S2.17

INTEGRATING TIME ALIGNMENT AND NEURAL NETWORKS FOR HIGH PERFORMANCE CONTINUOUS SPEECH RECOGNITION

Patrick Haffner*, Michael Franzini, and Alex Waibel

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

ABSTRACT

Successful application of existing connectionist methods to continuous speech recognition requires the use of time-alignment procedures. These procedures, usually based on dynamic programming, provide means for supervising the training of neural networks. This paper describes two systems in which neural network classifiers are merged with dynamic programming (DP) time alignment methods to produce high performance continuous speech recognizers. One system uses the Connectionist Viterbi Training (CVT) procedure, in which a neural network with frame-level outputs is trained using guidance from a time alignment procedure. The other system uses Multi-State Time Delay Neural Networks (MS-TDNNs), in which embedded DP time alignment allows network training with only word-level external supervision. CVT has been described previously [1]; only changes to the system and new results on the TI Digits task are reported here. The newest CVT results on the TI Digits are 99.1% word accuracy and 98.0% string accuracy. MS-TDNNs, introduced in this paper, are described in more detail here, with attention focused on their basic architecture, the training procedure, and results of applying MS-TDNNs to continuous speaker-dependent alphabet recognition: on two speakers, word accuracy is respectively 97.5% and 89.7%.

1 INTRODUCTION

Extending the classification capabilities of connectionist networks to continuous speech recognition is an important research direction in speech recognition. Time alignment presents the greatest problem for neural network (NN) based systems, since connectionist learning procedures are typically defined in terms of static pattern classification tasks. One effective solution to the problem of applying NN-based systems to continuous speech recognition is to combine an alignment procedure with the NN; the time alignment can be performed *externally* to the network [1, 2] or internally [3]. In most of these "hybrid" systems, the alignment procedure is based on dynamic programming (DP).

This paper describes two hybrid systems developed at Carnegie Mellon, both of which demonstrate that NN-based systems are able to achieve very high performance on continuous speech recognition tasks.

The Connectionist Viterbi Training (CVT) procedure uses NNs to perform frame-level classification and a hidden Markov model (HMM) to perform time alignment. The outputs of the network are used as emission probabilities associated with the transitions in the HMM. The training procedure is a straightforward adaptation of Viterbi Training, in which re-estimation of the HMM emission probabilities entails re-training the NN. An earlier version of the CVT system was described at ICASSP'90 [1]. This paper reports changes to the system and new results on the TI Digits task.

Multi-State Time Delay Neural Networks (MS-TDNNs) perform classification at the word level. Unlike CVT (and many other hybrid methods), these networks are not trained using frame-level supervision provided by the time-alignment section of the system. MS-TDNNs incorporate the DP procedure into their training, such that only external word-level supervision is required. Because of the novelty of the approach, the way the DP procedure is implemented inside the NN is described in some detail. The architecture of the MS-TDNN system can be characterized as purely connectionist, since the time alignment procedure is an integral part of the neural network training. Powerful learning techniques and the use of phonemic knowledge in the NN architecture enable this very large NN to learn within reasonable time with minimal external supervision. The system is compared to the discrete HMM based SPHINX system on continuous speaker-dependent alphabet recognition.

2 INTEGRATING TIME ALIGNMENT AND CONNECTIONIST NETWORKS

To take into account the time distortions that may appear within its boundaries, a word is generally modeled by a sequence of states that can have variable time durations. Even though some recurrent connectionist architectures have been proposed to learn sequences of states, they have only been demonstrated on very simple tasks. Training proceeds too slowly in these systems to make them practical in large speech tasks.

For speech tasks in which modeling sequential state information is not necessary, excellent recognition performance has been achieved using TDNNs or frame level classification NNs. To extend this performance to multi-state word recognition, one has to combine the NN with a procedure performing time alignment, usually based on Dynamic Programming. A NN produces the state scores, and, for each word in the vocabulary, a DP procedure finds the path through the set of states which provides the best cumulative score (generally the sum or the product of the state scores lying on this path). The word, or the string of words (in continuous speech) yielding the best cumulative score is taken to be the result of the recognition. A variety of training methods have been proposed for those systems.

3 CONNECTIONIST VITERBI TRAINING

At ICASSP '90, Franzini, Lee and Waibel [1] reported results for the Connectionist Viterbi Training (CVT) procedure applied to the TI Digits task. Although that version of the CVT system performed considerably better than the SPHINX system on the same task,[1] it significantly underperformed the best HMM-based continuous word recognizers [4]. Since then, we have reduced the string and word error rates of the CVT system on this task by more than 50%.

CVT is one of several hybrid systems combining neural networks and Viterbi alignment which have been proposed recently. Generally, in these systems, the frame level output score of each state is assumed to represent a likelihood associated with the input speech, and the overall emission probability of a word is obtained by multiplying the frame level outputs, which approximate probabilities.

---

* – Patrick Haffner is at Centre National d'Etudes des Telecommunications, Lannion, FRANCE.

This research was supported by the Defense Advanced Research Projects Agency (DOD) and by the Office of Naval Research under contracts N00014-86-K-0678 and N00014-86-G-0146. 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 Office of Naval Research or the US government.

[1] Very little tuning of the Sphinx system was performed.

CH2977-7/91/0000-0105 $1.00 © 1991 IEEE

These systems are "hybrid" not only in their architecture but also in their training procedures. The free parameters of the HMM/time-alignment part of the system are reestimated using probabilistic techniques, whereas the NN weights are modified using gradient descent.

Using the CVT procedure, a network is trained to estimate output probabilities associated with transitions in an HMM. The training procedure allows iterative re-estimation of these output probabilities, which are encoded in the weights of the network. Several changes have been made to the CVT system over the course of the past year, as shown in Table 1. These changes are described in more detail elsewhere [5].

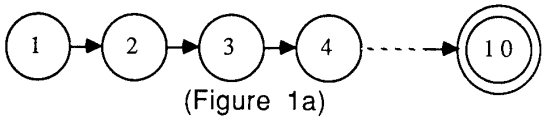
	Word Accuracy	String Accuracy	Incremental Improvement
Baseline ICASSP'90	98.5	95.0	
– recurrence	98.0	94.7	–6%
+ word models	98.7	96.8	+40%
+ corrective training	98.8	97.0	+6%
+ multiple models	**99.1**	**98.0**	+33%

Table 1: Improvements in CVT Results on TI Digits

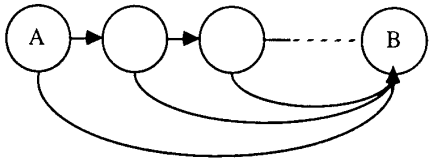
This table shows the incremental improvements to the CVT system since ICASSP'90. The third column shows the reduction in string error rate after making the specified change. The bottom row shows the best results that CVT has achieved on this task.

Since ICASSP '90, we have abandoned the recurrent portion of the neural network. As a result, we incurred a small decline in performance but gained a significant decrease in the computational complexity of the training procedure. Simplifying the network architecture in this way allowed us to devote computational resources to more complex training strategies.

The new implementation of CVT for the Digits task uses Bakis style word models, which allow modeling of finer subword distinctions than the Sphinx-style models [6] used in the previous version of the system.



(Figure 1a)



(Figure 1b)

Figure 1: The CVT Word Model

This figure shows the architecture of the word models used in the current version of the CVT system. Figure 1a is a simplified illustration of one of the word models; each pair of adjacent states shown in Figure 1a actually corresponds to the series of states shown in Figure 1b. For example, states **A** *and* **B** *in Figure 1b could correspond to states 1 and 2 in Figure 1a.*

As shown in Figure 1, the word model is a series of ten sub-series of states. All of the transitions within each sub-series are tied to the same output probability; hence, there is one output unit in the network for each of the ten sub-series in the word model. The time spent in each sub-series is controlled by transition probabilities associated with the transitions in that sub-series.

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%. In our corrective training procedure, the segments of the sentences which are misrecognized are set aside during the forced alignment phase. Segments of correctly recognized sentences in which the output probabilities are low are also isolated. Once these segments are identified, the network is retrained to suppress the score of the misrecognized segments and improve the score of the correctly recognized segments.

A second training strategy which has proved beneficial 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.

4 MULTI-STATE TIME-DELAY NEURAL NETWORKS

4.1 Embedding Time Alignment in a Neural Network

In order to integrate the DP procedure into a connectionist network, a connectionist representation of the DP accumulation of frame scores over time is necessary. The word score will be the output of a connectionist unit that is supervised using a classification based back-propagation learning algorithm.

Traditional DP-based methods attempt to minimize a sum of distances (Dynamic Time Warping) or maximize a product of probabilities (Viterbi alignment). Preliminary experiments have shown that those accumulation procedures are very difficult to integrate in a global connectionist architecture. On the other hand, the summation over time of unit activations has been a highly successful approach with Time-Delay Neural Networks (TDNNs)[7, 8]. However, TDNNs do not model time sequences with multiple states.

Units accumulating activations over time will be referred to as Temporal Accumulator Units or TAU-units in the rest of this paper.

With MS-TDNNs, we have extended the formalism of TDNNs to incorporate multiple sequential states. Suppose there are several states in a word (roughly corresponding to several phones), represented by a series of NN "state units" which are subject to certain sequential constraints. The word-level TAU-unit accumulates the activations of the first state unit over some period of time, then shifts attention to the next state unit and accumulates its activations, until it reaches the ending state of the word. We see in Figure 2 this word-level TAU-unit, which at each time is connected to the state unit in the optimal path, through a "winner take all connection." As in the one-state TDNN, it is essential to add a non-linearity (sigmoid)[7] into the word-level TAU-units so that small deviations at the frame level do not adversely affect overall classification performance.

This process may be seen as an integration of local phonemic decisions over time, where these decisions are constrained to follow the sequence of state (phone) units that make up the word. The MS-TDNN does not require precise phonemic decisions at each point in time (as many current hybrid approaches implicitly attempt), but only *sufficient* evidence for each of the phones that make up the word (subject to the order in which they occur) such that a decision of one word over another can be made. This limits the sensitivity of the system to frame-level classification errors.

Dynamic Neural Networks (DNNs) [3] were the first connectionist architecture to include dynamic programming, but they accumulate one score per state, whereas MS-TDNNs accumulate one score per time frame. The DP procedure used in MS-TDNNs is more easily extended to connected speech [9].

4.2 Architecture

In MS-TDNNs, the formalism of TDNNs has been extended to incorporate multiple sequential states. As seen in Figure 2, the network has 5 main layers of units. From the input layer to the second hidden layer (phone scores), the MS-TDNN is very similar to a phoneme-recognition TDNN [7]. Each unit of the first hidden layer receives input from a 3-frame window of the filterbank coefficients. Similarly, each unit in the second hidden layer receives input

from a 5-frame window of the first hidden layer. At this level of the system (2nd hidden layer), the network produces, at each time frame, the scores of the phones. Our phone set is an extension of the Arpabet phone set to perform continuous alphabet recognition. Some complex phones such as diphthongs or affricates are split into two separate phones.

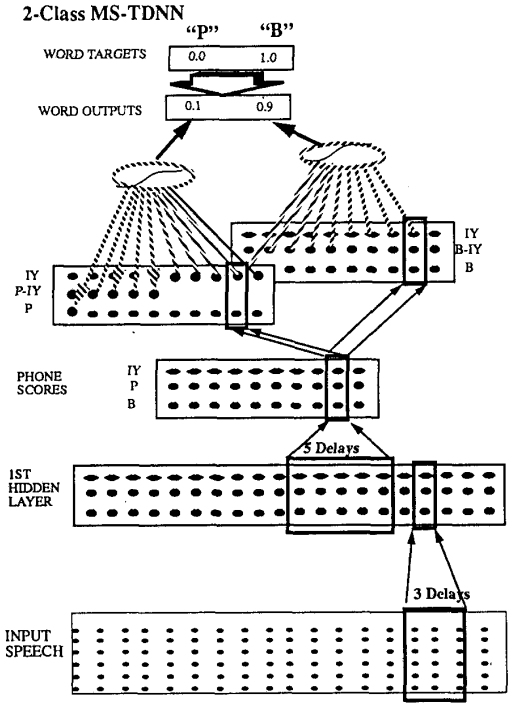


Figure 2: MS-TDNN Architecture

The units in the third hidden layer represent the scores of the word states. As output of the network, there is one TAU-unit per word to recognize. The 2-class MS-TDNN described here as an example has to recognize the words (spoken letters) 'P' and 'B', whose phonemic representations are /P IY/ and /B IY/.

4.3 Integrating Phonetic Knowledge

A closer examination of the units (corresponding to word states) in the third hidden layer and the way they combine phone scores from the second hidden layer will reveal how the phonetic description of the words is integrated in the connectionist architecture.

There are two categories of state units:

Stable state Units which represent the steady-state parts of a word – the parts in which there are no transitions between phones – are known as *stable state* units. For instance, the word 'P' has two stable states. The first one corresponds to the phone /P/, and its score is a weighted version of the /P/ phone score from the 2nd hidden layer. Similarly, the activation of the last state is a weighted version of the /IY/ phone score. It is therefore possible to weight differently the importance of each state belonging to the same word. We do not have to assume that each part of a speech pattern contains an equal amount of information.

Transition state Units which correspond to transitions between phones are known as *transition state* units. With each phone score unit, we associate a phone delta-score unit, which computes the derivative of the phone score. The transition state between state /P/ and state /IY/ will attempt to model the coarticulation between /P/ and /IY/, where the /P/ score is falling and the /IY/ score is rising. Its score is thus a weighted combination of the /P/ phone delta-score (negative) and the /IY/phone delta-score(positive). Those transition units have been found to have a useful segmenting role and to improve global performance.

4.4 Learning

The MS-TDNN training uses a fast back-propagation learning procedure that has been optimized on Japanese phoneme recognition [10]. The same global gradient back-propagation is applied to the whole system, from the output TAU-units down to the input units. For each sentence in the learning set, the sequence of input frames is time aligned with the desired string of words, using SPHINX in a forced alignment mode. Each desired word is therefore associated with a segment of speech with known boundaries, and this association represents a learning sample. The DP alignment procedure is applied between the known word boundaries, except in the bootstrapping phase, during which phone labels are used to give the alignment of the desired word.

Our optimization procedure explicitly attempts to minimize the number of word substitutions; this approach represents a move towards systems in which the training objective is maximum word accuracy. However, insertions and deletions are not yet taken into account, generating some problems when the system is used in recognition mode.

4.5 Recognition

During the recognition phase, time alignment is performed over the whole sentence – a series of connected words – not over individual speech segments which represent single words, as during training. A one-stage DP algorithm [9] for connected words is used in place of the isolated word DP algorithm used in the training phase. Our first recognition runs were plagued with a large number of insertions, which were mostly words corresponding to spelled vowels. The training procedure does not include any penalty for these insertions; however, several simple features of the recognition procedure have reduced the number of insertions to an acceptable level: (1) *Bounded Durations* – the duration of each word is bounded between the minimum and maximum values observed in the training data, (2) *Boundary detection* – word transitions are allowed only when specially trained boundary detectors fire over a given threshold, and (3) *Word entrance penalties* – a fixed penalty is added to the DP score every time a word is entered.

However, the optimization of these "external" features is not part of the main learning procedure described in the previous section. We will see that a possible direction for future improvements will be to integrate them in the connectionist architecture.

4.6 Experiments on Connected Alphabet

Recognizing spoken letters is considered one of the most challenging small-vocabulary tasks in speech recognition. The vocabulary, consisting of the 26 letters of the alphabet, is highly confusable, especially among subsets like the E-set ('B','C','D','E','G','P','T','V','Z') or ('M','N'). Fine acoustical modeling is necessary to discriminate among confusable phonemes such as /b/ and /d/, so that the letter 'B' can be distinguished from the letter 'D'. Many systems have been proposed so far to perform isolated letter recognition. DTW and HMM based systems [11] solve the problem of time alignment, but the necessary discrimination between confusable words is difficult to perform. Based on a classification paradigm, NNs have been shown to perform good discrimination on the isolated alphabet task [8], but those NNs do not implement time alignment. Some preprocessing is required to segment the part of the utterance on which the NN is going to focus its discriminant attention; therefore, this approach is difficult to extend to continuous speech.

Our database consists of 1000 connected strings of letters, some corresponding to grammatical words and proper names, others simply random. There is an average of five letters per string. Two speakers have been collected so far. The system should be able to recognize any kind of spelled word, including foreign proper names and non-grammatical strings of letters; recognition does not make use of any grammar. The strings are labeled with an automatic procedure using the discrete-HMM based SPHINX system [6] in a forced-alignment mode. The SPHINX phone models used in the forced-alignment were trained on a speaker-independent vocabulary-independent task. As input parameters, 16 filterbank melscale spectrum coefficients are computed at a 10 msec frame rate.

For comparison, we have also trained SPHINX speaker-dependently on the same data used in the MS-TDNN experiments. Two versions of SPHINX models were trained: one with context independent phones (monophones), and one with context dependent phones (triphones, descibed in [6]).[2] Three iterations of Baum-Welsh (standard MLE) training were performed. SPHINX used the same input representation as the MS-TDNN system. It should be noted that the SPHINX system was not tuned for this task.

Training the system on 500 strings from one speaker (2500 letters) takes about 1 day on a DEC-3100 workstation; training SPHINX on the same data takes a similar amount of time. Both systems have been tested on the other 500 strings form the same speaker. Table 2 shows the results of training and testing the system on two speakers.

Speaker		MS-TDNN Context Independent	Sphinx Context Independent	Sphinx Context Dependent
JMT	*Word Acc.*	**97.5**	94.0	**96.0**
	String Acc.	89.0	73.6	82.6
DBS	*Word Acc.*	**89.7**	78.3	**83.9**
	String Acc.	59.0	29.6	41.6

Table 2: Comparison of MS-TDNN and SPHINX Results on Continuous Alphabet Task

This table shows the speaker-dependent recognition accuracies of MS-TDNNs for the two speakers studied, JMT and DBS. For comparison, the results of the SPHINX system are shown.

The difference between the two speakers presented here lies partly in the uttering speed: regular and without pauses in the case of JMT; irregular, with random pauses and accelerations in the case of DBS. The difference in performance between the two systems is largely due to the discriminant power of MS-TDNNs, which leads to a lower substitution rate than SPHINX. The SPHINX results show the utility of context dependent phones, which are not yet modeled in MS-TDNNs.

4.7 Improvements

We have presented and tested a learning procedure aimed at reducing the number of word substitutions. The following planned improvements are designed to increase the MS-TDNN system's word accuracy, and especially reduce the number of insertions and deletions.

- Using strings of words as training tokens should more explicitly minimize the global string accuracy; however, pruning among the many possible string counter-examples is needed for this method to be computationally feasible.

- State durations represent information that can be used as input by small specialized networks whose output activations correct the MS-TDNN word score.

5 CONCLUSION

Both of the systems described in this paper demonstrate that high recognition accuracies can be achieved by systems which integrate time alignment and neural networks.

These systems represent two distinct approaches to applying classification-based connectionist learning procedures to continuous speech recognition. In CVT, NN supervision occurs at the state level and is guided by feedback from the time-alignment section of the system; what the CVT system maximizes is the input speech likelihood. In MS-TDNN training, NN supervision occurs at the word level; what the MS-TDNN system maximizes is the word recognition rate.

[2]The 1988 3-codebook version of the SPHINX system was used.

The experiments reported here with CVT applied to the TI Digits task demonstrate that hybrid NN/HMM systems can outperform HMM-based systems on a difficult real-world speech task. The improvements to the CVT system since ICASSP'90 have reduced the error rate on TI Digits by over 50%, to a word error rate of 0.9%.

Multi-State Time-Delay Neural Networks integrate a DP alignement procedure with a connectionist architecture. Their ability to recognize continuous speech, while retaining the discriminant capabilities of classification NNs, leads to a high word accuracy on speaker-dependent connected alphabet recognition: 97.5% and 89.7% for our two speakers.

6 ACKNOWLEDGEMENTS

The authors are indebted to Cindy Wood for diligently gathering the alphabet database, to Joe Tebelskis and Dave Sanner for lending us their voices for several hours, to Kai-Fu Lee for sharing his insightful understanding of speech recognition, to Jay McClelland for sharing his computing resources with us, to Hsiao-Wuen Hon for assistance with the SPHINX system, to Donn Hoffman for his help preparing the camera-ready copy for this paper, to Sondra Ahlen for proofreading this paper, and to Raj Reddy for his encouragement and support.

References

[1] Franzini, M.A., Lee, K.F., and Waibel, A.H., "Connectionist Viterbi Training: A New Hybrid Method for Continuous Speech Recognition," ICASSP'90.

[2] 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.

[3] Sakoe, H., Isotani, R., Yoshida, K., Iso, K., and Watanabe, T., "Speaker-Independent Work Recognition using Dynamic Programming Neural Networks," ICASSP'89.

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

[5] Franzini, M.A., Waibel, A.H., Lee, K.F., *The Connectionist Viterbi Training Procedure*, Tech Report, School of Computer Science, Carnegie Mellon University, February, 1991.

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

[7] Waibel, A.H., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K., "Phoneme Recognition using Time-Delay Neural Networks," in IEEE Transactions on Acoustics, Speech and Signal Processing. 1989.

[8] Lang. K., *A Time-Delay Neural Network Architecture for Speech Recognition*, Ph.D. thesis, Carnegie Mellon University. Available as tech report CMU-CS-89-185, July, 1989.

[9] Ney, H., "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition" in IEEE Transactions on Acoustics, Speech and Signal Processing. April, 1984.

[10] Haffner, P., Waibel, A.H., Sawai, H., and Shikano, K., "Fast Back-Propagation Learning Methods for Large Phonemic Neural Networks" in *Proceedings of Eurospeech*. September, 1989.

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