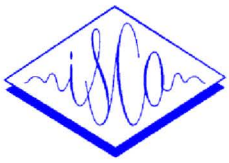

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

TIME-DELAY NEURAL NETWORKS EMBEDDING TIME ALIGNMENT: A PERFORMANCE ANALYSIS.

Patrick Haffner and Alex Waibel*.

Centre National d'Etudes des Télécommunications, Lannion, FRANCE

(*) School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT

Multi-State Time Delay Neural Networks (MS-TDNNs), using a new connectionist architecture with embedded time alignement, have been successfully applied to speaker-dependent continuous spoken letter recognition[1]. This shows the value of extending the classification capabilities of connectionist networks up to the word level in recognizing confusable vocabularies. This paper describes the application of MS-TDNNs to a very different task: speaker independent telephone-quality isolated digit recognition. The resulting 1.6% error rate demonstrates the value of embedded time alignement, since multi-feature TDNNs, which do not implement time alignement, have a 6.5% error rate on the same task. Comparisons with HMMs are also provided.

1 INTRODUCTION

Since connectionist learning procedures are typically defined in terms of static pattern classification tasks, time alignment presents the greatest problem in performing spoken word recognition for neural network (NN) based systems. To take the time distortions that may appear within its boundaries into account, a word is generally modeled by a sequence of states that can have variable time durations. Some recurrent connectionist architectures have been designed 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. One effective solution is to combine an alignment procedure with the NN, generally Dynamic Programming (DP). Traditional DP-based methods attempt to minimize a sum of distances (Dynamic Time Warping) or maximize a product of probabilities (Viterbi alignment).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 is preferably the output of a connectionist unit that is supervised using a classification based back-propagation learning algorithm.

For speech tasks in which modelling sequential state information is not necessary, excellent recognition performance has been achieved using TDNNs[2] or frame level classification NNs. TDNNs-based architectures which do not perform time alignment were experimented on word recognition[3], but the system is not robust to time distortions. To extend TDNN performance to multi-state word recognition, it is necessary to combine the NN with a procedure performing time alignment, usually based on Dynamic Programming[4]. Multi-State Time Delay Neural Networks (MS-TDNNs) perform classification at the word level. Unlike many other hybrid methods[5], these networks are not trained using external frame-level supervision provided by a separate time-alignment section of the system. MS-TDNNs incorporate the DP procedure into their training, such that only external word-level supervision is required. 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 enable this very large NN to learn within reasonable time with minimal external supervision. Recently, MS-TDNNs have been favorably compared to the discrete HMM based SPHINX system on continuous speaker-dependent alphabet recognition[1]. Experiments on the recognition of speaker-independent isolated digits (French, telephone quality) described here show a large increase in performance from static multiple feature TDNNs to MS-TDNNs.

2 TDNNS AS SINGLE FEATURE DETECTORS

TDNNs act as time-shift invariant feature extractors. They are particularly well suited to the recognition of speech patterns distinguished by the presence of a single feature in time. The first task assigned to them was the detection of a single significant acoustical feature used to recognize phonemes [2], this acoustical feature could happen anywhere in a speech segment made of a succession of parameter frames. To train a frame-level NN to perform the correct phoneme classification at each time frame is not desirable: the output unit corresponding to the feature to be recognized would be forced to assume its maximal value even at times when this feature is not present in the input speech frames. During training, TDNNs are only required to produce the correct classification, regardless of the position in time at which the significant feature has occurred. This is achieved by accumulating evidence which is local in time (the outputs of a frame-level NN) to produce an output which is time independent. It is essential to pass this accumu-

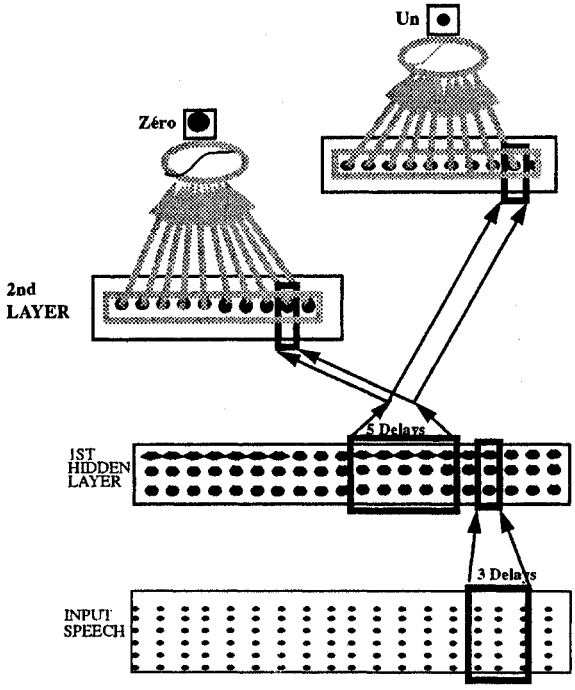

FIG.1 : A single feature TDNN for word recognition

lated sum through a non-linearity (sigmoid) so that small deviations at the frame level do not reduce overall classification performance.

FIG.1 shows a single feature TDNN used for the recognition of the French digits "zero" (0) and "un" (1). Each unit of the first hidden layer receives input from a 3-frame window of filterbank coefficients. Similarly, each unit in the second hidden layer receives input from a 5-frame window of outputs of the first hidden layer. At this level of the system (2nd hidden layer), the network produces, at each time frame, the scores for the desired phonetic features. However, to base the recognition of a word on the detection of only one of its acoustical feature - even when it is the most characteristic one - implies the loss of many other useful features. Single state TDNNs were applied to the recognition of the 10 isolated French digits to check whether the learning procedure could find one feature per digit which would be sufficient for discrimination among the words of this small vocabulary: this system never converged.

3 TDNNS AS MULTIPLE FEATURE DETECTORS

The recognition of one word generally depends on the detection of several consecutive acoustical features. The first attempts to handle multiple feature detection with TDNNs assumed that each of those feature occurred within some fixed temporal window. The output unit corresponding to a word to be recognized combines evidence from a succession of local feature detectors sampled at fixed intervals. FIG.3 shows the

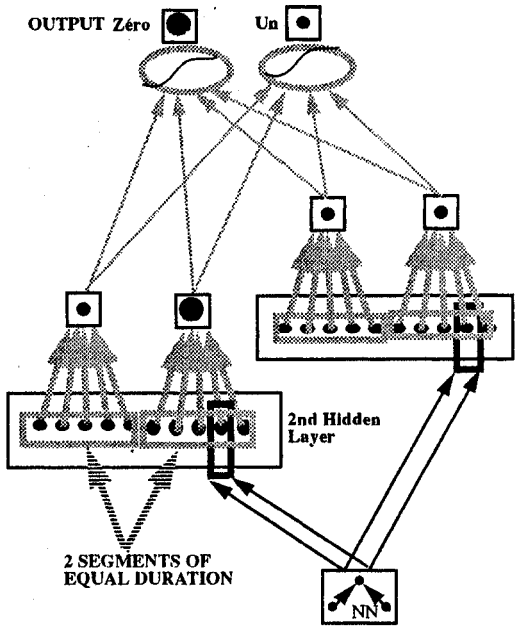

FIG3: A multiple feature TDNN

multiple feature TDNN used again for the recognition of the French digits "zéro" (0) and "un" (1). Up to the second hidden layer, the TDNN is the same as the one described in the previous section. Word boundaries are determined by an external end point detection procedure. Within these boundaries, the utterance is divided into several segments of equal duration (in our example, we have just used 2 segments for simplicity, each of those segments being 5 frame long). This method corresponds to a linear renormalization of time. The activations of the second hidden layer are averaged separately over these segments (the resulting averaged activations appear in the third hidden layer). The third layer is fully connected to the output layer. Our experiments described in section 6 show that this system is not very robust to time distortions.

4 MULTI-STATE TIME-DELAY NEURAL NETWORKS

With MS-TDNNs, we have extended the formalism of TDNNs to incorporate time alignment. Suppose there are several distinct states in a word (roughly corresponding to several phones). These states can be represented by a series of connectionist "state units" which are subject to certain sequential constraints. The word-level temporal accumulating unit accumulates the activation of the first state unit over some period of time, then shifts attention to the next state unit and accumulates its activations, and so on, until it reaches the ending state of the word. We see in FIG.4 this word-level temporal accumulating unit, which at each time is connected only to the currently ac-

tive state unit in the optimal alignment path. As in the one-state TDNN, it is essential that the accumulation units have a sigmoid output function.

This process may be seen as the 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 which limits the sensitivity of the system to frame-level classification errors. The 2-class MS-TDNN described here as an example again has to recognize the digits 'Zero' and 'Un' (FIG.4.). Each word has a 3-state model in this small example. More sophisticated phone models, which are described in a previous paper [1], have also been designed to perform spelled alphabet recognition. The main differences between word models and phone models are the following:

- In the *alphabet task*, some sections of the speech signal (for instance the stop consonant /p/ in the spelled letter 'p') may contain more discriminatory information than others. It is therefore justified to weight the importance of each phone belonging to the same word differently. In the *digit task*, where we use word models whose states are equally distributed in time, this differential weighting is not justified. In this case, experiments have shown it to worsen generalization performance, since the weights add superfluous parameters to the system.
- In *phone models*, a phonemic role is assigned to every state in a word. It is therefore possible to assume that two consecutive phone units fire on detecting different features. The transition from one phone to an other means most often that the score of the first phone is decreasing while the score of the second phone is increasing. It is therefore possible to add specialized transition units that are trained to detect this transition more explicitly: the resulting stabilization in segmentation yields an increase in performance. This cannot be done with *word models*, as some consecutive states may be redundant, since they detect the same feature.

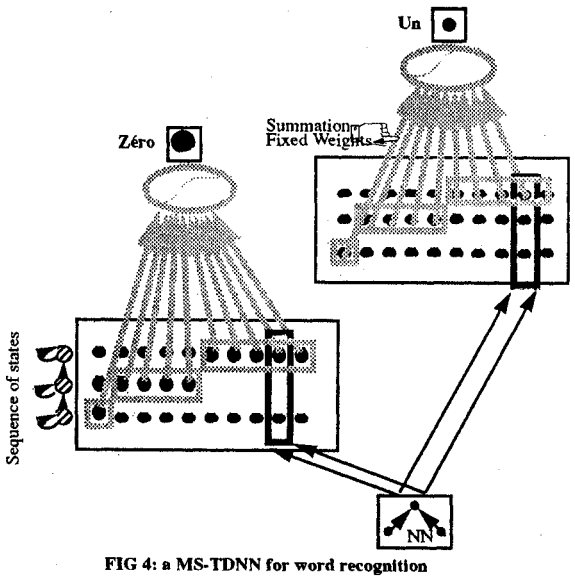

FIG 4: a MS-TDNN for word recognition

5 TRAINING MS-TDNNS

MS-TDNN training uses a fast back-propagation learning procedure that has was developed for a Japanese phoneme recognition task[6]. The same global gradient back-propagation is applied to the whole system, from the output word units down to the input units. Generally, each desired word is associated with a segment of speech with known boundaries, and this association represents a training pair. The DP alignment procedure is applied between the known word boundaries.

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.

6 ISOLATED TELEPHONE DIGITS

The speech material was collected from about 750 speakers, each of whom uttered the 10 digits (French) in isolation over the long distance telephone network. As input parameters, we use 6 Melscale Fourier Cepstral Coefficients (MFCC), computed at a 16msec frame rate, the Energy, and the Energy derivative. Training the network on 3500 digits (375 speakers) takes about one day on a IBM RISC/6000 workstation (100 epochs). Word models with 7 states are used. Results with continuous single density HMMs[7] are provided for comparison, using HMMs which have been developed on the CNET telephone digit database for several years. The best HMM results have been obtained with word models (41 states per word). Since precise end-point detection is extremely difficult on our telephone quality databases, the 6.5% recognition accuracy was the best we could obtain using a fixed sized TDNN. HMMs used in a forced alignment mode provided high quality word boundaries that reduced the error rate to 3.0%. Linear renormalization was applied within these word boundaries. Without a-priori knowledge about the word boundaries (silence models are used), MS-TDNNs achieve a much better performance: 1.6% error rate. This result is still far from the 0.7% error rate obtained with an HMM which has been optimized with respect to the word model (by allowing alternate paths), the number of states and the set of input parameters.

SYSTEM	Description	Error rate
Multi Feature TDNN	**No precise end point detection 4 states/word**	**6.5%**
Multi Feature TDNN	**Forced alignment word boundaries 4 states/word**	**3.0%**
MS-TDNN	**7 states/word**	**1.6%**
HMM (baseline)	**41 states/word Same input parameters as MS-TDNNs**	**2.8%**
HMM (optimized)	**41 states/word Extended set of 30 input parameters (derivatives)**	**0.7%**

CONCLUSION

Multi-State Time Delay Neural Networks were applied to speaker-independent isolated digit recognition, resulting in an four fold reduction in the error rate when compared to TDNNs that do not implement time alignment. No optimization has been tried yet on MS-TDNNs, as these baseline experiments were intended to show the efficiency of the time alignement procedure. It is believed that MFCC input parameters, which have been very successful with HMMs, are not the best possible choice as the input to a connectionist network. Alternate models or different classes of speaker and integrated connectionist speaker (or telephone noise) adaptation represent other possible directions for improvements.

Aknowledgements

The authors would like to express their gratitude to Denis Jouvet and Jean Monné for stimulating discussions, and to Michael Witbrock for reading the drafts.

References

[1] Haffner, P., Franzini.M. and Waibel A., "Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition" ICASSP'91.

[2] 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.

[3] Bottou, L.Y., Fogelman Soulié F., Blanchet P. and Liénard J.S, "Speaker-Independent Isolated Digit Recognition: Multilayer Perceptrons vs. Dynamic Time Warping" in Neural Networks, Volume 3, Number 4, 1990.

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

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

[6] 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.

[7] Jouvet D., Bartkova K. and Monné J., " On the modelization of allophones in an HMM based speech recognition system" Eurospeech'91.