6 Summary

Whilst not ruling out the possibility of carrying out time series based financial prediction, there are several aspects that must be borne in mind when carrying out research into such an application:

- Ensure that you have sufficient information in your training data to allow the possibility of making the kind of predictions you plan to carry out.
- When performing one step ahead prediction, check that you really are predicting ahead.
- When predicting several steps ahead, start the prediction from many different steps along the time series and check to see that there is no coincidence in your results. An example is a predictor which simply follows the current or overall trend and so appears to be correct for several steps. Only when the predictor consistently forecasts turning points can you claim success.
- Take into account trading costs and frequencies: You may be able to predict one step ahead every minute and accumulate sixty small profits an hour, but if you have to pay 0.05% on each transaction, your sum may actually dwindle to nothing. Similarly it is not possible to trade at such a speed and nor is it so easy to swap from a buying to a selling position at will without accounting for differences in brokers' buy and sell prices.

Above all, neural networks must be viewed as a statistical tool. They are bound by the same restrictions on data information content, data quality and model generalisation as any statistical technique. No statistical technique can extract information where there is none to be had.

References

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, **14**:179–211.

Hoptroff, R. G. (1992). The principles and practice of time series forecasting and business modelling using neural nets. *Neural Computing and Applications*, **1**:59–67.

LeBaron, B. (1993). Non linear diagnostics and simple trading rules for high frequency foreign exchange rates. In Weigend, A. S. and Gershenfeld, N. A., editors, *Time Series Prediction. Forecasting the Future and Understanding the past*, pages 457–474. Addison Wesley.

Pettit, R. R. (1972). Dividend announcements, security performance and capital market efficiency. *Journal of Finance*, **December**:993–1007.

Refenes, A. N. (1991). Constructive learning and its application to currency exchange rate forecasting. In Turban, E. and Trippi, R., editors, *Neural Network Applications in Investment and Finance Services*. Probus Publishing, USA.

Swingler, K. M. (1994). Sequence categorisation using multiple recurrent layers to create many trajectories through network state space. *Proc. ICANN*, **2**:1025–1028.

Waibel, A. (1989). Modular construction of time delay neural networks for speech recognition. *Neural Computation*, **1**:39–46.

Weigend, A. S. and Gershenfeld, N. A. (1993). *Time Series Prediction. Forecasting the Future and Understanding the past*. Addison Wesley.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, **1**:270–280.

and buys again at the predicted low turning point. His profit is gained from the fact that he sold at a high price an bought back at a lower price but we need to know what he payed for the stocks which he sold before we can know what his true profit was. Had he started with a zero holding, his profit before transaction costs would only have been 0.07%. [LeBaron, 1993] points out that it is not realistic to talk about the success or otherwise of a system unless trading costs are taken into account. A trading cost of between 0.1% and 0.01% should be accounted for before any claims can be made. Assuming such a trading cost sees the profit disappear.

4 Methods for Avoiding False Assumptions

Mean squared error (MSE) is not always the best measure of how well a network has learned a time series. You may get MSE down to 0.02 but if the average change from one time step to the next is also 0.02, then you've not achieved a great deal. Such small errors also cause very flat error gradients which require the learning rate to be managed very carefully.

Having obtained a good MSE there are several ways in which one can destructively test a network which appears to be making good predictions. One involves reducing the number of hidden units, or even using a perceptron on the same training task. If you have looked at your data set and decided that N hidden units would produce a good fit, and found that it seems to do so, then a lot can be learned by the deterioration (or lack of it) a network displays when N is reduced. Another way is to test the network on data with a totally unrelated structure; a sine wave for example. If you still seem to be getting good results, then it is clear that there is an error in your interpretation.

5 Market Modelling: Why Neural Networks are Useful After all

One arguement against technical analysis is based on the very premise which makes it seem possible. The technical analysts claim that all market information is reflected in the price levels and so these prices are the only data required. Detractors from this viewpoint argue that whilst the price levels do reflect available information, by the time they do so its too late. [Pettit, 1972] showed that when a new piece of information, an increase in dividend for example, becomes publicly available price change will make one large jump and then flatten out. By the time the information is reflected in the price level, the technical analysts are too late. As the time series prior to this change is said to contain all available information and the dividend announcement is a new fact, it cannot possibly be reflected in the time series. In order to profit from such a situation you must be able to look at a company's performance and its current stock value and know the value has been erroneously set too low.

[Hoptroff, 1992] used a feedforward network to model the performance of Britain's top 100 construction companies revealing a model of profitability and volatility against company size which showed the optimal size of company one should invest in for maximal return. Using such a model it may be possible to spot occasions when prices have been set at the wrong level and so make a profit when they reach equilibrium.

Figure 2: The prediction from Refenes et al (1992)

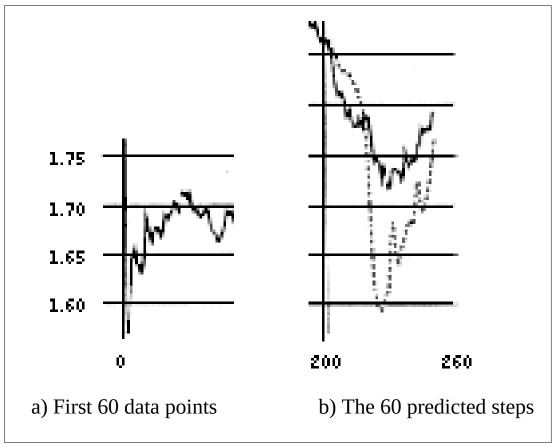


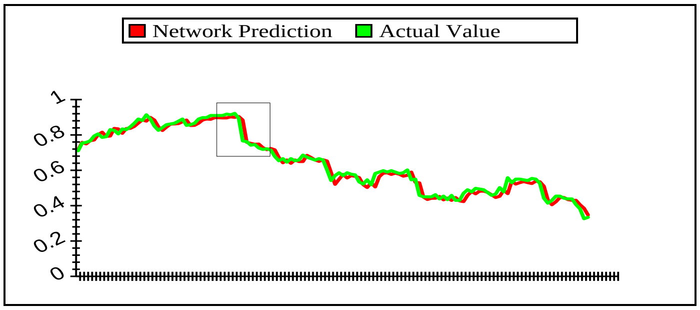
Figure 3: a) shows the first 60 steps in the training sequence, b) shows the 60 predicted steps. Note that once the prediction is near a state close to that of the start of the series, it simply repeats itself.

Refenes notes the importance of correctly predicting major turning points rather than simply predicting the next step ahead. Based on a system of buying and selling in anticipation of predicted turning points, the Refenes network made at least 22% profit on the last 60 trading days of 1989.

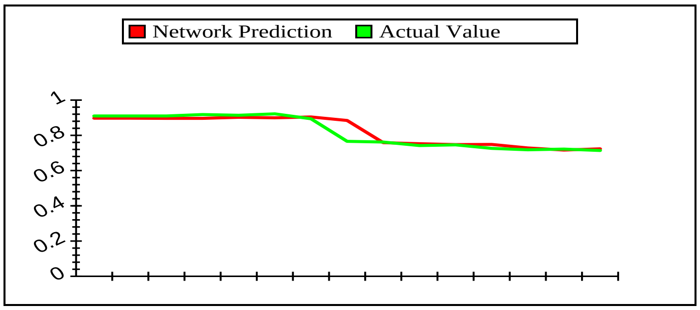
However, if we take a closer look at Refenes' findings a few cautionary tales are revealed. His return of 22% is based on a prediction which takes 200 data points from a non-stationary, noisy, non-periodic time series and predicts the next 60 values. It is always sensible to look at the amount of information you believe that you have extracted from a data set and ask yourself, *Was there sufficient information in the data in the first place or have I performed a conjuring trick?*.

Closer inspection of the results chart in figure 2 and figure 3 shows that the 60 point prediction falls dramatically to a point so low that is is outside the operating space of the model and then picks up following an almost identical path as the data followed from its original (and equally low) starting point. It appears that the match to the actual time series is quite coincidental.

Further, if we examine the figures quoted in the profit breakdown we see that the bulk (15 of the 22%) is made on the first pair of transactions. Refenes sells a holding of his stock at step 200 in the series



a) This one step ahead network prediction looks very good.



b) A closer look at the area in the box in a) shows how the chance of a profit is missed.

Figure 1: Predicting the hourly Dollar / Swiss Franc exchange rates

output of a recurrent neural network predicting one hour ahead and then taking the actual value, rather than the predicted value, as input in order to predict the next step. Looking at the graph and reasoning that *An hour is all you need* may give the impression that superior profits are possible. Clearly this is too good to be true: a closer inspection of the graph—as in figure 1 b)—shows that the network is simply predicting that the price level one hour from now will be the same as it is now. A commonly quoted effect of the efficient markets hypothesis is that such a policy is actually optimal. Figure 1 b) also illustrates the point that the new information is not reflected in the time series until too late: the drop in price drags the prediction down one step later but the price has already flattened out again and the chance of a profit has been lost.

3.2 Many Step ahead Prediction

[Refenes, 1991] used overlapping time windows to convert exchange rate series into spatial vectors. The data – hourly updates of 260 days worth of US dollar/DM exchange rate values – were coded into two moving windows with the intention of mapping the contents of the earlier window of size n onto those of the second, later, window of size m. The two windows move across the data series at a step of s. The task then is to predict the contents of window m from those of n. The authors show how important network design is; choosing the sizes of n, m and s carefully as well as constructively adding hidden units to maximise the data fit. One step ahead prediction was achieved by setting $m = 1$, i.e setting the output vector to be a single value. Multi step prediction was achieved by taking the single valued prediction and feeding it back as input rather than by extending the size of the output window, m.

2 Time Series Prediction

The two main neural network based approaches to time series forecasting are time windowing and recurrent networks. Both attempt to capture the dynamics of the system which underlies the data series by training a neural network to take as input a representation of the current state of the system and to output a prediction of the state of the system at some point in the future. In the case under consideration here, that system is a financial market.

2.1 Time Windowing

Several authors [Waibel, 1989], [Refenes, 1991] have shown how the temporal dimension can be transposed into a spatial vector by taking a moving window over the last n elements in a series. Using a feedforward network with n input units—one for each time step up to time t—and one output unit representing the value of the series at $t + 1$, we can learn to perform one step ahead prediction. By increasing output window size or—more usually—feeding the single output back to the top end of the input window it is possible to extend the predictions to several steps ahead. Predicting an unstable system more than one step ahead however can produce exponentially growing errors. Adding noise to the training set or using a regularisation term during learning can be used to reduce these errors but only if the information is in the data in the first place.

2.2 Recurrent Networks

[Elman, 1990] showed how an otherwise feedforward network with a recurrent context layer which took a copy of the network's hidden layer at time $t-1$ and re-applied it in addition to the input vector at time t was able to learn temporal dependencies. [Williams and Zipser, 1989] showed how a fully recurrent neural network was capable of simulating a Turing machine and so reproduce any deterministic sequence. [Swingler, 1994] showed how extra context layers could be added during training in order to allow a recurrent network to be trained on several examples of a time series from the same source system.

By building a recurrent network with one input unit representing the value of the time series at time t, one output unit representing the value of the time series at time $t+1$ and a recurrent layer to store and re-apply the state of the hidden layer from time t, we can forecast one step ahead along a time series. By taking the network output and feeding it back in as input, this method can be extended to multiple steps forward.

3 Common Pitfalls to Avoid

Having selected a network architecture and carried out rigorous data preparation to establish the limits of what one would expect from a prediction system, one must be aware of certain possible pitfalls awaiting the unsuspecting analyst. The following sections describe a few of these pitfalls in the hope that we can save a few people some time, effort and disappointment.

3.1 One Step Ahead Prediction

Taking the set of Swiss Franc, U.S Dollar exchange rate values from the Santa Fe time series forecasting competition [Weigend and Gershenfeld, 1993] and training a recurrent network on a one step ahead prediction task appears, at first glance, to give the key to a fortune. Figure 1 a) shows a plot of the

Financial Prediction, Some Pointers, Pitfalls, and Common Errors

Kevin Swingler

July 14, 1994

Center for Cognitive and Computational Neuroscience
Stirling University
Stirling
FK9 4LA

Abstract

There is growing interest both in the field of neural computing and in the financial world in the possibility of using neural networks to forecast the future changes in prices of stocks, exchange rates and commodities. Since networks have been shown to be capable of modelling the underlying structure of a time series, many attempts have been made at exploiting that capability in order to carry out a technical analysis of such prices. If the efficient markets hypothesis is true however, there is no underlying structure to be modelled and the whole endevour is doomed to failure. This paper investigates the common methods for such an approach and outlines the major pitfalls and common errors to avoid. It is the author's hope that in pointing out the possible pitfalls now, we can avoid making claims to the commercial world before we are properly ready to do so.

1 Introduction

There is a justifiable scepticism surrounding the idea that it is possible to make money by predicting price changes in a given market based only on its past behaviour and a number of publicly available indicators. Ignoring the cases involving insider information, this scepticism exists for a number of reasons; many of which are explained by the *efficient markets hypothesis*. The hypothesis is based firstly on the assertion that markets follow a random walk which cannot be predicted from past prices. Any chance of potential profits is snapped up immeadiatly, removing the opportunity almost as soon as it is created and certainly before the technical analyst has seen it in the data. It would seem that financial prediction, as far as the technical analyst is concerned, just has no future in it.

There is a possible way around the efficient markets hypothesis because it relies on the *public availability* of market information. If prices do not follow a random walk, but a chaotic one, then anybody who is able to model the price structure and make valid predictions using that model will have access to information which is not publicly available. The efficient markets hypothesis will no longer apply to the person with the model until everybody gains access to the same technology and things even out once more. Given that the efficient markets hypothesis relies on perfect knowledge; perfect prediction technology would only serve to enforce the conditions under which that technology is useless; as soon as a possible profit is predicted, it is snapped up and expected profits return to the level of the risk free return plus a risk premium associated with a stock holding. Put another way, chances of superior profits occur when brokers set their prices incorrectly and investors are able to spot the discrepancy before it is corrected. Perfect prediction for all would remove these discrepancies and with them the opportunities for superior profit.