Methods for integrating memory into neural networks applied to condition monitoring

J F Dale Addison, Stefan Wermter, Kenneth J McGarry, John MacIntyre

University of Sunderland, Centre for Adaptive Systems, St Peters Campus, St Peters Way, Sunderland, England, SR6 0DD

e-mail dale.addison@sunderland.ac.uk.

Abstract

A criticism of neural network architectures is their susceptibility to "catastrophic interference" the ability to forget previously learned data when presented with new patterns. To avoid this, neural network architectures have been developed which specifically provide the network with a memory, either through the use of a context unit, which can store patterns for later recall, or which combine high-levels of recurrency coupled with some form of back-propagation. We have evaluated two architectures which utilise these concepts, namely, Hopfield and Elman networks, respectively and compared their performance to self-organising feature maps using time- smoothed moving average data and Time delayed neural networks. Our results indicate clear improvements in performance for networks incorporating memory into their structure. However the degree of improvement depends largely upon the architecture used, and the provision of a context layer for the storage and recall of patterns.

Keywords: Neural networks, memory systems, recurrency

Introduction

Both multi layer perceptrons (MLP's) [1] and radial basis function networks [2] have been shown to be excellent function approximators, utilising either hidden units with sigmoidal transfer function, hidden units using a distance propagation rule and a Gaussian or some other transfer function respectively. For this reason they have been employed with great success in both classification and regression analysis problems. They are particularly effective in dealing with non-linear relationships between parameters. However in their deployment they require much more caution then is necessary for an equivalent linear time series estimation technique for the following reasons:-

- They require large amounts of sample data, due to the large number of "degrees of freedom" built into such models.
- Several serious problems can arise during training, such as overfitting, sub-optimal minima as a result of estimation (learning), etc., which are more severe than in the linear case (here overfitting can arise by choosing the wrong value for a model parameter)
- They have no means of providing for the linear case in a trivial way.

The first point is particularly relevant for many real-world applications where limited amounts of data are available. The second point concerns the type of learning algorithm employed, back-propagation of error being the algorithm of choice to obtain an optimal model. A crucial point in processing time varying signals is how to represent past inputs or "history", and how this history affects the response to the current inputs. This is achieved by storing inputs in the recent past and presenting them for processing along with the current input [7] [8]. Alternatively past events can be indirectly represented by a suitable memory device such as a series of possibly dispersive time-delays, feedback or recurrent connections, in the internal states of the neurons [3] [4] [5].

In this work we have compared and contrasted a variety of networks which incorporate storage of previous and current patterns and applied them to the domain of novelty detection of electrical output from gas turbine data. The architectures used here are, Hopfield networks [3], Elman Networks [4], simple two stage Time delayed neural networks [5] and a modified Kohonen network which incorporates a temporal element in their input structure allowing smoothed inputs to be presented to the networks [6], [7]. The structure of this paper is as follows. Section 1 will describe the architectures outlined above. Section 2 will discuss the experimental method applied. The results will be considered in section 3, and conclusions and further research will be suggested in section 4.

1 Networks incorporating memory.

In this section we briefly outline the means by which each of the architectures incorporates memory. We begin with the Hopfield network which uses associative memory and recall, a concept which allows for the recall of entire patterns, based upon a fraction of that patterns presented to the network which acts as a "cue" for the recall phase.

1.1 Hopfield networks

The version of the Hopfield network functionality used here can be found in [3]. Essentially it attempts to store a specific set of equilibrium points such that once an initial condition is provided, the network eventually comes to rest at that design point. The network is recursive, in that the output is fed back as the input once the network is in operation. Hopefully the network will settle on one of the original design points. The design utilised here attempts to minimise the chances of falling into a spurious equilibrium point by ensuring such points are made as small as possible [3].

1.2 Elman Networks

Elman networks contain an internal feedback loop which, which makes it capable of both detecting and generating temporal patterns [4]. Elman networks have the ability to approximate any input/output function with a finite number of discontinuities owing to their use of a two layer sigmoid/linear architecture. The provision of a context layer to which patterns can be copied directly following learning provides the memory comparison system for this type of network. Figure 1 illustrates the general architecture underlying networks with recurrent structures.

1.3 Time delay neural networks.

Time delay neural networks (TDNN's) consist of a complete memory temporal encoding stage followed by a feed forward neural network [5]. In this work, TDNN's exclude those architectures which include hidden unit delays, concentrating on simple two stage models. The TDNN architecture used here consists of a tapped delay line followed by a multi-layer perceptron [1]. The output of the tapped delay line is an *N*-dimensional vector, made up of the input signal at the current time, the previous input signal etc. This has the advantage of ease of mathematical analysis, and training regimes. Figure 2 at the end of this paper explains the architecture in more detail.

1.4 Time smoothed moving averages.

The final technique used in this evaluation borrows from the work of [6] and [7]. We utilise a flow vector, **f**=[x,y, dx. dy], containing second order information, motivated by the notion that the rate of change in the level of electrical output, provides an obvious source of discriminatory behaviour. The position first and second order elements to this vector, provide a vector which contains information regarding previous behaviour of the data. A time smoothing function was then applied to the elements in the vector. This can be done by using all instantaneous elements with a smoothed value for each. The feature vector used in the following experiments was made up of a subset of these elements. Using a simple (x,y) pair for clarity this becomes.

$$F=[x,y,s(x),s(y),s(dx),s(dy),s(|d^2x|),s(|d^2y|)] \quad (1)$$

where the function s(.) indicates a time smoothed average of the quantity, and the first and second order differences, dx and d^2 x are given by.

$$dx = x_t - x_{t-1}, and \quad (2)$$
$$d^2x = x_t - 2x_{t-1} + x_{t-2}$$

The smoothing function s(.) implements a moving average window defined as.

$$s_t(x) = (\upsilon)(s_{t-1}(x)) + (1-\upsilon)(x), \qquad (3)$$

The important point here being the use of a "short term memory" to store the recent position of the first and second order information regarding the status of the power output. This data together with the current data is used as input to a Kohonen Self-organising Feature Map (SOFM) [9]. For comparison purposes the data set used contains all of the input vectors as used in the Kohonen networks as described in [10].

2 Method

For training and testing purposes, the data provided comes from sensor readings from a gas turbine. Each sensor produces a pattern containing 240 rows of data. Each sensor reading was combined into a suitable training set, the target being the level of electrical output produced by the turbine. In the case of the Kohonen networks, this is not required and all of the data is incorporated into the input phase. Because of the size of the input feature space several dimensionality reduction techniques were applied to the data, to ensure non-redundant data was not incorporated into the models. These issues do not apply to Elman and Hopfield networks. For the Elman and Hopfield networks the input patterns were normalised to unit length, to ensure they fell within the required range of 0-1. Each network was trained on the profile of a normal event data set, split into a training, verification and test set, which allows the accuracy of the network to be predicted Subsequently the networks were given several pattern sets, to determine its ability to discriminate between normal and abnormal data readings.

3 Results

The results of our experiments are contained in tables 1 and 2 which show the results for the time shifted Kohonen networks and the Elman,, Hopfield and Time delayed networks respectively.

Number of Inputs	Percentage of Patterns classified correctly
41	100%
42	100%
42	100%
38	100%
38	98%
28	99%
13	100%
13	100%
13	95%

Table 1: Performance of Kohonen networks using time shifted inputs

Network type	Percentage of Patterns Correctly Classified
Elman	100%
Hopfield	25%
TDNN's	83.3%

Table 2: Results of Elman, Hopfield and Time delay neural networks (TDNN's).

The examples demonstrate that certain architectures are better at recognising novel patterns than others. The Hopfield networks are capable of discriminating between normal and extremely obvious error patterns, but have more difficulty in discriminating between normal patterns and other sets containing only a small percentage of error patterns. The Elman networks, have shown excellent discriminatory ability, recognising normal patterns as well as discriminating effectively between different type of error patterns. The self-organising feature maps with time delayed inputs also show good classification performance. The time delayed networks were able to discriminate between error and normal patterns, but experienced similar problems to the Hopfield networks, in failing to recognise errors in pattern sets which were substantially normal, however their performance is far better than for the Hopfield network.

4 Conclusions

We have experimented with neural network architectures capable of storing and recalling previously learned patterns, which include a "memory" facility. We have compared their performance to neural networks, which use some form of time delayed inputs, or time smoothed data to achieve the same effect. Our experiments suggest that architectures relying solely on associative learning are incapable of discriminating between patterns other than those which are very diverse from the original pattern on which they have been trained. Because of this Hopfield networks appear unsuitable for condition monitoring purposes owing to their eagerness to fall into a state space from which it is impossible to dislodge them.

Elman networks appear more promising, as the combination of a context layer together with a method of adjusting the difference between required and actual results gives the system the required "jolt" to escape from minima's. The use of time delayed inputs to neural networks such as multi-layer perceptrons and Kohonen networks, suggest that good classification performance can be achieved. However the verification error between the training and test sets is higher than may be expected, whilst the two stage time delay networks do not recognise all of the subsequent patterns with which they are presented. Our results suggest that the Kohonen network with time shifted inputs and Elman networks provide comparable performance to the architectures used in [10]. However the increase in dimensionality incurred by using a larger input set for the time shifted Kohonens may pose more sever problems for larger data sets than that used in this study.

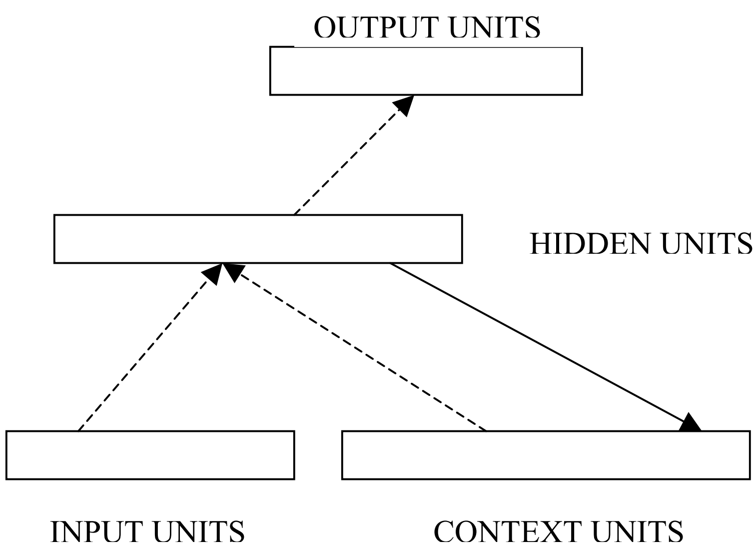

Figure 1: Simple recurrent network, showing connections between hidden and context layer

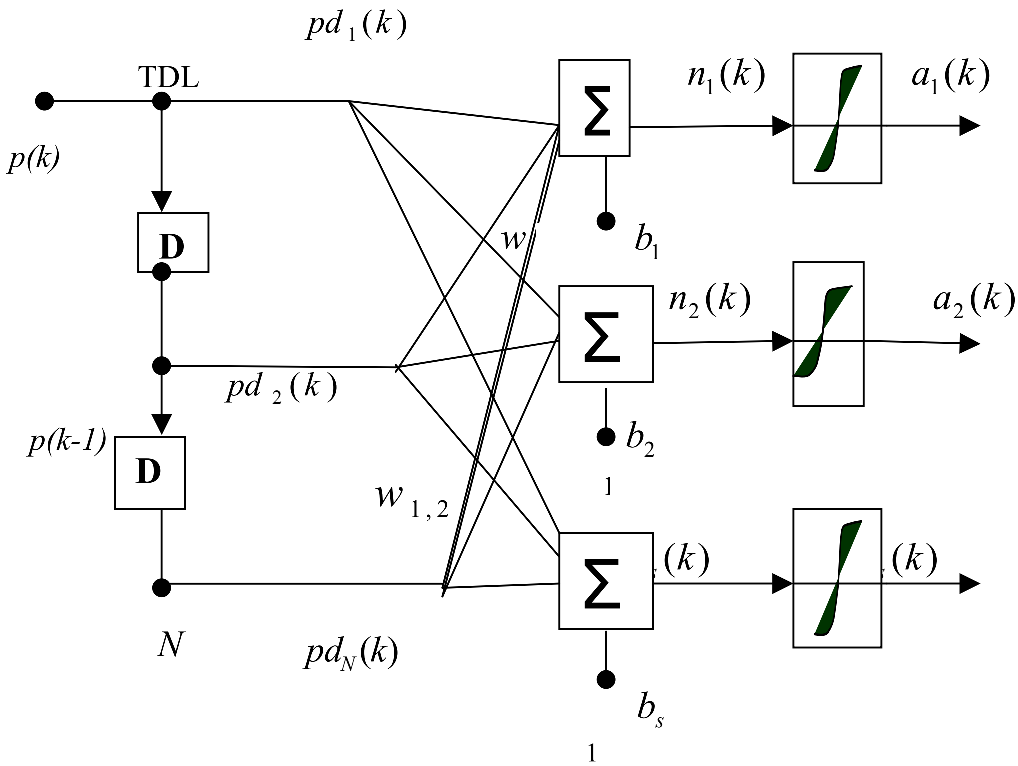

Figure 2: tapped delay line as input to MLP

References

[1] Haykin, S.; 1995 Neural Networks - A Comprehensive Foundation; Maxwell Macmillan International Publishing Company, 138.

[2] Lowe, D.; 1995 Radial Basis Function Networks, The Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press

[3] Li, J., A.N. Michel and W. Porod. "Analysis and Synthesis of a class of neural networks: linear systems operating on a closed hypercube". *IEEE transactions on Circuits and Systems,* vol. 36, no 11, November 1989, pp 1405-1422.

[4] Elman L J., "Finding structure in time" *Cognitive Science,* vol. 14, 1990, pp.179-211

[5] Waibel A, Hanazawa T, Hinton G, Shikano K, and Lang K.. Phenome recognition using time-delay neural networks. *IEEE Transactions on Accoustics, Speech, and Signal Processing,* 37:328-339,1989

[6] Bishop C, *Neural networks for pattern recognition,* Oxford University Press, 116-161, 1997

[7] Johnson N and Hogg D. Learning the Distribution of Object Trajectories for Event Recognition, *Proc, BMVC*, vol 2, 1995

[8] Owens, J and Hunter A. *Application of the Self-Organising Map to trajectory Classification* Proceedings of the Third IEEE International Workshop on Visual Surveillance, 2000

[9] Kohonen T. *Self-Organising Maps.* Springer-Verlag

[10] J F Dale Addison, Stefan Wermter, John MacIntyre "*Effectiveness of feature extraction in neural network architectures for novelty detection"*, ICANN-99, Ninth International Conference on Artificial Neural Networks, Edinburgh, UK, September 1999, pp 976-981