Unsupervised Learning for Financial Forecasting

Corchado J., Fyfe C. and Lees B.

Applied Computational Intelligence Group
University of Paisley, High St., Paisley, PA1 2BE,
United Kingdom

Tel.: +44 (0)141 848 3327
Fax: +44 (0)141 848 3542
email: corc-ci0@wpmail.paisley.ac.uk

Abstract

An unsupervised neural based approach to financial forecasting is presented in this paper; its performance is compared with that from a statistical technique and two other standard neural network techniques. We show that the unsupervised network outperforms multilayer perceptrons, radial basis function network and a standard ARIMA model.

1. Introduction

Every system, even the most complicated ones, such as the financial market, can be modelled if its behaviour is fully known and understood. However the current knowledge of this environment is not strong enough such as to create a model of its behaviour. Many researchers (Tailor S., 1986; Mill, T. C., 1993; Moody J., 1995) have investigated in this area and obtain successful results.

The goal of the experiments presented here is to create an Artificial Neural Network (ANN) capable of modelling a Financial Time Series and of forecasting its future values. This paper presents a novel Negative Feedback Artificial Neural Network for forecasting and extraction of the embedding function of time series. This network has previously been shown to be successful on standard iterated mappings (Fyfe C., 1997) and in complex oceanographic data sets (Corchado J., 1998).

This method is compared with the Autoregressive Integrated Moving Averages (ARIMA) model, with a standard Multilayer Perceptron Network (MLP) and with Radial Basis Function Networks (RBF), all results based on forecasting of Dow-Jones index from 30th of December 1988 to 15th of May 1985 (Figure 1).

Time series prediction is based on the assumption that an observable feature of a system is determined by an underlying deterministic system. If the evolution of the system can be described by a set of n ordinary differential equations in n variables, there exists a unique trajectory through every point **a** in R^n. In order to make a prediction it is necesary to know both the underlying rules of the deterministic system and the current state of the system. For example, consider a system in which we can observe a single scalar quantity, x_t, which is the value of x at time t and which is determined by the state, $\mathbf{a}_t$, of the system at time t. Now if the underlying system is based on the n dimensional set of differential equations referred to above, then the state, $\mathbf{a}_{t+1}$, of the system at time t+1 can be described by the set of equations $\mathbf{a}_{t+1}=F(\mathbf{a}_t)$, while the observable at time t+1 is given by $x_{t+1} = g(\mathbf{a}_{t+1}) = g(F(\mathbf{a}_t))$. It can be shown that there exists a value d such that the vector $\mathbf{x} = (x_1,x_2,\ldots,x_d)^T$, consisting of d consecutive observations of x, fully characterises the system. i.e. the evolution of the system may be specified absolutely using the function F() or equally by using the function H() where $x_{t+1} =H(\mathbf{x}_t) =H(x_t,x_{t-1},\ldots,x_{t-d+1})$. Therefore merely by finding a sufficiently long set of consecutive observations of x (d is known as the embedding dimension of the system) a complete specification of the future values or observations of the system may be obtained. It is well known, of course, that, for non-linear systems, the underlying dynamics are often such that there will be divergence of

trajectories from nearby initial conditions. Thus, since it is not possible to measure observables to infinite precision, such systems may only be predictable to a finite length into the future.

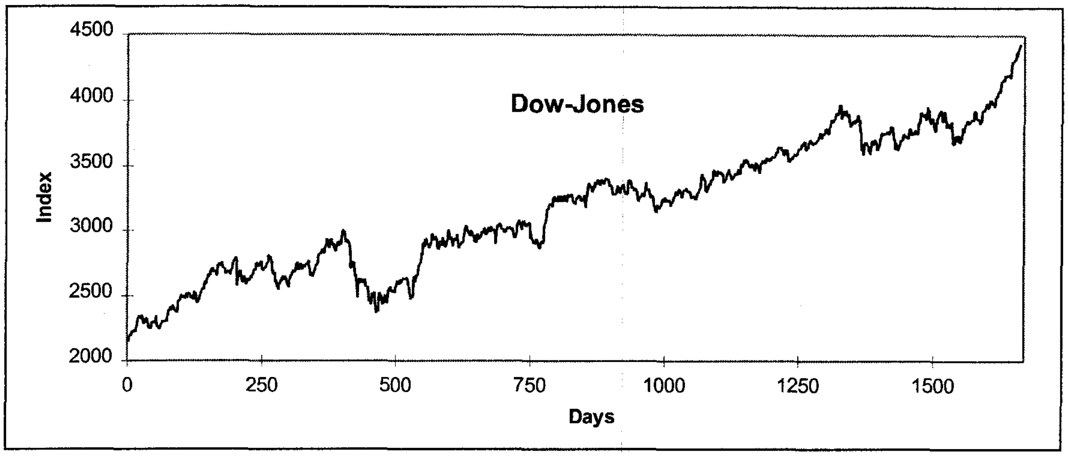

Figure 1: Dow-Jones Index Financial Data, from 30th December 1988 to 15th May 1985.

Time series prediction is based on the assumption that an observable feature of a system is determined by an underlying deterministic system. If the evolution of the system can be described by a set of **n** ordinary differential equations in n variables, there exists a unique trajectory through every point **a** in R^n. In order to make a prediction it is necesary to know both the underlying rules of the deterministic system and the current state of the system. For example, consider a system in which we can observe a single scalar quantity, x_t, which is the value of x at time t and which is determined by the state, $\mathbf{a}_t$, of the system at time t. Now if the underlying system is based on the n dimensional set of differential equations referred to above, then the state, $\mathbf{a}_{t+1}$, of the system at time t+1 can be described by the set of equations $\mathbf{a}_{t+1}=F(\mathbf{a}_t)$, while the observable at time t+1 is given by $x_{t+1} = g(\mathbf{a}_{t+1}) = g(F(\mathbf{a}_t))$. It can be shown that there exists a value d such that the vector $\mathbf{x} = (x_1,x_2,\ldots,x_d)^T$, consisting of d consecutive observations of x, fully characterises the system. i.e. the evolution of the system may be specified absolutely using the function F() or equally by using the function H() where $x_{t+1} =H(\mathbf{x}_t) =H(x_t,x_{t-1},\ldots,x_{t-d+1})$. Therefore merely by finding a sufficiently long set of consecutive observations of x (d is known as the embedding dimension of the system) a complete specification of the future values or observations of the system may be obtained. It is well known, of course, that, for non-linear systems, the underlying dynamics are often such that there will be divergence of trajectories from nearby initial conditions. Thus, since it is not possible to measure observables to infinite precision, such systems may only be predictable to a finite length into the future.

2. The Finite Impulse Response Hebbian Model.

The architecture of the ANN used for time series prediction, in this experiment, is shown in Figure 2. Input data is fed in from the left of the diagram and in successive time steps is passed onto the further neurons as shown. At time t, $y_i(t) = f_i(x_t)$, for $1 \le i \le n$, where n is the number of inputs to the network and the actual values of the inputs are a function of the observed time series.

The feedback lines signify trainable connections; there are no self connections i.e. from a set of inputs to the equivalent y-neuron. The equations governing the activation transfer and learning in the network are given by

$$y_i(t) = x_{i0} - \sum_{j=1, j\neq i}^{m} \sum_{k=1}^{d} w_{ijk} x_{jk}$$

where x_{jk} is the value of the jth input at time (t-k), and w_{ijk} is the weight from the ith neuron to this input. The parameter d measures the length of the embedding dimension and m is the number of neurons in parallel in each embedded layer.

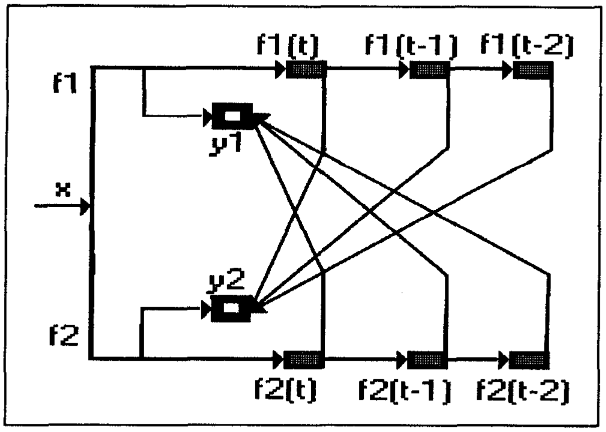

Figure 2: The negative feedback network

Learning is achieved by simple Hebbian learning with momentum:

$$\Delta w_{ijk}(t) = M(\eta y_i x_{jk}) + (1 - M)\Delta w_{ijk}(t-1)$$

where η is a learning rate which may be decreased during the course of the simulation, and M is a parameter which determines the magnitude of the momentum. The negative feedback in the network ensures that the network does not suffer from the usual Hebbian problem of weights growing without bound. Now since the expected value of $\Delta w_{ijk} = 0$ only when y_i and x_{jk} are decorrelated, y_i and x_{j0} are consecutively decorrelated and then y_i and x_{j1} etc. (similarly with respect to the time-delayed values of the other inputs). At convergence an approximation to independence exists between the neuron's outputs and the input data stream. To extract the value of the embedding function, H(), from the network's weights it is required to create a basis of the function space using predefined f() functions. Several experiments have been done using polynomials and sinusoidal functions. The best results have been found using a function of the type:

$$y_1(t) = x_t \text{ and}$$

$$y_n(t) = \sin((n-1) * x_t) \ \forall \ n > 1$$

$$\text{So} \qquad y_1(t) = x_{10} = data(t) \text{ and}$$
$$y_n(t) = x_n = \sin((n-1) * data(t)) \ \forall \ n > 1$$

$$\text{Thus } x_{1k} = data(t-k), \forall k \text{ and } x_{nk} = \sin((n-1) * data(t-k)), \forall k \text{ and } \forall n > 1$$

In this framework, a 500 day training set was used. A window of 8 days has been used to forecast the following day's Dow-Jones Index value. This window has been moved over the whole data set presented in Figure 1 from left to right. The elements of the data series here presented have very high values, between 2000 and 4500 units. To simplify operations, the time series values have been codified in order to scale them down to a value between 0 and 1. With the stated parameters, the average error has been 15.381 units of the Dow-Jones Index. The shape of the error can be seen in Figure 3.

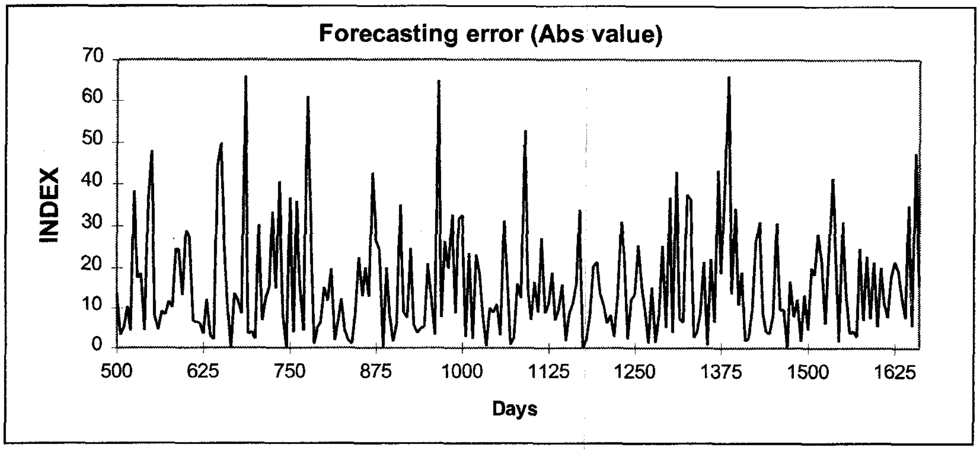

Figure 3: Error using the FIR model.

3. Comparison with other models

To assess, empirically, the performance and accuracy of the method presented in this paper, several experiments have been carried out using standard forecasting methods such us ARIMA, MLP ANN and RBF ANN.

The ARIMA methodology was developed by Box and Jenkins (1976) and it allows the modelling and forecasting of complex data sets. It has gained enormous popularity in many areas and research practice confirms its power and flexibility (Vandaele, 1983).

ARIMA(p,d,q) is composed of two common processes, Autoregressive (p) and Moving Averages (q), which are adequate to model and forecast the time series presented in this paper that consist of elements that are serially dependent and with a clear seasonal structure. The Autoregressive process requires a stationary data set, which can be obtained by differencing (d) the data. ARIMA estimates parameters so that the sum of squared residuals is minimized. The estimation process is performed, in this case, on transformed (differenced) data before the forecasts are generated.

The general trend of the Dow-Jones data set (Figure 1) is incremental: it is obviously a non-stationary data set. An ARIMA(1,1,0) model with one autocorrelations (p=1), of the differentiated data (d=1) and zero seasonal autocorrelations parameter have been found to yield an effective but still parsimonious model of this process. The average error obtain using this model to forecast is 16.97. In this case the forecast was performed on the same data set that was used to obtain the model. Different ARIMA models were tested and performed similarly.

A Multilayer Perceptron using back propagation (Bishop C. R., 1995) as the learning algorithm and a Radial Basis Function ANN (Bishop C. R., 1995) with a dynamic internal structure (so that the centres of the basis functions more accurately modelled the density of the inputs) have also been applied to the data set.

The input to the ANNs is a vector representing the values of the last 10 values of the Dow-Jones Index and the output is the value of next day's Dow-Jones Index value. The ANNs ware trained with 500 values to forecast the following one. In both cases this window was moved one value ahead after the forecast and retrained with the data from the new window and so on. The MLP had a fixed number of neurons in its single hidden layer. The RBF used a fast learning mechanism (Fritzke B, 1994) and the number of basis neurons was defined by the performance of the ANN. This number was always between 20 and 35. The best MLP average error was 16.41 Dow-Jones index units while the best RBF average error was 17.06. The slightly better performance of the MLP than the RBF is unsurprising since the data set is notstationary and so the ANN are extrapolating rather than interpolating, and it is a well known fact that MLP are more adequate for this purpose than RBF.

Conclusions

The forecasting capabilities of the Finite Impulse Response have previously been proven on standard iterated mappings and in several Oceanographic forecasting problems.

Reliable financial forecasting is difficult to achieve. In this paper we have compared the results of the model here described with the results of standard statistical techniques based on ARIMA models and two standard ANN: MLP and RBF. Those results are compared in the following table:

Model	*Type*	*Learning*	*Training**	*Avg. Error*
ARIMA	Stat. Model	None	-	16.97
MLP	ANN	Supervised	30 minutes	16.41
RBF	ANN	Supervised	14 minutes	17.06
FIR	ANN	Unsupervised	2-5 minutes	15.38

** Training is been done in PC Pentium 120, 32 Mb RAM*

The performance of the RBF, the ARIMA and the MLP is relatively similar, nevertheless, between these 3 models and the FIR model there is a bigger gap. The FIR model is also much more faster to train than any of the other two ANN.

References

Bishop C. R. (1995), Neural Networls for pattern recognition, Carendon Press, Oxford.

Corchado J., Fyfe C., Lees B.(1998), Unsupervised Neural Network For Temperature Forecasting, EIS'98, 11-13 February, Tenerife, Spain.

Fritzke B., (1994), Fast learning with Incremental RBF Networks. Neural Processing Letters, Vol 1, No 1,2-5.

Fyfe, C, (1997), An Unsupervised Neural Method for Time Series Analysis, Characterisation and Prediction, Proceedings of ICANNGA 97.

Mills, T. C. (1993), The econometric Modelling of Financial time Series, Cambridge University Press.

Moody, J. (1995), Economic forecasting: Challenger and neural network solutions, in "Proceedings of the International symposium on Artificial Neural Networks, Taiwan".

Tailor s. (1986), Modeling financial Time series, John Wiley & Sons, New York.

Vandaele W. (1983),. Applied Time Series and Box-Jenkins Models, New York: Academin Press.