Automatic Vowel Classification in Speech

An Artificial Neural Network Approach Using Cepstral Feature Analysis

Final Project for Math 196S; Dr. Garrett Mitchener, Prof.

Peter Merkx, Jadrian Miles

pm18@duke.edu, jadrian@math.duke.edu

Department of Mathematics, Duke University, Durham, NC, USA

April 26, 2005

Abstract

The recent resurgence of interest in artificial neural networks as a speech processing tool motivates the present investigation. The authors implement a near-realtime system based on a feed-forward artificial neural network that identifies vowels taken from natural speech samples from the TIMIT corpus of American speech. We report an recognition accuracy of 91.5% on a subset of 5 vowel phonemes, with indications for future work with expanded vowel sets.

1 Introduction

Neural networks have received varying degrees of attention over the last 40 years within the field of speech recognition. Recent advances in ANNs have led them again into the spotlight. However, the speech recognition systems employed in personal speech recognition software and even in more advanced systems tend to use Hidden Markov Models either in their place or in conjunction with them [5].

However, HMMs are not equally biologically plausible and have significant shortcomings [1]. The present work explores several neural network designs as the central engine in a vowel recognition system. Training and testing of two types of ANNs are carried out: perceptron networks and feed forward networks.

The neural networks designed by the authors are trained and tested on data preprocessed using Mel-frequency ceptral coefficient (MFCC) feaure analysis. Thus the role of the neural network here is to provide a mapping from 13-dimensional MFCC space the space of vowel phonemes. Variations in the number and characteristics of vowels used for these purposes are explored.

We examine several ANN types and architectures in addition to various training algorithms used for each of type. After preliminary research, we train and test two types of neural networks.

2 Background Theory

The vowel classification procedure described in this paper is based on three fundamental concepts: the IPA vowel classification scheme, mel-frequency cepstral coefficients (MFCCs), and feed-forward artificial neural networks (FF ANNs). In order to explain the process by which an ANN maps MFCCs to vowels, we describe each of these concepts below.

2.1 The IPA Vowel Classification Scheme

The International Phonetic Association maintains the International Phonetic Alphabet, or IPA, a standard set of characters to describe the sounds of human speech. The IPA separates phonemes into consonants and vowels, as well as a few other types of sounds that are not present in English, and then arranges each group according to the way the phonemes are constructed by the speaker. Vowels are described in terms of three parameters: the openness of the speaker's mouth when speaking the vowel, the position in the mouth where the sound is generated, and whether the mouth is rounded when producing the sound. See figure 1.

Our procedure treats these three parameters as dimensions of a phonological "vowel space", and assigns coordinates as shown in figure 1, with roundness values of 1 (unrounded) or 2 (rounded). A vowel then corresponds to a vector in this space, and vectors with small difference describe vowels that are similar

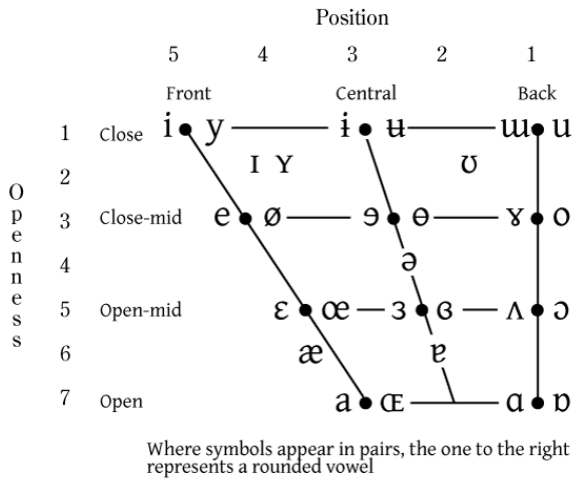


Figure 1: The IPA describes vowels in terms of three parameters: openness, position, and roundness [2].

to each other. Note that this space characterizes the process of *creating* vowel sounds, rather than the properties of the sounds themselves. This is because the true differences between vowels are based on their construction in the mouth of the speaker; the sounds representing all vowels are actually quite similar. One fundamental task that a human must undertake when listening to speech is the inverse mapping from sounds to oral constructions, and from there to phonemes, thereby interpreting the sounds. Our procedure attempts to mimic this process.

2.2 Mel Frequency Cepstral Coefficients (MFCCs)

A classic graphical description of an audio signal, especially one generated by speech, is the so-called "voiceprint", a two-dimensional spectrogram that plots time horizontally, frequency vertically, and intensity at a given time-frequency coordinate with some color scheme. In the example spectrogram below (figure 2), frequencies with higher intensities are shown in black, while unexpressed frequencies are white.

The process of creating a spectrogram involves the Fourier transform, a function that maps a periodic signal in the time domain into a collection of amplitudes or intensities, called a spectrum, in the frequency domain. An audio signal is a one-dimensional stream of data, $i(t)$: for each point in time, it describes an acoustic pressure. We wish to describe the "instantaneous" frequency spectrum at some time T; this is a vertical slice of the spectrogram.

Since the Fourier transform only works on a sequence of data points, we sample a short segment of the signal about time T by multiplying it by a windowing function. One common windowing function is the

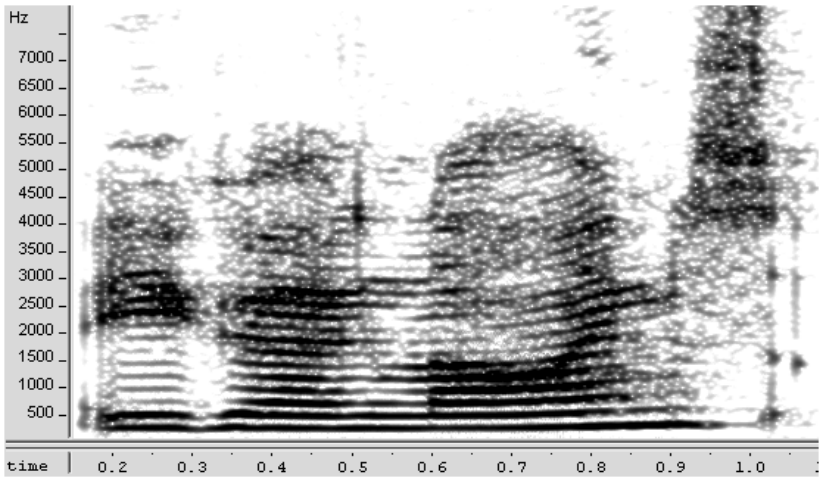
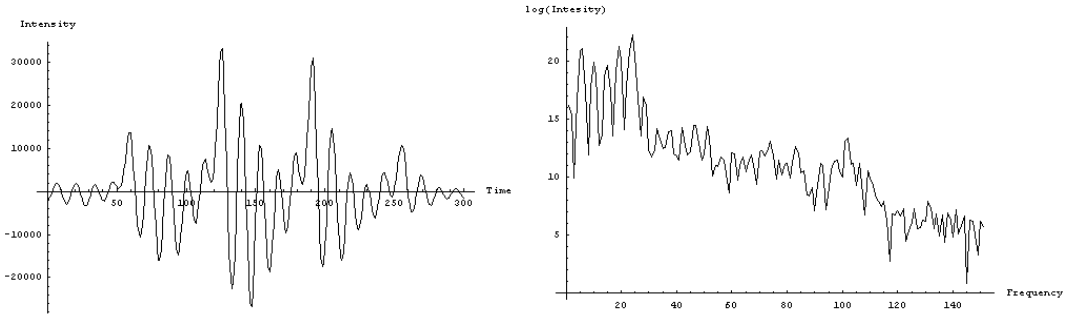


Figure 2: The spectrogram of a short speech sample. Note the frequency bands, which MFCCs can compactly describe.



(a) An audio signal

(b) The spectrum of the signal

Figure 3: A short audio signal and its spectrum, log-scaled.

Hamming window, given by the following equation [8]:

$$W(t) = \frac{46}{\sqrt{1691/2}} * \left(\frac{25}{46} - \frac{21}{46} * cos(2\pi t)\right) \tag{1}$$

The normalization factor at the left of the Hamming window equation is chosen so that the mean square of the function is 1.

Performing a Fourier transform on the windowed signal gives us a snapshot of the frequencies in the signal at time T. To create the spectrogram, we simply move T along the length of the signal in small steps, windowing the signal and generating a spectrum at each step. The spectrogram is a collection of all these spectra arranged side-by-side.

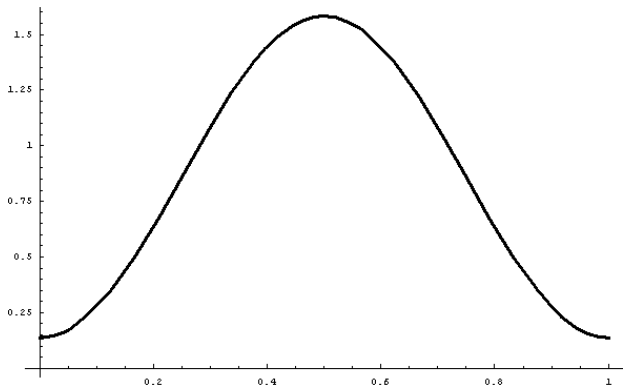


Figure 4: The Hamming window.

Dealing directly with the massive data set that the spectrogram represents is obviously not a practical way to algorithmically analyze a speech sample. We would instead prefer a compact summary, in the form of a relatively low-dimensional *feature vector*, of the perceptually meaningful characteristics of the sound at each point in time. Experimentally established feature vectors include linear prediction coefficients and mel-frequency cepstral coefficients [7]; we choose the latter of these.

While the name "mel-frequency cepstral coefficients" seems obscure, it is actually very descriptive. The mel frequency scale grew out of the early observation that humans can differentiate low-frequency sounds better than high-frequency ones; the gaps between tones that listeners judge to be equally spaced grow larger as frequency increases. Thus the mel scale, in which the distance between tones is approximately linear with their perceptual distance, is a logarithmic transform of the ordinary frequency scale citewikimel.

The cepstral domain is the result of a Fourier transform of the logarithm of *frequency* data for a sound—the frequency domain of the semilogged frequency domain of a time-varying signal. In the same way that the frequency domain indicates periodic patterns in a signal, the cepstral domain reveals precise data regarding time delays, harmonics, and the position of fundamental frequencies citeschroeder04.

MFCCs are thus the result of a Fourier transform of the log-log warped frequency spectrum, and can be plotted over time in a low-resolution plot similar to a spectrogram. While many terms of the Fourier transform may be calculated, only the low-order terms hold phonologically meaningful data. We use the first thirteen MFCCs; though the exact choice of cutoff term is arbitrary, this falls within a small range of traditional cutoffs used in phonological research.

2.3 Artificial Neural Networks (ANNs)

Neural networks are a common tool for attacking classification problems of any kind, with speech recognition only one example. While most modern speech recognition systems employed in the commercial sector use Hidden Markov Models (HMMs) as oppossed to neural networks or sometimes in conjunction with neural networks, this may be the result of the historical success using HMM systems rather than an actual superiority of these systems when using contemporary theory and methods.

Recent advances in the application of neural networks to speech recognition indicate that modified dynamic feed-forward networks capable of dealing with the time warping of phonemes found in actual speech are capable of achieving rates of successful phoneme recognition in continuous speech identical to that of the best HMM systems [1]. Graves *et al.* demonstrate this success and show that such memory-enabled neural network (called Long Short-Term Memory (LSTM) networks) are significantly more biologically plausible than HMM based speech recognition systems.

We explore the results of training two more basic types of neural networks for phoneme recognition: perceptron networks and feed-forward networks. Other network choices available to us for use in this project included radial basis function networks, dynamic neural networks, vector quantization networks, and Hopfield networks.

Each of these possible network designs was evaluated in the context of the vowel classification problem and determined to be suboptimal [6]. The most plausible network choice among these rejected types for our purposes is a dynamic neural network. This type of network does not successfully address the nonlinear time warping of input data found in continuous speech, however, and is thus unsuitable for classifying different phoneme samples [1].

Perceptron networks are the most basic type of neural network and are particularly useful for many classification problems. Perceptrons use the basic achitecture shown in figure 5. The network functions by combining the various inputs with some set of weights. This sum is then used as input for a single neuron's activation function. The output of the activation function is then taken to be the output of the network.

Perceptrons with multiple outputs are composed of several independant perceptron networks each determining the value of a single output. That is, if the output is a three-dimensional vector (X_1, X_2, X_3), then each X_i is computed by a separate network and the final output vector is the combination of these outputs.

Feed-forward networks function on a similar set of principals to those employed by perceptron networks. However, they may have hidden internal nodes spread over many layers as shown in figure 6 below.

This allows the network to define a highly nonlinear mapping from the input space to the output space. There are many known algorithms for use in training this type of network including backpropagation,

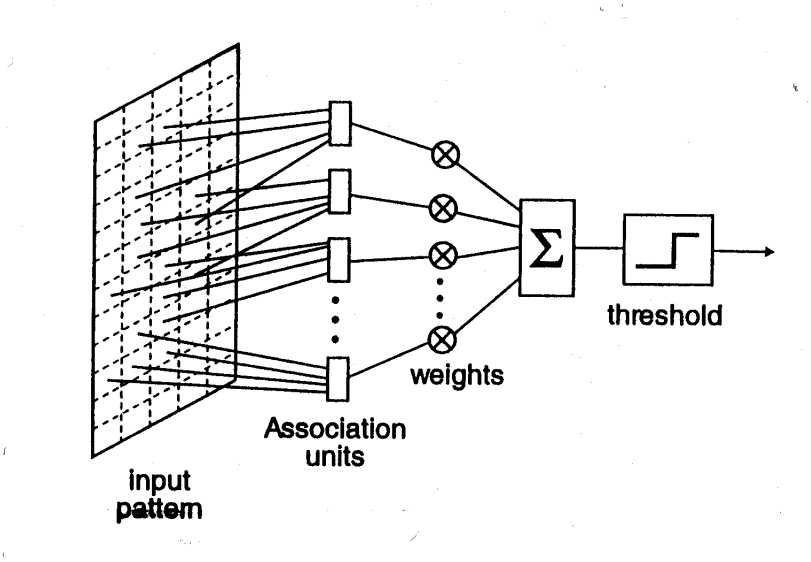


Figure 5: A Rosenblatt perceptron.

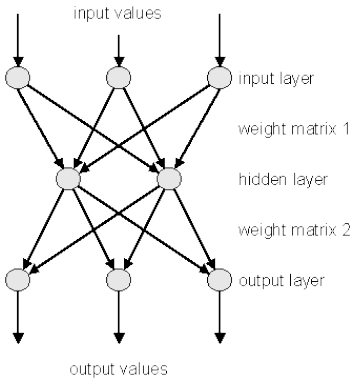


Figure 6: A feed-forward ANN.

steepest-descent, Gauss-Newton, and Levenberg-Marquardt. For our purposes, the latter was choosen due to its optimality in ill-conditioned problems and its quadradic rate of convergence [9].

The optimal number of hidden layers and number of neurons in each layer of the feed-forward networks we trained was determined experimentally and limited by the computational ability of the machines on which the training process was implemented; we ultimately found that a single layer was sufficient to acheive very good recognition rates.

3 Hypotheses

1. Each vowel sound in English is located in a distinct region of some high-dimensional phonological space that maximizes distances between each pair of distinct vowels [3]. Therefore it is possible to programmatically segment this space to classify vowel sounds.

2. The classification parameters used to describe vowels in the International Phonetic Alphabet approximate the phonological space in which vowel sounds are naturally described.

3. The first thirteen mel-frequency cepstral coefficients of a vowel sound describe its unique properties sufficiently to differentiate it from other vowels [8].

4. Certain artificial neural networks are appropriate tools for evaluating the MFCC profiles of extracted sound samples and classifying them based on this data. In general, an ANN learns an implicit mapping from its input space (in our case, MFCC data) to its output space (vowel classes) that matches its training data [4]. This mapping may then be applied to additional input data in order to classify them.

5. Training the ANN with data from multiple speakers will force it to correlate speaker-independent common properties of vowel sounds. Thus it will be able to classify vowels spoken by individuals whose voices were not included in the training data set.

4 Experimental Setup

4.1 Data Gathering and Preparation

All our training and testing data were extracted from the TIMIT corpus, licensed from the Linguistic Data Consortium at the University of Pennsylvania. We focused on eleven vowel sounds at first, using Unix shell scripts to parse the TIMIT phoneme annotation files and compile a list of every use of each vowel in the entire corpus. We then used Matlab scripts to extract these vowel samples, convert them to MFCC arrays, and subsample and compile them into one array per vowel. The training arrays had one MFCC vector per instance of the vowel in the corpus, while the testing arrays had ten vectors, gathered from adjacent Hamming windows, per sample. Figure 7 summarizes the amount of data we gathered; in total, there were 29999 training samples and 11149 testing samples.

Vowel	IPA	Example	Train	Test
ae	æ	bat	3979	1407
ah	ʌ	but	2004	772
ao	œ	bought	2852	1125
aw	a	bout	729	216
ax	ə	about	3745	1461
eh	ɛ	bet	3449	1291
ih	ɪ	bit	4085	1397
iy	i	beet	6135	2383
ow	o	boat	2120	775
uh	ʊ	book	399	168
uw	u	boot	502	154
Total			29999	11149
Core 5			17213	6530
Extended 8			25476	9434

Figure 7: Sample sizes and totals for eleven, eight, and five vowels.

4.2 Training the ANN

From the initial set of eleven distinct vowels, we trained and tested the ANN with subsets varying in length from five to the full set. Data was imported into Mathematica after preprocessing as described above. We generated two parallel matrices per vowel sound from the imported MFCC arrays. The first matrix stored each of the training input entries (a single 13-dimensional vector of MFCCs). The parallel matrix contained "vowel masks", vectors of length corresponding to the number of vowels under consideration, with a value of .8 in the entry corresponding to the vowel being sampled and .2 in all other entries. .2 and .8 were used, rather than 0 and 1 as in a traditional bitmask, to avoid backpropagation errors with the sigmoid neuron activation functions.

Once the data was prepared in this fashion, the Mathematica Neural Networks add-on package was used to train two different types of neural netorks: perceptron and feed-forward networks.

Several perceptron networks, each with a different structure, were trained with the entire input and output data set and allowed to reach a steady state in training. Ultimately, testing indicated that perceptrons gave suboptimal recognition results.

Several feed-forward networks with varying numbers of internal nodes and layers of internal nodes were then created by passing the training data as parameters for initialization. We specified a sigmoid as the neuron activation function. The number of nodes in each case was limited by the computational power of the machines on which the the training algorithms were run (1–2 hidden layers, 2–40 nodes per layer).

Each feed-forward network was iteratively trained on between 1 and 10 randomly generated subsets, each equally probalistically representative of each vowel (between 200 and 4500 entries) of the origional training data.

The number of iterations over which each training was allowed to run was again limited by the computational power of the machine the simulation was run using and the time constraints of the investigators for some network configurations. The most computationally intensive networks used more than one internal layer of nodes.

4.3 Testing the ANN

Each neural network created during training was tested on the entire portion of the testing data preprocessed for use with Mathmatica. Our evaluation of each network's success was based on the number of successful guesses defined according to two criteria:

1. The best match for the output corresponding to a single MFCC input vector was determined simply by the index of the maximum value of the output array. For example, $.2, .1, .9, .2, .1$ indicates that the corresponding MFCC input was the third vowel.

2. Since each vowel sample was represented by a series of ten MFCC vectors from adjacent Hamming windows, the statistical mode of the ten best matches was considered the final classification of the vowel sample.

These sets of outputs was then compared with the manually transribed values from the corpus for agreement. Evaluation of training using the criteria allowed the success of ANN phoneme recognition to be based on several Hamming windows of a given sample as opposed to only one as in earlier iterations of our technique. This means that network output was determined less strictly giving higher success rates; on average, recognition rates improved by eight percentage points when considering ten MFCCs per test sample rather than just one.

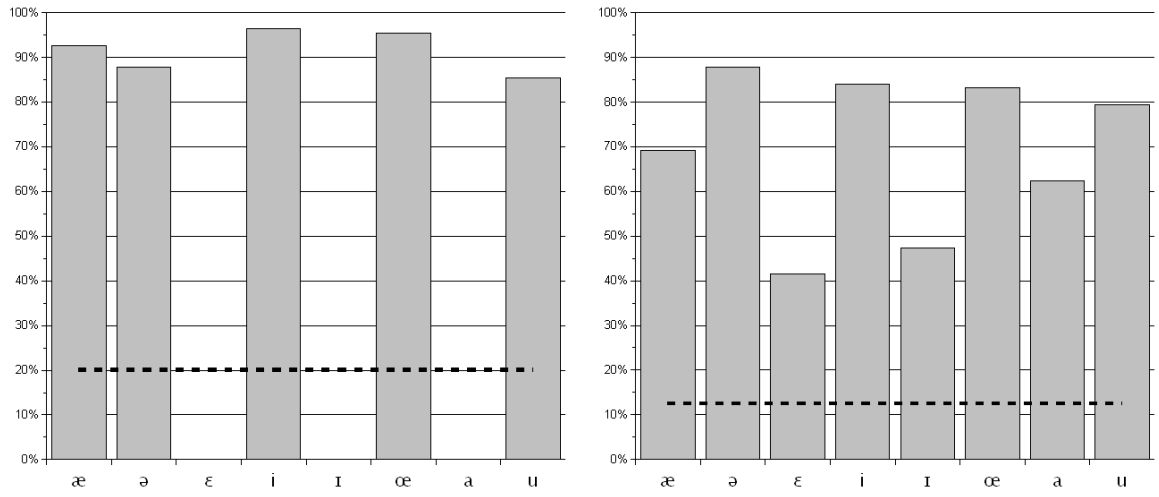
5 Results

Feed forward networks had a significantly higher overall accuracy than perceptron networks. Even so, perceptron networks were able to achieve an average accuracy for 5 vowels of 60%. Perceptrons failed to converge completely during training and were inconsistent in performance across vowels. Training the single-layer feed-forward ANN with 1000 training samples took an average of two minutes per iteration, with quadratic error convergence. Eight iterations (hence about fifteen minutes of training time) were usually sufficient for optimal recognition rates.

Each network architechture resulted in a different final accuracy rate. Additionally, the number of vowels used to train the network significantly affected the accuracy. Among the lowest scores in accuracy for all

networks were dipthongs. All dipthongs used to train networks had approx. 20% lower accuracy rates than non-dipthong vowels. The networks also displayed confusion between the ambiguous 'ih', 'eh', and 'uh' sounds that resulted in the lowest individual vowel accuracies.

The maximum average accuracy achieved for any network was 91.5%. This network had a single layer of 28 internal nodes and was trained to distinguish between five unique vowels. The network was trained for a total of 15 iterations using 3 subsets of the training data. The vowel the network recognized with the greatest reliability was 'iy' at 96.4%. The lowest accuracy for a single vowel with this network was 'uw' at 85.3%. Our success rates are summarized in figure 8 below.



(a) Five vowels, one hidden layer of 28 nodes

(b) Eight vowels, two hidden layers of 17 nodes

Figure 8: Recognition success rates with a feed-forward ANN. The dashed lines indicate the baseline success rate acheived with random guessing.

6 Discussion

The shortcomings of perceptron networks in their use as a speech recognition tool are unsuprising considering that they are incapable of accurately representing highly nonlinear maps as closely as feed forward networks.

The overall increase in accuracy achieved with the reduction of the number of vowels is unsuprising. However, the final rate of successful recognition when using 5 vowels indicates that the feed forward neural network is capable of successfully defining a map between the MFCC feature space and the output vowel space.

Many of the vowels the network was tested on in the larger sets of vowels were dipthongs and hence time

dependent and at any short time window highly similar to the component vowels of which the dipthong is comprised.

The high success rates when training and testing using 5 vowels indicate that this is a powerful method that could be used for general phoneme recognition when paired with statistical techniques for dealing with time dependant signals.

The failure of the ANNs used here to distinguish between certain pairs vowel sounds with a high degree of accuracy may indicate actual ambiguity in the feature characteristics of these sounds. The source of this error may be based in actual ambiguities between these vowels that are made clear to speakers by context, or, alternatively, the error may result from the inadequacy of MFCCs to complete characterize phonemes. It may be possible to resolve this ambiguity by including the time-derivatives of the MFCCs as an additional 13 values for any input to the network. Doing so may also remove the above mentioned shortcoming of the ANN to recognize dipthongs.

6.1 Strengths

- Despite using static, contextless training and testing data, we acheived a 91.5% recognition rate.
- Fully training the single-layer ANN takes about fifteen minutes.
- Once the ANN is trained, the recognition process, from speech signal to ANN output, is very fast, and could even be made real-time.
- Our ANN involves significantly less computational complexity than HMM-based procedures.
- While ANNs are known to be very inaccurate representations of biological neural networks, they are still more physiologically plausible than HMMs as a model for human perception.

6.2 Weaknesses

- The set of vowels we can successfully recognize is extremely limited; out of our initial set of eleven, we were only able to acheive good performance with five.
- Our general approach to this problem, using MFCCs and a feed-forward ANN, is poor at recognizing ambiguous vowels, like the short ‘e’, ‘i’, and ‘u’ sounds.
- Processing MFCCs contextlessly prevents our technique from recognizing time-dependent phonemes like dipthongs. In general, ignoring the time-dependent behavior of our input data ignores a significant feature that is likely important in natural speech recognition.

- While time-derivatives of MFCCs are sometimes used in recognition schemes, we left them out, again limiting our neural net's understanding of the time context of its input.

7 Future Work

Potential expansions on the procedure developed in this paper include:

- We could more effectively process the ANN's classification-mask output with mature statistical analysis, rather than just the mode of the maximum classification value. We could also use a second neural net, static or dynamic, to process the time-varying output.

- Alternately, rather than using the IPA vowel classification parameters to construct a three-dimensional perceptual space, we could adopt the more successful 14-dimensional Chomsky-Halle binary phonolgoical feature space as in [4]. This would allow us to use more mature output filtering techniques, rather than the simple statistical mode that gave us our best results in this experiment.

- We could use dynamic training data to provide context for the input. This could be accomplished through the inclusion of time-derivatives of the MFCCs or with a dynamic feed-forward ANN.

References

[1] Alex Graves, Douglas Eck, Nicole Beringer, and Juergen Schmidhuber. Biologically plausible speech recognition with LSTM neural nets. In *Proceedings of the First International Workshop of Biologically Inspired Approaches to Advanced Information Technology*, pages 127–136, 2004.

[2] International Phonetic Association, Department of Theoretical and Applied Linguistics, School of English, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. *International Phonetic Alphabet chart*, 1996.

[3] Keith Johnson. Adaptive dispersion in vowel perception. *Phonetica*, 57:181–188, 2000.

[4] Simon King and Paul Taylor. Detection of phonological features in continuous speech using neural networks. *Computer Speech and Language*, 14(4):333–353, October 2000.

[5] R. J. Mammone, editor. *Artificial Neural Networks for Speech and Vision*. Chapman & Hall, New York, 1993.

[6] Wolfram Research. Website: http://documents.wolfram.com/applications/neuralnetworks.

[7] Manfred R. Schroeder. *Computer Speech: Recognition, Compression, Synthesis*. Number 35 in Springer Series in Information Sciences. Springer, second edition edition, 2004.

[8] Rivarol Vergin, Douglas O'Shaughnessy, and Azarshid Farhat. Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. *IEEE Transactions on Speech and Audio Processing*, 7(5):525–532, September 1999.

[9] N. Yamashita and M. Fukushima. On the rate of convergence of the levenberg-marquardt method. *Computing*, 15:239–249, 2001.