CONTINUOUS SPEECH RECOGNITION WITH THE CONNECTIONIST VITERBI TRAINING PROCEDURE: A SUMMARY OF RECENT WORK

Michael Franzini
Telefónica Investigación y Desarrollo; Emilio Vargas, 6; 28043 Madrid SPAIN

Alex Waibel
School of Computer Science; Carnegie Mellon University; Pittsburgh, PA 15213 USA

Kai-Fu Lee
Apple Computer Corporation; Cupertino, CA 95014 USA

18-22 November 1991

ABSTRACT

Hybrid methods which combine hidden Markov models (HMMs) and connectionist techniques take advantage of what are believed to be the strong points of each of the two approaches: the powerful discrimination-based learning of connectionist networks and the time-alignment capability of HMMs. Connectionist Viterbi Training (CVT) is a simple variation of Viterbi training which uses a back-propagation network to represent the output distributions associated with the transitions in the HMM. The work reported here represents the culmination of three years of investigation of various means by which HMMs and neural networks (NNs) can be combined for continuous speech recognition. This paper describes the CVT procedure, discusses the factors most important to its design and reports its recognition performance. Several changes made to the system over the past year are reported here, including: (1) the change from recurrent to non-recurrent NNs, (2) the change from SPHINX-style phone-based HMMs to word-based HMMS, (3) the addition of a corrective training procedure, and (3) the addition of an alternate model for every word. The CVT system, incorporating these changes, achieves 99.1% word accuracy and 98.0% string accuracy on the TI/NBS Connected Digits task ("TI Digits").

1. Introduction – The Hybrid Approach

Recent work in continuous speech recognition has focused on augmenting existing hidden Markov model (HMM) based techniques with other methods. One direction this research has taken is towards the use of powerful *discrimination* methods instead of the Maximum Likelihood Estimation (MLE) procedures typically used for training HMMs. Since speech recognition entails *discriminating* among speech units, learning procedures which are defined explicitly in terms of performing a discrimination task may be better suited to the task than MLE.

Another focus of recent work with HMM-based speech recognizers has been on modeling speech parameters directly, rather than using the drastically reduced representations of the speech signal produced by vector quantization (VQ). Systems which vector quantize have a distinct disadvantage, being deprived of information which may be of use in the recognition process. One approach to this problem has been to use continuous density HMMs. However, these systems incorporate assumptions about the distributions of speech parameters which may be inaccurate. (See [1].)

Connectionist learning procedures are designed to perform accurate *discrimination*, and they operate directly on real-valued parameters, without making any strong assumptions about the distributions of these parameters. Since the energy functions typically used in connectionist learning maximize the system's ability to discriminate among classes of input patterns, these procedures are well suited to speech recognition applications, in which the usual goal is to discriminate among words or phones. Most connectionist models include inputs defined over a continuous range of real numbers and exhibit no advantage with discrete inputs. Integrating these models into

CH 3065-0/91/0000-1855 $1.00 © IEEE

HMMs can relieve the need for VQ, while adding discrimination-based learning. Hence, such hybrid methods have been the subject of a great deal of recent investigation (e.g., [2, 3]).

In building hybrid connectionist/HMM systems, speech recognition is viewed as a *static pattern classification* problem combined with a *time alignment* problem. These systems take advantage of the ability of connectionist networks to discriminate accurately among classes in static pattern classification problems. They use HMM technology to find the optimal time alignment based upon the output of the connectionist component of the system.

In this paper, we describe the Connectionist Viterbi Training (CVT) procedure, which is one such hybrid system. We present a general overview of the system, describe its components, and report a series of recent experiments in which we improved the performance of the system by more than 50% on the TI Digits task.

2. System Overview

The CVT system consists of a neural network (NN) and a hidden Markov model (HMM). These two components are not independent; the training of each depends on the other.

A fundamental idea underlying the architecture of the CVT system is that the connectionist section of the system performs a speech *classification* task and the HMM part of the system performs a *time alignment* task.

In the earliest version of this system, the NN looked at a wide window of speech and produced as its output a hypothesis about the identity of the word in its input window. These hypotheses were generated for input windows in every position on the input data. Then, a viterbi search was used to find the optimal path through these hypotheses. In this version of the system, the NN and HMM components were entirely independent; the outputs of the NN were simply passed for processing to the HMM.

In the most recent version of the CVT system, the two components of training are integrated. The outputs of the neural network no longer correspond to *linguistic* entities (as they did in a previous version of the system, which had output units corresponding to words and phones); they now are defined in terms of the HMM architecture. Each NN output unit maps to one transition in the HMM.

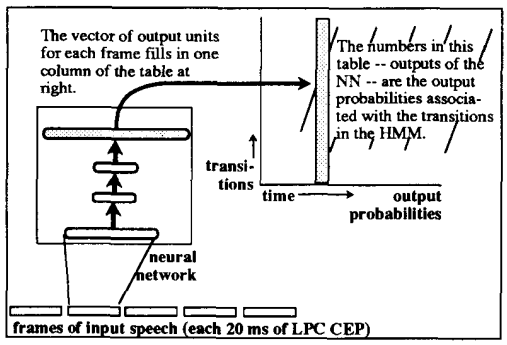

Figure 1a: NN Generation of Output Probabilities

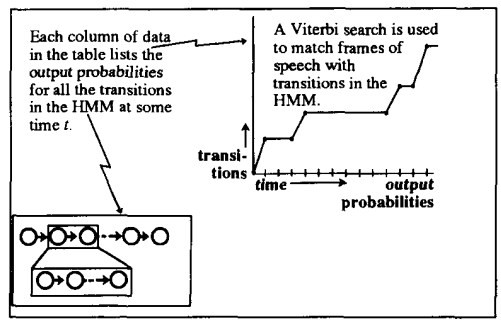

Figure 1b: HMM alignment of speech

Figures 1a and 1b illustrate the the primary components of the system. In the first phase of processing, the system passes one frame of speech (along with several frames of context) to the NN, which outputs a vector of floating point numbers; this vector will serve as the output probabilities for the HMM. In the second phase of processing, once one vector of output probabilities has been generated for every frame of input speech, a Viterbi alignment is performed to determine the most likely path through the HMM. During training, this is a "forced alignment" (i.e. forced to pass through the correct word sequence), and the results of the alignment are used for re-training the NN. During recognition, the Viterbi alignment is free to pass through all words, and the sentence recognized is determined by observing the words entered.

3. The TI Digits Task

The Texas Instruments Connected Digits Recognition Task (commonly known as "TI Digits") has become one of the standard tasks on which recognition performance of systems is assessed. The database consists of studio-quality dialectically-balanced recordings of about 10,000 utterances of digit strings ranging in length from one

to seven. The vocabulary includes the words "one" through "nine," "oh" and "zero."

The data, as provided by the NBS, was sampled at 20 KHZ. Before use for training or testing our system, the speech was downsampled to 16 KHz and pre-emphasized with a filter of $1 - 0.97z^{-1}$. Then, a Hamming window with a width of 20 ms was applied every 10 ms. Autocorrelation analysis with order 14 was followed by LPC analysis with order 14. Finally, 12 LPC-derived cepstral coefficients and one power value were computed for each frame.

4. Connectionist Architecture

4.1. Current CVT NN Architecture

The current version of the CVT system uses a four-layer[1] network which accepts as input 91 speech coefficients and produces as output 120 floating point numbers between zero and one. The input consists of one 20 ms. frame of speech with three frames of context on each side (10 ms overlap between adjacent frames), and the output includes one value for every transition in the HMM. The two hidden layers each contain 34 units. Hence, the total number of connections in the network is 8,330. The network is illustrated in Figure 2.

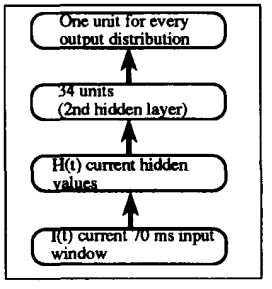

Figure 2:
The Network Used in the Current Version of the System

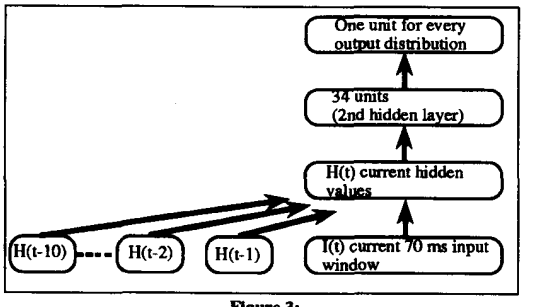

Figure 3:
The Recurrent Network Used in a Previous Version of the System

The considerations that were made in designing the network included:

1. *Choosing the optimal number of layers* – We found that there was a significant performance benefit for using four layers instead of three. The classification error rate was reduced by nearly an order of magnitude when the number of layers in the network was increased from three to four.[2] The addition of a fourth layer resulted in a significantly longer training period, due to the slower convergence which is typically observed is deeper networks.[3] Adding a fifth layer did not produce any performance benefit and increased the training time by an order of magnitude over that of the four-layer network.

2. *Choosing the optimal number of units per hidden layer* – We found that the recognition performance of the network did not depend on the number of units in each hidden layer. However, when there were very few units in each hidden layer (fewer than ten), the convergence of the network was so slow that we were unable to complete training. [4]

3. *Choosing between recurrence and non-recurrence (and the structure of the recurrent mechanism when present)* – We have performed extensive investigations of the benefits of various types of recurrent networks,

[1]The convention used here is that the term "layer" refers to a layer of *units* in the network; hence, a *four-layer* network has three layers of *weights*.

[2]This experiment was performed with a network trained to classify 500 ms blocks according to the digit to which they belonged. This network will not be described here.

[3]This is an empirical observation made by the authors and applies to networks trained for speech recognition. The same observation has been made by others not working on speech recognition – e.g., Hinton (personal communication).

[4]A past experiment [5] showed that NN performance on a speech classification task degraded slightly when the number of hidden units was reduced drastically (to about 2 or 3) and reached a plateau quickly (when the number of hidden units reached about 8).

which we will mention only briefly here. We found that the best configuration for a recurrent network was that shown in Figure 3. This architecture is similar to that described by Elman [4]; however, in our network, there are ten groups of "history" or "representation" units, where Elman has only one. In our experiments, the network was unable to retain information across more than two time steps when only one set of history units was used. The system performed 6% better with the recurrent version of the network than with the non-recurrent version, but, as discussed below, we felt that the computational cost was too high to justify this benefit.

4. *Determining the topology of connections between layers* – Although we have not examined different patterns of connection in the context of the complete CVT system, we did investigate the impact of using sparse connections between layers on training data classification performance. We found that accuracy was degraded by 10 - 50%, as the density of connections between layers varied between the maximum ("fully connected") and a pattern of local connections in which each unit had a fan-in of ten.[5]

5. HMM Architecture

5.1. Current CVT HMM Architecture

In the current version of the system, the HMM architecture is similar to that described by Bakis [5]. The system uses word models in which each transition corresponds on average to two frames of speech in a word. This is close to the optimal HMM topology reported by Picone [6] for this task; he found that the best configuration uses word models with one transition for every frame of speech in a prototypical utterance of the word modeled. We used half this number of transitions in order to reduce the computational cost of training and recognition.

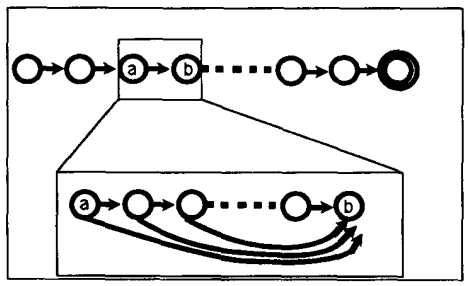

Figure 4a: Our Word Model

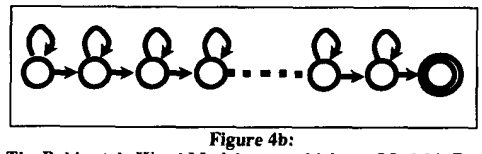

Figure 4b:
The Bakis-style Word Model upon which our Model is Based

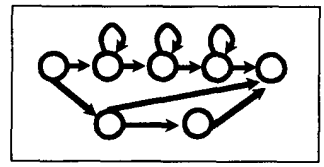

Figure 5:
The Phone Model Used in a Previous Version of the System

Figure 4a illustrates the word models used in the current version of the CVT system, and Figure 4b shows the simpler Bakis-style models upon which our models are based. In our version of the word models, the duration controls are significantly tighter, since self-loops are not permitted.

As shown in Fig. 4a, every adjacent pair of states (e.g., those labeled *a* and *b* in the figure) actually corresponds to a series of states (illustrated in the lower portion of Fig. 4a). This series of states serves as a replacement for the self-loop that would have appeared on state *a* in the architecture shown in Fig. 4b. All of the transitions in the series are tied; i.e., they all share the same output probability (and therefore all correspond to the same output unit in the network). Furthermore, each of the transitions to state *b* has a probability associated with it, which allows duration modeling at the state level. (Hence, the original Bakis architecture shown in Fig. 4b would be equivalent to the design in Fig. 4a if (i) a ceiling were placed on the number of times a self loop could be taken, and (ii) a probability $p(a, n)$ of taking a self loop *a n* times were calculated.)

[5]The term "fan-in" refers to the number of incoming connections to a unit.

5.2. Phone Models *vs* Word Models

In a previous version of the system, we used phone-based HMMs with exactly the same topology as those used in the SPHINX system [7]. (See Figure 5.) When we switched from these to the word-based HMMs described above, the system performance improved by 40%.

The disadvantage of the phone models is that each transition in the HMM has to model a variety of speech frames, which are not highly localized within words. Specifically, we believe that the disadvantages of the phone models are owed to (1) parallel transitions, (2) self-loops, and (3) too few transitions per word; the best HMM architecture is that which models at the lowest level, with the most rigid correspondence between transitions and speech.

6. Recent Improvements to the CVT Procedure

6.1. Corrective Training

Using a form of corrective training, we have further reduced the error rate in the new non-recurrent word-based CVT system by about 6%. The general idea of the corrective training procedure is that emphasis in training should be placed on sentences in which the system is likely to commit recognition errors. This emphasis is achieved simply by performing extra training on misrecognized sentences. However, given the rate of recognition errors, there is not a large corpus of misrecognized sentences to use in this manner.

In order to generate more misrecognitions for the corrective training procedure, we suppress correct recognition of a random subset of training sentences; we prevent the Viterbi forced alignment from entering the correct word at certain randomly selected times. Not only does this augment the size of the corrective-training corpus, it also produces sentences which are likely to include realistic recognition errors – since the system is in effect making a "second choice" recognition, which we assume often corresponds to the sorts of errors made in actual recognition.

6.2. Multiple Models

A second training strategy which has proved beneficial – yielding a 33% increase in performance – uses multiple models for each word. Once the single-model-per-word system was fully trained, an extra output unit for every transition in every word was added to the neural network. These weights were set equal to the corresponding previously trained weights, with the addition of a small (5%) random perturbation. Then, an additional HMM was created for every word, and these new models were associated with the new network output units.

CVT training proceeded as before; however, during the forced alignment phase, the system was permitted to enter *either* of the models for a word, based on the network scores. Hence, the system was able to develop models specialized for two primary pronunciations of each word. For example, we observed that the word 'eight' (phonetically represented as */ay/ /t/*) has two primary pronunciations: one in which the final stop is strongly pronounced and another in which it is hardly detectable. Using the new system configuration, the two pronunciations could be modeled separately.

7. Results & Conclusion

Table 1 is a summary of the performance of the CVT system on the TI Digits task, showing the changes in performance which accompanied the recent changes to the system. The current version of the system achieves 99.1% word accuracy and 98.0% string accuracy on the TI Digits.

The goal of this work was to build a continuous-speech recognition system which combined the pattern-classification ability of connectionist networks with the time-alignment ability of hidden Markov models. We began with a system built of two distinct components: a NN frame classifier, and an HMM post-processor. Then, using the Connectionist Viterbi Training procedure, we integrated the training of the two parts of the system, such that the classification task being performed by the NN was in effect controlled by the HMM.

	Word Accuracy	String Accuracy	Incremental Improvement
Baseline CVT System 1990	98.5	95.0	
− recurrence	98.0	94.7	−6%
+ word models	98.7	96.8	+40%
+ corrective training	98.8	97.0	+6%
+ multiple models	**99.1**	**98.0**	+33%

Table 1: Recent Improvements in Results on TI Digits

In making the most recent revisions to the system, we have reached several conclusions, which may be extensible to other approaches and other tasks as well: (1) that modeling speech at the lowest level possible appears to produce the best results and, when permitted by the task (i.e., when the vocabulary is sufficiently small and the size of the training corpus sufficiently large), word-based HMMs should be used in place of phone-based HMMs, (2) that the performance of non-recurrent NNs is only slightly worse than recurrent NNs, and the former allows a significant computational saving, (3) that a corrective training procedure can reduce the error rate by providing additional training on error-prone data, and (4) that using multiple models per word can result in a higher overall recognition rate, by allowing distinct representations of different pronunciations or different speaker characteristics.

The most general conclusion to be drawn from this work is that NN-HMM hybrid systems show great promise in the domain of continuous speech recognition. These systems, which have been under investigation for only about three years, have already achieved error rates within one order of magnitude of the best results on a task for which HMM-based recognizers have been under development for nearly a decade. This early success suggests that these hybrid systems may be one of the most viable means for performing high-accuracy continuous speech recognition.

8. Acknowledgements

The authors wish to thank Juan Siles at Telefónica I+D and Raj Reddy at Carnegie Mellon for their continued encouragement and support, Patrick Haffner and Michael Witbrock for their invaluable advice and assistance throughout this project, and Daniel Tapias for facilitating the first author's survival in Spain long enough to write this paper.

References

[1] Brown, P., *The Acoustic Modeling Problem in Automatic Speech Recognition*, Ph.D. Thesis, Carnegie Mellon University, May, 1987.

[2] Huang, W., Lippmann, R. "HMM Speech Recognition with Neural Net Discrimination," *Proc. Neural Information Processing Systems (NIPS) Conference*, November, 1989.

[3] Bourlard, H. and Morgan, N. *Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition*, Tech. Report TR-89-033, July, 1989, International Computer Science Institute, Berkeley, CA.

[4] Elman, J.L. *Finding Structure in Time*, Tech. report, Center for Research in Language, University of California, San Diego, April, 1988.

[5] Bakis, R. "Continuous Speech Recognition via Centisecond Acoustic States," *Proc. 91st Meeting Acoustical Soc. of America*. April, 1976.

[6] Picone, J. "On Modeling Duration in Context in Speech Recognition," *Proc. ICASSP*, April, 1989.

[7] Lee, K.F. *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*, Ph.D. Thesis, Carnegie Mellon University. 1988.