Proceedings of The Institute of Acoustics

# THE MULTI-LAYER PERCEPTRON AS A TOOL FOR SPEECH PATTERN PROCESSING RESEARCH

S.M. Peeling, R.K. Moore and M.J. Tomlinson

Speech Research Unit, RSRE, Malvern, Worcs, UK

## INTRODUCTION

A pressing problem in speech pattern processing is the construction of stochastic models which have appropriate internal *hidden* structures [1]. Normally, the structure of a model is defined using a-priori constraints based on some understanding of the nature of the relevant patterns; for example in hidden Markov modelling the number of states in a model may be defined by the expected number of roughly stationary spectrum regions in a spoken word [2]. The stochastic parameters of such a model are then estimated using information extracted from a set of training examples of actual speech patterns; in hidden Markov modelling the state transition probabilities and the state output probabilities are derived in this way.

However, recent work in this laboratory [3,4] and elsewhere [5,6] has been concerned with model building strategies which are capable of learning structural as well as stochastic information.

This paper introduces the *multi-layer perceptron* (MLP) [7,8] as a new approach to discovering appropriate internal representations of speech patterns.

## THE MULTI-LAYER PERCEPTRON

Like the Boltzmann machine [10], the MLP is a member of the class of self-organising machines known as *adaptive parallel distributed processing networks* [9]. In this formalism, a-priori speech knowledge is expressed in the pattern of weighted connections in a network of very simple processing units. Input data is presented to the network as a pattern of activity on the input units, and the interpretation of the input data is represented by the resulting activity on the output units. The information embedded in the network is refined by adjusting the weights in order to produce the required input-output relationship. The advantage of the MLP over the Boltzmann machine is that it is more tractable computationally.

### The Perceptron

As its name suggests, the MLP is related to work done in the 1960's on simple two-layer associative networks with no hidden units known as *perceptrons* [11]. In the perceptron a set of input patterns is mapped directly to a set of output patterns and a learning algorithm is used to adjust the weights on the input-output connections in order to optimise the accuracy of the mapping. However, it was found that two-layer networks cannot compute many important functions (for example, it is impossible to perform an exclusive-or operation without at least one hidden unit) and there was no known learning algorithm for networks with hidden units. Such a learning algorithm has now been discovered;

Rumelhart et al [7] have shown that it is possible to generalise the original perceptron learning algorithm to handle *multi-layer* feedforward networks.

Multi-Layer Networks

The units in a multi-layer perceptron are arranged with a layer of input units at the bottom, any number of intermediate layers, and a layer of output units at the top. Connections within a layer or from higher to lower layers are not permitted. Each unit has a real-valued output (between 0 and 1) which is a non-linear function of its total input. For example, the total input, $x_j$, to unit j is given by:-

$$x_j = \sum_i y_i w_{ij}$$

A unit has an output, $y_j$, which is a non-linear function of its total input.

$$y_j = \frac{1}{1 + e^{-x_j}}$$

Thus, given an input pattern, the output pattern can be computed in a single *forward pass* through the network.

Error Back-Propagation

If a unit j is an output unit then, for a given target value $t_j$, the total error E at the output is defined by the following expression:-

$$E = \frac{1}{2} \sum_c \sum_j (t_{jc} - y_{jc})^2$$

where c is an index over input-output pairs.

The learning algorithm minimises E by gradient descent. This involves changing the weights according to the following rule:-

$$w_{ji}(n+1) = \varepsilon \delta_j y_i + \alpha w_{ji}(n)$$

where $\varepsilon$ is the learning rate, $\alpha$ is a 'momentum' term and $\delta$ is a measure of the local error at unit j.

For an output unit, the error term is given by the expression:-

$$\delta_j = (t_j - y_j).y_j.(1-y_j)$$

and for an internal (hidden) unit the expression is:-

$$\delta_j = \sum_k \delta_k w_{kj}.y_j.(1-y_j)$$

From the foregoing it can be seen that the learning algorithm changes the weights by apportioning the error at the output using a *backward pass* from the output layer to the input layer; hence the term 'error back-propagation'.

The effect of the learning algorithm is thus to 'discover' a set of weights which produce an appropriate non-linear transformation between input and output. The MLP is thus a powerful technique for deriving high-order internal representations and its computational requirements are quite modest.

## APPLICATION TO SPEECH PATTERN PROCESSING

In order to assess the relevance of the MLP to speech pattern processing, a number of speech related problems have been studied. Of particular interest is the MLP's ability to derive internal structure automatically. Plaut et al [12] have described a synthetic problem which involves a MLP learning to discriminate between 'rising' and 'horizontal' patterns in a spectrogram-like representation. However, the experiments reported here are based on real speech patterns.

Three pilot experiments were performed using vowel spectra, whole-word patterns and facial images. These are described in more detail below.

After each MLP was trained, the resulting weight patterns could be displayed graphically by mapping the input array onto each hidden unit. Each point in this unit then represents the weight associated with the connection between that input element and the hidden unit. The size of the point is proportional to the strength of the connection and colour is used for the sign: black for a negative weight and white for a positive.

### Vowel Spectra

The input patterns consisted of average spectra for six different vowels /æ,ɛ,ɯ,ʌ,ɑ,ɘ/ obtained from a single speaker. Synthetic noise was added in order to simulate different levels of variability. The spectra were mapped onto a ten-by-sixteen grid of input units reflecting the amplitude and frequency dimensions of the pattern. All the input units which lay below the spectrum were set to 1 ('on'). The system contained twelve hidden units and six output units (each corresponding to a vowel class).

Very little structure was apparent in the weights learnt by the system, except for those between one hidden unit and the input units shown in Figure 1. From the weight pattern between the hidden units and output units it could be seen that this unit discriminates between the vowels /ʌ/ and /ɑ/. This unit is 'on' for the vowel /ʌ/, and 'off' for /ɑ/. The lower graphs in Figure 2 show the prototype spectra for these two vowels whilst the upper show the maximum possible variation with noise added. The several strong positive weights in Figure 1 can be seen, by reference to Figure 2, to be in regions only attained by /ʌ/. Similarly the strong negative weight on the right hand side is at a value more likely to be attained by the vowel /ɑ/.

The recognition rate of the MLP was over 90% for an effective signal-to-noise ratio of 3 dB.

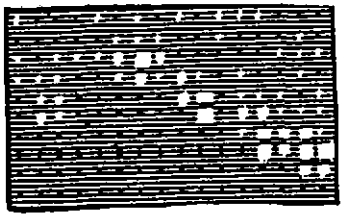

Fig.1:Weights between a hidden unit and the input units for vowel spectra.

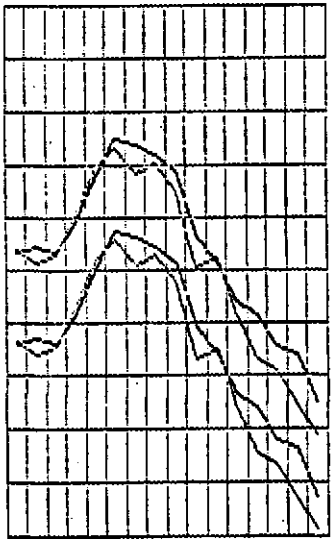

Fig.2:Prototype spectra for the vowels /A/ (solid line) and /Q/ (dotted line).

Whole-Word Patterns

The MLP was also applied to a more complex problem, on which established speech recognition techniques had already been used. The first set of data consisted of the ten digits ("zero" to "nine") spoken by three different speakers under controlled conditions, chosen from the lower end of the consistency table [13]. In the list of 40 speakers JR is 40th, DJ is 36th and AM is 32nd. The training set consisted of ten examples of each of the digits.

The second set of data consisted of minimal pairs e.g. "league/leek", "close/cloze", "chip/ship" and "five/fife". Again direct comparison with existing techniques was possible.

The data were obtained from a 19 channel vocoder with a 20ms frame length. The words occupied between 35 and 60 frames. In each case the input array was set to be 19 x n, where n is the maximum word length in that experiment. Shorter words were padded with silence (zeros). Initially the words were left justified but for the results reported here the words were randomly positioned within the input array.

| DATA | ERRORS | | | |
|---|---|---|---|---|
| | Template | Statistical | Adaptive Network No Hidden | Adaptive Network With Hidden |
| Digits (errors in 100) | | | | |
| spkr AM | 0.98 | 0 | 4 | 1 |
| spkr DJ | 3.82 | 0 | 5 | 0 |
| spkr JR | 11.25 | 1 | 4 | 3 |
| Pairs (errors in 20) | | | | |
| league/leak | 5.87 | 1 | 0 | 0 |
| chip/ship | 6.12 | 5 | 4 | 1 |
| five/fife | 5.47 | 3 | 4 | 1 |
| cloze/close | 2.32 | 0 | 0 | 0 |

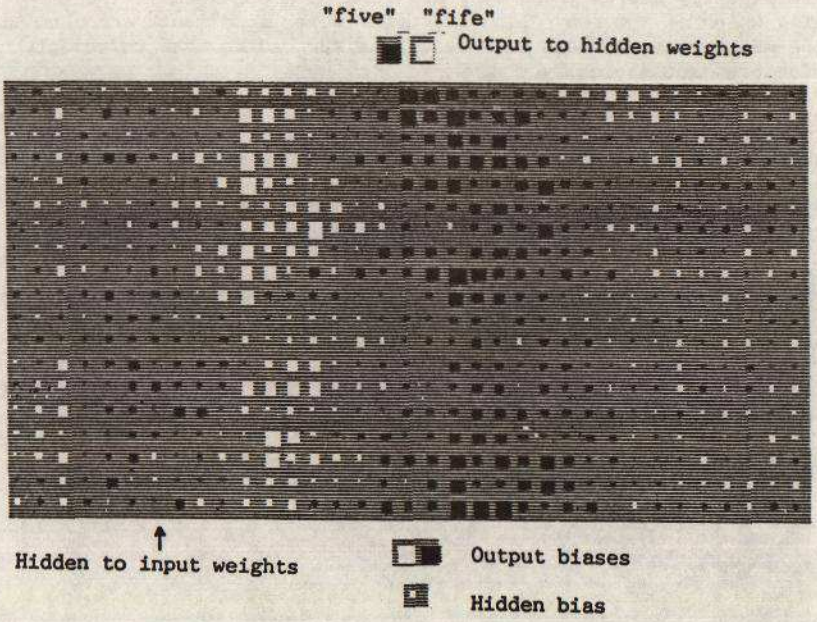


Fig.3:Weight pattern for the minimal pair "five/fife" with 1 hidden unit.

Experiments were conducted using varying numbers of hidden units, including

none.

The results obtained are shown in the Table. Direct comparison was possible with template matching methods (i.e. dynamic time-warping) and statistical methods (i.e. hidden Markov modelling). Full details of these two sets of results can be found in [14].The weight patterns obtained from the 10 digits are difficult to interpret, mainly due to the large number of outputs. A typical weight pattern obtained for the minimal pairs "five/fife" is shown in Figure 3.

From these results it would appear that the MLP is less data dependent than the more traditional methods and is achieving a level of performance which is comparable with current statistical methods.

Facial Images

It has already been shown that lip movements can provide valuable infomation for human speech perception [15] and for automatic speech recognition [16]; the extra information provided by lip movements is speech related but non-acoustic so could be especially useful in a noisy environment. Previously, the problem lay in deciding which of the many lip features were important and how to instrument the measurement of such features. The MLP can be used to reveal the interesting features automatically.
The data consisted of digitised images of lip shapes for the vowels /u/, /i/ and /a/. The MLP used images containing 16 x 16 pixels.

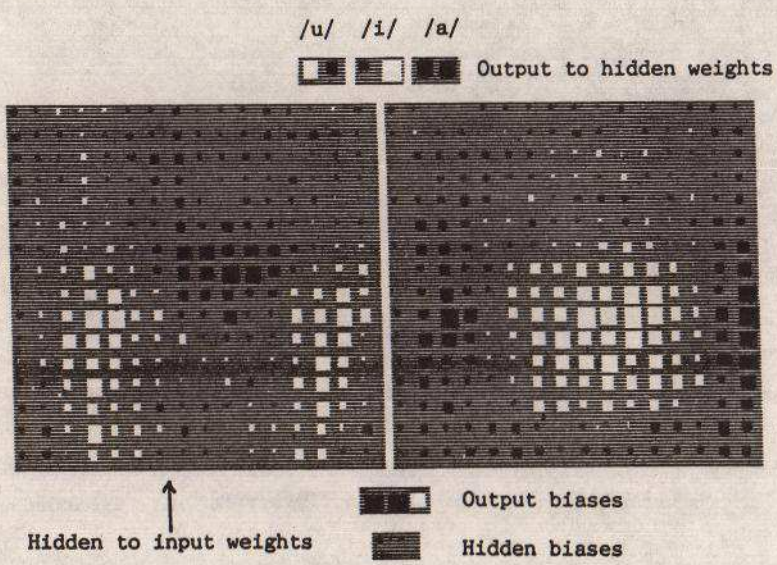


Fig.4:Weight pattern for a lip picture with 2 hidden units.

The MLP is only able to perform the task with two or more hidden units. In the case of two hidden units it is easy to explain the behaviour of the system. From the output and hidden weight patterns in Figure 4 it can be seen that /u/ classification relies upon no teeth being visible and there being no gaps at the side of the mouth. For /i/ there should be no gaps but both teeth and tongue should be visible. (The tongue is the line of strong positive weights in the right-hand hidden-to-input unit.) For /a/, classification relies upon the tongue not being visible.

The MLP was able to classify the lip pictures with no errors provided two or more hidden units were used. It proved totally incapable of discriminating with zero hidden units. If more than two hidden units are used, the weight pattern repeats.

## CONCLUSIONS

The initial choice of startup weights can be crucial. For some choices the MLP will totally fail to find a solution and will remain stuck in a local minimum. For this reason it is advisable to run several experiments on the same data using different startup values. Experience suggests that the MLP is likely to get stuck about 20% of the time.

There is no easy way of deciding how many hidden units should be used. A good initial choice seems to be two per output class. With the ten digits, using twenty hidden units certainly overspecifies the problem. However it is not safe to just pick out the units which seem to be useful. Even weights which appear to have no structure can be performing a useful task. Using too few hidden units can result in a system which fails to learn. In some cases it is difficult to know whether the failure is due to startup values or too few hidden units.

Using too may hidden units can be useful for experimenting with other parameters such as learning rate and scaling for the momentum term. We have found that a learning rate of 0.5 with a momentum term of 0.5 is a good starting point in most cases.

A decision must be made on terminating the learning process. To date, termination has usually been after a specified number of pattern presentations. It is difficult to know when the learning process has converged. Termination on reaching a specified error can be dangerous in cases where the error is oscillating. In other cases, after fairly rapid convergence the learing can slow down drastically so that an extra 1000 iterations may reduce the error very little. Certainly less than 95% correct recognition on the training set indicates that the MLP has not finished learning. In the successful experiments reported here training has usually resulted in zero errors when testing on the training data.

The results obtained here and in [17] suggest that the MLP is a very useful tool for speech pattern processing research.

REFERENCES

[1] R.K. Moore and J.S. Bridle, 'Speech Research at RSRE', Proc. IoA Autumn Conf. on Speech and Hearing, (1986).

[2] R.K. Moore, 'Computational techniques', *Electronic Speech Recognition*, G. Bristow (ed.), Collins, 130-157, (1986).

[3] R.K. Moore, M.J. Russell and M.J. Tomlinson, 'The discriminative network; a mechanism for focusing recognition in whole-word pattern matching', Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, 1041-1044, (1983).

[4] J.S. Bridle and R.K. Moore, 'Boltzmann machines for speech pattern processing', Proc. IoA, Vol.6, Part 4, 315-322, (1984).

[5] G.D. Tattersall and R.D. Johnston, Self organising arrays for speech recognition', Proc. IoA Autumn Conf., 323-331, (1984).

[6] T. Kohonen, H. Riittinen, E. Renhkala and S. Haltsonen, 'On-line recognition of spoken words from a large vocabulary', Info. Sciences, Vol.33, 3-30, (1984).

[7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, 'Learning internal representations by error propagation', *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, D.E. Rumelhart and J.L. McClelland (eds.), MIT Press, (1986).

[8] G.E. Hinton, 'Learning distributed representations of concepts', Proc. 8th Annual Cognitive Science Soc. Conf., Amherst, Mass., (1986).

[9] J.S. Bridle, 'Adaptive networks for speech pattern processing', Proc. NATO ASI on Pattern Recognition Theory and Applications, Spa, Belgium, (1986).

[10] D.H. Ackley, G.E. Hinton and T.J. Sejnowski, 'A learning algorithm for Boltzmann machines', Cognitive Science, 9, 147-169, (1985).

[11] M.L. Minsky and S. Papert, *Perceptrons*, MIT Press, (1969).

[12] D.C. Plaut, S.J. Nowlan and G.E. Hinton, 'Experiments on learning by back propagation', Technical Rept. CMU-CS-86-126, Dept. Computer Science, Carnegie-Mellon Univ., (1986).

[13] D.C. Smith, M.J. Russell and M.J. Tomlinson, Rank-ordering of subjects involved in the evaluation of automatic speech recognisers', RSRE Memo. No.3926, (1986).

[14] M.J. Russell, 'Experiments in isolated word recognition using hidden semi-Markov models', RSRE Memo. (in preparation).

[15] N.M. Brooke and A.Q. Summerfield, 'Analysis, synthesis and perception of visible articulatory movements', J. Phonetics, Vol.11, 63-76, (1983).

[16] N.M. Brooke and E.D. Petajan, 'Seeing speech: investigations into the synthesis and recognition of visible speech movements using automatic image processing and computer graphics', Proc. IEE Conf. on Speech Input/Output; Techniques and Applications, 104-109, (1986).

[17] S.M. Peeling and J.S. Bridle, 'Experiments with a learning network for a simple phonetic recognition task', Proc. IoA Autumn Conf. on Speech and Hearing, (1986).