MULTI-SPEAKER DIGIT RECOGNITION USING NEURAL NETWORKS

Danqing Zhang, J.Bruce Millar and Iain Macleod

Computer Sciences Laboratory
Research School of Physical Sciences
Australian National University

ABSTRACT

Application of neural network architectures to the problem of digit recognition is investigated using two different forms of a multi-layer perceptron. The problem of digit recognition is studied from three points of view: firstly, selection of input features representing the spoken digits; secondly, minimisation of training time; thirdly, optimisation of the architecture of neural nets.

INTRODUCTION

Our work on digit recognition addresses the issues of the best form of preprocessing of digit data for input to a network, and the optimum network architecture for this work. Eleven classes of isolated digit utterances spoken by female speakers were selected for this study. They are "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" and "naught". The multi-layer perceptron neural network based on a conventional back-propagation training algorithm has been found to present problems in achieving a set of weights which satisfactorily classifies the training data (Zhang, 1989). Furthermore, the training speed was also very slow. A new fast training algorithm (Brent, 1990) provided a potential advantage in training time and also assisted in the choice of the number of hidden units.

SELECTION OF INPUT FEATURES FOR DIGITS

The first issue to be determined is the selection of features of the spoken digits which may be used to compare reference templates against an unknown input utterance. Features must be selected in the time domain and in the spectral quality domain. We selected low-order cepstral coefficients to represent spectral quality based on the work of Davis and Mermelstein (1980). In the time domain a traditional approach for isolated utterance recognition is to dynamically "time-warp" each input utterance so that by a process of stretching and compressing various parts of the input utterance the best match with a reference utterance is made, according to some parameter such as energy (e.g., Clermont and Butler, 1988). The input frames can then be sampled from a common time-base either using regular or irregular sampling. In the latter case the input utterance can be sampled at fixed points determined by features of the utterance. In selecting input features for these digits, the approach we adopted was to choose a small number of frames located at defined points within each utterance and to use parameters calculated from these frames as the input features. We propose to select these features on a digit by digit basis. However we have not yet achieved robust feature selection from a wide range of speakers and, therefore, for the purposes of this paper a simplified method is used.

In our current system, using this simplified input feature set, the training and classification behaviour of alternative neural net architectures are studied. If a pair of frames positioned at the energy peak in the strongly voiced vowel section and at the point where the energy has fallen to half its peak value are used, then information about the vowel (or diphthong) and also (with the latter frame) about movement of the articulators towards any following consonant is acquired. Similarly, a frame positioned at the half-energy point ahead of the peak energy will capture information about movement of the articulators away from any leading consonant.

Thus, for the current phase of this project we use three frames chosen at the peak of voiced energy and at the leading and trailing half-power points. Identifying significant frames in this manner avoids any requirement for dynamic time warping. There is a potential ambiguity here with respect to the digit "seven", which has two vowels, but all the examples we have inspected from our speech database have significantly higher energy in the first vowel and so this does not present a problem in our initial experiments. Should such a problem arise in other data, two models of the digit "seven" could be used, one having a dominant first syllable and the other a dominant second syllable.

DATA ANALYSIS

The digits were originally recorded with a sample rate of 16, 000 samples per second and a low-pass anti-aliasing filter with a −3dB point of 7.2 kHz. The data used have been resampled at 8, 000 samples per second following digital low-pass filtering to 3.6kHz. In the recording set, each digit is repeated ten times by five different female speakers.

The speech data were first segmented using an energy and zero crossing segmentation process (Clermont, 1990) in order to isolate individual digits. After each utterance was segmented and stored into one file, the data were analysed using a frame of 256 points and a 50% overlap between adjacent frames. Each frame was subjected to a 10th order LPC autocorrelation analysis using a pre-emphasis factor of 0.98. The 10 low-order cepstral coefficients were then derived from the reflection coefficients. The gain of the LPC process was stored for each frame as a measure of the overall signal energy.

Once all frames had been processed, a group of three frames was chosen with one at the "peak" energy and two frames before and after which were closest to half this energy. LPC cepstral coefficients for these three frames were then written to a text file to be used as input data for the neural network.

THE MULTI-LAYER PERCEPTRON USING BACK-PROPAGATION TRAINING

The architecture of the MLP used is a two-layer perceptron with N continuous valued inputs, M outputs and one layer of hidden units. The architecture is shown in Fig.1.

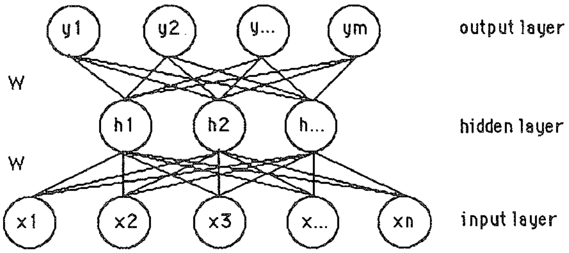

Fig.1

The back-propagation training algorithm is an iterative gradient algorithm designed to minimise the mean square error between the actual output of a multilayer feed-forward perceptron and the desired output (Lippmann, 1987). The network weights are initially undetermined, and a back-propagation algorithm is employed to adjust the weights. The weights are initialised by setting them to small random values. The algorithm goes through the network iteratively, changing the weights in each layer according to the following formula:

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x_i{}' + \alpha(w_{ij}(t) - w_{ij}(t-1))$$

where $w_{ij}(t)$ is the weight from hidden node i (or from an input) to node j at time t, $x_i{}'$ is either the output of node i or is an input, η is a gain term, δ_j is an error term for node j, and α is a smoothing factor which reduces sharp changes in the weight space. The range of α is: $0 < \alpha < 1$. If node j is an output node, then

$$\delta_j = y_j(1 - y_j)(d_j - y_j)$$

where d_j is the desired output of node j and y_j is the actual output. If node j is an internal hidden node, then

$$\delta_j = x_j'(1 - x_j')\sum_k \delta_k w_{jk}$$

where k is over all nodes in the output layer.

RESULTS USING A MULTI-LAYER PERCEPTRON

An initial experiment was performed to determine the optimum number of hidden units. The number of hidden units was varied from 10 to 80 and the performance of the network was monitored. Performance increased significantly when the number of hidden units increased up to 30 but showed no significant increase for more than 30 hidden units. In order to minimise the training time, 30 hidden units were used.

Two methods were adopted to choose training and testing data: (1) four utterances for each class of digit from a given speaker were selected for training and a different utterance for testing, this was named the 4U-1U speaker-dependent test; (2) one utterance from each of four different speakers for each class of digit were selected for training and another utterance from the fifth speaker for testing, this was named the 4F-1F speaker-independent test. Different numbers of initialisations (random starting points in the search space) were tested with each task in both speaker-dependent and speaker-independent tests. The results of the speaker-dependent test shown below give the average and range of four runs, and the results of the speaker- independent test shown below give the average and range of twelve runs.

Speaker-dependent : 4U-1U

Training	Testing	Hidden U.	Ave.Recog.Rate	Range
spk1	spk1	30	100%	100%-100%
spk2	spk2	30	90.33%	81%-100%
spk3	spk3	30	75.00%	63%-90%
spk4	spk4	30	58.50%	54%-63%
spk5	spk5	30	76.50%	72%-81%

Speaker-independent : 4F-1F

Training	Testing	Hidden U.	Ave.Recog.Rate	Range
spk2345	spk1	30	67.5%	63%-72%
spk1245	spk3	30	63%	54%-72%
spk1235	spk4	30	49.50%	45%-54%
spk1234	spk5	30	67.5%	54%-81%

With the experiments using back-propagation training in an MLP, we found that obtaining convergence was a major problem. For example, we failed to train the group of speaker one, speaker three, speaker four and speaker five in our 4F-1F speaker-independent test even with more than fifteen different initialisations. Another drawback of the back-propagation algorithm is that the training time could be very long and we sometimes couldn't get convergence in a practical time. Furthermore, different starting points in the search space could lead to quite different results as evidenced by the range listed above.

It is not clear whether this lack of convergence was due to the back-propagation algorithm used, and specifically its termination criterion, or to the data itself. In a similar study by Millar and Hawkins (1990), this problem did not occur, but in that study both the data and the algorithm used were different. We are investigating this issue.

Another suggestion made by Millar and Hawkins (1990) is that variables whose effect cannot be predicted should be allowed to vary randomly and generate a distribution of results which adequately predicts the range of performance of the system. Hence we report both average recognition rate and the range of recognition rates achieved using different starting points in the search space.

THE FAST TRAINING ALGORITHM

A new training algorithm called Fast Training Algorithm has been introduced by Brent (1990). The architecture used to implement the fast training algorithm is a three-layer net with two hidden layers and one output layer. The basic idea of this approach is to determine an architecture and weights for an MLP by developing a decision tree which is optimised to discriminate between the classes of data in the training set. A trained-MLP is then constructed from the parameters of the decision tree. It is not necessary to specify the number of hidden units in advance, rather a minimum number is determined by this process.

Suppose the training set is S, then we may construct a decision tree T with t nonterminal nodes and $t+1$ leaves. The concept of the fast training algorithm is to find a hyperplane H to split the training set S into two sets S_0 and S_1 in an optimal way. Several criteria can be employed in the optimisation, which are then applied recursively to S_0 and S_1, as often as necessary to construct a decision tree T which correctly classifies all points in S. One criterion shown by Brent to give good results is to maximise

$$\log \frac{\prod_{k=1}^{K} m_{0,k}!m_{1,k}!}{(\sum_{k=1}^{K} m_{0,k})!(\sum_{k=1}^{K} m_{1,k})!}$$

where $m_{i,k}$ is the number of training points of class k in S_i and K is an upper bound on the number of classes. On each recursion, the search for the next hyperplane is initiated using a vector of random weights. These weights are determined from a single seed at the commencement of the algorithm.

The correspondence between the decision tree T and the derived MLP neural net N is that there are t units in the first hidden layer, $t+1$ units in the second hidden layer and at most K output units, where the K is an upper bound on the number of classes. In our test task, K is eleven.

This fast training algorithm is faster by a factor of order $t^2/\log t$ compared to the back-propagation algorithm. The main reason is that maximising the function above is a problem with n degrees of freedom, whereas back-propagation aims to optimise at least nt parameters simultaneously (where t is the number of hidden units in the first hidden layer of the neural net and n is the number of output units).

RESULTS USING THE FAST TRAINING ALGORITHM

The fast training algorithm was used with speaker-dependent and speaker-independent tasks in the same way as introduced for the back-propagation based MLP. Four different random seeds were tested with the speaker-dependent task and no differences were found – the recognition rate was always 100%. In the speaker-independent task, twelve different random seeds were tested to get average recognition rates. In addition, the results also indicated 21 hidden units were required in speaker-dependent tests and different numbers of hidden units were required in speaker-independent tests (see table).

Speaker-dependent : 4U-1U

Training	Test	Hidden U.	Ave.Recog.Rate	Range
spk1	spk1	21	100%	100%-100%
spk2	spk2	21	100%	100%-100%
spk3	spk3	21	100%	100%-100%
spk4	spk4	21	100%	100%-100%
spk5	spk5	21	100%	100%-100%

Speaker-independent : 4F-1F

Training	Testing	Hidden U.	Ave.Recog.Rate	Range
spk2345	spk1	33	73.49%	67.27%-78.18%
spk1345	spk2	27	83.36%	78.18%-87.27%
spk1245	spk3	27	83.33%	78.18%-85.45%
spk1235	spk4	27	76.06%	72.73%-78.18%
spk1234	spk5	27	82.43%	80%-85.45%

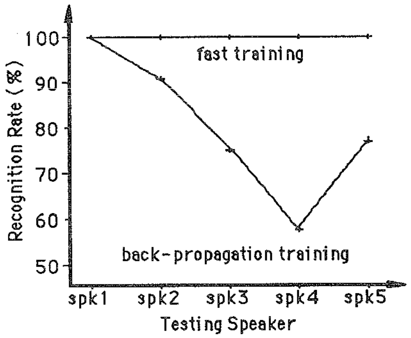

Fig.2 Speaker-dependent tests

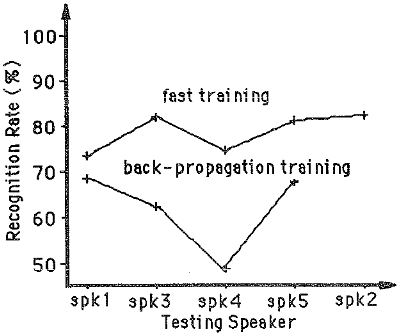

Fig.3 Speaker-independent tests

DISCUSSION

This paper has identified three domains in which the use of neural nets for isolated digit recognition can be explored and optimised. The first, the selection of input features has been fixed in this study, hence cannot contribute to the discussion of our current results. The second, is the training and the third is the architecture of a neural network. A basic MLP architecture is used but varied in terms of its training procedure and number of hidden units. The back-propagation training algorithm with a fixed number of hidden units (30) is compared to the fast training algorithm with a data-dependent number of hidden units. The fast training algorithm provides better results both in a speaker-dependent mode (Fig.2) and a speaker-independent mode (Fig.3). Important advantages of the fast training approach are that it avoids problems with lack of convergence experienced using a back-propagation algorithm and results in a dramatic decrease in training time. On the other hand, the fast training algorithm based MLP architecture tries to reduce the system structure by minimising the number of hidden units solely on the basis of the training data. This could have the effect of encoding highly detailed information about that training data into the neural network weights. We are investigating the degree to which this stored information allows generalisation to test data outside the training set in a separate set of experiments. Some modifications to the fast training algorithm (e.g., setting a minimum number of hidden units) may be necessary if we find that this algorithm gives inadequate generalisation.

Peeling and Moore (1988) indicate the choice of start-up weights was not found to be very important in their similar study using a back-propagation based MLP architecture. Moreover, an MLP with one hidden layer gave a better performance than that an MLP using two hidden layers. In our experience, start-up weights used in an MLP with one hidden layer could directly affect the speed of convergence of training and lead to quite different results when applied in recognition mode.

Our experiments indicate that the fast training algorithm which gives higher average recognition scores and generally reduced ranges of scores depending on initiation of training procedures is a suitable basis for further work on the digit recognition task. The next stage in our work is to define an appropriate set of input features for the digits. The digits consist of at most two vowels with an initial, middle and/or final consonant. The vowels are easily located by finding peaks in voiced energy and frames selected in the vicinity of these peaks will convey useful information about the vowel class. Of the eleven digits, six can have distinct diphthongs in Australian English. In general, the energy peak of the diphthongs in this context occurs on the first of the two vowel qualities, with the energy decreasing through the transition and into the second vowel quality. Thus, to identify the diphthong we need to select at least two frames within the strongly voiced section – say, one frame near the peak energy (to capture information about the first vowel quality) and one at a later point where the energy has been reduced to half. In the case of both monophthongs and diphthongs, the later frame may also capture some information about the movement of the articulators towards any following consonant (as noted earlier).

The consonants present more of a problem because they can come from one of several broad classes (liquid, nasal, fricative or plosive). In addition to those frames which are positioned to capture information about vowel classes, we also want to select frames which help identify initial, middle and final consonants. Because of the generally much lower energy associated with consonants, particularly if they are unvoiced, we intend to use different parameters measured from the speech waveform to help position these frames. There is no simple solution to this problem because of the very different characteristics of the different classes of consonants and the importance and subtlety of the acoustic dynamics (such as in the plosives). Note, however, that this scheme for identifying significant frames allows a very flexible definition of frame locations.

Millar and Hawkins (1990) show that the selection of speakers used to train the system is a significant factor in determining speaker performance. Given a large number of speakers the principles of speaker selection outlined by Millar and Hawkins could also be used to optimise digit recognition.

ACKNOWLEDGEMENTS

The authors are most grateful to the Australian Telecommunications Corporation and P. Duke for supplying the digits data for our research. The author D. Zhang would also like to thank F. Clermont for all the help he has given in this project, and A. C. Tsoi for exciting her interest in neural networks while she was studying in University College (ADFA), University of New South Wales.

REFERENCES

Brent,R.P. (1990) *Fast training algorithm for multi-layer neural nets,* Technical Report, Computer Sciences Laboratory, the Australian National University, TR-CS-90-01.

Clermont,F., Butler,S.J., (1988) *Prosodically guided methods for nearest neighbour classification of syllables,* Proceedings of the Second Australian International Conference on Speech Science and Technology, November 1988, pp216-221.

Clermont,F., (1990) *Unpublished PhD thesis,* Computer Sciences Laboratory, Research School of Physical Sciences, the Australian National University.

Davis,S.B., Mermelstein,P., (1980) *Comparison of parametic representations for monosyllabic recognition in continuously spoken sentences,* IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.28, 1980, pp357-366.

Lippmann,R.P., (1987) *An introduction to computing with neural nets,* IEEE ASSP magazine, April, 1987, pp5-22.

Millar,J.B., Hawkins,S.R. (1990) *Selecting representative speakers,* ESCA/ETR workshop on Speaker Characterisation, Edinburgh, June 25-28, pp161-166.

Peeling,S.M., Moore,R.K. (1988) *Isolated digit recognition experiments using the multi-layer perceptron,* Speech Communication, Vol.7, 1988, pp403-408.

Zhang,D., (1989) *Report on neural network project - consonant-vowel pair recognition,* Dept. of Electrical Engineering, University College, University of New South Wales, 1989, 17pp.