Traffic Trends Analysis using Neural Networks

T. Edwards, D.S.W. Tansley, R. J. Frank, N. Davey

`T.J.Edwards@nortel.co.uk, D.S.W.Tansley@nortel.co.uk`
Northern Telecom (Nortel) Limited
Harlow, Essex, UK, CM17 9NA

`N.Davey@herts.ac.uk, R.J.Frank@herts.ac.uk`
Faculty of Information Sciences
University of Hertfordshire
Hatfield, Herts., UK, AL10 9AB

Abstract

An application of time series prediction, to traffic forecasting in ATM networks, using neural nets is described. One key issue, the number of data points needed to be included in the input representation to the net is discussed from a theoretical point of view, and the results are applied in the model under discussion. Experimental results are discussed and analysed.

1 Introduction

To be successful in a competitive world, any organisation must not only maintain a good understanding of the present, but be able to forecast the future – as accurately and precisely as possible. Analysis of historical trends in data, with a view to making predictions is therefore a key task, and one for which neural network technology is well suited. At Northern Telecom (Nortel) Limited we have investigated two applications of predictive neural networks:

- The early identification of problems in Asynchronous Transfer Mode (ATM) networks. ATM networks produce a continually varying and often heavy stream of alarms and other symptomatic information. We are investigating the use of Artificial Neural Networks (ANNs) to identify when a sequence of events is indicative of an incipient, major component failure.

- The prediction of traffic on ATM networks. This is a standard time series prediction task and it is discussion of this problem that forms the main subject of this paper.

Analogous work is reported widely. For example, there are also a host of applications using ANN predictors to forecast aspects of the financial markets, see for example [5].

This area is also closely related to the similar problem of detecting anomalous patterns in time series, as used in our work on the detection of fraudulent use of cellular phones [2].

Section 2 briefly discusses times series prediction, Section 3 the method used to set the window size, and section 4 gives summary results.

2 Time Series Prediction

One aspect of time series analysis involves forecasting the value of a variable from a series of observations of that variable up to that time.

We suppose that observations are available at discrete, equispaced historical time intervals, $z[T]$, $z[T-\tau]$, $z[T-2\tau]$, $z[T-3\tau]$ …, with time interval τ. The observations $\{z(T), ... z[t - (N-1)\tau]\}$ constitute a "window" and N is referred to as the window size. The aim is to forecast the value of z at some later time, $z_f(T+L)$, where L is the lead-time and is an integral multiple of τ. More formally the objective is to obtain a forecast function $z_f(T+L)$ which minimises the mean square of the deviations $z_f(T+L) - z(T+L)$ for each lead time L.

2.1 Time Series Prediction using Neural Nets

The simplest way to estimate z_f using a neural net is to use a feedforward net with one input for each member of the window and one output for $z_f(N+ \tau)$[5]. An obvious extension of this model is to have multiple outputs corresponding to multiple lead times. These models are often called sliding window nets. As discussed in [3] a more sophisticated predictor may be produced by buffering either the hidden units or the output units and recurrently adding these activations to the input vector, as in respectively Elman or Jordan type nets. These predictors are particularly useful when the data is inherently noisy [3]. In the work reported here we have used a single hidden layer feedforward net with a sliding window input, trained with a scaled conjugate gradient algorithm.

A sliding window feed forward neural network is shown below using a window size of 4.

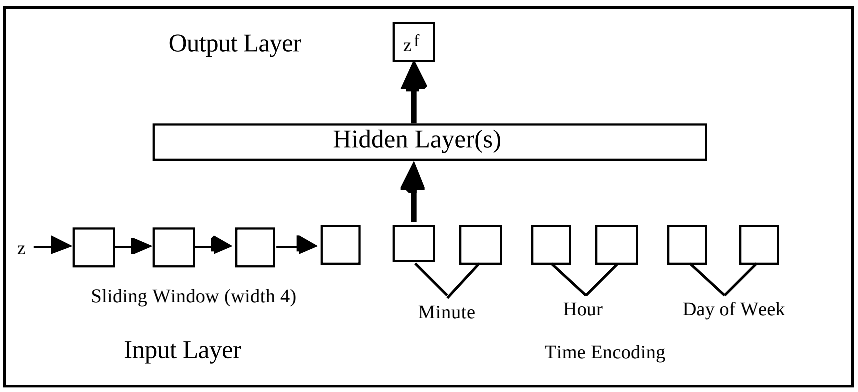

Figure 1: Neural network inputs and outputs

One of the characteristics of telecoms traffic is the superimposing of many cyclical effects. For instance, there are hourly trends corresponding to the business day, daily trends according to which day of the week (some working days are typically busier than others and weekends have very little traffic) trends according to the day of the month (end of month can be busier) and seasonal trends. Each of these trends are cyclical with differing periodicities. One approach to this problem is to de-trend the data by working out what these periodicities are, and what is the average effect from each of these influences. The trend(s) are then removed and prediction made on the resulting data. The obvious simplification of this problem is that we can encode these time factors in the ANN and allow the model to determine how much effect each of these factors has on the prediction. The model can also deal with the interactions between effects.

The tool we have developed has been specifically designed to allow the addition of time inputs which are appropriate to the application. Each of the time representations are encoded using a standard sin/cos pair [6]. The addition of the time units makes the whole prediction task simpler and the tool more suitable for embedding into a host system, such as a network management system. In some circumstances the addition of an ancillary input variable is necessary to make the prediction more accurate where the variable being predicted is dependent on the ancillary input variable.

3 Window Size

A central problem in using this approach to forecasting is to ascertain an appropriate window size. Whilst this can be approached from a purely empirical point of view, as is often done, results from the theoretical analysis of non-linear dynamic systems [1] can be applied to give a more well motivated choice. A window size that is too small will not allow the model to properly represent the system dynamics, and will therefore lead to a poor generalisation performance. Conversely, if the window is too large then the computational complexity of the network is unnecessarily increased. Moreover the network may also learn to represent both the signal and noise in the data [1] which will again lead to poor generalisation. Below we present an algorithm to determine the correct window size.

3.1 The Algorithm

The algorithm is based on analysing consecutive values of the input data to determine the correct window size. It works by taking one dimensional samples, z(T) and combines sequential values together to form a multidimensional vector **s** of dimension d.

For example, for dimension d = 2, the vectors S = {**s**(0), **s**(1), … }, can be formed from the sequential values as follows:

$\mathbf{s}(0) = [z(0), z(1)]$

$\mathbf{s}(n) = [z(n), z(n+1)]$

$\mathbf{s}(N-1) = [z(N-1), z(N)]$

The theoretical results imply that with a sufficiently large value of d, the path of these vectors in $\mathbf{R}^d$, is representative of the dynamics of the system, of which z is the observed variable. The goal is to find the smallest value of d that has this property. Following the method of Kennel [4] we use the *nearest neighbour heuristic* to ascertain a minimal value for d. The idea is that for each of the **s**(n) its nearest neighbour in S is found and the distance between the vectors recorded, as *NearestNeighbourDistance(n,d).* This distance is then recalculated for **s**(n) and its nearest neighbour but now with an incremented window size to give: *NearestNeighbourDistance(n,d+1)*. If the difference between these two values is large in proportion to the original separation then they are judged as false nearest neighbours. Formally, when:

$$\frac{\left| NearestNeighbourDistance(n,d) - NearestNeighbourDistance(n,d+1) \right|}{NearestNeighbourDistance(n,d)} > R$$

s(n) is judged to have a false nearest neighbour. A suitable value for the threshold R lies in the range 10 to 50 [1]; we use a value of 10.

To find the appropriate window size then, the number of false nearest neighbours for the whole training set is computed for incrementally increasing window sizes. When the number approaches zero the window size is fixed. At this point the dynamics of the system is represented with reasonable fidelity.

4 Traffic Prediction Example

4.1 Results

The experimental set up was to present patterns of voice traffic demand to the ANN (as previously described). This ANN was linked to a host network management system. In this experiment, data of 1339 time series points representing network telephony traffic was used. As can be seen (graph below) the number of false neighbours declines dramatically reaching 11 by window size 4. After this the graph varies little. A window size of 21 reduces the number of false neighbours to 5 and a window size of 42 reaches 4.

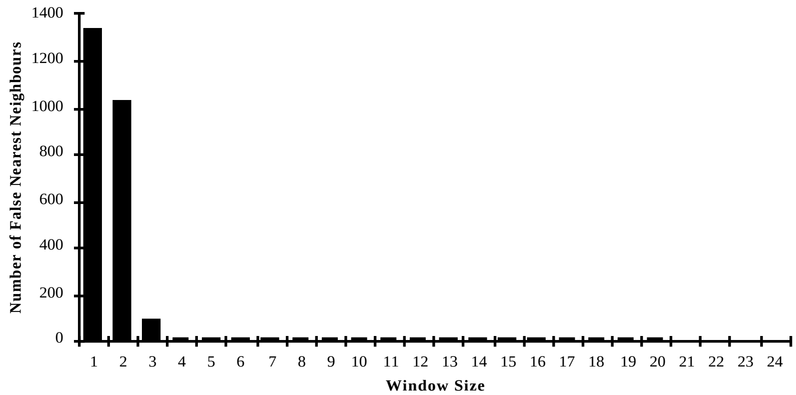

Figure 2: Graph of number of false nearest neighbours

It was therefore shown that a window size of 4 should be used. With this representation the network performed well as a predictor for the training set and moreover produced good generalisation behaviour on unseen data.

The diagram below shows the generalisation performance of the trained neural net when the amount of traffic increases. Although the ANN demonstrates good generalisation properties (seen before the retrain) the traffic has increased to a level significantly above the usual traffic patterns. The host network management system is constantly monitoring a window of actual and corresponding predicted values. When the difference between predictions and actual values falls below a user defined threshold then the network management system recognises that the performance is sub-optimal and requests a retrain. A copy of the net is then retrained on the most recent window of data and if predictive performance is improved then the current net is switched with the newly retrained net and the old net deleted. Retraining is sometimes postponed until processor time on the host machine becomes available as this is a low priority task.

The net can be retrained in under 2 minutes on a HP9000/700 series workstation (typical of a modern network management system). This can be done easily between successive predictions (made at 30 minute intervals).

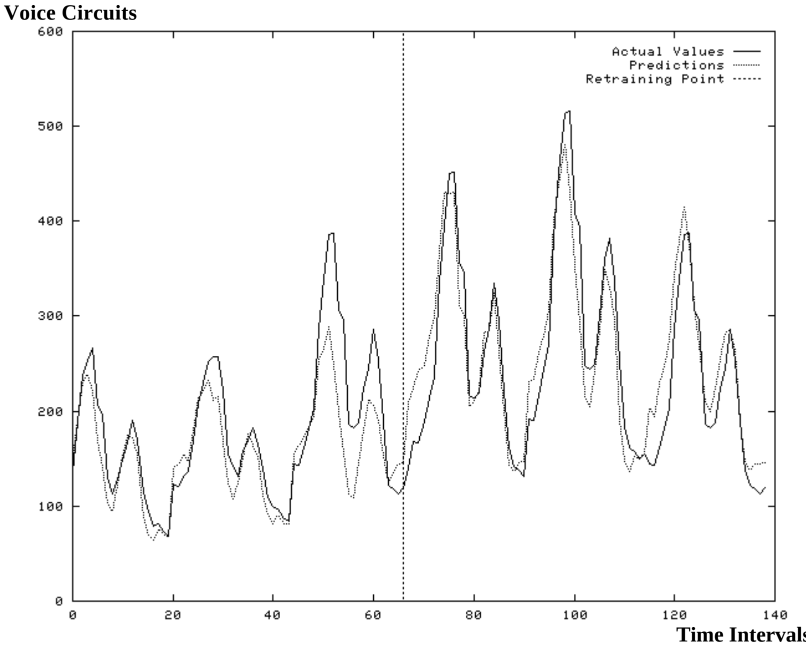

Figure 3: Performance of the final model

5 Conclusion

Neural networks provide a useful tool for time series prediction in the telecoms domain. Critical to the performance of the predictor is the selection of an appropriate window size for the data we need to model. The nearest neighbour algorithm, described earlier, has been tested empirically and shown to be a valuable technique for allowing us to define this window size from analysis of the data.

6 Acknowledgements

The authors of this paper would like to thank Jon Coward and Peter Hamer for their invaluable contribution to the implementation of this work, Phil Hobson for managing this project and Simon Field and Paul Barson for their contributions to engineering the core ANN.

References

[1] H. D. I. Abarbarnal, R. Brown, J. L. Sidorowich and L. S. Tsimring, "The analysis of observed chaotic data in physical systems," *Reviews of Modern Physics*, Vol. 65, No. 4 pp1331-1392, 1993.

[2] P. Barson, N. Davey, S. D. H. Field, R. J. Frank, G. McAskie, "The Detection of Fraud in Mobile Phone Networks," *Neural Network World* Vol 6, No 4., 1996.

[3] G. Dorffner, "Neural Networks for Time Series Processing," *Neural Network World* 4/96, 447-468, 1996.

[4] M. B. Kennel, R. Brown, H. D. Ababarnal 1192, *Physics Review*, A 45, 3403.

[5] D. W. Patterson, K. H. Chan, C. M. Tan, "Time Series Forecasting with neural nets: a comparative study," *Proc. the international conference on neural network applictions to signal processing*. NNASP 1993 Singapore pp 269-274.

[6] Department of Trade and Industry, *Best Practice Guidelines for Developing Neural Computing Applications*, pp86-87, 1994.