Post Softmax	34.5	3.1
TMLP	33.9	4.8

Table 5.7: Table of results for systems combined with the *9 Frame PLP MLP* features and augmenting the *HLDA(PLP+3d)* features. WER and relative improvements over the baseline *9 Frame PLP MLP* augmented system on Eval2001 are reported.

5.4.7 Overall Comparison of Temporal Systems

Table 5.8 shows the rankings for each of the various temporal systems in all of the different feature configurations and their frame accuracies. The *15 x 51 MLP4* system does the best at the frame level as well as in the stand-alone feature configuration; however, when combined with the other full-band features, *HAT* and *TMLP* perform better than the *15 x 51 MLP4* system. We mention again that this is because of the narrow-frequency constraints imposed by the *HAT* and *TMLP* systems, which force these two systems to model critical-band temporal patterns. The *15 x 51 MLP3* and *Neural TRAP Post Softmax* systems almost always perform the worse, while all the other systems show no predictable pattern of performance. The nonlinear constrained approaches consistently perform better than their linear counterparts only when using *HAT* and *TMLP*. To summarize these findings:

1. The narrow-band constraints are most helpful in combination with either the full-band short and intermediate-term feature streams if implemented in the form of HAT or TMLP.

2. The *HAT* and *TMLP* nonlinear constrained systems perform better in all feature configurations than the linear constrained systems.

System Description	Frames Correct Rank	Stand-Alone Rank	Augment Rank	Combined-Augment Rank
15 x 51 MLP3	8	8	8	8
15 x 51 MLP4	1	1	2	3
PCA40	7	4	4	7
LDA40	6	7	6	4
HAT Before Sigmoid	5	5	5	9
HAT	3	2	2	2
Neural TRAP	4	5	7	4
Neural TRAP Post Softmax	9	9	9	4
TMLP	2	3	1	1

Table 5.8: Rankings of the various temporal systems on Eval2001

5.4.8 *Neural TRAP* With More Hidden Units

In previous implementations of Neural TRAP (e.g [53, 112, 62]), researchers use many more hidden units than the 40 hidden unit implementations in this chapter. Table 5.9 shows the performance of Neural TRAP systems with both 40 and 300 hidden units per critical-band. The TRAP systems with 300 hidden units per critical-band have about 380,000 more total parameters than the ones with 40. In general both the 40 and 300 hidden unit versions perform equally except in two cases: 1) *Neural TRAP* in the augmented feature configuration where the 300 hidden unit version is significantly better (36.0% versus 36.5%), and 2) *Neural TRAP Post Softmax* in the stand-alone feature configuration where the 300 hidden unit version is much worse. We cannot conclude that increasing the number of critical-band hidden units to 300 always improves performance for *Neural TRAP Post Softmax*, but in the *Neural TRAP* systems increasing to 300 never leads to performance degradation.

5.5 Frame Accuracy Analysis of the Best Temporal Systems

In the previous sections we have seen how the MLP-based features derived from temporal systems have complemented both the intermediate and short-term features, leading to substantial reductions in the word error rate on a CTS task. In this section we would

System Description	Frames Correct (%)	Stand-Alone WER (%)	Augment WER (%)	Combined-Augment WER (%)
40 *Neural TRAP*	65.85	45.9	36.5	34.5
300 *Neural TRAP*	66.43	45.9	36.0	34.3
40 *Neural TRAP Post Softmax*	63.96	48.2	36.8	34.5
300 *Neural TRAP Post Softmax*	63.73	49.1	36.6	34.2

Table 5.9: System performances on Eval2001 of *Neural TRAP* with 40 hidden units versus *Neural TRAP* with 300 hidden units per critical-band. With 300 hidden units per critical-band *Neural TRAP* and *Neural TRAP Post Softmax* perform at about the same level as *HAT* in the combined-augmented configuration using 380,000 more parameters.

like to dig a little deeper and find out what phone categories these temporal systems do particularly well on compared to the intermediate-term *9 Frame PLP MLP* as well as to each other. Because *HAT*, *TMLP*, and *15 x 51 MLP4*, outperformed all other temporal systems, we focus our attention on these three temporal systems in our analysis.

All temporal systems and the intermediate-term *9 Frame PLP MLP* system are MLP-based classifiers that we train to learn 46 phone classes. As described earlier, the phone targets for MLP training come from forced-alignments from the SRI recognizer whose dictionary of words consists of sequences of these 46 phones. Table 5.10 lists all of the phone classes (as well as an example or description of its usage) that we train our MLPs on. Once trained, our MLP-based classifiers output a phone probability distribution for every frame of speech. We consider a classifier to have correctly classified a particular frame of speech when the maximum phone probability output corresponds to the labeled phone target. As described in Section 3.1, frame accuracy is calculated by counting how many frames a classifier gets correct divided by the total number of test frames.

When comparing two classifiers at the frame level, we can do better than simply comparing the gross frame accuracy measure. We can calculate accuracy measures on a per phone class basis to see which classifier does better on what phone. For any frame, one of four outcomes is possible: 1) both classifiers get the frame correct, 2) only the first classifier gets it correct, 3) only the second classifier gets it correct, or 4) both get it wrong. If we sum up the counts of these outcomes for frames labeled a certain phone, we can immediately see which classifier is better at classifying this phone. For example, the first classifier is better for this phone if the counts of case 2 outcomes is greater than the

ASR Phoneme Symbols			
SRI 46	Example	SRI 46	Example
sil	(silence)	k	**k**ey
aa	f**a**ther	l	**l**ike
ae	b**a**t	lau	(laughter)
ah	b**u**t	m	**m**oon
ao	b**ou**ght	n	**n**oon
aw	ab**ou**t	ng	si**ng**
ax	**a**bout	ow	b**oa**t
ay	b**i**te	oy	b**oy**
b	**b**ee	p	**p**ea
ch	**ch**oke	puh	(filled-pause vowel)
d	**d**ay	pum	(filled-pause nasal)
dh	**th**en	r	**r**ight
dx	dir**t**y	s	**s**ound
eh	b**e**t	sh	**sh**out
er	b**ir**d	t	**t**ea
ey	b**ai**t	th	**th**in
f	**f**ish	uh	b**oo**k
fip	(word fragment interruption point)	uw	b**oo**t
g	**g**ay	v	**v**ote
hh	**h**ay	w	**w**ire
ih	b**i**t	y	**y**es
iy	b**ee**t	z	**z**oo
jh	**j**oke	zh	a**z**ure

Table 5.10: The 46 monophone targets used for MLP training.as defined for SRI's recognition system.

counts of case 3 outcomes. The counts of case 1 and case 4 outcomes reveal the difficulty of classifying a particular phone and possibly the inaccuracy of the labeling of the phone that we use as ground truth. Mostly, we are interested in the counts of case 2 and case 3 because they give us an indication of which classifier is better.

In Tables 5.11-5.13, we calculate the counts for all of the above cases normalized by the total number of frames for a particular phone. The result is the percentage of frames that the outcome occurs for a particular phone. The phone are listed in order of how well the temporal system does on that phone compared with the *9 Frame PLP MLP* system, and we only list those phones for which the temporal system is better. The tables also list the average phone duration in frames and the total number of frames labeled with that phone on the Eval2001 set. Using these tables, we address the following questions:

1. Do the temporal systems perform better on longer phones?
2. What phones do the temporal systems do consistently better on than the *9 Frame PLP MLP* system?
3. As we remove constraints in the learning of long-term information, what phones are more accurately classified?
4. As we add constraints in the learning of long-term information, what phones are more accurately classified?

5.5.1 Temporal Systems and Longer Phones

To answer the first question, we calculate the average phone durations for all the phones that a particular temporal system is better at classifying than the *9 Frame PLP MLP* system and vice versa. When comparing *HAT* and *9 Frame PLP MLP*, the average phone duration of all the phones that *HAT* is better at classifying is 13.0 frames, while the the average phone duration for the phones that *9 Frame PLP MLP* is better at is 8.7 frames. The phones that *TMLP* is better at classifying have an average duration of 11.1 frames compared to 9.4 frames for the phones that *9 Frame PLP MLP* is better at. Finally, when comparing *15 x 51 MLP4* with *9 Frame PLP MLP*, the average durations are 10.5 frames for *15 x 51 MLP4* and 10.0 frames for *9 Frame PLP MLP*. Overall, the temporal systems do perform better on longer phones. These results are consistent with

Phone	Avg. Dur. (Frames)	Both Right (%)	*HAT* Right (%)	*PLP MLP* Right (%)	Both Wrong (%)	Total Phone Counts
oy	14.00	11.0	17.6	8.5	63.0	2028
ae	11.00	49.6	17.4	9.6	23.4	73780
hh	5.00	25.6	14.3	8.8	51.3	32070
zh	10.00	24.3	14.8	10.2	50.7	391
ay	15.00	53.8	13.2	9.5	23.5	73458
z	3.00	38.3	14.8	11.4	35.5	32521
ey	12.00	40.7	14.5	11.6	33.1	35987
ow	11.00	34.8	16.2	13.3	35.6	56401
puh	19.00	48.7	14.8	12.2	24.3	45210
dx	5.00	28.7	12.1	10.1	49.1	5284
pum	11.00	30.4	15.9	14.7	39.0	33913
ax	5.00	34.2	11.9	10.7	43.2	75242
th	21.00	23.3	13.2	12.0	51.5	10507
lau	41.00	45.5	13.8	13.2	27.5	38014
aw	21.00	20.5	14.8	14.3	50.5	14675
fip	3.00	0.2	2.0	1.8	95.9	4351

Table 5.11: Frame level classification statistics for *HAT* versus *9 Frame PLP MLP*.

what we would expect because the long-term systems are learning patterns spanning 51 frames, while the *9 Frame PLP MLP* system only gets 9 frames of input context to work with.

Recall, that we can view the progression of going from *HAT* to *TMLP* to *15 x 51 MLP4* as a progression of loosening constraints. As we move from *HAT* to *TMLP*, we are loosening the constraint of learning critical-band level phone labels. As we move from *TMLP* to *15 x 51 MLP4*, we remove the narrow-frequency channel constraint. As we loosen the constraints on the learning of long-term patterns (i.e., going from *HAT* to *TMLP* to *15 x 51 MLP4*), the difference between the average duration from the temporal system and the average duration from *9 Frame PLP MLP* decreases. It seems that the each of the constraints help the temporal systems better focus on learning long-term information from phones that have higher average durations.

Phone	Avg. Dur. (Frames)	Both Right (%)	*TMLP* Right (%)	*PLP MLP* Right (%)	Both Wrong (%)	Total Phone Counts
ae	11.00	50.4	18.0	8.8	22.8	73780
oy	14.00	10.2	16.1	9.3	64.4	2028
puh	19.00	51.0	16.4	9.9	22.7	45210
ow	11.00	36.7	17.0	11.5	34.9	56401
th	21.00	25.6	15.0	9.6	49.9	10507
hh	5.00	26.0	13.5	8.4	52.0	32070
ay	15.00	54.2	13.4	9.0	23.4	73458
ey	12.00	41.6	14.9	10.7	32.8	35987
z	3.00	39.4	14.5	10.3	35.9	32521
lau	41.00	47.9	14.9	10.8	26.4	38014
aw	21.00	22.5	16.2	12.3	49.0	14675
ax	5.00	35.7	13.0	9.2	42.1	75242
f	7.00	40.8	14.5	11.5	33.1	24710
dh	3.00	33.2	14.5	12.4	40.0	29534
dx	5.00	29.2	11.7	9.7	49.4	5284
pum	11.00	30.7	15.9	14.4	39.0	33913
y	9.00	54.1	12.2	10.7	23.1	38136
uw	3.00	36.7	12.4	11.5	39.4	29316
d	6.00	22.8	10.9	10.2	56.2	35311
aa	9.00	24.7	14.7	14.1	46.6	24764
jh	13.00	40.5	11.9	11.5	36.1	8795
fip	3.00	0.1	2.2	1.9	95.8	4351
sil	16.00	92.0	2.7	2.4	3.0	762542
ng	11.00	37.8	10.6	10.3	41.3	17417
ah	3.00	22.3	12.4	12.3	53.0	30033

Table 5.12: Frame level classification statistics for *TMLP* versus *9 Frame PLP MLP*

Phone	Avg. Dur. (Frames)	Both Right (%)	*MLP4* Right (%)	*PLP MLP* Right (%)	Both Wrong (%)	Total Phone Counts
oy	14.00	11.2	19.8	8.2	60.8	2028
aw	21.00	24.8	19.9	10.1	45.3	14675
ae	11.00	50.3	17.5	8.9	23.3	73780
ow	11.00	37.0	18.0	11.1	33.9	56401
puh	19.00	49.7	16.4	11.2	22.7	45210
ay	15.00	54.5	13.7	8.8	23.1	73458
ey	12.00	42.2	14.8	10.1	32.9	35987
hh	5.00	26.1	12.4	8.4	53.2	32070
z	3.00	38.6	14.3	11.0	36.1	32521
ax	5.00	35.3	12.9	9.6	42.3	75242
dx	5.00	29.2	12.7	9.6	48.5	5284
lau	41.00	47.2	14.4	11.4	27.0	38014
pum	11.00	32.1	15.9	13.1	39.1	33913
aa	9.00	25.4	15.6	13.3	45.6	24764
th	21.00	23.7	13.7	11.5	51.1	10507
s	6.00	54.2	13.2	11.2	21.4	70534
r	6.00	47.4	14.6	12.6	25.4	51308
y	9.00	53.9	12.2	10.8	23.1	38136
eh	5.00	21.0	12.9	11.8	54.3	33454
dh	3.00	32.5	14.0	13.0	40.4	29534
f	7.00	40.0	13.3	12.4	34.3	24710
ah	3.00	22.2	13.3	12.4	52.1	30033
zh	10.00	21.5	13.8	13.0	51.7	391
d	6.00	22.5	11.1	10.5	55.9	35311
jh	13.00	40.5	11.8	11.6	36.2	8795
uw	3.00	36.2	12.2	12.0	39.6	29316
t	4.00	33.4	11.5	11.3	43.8	73020
sil	16.00	91.9	2.5	2.4	3.1	762542

Table 5.13: Frame level classification statistics for *15 x 51 MLP4* vs. *9 Frame PLP MLP*

5.5.2 Temporal Systems Versus *9 Frame PLP MLP*

To answer the second question of what phones the temporal systems are generally better at classifying than *9 Frame PLP MLP*, we examined the intersection of the phones that appear in Tables 5.11, 5.12, and 5.13. These phones are the phones for which all three temporal systems are better at classifying than *9 Frame PLP MLP*. The most prominent observation from this is that all of the temporal systems consistently classify diphthongs (/aw/, /ay/, /ey/, /ow/, and /oy/) better than *9 Frame PLP MLP*. Diphthongs are phones that start off sounding like one vowel and end sounding like another vowel. The average duration of diphthongs is 13.6 frames in the Eval2001 data set, which is 4.6 frames more than the input context to *9 Frame PLP MLP*. Because the temporal systems have 51 frames of context to work with, they can better model these diphthongs.

Other phones which these temporal systems are consistently better at classifying include: /ae/, /puh/, /pum/, /hh/, /th/, /z/, /ax/, /lau/, and /dx/. /ae/, /puh/, /pum/, /th/, and /lau/ have average durations longer than 9 frames. The filled paused vowel /puh/ (as used when people say "uh") and the filled paused nasal /pum/ (as used when people say "ummm"), seem like phones that can be easily confused with regular phones like /ah/ and /m/. With more temporal context, *HAT*, *TMLP*, and *15 x 51 MLP4* seem to be able to disambiguate these filled pause phones better than the *9 Frame PLP MLP* system. It is interesting that these temporal systems outperform *9 Frame PLP MLP* on some short phones also (i.e., /hh/, /z/, /ax/, and /dx/). Perhaps, there is a lot of contextual information about these phones that the temporal systems are able to capture and exploit.

5.5.3 Temporal Systems Versus Each Other

In the context of our augmented combination system, where we combine the outputs of one of the temporal systems with the outputs from the intermediate-term *9 Frame PLP MLP* system, and use this combination to augment the conventional short-term features, it is interesting to analyze what happens when we remove or add learning constraints on the temporal systems. As we move from *HAT* to *TMLP* to *15 x 51 MLP4*, we are removing constraints on the learning of long-term information. We can see the effect of removing constraints on performance by looking at all the phones for which a more

constrained temporal system performs better at than *9 Frame PLP MLP* but that the less constrained temporal system does not perform better at than *9 Frame PLP MLP*. For example, by looking for phones that appear in Table 5.12 but do not appear in Table 5.11, we can see which phones are better classified when removing the constraint of learning critical-band phone labels. Similarly, by looking for phones that appear in Table 5.13 but do not appear in Table 5.12, we can see which phones are better classified when removing the constraint of learning within critical-bands. To see the effect of adding constraints, we simply reverse the order of our table comparisons and look for phones which appear in the table for the more constrained system but do not appear in the less constrained system's table.

Comparing *HAT* to *TMLP*, we see from Tables 5.11 and 5.12 that the following phones appear in Table 5.12 but not in Table 5.11: /f/, /dh/, /y/, /uw/, /d/, /aa/, /jh/, /sil/, /ng/, and /ah/. Removing the critical-band constraints (going from *TMLP* to *15 x 51 MLP4*, we see from Tables 5.12 and 5.13 that the phones /s/, /r/, /eh/, /zh/, and /t/ are better classified. When tightening the constraints from *15 x 51 MLP4* to *TMLP*, /ng/ is the only phone that is improved, while going from *TMLP* to *HAT* only /zh/ is improved. Generally, loosening the constraints helps the temporal systems to better classify phones, but we have also noticed that in combination with *15 x 51 MLP4*, the narrow-band constraint does make the temporal systems more complementary leading to larger reductions in word error rates (e.g., compare the combined-augmented results for *TMLP*, 33.9%, versus *15 x 51 MLP4*, 34.3%).

5.6 Narrow-Band Discriminant Temporal Patterns

In Section 3.10, we discussed the nature of the discriminant temporal patterns learned by HAT and TMLP on TIMIT speech data. In this section, we not only examine what was learned by *HAT* and *TMLP* on CTS data, but we also look at what temporal patterns were learned by *PCA40* and *LDA40*. As explained in Section 3.10, the critical-band hidden units of HAT and TMLP perform filtering operations on the log critical-band energy trajectories of speech. When trained on TIMIT data these matched temporal filters coming from both HAT and TMLP tended to filter out modulation frequencies above 20 Hz. The *PCA40* and *LDA40* transformations that we trained on the log critical-band

energy trajectories can similarly be considered as matched temporal filters as well. In Appendix D, we show plots of the cluster centroids of input-to-hidden weights of critical-band hidden units for the *HAT* and *TMLP* systems trained on the female portion of the CTS training set in this chapter. In Appendix E, we show plots of the cluster centroids of the *PCA40* and *LDA40* transformation vectors trained on the female portion of the CTS training set in this chapter.

Comments similar to the ones in Section 3.10 can be made here also. The *HAT* and *TMLP* discriminant temporal patterns mostly tend to emphasize only modulation frequencies below 20 Hz which has been shown to be important for speech recognition. *TMLP* patterns tend to also exhibit more shifting in time than the *HAT* patterns: there seem to be more patterns where the regions of varying magnitudes are not centered at frame 0. Like the patterns in Section 3.10 and Appendix C, the patterns learned by *HAT* and *TMLP* in this chapter do resemble previous patterns found in literature [10, 124, 115, 67, 112]. There are onset detector patterns, "Mexican hat" energy detector patterns, and patterns that resemble Mean TRAPs.

The most striking differences come from looking at the patterns learned by *HAT* and *TMLP* versus those learned by *PCA40* and *LDA40*. The first main difference between the sets is that both *PCA40* and *LDA40* have learned some patterns that are sensitive to modulation frequencies greater than 20 Hz. These patterns are capturing temporal information that is not necessarily essential for speech recognition which explains to some extent why the *PCA40* and *LDA40* temporal systems in this chapter were less effective than the *HAT* and *TMLP* systems for improving performance. The next striking difference is that all of the *PCA40* patterns look like sinusoids of different frequencies. What this implies about speech within narrow-frequency bands is interesting: this means that the directions of highest variance all correspond to sinusoidal functions with different oscillation frequencies. Finally, the *LDA40* patterns look somewhat like a mix between *PCA40* patterns and *HAT* patterns. Of all the *LDA40* patterns some also look like rapidly varying sinusoids, but there are other patterns that more resemble those learned by *HAT* and *TMLP*. We have also observed that the top *LDA40* discriminants (i.e., the ones corresponding to the highest eigenvalues) look like the onset detectors and "Mexican hat" patterns consistent with previous LDA studies.

5.7 HAT and TMLP Practical Trade-offs

There are some notable observations concerning the training process of HAT and TMLP. As explained earlier, the training of HAT proceeds in two stages. The first stage is to train all the critical-band MLPs. In the second stage, we first compute the hidden unit outputs of all critical-band MLPs from the input critical-band energy trajectories of the training data. The set of all of these hidden unit outputs becomes the input training data for the second stage merger MLP training. The first stage can be parallelized to train on several computers simultaneously. There is some savings in time by training this first stage in parallel; however, the first stage training is much quicker than the second stage merger training because the critical-band MLPs are rather small (only 20 hidden units in Chapter 3 and 40 hidden units in this chapter), and so the overall training time is dominated by the second stage merger training.

One potential drawback from our implementation of this two-stage HAT training is the need for temporary disk space to store the hidden unit outputs from all the critical-band MLPs on the complete training set. A small training set such as the one in Chapter 3 (about 1 million frames), requires 380 million 4 byte floats (1 million frames x 20 hidden units per critical-band x 19 critical-bands x 4 bytes = 1.52 gigabytes) of temporary disk storage. The CTS training sets in this chapter have about 12 million frames per gender which means that we need about 29 gigabytes of temporary disk storage (12 million frames x 40 hidden units per critical-band x 15 critical-bands x 4 byte floats = 28.8 gigabytes) per gender for training *HAT*[3]. *TMLP*, in contrast, requires no such temporary disk space since all critical-band hidden unit outputs are propagated within the network during training.

Although temporary disk space is not an issue for training TMLP, there is a trade-off with the time needed for training. The time required for training TMLP is typically longer than that for training HAT. In HAT the critical-band hidden units can be trained in parallel, but in TMLP the critical-band hidden units are trained along with all the other TMLP parameters within a single network optimization routine. Another reason why TMLP trains slower than HAT is that the optimized linear algebra routines run less efficiently because the TMLP's band-constrained 2 hidden layer topology leads to

[3]The training set used for training SRI's 2004 CTS recognizer has about 40 times more frames per gender than the training set used in this chapter which would require over a terabyte of temporary disk storage!

	33 Hour Set		66 Hour Set	
System	Approx. Training Time	Temporary Disk Space	Approx. Training Time	Temporary Disk Space
HAT	28.5 hours	28.8 GB	100.5 hours	86.4 GB
TMLP	32.9 hours	0	140.8 hours	0

Table 5.14: A comparison of training time and disk space requirements for HAT and TMLP trained on a 33-hour and 66-hour training set. The systems trained on the 33-hour set have about 516,000 parameters and 40 hidden units per critical-band, and the systems trained on the 66-hour set have about 1,032,000 parameters and 60 hidden units per critical-band.

operations on matrices that are either thin and tall or short and wide. Because each stage of HAT training operates on single hidden layer MLPs with regular topologies, the linear algebra routines run faster allowing for quicker HAT training.

To compare training times of HAT and TMLP, we trained both HAT and TMLP on two different training sets. The first training set is the one used in this chapter for training male systems (about 33 hours and 12 million frames). The second training set is a superset of the first and contains about twice as much male speech data (about 66 hours and 24 million frames). The HAT and TMLP systems trained on the first set have about 516,000 total parameters and 40 hidden units per critical-band, while the systems trained on the second set have twice as many total parameters and 60 hidden units per critical-band. Table 5.14 shows the actual training times and temporary disk space required for training each of the four systems on an Intel Xeon 2.80GHz machine with 3 GB of memory. The HAT training times includes the savings from parallelizing the critical-band hidden unit training and the time needed to process the intermediate hidden unit activation files for the second stage training.

Table 5.14 illustrates the practical trade-off of training HAT and TMLP. HAT trainings run faster at the cost of large amounts of temporary disk space, while TMLP trainings run slower and save in disk space as well as human operator effort required for preparing the intermediate HAT training files.

5.8 Conclusions

In this chapter we have compared various temporal systems for the learning of long-term (about 500 milliseconds) information useful for ASR on CTS. We compared their performance using three different Tandem ASR configurations: stand-alone Tandem, augmented Tandem, and combined-augmented Tandem. The various temporal systems constrain the learning of long-term information in different ways. The *15 x 51 MLP3* and *15 x 51 MLP4* systems do not constrain the learning within the 15 critical-bands by 51 frames matrix of log energies. The *TMLP* system constrains the classifier to learn important distinctions within individual 51-frame critical-band energy trajectories. Finally, the *PCA40*, *LDA40*, *HAT Before Sigmoid*, *HAT*, *Neural TRAP*, and *Neural TRAP Post Softmax* systems constrain the learning within critical-bands, but also forces the systems to learn transformations useful for classifying phone labels at the critical-band level (except for the *PCA40* system which learns transformations in directions of highest variance).

We found that three temporal systems outperformed all others in all three system configurations: the unconstrained *15 x 51 MLP4*, *TMLP*, and *HAT* temporal systems. When comparing these three systems, we saw an advantage to the critical-band constrained *TMLP*, and *HAT* temporal systems in combination with the intermediate-term *9 Frame PLP MLP* system, suggesting that the critical-band constraints help to make our temporal systems more complementary to the *9 Frame PLP MLP* system. Also, the two best nonlinear critical-band constrained systems, *TMLP* and *HAT*, outperformed all linear critical-band constrained systems, *PCA40* and *LDA40*, in all system configurations. This suggests that it is important to learn "probabilities" of something fundamentally discriminant at the critical-band level for later stages in the MLP classifier.

Performing further analysis as to which phone classes our temporal systems classify better, we found that the temporal systems tend to do better on phones that have longer average durations. Compared with the intermediate-term *9 Frame PLP MLP* system, we also found that the temporal systems consistently perform better on diphthongs, filled pauses, laughter, and a few other phones (/ae/, /hh/, /th/, /z/, /ax/, and /dx/).

The narrow-band frequency patterns learned by *HAT* and *TMLP* systems again preserve the important low modulation frequencies of speech needed for recognizing words. The patterns learned by *LDA40* and *PCA40* differ from those learned by *HAT* and *TMLP*

in that they also pass modulation frequencies greater than 20 Hz. Moreover, the patterns learned by *PCA40* all look like sinusoidal functions of different frequencies. Lower order *LDA40* basis vectors look somewhat like noisy sinusoids, while the higher order ones resemble patterns learned by *HAT* and *TMLP*.

When training *HAT* and *TMLP* systems, we commented that in general *HAT* systems train faster, but *TMLP* systems do not require any temporary disk space for training.

Our best system in this chapter, the combination of *TMLP* and *9 Frame PLP MLP* features augmenting the conventional *HLDA(PLP+3d)* features, achieved a WER of 33.9% on Eval2001. The conventional *HLDA(PLP+3d)* features get a WER of 37.2%. This is an absolute reduction in WER of 3.3% (or 8.9% relative) compared to using *HLDA(PLP+3d)* features alone, which was the state-of-the-art feature used in 2003. An improvement of this magnitude on CTS is considered impressive and is about half the gain achieved by most evaluation teams after a year of collective work.

Chapter 6

Further Explorations With TMLP

In previous chapters we developed several new neural net architectures for the learning of long-term narrow-frequency band information useful for ASR. We started by testing HAT, TMLP, and Neural TRAP on a small recognition task - recognizing phones from the TIMIT corpus. Then we set up a series of recognition tasks leading to the development of a baseline system utilizing new front-end feature for the recognition of conversational telephone speech (CTS). In Chapter 5, we compared various neural net systems learning long-term information for recognizing CTS. In this chapter we further explore one of the best long-term systems: the TMLP.

We begin by examining the choice in the number of critical-band hidden units in the TMLP. Specifically, we are interested in determining how performance is affected by the choice in the number of critical-band hidden units as the amount of training data and total parameters are varied. Because the critical-band hidden units can be thought of as probability estimators of discriminant temporal patterns, choosing how many of them to use is equivalent to choosing how many discriminant temporal patterns we would like the TMLP to learn. From previous work on Mean TRAPs, there seems to be a finite number of important temporal patterns at the critical-band level necessary for high accuracy. Likewise, we find that the optimal number of critical-band hidden units does not grow when increasing the amount of training data.

In the second part of this chapter, motivated by previous work on UTRAP [50] which hypothesized that multiple critical-bands have similar temporal patterns, we inves-

tigate whether the discriminant temporal patterns can be shared across multiple critical-bands. We develop TMLPs that can share the parameters of critical-band hidden units among different critical-bands. These shared hidden units are trained and applied on speech data coming from multiple critical-bands. We find that performance remains high when sharing critical-band hidden units suggesting that different critical-bands share similar discriminant temporal patterns useful for ASR.

6.1 The Growth of Critical-Band Hidden Units

When we moved to apply HAT and TMLP on CTS, the optimal number of hidden units per critical-band jumped from 20 for the smaller TIMIT task to 40 for the CTS task. Perhaps this comes from having much more training data in the CTS task than in TIMIT training (3.12 hours of TIMIT training data versus about 35 hours of CTS training data per gender[1].). This leads us to the question that we wish to answer in this section:

- How does the amount of training data affect the optimal choice for the number of hidden units per critical-band in the TMLP?

To answer this question, we created four CTS training sets differing in the total number of hours of speech. These four CTS training sets come from the same sources used for creating the baseline CTS training set in Chapter 5: English CallHome [19], Switchboard I with transcriptions from Mississippi State [41, 28], and Switchboard Cellular [43]. The first new training set consists of about 124.9 hours (about 20 million frames per gender) of speech data from the above sources. In all of the four new training sets, we maintained an equal balance between the amount of male and female training data. Subsampling the 124.9 hour set by 2, 4, and 8 resulted in a 62.4 hour (about 10 million frames per gender), 31.2 hour (about 5 million frames per gender), and 15.6 hour (about 2.5 million frames per gender) training set respectively.

Once these training sets were completed, we started to investigate the interactions between the number of critical-band hidden units, the total number of trainable parameters, and the amount of training data. We trained TMLPs with 20, 30, 40, 50, and 60

[1]Recall that the TIMIT nets are gender independent nets, while the CTS nets are gender dependent nets (one net for each gender), so for fairness of comparison, we compare how much data it takes to train single networks.

hidden units per critical-band, and for each of these cases, we chose the second hidden layer size such that the total number of parameters was either 250,000, 500,000, 1,000,000, or 2,000,000. Training for each TMLP setting was done separately for each gender, and the performance numbers that follow reflect the average performance from both genders. The training procedure was the same procedure used for training TMLPs in Chapter 5. Basically, we calculated 15 log critical-band energies for every 10 milliseconds of speech, normalized the mean and variance of these energies over every utterance, and used these as input features for the TMLPs. Holding out 10% of the training data as a cross-validation set, we used the error back-propagation algorithm to minimize the cross-entropy between the TMLP outputs and the phone targets. These phone targets were the same kind of phone targets derived from forced alignments from the SRI recognizer in Chapter 5.

Once training completed, we measured the frame accuracies on the separate Eval2001 CTS test set as described in Subsection 5.3.1. Figure 6.1 shows four graphs of frame accuracy on Eval2001 versus the number of hidden units per critical-band of TMLPs for the four different amounts of training data. Each panel corresponds to one of the four training set sizes (15.6 hours, 31.2 hours, 62.4 hours, or 124.9 hours), and within each panel there are four curves of frame accuracies corresponding to the four TMLP sizes (250,000, 500,000, 1,000,000, or 2,000,000).

All curves in Figure 6.1 exhibit a max accuracy between 30 and 50 hidden units per critical-band except for the 1M parameters/15.6 hour case which has a max at 60. Only the 500k parameters/15.6 hour and 1M parameters/15.6 hour cases show trends that may indicate higher accuracies for greater than 60 hidden units per critical-band. To answer whether increasing the amount of training data leads to an increasing number of hidden units per critical-band for optimal performance, compare the lines corresponding to TMLPs with the same number of trainable parameters in each of the four panels. The curves for 250,000 parameters exhibit a maximum at 30 hidden units per critical-band regardless of the amount of data. For the TMLPs with 500,000 parameters, the maximum accuracy moves from 40, to 50, to 40, to 40 hidden units per critical-band as we double the amount of training data. In the 1,000,000 parameters case, the maximum accuracies goes from 60, to 60, to 40, to 40 hidden units per critical-band, and in the 2,000,000 parameters case, the maximum goes from 50, to 50, to 40, to 40 hidden units per critical-band for each doubling in the amount of training data.

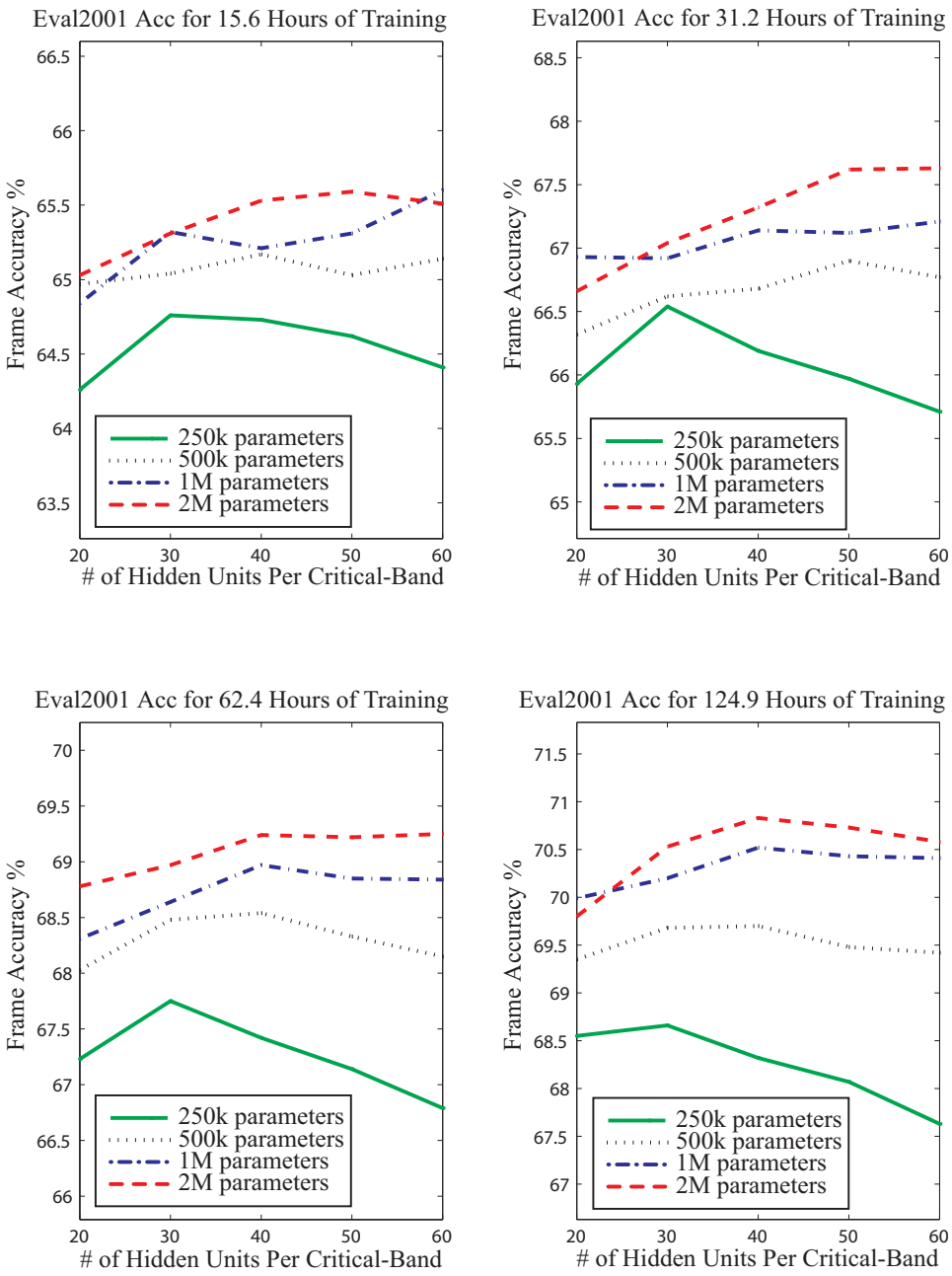

Figure 6.1: Frame accuracies on Eval2001 for various TMLPs.

From these observations, we find that as we increase the amount of training data, the optimal choice for the number of hidden units per critical-band actually decreases when keeping the total number of parameters fixed. However, it does appear to be the case that as the number of total parameters increases, the best number of critical-band hidden units increases slightly. This can be seen clearly in the 62.4 and 124.9 hour panels. See how the best number of hidden units goes from 30 for 250k parameters to between 30-40 for 500k parameters, and to 40 for 1M and 2M parameters.

In a previous empirical study on training MLPs for use in a hybrid ANN/HMM system on Broadcast News [33], Ellis et al. explored the optimal ratio of the number of training examples to number of trainable MLP parameters for a fixed training time. They found that the optimal ratio of number of training example frames to number of parameters was in the range of 10 to 40 for a constant product of training frames and parameters. The product of training frames and parameters gives a measure of how long it takes to train an MLP because in each epoch of training all the parameters are updated N times where N is a number proportional to the number of total training frames[2].

We plot the average frame accuracies for TMLPs of constant N (connection updates (CUPs) per epoch) versus the ratio of frames to parameters in Figure 6.2. From this figure we can see a slowing of accuracy improvements as the ratio of frames per parameter increases. There is a decrease in accuracy for the 19.5 million connection updates per epoch (19.5 MCUP) line when frames per parameter is greater than 20. Table 6.1 show word error rate results on the Eval2001 test set for stand-alone Tandem systems using posterior-based features from TMLPs of this constant 19.5 MCUP. Each of the TMLPs in the table have 40 hidden units per critical-band, and the SRI recognizer HMMs were trained using the same training set as in Chapter 5. The TMLP with 80 frames-to-parameters performed the best achieving a 43.9% WER on Eval2001. Lowering the frames-to-parameters ratio to 20, causes WER to go up to 44.1%, while lowering this ratio to 5 and 1.25 causes WER to go up to 45.8% and 48.1% respectively. From Figure 6.2 and Table 6.1, it is unclear where the optimal ratio of training frames-to-parameters lies. We cannot conclude as in [33] that the optimal range of training frames-to-parameters is between 10 and 40, but we can say that the range is likely to begin at 40. It is interesting to note that the systems with 40 or more

[2] N depends on the degree to which the training is done online or in batch mode. Our trainings are done in a bunch (or semi-batch) mode where the parameter updates happen once every 256 training frames.

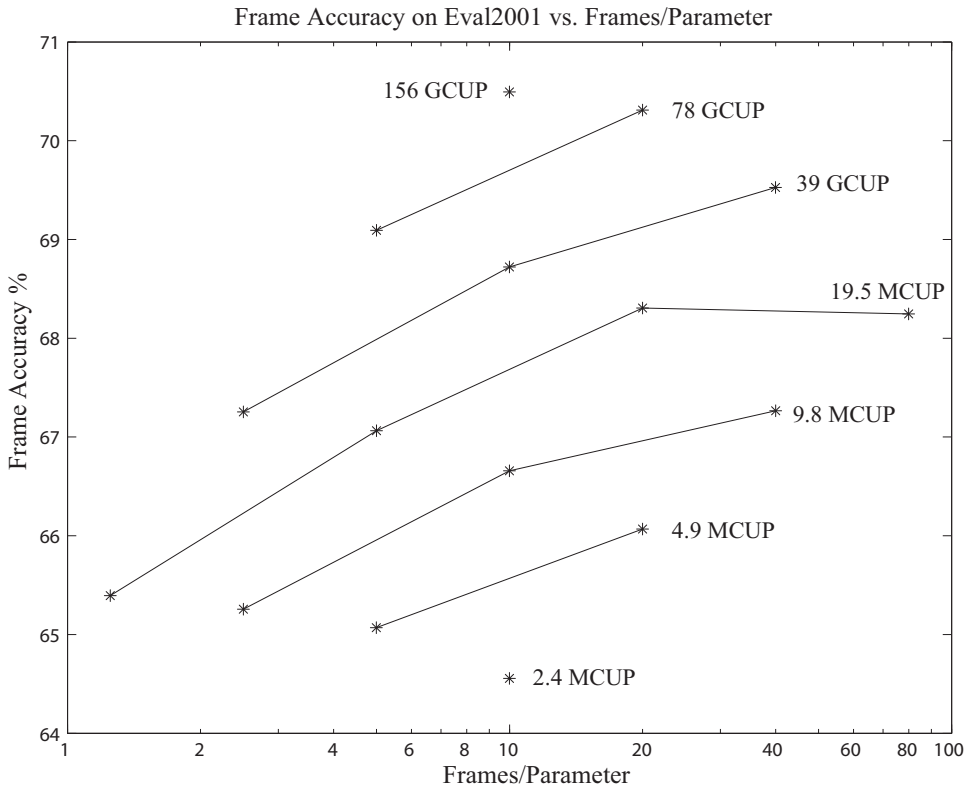

Figure 6.2: Frame accuracies on Eval2001 for TMLPs of equal training time.

frames per parameter (i.e., 124.9 hours/250k parameters, 124.9 hours/500k parameters, and 62.4 hours/250k parameters) have 30 or 40 as the best number of critical-band hidden units.

To summarize the findings from this section we make several concluding statements.

- Overall, the dominant conclusion that one can draw from these experiments is that the optimal number of critical-band hidden units is not all that sensitive to the amount of training data or the total number of parameters.

    1. For a fixed number of TMLP parameters, increasing the amount of training data does not lead to an increase in the optimal number of hidden units per critical-band. This is true even when increasing the amount of training data by almost 10-fold (15.6 hours versus 124.9 hours).
    2. For a fixed amount of training data, increasing the number of total parameters

	Frames-to-Parameters			
System Description	1.25 WER (%)	5 WER (%)	20 WER (%)	80 WER (%)
19.5 MCUP	48.1	45.8	44.1	43.9

Table 6.1: Word error rate results on Eval2001 for stand-alone Tandem systems using TMLPs of a constant training complexity (19.5 MCUP), 40 hidden units per critical-band, and varying training frames-to-parameters ratio. Even though the TMLPs were trained using different training set sizes, the SRI recognizer models were all trained using the training set used in Chapter 5.

leads to only a slight increase in the optimal number of hidden units per critical-band.

- For a fixed training time constraint, the optimal ratio of frames-to-parameters is greater than 40. Furthermore, TMLPs with ratios in this range have between 30 and 40 hidden units per critical-band.

6.2 Sharing Critical-Band Hidden Units

When looking at critical-band mean temporal patterns like the ones shown in [112, 62] and in Figure 2.5, we immediately notice that many temporal patterns are very similar within a particular critical-band and also among different critical-bands. In [50], Hermansky et al. developed a version of Neural TRAP called UTRAP, which used a single critical-band MLP for all critical-bands. They reasoned that since the critical-band temporal patterns are so similar even among temporal patterns from different critical-bands, then a single "universal" MLP could be used to extract the discriminant temporal information for all critical-bands. Besides reducing the amount of memory and computation requirements, another reason for developing UTRAP is that sharing this one "universal" MLP across all critical-bands in this way, offered potential for improving generalization by lessening the sensitivity to training and test set variations. Their experiments on a digit recognition task showed that UTRAP performed comparably to a Neural TRAP system [62].

Another interesting observation about the temporal information learned by Neural TRAP-like systems comes from examining the input-to-hidden weights of the critical-

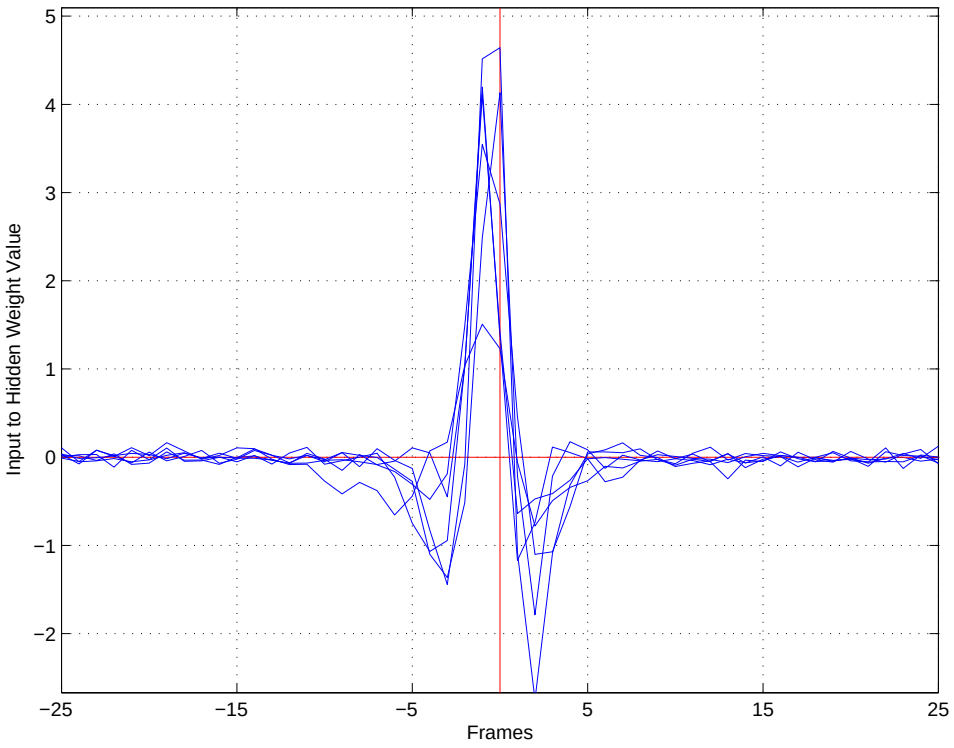

Figure 6.3: Input to hidden weights of various critical-band hidden units from a female HAT network trained on the female CTS training data in Chapter 5. These hidden units are gathered from different critical-bands.

band hidden units in HAT and TMLP. As described in Chapter 3, these input-to-hidden weights are critical-band matched filters acting on the log critical-band energy trajectories of speech. Each of these filters has a frequency response which tells us how the filter affects certain modulation frequencies. When looking at plots of these critical-band input-to-hidden weights for HAT and TMLP, we notice that many of these weights have similar shapes. Figures 6.3 and 6.4 show several input-to-hidden weights from hidden units at different critical-bands for HAT and TMLP respectively. Notice how similar they are. Appendices C and D contain many similar plots of input-to-hidden weights of critical-band hidden units for HAT and TMLP trained on TIMIT and CTS. Appendix E has corresponding temporal patterns learned by PCA and LDA on CTS.

The comparable performance of UTRAP to Neural TRAP and the similarity of the input-to-hidden weights of critical-band hidden units learned by HAT and TMLP suggest that discriminant temporal patterns can be shared across different critical-bands.

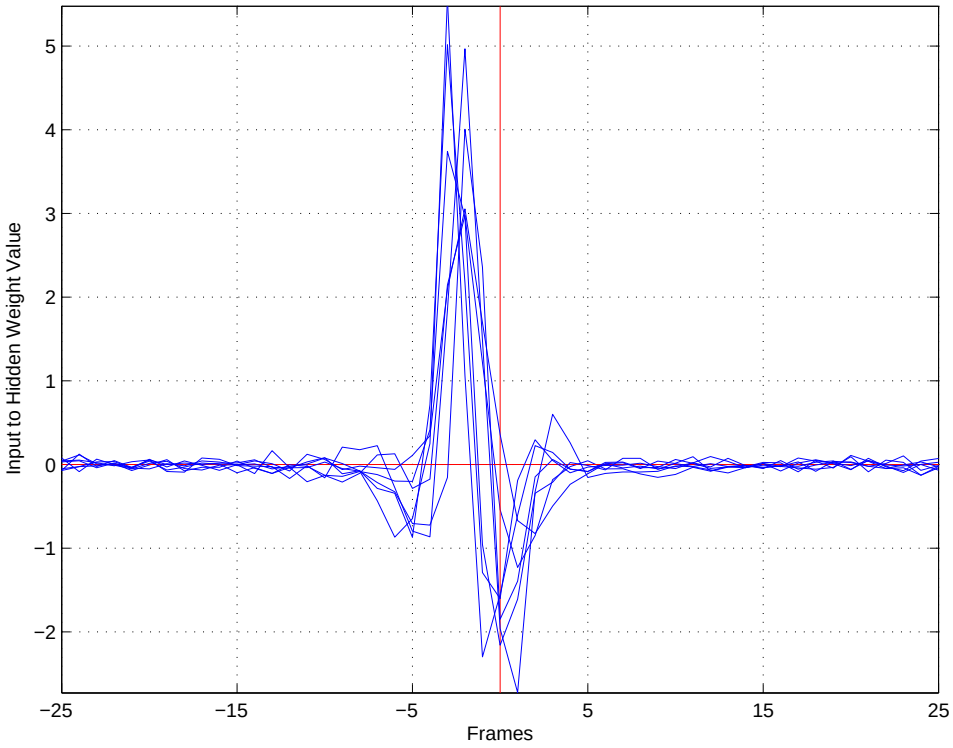

Figure 6.4: Input to hidden weights of various critical-band hidden units from a female TMLP network trained on the female CTS training data in Chapter 5. These hidden units are gathered from different critical-bands.

We further explore this suggestion in this section using our TMLP. The main idea is to share (or tie) critical-band hidden units across multiple critical-bands in the TMLP. This means that certain critical-band hidden units will have the same weights and biases but appear in different positions within the TMLP. For example, if we specify that hidden unit 4 of critical-band 8 be shared with hidden unit 15 of critical-band 5, then these two critical-band hidden units will have the same weights and biases. Training proceeds normally, but when the parameters of hidden unit 4 of critical-band 8 are updated, the parameters of hidden unit 15 of critical-band 5 are updated identically. In this way we effectively have identical critical-band discriminators that are trained and applied over multiple critical-bands.

In Chapter 5 we trained gender dependent TMLPs on a 68-hour CTS training set. In this section we train our gender dependent TMLPs that share critical-band hidden units on the same training set. One configuration for the sharing of critical-band hidden units is to share each critical-band hidden unit across all critical-bands. For example, if we choose to have 30 total hidden units per critical-band, this type of sharing means that each of the 30 hidden units appears in every one of the critical-bands. To make this clearer, hidden unit 1 of critical-band 1 shares parameters with hidden unit 1 of critical-bands 2-15. Similarly, hidden unit 2 of critical-band 1 shares parameters with hidden unit 2 of critical-bands 2-15, and so on and so forth. Figure 6.5 shows the frame accuracy performance on Eval2001 for TMLPs whose critical-band hidden units are shared across all critical-bands. Each of the TMLPs only differ by the total number of critical-band hidden units (each of which is shared across all critical-bands).

Frame accuracy performance starts to plateau after 25 critical-band hidden units. It is safe to assume that 40 shared critical-band hidden units are sufficient for achieving high frame accuracy. We compare recognition performance between a comparable non-weight sharing TMLP with 40 hidden units per critical-band (this is the same *TMLP* in Chapter 5) with the weight sharing TMLP with 40 shared critical-band hidden units (*TMLP S40*) in Table 6.2. We measure the performance in terms of frame accuracy and word error rates (WER) on Eval2001, and the WERs come from the 3 ASR system configurations tested in Chapter 5 (e.g., stand-alone Tandem, augmented Tandem, and combined-augmented Tandem). The performance of *TMLP S40* is always worse than *TMLP* except in the case of frame accuracy where *TMLP S40* gives a higher accuracy than *TMLP*. The WERs for

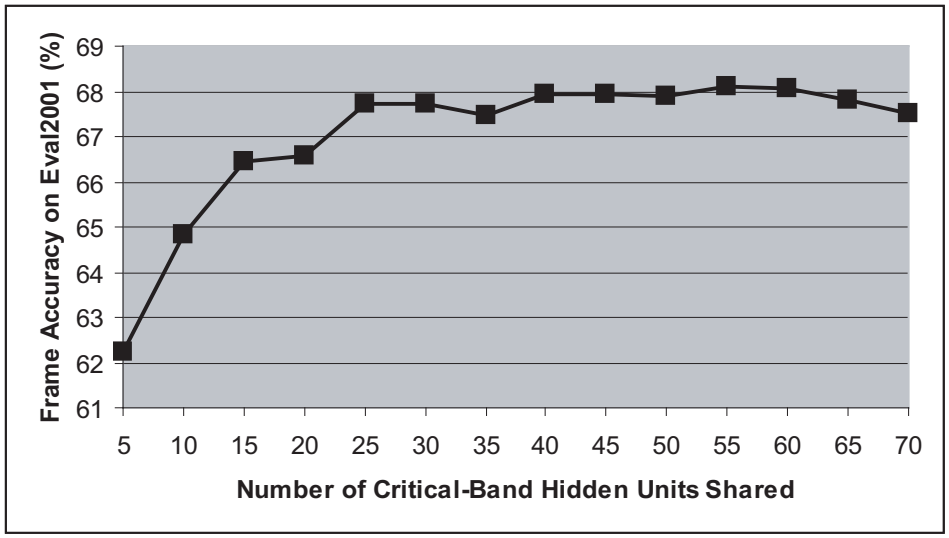

Figure 6.5: Frame accuracy on Eval2001 for TMLPs whose critical-band hidden units are shared across all critical-bands.

TMLP S40 are worse by .4%-.6% absolute which is a statistically significant margin.

It makes sense that the weight sharing *TMLP S40* would produce worse results than the non-weight sharing *TMLP* because weight sharing further constrains the model reducing the size of the family of distributions that *TMLP S40* can model. However, what is somewhat surprising is that the *TMLP S40* does so well. The margins in performance between *TMLP S40* and *TMLP* are not very large. Our motivation for exploring weight sharing in the TMLP came from observations that temporal patterns (either those from Mean TRAPs or from input-to-hidden weights of critical-band hidden units in HAT and TMLP) from different critical-bands look similar. Because *TMLP S40* performs comparably to *TMLP*, discriminant temporal patterns can indeed be shared by different critical-bands without incurring a large penalty in performance. This may be especially crucial in applications where the amount of memory and computation is limited (i.e., in mobile devices). The *TMLP S40* has about 30,000 fewer parameters than the *TMLP*. Moreover, the *TMLP S40* can potentially give better generalization performance in more mismatched training and testing conditions because of its more parsimonious representation.

System Description	Frames Correct (%)	Stand-Alone WER (%)	Augment WER (%)	Combine Augment WER (%)
TMLP	67.12	44.9	35.5	33.9
TMLP S40	67.92	45.5	35.9	34.3

Table 6.2: Performance of TMLPs with 40 hidden units per critical-band on Eval2001. *TMLP* does not have weight sharing, while *TMLP S40* shares all 40 hidden units over all critical-bands.

6.2.1 Narrow-Band Discriminant Temporal Patterns

In Section 5.6 we discussed the temporal patterns learned by HAT and TMLP trained on CTS data. In this subsection, we do the same for the weight sharing *TMLP S40*. Appendix D contains plots of the critical-band input-to-hidden unit weights of *TMLP S40* as well as corresponding modulation frequency responses.

The narrow-band discriminant temporal patterns learned by *TMLP S40* resemble the centroids of the patterns learned by *TMLP* in Chapter 5 and also displayed in Appendix D. Present are the ubiquitous onset "derivative" patterns, the energy detecting "Mexican hat" patterns, and other patterns that have also been learned by HAT and TMLP in previous chapters. What is important is that all of the modulation frequency responses pass speech modulations between 0 and 20 Hz. Again, these low modulation frequencies have been shown to be important for speech recognition.

Chapter 7

Conclusion

7.1 Summary

Conventional state-of-the-art speech recognition systems typically only extract information from speech within short-term spectral slices lasting about 25 milliseconds. Relying solely on short-term spectral slices for the modeling of speech, these speech recognition systems are vulnerable to variabilities in speech that do not affect human speech recognition performance. The work presented in this thesis further showed a novel way of modeling speech and integrated it within the framework of a state-of-the-art large vocabulary continuous speech recognizer. Instead of using just short-term spectral slices, the systems developed in this thesis extract information useful for automatic speech recognition (ASR) from long-term narrow-frequency bands of speech spanning about 500 milliseconds.

The motivation for extracting information within narrow-frequency bands comes mainly from human listening experiments that showed that human recognition performance remains quite high when given band-limited speech. Humans can also accurately detect certain characteristics of speech quite robustly from narrow-frequency bands of speech. The motivation for extracting long-term information comes from human listening experiments showing how humans rely on longer acoustic context for the accurate recognition of nonsense syllables. Moreover, information theoretic analyses of speech showed that significant amounts of discriminant information about the identity of a phone exist at times up to several hundred milliseconds before and after. Finally, it was our hope that by extracting speech information in this radically different way, our new long-term narrow-band (tem-

poral) systems would be able to complement the traditional systems leading to significant reductions in word error rates.

Prior to this work, Hermansky and Sharma developed a system that improved ASR performance by extracting information from long-term narrow-frequency bands. The Neural TRAP system [52] proved to be quite comparable with traditional ASR systems; however, in combination with traditional systems, Neural TRAP further reduced word error rates. Building off their success, we developed some Neural TRAP extensions that addressed one of the major issues in Neural TRAP: the choice of narrow-band information to extract. Neural TRAP uses critical-band level phone posteriors as the narrow-band information source. Multi-layer perceptrons (MLPs) are trained on critical-band level labels of phones to learn phone posterior probabilities from critical-band log energy trajectories of speech lasting about 500 milliseconds. Phone posteriors from all critical-bands are then used as inputs to a merger MLP that estimates the overall phone posterior probabilities. The problem here is that critical-band level phone posterior estimation is quite difficult because of the dearth of information for classifying phones within critical-band log energy trajectories. Because of these difficulties, we developed two new neural net architectures for extracting long-term narrow-band speech information: Hidden Activation TRAP (HAT) and Tonotopic Multi-Layer Perceptron (TMLP).

HAT was built on the premise that the mappings from the critical-band hidden unit space to the critical-band phone posteriors of critical-band MLPs in Neural TRAP were extraneous and inaccurate. Whatever useful information for discriminating between phones at the critical-band level is already captured by the input-to-hidden weights of the critical-band MLPs. These input-to-hidden weights act as matched filters on the input critical-band log energy trajectories, and they emphasize/deemphasize certain modulation frequencies of speech. Unlike Neural TRAP, HAT uses the critical-band hidden unit activations as the input to the merger MLP instead of the critical-band phone posteriors.

TMLP has the same network connections as HAT, but in TMLP the critical-band hidden unit connections are learned as a result of the global gradient descent error minimization training algorithm. Instead of constraining the critical-band hidden unit connections to learn what is best for critical-band level phone classification, TMLP critical-band hidden unit connections are set to whatever is best for the overall phone classification. Thus, the family of distributions that TMLP can model is greater than HAT.

In Chapter 3, we compared the performance of HAT, TMLP, and Neural TRAP systems on the TIMIT phone recognition task. We used the hybrid ANN/HMM ASR setup and found that HAT and TMLP outperform standard Neural TRAP in clean conditions while using 84% fewer parameters. We also compared these temporal systems with a more traditional system that used 9 frames of Perceptual Linear Predictive (PLP) features plus energy and deltas and double deltas as inputs to the MLP. The temporal systems performed comparably in clean conditions to this PLP system, but in reverberant conditions all temporal systems outperformed this PLP system. We also tested these systems in the presence of additive car and exhibition hall noise at various signal-to-noise ratios. No clear winner was declared from these tasks. The main finding which supported earlier findings on Neural TRAP was that in combination with the PLP system, HAT and TMLP significantly improved performance. Because HAT and TMLP performed very well compared to Neural TRAP we concluded that is was good to skip the mapping to critical-band phones. Because TMLP did not significantly outperform HAT, there was not yet a clear advantage from further unconstraining the learning of critical-band hidden unit weights in TMLP; however, TMLP did have the practical advantage of not having to use large amounts of temporary disk space to store all of the critical-band hidden unit activations. This practical advantage was especially clear when we worked on training sets with much more data.

In Chapter 4, we integrated the Neural TRAP system with a state-of-the-art recognizer for conversational telephone speech (CTS). In particular, we combined the phone posteriors estimated by Neural TRAP with the phone posteriors from a 9 frame PLP MLP and transformed the combined phone posteriors into front-end features. These features were then concatenated with conventional PLP features, resulting in an augmented feature vector that captured speech information from multiple time scales. We tested this setup over a series of increasingly complex recognition tasks (numbers, 500 most commonly used words from Switchboard, and full vocabulary CTS) and found that this approach consistently reduced recognition errors. We showed that the simple posterior combination methods tested (e.g., averaging the posteriors, averaging the log posteriors, and inverse entropy weighted averaging of the posteriors) all performed roughly the same, but the inverse entropy weighted combination method demonstrated some robustness to catastrophic errors within a single posterior stream. We also cited the importance of tuning the Gaussian

Weight parameter[1] to reduce the importance of picking the optimal number of dimensions to keep from the posterior stream. Concatenating the combined posterior features with the conventional PLP features led to a larger dimensional front-end feature vector which had to be compensated for by adjusting the Gaussian Weight.

After successfully integrating Neural TRAP within a state-of-the-art recognizer, we proceeded to compare various approaches for the extraction of useful information from long-term contexts for recognizing CTS in a variety of ASR system configurations. The three main types of long-term or temporal systems were:

1. Totally unconstrained - These systems simply took the 15 bands by 51 frames of log energies as inputs. *15 x 51 MLP3* used a single hidden layer MLP, while *15 x 51 MLP4* used a double hidden layer MLP.

2. Band-constrained linear - These systems calculated linear transforms on the log critical-band energy trajectories. *PCA40* used principal components analysis to project the input trajectories along directions corresponding to the top forty dimensions. *LDA40* used linear discriminant analysis to transform the input trajectories along the top forty most discriminant directions.

3. Band-constrained nonlinear - These systems used some form of critical-band MLP to extract information from the input critical-band trajectories. *HAT Before Sigmoid*, *HAT*, *Neural TRAP*, *Neural TRAP Post Softmax* used outputs from various points within critical-band MLPs trained to learn critical-band level phone posteriors. *TMLP* was like *HAT* except that the critical-band hidden unit connections were learned to optimize the overall phone posterior estimate.

The three types of ASR system configurations for the comparison tests were:

1. Stand-Alone Tandem - The phone posterior outputs of the temporal systems were transformed and used as the front-end features for a conventional Gaussian mixtures-based HMM recognizer.

2. Augmented Tandem - The phone posterior outputs of the temporal systems were transformed and concatenated with conventional short-term front-end features. The

[1] Recall that this is a specific weighting factor found in the SRI recognizer.

resulting feature vector was then used as the front-end feature vector for a conventional Gaussian mixtures-based HMM recognizer.

3. Combined-Augmented Tandem - The phone posterior outputs of the temporal systems were combined with the phone posterior outputs from an MLP whose inputs were 9 frames of PLP features (*9 Frame PLP MLP*). These were transformed and concatenated with conventional short-term front-end features. The resulting feature vector was then used as the front-end feature vector for a conventional Gaussian mixtures-based HMM recognizer.

We found that *15 x 51 MLP4, HAT* and *TMLP* consistently outperformed all other temporal systems in all ASR system configurations. The band-constrained *HAT* and *TMLP* systems performed better in the combined-augmented Tandem configuration than the unconstrained *15 x 51 MLP* suggesting that the critical-band constraint found in *HAT* and *TMLP* are more helpful for learning complementary information to the *9 Frame PLP MLP*. *HAT* and *TMLP* outperformed band-constrained linear temporal systems, suggesting that probabilities of certain critical-band categories are important for higher recognition performance. *HAT* and *TMLP* outperformed other band-constrained nonlinear temporal systems, suggesting that phone posteriors at the critical-band level are not the optimal critical-band level information to extract. Rather, it is the information captured by the critical-band hidden units (i.e., the matched temporal filters) that is best for the classification of phones.

Toward the end of Chapter 5, we investigated what phone categories the temporal systems consistently performed better on compared with the intermediate-term *9 Frame PLP MLP*. *15 x 51 MLP4, HAT* and *TMLP* consistently classified diphthongs, filled pauses, laughter, and a few other phones (/ae/, /hh/, /th/, /z/, /ax/, and /dx/) better than *9 Frame PLP MLP*. The very best ASR system developed, the *TMLP* in combined-augmented Tandem configuration, achieved an impressive 8.9% relative reduction in word error rate on CTS compared with only using the short-term state-of-the-art front-end feature vector alone. The scale of this relative reduction in word error rate also carried over when using the full state-of-the-art speech recognition system on the CTS evaluations in 2004 [135].

In Chapter 6 we further explored the settings for TMLP. We found that the

optimal number of critical-band hidden units in TMLP does not increase with more training data. The optimal ratio of training frames to trainable parameters in the TMLP was greater than 40. Finally, we showed that since many critical-band matched filters learned by TMLP and HAT looked similar across different critical-bands, it was possible to maintain comparable performance by sharing the critical-band hidden units across all critical-bands in the TMLP, thereby reducing the total number of parameters by 30,000.

As mentioned in Chapters 3, 5, and 6, the temporal patterns learned by HAT and TMLP systems as well as PCA and LDA systems are displayed in Appendices C, D, and E. Almost all of the patterns learned by HAT and TMLP systems preserve the low modulation frequencies of speech (0 to between 16 and 20 Hz) which are important for speech recognition. The patterns learned by PCA and LDA also pass higher modulation frequencies. Also, patterns learned by PCA resemble sinusoidal basis functions.

7.2 Contribution

The work in this thesis further developed the techniques of extracting information from speech over long time spans within narrow-frequency channels. Previously, all such approaches (the original Neural TRAP and its variants) were designed and tested only on smaller tasks of limited complexity like numbers, digits, and read speech. One of the major contributions of this thesis was to integrate these long-term approaches within the framework of a state-of-the-art large vocabulary continuous speech recognizer for the recognition of conversational telephone speech. We have also developed two new Neural TRAP-like classifiers that outperform Neural TRAP and use fewer parameters. Using HAT and TMLP, we were able to achieve significant word error rate reductions on the challenging task of recognizing conversational telephone speech. In fact, combined-augmented Tandem features derived with HAT were used in SRI's state-of-the-art 2004 recognition system [135]. With these features, system performance was improved by about 10% relative compared to a system without HAT-based features.

In addition to reducing word error rates on a challenging ASR task, we have gained some understanding from comparing various methods of extracting information from long-term narrow-band speech. We have seen in many cases that extracting information in this way leads to systems that combine well with more traditional methods that

extract information from shorter time contexts over the entire spectrum. By comparing various temporal systems, we learned that it is important to extract probabilities of certain sub-phonemic categories of speech from the long-term energy trajectories. These categories correspond to temporal patterns that are useful in discriminating between speech sounds. Using phone posteriors at the critical-band level was consistently worse than using probabilities of these temporal patterns. This work also examined what phones are better classified by temporal systems.

Finally, this work began exploring the reuse of certain discriminant critical-band temporal patterns for ASR. By sharing all critical-band hidden units in TMLP across all critical-bands, we were able to show that discriminant temporal patterns can indeed be trained and applied on different critical-bands without a gross reduction in performance. Further studies are required, however, to determine which specific patterns can be shared across which critical-bands. The discriminant temporal patterns learned in this thesis further support previous studies on the importance of modulation frequencies between 0-20 Hz for ASR. The patterns learned by TMLP and HAT (displayed in Appendices C and D) mostly have modulation frequency responses that emphasize these important frequencies.

7.3 Future Work

The work on HAT and TMLP has shown the basic effectiveness of using critical-band hidden units to derive discriminant temporal filters. Throughout, we have been using a constant number of hidden units per critical-band. It is likely that improvements in performance as well as reductions in total parameters can be achieved by customizing each critical-band with its own optimal number of hidden units. For example, high frequency critical-bands probably do not need as many hidden units as critical-bands around 500 Hz where a lot of phonetic information exists.

Further reductions in model size can also be achieved by exploring more weight sharing schemes in TMLP. We tried the simplest scheme of sharing all critical-band hidden units across all critical-bands. Some of the filters learned by the hidden units may not be useful for certain bands. It is also likely that only some critical-bands share certain discriminant temporal filters. For example, adjacent critical-bands are more likely to contain similar discriminant temporal filters than critical-bands separated by 2,000 Hz. An

exhaustive study of various sharing schemes would be able to discover which bands share which kind of temporal filters. Another useful byproduct of such a study would be that the learned discriminant temporal filters could be fixed and reused over and over again as a part of a preprocessing step for front-end feature extraction. This is becoming more attractive each year as we continue to gain access to more training data, which requires longer training times for our methods.

All of the comparisons in Chapter 5 were tested on CTS, which is a very difficult task but has relatively little noise coming from outside sources like cars, sirens, fans, other people, etc., so given a lot of training data, narrow-band constraints may make less of a difference than you might see in other tasks. Therefore, it would be a great interest to repeat some of the comparisons between the unconstrained temporal systems and the narrow-band constrained temporal systems in Chapter 5 on a large vocabulary continuous speech task containing more naturally occurring noises (e.g., recordings of meetings). It is likely that the narrow-band constraints will show more of a benefit on such a task.

Finally, in all of the HAT and TMLP experiments in this thesis, we used log critical-band speech energy trajectories lasting 51 frames or about 500 milliseconds. Further explorations of HAT and TMLP by varying the input time context as well as widening the frequency band (i.e., using more than one critical-band) of the inputs to the band specific hidden units may lead to additional performance improvements. HAT and TMLP classifiers of varying time context and bandwidth can offer separate and complementary snapshots of the speech signal leading to increased robustness. The ASR system framework is already in place because we can combine any number of HAT and TMLP classifiers using the simple posterior combination techniques explored in Chapter 4. Once combined, these MLP-based features can augment the conventional short-term features offering an almost limitless number of different snapshots extracting the redundant information found in speech.

Appendix A

Critical-Band Cuttoff Frequencies for TIMIT

This appendix lists for reference the half power cut-off frequencies for the critical-band filters on the TIMIT database which is sampled at 16 kHz.

Critical-Band	Frequency Range (Hz)
1	18-163
2	118-267
3	220-379
4	329-502
5	446-637
6	575-790
7	720-965
8	885-1165
9	1073-1397
10	1290-1667
11	1542-1982
12	1836-2350
13	2180-2782
14	2582-3289
15	3055-3885
16	3609-4587
17	4262-5412
18	5030-6383
19	5933-7527

Table A.1: The half power cut-off frequencies of each critical-band for speech data sampled at 16 kHz.

Appendix B

Critical-Band Cuttoff Frequencies for CTS

This appendix lists for reference the half power cut-off frequencies for the critical-band filters on the CTS data which is sampled at 8 kHz.

Critical-Band	Frequency Range (Hz)
1	17-161
2	115-265
3	216-375
4	323-495
5	439-629
6	565-779
7	707-949
8	868-1144
9	1051-1370
10	1262-1632
11	1506-1937
12	1790-2293
13	2122-2709
14	2509-3197
15	2963-3769

Table B.1: The half power cut-off frequencies of each critical-band for speech data sampled at 8 kHz.

Appendix C

HAT and TMLP Critical-Band Patterns for TIMIT

In Appendices C, D, and E, we display plots of critical-band temporal patterns that our methods have learned. In the HAT and TMLP networks, these patterns come from the input-to-hidden weights of critical-band hidden units. As described in Chapter 3, these input-to-hidden weights are matched filters acting on the long-term, narrow-frequency log energy input trajectories of speech. As such, each filter has a corresponding modulation frequency response. For speech recognition, it has been shown that modulation frequencies between 0-16 Hz are important (see Chapter 2 for a detailed discussion about modulation frequencies and speech recognition).

In this appendix, we display pictures of critical-band discriminant temporal patterns learned by the HAT and TMLP networks from Chapter 3 trained on TIMIT data. There are a total of 380 discriminant temporal patterns (19 critical-bands times 20 hidden units per critical-band), which is too many to plot. Since many of these discriminant temporal patterns look similar, we have clustered all of them using agglomerative clustering with the correlation based similarity measure described in Chapter 2 (Eq.2.5). We stop clustering at 20 clusters and average all patterns belonging to a particular cluster. We call this average pattern a centroid, and we display the tables showing which critical-bands contain hidden unit patterns that make up a particular centroid in Table C.1 for HAT and Table C.2 for TMLP. We also plot the centroid patterns with their corresponding modu-

Centroid	Critical-Band(s)
Centroid 1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 2	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 3	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 5	19
Centroid 6	6, 11, 15, 17
Centroid 7	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19
Centroid 8	2, 17, 19
Centroid 9	2
Centroid 10	3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19
Centroid 11	12, 14
Centroid 12	4, 5, 10, 11, 14
Centroid 13	1, 2, 3, 4, 5, 7, 12, 13, 14, 15, 16, 17, 18
Centroid 14	3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19
Centroid 15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 16	16, 17
Centroid 17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 18	1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 13, 19
Centroid 19	1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18
Centroid 20	19

Table C.1: Centroid composition table for critical-band hidden units of HAT trained on TIMIT. The originating critical-bands of all the hidden units clustered within a particular centroid are listed.

lation frequency responses in Figures C.1 and C.2 for HAT and in Figures C.3 and C.3 for TMLP.

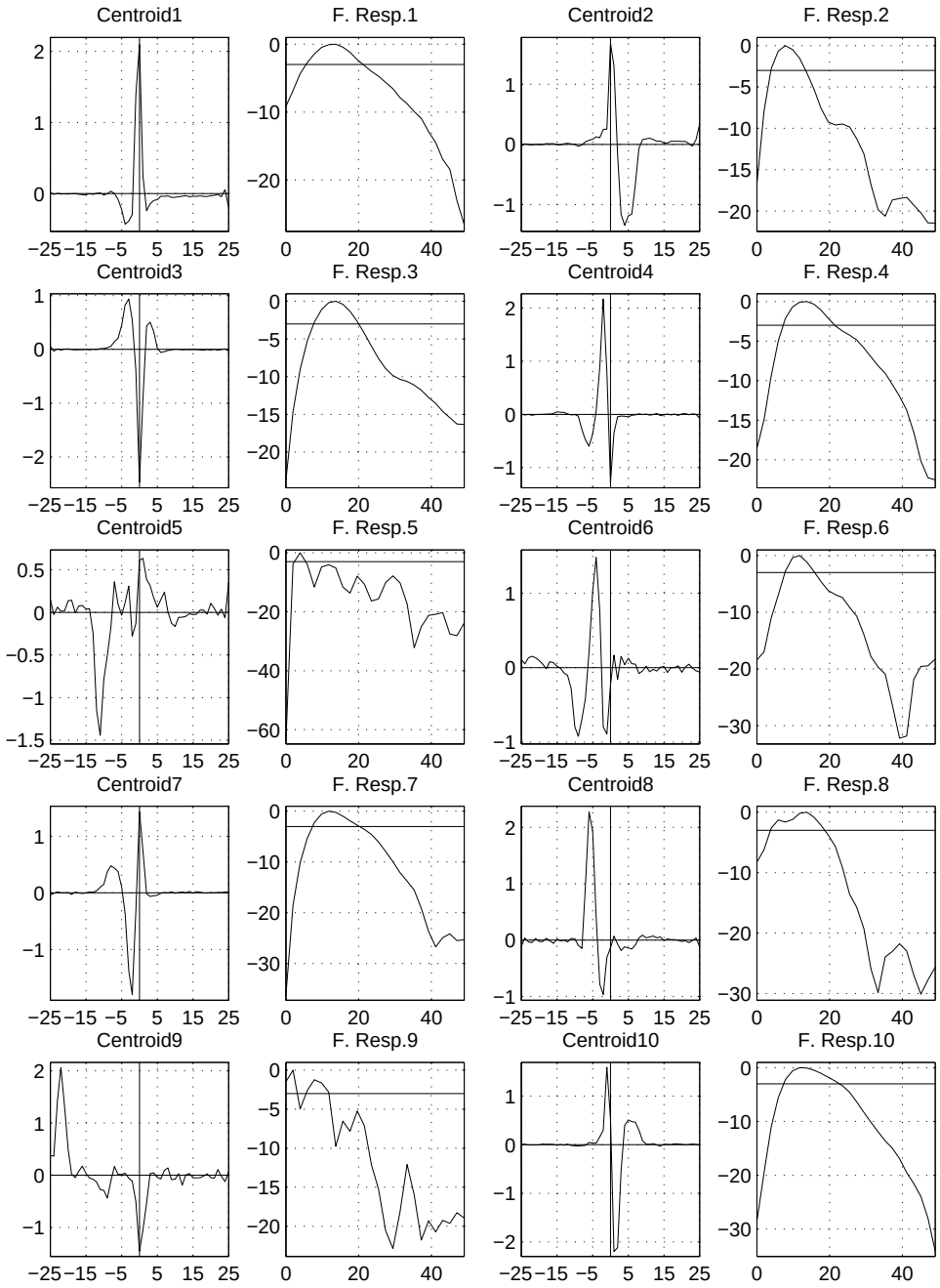

Figure C.1: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from HAT trained on TIMIT (Centroids 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

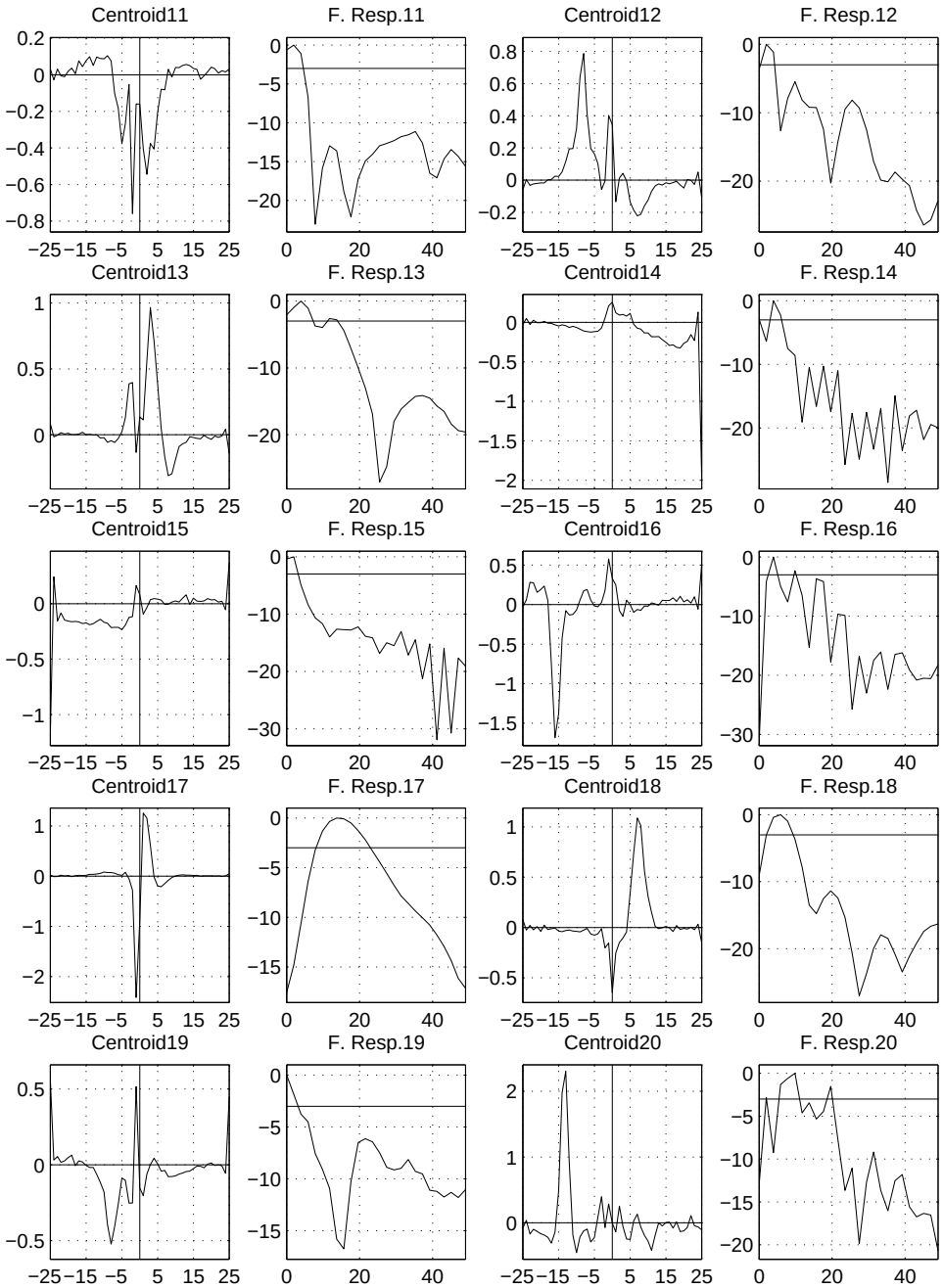

Figure C.2: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from HAT trained on TIMIT (Centroids 11-20). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

Centroid	Critical-Band(s)
Centroid 1	1, 2, 3, 5, 7, 10, 11, 12, 14, 15, 18, 19
Centroid 2	1, 5, 7, 9, 10, 12, 13, 17, 18
Centroid 3	2, 4, 5, 6, 7, 10, 14, 15, 17, 19
Centroid 4	2, 3, 4, 7, 8, 10, 12, 13, 14, 15, 16, 17, 19
Centroid 5	5, 16, 17
Centroid 6	2, 4, 7, 8, 10, 13, 14
Centroid 7	1, 4, 5, 6, 12, 13, 16, 18
Centroid 8	4, 17
Centroid 9	2, 3, 5, 6, 7, 10, 11, 16, 18
Centroid 10	1, 2, 3, 4, 8, 10, 11, 12, 13, 14, 15, 18, 19
Centroid 11	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 12	1, 2, 3, 4, 7, 8, 12, 13, 14, 15, 16, 18, 19
Centroid 13	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 14	3, 6, 10, 11, 13, 15
Centroid 15	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Centroid 16	1, 5, 6, 9, 10, 11, 13, 14, 15, 16, 17
Centroid 17	1, 4, 6, 7, 9, 11, 12, 13, 14, 16, 17, 19
Centroid 18	2, 3, 6, 7, 8, 9, 11, 12, 13, 16, 18, 19
Centroid 19	3, 4, 6, 7, 8, 9, 19
Centroid 20	3, 4, 5, 6, 9, 11, 12, 13, 16, 17, 19

Table C.2: Centroid composition table for critical-band hidden units of TMLP trained on TIMIT. The originating critical-bands of all the hidden units clustered within a particular centroid are listed.

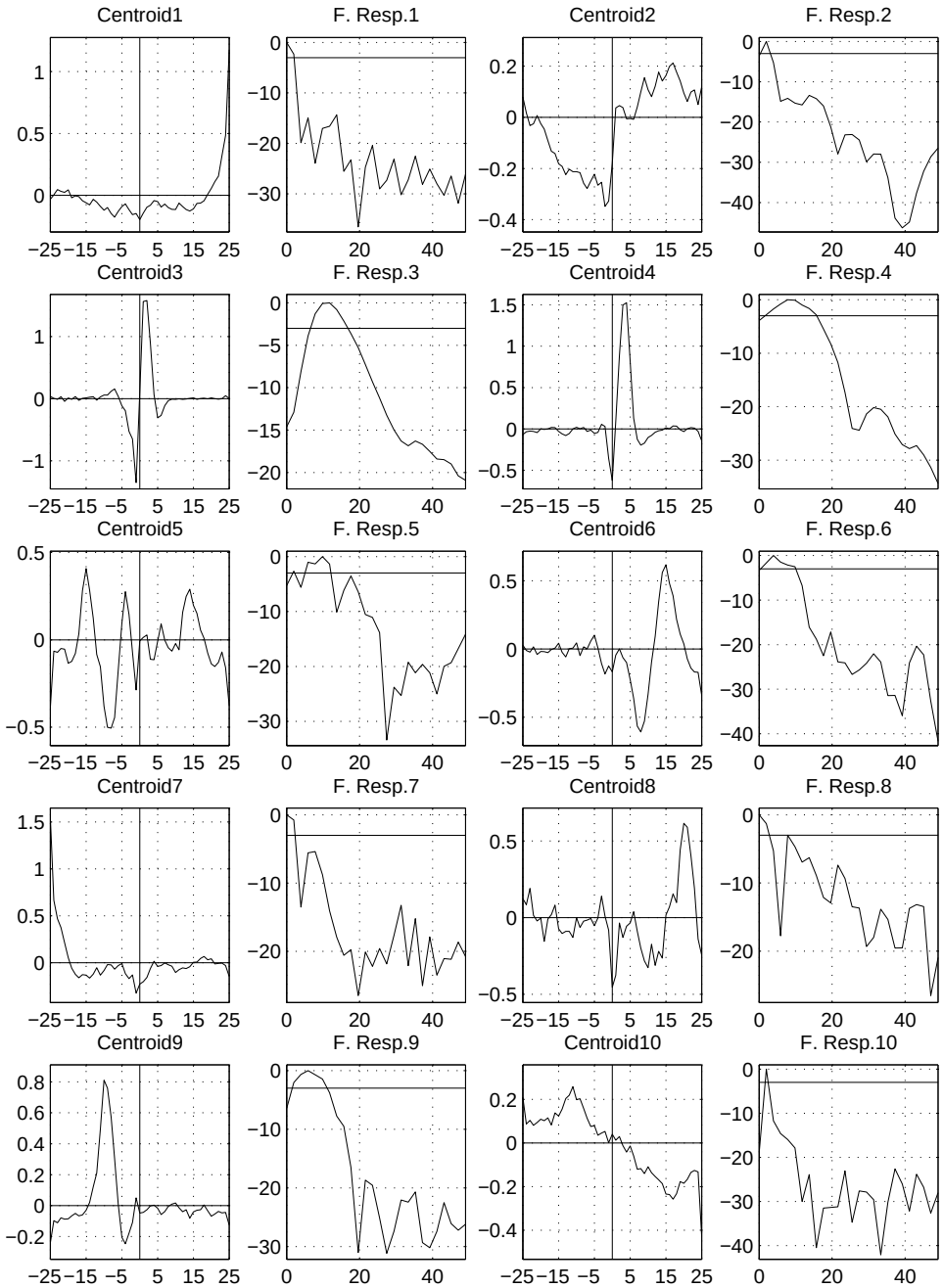

Figure C.3: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from TMLP trained on TIMIT (Centroids 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

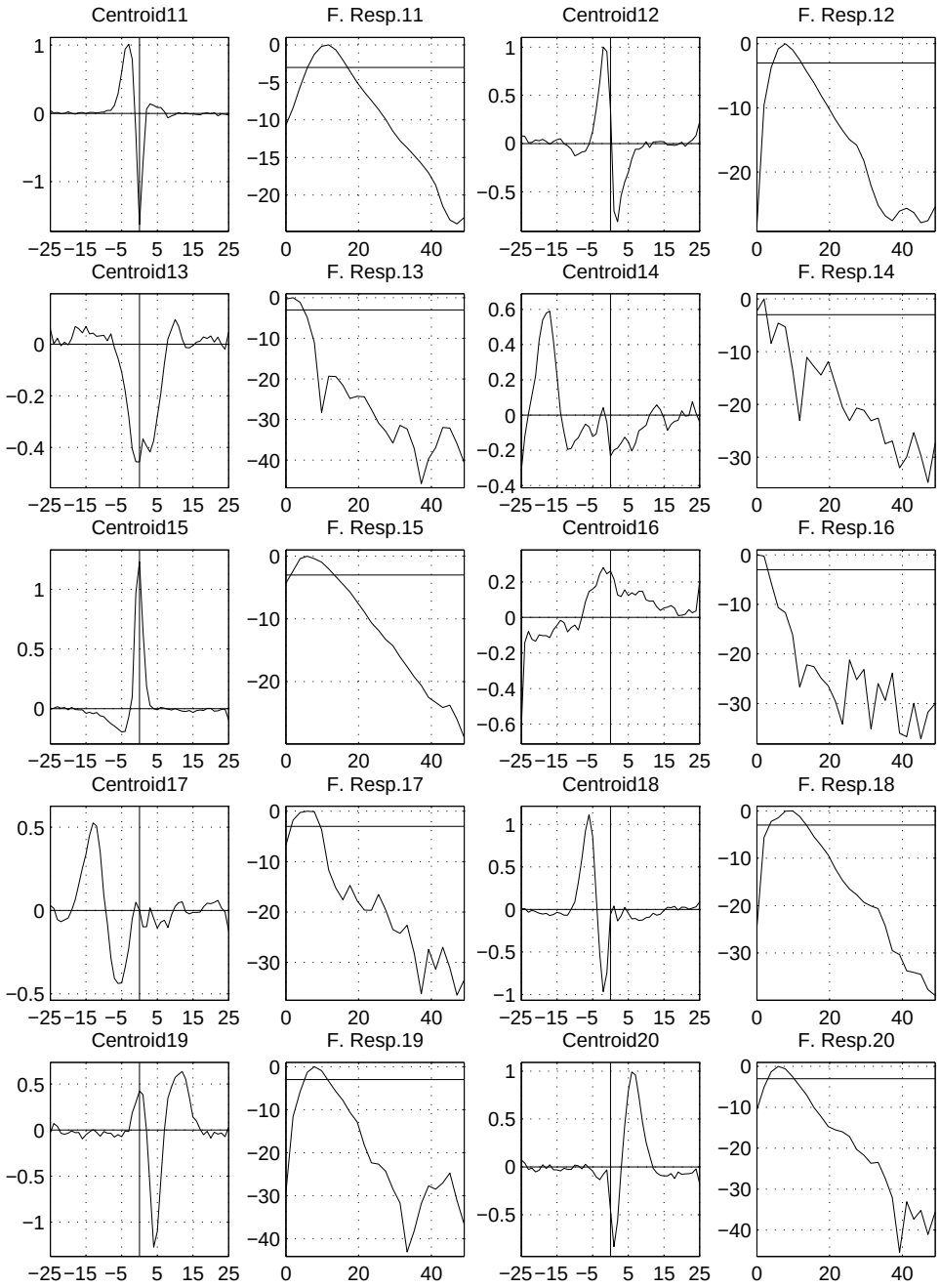

Figure C.4: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from TMLP trained on TIMIT (Centroids 11-20).The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

Appendix D

HAT and TMLP Critical-Band Patterns for CTS

In this appendix, we display pictures of critical-band discriminant temporal patterns learned by HAT and TMLP networks from Chapter 5 trained on 34 hours of female CTS data. These patterns are found in the input-to-hidden unit weights of the critical-band hidden units. There are a total of 600 discriminant temporal patterns (15 critical-bands times 40 hidden units per critical-band), which is too many to plot. Since many of these discriminant temporal patterns look similar, we have clustered all of them using agglomerative clustering with the correlation based similarity measure described in Chapter 2 (Eq.2.5). We stop clustering at 40 clusters and average all patterns belonging to a particular cluster. We call this average pattern a centroid, and we display the tables showing which critical-bands contain hidden unit patterns that make up a particular centroid in Tables D.1 and D.2 for HAT and Tables D.3 and D.4 for TMLP. We also plot the centroid patterns with their corresponding modulation frequency responses in Figures D.1, D.2, D.3, and D.4 for HAT and in Figures D.5, D.6, D.7, and D.8 for TMLP.

In addition to the HAT and TMLP networks trained on female CTS data from Chapter 5, we also display plots from the weight-sharing *TMLP S40* in Chapter 6. There are a total of 40 shared critical-band hidden units for *TMLP S40*. We plot the input-to-hidden weights of these 40 shared critical-band hidden units (discriminant critical-band matched filters) as well as their corresponding modulation frequency responses in Fig-

Centroid	Critical-Band(s)
Centroid 1	2, 6
Centroid 2	1, 3, 4, 5, 10, 12, 13, 14, 15
Centroid 3	8, 9, 14
Centroid 4	1, 2, 3
Centroid 5	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 6	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15
Centroid 7	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 8	4, 5, 6, 7, 8, 9, 10, 11, 12, 14
Centroid 9	1, 2, 3, 4, 13, 15
Centroid 10	1, 6
Centroid 11	1, 3, 4, 8, 9, 10, 11, 12, 14, 15
Centroid 12	1, 2, 5
Centroid 13	4, 6, 10, 11, 13, 15
Centroid 14	3
Centroid 15	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15
Centroid 16	3, 4, 5, 6, 7, 8, 9, 10, 15
Centroid 17	1, 2, 8, 11, 12, 13, 14
Centroid 18	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 19	12
Centroid 20	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Table D.1: Centroid composition table (Centroids 1-20) for critical-band hidden units of HAT trained on 34 hours of female CTS. The originating critical-bands of all the hidden units clustered within a particular centroid are listed.

ures D.9, D.10, D.11, and D.12.

Centroid	Critical-Band(s)
Centroid 21	3, 6, 7, 8, 9, 10, 12, 13, 14, 15
Centroid 22	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 23	6, 8, 9, 11, 12, 14, 15
Centroid 24	7, 8, 9, 12, 14
Centroid 25	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 26	1, 2, 3, 4, 5
Centroid 27	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 28	7
Centroid 29	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 30	1, 2, 3, 4, 5, 6, 11, 13, 14, 15
Centroid 31	2, 3, 6, 7, 9, 10, 11, 12, 14, 15
Centroid 32	1, 2, 4, 5, 11, 12, 13, 15
Centroid 33	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 34	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 35	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14
Centroid 36	2, 5, 6, 7, 13, 15
Centroid 37	3, 8, 9, 10, 11, 14
Centroid 38	3, 4, 5, 6, 7, 8, 9, 10, 15
Centroid 39	2, 3, 6, 8, 9, 10, 11, 12, 13, 14
Centroid 40	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Table D.2: Centroid composition table (Centroids 21-40) for critical-band hidden units of HAT trained on 34 hours of female CTS. The originating critical-bands of all the hidden units clustered within a particular centroid are listed.

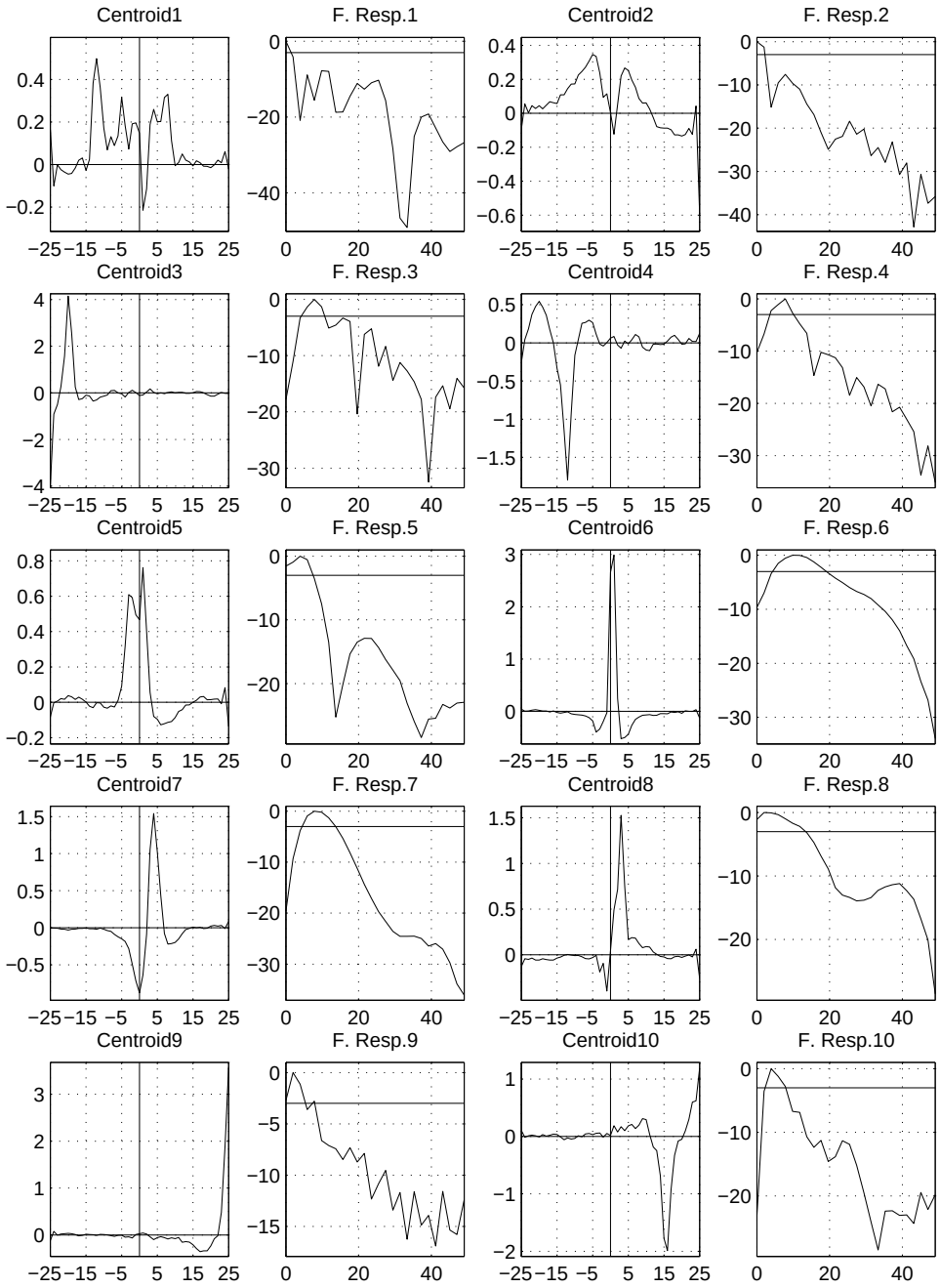

Figure D.1: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from HAT trained on 34 hours of female CTS (Centroids 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

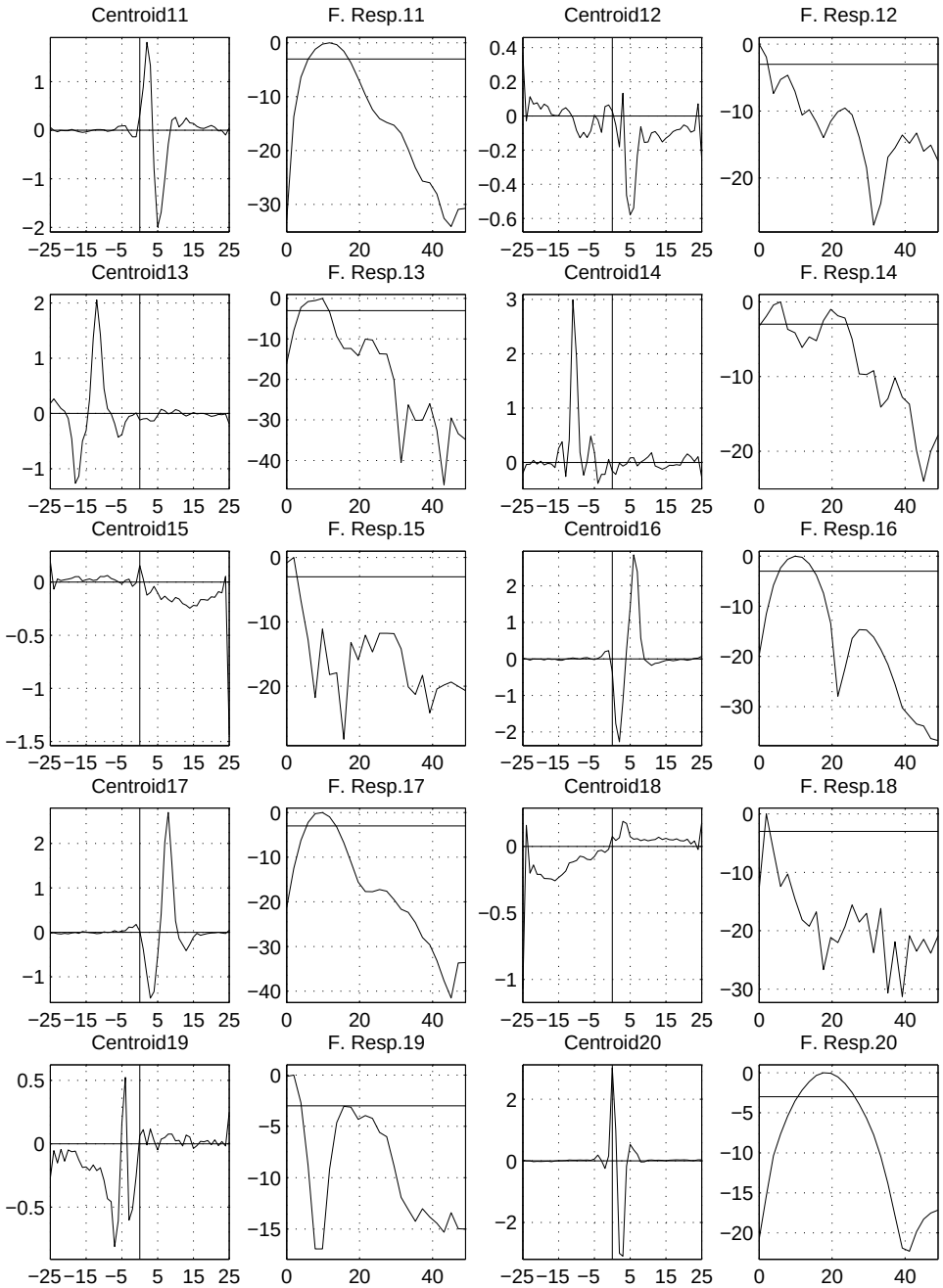

Figure D.2: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from HAT trained on 34 hours of female CTS (Centroids 11-20). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

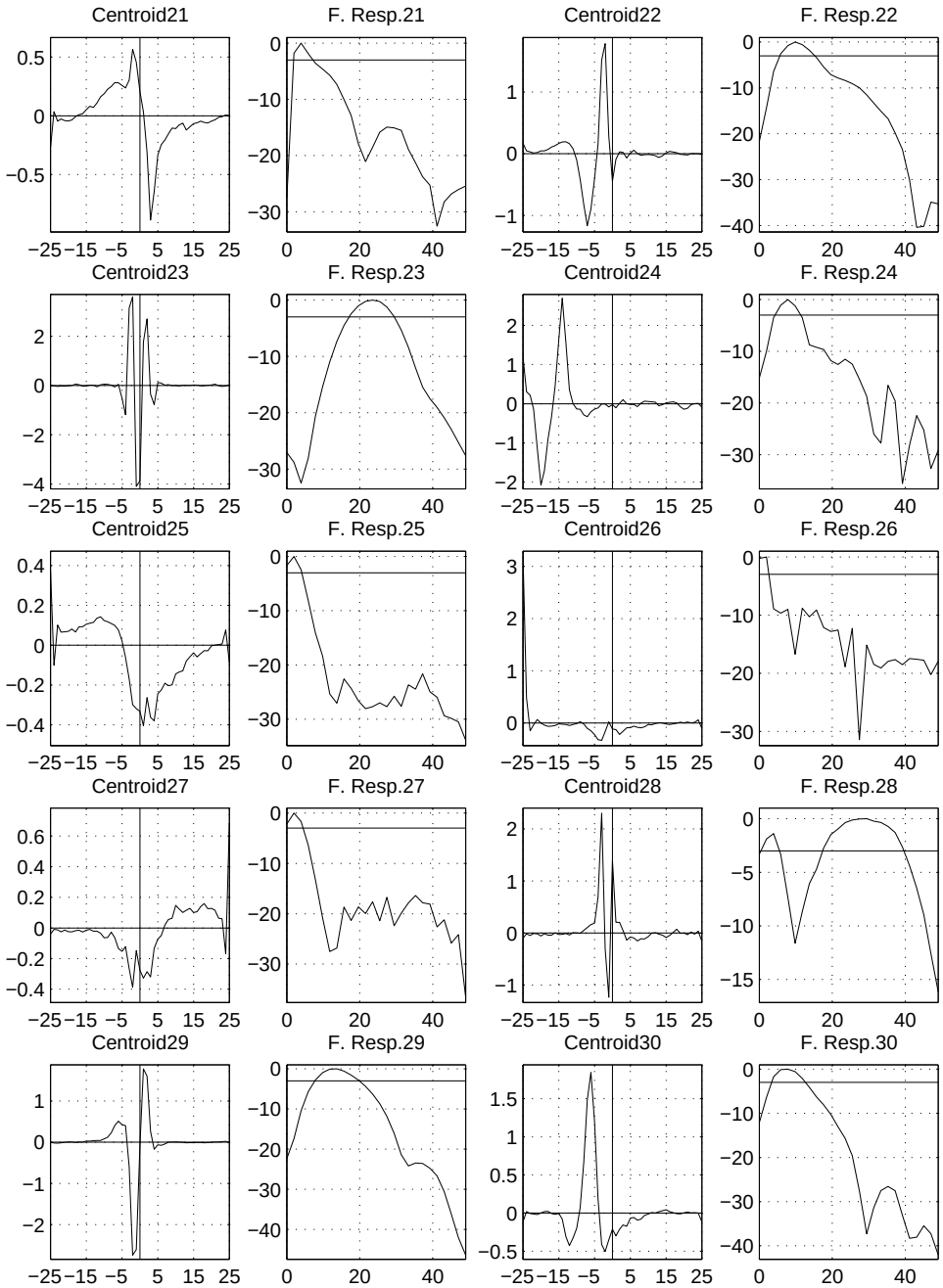

Figure D.3: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from HAT trained on 34 hours of female CTS (Centroids 21-30). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

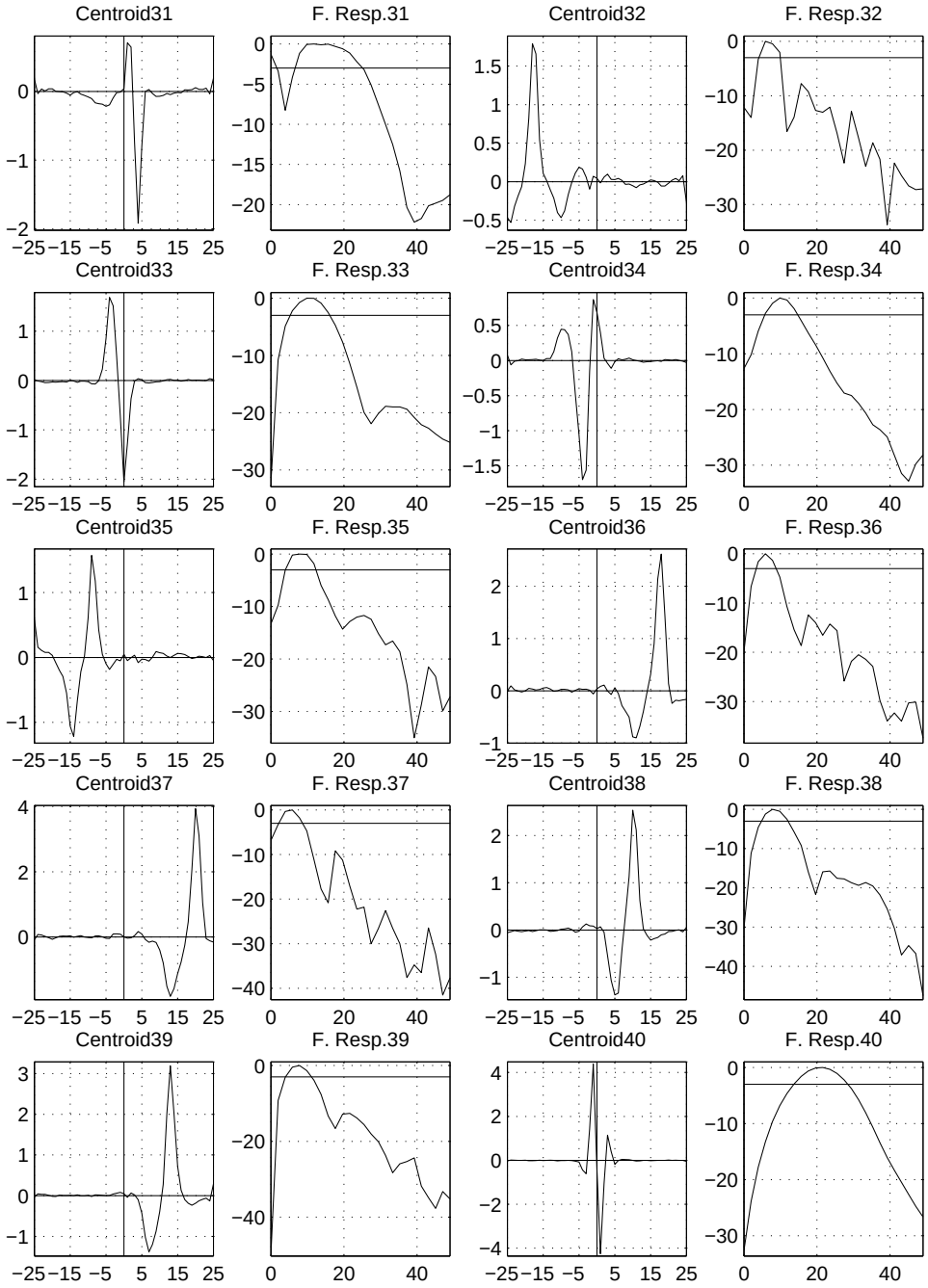

Figure D.4: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from HAT trained on 34 hours of female CTS (Centroids 31-40). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

Centroid	Critical-Band(s)
Centroid 1	2, 4, 5, 6, 7, 9, 11, 12, 13, 15
Centroid 2	1, 2, 3, 6, 7, 8, 10, 12, 13
Centroid 3	1, 3, 4, 5, 12, 13, 14
Centroid 4	1, 2, 7
Centroid 5	1, 2, 3, 4, 6, 7, 13
Centroid 6	8
Centroid 7	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15
Centroid 8	3, 4
Centroid 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 10	1, 5, 6, 8, 10, 11, 12, 13, 15
Centroid 11	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15
Centroid 12	4, 11, 15
Centroid 13	1, 3, 6, 7, 8, 9, 11, 15
Centroid 14	1, 3
Centroid 15	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 16	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15
Centroid 17	1, 3, 6, 9, 12, 14
Centroid 18	1, 2, 4, 7, 8, 10, 12, 14
Centroid 19	2, 3, 6, 7, 14, 15
Centroid 20	2, 3, 8, 12

Table D.3: Centroid composition table (Centroids 1-20) for critical-band hidden units of TMLP trained on 34 hours of female CTS. The originating critical-bands of all the hidden units clustered within a particular centroid are listed.

Centroid	Critical-Band(s)
Centroid 21	2, 5, 10
Centroid 22	1, 6, 7, 9, 10, 11, 13, 14, 15
Centroid 23	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 24	3, 4, 6, 13, 15
Centroid 25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Centroid 26	4, 6, 15
Centroid 27	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 28	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15
Centroid 29	2, 5, 8, 9, 14, 15
Centroid 30	1, 2, 4, 5, 6, 7, 8, 10, 11, 12, 14, 15
Centroid 31	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 32	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 33	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15
Centroid 34	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Centroid 35	1, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15
Centroid 36	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15
Centroid 37	14, 15
Centroid 38	4, 8, 9, 14
Centroid 39	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15
Centroid 40	2, 3, 4, 6, 7, 10, 12, 14

Table D.4: Centroid composition table (Centroids 21-40) for critical-band hidden units of TMLP trained on 34 hours of female CTS. The originating critical-bands of all the hidden units clustered within a particular centroid are listed.

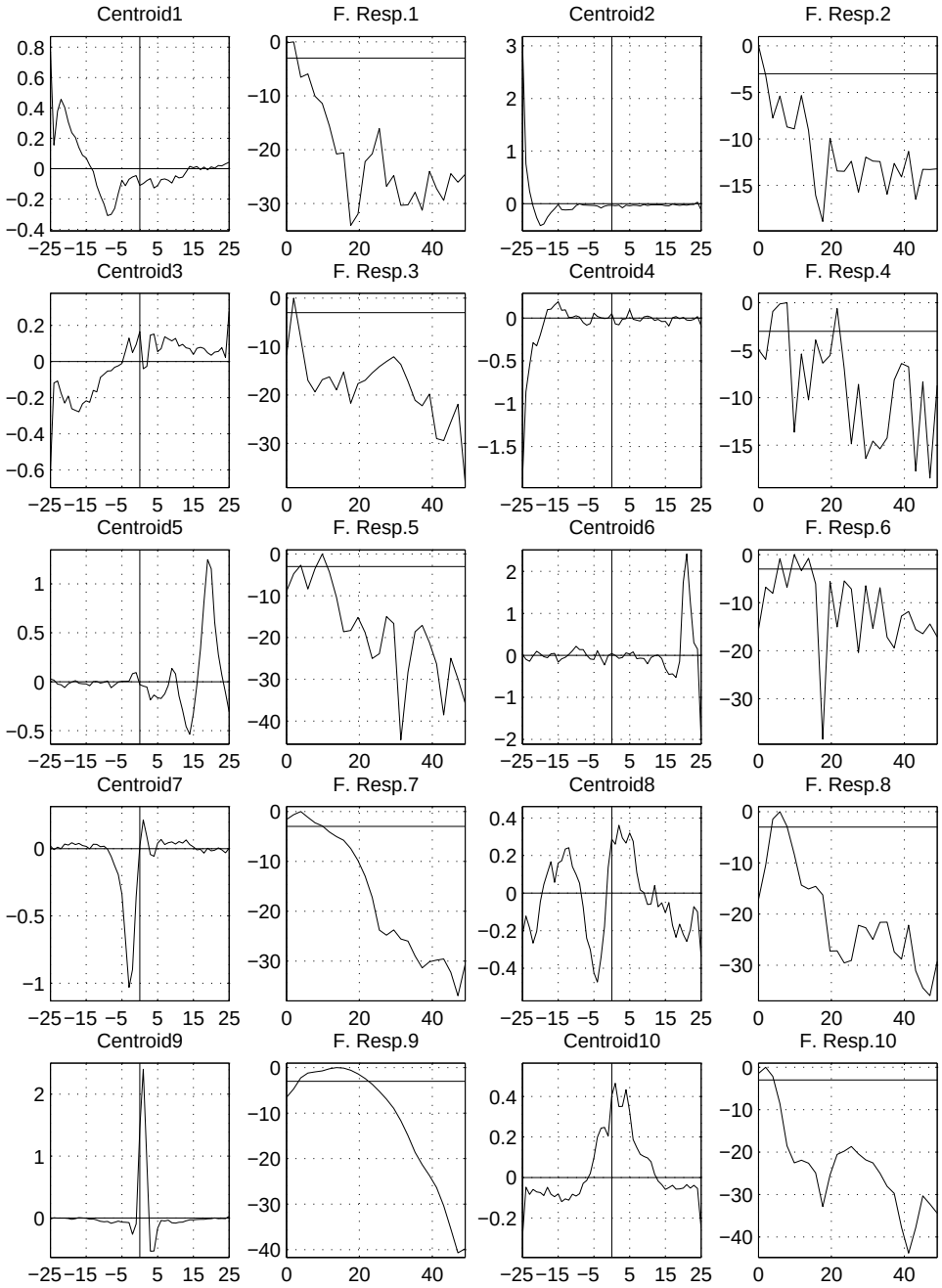

Figure D.5: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from TMLP trained on 34 hours of female CTS (Centroids 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

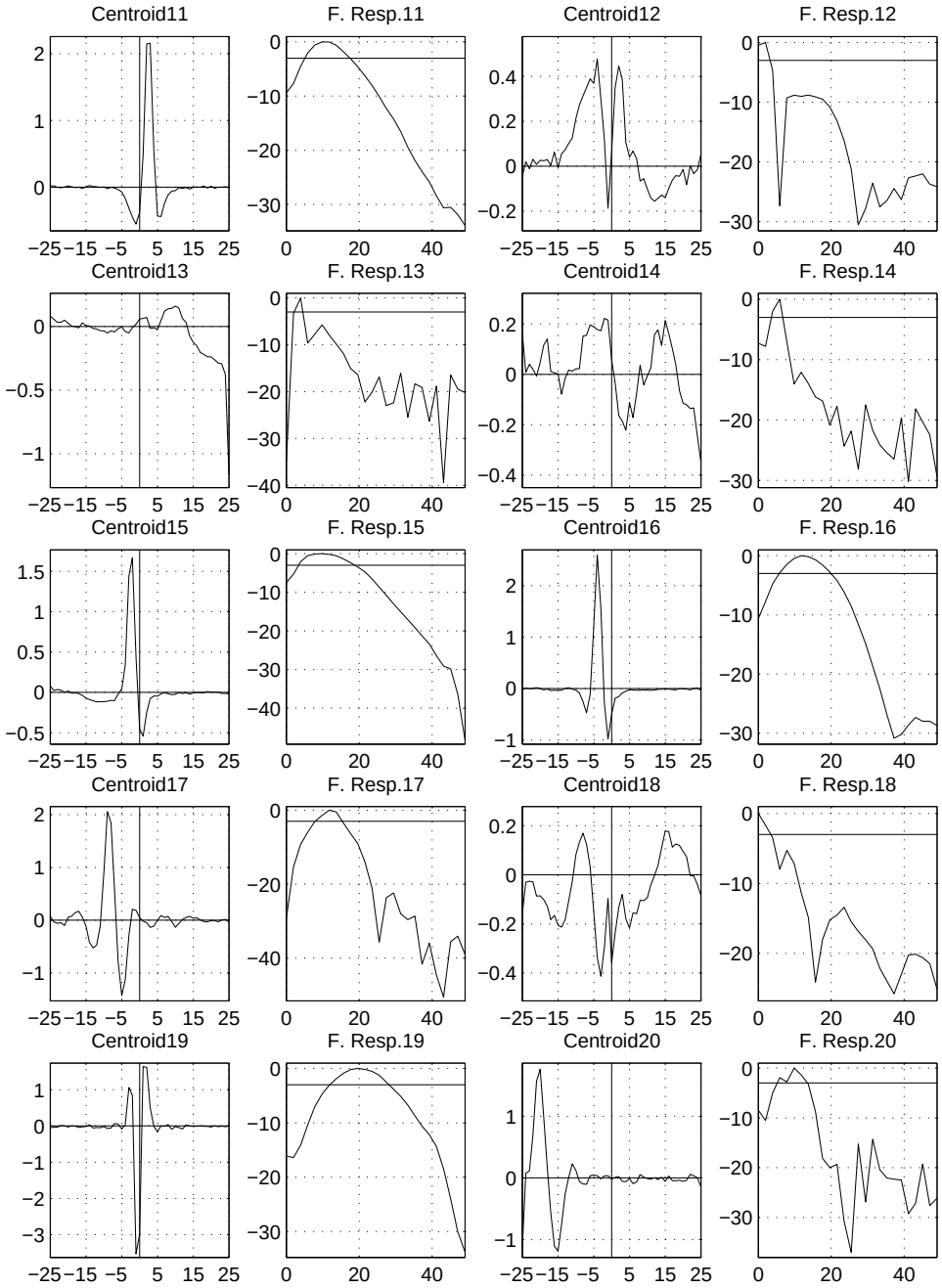

Figure D.6: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from TMLP trained on 34 hours of female CTS (Centroids 11-20). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

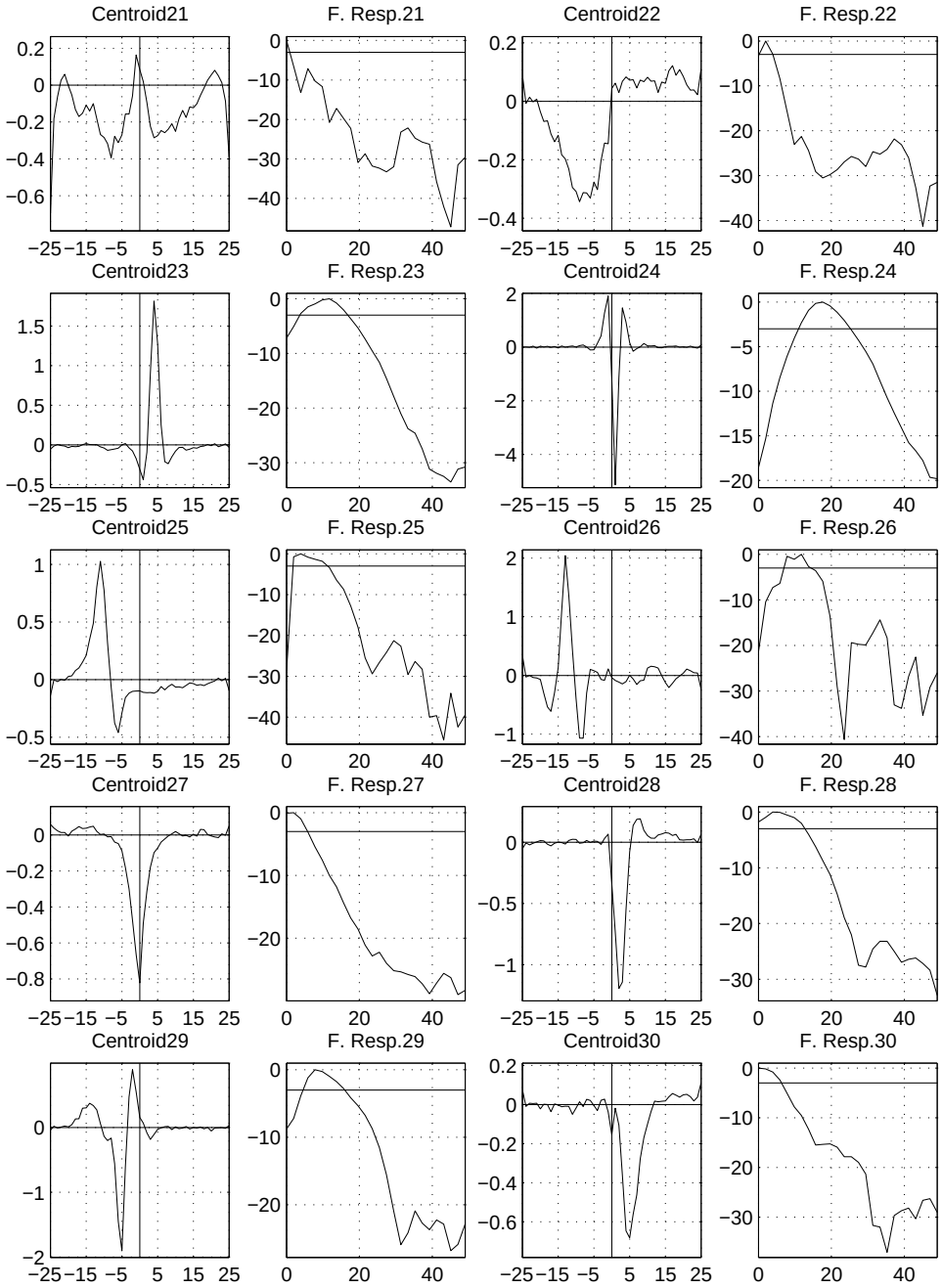

Figure D.7: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from TMLP trained on 34 hours of female CTS (Centroids 21-30). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

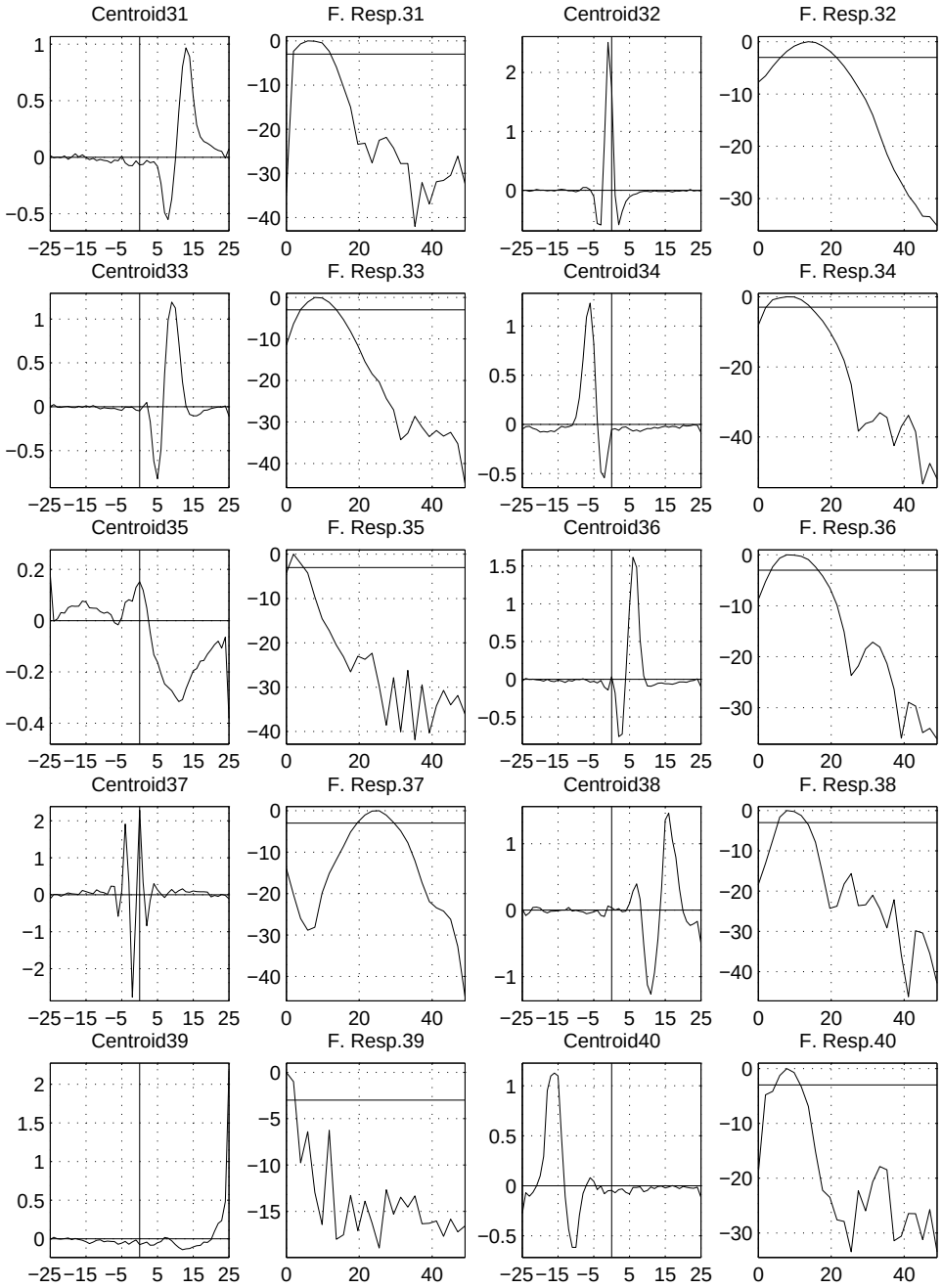

Figure D.8: The input-to-hidden weights and corresponding modulation frequency responses of critical-band hidden units from TMLP trained on 34 hours of female CTS (Centroids 31-40). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

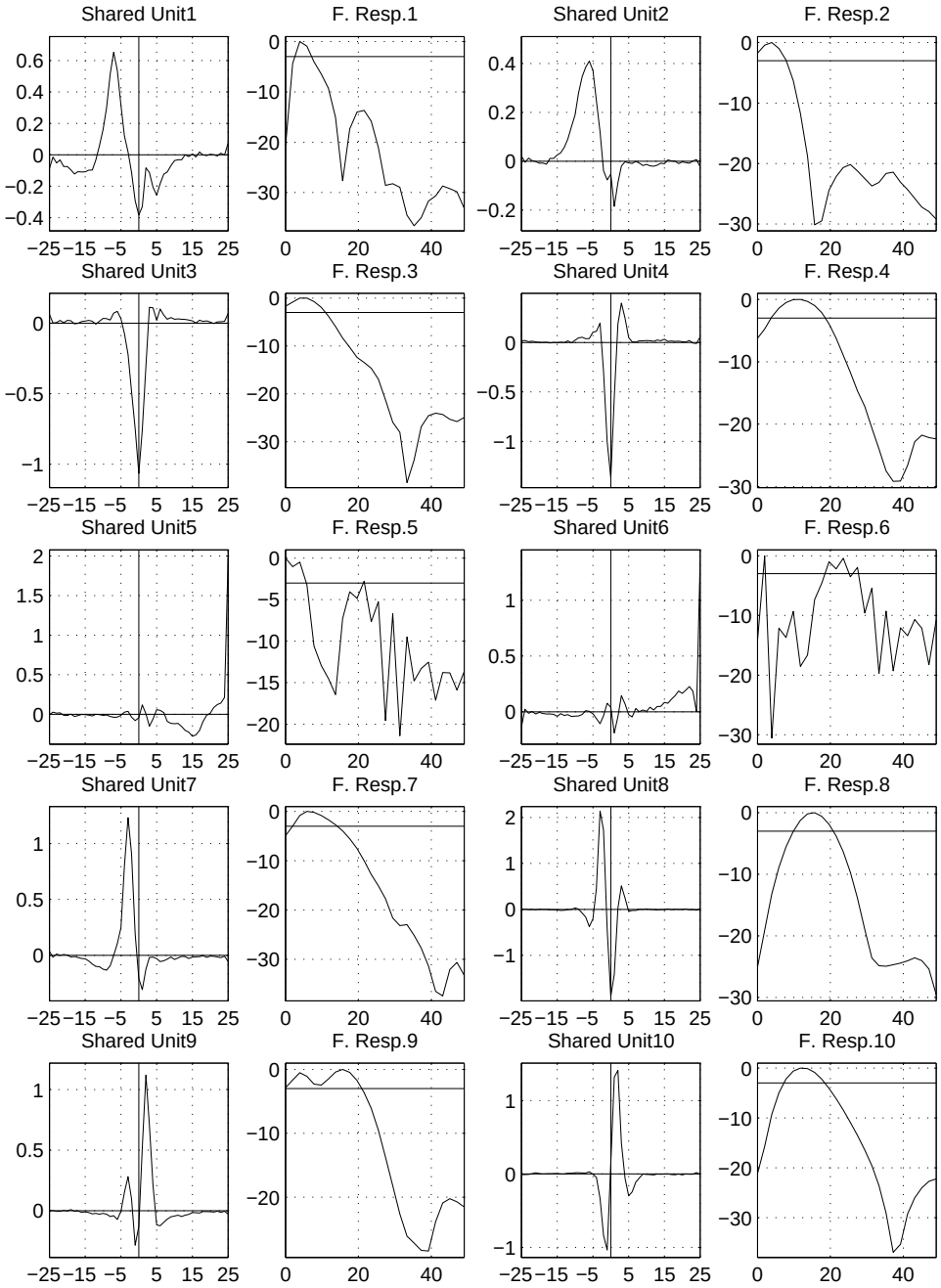

Figure D.9: The input-to-hidden weights and corresponding modulation frequency responses of shared critical-band hidden units from the weight-sharing TMLP (*TMLP S40*) trained on 34 hours of female CTS (shared weights 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

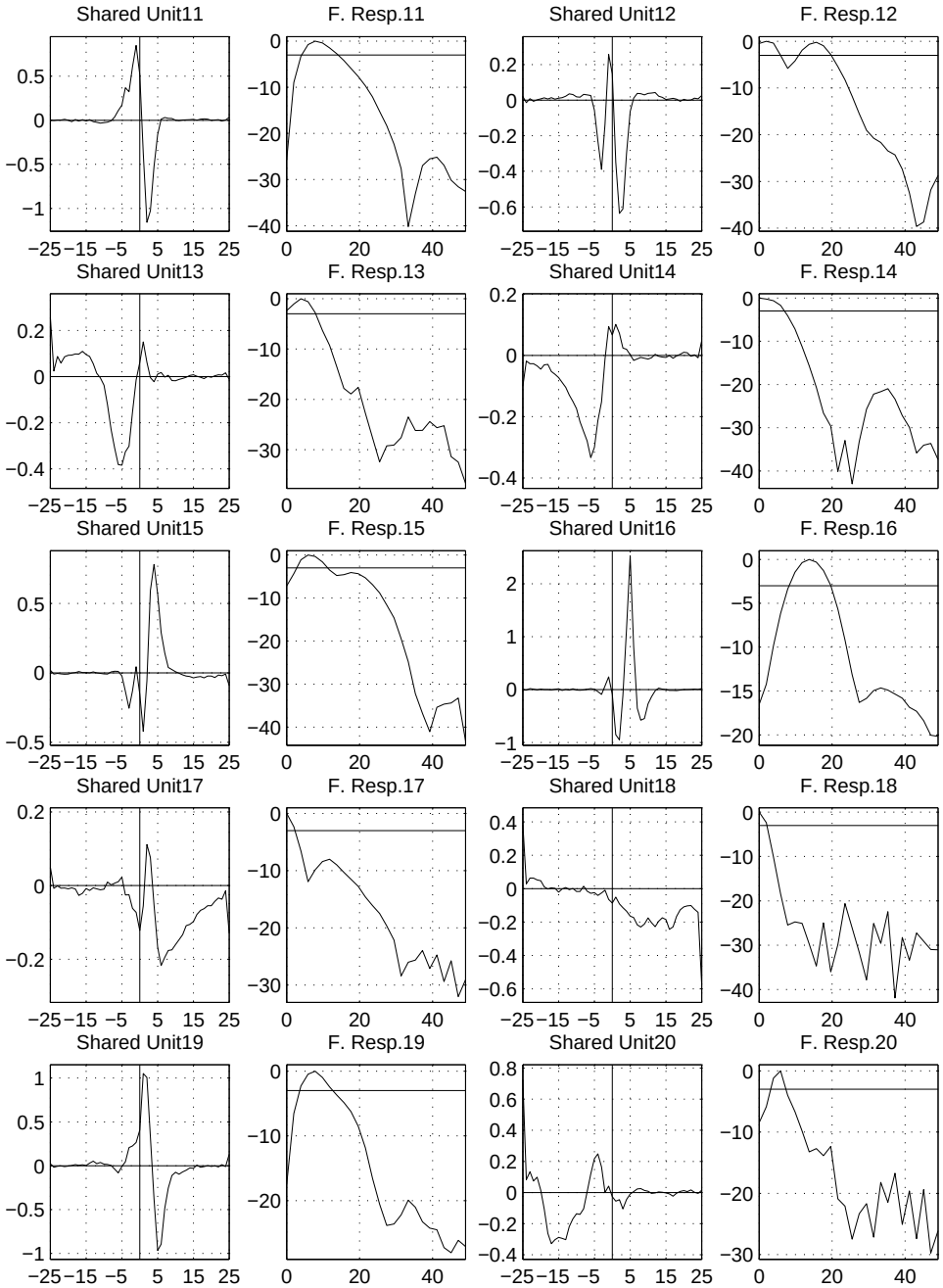

Figure D.10: The input-to-hidden weights and corresponding modulation frequency responses of shared critical-band hidden units from the weight-sharing TMLP (*TMLP S40*) trained on 34 hours of female CTS (shared weights 11-20). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

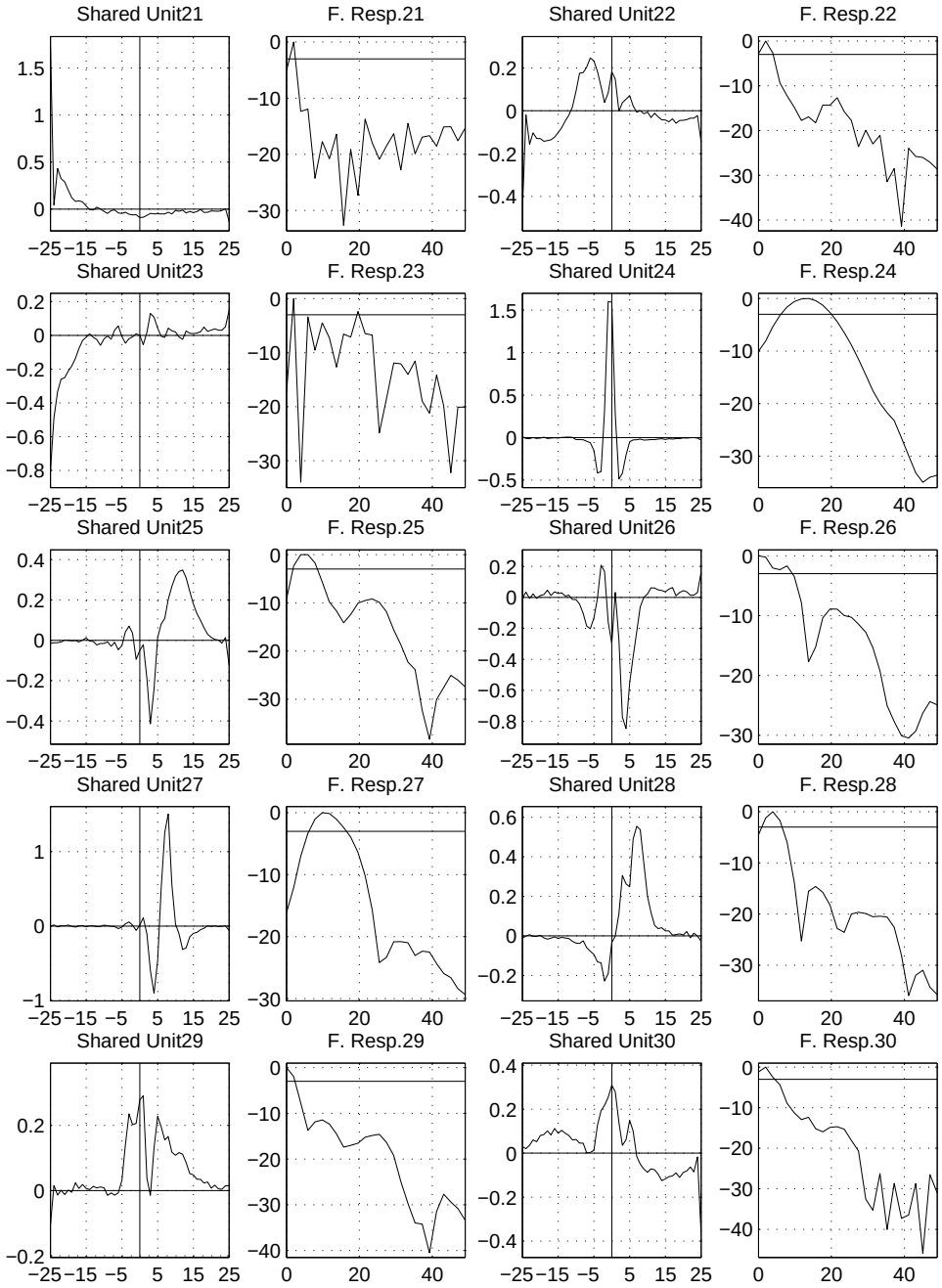

Figure D.11: The input-to-hidden weights and corresponding modulation frequency responses of shared critical-band hidden units from the weight-sharing TMLP (*TMLP S40*) trained on 34 hours of female CTS (shared weights 21-30). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

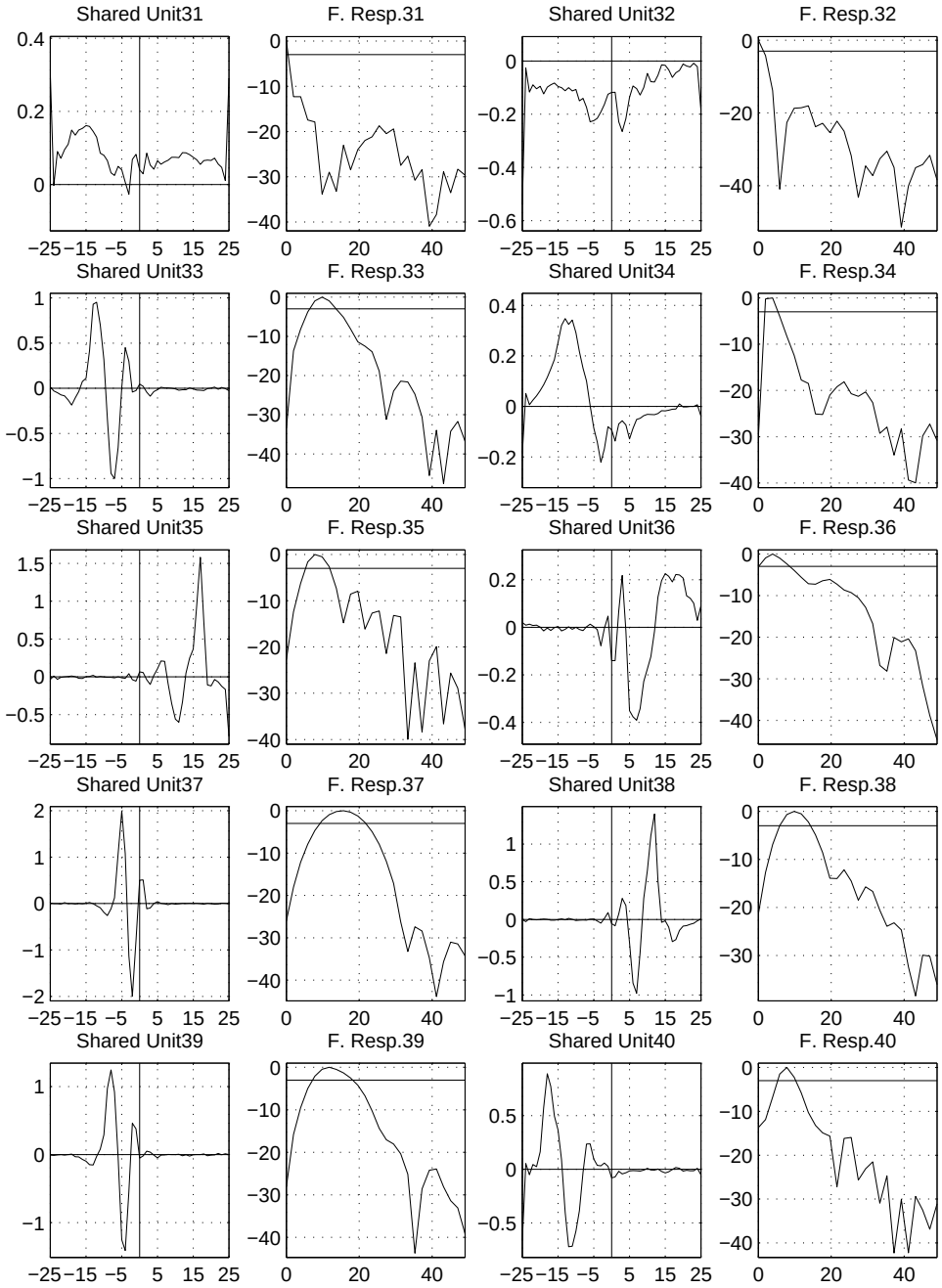

Figure D.12: The input-to-hidden weights and corresponding modulation frequency responses of shared critical-band hidden units from the weight-sharing TMLP (*TMLP S40*) trained on 34 hours of female CTS (shared weights 31-40). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the weight magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

Appendix E

PCA and LDA Critical-Band Patterns for CTS

In this appendix, we display pictures of critical-band temporal patterns learned by PCA and LDA methods from Chapter 5 computed on 34 hours of female CTS data. These patterns are the ones used to transform the input log critical-band energy trajectories. There are a total of 765 temporal patterns (15 critical-bands times 51 dimensions per critical-band), which is too many to plot. Since many of these temporal patterns look similar, we have clustered all of them using agglomerative clustering with the correlation based similarity measure described in Chapter 2 (Eq.2.5). We stop clustering at 40 clusters and average all patterns belonging to a particular cluster. We call this average pattern a centroid and plot the centroid patterns with their corresponding modulation frequency responses in Figures E.1, E.2, E.3, and E.4 for PCA and in Figures E.5, E.6, E.7, and E.8 for LDA.

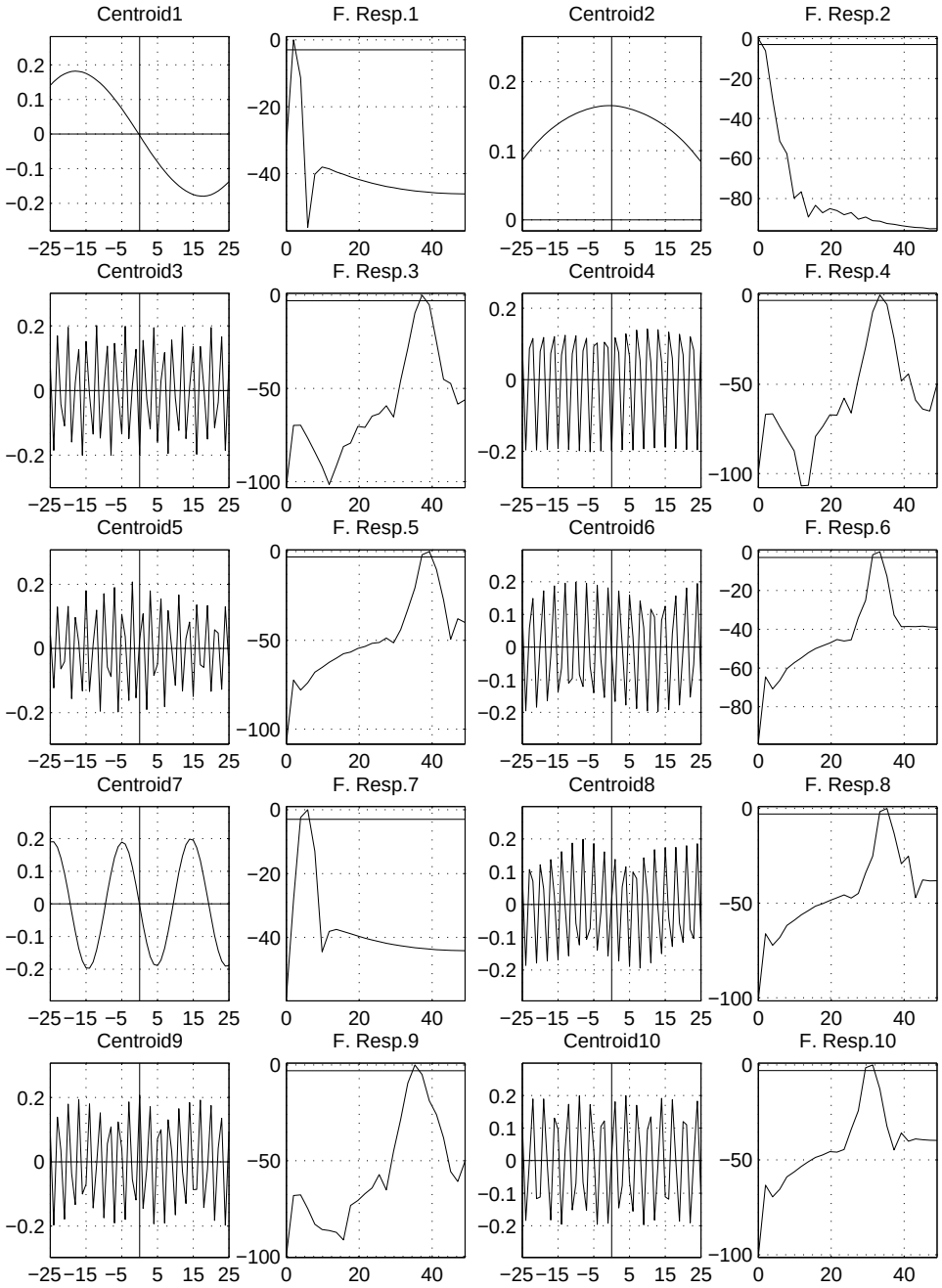

Figure E.1: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of PCA computed over 34 hours of female CTS (Centroids 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

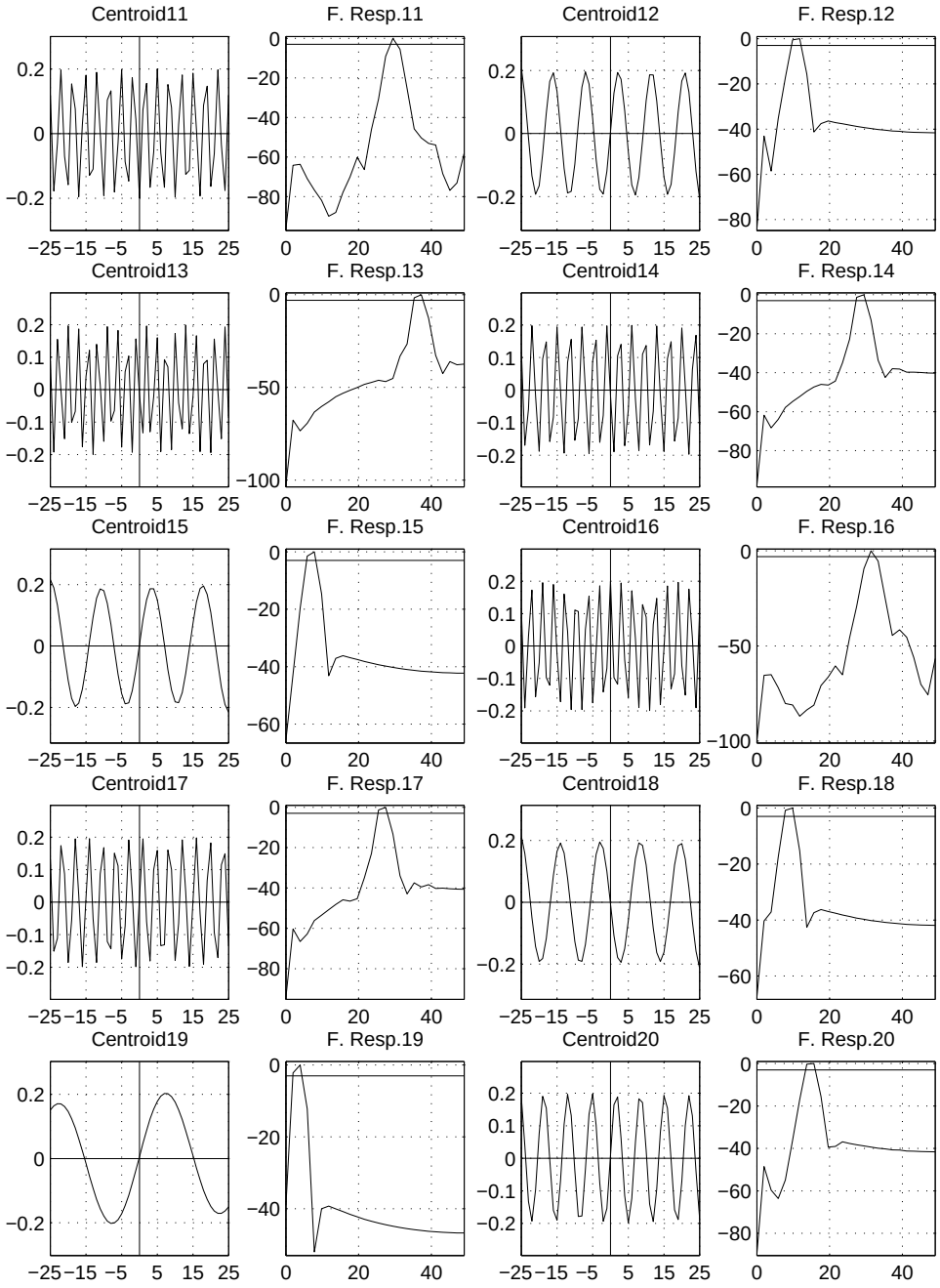

Figure E.2: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of PCA computed over 34 hours of female CTS (Centroids 11-20). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

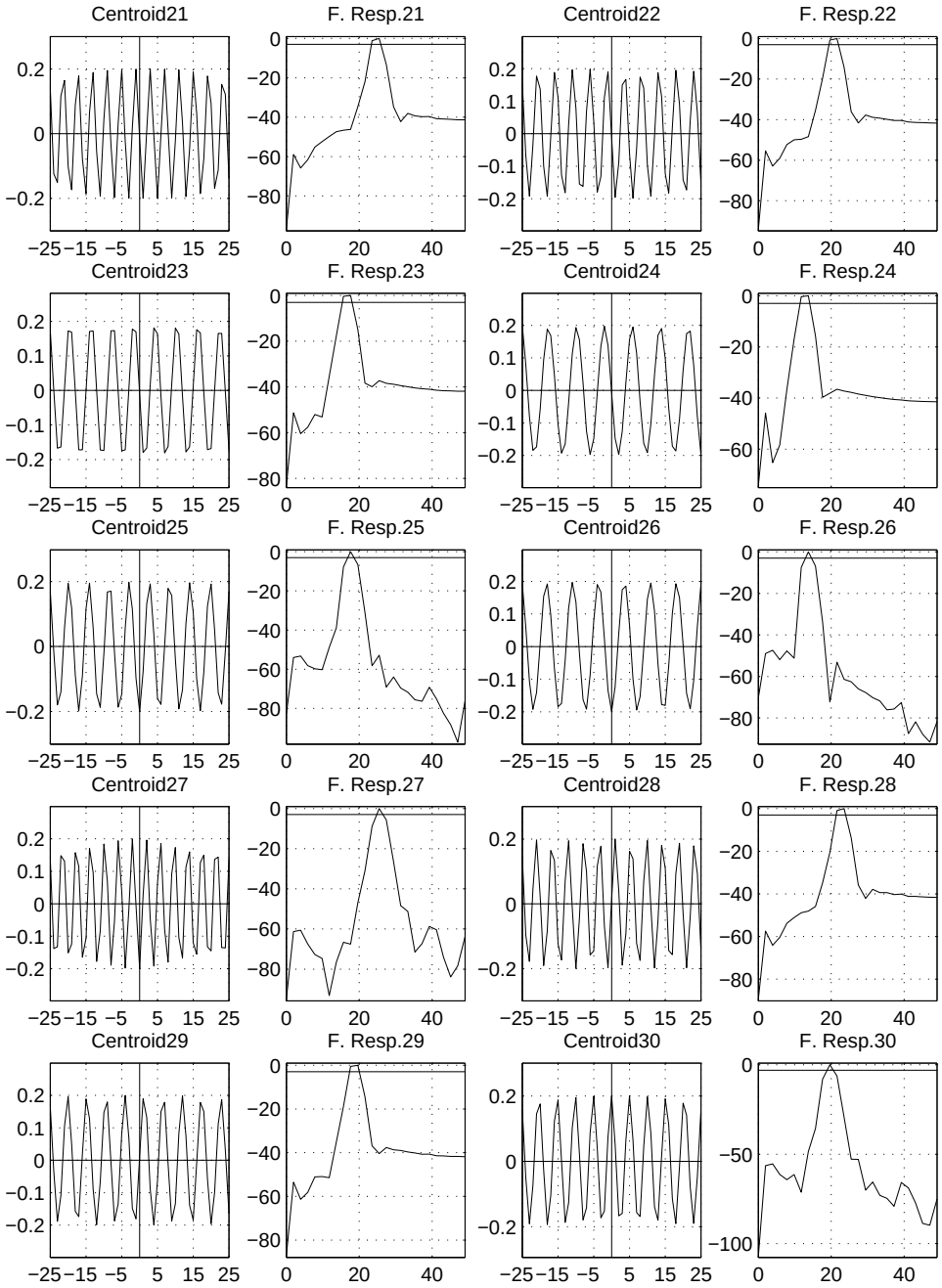

Figure E.3: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of PCA computed over 34 hours of female CTS (Centroids 21-30). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

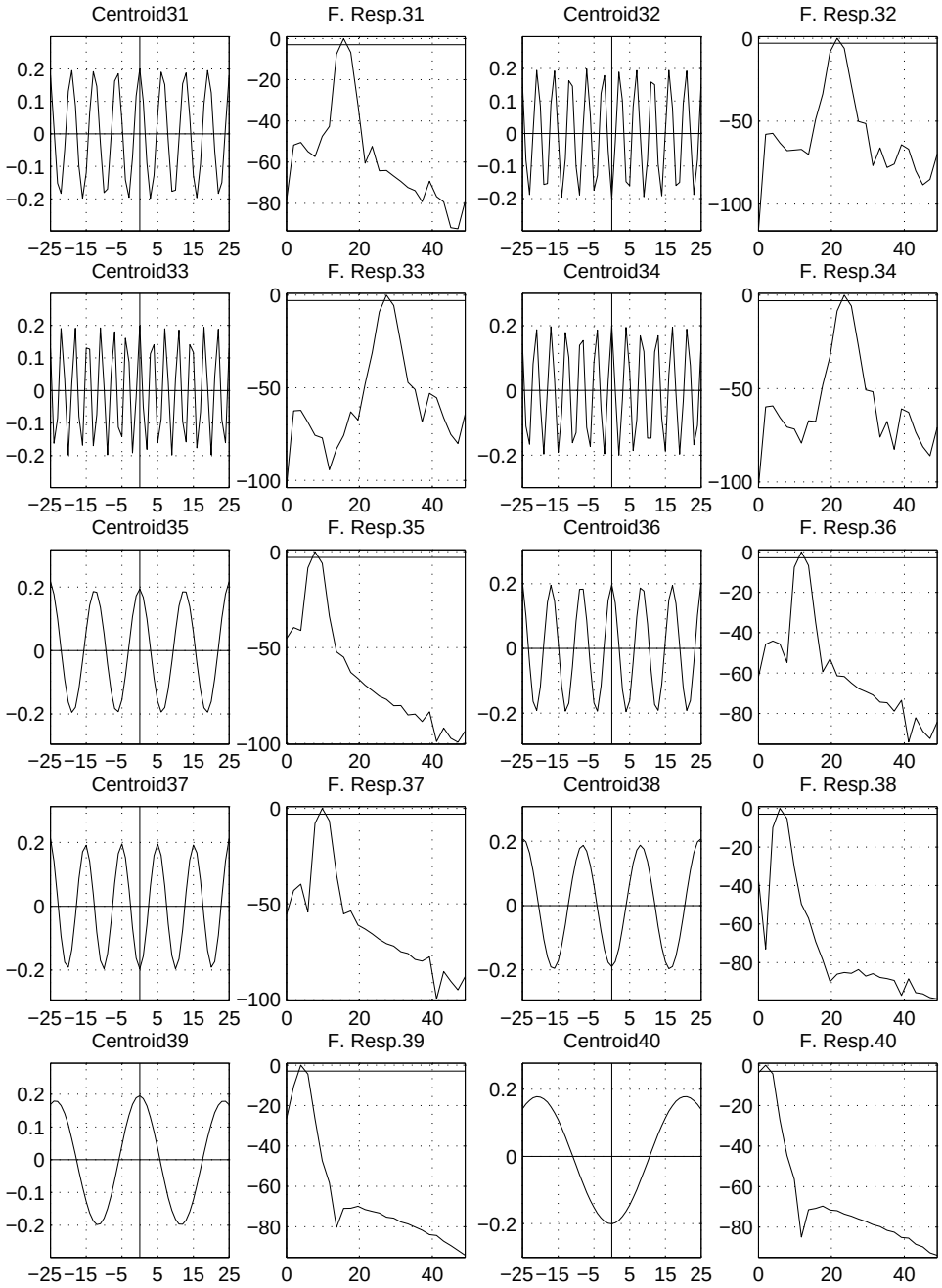

Figure E.4: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of PCA computed over 34 hours of female CTS (Centroids 31-40). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

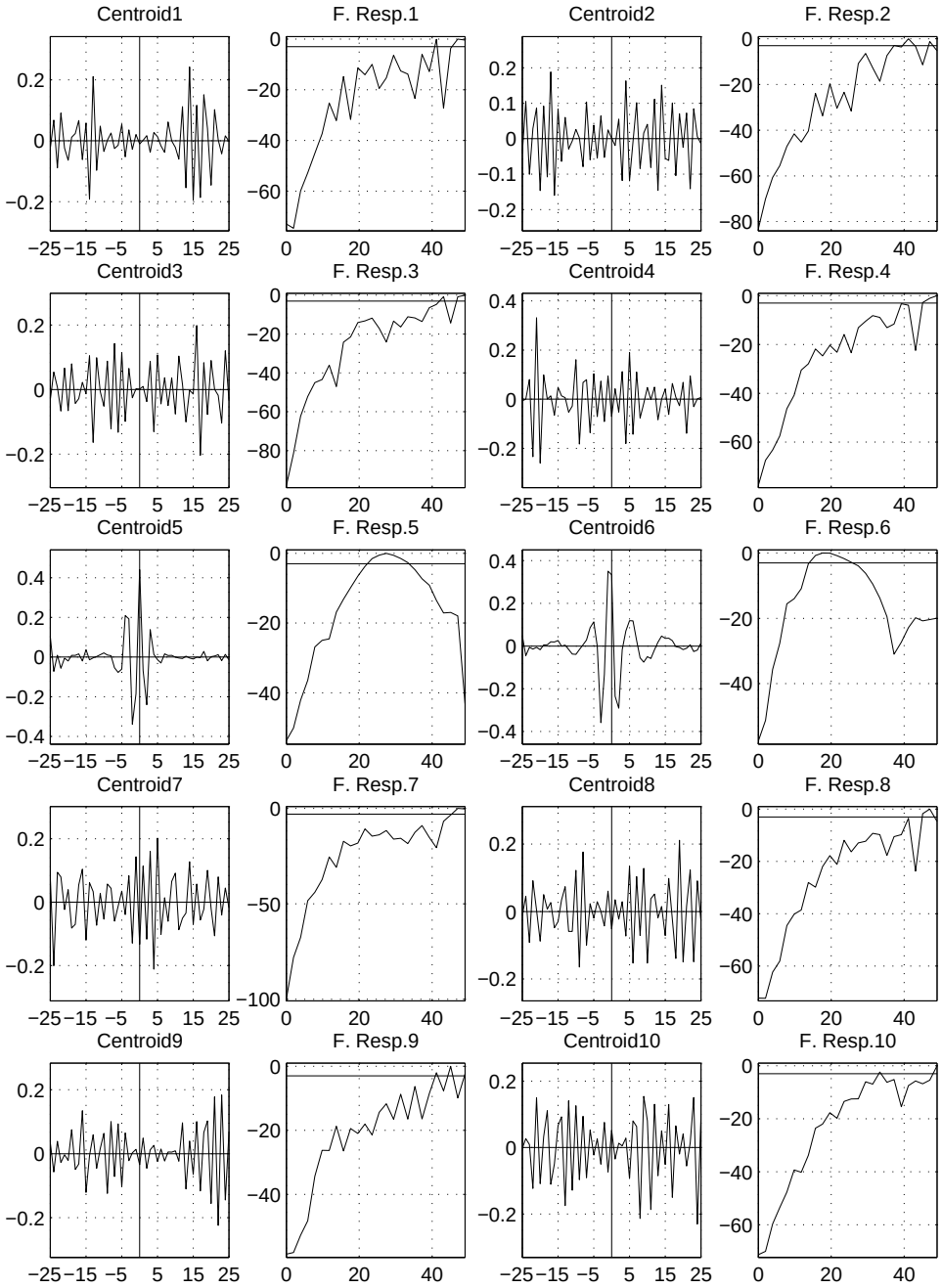

Figure E.5: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of LDA computed over 34 hours of female CTS (Centroids 1-10). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

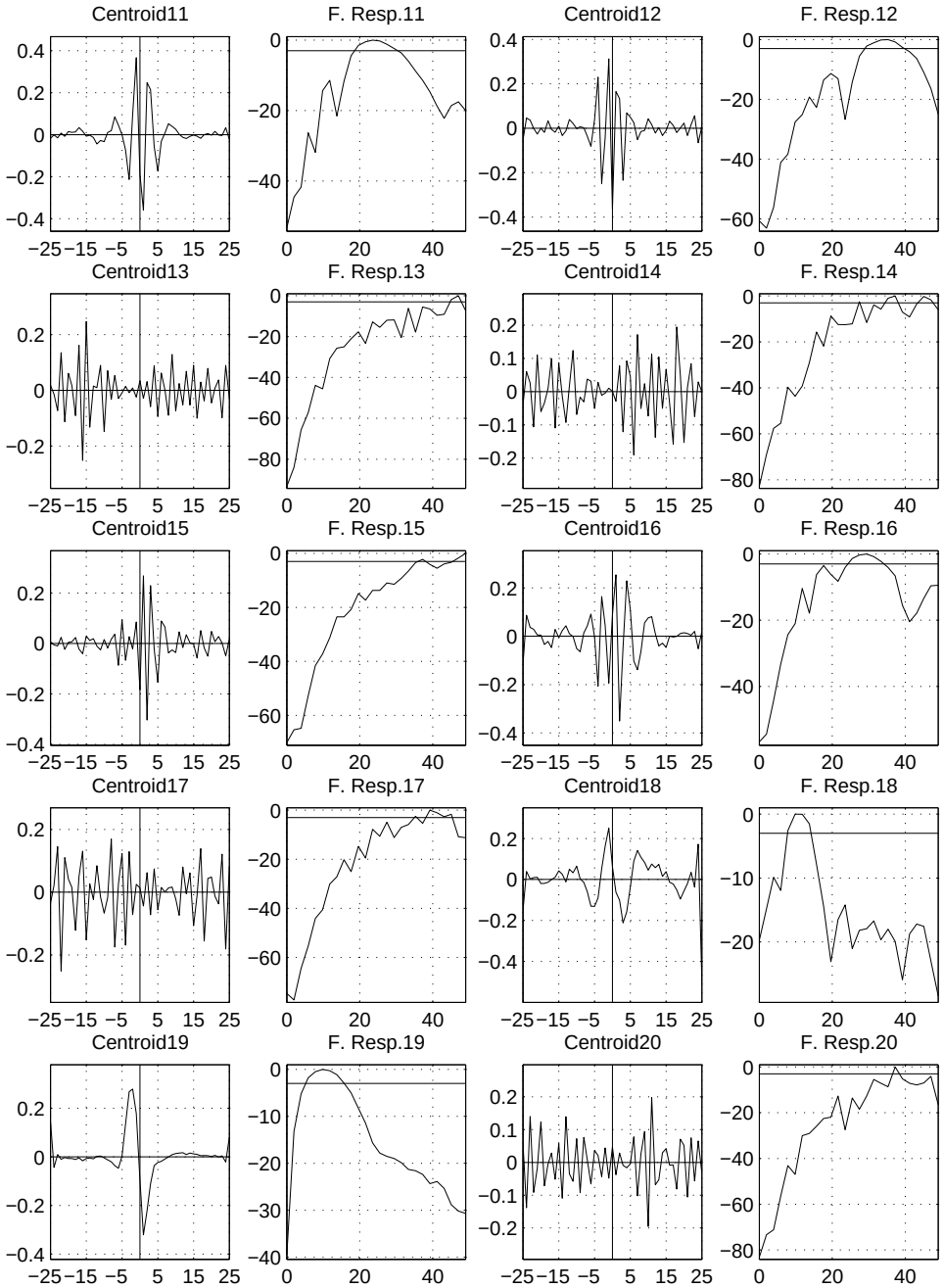

Figure E.6: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of LDA computed over 34 hours of female CTS (Centroids 11-20). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

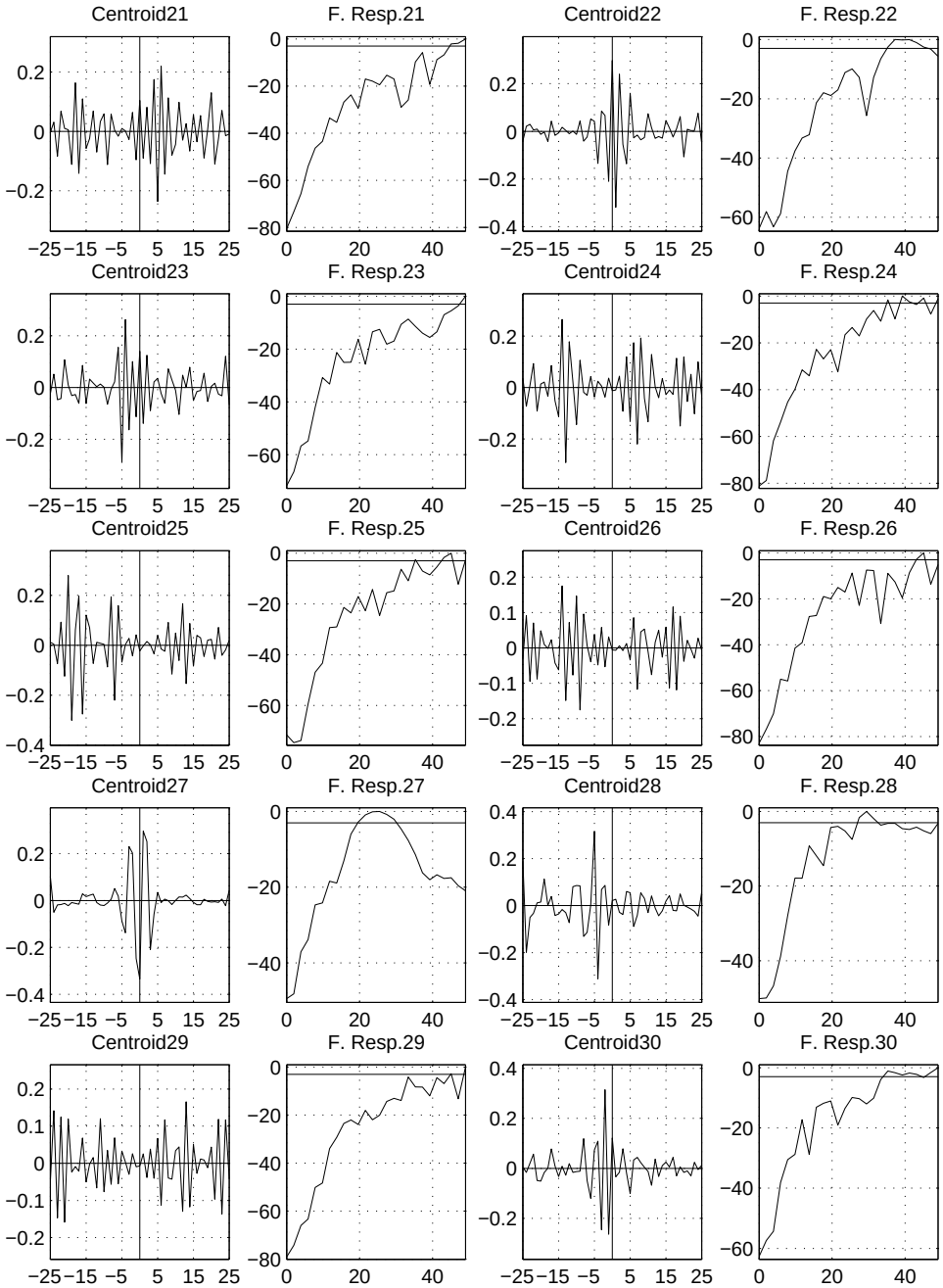

Figure E.7: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of LDA computed over 34 hours of female CTS (Centroids 21-30). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

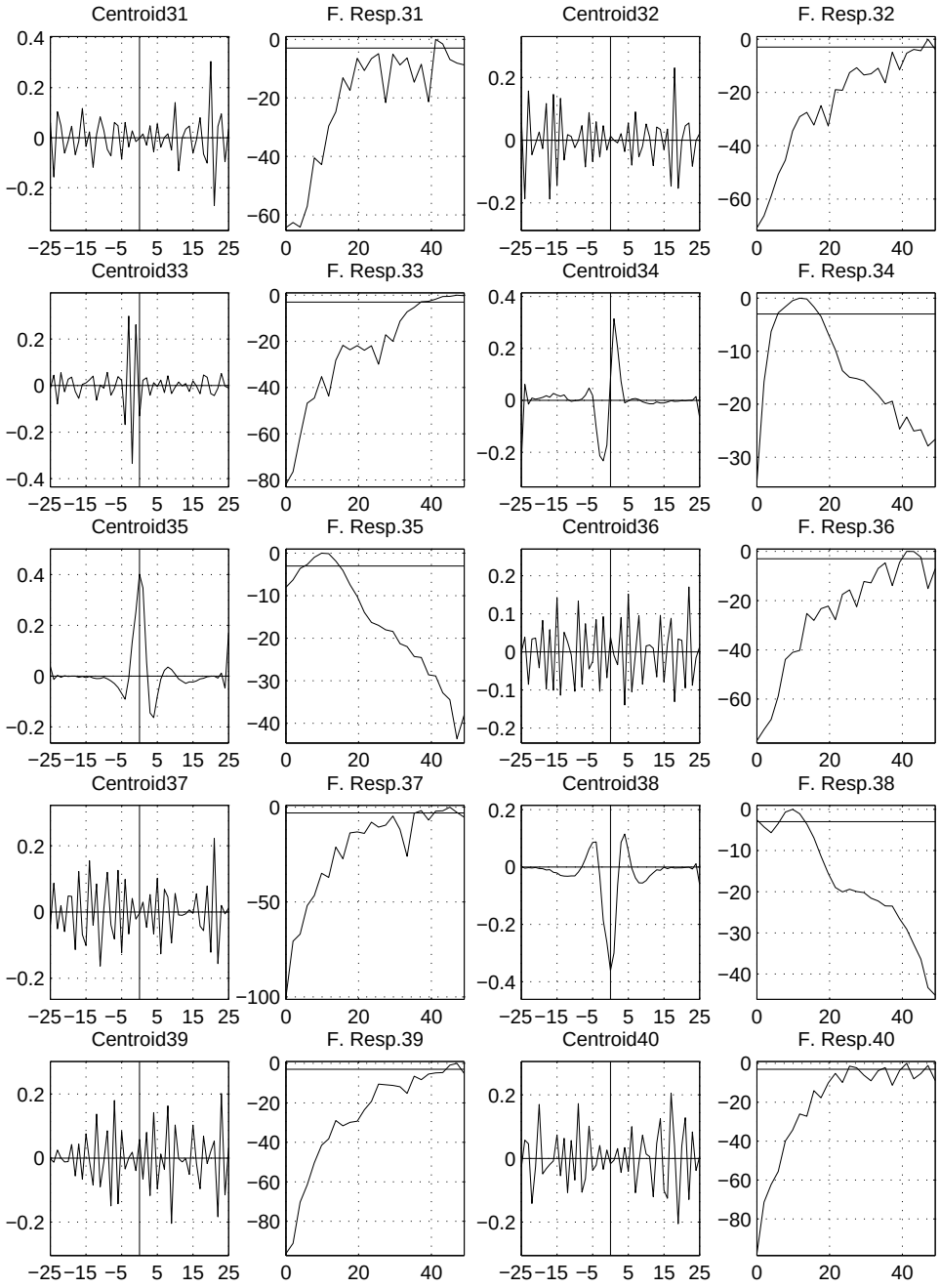

Figure E.8: The critical-band log energy trajectory transformation vectors and corresponding modulation frequency responses of LDA computed over 34 hours of female CTS (Centroids 31-40). The x-axes correspond to the frame index and modulation frequency respectively, and the y-axes correspond to the tranform magnitude and gain in decibels respectively. The horizontal line in the modulation frequency response is the -3 dB half power point.

Bibliography

[1] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. Qualcomm-ICSI-OGI features for ASR. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, USA, September 2002.

[2] J. B. Allen. How do humans process and recognize speech? *IEEE Transactions on Speech and Audio Processing*, 2(4):567–577, October 1994.

[3] C. Antoniou. Modular neural networks exploit large acoustic context through broad-class posteriors for continuous speech recogition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2001.

[4] C. A. Antoniou and T. J. Reynolds. Acoustic modelling using modular/ensemble of combinations of heterogeneous neural networks. In *Proceedings of the International Conference on Spoken Language Processing*, Beijing, China, 2000.

[5] T. Arai, M. Pavel, H. Hermansky, and C. Avendano. Intelligibility of speech with filtered time trajectories of spectral envelopes. In *Proceedings of the International Conference on Spoken Language Processing*, Philedelphia, 1996.

[6] B. S. Atal. Effectiveness of linear prediction characteristics of speech wave for automatic speaker identication and verication. *Journal of the Acoustical Society of America*, (55):1304–12, 1974.

[7] M. Athineos, H. Hermansky, and D. P. W. Ellis. LP-TRAP: Linear predictive temporal patterns. In *Proceedings of the International Conference on Spoken Language Processing*, Jeju, Korea, 2004.

[8] L. Atlas. Modulation spectral filtering of speech. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[9] C. Avendano. *Temporal Processing of Speech in a Time-Feature Space*. PhD thesis, Oregon Graduate Institute of Science and Technology, 1997.

[10] C. Avendano, S. van Vuuren, and H. Hermansky. Data based filter design for RASTA-like channel normalization in ASR. In *International Conference on Spoken Language Processing*, volume 3, pages 2087–90, Philadephia, Pennsylvania, October 1996.

[11] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[12] C. Benitez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, and S. Sivadas. Robust ASR front-end using spectral-based and discriminant features: experiments on the aurora tasks. In *Proceedings of Eurospeech*, Aalborg, Denmark, September 2001.

[13] J. Bernstein, K. Taussig, and J. J. Godfrey. MACROPHONE. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC94S21, 1994.

[14] J. Bilmes. Maximum mutual information based reduction strategies for cross-correlation based joint distributional modeling. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, pages 469–472, Seattle, 1998.

[15] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[16] H. Bourlard and S. Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, 1996.

[17] H. Bourlard and S. Dupont. Sub-band based speech recognition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 1251–1254, Munich, Germany, April 1997. IEEE.

[18] H. Bourlard and N. Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[19] A. Canavan, D. Graff, and G. Zipperlen. CALLHOME american english speech. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97S42, 1997.

[20] Center for Spoken Language Understanding, Department of Computer Science and Engineering, Oregon Graduate Institute. Numbers corpus, release 1.0, 1995.

[21] C. Cerisara, J.-P. Haton, and D. Fohr. Towards a global optimization scheme for multi-band speech recognition. In *Proceedings of Eurospeech*, 1999.

[22] C. Cerisara, J.-P. Haton, J.-F. Mari, and D. Fohr. Multi-band continuous speech recognition. In *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[23] C. Cerisara, J.-P. Haton, J.-F. Mari, and D. Fohr. A recombination model for multi-band speech recognition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Seattle, 1998.

[24] B. Y. Chen, S. Chang, and S. Sivadas. Learning discriminative temporal patterns in speech: Development of novel TRAPS-like classifiers. In *Proceedings of Eurospeech*, Geneva, Switzerland, September 2003.

[25] K. Daoudi, D. Fohr, and C. Antoine. A new approach for multi-band speech recognition based on probabilistic graphical models. In *Proceedings of the International Conference on Spoken Language Processing*, volume 1, pages 329–332, October 2000.

[26] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[27] L. Deng and D. X. Sun. Phonetic classification and recognition using HMM representation of overlapping articulatory features for all classes of english sounds. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 45–48. IEEE, 1994.

[28] N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker, and J. Picone. Resegmentation of Switchboard. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1543–1546, Sydney, Australia, November 1998.

[29] R. Drullman, J. M. Festen, and R. Plomp. Effect of temporal envelope smearing on speech reception. *Journal of the Acoustical Society of America*, 95(2):1053–1064, 1994.

[30] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, Inc., New York, NY, 2 edition, 2001.

[31] D. Ellis and J. Bilmes. Using mutual information to design feature combinations. In *Proceedings of the International Conference on Spoken Language Processing*, Beijing, October 2000.

[32] D. Ellis and M. R. Gomez. Investigations into Tandem acoustic modeling for the Aurora task. In *Proceedings of Eurospeech, Special Event on Noise Robust Recognition*, Denmark, September 2001.

[33] D. Ellis and N. Morgan. Size matters: An empirical study of neural network training for large vocabulary continuous speech recognition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 1999.

[34] D. Ellis, R. Singh, and S. Sivadas. Tandem acoustic modeling in large-vocabulary recognition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Salt Lake City, May 2001.

[35] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, X. Liu, D. Mrva, K. C. Sim, L. Wang, P. C. Woodland, and K. Yu. Development of the 2004 CU-HTK English CTS systems using more than two thousand hours of data. In *Proceedings of the EARS RT-04F Workshop*, Palisades, New York, November 2004.

[36] J. G. Fiscus, J. S. Garofolo, A. Le, A. F. Martin, D. S. Pallet, M. A. Przybocki, and G. Sanders. Results of the fall 2004 STT and MDE evaluation. In *Proceedings of the EARS RT-04F Workshop*, Palisades, New York, November 2004.

[37] H. Fletcher. *Speech and Hearing in Communication.* D. Van Nostrand Company, Inc., 120 Alexander St., Princeton, New Jersey, 1953.

[38] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transaction on Acoustics Speech and Signal Processing*, (ASSP-34):52, 1986.

[39] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[40] J. S. Garofolo. Getting started with the DARPA TIMIT CD-ROM, 1988. National Institute of Standards and Technology (NIST).

[41] J. Godfrey, E. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development, 1992.

[42] J. J. Godfrey and E. Holliman. SWITCHBOARD credit card. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S8, 1993.

[43] D. Graff, K. Walker, and D. Miller. Switchboard Cellular part 1 audio. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001S13, 2001.

[44] S. Greenberg, T. Arai, and R. Silipo. Speech intelligibility derived from exceedingly sparse information. In *Proceedings of the International Conference on Spoken Language Processing*, volume 6, pages 2803–2806, December 1998.

[45] S. Greenberg, S. Chang, and J. Hollenbeck. An introduction to the diagnostic evaluation of Switchboard corpus automatic speech recognition systems. In *NIST Speech Transcription Workshop*, 2000.

[46] F. Grézl and H. Hermansky. Local averaging and differentiating of spectral plane for TRAP-based ASR. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[47] D. B. Guralnik, editor. *Webster's New World Dictionary of the American Language: Second College Edition.* Simon and Schuster, 1982.

[48] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, April 1990.

[49] H. Hermansky, D. Ellis, and S. Sharma. Tandem connectionist feature stream extraction for conventional HMM systems. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, volume III, pages 1635–1638, Istanbul, 2000.

[50] H. Hermansky and P. Jain. Band-independent speech-event categories for TRAP based ASR. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[51] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, October 1994.

[52] H. Hermansky and S. Sharma. TRAPs: Classifiers of TempoRAl Patterns. In *Proceedings of the International Conference on Spoken Language Processing*, Sydney, 1998.

[53] H. Hermansky and S. Sharma. Temporal patterns (TRAPS) in ASR of noisy speech. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Phoenix, Arizona, 1999.

[54] H. Hermansky, S. Sharma, and P. Jain. Data-derived nonlinear mapping for feature extraction in HMM. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Keystone, Colorado, 1999.

[55] H. Hermansky, S. Tibrewala, and M. Pavel. Towards ASR on partially corrupted speech. In *International Conference on Spoken Language Processing*, volume 1, pages 462–5, Philadephia, Pennsylvania, October 1996.

[56] G. Hirsch. Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task. In *ETSI STQ Aurora DSR Working Group*, June 2001.

[57] M. Hochberg. y0 – Y0 recognizer from ICSI. manpage, August 1993. WERNICKE distribution.

[58] T. Houtgast and H. J. M. Steeneken. The modulation transfer function in room acoustics as a predictor of speech intelligibility. *Acustica*, 28:66–73, 1973.

[59] T. Houtgast and H. J. M. Steeneken. A review of the MTF concept in room acoustics and its use for estimating speech intelligibility. *Journal of the Acoustical Society of America*, 77(3):1069–1077, 1985.

[60] J.-W. Hung and L.-S. Lee. Data-driven temporal filters obtained via different optimization criteria evaluated on AURORA2 database. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2002.

[61] J.-W. Hung, H.-M. Wang, and L.-S. Lee. Comparative analysis for data-driven temporal filters obtained via principal component analysis (PCA) and linear discriminant analysis (LDA) in speech recognition. In *Proceedings of Eurospeech*, Aalborg, 2001.

[62] P. Jain. *Temporal Patterns of Frequency-Localized Features in ASR.* PhD thesis, OGI School of Science and Engineering, 2003.

[63] P. Jain and H. Hermansky. Beyond a single critical-band in TRAP based ASR. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[64] P. Jain, H. Hermansky, and B. Kingsbury. Distributed speech recognition using noise-robust MFCC and TRAPS-estimated manner features. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, 2002.

[65] A. Janin, D. Ellis, and N. Morgan. Multi-stream speech recognition: Ready for prime time? In *Proceedings of Eurospeech*, Budapest, 1999.

[66] B.-H. Juang, W. Chou, and C.-H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3), May 1997.

[67] S. S. Kajarekar, B. Yegnanarayana, and H. Hermansky. A study of two dimensional linear dicriminants for ASR. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Salt Lake City, May 2001.

[68] N. Kanedera, T. Arai, H. Hermansky, and M. Pavel. On the importance of various modulation frequencies for speech recognition. In *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[69] M. Karafiát, F. Grézl, and J. Černocký. TRAP based features for LVCSR of meeting data. In *Proceedings of the International Conference on Spoken Language Processing*, Jeju, Korea, 2004.

[70] B. Kingsbury, P. Jain, and A. Adami. A hybrid HMM/TRAPS model for robust voice activity detection. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, 2002.

[71] B. Kingsbury, N. Morgan, and S. Greenberg. Robust speech recognition using the modulation spectrogram. *Speech Communication*, 25:117–132, 1998.

[72] M. Kleinschmidt. Localized spectro-temporal features for automatic speech recognition. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[73] M. Kleinschmidt and D. Gelbart. Improving word accuracy with Gabor feature extraction. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, USA, September 2002.

[74] A. N. Kolmogorov. On the representation of continuos functions of several variables by superposition of continuous functions of one variable and addition. *Doklady Akademiia Nauk SSSR*, 114(5):953–956, 1957.

[75] V. Kürková. Kolmogorov's theorem and multilayer neural networks. *Neural Computation*, 5(3):501–506, 1992.

[76] L. Lamel, F. Lefevre, J.-L. Gauvain, and G. Adda. Portability issues for speech recognition technologies. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001.

[77] K. F. Lee and H. W. Hon. Speaker-independent phoneme recognition using hidden markov models. *IEEE Transactions on Acoustic Speech, and Signal Processing*, 37(12):1641–1648, November 1989.

[78] S. Lee and J. Glass. Real-time probabilistic segmentation for segment-based speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, Sydney, 1998.

[79] T. W. Lee. *Independent Component Analysis - Theory and Applications.* Kluwer Academic Publishers, 1998.

[80] C. J. Legetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9(2):171–186, April 1995.

[81] M. Lieb and R. Haeb-Umbach. LDA derived cepstral trajectory filters in adverse environmental conditions. In *Proceedings International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 1105–8, Istanbul, Turkey, June 2000. IEEE.

[82] J. lin Shen and W. L. Hwang. New temporal features for robust speech recognition with emphasis on microphone variations. *Computer Speech and Language*, 13:65–78, 1999.

[83] R. P. Lippmann. Accurate consonant perception without mid-frequency speech energy. *IEEE Transactions on speech and audio processing*, 4(1):66–69, January 1996.

[84] R. P. Lippmann. Speech perception by humans and machines. *Speech Communication*, 22(1):1–15, 1997.

[85] N. Malayath and H. Hermansky. Data-driven spectral basis functions for automatic speech recognition. *Speech Communication*, 40:449–466, 2003.

[86] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, 2000.

[87] P. Mermelstein and S. Davis. Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics and Speech Signal Processing*, 28:357–366, 1980.

[88] G. A. Miller and P. E. Nicely. An analysis of perceptual confusions among some english consonants. *The Journal of the Acoustical Society of America*, 27(2):338–352, March 1955.

[89] B. Milner. Inclusion of temporal information into features for speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, 1996.

[90] B. Milner. Cepstral-time matrices and LDA for improved connected digit and subword recognition accuracy. In *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[91] N. Mirghafori. *A Multi-Band Approach to Automatic Speech Recognition*. PhD thesis, University of California at Berkeley, November 1998.

[92] N. Mirghafori and N. Morgan. Combining connectionist multi-band and full-band probability streams for speech recognition of natural numbers. In *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[93] N. Mirghafori and N. Morgan. Transmissions and transitions: A study of two common assumptions in multi-band ASR. In *International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 713–16, Seattle, Washington, May 1998. IEEE.

[94] H. Misra, H. Bourlard, and V. Tyagi. New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Hong Kong, 2003.

[95] R. K. Moore. Modeling data entry rates for ASR and alternative input methods. In *Proceedings of Interspeech-2004*, pages 2285–2288, 2004.

[96] N. Morgan and H. Bourlard. Continuous speech recognition. *IEEE Signal Processing Magazine*, 12(3):25–42, May 1995.

[97] N. Morgan, B. Chen, Q. Zhu, and A. Stolcke. TRAPping conversational speech: Extending TRAP/Tandem approaches to conversational telephone speech recognition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Montreal, May 2004.

[98] A. C. Morris, A. Hagen, and H. Bourlard. The full combination sub-bands approach to noise robust HMM/ANN based ASR. In *Proceedings of Eurospeech '99*, pages 599–602, 1999.

[99] P. Motlíček and J. Černocký. Time-domain based temporal processing with aplication of orthogonal transformations. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[100] C. Nadeu, D. Macho, and J. Hernando. Time and frequency filtering of filter-bank energies for robust HMM speech recognition. *Speech Communication*, 34:93–114, 2001.

[101] C. Nadeu, P. Pachès-Leal, and B.-H. Juang. Filtering the time sequences of spectral parameters for speech recognition. *Speech Communication*, 22:315–332, 1997.

[102] S. Okawa, E. Bocchieri, and A. Potamianos. Multi-band speech recognition in noisy environments. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Seattle, 1998.

[103] S. Okawa, T. Nakajima, and K. Shirai. A recombination strategy for multi-band speech recognition based on mutual information criterion. In *Proceedings of Eurospeech*, Budapest, 1999.

[104] D. S. Pallett. A look at NIST's benchmark ASR tests: Past, present, and future. http://www.nist.gov/speech/history/pdf/NIST_benchmark_ASRtests_2003.pdf, 2003.

[105] S. Renals and M. Hochberg. Efficient evaluation of the LVCSR search space using the Noway decoder. In *International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 149–152, Atlanta, Georgia, May 1996. IEEE.

[106] A. Robinson, M. Hochber, and S. Renals. IPA: Improved modelling with recurrent neural networks. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, pages 37–40, 1994.

[107] T. Robinson and J. Christie. Time-first search for large vocabulary speech recognition. In *International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 829–32, Seattle, Washington, May 1998. IEEE.

[108] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[109] L. K. Saul, M. G. Rahim, and J. B. Allen. Learning from examples in critical bands of speech. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Keystone, Colorado, USA, December 1999. IEEE.

[110] L. K. Saul, M. G. Rahim, and J. B. Allen. A statistical model for robust integration of narrowband cues in speech. *Computer Speech and Language*, 2001.

[111] P. Schwarz, P. Matějka, and J. Černocký. Recognition of phoneme strings using TRAP technique. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[112] S. Sharma. *Multi-Stream Approach to Robust Speech Recognition*. PhD thesis, Oregon Graduate Institute of Science and Technology, 1999.

[113] S. Sharma, D. Ellis, S. Kajarekar, P. Jain, and H. Hermansky. Feature extraction using non-linear transformations for robust speech recognition on the Aurora database.

In *International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 1117–20, Istanbul, Turkey, June 2000. IEEE.

[114] M. Shire. *Discriminant Training of Front-End and Acoustic Modeling Stages to Heterogeneous Acoustic Environments for Multi-Stream Automatic Speech Recognition*. PhD thesis, University of California, Berkeley, 2000.

[115] M. L. Shire. Data-driven modulation filter design under adverse acoustic conditions and using phonetic and syllabic targets. In *EUROSPEECH*, pages 1123–6, Budapest, Hungary, September 1999. ESCA.

[116] M. L. Shire and B. Y. Chen. Data-driven RASTA filters in reverberation. In *International Conference on Acoustics Speech and Signal Processing*, volume 3, pages 1627–30, Istanbul, Turkey, June 2000. IEEE.

[117] M. L. Shire and B. Y. Chen. On data-derived temporal processing in speech feature extraction. In *International Conference on Spoken Language Processing*, volume 3, pages 71–4, Beijing, China, October 2000.

[118] R. Silipo, S. Greenberg, and T. Arai. Temporal constraints on speech intelligibility as deduced from exceedingly sparse spectral representations. In *Proceedings of Eurospeech*, volume 6, pages 2687–2690, September 1999.

[119] P. Somervuo. Experiments with linear and nonlinear feature transformations in HMM based phone recognition. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Hong Kong, 2003.

[120] P. Somervuo, B. Chen, and Q. Zhu. Feature transformations and combinations for improving ASR performance. In *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[121] A. Stolcke. STT research and development at SRI-ICSI-UW. In *EARS RT-04F Workshop*, Palisades, New York, November 2004.

[122] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. R. Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, F. Weng, and J. Zheng. The SRI March 2000 Hub-5 conversational speech transcription system. In *Proceedings of NIST Speech Transcription Workshop*, College Park, MD, 2000.

[123] S. Tibrewala and H. Hermansky. Sub-band based recognition of noisy speech. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, volume 11, pages 1255–1258, Munich, Germany, April 1997. IEEE.

[124] S. van Vuuren and H. Hermansky. Data-driven design of RASTA-like filters. In *EUROSPEECH*, volume 1, pages 1607–1610, Rhodes, Greece, September 1997. ESCA.

[125] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989.

[126] W. C. Ward, G. W. Elko, R. A. Kubli, and W. C. McDougald. The new varechoic chaber at AT&T Bell Labs. In *Proceedings of the Wallace Clement Sabine Centennial Symposium*, pages 343–346, Woodbury, NY, USA, 1994. Acoustical Society of America.

[127] R. M. Warren and J. A. Bashford, Jr. Intelligibility of 1/3-octave speech: Greater contribution of frequencies oustide than inside the nominal passband. *Journal of the Acoustical Society of America*, 106:L47–L52, 1999.

[128] R. M. Warren, K. R. Reiner, J. A. Bashford, Jr, and B. S. Brubaker. Spectral redundancy: Intelligibility of sentences heard through narrow spectral slits. *Perception and Psychophysics*, 57(2):175–182, 1995.

[129] A. R. Webb and D. Lowe. The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis. *Neural Networks*, 3:367–375, 1990.

[130] H. Yang, S. V. Vuuren, S. Sharma, and H. Hermansky. Relevance of time-frequency features for phonetic and speaker-channel classification. *Speech Communication*, 31:35–50, 2000.

[131] S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of ARPA Human Language Technology Workshop*, 1994.

[132] K.-H. Yuo and H.-C. Wang. Robust features for noisy speech recognition based on temporal trajectory filtering of short-time autocorrelation sequences. *Speech Communication*, 28:13–24, 1999.

[133] F. Zheng and J. Picone. Robust low perplexity voice interfaces. www.isip.msstate.edu/projects/robust_low_perplexity/html/performance.html, 2001.

[134] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke. On using MLP features in LVCSR. In *Proceedings of the International Conference on Spoken Language Processing*, Jeju, Korea, 2004.

[135] Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan. Incorporating Tandem/HATS MLP features into SRI's conversational speech recognition system. In *Proceedings of the EARS RT-04F Workshop*, Palisades, New York, November 2004.