CONNECTIONIST VITERBI TRAINING: A NEW HYBRID METHOD FOR CONTINUOUS SPEECH RECOGNITION

S8.4

Michael Franzini, Kai-Fu Lee, and Alex Waibel

School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213

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 CVT procedure is an extension of the procedure we described at ICASSP'89; however, *CVT integrates the connectionist and HMM components* of the system more tightly than the ICASSP'89 approach. Unlike the previous procedure, CVT can be run iteratively and can be applied to large-vocabulary recognition tasks. Successful completion of training the connectionist component of the system, despite the large network size and volume of training data, depended largely on several measures taken to reduce learning time. The system was trained and tested on the TI/NBS Speaker-Independent Continuous Digits database. Performance on test data for unknown-length strings was 98.5% word accuracy and 95.0% string accuracy. Several improvements to the current system are expected to increase these accuracies significantly.

1 Introduction

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 modelling 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 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., [2,3,4]).

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.

The work described here is an extension of that reported at ICASSP'89 [2], in which a connectionist network was used to generate hypotheses concerning the phones and words present in the input, and these hypotheses were processed by an HMM. In our present work, the two components of the system are more tightly integrated. Using a simple extension of the Viterbi (or "Segmental K-Means") training procedure [5], a connectionist network was used to encode the output probabilities associated with the transitions of an HMM.

2 The Speech Database

As in our work reported at ICASSP'89, the adult portion of the TI/NBS Connected Digit Database [6] was used for assessing the effectiveness of the training and recognition procedures. The vocabulary consisted of the digits *one* through *nine*, *oh* and *zero*. The database includes approximately 6000 sentences recorded in a quiet environment and is dialectically balanced.

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.

3 Training the System

Phone-based HMMs similar to those used in the SPHINX system [7] were used. The sentence model was composed of a concatenated series of word models, which in turn was composed of a concatenated series of word-dependent phone models. The architecture of a typical phone model is illustrated in Figure 1. The system was initialized by training a discrete HMM with this architecture using three iterations of the Baum-Welch procedure. This HMM was used to produce the initial alignment for the Connectionist Viterbi Training (CVT).

The CVT procedure was used to re-estimate the output distributions, which were associated with transitions in the HMM. The output probabilities, calculated in typical discrete HMMs by VQ followed by a table lookup, were calculated here by presenting frames of speech to a connectionist network. The outputs of the network were used as HMM output probabilities.

This research was supported by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 5167, under contract number N00039-85-C-0163, by the Office of Naval Research under contracts N00014-86-K-0678 and N00014-86-G-0146, and by the National Science Foundation under contract EET-8716324. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the National Science Foundation, the Office of Naval Research or the US government.

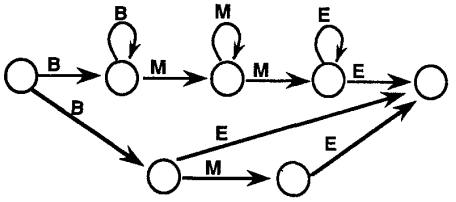

Figure 1: A Typical Phone Model; All transitions with the same label (B/M/E) are tied.

The training procedure proceeded as follows:

1. *Initialization.*
   - Use SPHINX to train a set of discrete HMMs on this task.
   - Perform a forced Viterbi alignment of all utterances in the training set using this set of HMMs. This alignment establishes a mapping from frames of speech in the input to transitions in the HMMs.
2. *First iteration.*
   - Train a connectionist network on the pairings (from frames of speech to HMM transitions) produced by the initial alignment.
   - Use the initial HMMs, including the SPHINX-trained transition probabilities, but replace the discrete-HMM output distributions with the distributions encoded by the network. That is, discard the VQ codebooks and lookup tables, and use the network to generate output probabilties from this point on.
3. *Subsequent iterations.*
   - Perform another forced Viterbi alignment of all the training data using the new HMMs (which now include the connectionist network).
   - Re-train the network on the new pairings from this alignment.
   - Re-estimate the transition probabilities. The probability of taking a transition from state i to state j is re-estimated as the ratio of the number of times transition ij was taken (in the state sequences generated by the Viterbi alignment) to the total number of times that transitions were taken from state i to any state.
   - Check the performance of the new model on a "validation set" of utterances. If improvement is observed, perform another iteration, beginning with a new forced Viterbi alignment.

Figure 2 shows the recurrent connectionist network used in the CVT system. The network takes 70 ms of speech as input – 1 central frame plus 3 frames of left context and 3 frames of right context. The network produces one output value for each transition in the HMM. (However, sets of tied transitions have only one output unit each.) Training was performed using the back-propagation learning procedure [8].

During training, for each frame of speech presented to the network, the desired value for the output unit corresponding to the transition to which the frame of speech was assigned in the forced alignment is set to 1.0. Desired values for all other output units are set to 0.0.[1]

The 70 ms input window was shifted accross the utterance from left to right in 50 ms steps. The network's recurrent mechanism (illustrated with dashed lines in Figure 2) retained a history of the internal state of

[1]Somewhat faster learning was observed when a desired value of 0.4 was assigned to output units which corresponded to different transitions within the correct phone. That is, the desired value for the unit corresponding to one transition in a phone model was set to 1.0, the desired value for units corresponding to other transitions in that phone model was set to 0.4, and all other desired values were set to 0.0.

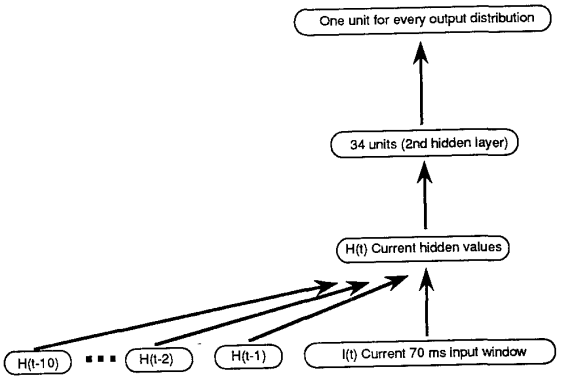

Figure 2: The Recurrent CVT Network

the network over 10 such steps, or a total of 500 ms. The design of this network is an extension of the design used by Elman ([9]); we found that a single set of recurrent context units, as used by Elman, was not able to retain context over a sufficiently long duration for this application. Hence, we augmented Elman's network with nine other groups of recurrent units, as shown in Figure 2.

The following were the steps in the training of the network:

1. Initialize the outputs of all *history units* (see Figure 2) to 0.
2. Place the first 70 ms of processed speech on the inputs, and forward propagate.
3. Set up the desired values for the output units based on the central frame in the input layer,as described above. Back propagate.
4. Copy the output values from units in the first hidden layer to the units in the first history group.
5. Forward propagate.
6. Set up the desired values, as described above. Back propagate to all input units, including the history units.
7. Shift all the history groups to the left. Discard the values from the 10th group, shifting all others one group to the left.
8. Go to step 4; continue until the whole utterance has been processed.

Once the CVT network was trained, its output values were distributed in the range between 0 and 1 (see *Results* section below).

3.1 Notes on Connectionist Training

The volume of training data and the large number of connections in the network presented a formidable implementation problem – that of how to perform the training within an acceptable time frame. The slowness of connectionist training algorithms and the poor scaling behavior they exhibit as the size of the task domain grows have interfered with the successful application of these algorithms to real-world speech recognition problems.

The success of our current approach is due in large part to several measures taken to increase the speed of the back-propagation learning procedure. Based on our experience with connectionist learning in other task domains, we estimate that these simple measures have contributed several orders of magnitude of reduction in learning time. The complete training procedure entailed on the order of 10^{12} floating point computations, which was easily manageable using a Convex C-1 computer.

The measures which we found to be most effective for reducing learning time were the following:

- *Split training corpus & Pooled updates.* We have found that learning progresses in a more stable and uniform manner when weight updates are performed after forward and backward passes through all tokens in the training set, rather than after each back propagation. However, training on the entire 6000-utterance database in this manner was prohibitively time consuming, so several smaller training sets were constructed by randomly sampling utterances from the database. Extensive training was performed using each training set (which included about 1000 sentences) with pooled updates. Although some unlearning takes place when moving from one training set to the next, most of the error reduction achieved for each training set generalized to the others.

- *Dynamic adjustment of learning rate.* The single most effective heuristic used to decrease learning time was an adjustment procedure which maintained the maximum value of the learning rate parameter for which learning would remain stable (first described in [10]). The heuristic monitored the angle, θ, between the error derivative vector at epoch t and that at epoch $t-1$:

$$\cos\theta = \frac{\sum_{i,j}(d_{ij}(t-1)d_{ij}(t))}{\sqrt{(\sum_{i,j}d_{ij}(t-1)^2)(\sum_{i,j}d_{ij}(t)^2)}}, \text{ where } d_{ij} = \frac{\partial E}{\partial w_{ij}} \quad (1)$$

where E was the network error measure, and w_{ij} was the weight on the connection from unit i to unit j. The learning rate, ε, was then updated according to the "epsilon scaling" rule:

$$\varepsilon(t) = \varepsilon(t-1) \times \beta\frac{\cos\theta + 1}{2} \quad (2)$$

where β, the "epsilon-scaling factor," typically 1.005, determined the rate of increase of ε when $\cos\theta$ was near 1.0. Epsilon scaling was effective only when (1) pooled updates were used, (2) the number of patterns in the pool was large, and (3) the composition of the pool did not vary from one epoch to the next.

- *Epsilon splitting.* Our experience confirmed previous findings [11] that dividing the learning rate on connections into a unit by the unit's fan-in[2] leads to more stable behavior during learning.

- *High momentum.* After an initial sharp descent down the network error surface, during which the momentum was set to 0 (for about 100 epochs), the speed of learning increased considerably as the value of the momentum parameter was increased, as long as it remained below 1.0.

- *Avoidance of overlearning.* When the error for an output unit for a given pattern dropped below a threshold of 0.01, the error value for that unit and pattern was taken to be 0. This effectively avoided detrimental effects of "overlearning."

4 Results

	Word Accuracy	String Accuracy
Unknown Length	98.5%	95.0%
Known Length	99.1%	96.1%

Table 1:
Summary of Results

[2]A unit's "fan-in" is the number of connections into that unit.

The overall results of the CVT procedure on the TI Digits task are summarized in Table 1.[3] These results are based on recognition using a full Viterbi search [7].

These results are from experiments using word models constructed from phone models of the form illustrated in Figure 1. Experiments in which information about the distributions of phone and word durations was encoded in the HMM transition probabilties exhibited no benefit over the experiments reported here, in which no duration modelling was used.

Likewise, experiments in which the outputs of the network were divided by the prior probabilities associated with each of the network's output classes, as suggested by Bourlard [4], showed no improvement over the conditions of the reported experiments, in which the network outputs were used directly.

The connectionist component of the system achieved a correct first-choice phone classification rate of 92% for training data. This value was not computed for test data, but, as indicated by overall system performace, recognition rates for training data and test data were quite close.

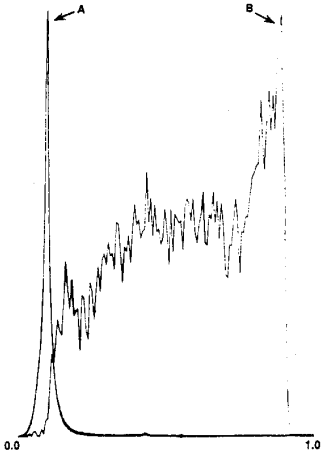

Figure 3: Histograms Showing Separation Between Two Classes

In order to assess the accuracy of the separation between unit outputs corresponding to transitions to which the input should not be mapped and those corresponding to transitions to which the input should be mapped, we produced histograms of the output values in these two classes. These two histograms are shown, superimposed, in Figure 3.

5 Discussion

The goal of this work was to take advantage of the superior discrimination ability of connectionist networks in an HMM-based system. The CVT procedure achieved this goal by isolating the speech pattern discrimination problem from the temporal alignment problem and applying connectionist methods to the former and HMM methods to the latter.

The primary advantages of CVT over the pure HMM approach are the following:

- It uses connectionist techniques for low-level pattern classification. There is some evidence (e.g., [12]) that these techniques are superior to HMMs at static pattern classification.

- It does not require the use of vector quantization (as discrete HMMs do) and the loss of precision which accompanies it.

[3]One set of models and one network was trained for male speakers and one for female speakers. The models and network were chosen for each test utterance according to the gender of the speaker.

- It is able to represent the output distributions without making the strong assumptions concerning the distributions of speech parameters made in continuous-density HMM systems.

The primary advantages of CVT over the purely connectionist approach are the following:

- It uses HMMs to perform the temporal modelling and sequencing for which no highly effective connectionist approach has been found.
- It provides a means for performing large-vocabulary speech recognition; no purely connectionist method has been applied to this task.

The primary advantages of CVT over the method we described at ICASSP'89 are the following:

- It integrates the connectionist and HMM components of the system for both training and recognition. Previously, these two parts of the system were disjoint.
- It uses a training procedure which can be run iteratively to improve performance.
- It is extensible to large vocabulary recognition, since the phone is the largest unit of speech used in the connectionist network; the size of the network does not increase with the vocabulary size.[4]

6 Future Work

Our short-term goal is to match or surpass the best performance achieved on this task – 99.5% word accuracy [13] – i.e., a $\frac{2}{3}$ decrease in word errors. In order to do this, we are focusing on two aspects of our system:

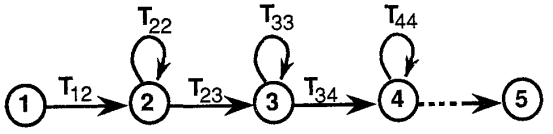

Figure 4: An Alternative to Our Current Phone Model

1. *HMM architecture.* When sufficient training data is available, the best results are achieved when each frame of a typical utterance is modelled individually by a transition in the HMM [14]. In our current system, each transition in the HMM models several frames of speech in a typical utterance. Our initial choice of model topology was based chiefly on the availability of the model specifications for the SPHINX system. Since ample training data is available, a simple progressive HMM architecture (as shown in Figure 4) is likely to produce better results than the current SPHINX-style HMMs. By increasing the resolution of the modelling in this manner, we will increase the system's capacity to represent useful information about the task domain.
2. *Speech representation.* Although LPC Cepstrum Coefficients are widely used in HMM-based recognition systems, there are conflicting results concerning the effectiveness of this representation, particularly as input to a connectionist system. Mel-scale FFTs may represent the data in a form better suited to connectionist processing.

At the time of this writing, experiments using the linear HMM architecture and mel-scale FFT representations are underway. In these experiments, we are continuing to use the TI Digits database. In subsequent experiments, we will investigate the applicability of CVT to large-vocabulary recognition.

[4]However, the current phone models are word-dependent, and, for a large-vocabulary task, it would probably be necessary to use context-dependent phone models. (See [7]). Depending on the task, the number of these models might be prohibitively large to use in a CVT system.

7 Acknowledgements

The authors would like to thank Michael Witbrock for his helpful advice, Mei-Yuh Hwang and Hsiao-Wuen Hon for providing the labeled training data, Donn Hoffman for proofreading this paper, and Raj Reddy for his encouragement and support.

References

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

[2] Franzini, M.A., Witbrock, M.J., Lee, K.F., "A Connectionist Approach to Continuous Speech Recognition," *Proc. ICASSP*, April, 1989.

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

[4] 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.

[5] Rabiner, L.R., Wilpon, J.G., and Juang, B.H., "A Segmental K-Means Training Procedure for Connected Word Recognition," *AT&T Technical Journal*, May, 1986.

[6] Leonard, R.G. "A Database for Speaker-Independent Digit Recognition", *Proc. ICASSP*, March, 1984.

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

[8] Rumelhart, D.E., Hinton, H.E., and Williams, R.J. "Learning Internal Representations by Error Propagation" in Rumelhart, D.E., McClelland, J.L., eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol I. 1986. The MIT Press. Cambridge.

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

[10] Franzini, M.A. "Speech Recognition with Back Propagation," *Proc. Ninth Conf. of the IEEE Engineering in Medicine and Biology Society*, November, 1987.

[11] Plaut, D.C., Nowlan, S.J., and Hinton, G.E. *Experiments on learning by back propagation*, Tech. report, Carnegie Mellon University, June, 1986.

[12] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K., "Phoneme Recognition: Neural Networks vs. Hidden Markov Models", *Proc. ICASSP*, April, 1988.

[13] Doddington, G.R., "Phonetically Sensitive Discriminants for Improved Speech Recognition," *Proc. ICASSP*, April, 1989.

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