TDNN VS. FULLY INTERCONNECTED MULTILAYER PERCEPTRON: A COMPARATIVE STUDY ON PHONEME RECOGNITION

David B. Grayden and Michael S. Scordilis

Department of Electrical & Electronic Engineering
University of Melbourne

ABSTRACT - The development and performance of a Time-Delay Neural Network (TDNN) and a Fully Interconnected Neural Network (FINN) is compared for continuous speech, speaker-independent recognition of voiced stops and unvoiced fricatives from the DARPA TIMIT speech database. The results conclusively show that the TDNN is the preferred network for phoneme recognition. A major enhancement of the back-propagation is also included, and it makes possible the speedy development of large neural networks on general purpose workstations.

INTRODUCTION

For an unrestricted speech recognition system, it is best to recognise the phonemes at the lowest level and then build words from the phonemes detected. The main difficulty in phoneme recognition is the enormous amount of variability that exists in spoken speech. A single phoneme may differ in duration and distribution of spectral energy depending on the context that it is in. Phonemes are also affected by the manner of articulation, interspeaker variations, environmental conditions and the prosodics of the sentence. These factors all combine to make the phoneme recognition problem very difficult.

One problem is that phonemes spoken in continuous speech tend to blur into each other. This makes it hard to determine where a phoneme begins and ends leading to alignment difficulties in phoneme recognition. It is important to overcome the phoneme alignment problem to achieve optimal performance.

Two neural architectures for phoneme recognition have been investigated. First, the Time-Delay Neural Network (TDNN) was designed in an effort to try and overcome the temporal alignment problem. By constructing a network that is invariant to time shifts while still keeping the number of training passes small, it should be possible to recognise phonemes without having to align them at all. The utterance could just be passed frame by frame through the input to the network and the phonemes detected as they appear.

To compare performance with the TDNN, a Fully Interconnected Neural Network (FINN) was created with the same dimensions as the TDNN. This network was the standard feed forward neural network with all nodes on one layer connected to all nodes on the layer above. Both were trained and tested on presegmented phonemic sequences.

SPEECH FEATURE EXTRACTION

The DARPA TIMIT database was used to provide the training and testing sets. This database contains speech from 630 speakers from 8 major dialects of American English. Each speaker utters 10 sentences. Time-aligned phonetic and word transcriptions provided with the database ease the extraction of the phonemes wanted for training and testing. The database is divided up into standard training and testing utterances. Each speaker utters sentences sa1 and sa2 as well as some more diverse sentences, which differ from speaker to speaker, called si and sx sentences. The testing set does not contain any speakers that appear in the training set.

Speech samples in the TIMIT database are 16 bit integers at a sampling rate of 16,000 Hz. Phonemes were extracted from the database by searching through the phonetic transcription for the required phonemes and copying the relevant section of the speech waveform data. The phonemes were extracted inside 166 ms segments of speech data with the onset of a phoneme at a given distance from the beginning of the segment.

Features to be used as input to the neural networks were extracted via a 256 point Hamming window applied at 5 ms intervals across the speech segment. An FFT was performed on each windowed speech portion giving 30 spectra throughout the segment. The log-magnitude was taken of each of these and then the spectra was rescaled to a 16 point melscale. Adjacent spectra were averaged to obtain 15 frames of data where each frame comprised 10 ms of speech features. Finally, the data was normalised to values between -1 and +1.

For training, all the voiced stops and unvoiced fricatives were extracted from the si and sx sentences of TIMIT. These were formed into two training sets, one containing /b/, /d/ and /g/, and the other /s/, /sh/, /f/ and /th/. The sa sentences were not used as these would give a bias to the phonemes that they contain (i.e., neither of them contain /b/). Each training set was then passed through the above procedure to produce input data whose desired output was +1 for the correct phoneme and -1 for the others. The testing set was extracted and prepared in the same way from the si and sx sentences in the test set designated by TIMIT.

The computers used to develop and run the above and following programs were several Sun SPARCstations. Because of the large amount of training that had to be performed, different training and testing runs were often executed on separate machines.

NEURAL NETWORK CONSTRUCTION

Time-Delay Neural Network (TDNN)

The TDNN was the same structure as that used by Waibel, et al (1989). Each set of connections was made equal to the next set by averaging the weights after each back-propagation update. This was an attempt to make the network independent of temporal shifts across the input frames. A bipolar (-1,+1) sigmoid function was employed (Rumelhart, Hinton, Williams, 1986; Rumelhart, McClelland, 1986).

Initially, training time was excessively long so a number of speed-ups over the conventional back-propagation algorithms were used. The McClelland error as outlined in Haffner (1989) was used to remove the local minima problems as well as increase training speed when error was large. This way, the error is given by

$$E = -\sum_{\text{samples}} \left[\sum_j \ln(4 - (y_j - d_j)^2 \right],$$

where E is the total error, j is an index of output units, y_j is the actual output from the network and d_j is the desired output.

Dynamic epsilon adjustment was also used where the value of epsilon was adjusted for each node depending on the delta values of the weights feeding into it (Haffner, 1989). Thus, the epsilon value for all weights entering node i on a given layer is calculated by

$$e_i = \frac{e}{1 + \left(\frac{e}{w}\right)\sqrt{\sum \left(\frac{\partial E}{\partial W_{ij}}\right)^2}},$$

where e is the global value of epsilon, w is a weighting value (usually 1) and j is an index of nodes in the layer below.

The number of samples passed through the network between weight adjustments was also varied. One option was to gradually increase this number but it was found that the TDNN trained best with 3 samples presented between weight updates.

Significant improvement in the rate of convergence was achieved by calculating the momentum term as an accumulation of the back-propagation passes. This way each weight derivative was added to an array which was used when updating weights and, after updating weights, the value of the sum was multiplied by a momentum factor rather than clearing all past values. Further back-propagation passes were then added to this value so that previous passes still had some diminishing effect on later ones. This increased the convergence rate by about one order of magnitude compared to the conventional momentum calculation (Rumelhart, 1986).

Staged learning strategies were also tried. As described in Waibel (1989), this involved training the network on only a few training samples, and then gradually increasing the number of samples presented to the network. The motivation was that the initial few samples should quickly "teach" the network the basic pattern to recognise and then fine tuning can be done later at a faster rate. However, in these simulations, it was found that the staged learning makes the network significantly "over-learn" portions of the training sets and that it performed significantly better when all data was presented at once.

The training algorithm was given a threshold of error. If the greatest error at an output node for a particular training sample was below this value, then the back-propagation pass was not performed for a number of epochs proportional to the difference between error and the threshold value. This technique decreased the amount of time required to complete one epoch.

The error threshold could also be varied as the network trained. This would allow the network to complete the initial iterations faster when it is training those samples that have large error. However, this method also resulted in over-learning. The network most likely learned the more difficult training samples and ignored the important clues found in the samples that were mostly the same. The network performed best when a low, fixed error threshold was used.

Finally, the back-propagation routine was adjusted so that those output nodes with very small error compared to the others were taken to have zero error. This allowed many of the multiplications to be skipped in back-propagation thus speeding up the algorithm but also decreasing performance somewhat.

In summary, measures taken for network speed-ups were:

- use of the McClelland error
- dynamic epsilon adjustment
- accumulating momentum
- ignoring trained samples

Fully Interconnected Neural Network (FINN)

The same speed-up techniques used in TDNN training strategy could also be used for the FINN. However, the FINN was much faster to train so only a few of them were used, and where:

- use of McClelland error
- ignoring trained samples

Normal momentum was again not used as this tended to make it unstable. Accumulating momentum increased the speed of training but it also decreased performance so it was not used either.

TRAINING AND TESTING

All voiced stops (/b/, /d/, /g/) were extracted from the si and sx sentences in the TIMIT training and testing sets. The unvoiced fricatives (/s/, /sh/, /f/, /th/) were also extracted from the same sentences. Because of the large number of unvoiced fricatives in the training sentences, only 5000 were actually used. Table 1 shows the number of phonemes in each set. The training sets were extracted in random order to prevent biasing.

Due to the averaging performed, the TDNN took much longer to train than the FINN. With more that 1000 samples, the TDNN would not converge for the training set even after one week of training. After 40,000 epochs, which was the limit set for any stage of the training, the average squared error of the network was still fairly high. However, for the FINN, the network trains much faster. After less than 100 epochs of training with the full 5804 sample of the voiced stops training set, the average squared error had converged to less than 0.05, which was the limit set for training termination. For comparison, see Figures 1 and 2 which compare the training curves of the TDNN and FINN respectively. Note that the scale is different in the two graphs.

COMPARISON OF RECOGNITION PERFORMANCE

Both the TDNN and FINN tended to overlearn the training set after a lot of training. The recognition of the aligned testing sets would rise fairly quickly, and then fall gradually as more training was performed.

The performance of the developed networks was first tested for correctly aligned phonemic data. Using the TDNN, the best average recognition for the voiced stops was 87.88%, while for the unvoiced fricatives it reached 89.85%. Using the FINN, the recognition reached 89.08% for the voiced stops and 90.06% for the unvoiced fricatives. Thus, for aligned testing data, the FINN outperforms the TDNN. This is due to the ability of the FINN to converge for the training set while the TDNN must be stopped early. The difference in recognition, however, is small.

To test the sensitivity of the networks to misaligned data, the testing sets were presented to the networks with different offsets of the phonemes from the position used during training. It was this test that revealed the most significant difference in performance between the two networks. As can be seen in Figure 3, in the case of voiced stops, the recognition performance of the FINN dropped off much faster than that of the TDNN. Offsets of up to 2 frames (20 ms) on either side of the training position did not greatly affect the recognition performance of the TDNN, while the FINN's performance was significantly degraded. For offsets greater than 2 frames, the TDNN performed significantly better than the FINN.

In the case of unvoiced fricatives, the TDNN performed better than the FINN for increased amounts of offset, as shown in Figure 4. The difference in performance was much greater for positive offsets. Reduced degradation with negative offset for the FINN may be attributed to the difference in acoustic characteristics between stops and fricatives. The reduced length of stops compared to fricatives, the dynamic nature of stops (closure + burst), and the fact that the identity of stops is provided by its neighbouring phonemes, are all factors that would make stops more sensitive to offset and hence harder to recognize than other sounds (Denes, Pinson, 1963).

CONCLUSION

This paper shows a comparison between the Time-Delay and the Fully Interconnected Neural Network for continuous speech, speaker-independent phoneme recognition. Voiced stops and unvoiced fricatives from the DARPA TIMIT speech database were used for training and recognition. The TDNN was slower in training than the FINN. While the FINN could learn the training data almost perfectly, the TDNN would not converge. However, both networks achieved high recognition rates for correctly aligned phonemes. The TDNN was able to generalize with the training data while not learning the poorer phoneme examples to the same extent that the FINN was able. Evidence of the existence of "harder" examples of phonemes was shown by the FINN which tended to decrease in performance as average error decreased to very low values.

Significant differences in performance appeared when the phonemes became misaligned. The TDNN showed much greater performance although this also decreased with increased misalignment. Employing a phonemic segmenter with the FINN would still not make this network significantly better than the TDNN. Despite the increased learning time, the comparable performance of the networks for correctly aligned data and, more importantly, the robustness of TDNN to misaligned data make it the preferred configuration for recognition of phonemes in continuous speech.

Phoneme	Training set	Testing set
/b/	2181	886
/d/	2432	1245
/g/	1191	775
Total	5804	2886
/s/	2947	2172
/sh/	611	460
/f/	1061	911
/th/	381	259
Total	5000	3802

Table 1. Distribution of phonemes in training and testing sets.

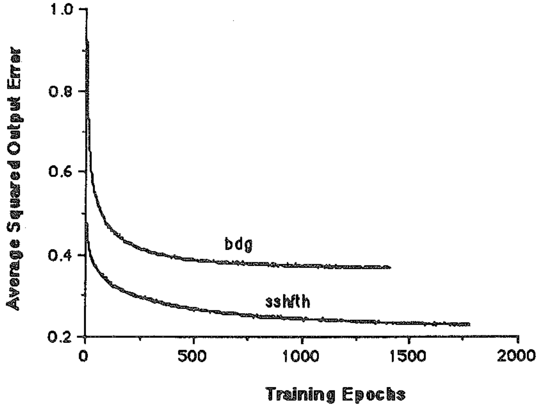

Figure 1. TDNN Training Curves.

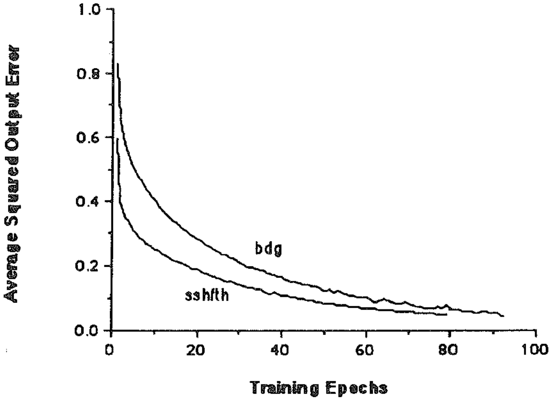

Figure 2. FINN Training Curves.

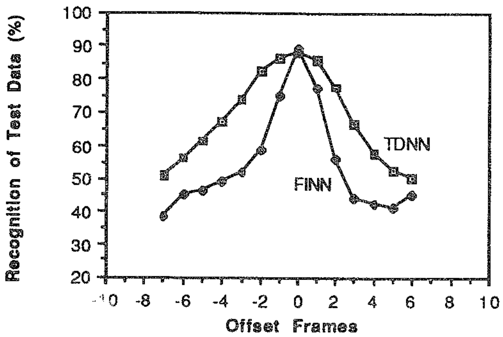

Figure 3. Recognition performance with misaligned testing data: voiced stops.

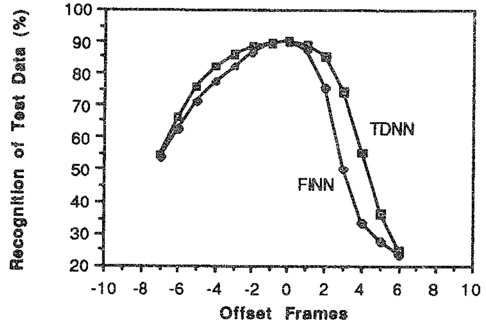

Figure 4. Recognition performance with misaligned testing data: unvoiced fricatives.

REFERENCES

Denes, P.B., Pinson, E.N. (1963) *The Speech Chain: The Physics and Biology of Spoken Language,* (Bell Telephone Laboratories: USA), p.130-5.

Haffner, P., Waibel, A., Sawai, H., Shikano, K. (1989) *Fast Back-Propagation Learning Methods for Large Phonemic Neural Networks,* European Conference on Speech Communication and Technology, Paris, p.553-6.

Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986) *Learning Representations by Back-propagating Errors,* Nature, vol.323, p.533-6.

Rumelhart, D.E., McClelland, J.L., PDP Research Group (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* (MIT Press: Cambridge, Massachusetts; London, England).

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K.J. (1989) *Phoneme Recognition Using Time-Delay Neural Networks,* IEEE Trans. on Acoust., Speech and Signal Proc., vol.37, no.3, p.328-39.