SPEAKER-INDEPENDENT PHONEME RECOGNITION IN A CONTINUOUS SPEECH CONTEXT USING TIME-DELAY FEED-FORWARD NEURAL NETWORKS

Bret D. Whissel

Artificial Neural Networks
Florida State University

ABSTRACT

In scaling the problem of automatic speech recognition down to a smaller sub-problem — that of recognizing individual utterances — this exploration into a connectionist model of phoneme recognition stems from the assumption that phonemes should be identifiable as part of a surrounding context. By including contextual clues in the form of multiple contiguous spectral speech frames, a feed-forward neural network trained through error back-propagation will have more information from which to derive phonemic classifications. An additional assumption is that multiple speaker data is not only desirable, but even necessary for proper training of a robust recognition system.

1. INTRODUCTION

Speech recognition as a field of research is hardly new. From the 1950's, it was realized that recognition of speech by machines could become the most efficient means of interaction with systems. Speech is the most natural form of human communication requiring no special training or expertise, unlike typing, for example. If machines could be built which understand speech, then little additional training would be required for a person to have access to information, communication, and control functions currently provided by auxiliary human translators. Progress has been made in many facets of this recognition problem, but in over 50 years of research, the complete solution to the speaker-independent large-vocabulary understanding of continuous speech eludes us.

1.1 *History*

Ainsworth (1988) provides some discussion of the various approaches that have been applied to the speech recognition problem date. Early efforts of the 1950's attempted to exploit knowledge of acoustic phenomena, recognizing that different utterances had characteristic sound spectrogram patterns. By dividing a speech signal into a number of frequency bands using analog circuits, energy concentrations in each band could be correlated with stored patterns for spoken digit recognition. While these early systems claimed recognition accuracies of 94% to 99% for a single speaker, the accuracy for additional speakers could drop to as low as 50%. Another early attempt involving the classification of phonemes tried to parameterize the means by which a phoneme was generated in the vocal tract. It was found that 12 binary features could be defined to differentiate between the 50-odd phonemes of English.

It came to be understood that comparing simple acoustic patterns could not account for the wide variety of intensity and duration values found in everyday speech. In the 1960's, the concept of signal transformations was borrowed from vision researchers, whereby comparison of the transformed signals could be compared with templates to accomplish recognition. While some progress was again made using this technique, success stalled as researchers attempted to identify units larger than isolated vowels. Also during this period, Ainsworth reports early attempts at applying neural networks to speech recognition by Talbert *et al.* (1963) and Nelson *et al.* (1967).

The next phase of advancement came

in the 1970's, when linguistic knowledge is added to the recognition task. While all the information for a particular utterance may be located in the acoustic signal, the understanding of that utterance depends upon the listener's knowledge of the language and expectation based on previous and projected context. Systems developed during this time frequently incorporate dictionaries, language syntax, or other forms of linguistic information into recognition systems, in conjunction with acoustic analysis.

Trends of the recent decade continue to include linguistic information in speech recognition, enhanced by time-scale distortion for normalizing variances in that domain. Stochastic models, hidden Markov models (HMMs) in particular, are currently among the most popular tools in recognition research today. A conference in early 1995 sponsored by the Advanced Research Projects Agency (ARPA) consisted almost entirely of papers pertaining to HMMs.

1.2 *Current Research*

Speech recognition is a vigorous field of research with hundreds of participants world-wide. A recent query of the `comp.speech` archive at Carnegie Mellon University (`http://www.speech.cs.cmu.edu/comp.speech`) listed 598 links to speech-related network resources, if this can be any indication of the level of interest. With so much activity, it would be no surprise that there are many and varied viewpoints on the successful approaches to the speech recognition problem. Waibel and Lee (1990) have classified the major efforts of researchers into four categories.

Template-based speech recognition stores prototypes for speech patterns in the recognizer's lexicon. Unknown utterances are compared to each template, and the closest match within some tolerance is selected. Templates may span words and short phrases or be more narrowly-defined. Processing requirements become prohibitive as the recognition dictionary grows large, but the advantage is that small-scale variances at the phoneme level are less likely to affect the final result.

Knowledge-based approaches to speech recognition attempt to codify experts' knowledge explicitly into rule-based expert systems. A wide range of linguistic and acoustic expertise is available, but combining the many layers of knowledge has been difficult. While knowledge-based attempts have themselves been largely unsuccessful, information gained from experts has proved valuable to other endeavors.

Stochastic methods in speech recognition are dominated by the highly successful hidden Markov model (HMM). Both temporal and spectral variabilities of speech signals are captured by HMMs. One factor limiting the accuracy of these models is the assumption that signal probabilities depend upon the current state only, and not on previous context. Also, a large corpus of text is required to train such models. Nonetheless, most success in speech recognition today is achieved by using some form of HMM. ARPA (1995) suggests that current HMM implementations may be limited in accuracy to vocabularies of only 50 000 to 60 000 words, beyond which spectral variability degrades performance.

Finally, connectionist, or parallel distributed processing models of speech recognition employ artificial neural networks. The ability of networks to classify highly complex input vectors makes them well-suited to the variability inherent in speech signals. Neural networks are the newest tools in the speech recognition arsenal, and there is yet much to be learned about how they may be applied to the recognition task.

2. MOTIVATION

The author's emphasis for this project is to become familiar with some of the terminology and methods surrounding basic research in speech recognition using a neural network as a recognition tool. Speech recognition has been a long-time interest, and it seemed appropriate to test the viability of a neural network system on a rather naïve approach to this problem. Since it is known that utterances have characteristic spectral distributions, and that a human can be trained to read a sound spectrogram (Waibel & Lee, 1990), it is logical that a network properly trained should be able to perform a similar task.

One of the unsolved difficulties with continuous speech recognition over single word recognition is that of word separation and time invariance. For this study, this issue has been side-stepped by focusing on phoneme recognition rather than word recognition. Phoneme recognition is more directly tied to the acoustic signal, and relies less on the language modeling as required by a hidden Markov model.

An individual phoneme, while recognized as an entity, may actually be composed of smaller discernable units of sound. It may be that some or all of these components constitute what is essential for identifying the utterance. It is therefore clear that if the speech signal is to be partitioned into small segments in time, that several contiguous segments should be presented to the network as a group for training purposes.

Almost all speech recognition research requires that a speech signal be analyzed using some means of spectral decomposition. The author chose to use a tool with which he had some familiarity and confidence, a Fourier transform. Many other spectral decompositions and analysis methods exist and have been employed in speech recognition research. Papamichalis (1987) provides several interesting methods for the coding of speech signals, some of which may be useful for feature development. An algorithm for the Fast Fourier Transform (FFT) is found in Press *et al.* (1986) with a complete treatment given by Oppenheim and Schafer (1975).

Training data should be derived from naturally spoken contexts, rather than single word utterances because many phonemes assume slightly different sounds depending on the phonemes surrounding them. To be precise, a phoneme is slightly more than a unit of sound. As described by Yannakoudakis and Hutton (1987), a phoneme is actually a symbolic identifier for a small cluster of sounds. Changing the identifier changes the sense of the utterance, as in "/t/in" versus "/p/in". Phones, on the other hand, are more tightly coupled to the actual sound, and can display the variability that phonemes possess in different contexts. As an example, the phoneme /l/ in the word "/l/et" takes on a slightly different phone when used in the context "peop/l/e". This study attempts to classify the phonemes from their various context-dependent phones, and therefore the training data should attempt to provide instances of phonemes in different positional contexts.

There have been some successes in producing recognition systems that have been tuned to a small number of speakers. To be truly robust as a recognition system, it is hoped that speaker independence can be achieved by training from data provided by a number of different speakers.

3. METHOD

3.1 *The Network*

The feed-forward back-propagation network was chosen for the recognition task for several reasons. Class discussions and readings from Haykin (1994) have de-mystified the operation of these networks, and provided a rudimentary foundation for understanding error minimization in a complex n-dimensional space. However, experience has shown that hidden misconceptions are brought to light when faced with implementing one's understanding in the form of a computer program. Indeed, this has proved true more than once during the course of this investigation.

In addition, feed-forward networks are relatively simple to implement, and the connections of these networks are more easily traced and understood than, for example, recurrent networks. Since one goal of this study is to learn more about the difficulties associated with speech recognition, having a tool from which feature data can later be extracted could be very valuable. Nevertheless, it was tempting to wonder how an Adaptive

Resonance Network might have been employed somewhere along the way.

In implementing the back-prop network, the output (squashing) function $\phi(v)$ with output range $[-A, A]$ was chosen since it leads to quicker learning in many cases, as discussed in Haykin (1994).

$$\phi(v) = A\tanh(Bv) = A\left(\frac{e^{Bv} - e^{-Bv}}{e^{Bv} + e^{-Bv}}\right)$$

with derivative function (for back-prop)

$$\begin{aligned}\phi'(v) = \frac{d}{dv}A\tanh(Bv) &= AB\,\mathrm{sech}^2(Bv)\\ &= \frac{4AB}{\left(e^{Bv} + e^{-Bv}\right)^2}.\end{aligned}$$

where $A = 1.716$ and $B = 2/3$ are constants. A plot of these functions is given in Fig. 1.

The other major component of a back-propagation implementation is the learning rule. For this study, the generalized delta rule is used for weight adjustment given by the equation

$$\begin{aligned}w_{ji}(n+1) = w_{ji}(n) + \alpha\,[w_{ji}(n) - w_{ji}(n-1)]\\ + \eta\,\delta_j(n)y_i(n),\end{aligned}$$

where η is the learning rate parameter, and α is a momentum term. These parameters are modifiable so that different values may be selected to optimize learning.

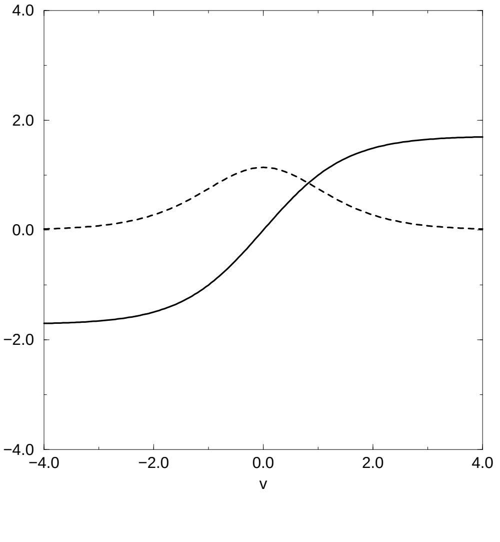

Fig. 1. A plot of the output function $\phi(v)$ and its derivative $\phi'(v)$.

Since several network configurations were to be tested, the network program was written with sufficient flexibility for varying the network architecture over the course of several experiments. By changing parameters in an experiment configuration file, a new network configuration consisting of different numbers of layers or nodes per layer could be built, and momentum and learning rates could be assigned. Once the network software was built, it was tested and debugged using the XOR problem until it performed as expected. The network software will also produce a file containing its complete state at selectable training intervals so that training could be halted and resumed with little computational penalty.

3.2 *Phoneme Classes*

Back-propagation is a supervised learning technique requiring that input patterns have known classifications. The job of the network is to learn how to reproduce a classification given a similar input pattern. The next task was to generate appropriate datasets for training.

A list of phonemes was found through the `comp.speech` archives at Carnegie Mellon. Not all sources agree on the number of unique phonemes for English and American English. Rather than try to resolve the debate, this author considered a subset of 39 phonemes. These phonemes, their International Phonetic Alphabet equivalent, and contextual sound are shown in Table 1. This subset was chosen by occurrences in a list of the most common words of English literature as derived from a word count of the archives of Project Gutenberg. The word and phoneme list are not necessarily definitive, but they are sufficient for the purposes of this study.

From the list of most common words, sentences were constructed with the goal of using as many phonemes as possible with the fewest number of words. A secondary goal was to avoid common word configurations

Table 1. Phonemes and their IPA equivalents with examples.

Phoneme	IPA	Example		Phoneme	IPA	Example	
&	æ	and	/&nd/	j	j	yet	/jet/
A	ɑ	was	/wAz/	k	k	could	/kUd/
e	ɛ	yet	/jet/	l	l	let	/let/
i	i	she	/Si/	m	m	my	/maI/
I	I	it	/It/	n	n	one	/wVn/
O	ɔ	all	/Ol/	p	p	point	/poInt/
u	u	you	/ju/	r	r	for	/fOr/
U	ʊ	good	/gUd/	s	s	such	/sVts/
V	ʌ	but	/bVt/	t	t	but	/bVt/
aI	ɑi	my	/maI/	v	v	voice	/voIs/
aU	ɑʊ	about	/VbaUt/	w	w	was	/wAz/
eI	ɛi	they	/DeI/	z	z	was	/wAz/
oI	ɔi	point	/poInt/	D	ð	they	/DeI/
oU	ɑʊ	own	/oUn/	T	θ	thought	/TOt/
3R	ɜʀ	turn	/t3Rn/	N	ŋ	being	/biIN/
b	b	but	/bVt/	S	ʃ	she	/Si/
d	d	good	/gUd/	Z	ʒ	vision	/vIZVn/
f	f	for	/fOr/	tS	ʧ	such	/sVtS/
g	g	good	/gUd/	dZ	ʤ	just	/dZVst/
h	h	her	/h3R/				

which might lead to word elision. But since target of this project is to study phoneme occurrence in natural speech, the sentences needed to be more than word lists: they had to follow some grammatical rules, even if they were semantically nonsensical. The resulting sentences are rather odd, and perhaps even mildly amusing, but they fall within the prescribed objectives. Table 2 lists the sample sentences and their phonetic spellings.

3.3 *Speech Samples*

Once the phonemes were chosen and sentences were written, samples of speakers saying the sentences were recorded onto a cassette tape. It would have been possible to record the samples in the digital domain directly, but should it be decided that a different digital sampling rate might be desirable, having a cassette recording would avoid the necessity of obtaining new readings from the speakers. Speakers were allowed rehearsal readings so that speech would sound natural, in spite of the awkwardness of the sentences. In all, eight speakers were recorded for each of the three sentences, five female and three male speakers.

3.4 *Feature Development*

In speech recognition research, a signal is often divided into short stretches of time called frames, usually 10 ms in length. Although speech is a continuously varying signal, it is assumed that the spectral information changes little over the span of the frame owing to the relatively slow movement of articulators in the vocal tract. After spectral analysis, each frame is considered a spectral snapshot of the speech signal.

When the cassette recordings were finished, the speech samples for each of three sentences were digitized at 16 000 Hz using the audio device of a Sun Sparc5 workstation. This sample rate was chosen so that the FFT might provide a generous frequency resolution without blending the sounds from too great a stretch of time. A 256-point FFT resolves the sampled signal into 128 frequency bins from 0 Hz to 8000 Hz. An FFT of this length spans 16 ms, which is slightly longer than the standard speech frame.

Digitized speech samples were stored in files, one file per sentence per speaker. Figure 2 displays a sampled signal. A spectral analysis program was written to convert the raw sample data into frames of spectral power densities. The transformed data were also converted into a PostScript form for printing and display. The original output of the FFT shows a tremendous dynamic range of several orders of magnitude. Since the neural network software is sensitive to linear changes in input, it would be necessary to normalize the FFT data in some way.

As with many other human senses, the auditory system is sensitive to logarithmic changes in input, rather than linear changes. This is true for both power and frequency. Because it is not without precedent, FFT frequency bins f_i were scaled by $\log_{10}(1 + f_i)$ to produce a more linear distribution of power. Power values scaled in this way may still lie beyond the input range of a network input unit. Further nomalization would be required.

The normalization curve was derived by calculating the mean and standard deviation of logarithmic power for each frequency bin. An upper bound of the scale for each bin was arbitrarily set to two standard deviations above the mean, and the lower bound was set

Table 2. Sample Sentences and Their Phonemic Spellings.

1. But for the good of all such people she thought with one voice that her own vision was not so just.

   bVt fOr DV gUd Vv Ol sVtS pipVl Si TOt wIT wVn voIs D&t h3R oUn vIZVn wAz nAt soU dZVst

2. And yet he could let you in if the point about which they turn was being taken through my great project.

   &nd jet hi kUd let ju In If DV poInt VbaUt wItS DeI t3Rn wAz biIN teIkVn Tru maI greIt prAdZekt

3. Give me change for noticing you.

   gIv mi tSeIndZ fOr noUtIsIN ju

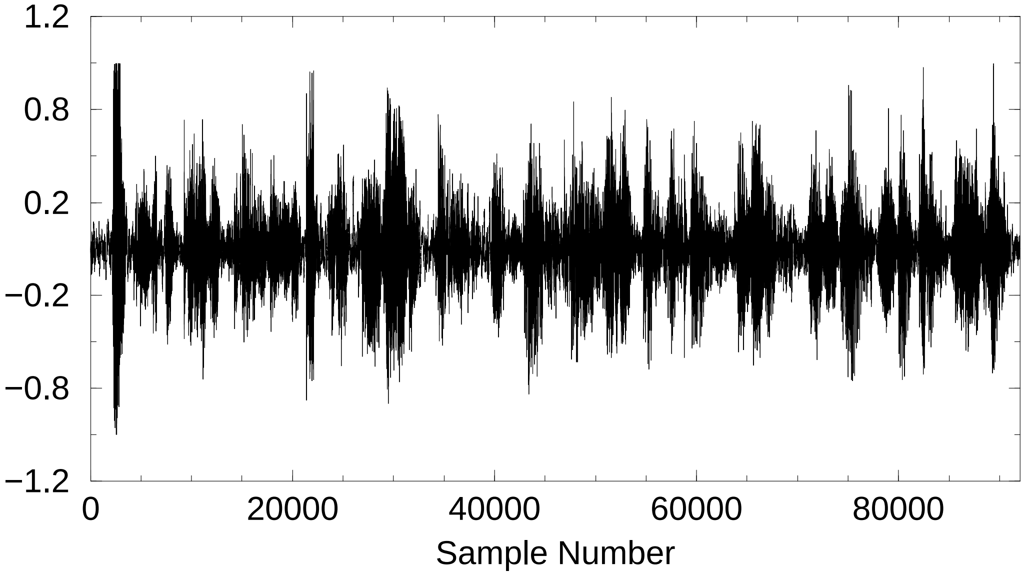

Fig. 2. Sample of a speech signal of phrase “But for the good of all such people she thought with one voice that her own vision was not so just,” spoken by a female speaker.

to one standard deviation below the mean. The logarithmic power data in each frequency bin f_i were then scaled according to the following formula and hard limited to 0 and 1:

$$norm_i = \frac{(f_i - lower_i)}{(upper_i - lower_i)}.$$

This scaling is admittedly *ad hoc*, but not unreasonable.

3.5 *Frame Tagging*

Once the frame data had been prepared, it was necessary to label each frame with its appropriate phoneme value. Additional software was written to assist in the tagging of frames. Each frame of each sentence by each speaker was tagged and written to an annotated frame file, a file which could be read by the network software.

The annotation software reads the original sampled data. A file of frames is initially tagged by distributing the phonemic spelling of a sentence across all of the frames in the file. Each phoneme was therefore initially assigned several contiguous frames. Then, for each phoneme, the data samples represented by the frames tagged by that phoneme were played. The frames allocated to the phoneme were adjusted by listening to the played samples, and making the appropriate adjustments. Once every frame in the file was properly labeled, the annotated frame file was written to disk.

While listening to sample playback, it was sometimes necessary to alter or delete a phoneme. One common example was substituting the phoneme /tS/ for the combination of phonemes /t/ /j/ between the words 'let you'. A phoneme that was commonly deleted was /h/ between the words 'yet he'. Figure 3 shows an example of an annotated frame file with tags.

3.6 *Operation*

At this point, the frame spectral parameters have all been scaled to the range $[0, 1]$, and each frame has been tagged with a single phoneme value. For each experiment, a network architecture is assigned and learning rate and momentum parameters are set from the experiment description file. Also, any number of training sets and cross-validation sets are selected from among the available annotated frame files. Then, until some maximum number of training epochs is reached, network training is accomplished according to the following description.

First, a list of available frames from all training datasets is generated. Then a random frame selection is made and removed from the list. The selected frames' spectral parameters are transferred to the input layer. The input signal is propagated forward through the network. The tag of the chosen frame is used to indicate the output node which should be most active, and error is propagated backward through the network. Weights are adjusted according to the learning rule. Additional frames are selected from the list in this fashion until all frames have been seen. At the end of the epoch, any cross-validation datasets are run through a similar process, except that error is not propagated backward through the network and weights are not adjusted.

Error calculations are according to Haykin (1994) summarized here. Output of node j is denoted by $y_j(n)$ for pattern n, and $d_j(n)$ represents the desired output for that node. For the class of output nodes, the desired output is $A - \epsilon$ if the node is active according to the frame tag, or $-A + \epsilon$ otherwise. The error signal at the output layer is given by the equation

$$e_j(n) = d_j(n) - y_j(n).$$

Training error and cross-validation error are calculated according to the following formula:

$$\mathcal{E}_{t,c} = \frac{1}{N_{t,c}} \sum_{n=1}^{N_{t,c}} \sum_{j \in C} e_j^2(n)$$

where the subscripts t and c denote training or cross-validation datasets accordingly, and C is the class of output nodes.

The experiment description file may also indicate that the network state should be dumped to a checkpoint file after a certain number of epochs. Not only may training be halted and resumed at the most recent checkpoint, but the training progress may also be tracked.

4. RESULTS

Training was attempted using several learning rates, momentum rates, and network architectures, with varying degrees of success. The first experiment to be tried had selected a learning rate $\eta = 0.01$ and a momentum term of $\alpha = 0.8$ and a network configuration of 384–64–39 (three consecutive input frames). The training error is shown in Fig. 4. Because the network had ceased reducing error very early, it was thought that learning and/or momentum rates were set too high. For this reason, training was halted after epoch 70 and resumed with a new learning rate $\eta = 0.001$ and momentum rate of $\alpha = 0.7$. This discontinuity can be seen by the graph in Fig. 4. Because over 2900 sets of frames were available for training during this experiment, 270 training epochs required over 4 hours of compute time on a Sun Ultra 3000 dual processor machine with 128 Mb of physical memory.

Experiment one was run a second time, but with the final learning rates. After 240 epochs, the training error curves became nearly identical. Experiments two and three changed the network architecture to 128–64–39 and 384–128–39, respectively, with the learning rates of $\eta = 0.001$ and momentum factors of

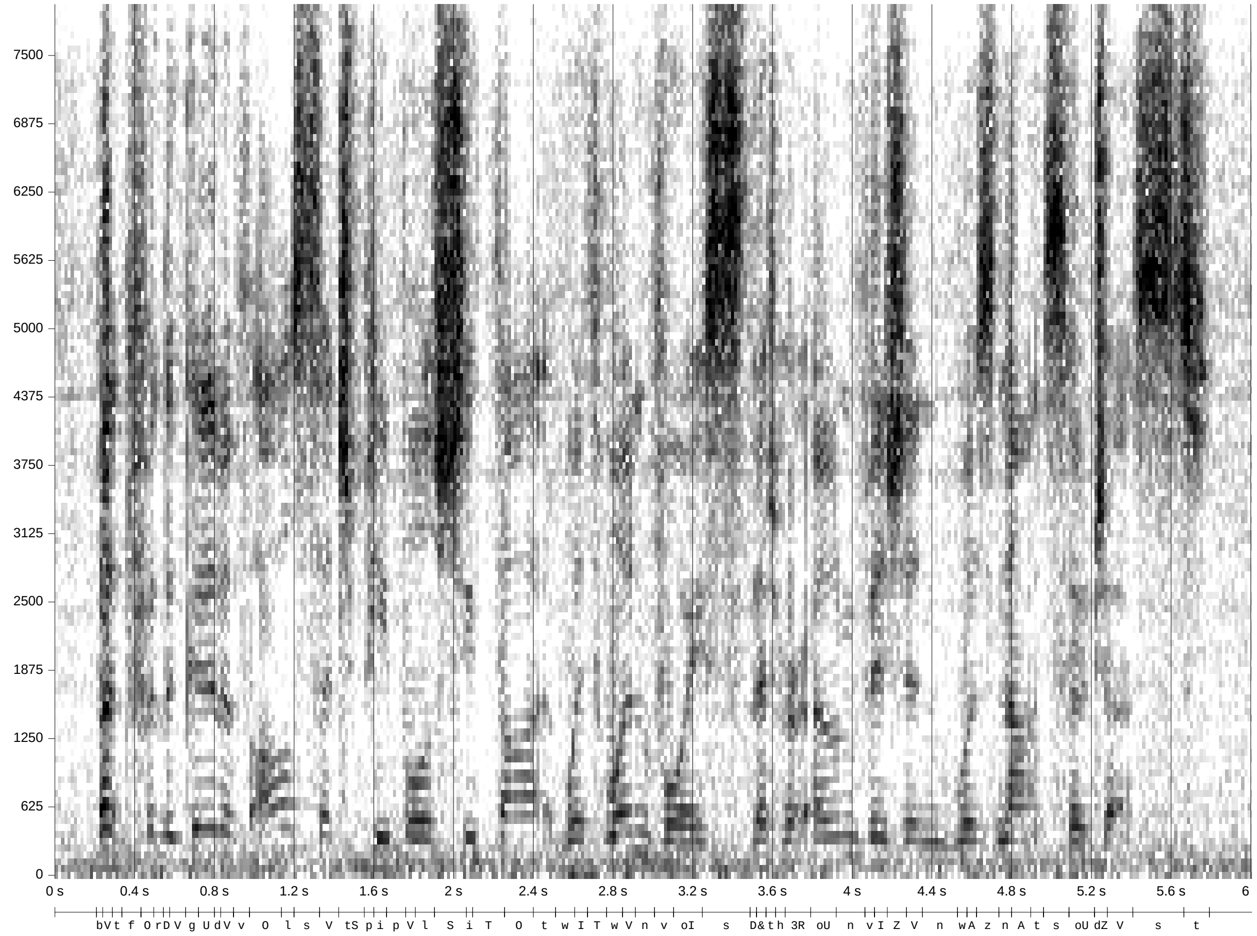

Fig. 3. The speech signal from Fig. 2 after spectral analysis, normalization, and tagging.

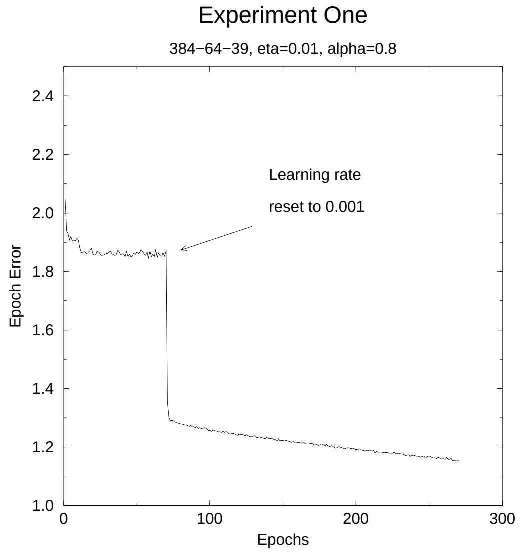

Fig. 4. Graph of the training error in experiment one with a network configuration of 384–64–39 and initial learning and momentum rates of 0.01 and 0.8, respectively. The discontinuity at epoch 70 is due to the resetting of the learning rate.

$\alpha = 0.1$. The training error rates leveled off sooner than in the curve of experiment one.

The evidence of experiments two and three seems to suggest that these network topologies are stressed by the variabilities inherent in the training datasets, and that deeper networks may provide better results. Experiment four, with a network topology of 384–128–64–39 did indeed display a steeper error descent rate than the previous experiments. However, with nearly 67 000 weights to adjust for each presentation of 2900 frames of training data, approximately 37 hours of computer time was required before the epoch training error dropped below 0.5. Training was stopped after epoch 1530 where network performance on the training sets was quite acceptable. However, generalization on a dataset not seen by the network was abominable. Since the calculation of cross-validation error had not been implemented yet, it was not possible to see where over-training had begun. When cross-validation error had been implemented in the network software, experiment four was run again with the curious result shown in Fig. 5.

5. ANALYSIS

Time would not allow continuing the investigation into the cause of the strange perpetually increasing cross-validation error. From this result, it seems as if the network is unable to generalize at all, and that its learning effort is being applied exclusively to the memorization of the specific input/output correspondences in the training data. This result is disturbing.

There is an unresolved concern pertaining to the mapping of input data to the input layer of the network. The network squashing function $\phi(v)$ in its unscaled form produces output in the range $[-1, 1]$. An interesting question is raised if power spectrum data is biased and re-scaled into this range. Power values are scaler, not vector, quantities, yet biasing an re-scaling the power value could cause the presence of a signal to both excite and inhibit attached nodes, depending on whether the power value is above or below its midway point. A quick experiment replacing the $\phi(v)$ and $\phi'(v)$ with their logistic function

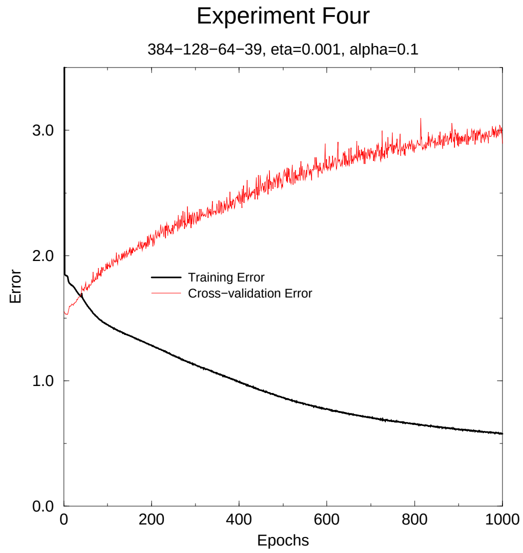

Fig. 5. Graph of training and cross-validation error for experiment 4. Cross-validation error continually increases from the onset of training.

counterparts did not yield any definitive difference, but this should be considered further.

In all, the two principal goals of this project were realized: a better understanding of the terminology and difficulties surrounding speech recognition research was gained, but there is much more to learn. In addition, this author is much has a much more concrete grasp of the back-propagation algorithm and the implementation of neural nets.

6. FUTURE WORK

It may be that the most common result of any research project is the recognition that substantially more work can be done to further our understanding. Perhaps, because of its superficial nature and naïve assumptions, this paper will yield more such fruit than it should. However, before this work can be extended usefully, a thorough review of current research, both in speech recognition and neural networks, should be undertaken. Many questions raised here may have been answered already. What follows is merely a shopping list of ideas generated by this work.

(1) Find the cause of the non-decreasing cross-validation error. For there to be no reduction in this value implies that there may be no identifiable patterns in the feature space. Since this is unlikely, there may be some unseen bug in the network software.

(2) Partially to address the problem #1, the input data should be analyzed carefully. At the very least, it should be possible to discover the variability amongst frames assigned the same tag value, both speaker-to-speaker variability, and same-speaker contextual variability for different instances of the same phoneme. Perhaps speech frames could be clustered by an ART network for comparison to the output tags.

(3) Many more speech samples should be gathered from different speakers. Perhaps the insufficient data led to the effect of problem #1.

(4) Fourier transforms produce linear divisions of the frequency domain. However, humans do not perceive linear changes in frequency as linear changes in pitch. Perhaps the input feature space could be reduced by applying a different transform method, yielding perhaps 3 poles per octave. High upper frequency resolution is almost certainly wasted.

(5) Different network architectures may be tried, but higher numbers of nodes will probably not be effective until the generalization problem is identified. Evaluation of the mapping function of power spectra to input nodes should be done: perhaps the $[0, 1] \rightarrow [-1, 1]$ mapping is counter-productive.

(6) Software could be written to analyze the state of the network. Perhaps output signals could be traced back to the input by following the highest-weighted connections. For feed-forward networks, this should not be too difficult to understand.

(7) Phoneme recognition is only part of the speech regcognition problem. The next phase of research should investigate how the output of a phoneme-recognizer could be incorporated into a word-recognizer.

7. ACKNOWLEDGMENTS

I am grateful for the many information, database, and software resources that have been made electronically available via the World Wide Web and FTP. In particular, I wish to acknowledge Carnegie Mellon University for providing the CMU pronouncing dictionary (`cmudict.0.4`) maintained by Bob Weide (`weide@cs.cmu.edu`), which provides American phonetic translations of over 100,000 words. I also wish to acknowledge the author of the `rsynth` program, Nick Ing-Simmons (`nik@tiuk.ti.com`), for providing his integrated version of the Klatt synthesizer, which I used in conjunction with `cmudict.0.4` for deriving phonemic translations. Finally, the most common words list was calculated by Jim Gillogly, and made available at `http://www.cse.ogi.edu/∼brianh/most_common_words.html`.

I owe thanks to my boss, Dr. Peter S. Ray, for his patience during my pre-occupation with this project and for allowing me the use of his computing resources. I also wish to express gratitude to those who participated

by providing speech samples for this research (and who also endured my ceaseless babbling and unbridled enthusiasm): Paul Beiringer, Debi Chandler, Caroline Emmons, Darci Kelly, Curtis Knox, Tina Cartwright, and Paige Wagner.

8. REFERENCES

Advanced Research Projects Agency, *Proceedings of the Spoken Language Systems Technology Workshop, January 22–25, 1995*, Morgan Kaufmann Publishers, San Francisco, 1995. *(selected articles)*, 305 pp.

Ainsworth, William A., *Speech Recognition by Machine*, Peter Peregrinus Ltd., London, 1988. 206 pp.

Haykin, Simon, *Neural Networks, A Comprehensive Foundation*, Macmillan Publishing Company, Englewood Cliffs, 1994. 696 pp.

Hunt, Andrew, ed., *Comp.Speech FAQ and WWW Site*, `http://www.speech.cs.cmu.edu/comp.speech`, 1993-1996.

Oppenheim, Alan V., and Ronald W. Schafer, *Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, 1975. 585 pp.

Papamichalis, Panos E., *Practical Approaches to Speech Coding*, Prentice-Hall, Inc., Englewood Cliffs, 1987. 322 pp.

Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, 1986. pp. 381–429.

Waibel, Alex, and Kai-Fu Lee, eds., *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo, 1990. *(selected articles)*, 629 pp.

Yannakoudakis, E. J., and P. J. Hutton, *Speech Synthesis and Recognition Systems*, Ellis Horwood Ltd., Chichester, 1987. 184 pp.