Time Series Prediction and Neural Networks

N.Davey, S.P.Hunt, R.J.Frank,
University of Hertfordshire
Hatfield, UK.
Email: {N.Davey, S.P.Hunt, R.J.Frank}@herts.ac.uk

Abstract

Neural Network approaches to time series prediction are briefly discussed, and the need to specify an appropriately sized input window identified. Relevant theoretical results from dynamic systems theory are introduced, and the number of false neighbours heuristic is described, as a means of finding the correct embedding dimension, and thence window size. The method is applied to three time series and the resulting generalisation performance of the trained feed-forward neural network predictors is analysed. It is shown that the heuristics can provide useful information in defining the appropriate network architecture.

I. INTRODUCTION

Neural Networks have been widely used as time series forecasters: most often these are feed-forward networks which employ a sliding window over the input sequence. Typical examples of this approach are market predictions, meteorological and network traffic forecasting. [1,2,3]. Two important issues must be addressed in such systems: the frequency with which data should be sampled, and the number of data points which should be used in the input representation. In most applications these issues are settled empirically, but results from work in complex dynamic systems suggest helpful heuristics. The work reported here is concerned with investigating the impact of using these heuristics. We attempt to answer the question: can the performance of sliding window feed-forward neural network predictors be optimised using theoretically motivated heuristics? We report experiments using three data sets: the sequence obtained from one of the three dimensions of the Lorenz attractor, a series of 1500 tree ring measurements, and measurements of the traffic load on an ATM network.

II. TIME SERIES PREDICTION

A time series is a sequence of vectors, $\boldsymbol{x}(t)$, t = 0,1,… , where t represents elapsed time.
For simplicity we will consider here only sequences of scalars, although the techniques considered generalise readily to vector series. Theoretically, x may be a value which varies continuously with t, such as a temperature. In practice, for any given physical system, x will be sampled to give a series of discrete data points, equally spaced in time. The rate at which samples are taken dictates the maximum resolution of the model; however, it is not always the case that the model with the highest resolution has the best predictive power, so that superior results may be obtained by employing only every nth point in the series. Further discussion of this issue, the choice of time lag, is delayed until section 3, and for the time being we assume that every data point collected will be used.

Work in neural networks has concentrated on forecasting future developments of the time series from values of x up to the current time. Formally this can be stated as: find a function $f: \quad ^{N} \rightarrow \quad$ such as to obtain an estimate of x at time t + d, from the N time steps back from time t, so that:

$$x(t+d) = f(x(t), x(t-1), \ldots, x(t-N+1))$$

$x(t+d) = f(\mathbf{y}(t))$ where $\mathbf{y}(t)$ is the N- ary vector of lagged x values
Normally d will be one, so that f will be forecasting the next value of x.

Neural Network Predictors

The standard neural network method of performing time series prediction is to induce the function f in a standard MLP or an RBF architecture, using a set of N-tuples as inputs and a single output as the target value of the network. This method is often called the sliding window technique as the N-tuple input slides over the full training set. Figure 1 gives the basic architecture.

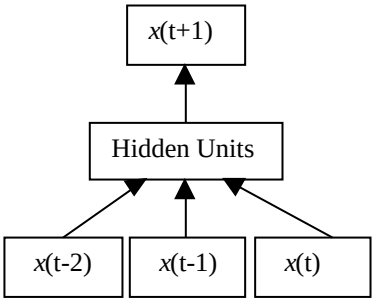

Figure 1: The standard method of performing time series prediction using a sliding window of, in this case, three time steps.

As noted in [4] this technique can be seen as an extension of auto-regressive time series modelling, in which the function f is assumed to be a linear combination of a fixed number of previous series values. Such a restriction does not apply with the MLP approach as MLPs are general function approximators.

III. THEORETICAL CONSIDERATIONS

Time series are generally sequences of measurements of one or more visible variables of an underlying dynamic system, whose state changes with time as a function of its current state vector $\mathbf{u}(t)$:

$$\frac{d\mathbf{u}(t)}{dt} = G(\mathbf{u}(t))$$

For the discrete case, the next value of the state is a function of the current state: $\mathbf{u}(t+1) = F(\mathbf{u}(t))$

Such dynamic systems may evolve over time to an attracting set of points that is regular and of simple shape; any time series derived from such a system would also have a smooth and regular appearance. However another result is possible: the system may evolve to a chaotic attractor. Here, the path of the state vector through the attractor is non-periodic and because of this any time series derived from it will have a complex appearance and behaviour.

In a real-world system such as a stock market, the nature and structure of the state space is obscure; so that the actual variables that contribute to the state vector are unknown or debatable. The task for a time series predictor can therefore be rephrased: given measurements of one component of the state vector of a dynamic system is it possible to reconstruct the (possibly) chaotic dynamics of the phase space and thereby predict the evolution of the measured variable? Surprisingly the answer to this is yes. The *embedding theorem* of Mañé & Takens [5] shows that the space of time lagged vectors $\mathbf{y}$ with sufficiently large dimension will capture the structure of the original phase space. More specifically they show that if N, the arity of $\mathbf{y}$, is at least twice the dimension of the original attractor, "then the attractor as seen in the space of lagged co-ordinates will be smoothly related" [5] to the phase space attractor. Of

course this does not give a value for N, since the original attractor dimension is unknown, but it does show that a sufficiently large window will allow full representation of the system dynamics. Abarbanel et al. [5] suggest heuristics for determining the appropriate embedding size and time lag, and these are discussed below.

False Nearest Neighbours

Having a sufficiently large time delay window is important for a time series predictor - if the window is too small then the attractor of the system is being projected onto a space of insufficient dimension, in which proximity is not a reliable guide to actual proximity on the original attractor. Thus, two similar time delay vectors $\mathbf{y}^1$ and $\mathbf{y}^2$, might represent points in the state space of the system which are actually quite far apart. Moreover, a window of too large a size may also produce problems: since all necessary information is populated in a subset of the window, the remaining fields will represent noise or contamination. In order to find the correct embedding dimension, N, an incremental search, from N = 1, is performed. A set of time lagged vectors $\mathbf{y}_N$, for a given N, is formed. The nearest neighbour relation within the set of $\mathbf{y}_N$'s is then computed. When the correct value of N has been reached, the addition of an extra dimension to the embedding should not cause these nearest neighbours to spring apart. Any pair whose additional separation is of a high relative size is deemed a false nearest neighbour pair. Specifically, if $\mathbf{y}_N$ has nearest neighbour $\tilde{\mathbf{y}}_N$, then the relative additional separation when the embedding dimension is incremented is given by:

$$\left| \frac{d(\mathbf{y}_N, \tilde{\mathbf{y}}_N) - d(\mathbf{y}_{N+1}, \tilde{\mathbf{y}}_{N+1})}{d(\mathbf{y}_N, \tilde{\mathbf{y}}_N)} \right|.$$

When this value exceeds an absolute value (we use 20, following [5]) then $\mathbf{y}_N$ and $\tilde{\mathbf{y}}_N$ are denoted as false nearest neighbours.

Sampling Rate

Since it is easily possible to over-sample a data stream, Ababarnel et al. suggest computing the average mutual information at varying sampling rates, and taking the first minimum as the appropriate rate. See [5] for further details.

IV. EXPERIMENTS

We examine the relationship between embedding dimension and network performance for two data sets.

Lorenz Data

The first data set is derived from the Lorenz system, given by the three differential equations:

$$\frac{dx}{dt} = \sigma(y_t - x_t) \quad \frac{dy}{dt} = -x_t z_t + r(x_t - y_t) \quad \frac{dz}{dt} = x_t y_t - b z_t$$

We take parameter settings r = 45.92, b = 4.0 and σ = 16.0 [5], and use 25,000 x-ordinate points derived from a Runge-Kutta integrator with time step 0.01. Mutual information analysis gives a time lag of 13, so that every thirteenth point is taken. A data set consisting of 1923 points remains and a nearest neighbour analysis was undertaken, with results given in Table 1. This result suggests that an embedding of 3 or 4 should be sufficient to represent the attractor. This corresponds well with the theoretical upper bound of 5, from the embedding theorem.

Embedding Dimension	*Percentage of False Nearest Neighbours*
2	77%
3	3.3%
4	0.3%
5	0.3%

Table 1: The percentage of false nearest neighbours in the Lorenz data set

We train a feedforward neural network with 120 hidden units, using conjugate gradient error minimisation. The embedding dimension, the size of the input layer, is increased from 1 unit to 9 units. The data is split into a training set of 1200 vectors and test set of 715 vectors. Each network configuration is trained 10 times with different random starting points, for 500 epochs. A typical run is shown in Figure 2 and the overall results in Figure 3. For this data set over-training was not a problem – the test set error was not seen to rise, with test and training set generally producing similar mean square error (MSE) values. The final MSE values as seen in Figure 2, show that an embedding of 3 produces a significant decrease in MSE, when compared with 1 or 2. Further increases in the embedding size produce decreases in the test set error (with the exception of 6 input units), but the decreases are far smaller.

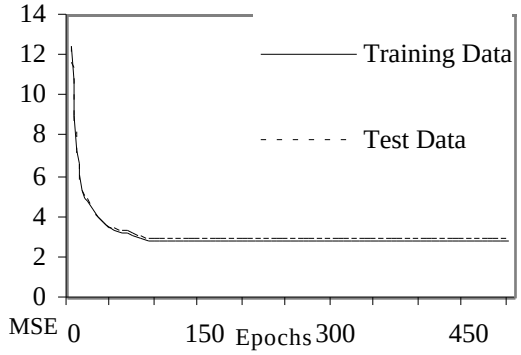

Figure 2 The mean square error for the Lorenz data with an input window of 5 units.

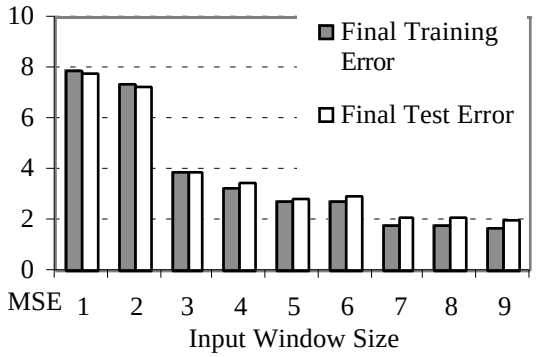

Figure 3 The final MSE for the test and training set of the Lorenz data, as the embedding dimension is varied. All results are averages of 10 runs, at 500 epochs of conjugate gradient training.

Figure 4 shows the distribution of errors over the test set for typical trained networks. There is an obvious transition in the error pattern between 2 and 3 inputs, after which there is no significant change in the pattern of errors. In this case the correspondence with the predicted embedding dimension from the false nearest neighbours analysis is noteworthy. The lack of any noise in the Lorenz data set may account for the gentle decrease in error past the optimal predicted embedding dimension

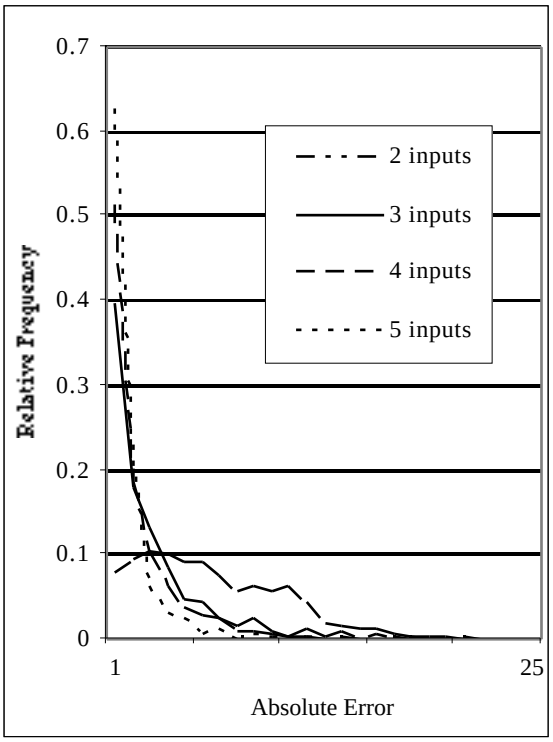

Figure 4: The relative frequency of errors for the Lorenz data test set, and a variety of networks configurations.

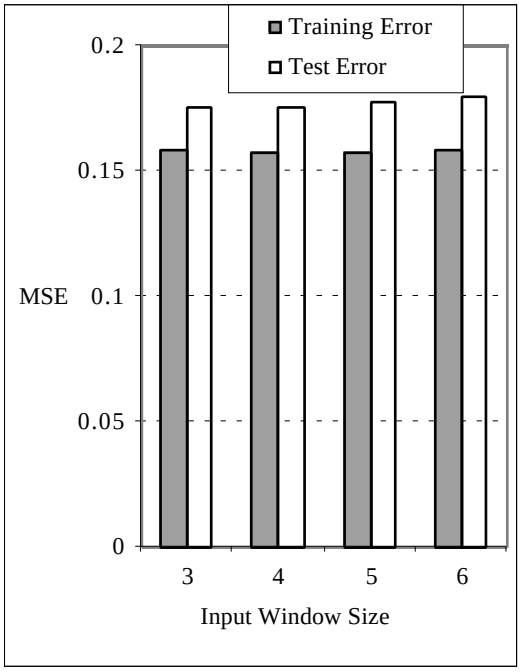

Figure 5: The final MSE for the test and training set of the Tree Ring data, as the embedding dimension is varied. All results are averages of 5 runs, at 200 epochs of conjugate gradient training

Voice traffic demand, over an ATM network

In this experiment, data of 1339 time series points representing network telephony traffic was used [6]. The false nearest neighbour analysis gave the result shown in Table 2. In this case a window of size four was chosen.

One of the characteristics of telecomms 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. To help the forecaster deal with this the day, hour and minute were added to the input using a periodic code, wit two bits for each feature, so that six additional input units are used, to give ten input units in total.

Tree Ring Data

This data set consists of 5405 data points recording tree ring data (measuring annual growth) at Campito mountain from 3435BC to 1969AD, [7]. The mutual information analysis of the data suggested a sampling rate of one. The subsequent false nearest neighbour analysis is summarised in Table 3.

Embedding Dimension	*Percentage of False Nearest Neighbours*
1	100%
2	82.1%
3	7.1%
4	0.8%
5	0.7%

Table 2: The percentage of false nearest neighbours in the ATM data set

Embedding Dimension	*Percentage of False Nearest Neighbours*
1	100%
2	14.0%
3	68.9%
4	16.3%
5	0.0%

Table 3: The percentage of false nearest neighbours in the Tree Ring data set

The results show a strange pattern in which the number of false neighbours falls and then rises again. The data was split into two: a training set, of 3000 points and a test set of 2400 points. Figure 5 shows the MSE for various embedding dimensions after conjugate gradient training, of feedforward nets with 120 hidden units, averaged over 5 runs. The results show no significant variation of error as the input window size is varied. This data set does not appear to be amenable to prediction by this form of model.

V. CONCLUSIONS

The results suggest that the embedding theorem and the false nearest neighbour method can provide useful heuristics for use in the design of neural networks for time series prediction. With two of the data sets examined here, the predicted embedding size corresponded with a network configuration that performed well, with economical resources. With our data sets we did not observe an overlarge embedding size to have a deleterious effect on the network. The tree ring data, however, showed that conclusions must be treated with caution, since poor predictive results were produced whatever the window size.

References

[1] Edwards, T., Tansley, D, S. W., Davey, N., Frank, R. J.(1997). Traffic Trends Analysis using Neural Networks. *Proceedings of the International Workshop on Applications of Neural Networks to Telecommuncations 3*. pp. 157-164.

[2] Patterson D W, Chan K H, Tan C M. 1993, 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.

[3] Bengio, S., Fessant F., Collobert D. A Connectionist System for Medium-Term Horizon Time Series Prediction. *In Proc. Intl. Workshop Application Neural Networks to Telecoms* pp308-315, 1995.

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

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

[6] Edwards, T., Tansley, D, S. W., Davey, N., Frank, R. J.(1997) Traffic Trends Analysis using Neural Networks. Proceedings of the International Workshop on Applications of Neural Networks to Telecommuncations 3. pp. 157-164

[7] Fritts, H.C. et al. (1971) Multivariate techniques for specifying tree-growth and climatic relationships and for reconstructing anomalies in Paleoclimate. *Journal of Applied Meteorology*, 10, pp.845-864.