Predicting Financial Markets with Neural Networks

Leonid Kuvayev
Seminar in Capital Markets
Review Paper

May 28, 1996

1 Introduction

In the past five years, neural networks have received a great deal of attention because of their ability to solve several classes of problems that are difficult, and sometimes impossible, to solve any other way. Neural networks are particularly well suited to finding accurate solutions in an environment characterized by complex, noisy, irrelevant, or partial information. They address these limitations by deriving nonlinear maps between high-dimensional, input pattern spaces and outputs.

A primary advantage of mapping networks over classical statistical regression analysis is that the neural networks have more general functional forms than the well developed statistical methods can effectively deal with. Neural networks are free from depending on linear superposition and orthogonal functions — which linear statistical approaches must use.

In the following section we describe the experiments involving neural networks and accomplished results.

2 Predicting Financial Series

2.1 Futures Prices

In (Grudnitski et al, 1993) the neural networks were used to forecast S & P and Gold Futures prices. As inputs for their neural network they used the following:

1. The monthly growth rate of the aggregate supply of money. It is intended to represent an underlying economic factor that influences both the S&P and Gold markets.

2. The change in and volatility of S&P and Gold futures prices.

3. End-of-month net percentage commitments of large speculators, large hedgers, and small traders.

For forecasting purposes, the networks have one output. This output is the change of the monthly centered price mean for the forecast month. The forecast networks have 24 input nodes. These input nodes represent six inputs per month (i.e. price change, volatility, three sentiment percentages, and money supply) presented four month at a time. The choice of window size is critical in a network's facility to identify high order regularities in a series contaminated by noise, and thus approximate the hidden correlations accurately. The basic assumption is that the sequence of values of fixed size n in the window $I_0 ... I_n$, is somehow related to the following sequence of output values $O_0 ... O_n$, and is defined entirely within the date set.

The forecasting ability of the neural networks is assessed within the framework of a two-step process. The first step in this process is to decide, on a test-period-by-test-period basis, whether to engage in or refrain from trading. The key factor in the decision to trade is the quality of the network's prediction. Quality of the network's prediction is defined as the similarity between a test pattern's input and one or more training input patterns. That is, when a test period's input is found to be different from all of its training input patterns, its corresponding forecast is deemed to be unreliable because that forecast would be produced using parameters derived from dissimilar training patterns.

On the basis of a network assessment of the similarity of the input of a test pattern to one or more of its training pattern inputs, 41 S&P and Gold trades are simulated. The networks are able to predict the correct sign of the next month's price change 75% and 61% of the time for S&P and Gold trades, respectively. When transaction costs and margin requirements are considered, the 41 trades of a S&P and Gold futures contract resulted in an average per-period return on investment of 17.04% and 16.36%, respectively.

This study concludes that the success with which a neural network methodology can be applied to forecast price changes of these markets' futures is related to four factors. Derivation of the parameters of the networks relies heavily on published commitments data as a surrogate for the expectations of major trading groups. A second important factor is the selection of futures whose prices are relatively insulated from natural phenomena such as floods and pestilence. An appropriate length for the training period avoids the potential danger of deriving network parameters based on patterns that are no longer relevant in representing market behavior. Last, the research is aided by trading selectively rather than in every period.

2.2 Futures Volume

This study (Kaastra and Boyd, 1995) attempts to forecast futures trading volume on out-of-sample data. The backpropagation neural network is used to forecast futures trading volume for six commodities traded on the Winnipeg Commodity Exchange. The forecasts are designed to improve the budgeting of transaction fees. Neural networks are used because of their ability to approximate any continuous function and learn nonlinear patterns that may be present in the data and incorporate them into forecasts.

Separate neural network forecasting models are created for barley, canola, flax oats, rye, and wheat. Trading volume is forecast up to nine months ahead with a single neural network specializing in each step-ahead forecast. The five independent variables used as inputs to the neural networks include: three-period moving averages of lagged trading volume, open interest, futures price variability, mean cash prices, and producer grain deliveries. Three, five, seven, and nine hidden neurons are trained and evaluated on the testing sets.

The results indicate that the neural networks are able to forecast up to nine months ahead and outperform the naive model for all commodities except barley an rye. The best forecasts are obtained for canola and wheat which together account for over three-quarter of total trading volume and are therefore the most important commodities for budgeting purposes. The neural network forecasts relative to the naive model currently in place to forecast futures trading volume at the WCE do not deteriorate as the forecast horizon increases. The neural network also outperforms an ARIMA model.

In terms of network topology, neural networks with three hidden neurons are selected for 23 out of the 54 networks. Networks with five and seven hidden neurons are selected almost equally often, while no network with nine hidden neurons performs well on the testing set.

3 Analysis of existing software

4Thought is a computer package developed by Right Information Systems Ltd. It is based on a special case of feedforward neural network known as a multi-layer perceptron. The basic idea is to model the series of interest, the output, in terms of explanatory variables, or inputs. Lagged values of both output and the inputs may also be used. Below are some of the experiments that were tried on '4Thought' (Harvey and Toulson, 1994).

Random walks — 100 observations were generated from a random walk and 4Thought was asked to fit a model with a single lag and a time trend. The coefficient of the lagged observation changed with the observation but was always less than 0.8, which is exactly what one would expect from unit root theory. Nevertheless the results were quite reasonable in so far as the eventual forecast function was not too far from a horizontal line projected out from the last observations. However, given that many financial time series are random walks, this is an important hypothesis to test at the outset.

Trends and cycles — The authors next considered quarterly seasonably adjusted US real investment over the period 1951 Q1 to 1985 Q4. This series has a strong upward trend and exhibits marked cycles. It can be modeled by a structural time series model, in which there is a stochastic trend and cycle. Since a program may not pick up the cycle if it is chosen to be too parsimonious the authors allowed four lags as well as a time trend. The forecasts were similar to those produced by the structural time series model with a weak cycle being projected on top of the trend.

Outliers — In order to get some idea of how robust 4Thought is to outliers, the authors took a regression example that concerns a time series of 24 annual observations on international telephone calls from Belgium. The data follows a linear trend but there is a block of six observations which are well above the trend due to a different measuring system being used. Least squares regression is adversely affected by these outliers. As a result its forecasts of future observations are too high, though the outliers are easily spotted and so can be omitted. 4Thought fitted a nonlinear function which passed close to the outlying observations and consequently produced forecasts heading sharply downwards rather than slowly upwards. When they modified the data by replacing the last four outliers by observations close to the trend line, 4Thought again produced downward forecasts though not as extreme as before. Finally with only one outlier 4Thought produced forecasts with the mild upward pattern which the data obviously requires. Thus although 4Thought shows some degree of robustness, the very flexibility of the neural net approach means that it cannot adequately cope with some of the difficult situations which have been studied in the statistical literature and which have led to the development of procedures which statisticians regard as being robust.

Volatility of financial time series — In order to examine the claims of 4Thought to be an effective method of extracting information from noisy data financial time series were considered, namely the daily exchange rate of the US dollar against the deutschmark. There are 946 observations starting on 1 October 1981 and ending on 28 June 1985. The data analysed in Harvey, Ruiz and Shephard (1994). The first differences of the logged exchange rates, y_t, appear to be random, but the squares exhibit strong serial correlations, indicating that volatility changes over time. As an alternative to ARCH modeling, Harvey, Ruiz and Shephard build a model in which the logarithms of the squared observations are a random walk plus noise. In order to see how 4Thought copes with volatility, it was ran on $|y_t|$, y_t^2 and $\log y_t^2$. Lags were allowed to enter the model, but no time trend. While the very parsimonious model in Harvey, Ruiz and Shephard was successful in capturing an underlying pattern in changing volatility, nothing very clear emerged from 4Thought. This is probably related to the fact that a pure autoregression does a poor job when the signal is dominated by noise, so that the series is close to being noninvertible.

The author conclude that 4Thought does a reasonable job in many cases and would probably beat many other semi-automatic model fitting programs, not to mention the use of simple regression techniques. However, it is no no substitute for careful modeling carried out with state of the art techniques. If neural nets are to play a useful role in forecasting economic time series, they will need to be combined in some way with the more traditional techniques. At the very least, some kind of preliminary time series analysis, possibly involving the fitting of a few basic models, is needed to establish the salient features of the data and to test some fundamental hypotheses.

4 Learning from Hints

When a neural network learns its target function from examples (training data), it knows nothing about the function except what it sees in the data. In financial market applications, it is typical to have limited amount of relevant training data, with high noise levels in the data. The information content of such data is modest, and while the learning process can try to make the most of what it has, it cannot create new information on its own. This poses a fundamental limitation on the learning approach, not only for neural networks, but for all other models as well. It is not uncommon to see simple rules such as the moving average outperforming an elaborate learning-from-examples system. Learning from hints is a value-added feature to learning from examples that boosts the information content in the data. The method allows us to use prior knowledge about the target function, that comes from common sense or expertise, along with the training data in the same learning process.

In (Abu-Mostafa, 1995) paper, he explains the difficulties in learning from the very noisy data. Abu-Mostafa defines input, x, as 'very noisy' if forecast, y, and target output, $\hat{y}$, agree only $\frac{1}{2}+\epsilon$ of the time (50% performance range). In the 50% range, a performance of $\frac{1}{2}+\epsilon$ is good, while a performance of $\frac{1}{2}-\epsilon$ is disastrous. During learning we need to distinguish between good and bad hypotheses based on a limited set of N examples. The problem with the 50% range is that the number of bad hypotheses that look good on N points is huge. This is in contrast to the 100% range where a good performance is as high as $1-\epsilon$. The number of bad hypotheses that look good here is limited. Therefore, one can have much more confidence in a hypothesis that was learned in the 100% range that one learned in the 50% range. It is not uncommon to see a random trading policy making good money for a few weeks, but it is very unlikely that a random character recognition system will read a paragraph correctly.

For the experiment the symmetry hint was crafted into the neural network algorithm. It asserts that if a pattern in the price history implies a certain move in the market, then this implication holds whether you are looking at the market from the U.S. Dollar or the exchanged currency viewpoint. Formally, in terms of normalized prices, the hint translates to invariance under inversion of these prices. It was also shown in the paper that the information content of the hint, rather than the incidental regularization effect resulted in the performance differential.

The experiment was ran to learn foreign-exchange trading of the U.S. Dollar versus the major currencies over a period of 32 months. Without hints neural network was reaching $\sim 5\%$ in annualized percentage returns (cumulative daily, unleveraged, transaction cost included), for a sliding one-year test window in the period from April 1988 to November 1990, averaged over the four major FX markets with more than 150 runs per currency. The symmetry hint described above helped to achieve $\sim 10\%$ in annualized returns which is significant improvement over no-hint algorithm.

5 Conclusions

In this paper we reviewed several successful applications of neural networks to financial forecasting. We showed how the neural networks are used to predict the futures prices and trading volume. Also the analysis of the commercial software was presented, including the test on forecasting the financial time series. Currency exchange markets were used for applying hints. Thus this review covered many important areas of financial forecasting where neural networks may be successfully used.

A new idea of using hints may prove to be the crucial element of the future learning systems, since the information contained in the past observations is not always enough for successful forecasting. In the future research we would like to apply the newest machine learning techniques for the financial forecasting, for example model-based learning (Kuvayev and Sutton, 1996) that can learn the model of the environment and use it to facilitate learning. The financial forecasting field is appealing to us because of the challenges that arise in learning from very noisy data.

Finally we would like to pose a paradox. As the forecasting methods evolve and perform better, people should be able to make profits from using these methods. However the efficient market hypothesis asserts that the arbitrage opportunities will cease quickly. Would it mean that the learning and statistical methods are not predicting anymore? It should not, since there still will be dependencies and correlation in the data. But here the cycle repeats. Another philosophical question is whether it is possible, at least theoretically, to build a gigantic neural net that is complicated enough to learn the average person's behavior in response to various price changes and events. If such network is possible then how much profit can it make if it used in trading.

References

[1] Abu-Mostafa, Y.S. (1995), Financial applications of learning from hints, *Advances in Neural Information Processing Systems*, 7:73-80, Morgan-Kaufmann.

[2] Grudnitski, G., and Osburn, L. (1993), Forecasting S&P and Gold Futures Prices: An Application of Neural Networks, *The Journal of Futures Markets*, 13:631-643.

[3] Harvey, A. C., Ruiz, E., and Shepard, N. (1994), Multivariate stochastic variance models, *Review of Economic Studies.*

[4] Harvey, A. C., and Toulson, S. (1994), Review of '4Thought', *International Journal of Forecasting*, 10:35-41.

[5] Hill, T., Marquez, L., O'Connor, M., Remus, W., (1994), Artificial neural network models for forecasting and decision making, *Internation Journal of Forecasting*, 10:5-15.

[6] Kaastra, I., and Boyd M.S. (1995), Forecasting futures trading volume using neural networks, *The Journal of Futures Markets*, Vol. 15, No. 8, 953-970.

[7] Kuvayev, L., and Sutton, R.S. (1996), Model-based reinforcement learning with an approximate, learned model, submitted to *Advances in Neural Information Processing Systems, 8.*