A Neural Network Model for the Gold Market

Peter J. McCann
Department of Computer Science
Washington University
Campus Box 1045
St. Louis, Missouri 63130-4899
pjm3@cs.wustl.edu

Barry L. Kalman
Department of Computer Science
Washington University
Campus Box 1045
St. Louis, Missouri 63130-4899
barry@cs.wustl.edu

Abstract

A neural network trend predictor for the gold bullion market is presented. A simple recurrent neural network was trained to recognize turning points in the gold market based on a to-date history of ten market indices. The network was tested on data that was held back from training, and a significant amount of predictive power was observed. The turning point predictions can be used to time transactions in the gold bullion and gold mining company stock index markets to obtain a significant paper profit during the test period. The training data consisted of daily closing prices for the ten input markets for a period of about five years. The turning point targets were labeled for the training phase without the help of a financial expert. Thus, this experiment shows that useful predictions can be made without the use of more extensive market data or knowledge.

Keywords — Neural networks, financial analysis.

1 INTRODUCTION

In recent years, the financial analysis community has paid increasing attention to neural network applications. Understandably, the desire for a good financial prediction system is a strong motive for such research. Many applications have been

developed, some of them quite successfully.

The applications seem to fall naturally into several broad categories. First, there are individual or single corporation analyses, such as credit assessment and bankruptcy prediction, that consider input data for only one entity and attempt to make a guess about future performance [10]. The alternative to this approach would be a system that makes guesses about a large scale commodities market or an aggregate stock index [2].

The second distinction is the one between pattern classification approaches and those that attempt to predict a numeric value such as a stock price or dividend return. White [14] takes the time series prediction approach for IBM daily stock dividends. On the other hand, Trippi [12] developed a trading system based on pattern classification. He relied on his neural networks only for a long/short recommendation.

Note that the two distinctions are more or less orthogonal in that there are applications representing each of the four combinations of features. This work focuses on forecasting price trends in the gold market. The problem addressed here thus falls into the large market, pattern classification category.

One of the main tools financial analysts use to solve the market prediction problem is multivariate linear regression. Linear regression, like the neural network approach, attempts to fit a data set by deriving a set of coefficients that minimize some error function. Unlike a neural network, however, hypotheses arrived at by way of linear regression can have only a very simple functional form, that is, a linear combination of their inputs. On the other hand, a neural network is a complicated nonlinear function, sometimes with two or more nested levels of nonlinearity. There have been several direct comparisons made in the literature between neural network and linear regression techniques applied to data fitting [9]. This might be a mistake.

The most successful financial neural network applications seem to be the ones that do not attempt exact numeric prediction of a price, but that rather attempt to recognize patterns in the input data that can give clues to where the market might be headed.

Also, most of the work to date has been directed at introducing the concept of a neural network to the financial analysis community. As a result, the networks used tend to be of the simple feed-forward variety, and are usually trained with first order back-propagation algorithms. Except for Kamijo [11], very little work in this area has made use of recurrent networks. We believe our second order conjugate gradient training method [5] and our data pretreatment procedures using singular value decomposition [6] represent significant steps forward in this area. They allow us to train networks in fewer epochs and with poorer data, and to obtain better generalization than previous methods.

2 METHODS

Figure 1 shows the behavior of the gold market from September 1, 1988 through January 24, 1994. Daily closing indices during the same period for nine other major markets were also included. These were an oil index, an aggregate commodities

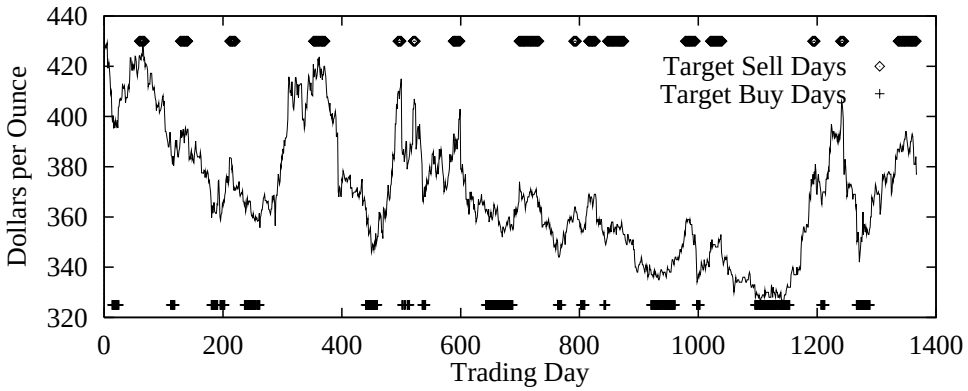

Figure 1: The target trading days. The goal was to train a network to recognize the optimal buying and selling dates from purely historical data.

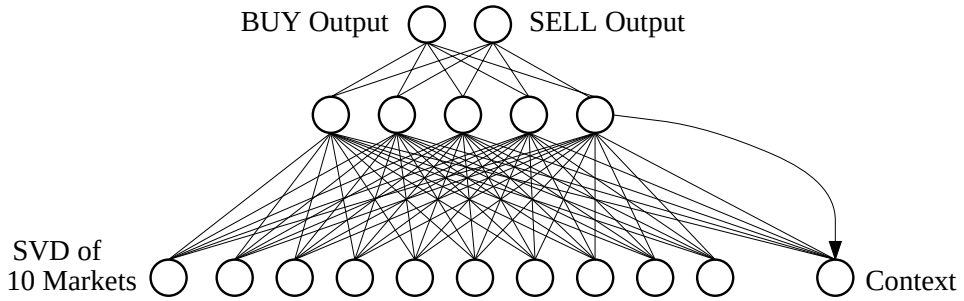

Figure 2: The best performing network, consisting of 10 input units, 5 hidden units, and 1 feedback unit. The feedback or context unit is a copy of one of the hidden units from the previous feedforward pass. Not pictured are the skip connections between the input layer and the output layer.

index, the Standard and Poor's 500 index, a dollar index, a bond index, the 30 year and 10 year bond yields, the Sterling currency index, and the gold mining index.

The objective of this experiment was to identify the turning points in the market as they happen, i.e., the network was to predict, based on a to-date history of the ten input markets, whether the price of gold was at the top of a peak or the bottom of a trough. The architecture used is shown in figure 2. This arrangement was arrived at mainly through trial and error. The single context unit provided the network with some means of recording the market history while not introducing so many new variables that generalization was impeded.

The training targets were derived without the aid of an expert, but in a somewhat ad hoc manner. The author simply looked at the performance of the gold market and chose trading days on which it would have been profitable to buy or sell, given perfect 20/20 hindsight. This implicitly defined a trading frequency and therefore a risk factor [1]. Figure 1 is marked with the trading day targets that

Table 1: The target output encodings.

ENCODING	MEANING
(+1,-1)	Gold is in a trough. BUY.
(-1,+1)	Gold is at a peak. SELL.
(0,0)	No decision.

were chosen. Although the target patterns were not derived in any methodical way such as filtering the data and then performing peak detection, there was useful information correlated with these trading days that the network was able to take advantage of, as will be shown by the validation phase.

The target for a given day was encoded in a two dimensional vector, as shown in table 1. Note that our in-house training software uses a squashing function with domain (-1,+1). We have found that a hyperbolic tangent squashing function coupled with a scaled square error function to be much more effective in training. See [7] for a complete derivation of these embellishments on the standard neural network model.

The 1369 labeled input patterns were then preprocessed by singular value decomposition. This is a linear data transformation that finds a new orthogonal basis set for the 1369 input vectors. It operates only on the input values and leaves the target outputs untouched. It outputs a new set of input vectors and the transformation matrix that it used to create them, so that more patterns can be added to the training corpus if needed. In addition, weights can be back-transformed into the original input space after training so that new data can be tested without reapplying SVD. The decomposed vectors in the new orthogonal basis set often allow for faster training. See [6] for a more complete discussion of the benefits of this pretreatment method.

The first training attempt was performed with a slightly larger network than the one shown in figure 2. The first attempt had one more context unit than the final, most successful network. The first 900 patterns were used for training and the remaining 469 for testing. An indication of the network's confidence in the market was produced by subtracting the "sell" output from the "buy" output. The result is graphed in figure 3a.

This network may have minimized the error function with respect to the targets it was given during training, but evaluation of its true merit should be based on the profitability of a trading strategy derived from it. This set of outputs, not surprisingly, did not perform very well. The best trading strategy found involved the use of a buy threshold $\overline{b}$ and a sell threshold $\underline{s}$. Call the network indicator I. The trading strategy was defined as follows:

> If we are in the market, and it has been the case that $I < \underline{s}$ for n consecutive days or more, then sell.
>
> If we are not in the market, and it has been the case that $I > \overline{b}$ for n consecutive days or more, then buy.

The parameters $\overline{b}, \underline{s}$ and n were adjusted experimentally for optimum performance.

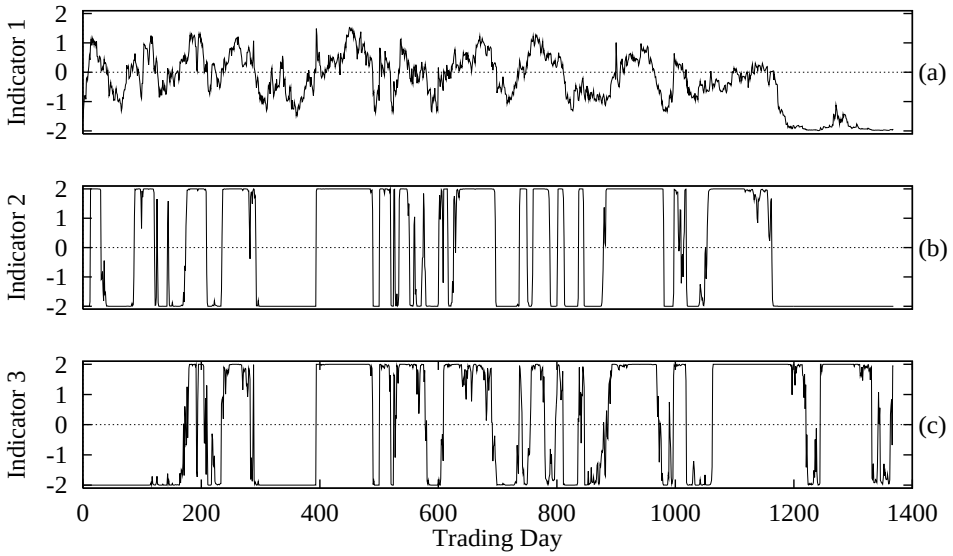

Figure 3: The three experimental indicators. (a) The original network trained on the first 900 patterns. (b) The original network trained on the first 900 patterns, with (0,0) patterns subtracted from the error function. (c) The smaller network trained on the first 1169 patterns, with (0,0) patterns subtracted from the error function.

The price used in the calculation of profit is the closing price on the day after the decision to buy or sell has been made. The profit was evaluated over the 469 trading days in the testing corpus, and only a 5% profit was observed. Note also that no trades were performed during the last 200 trading days, even though there seem to be many opportunities in the market during this period.

The second training run was made on the same network architecture, but with the error from (0,0) targets subtracted from the error function. Note that we still need to present these patterns to the network because of the recurrent connections. While the network is not being trained on these patterns directly, they do make up part of the market history and are therefore important clues to the classification of the (+1,-1) and (-1,+1) patterns. As expected, the indicator derived from this network is much less ambiguous about its predictions, as shown in figure 3b.

The second network still exhibits a problem when it is asked to perform too far from the training set. The indicator drops to -2 after trading day 1174 (April 23, 1993) and stays there for the rest of the testing set. It seems reasonable to assume that at this point, the underlying market forces are simply very different from the ones the network was trained to recognize. To check this, some of the testing patterns were added to the training set, and the testing set was made smaller. The graph in figure 3c shows the derived indicator after training on patterns from day 1 through day 1169.

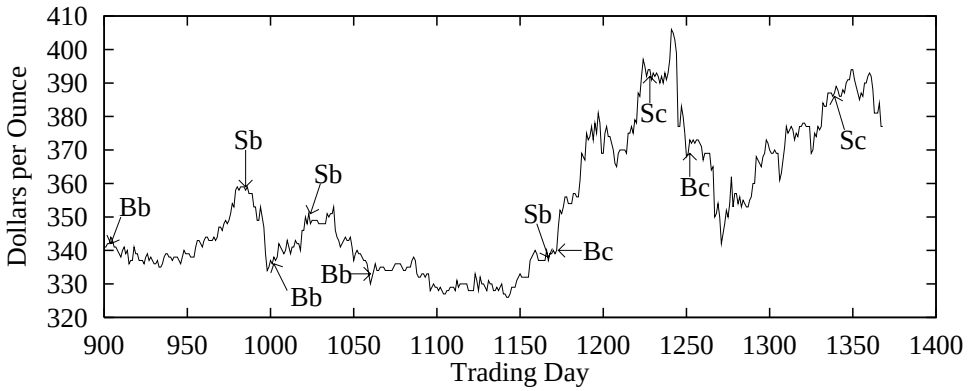

Figure 4: Trades made according to the indicators in figure 3. The trades made by indicator b from figure 3 are marked with a b. The testing period is from trading day 901 to trading day 1369. Note that no trades are made after day 1174. The trades made by indicator c from figure 3 are marked with a c. The testing period is from trading day 1170 to trading day 1369.

3 RESULTS

The indicators were used to make simulated trades according to the strategy defined above, with $\overline{b} = 1.9$, $\underline{s} = -1.9$, and $n = 3$. The resulting trades are shown in figure 4 along with the gold market performance. Buy transactions are shown with a "B" and sell transactions are shown as an "S" on the trading day and at the price each was made. The indicator from figure 3b made six trades before its performance degraded, for a profit of 11% in about 13 months. (270 trading days). The indicator from figure 3c made four trades during a test period of about 9 months (200 trading days), for a profit of 21%. If these trades are made on the same dates in the gold mining company stock index, the paper profits are 50% and 64% respectively. The gold mining company stock index is a much more volatile market than the gold bullion market, but one which follows the same trends. This is because the profitability of gold mining concerns increases dramatically with a small increase in the price of gold, and decreases dramatically with a small decrease in the price of gold.

Perhaps more important than these numbers, however, is how well the network seems to recognize the peaks and troughs as they are happening. The early test cases in figure 4 are a case in point. The network seems to have learned to predict a market turning point of a certain time scale almost precisely.

The reason for the poor performance of the first network is that the indicator is very nearly an exact reflection of the market over the period in which it was trained. The network is really not performing a pattern classification task, but is attempting to fit the targets it was given. By taking the spurious (0,0) targets out of consideration, the network learns to make a more definite commitment at each data point. This makes the resulting indicator much more useful.

The degradation of performance late in the testing set of the first two networks is to be expected as conditions in the world financial markets slowly evolve. In fact, it is surprising that the nonlinear regularities found by the second and third networks continue to be present for so long (13 months in the case of the second network) after training.

An additional clue is given by the failure of the third network to adequately fit its early training set. It is perhaps fortunate that it did not, since fitting this data would probably make it perform less well on its testing set. The training of this third network seemed especially sensitive to the initial random choice of weights, and this might be because the early training cases were acting as a hindrance rather than a help. This seems to give rise to a natural training window of about 1000 trading days. Data points older than this should probably be dropped from training. Future experiments will test this hypothesis.

4 CONCLUSIONS

The neural networks trained here have been shown to make accurate predictions about gold market turning points. A trading strategy was developed that used a market indicator derived from a trained network's outputs to make profitable trades. The training patterns were derived without the help of a financial expert, but the networks obtained still were able to recognize the underlying structure inherent in the market. The main conclusion that can be drawn from this work is that there are indeed nonlinear regularities present in the interactions of market forces, and that a neural network can be trained to recognize them.

Much of this result is probably dependent on the scale of the turning points chosen as examples. A shorter trading cycle would perhaps lead to higher profits, but it would also increase the risk associated with each trade. A longer trading cycle probably increases the accuracy of the predictions, because the long term trends are less affected by random daily noise. Here, the scale was chosen somewhat arbitrarily by the author as a guess at what a good tradeoff between the two would be. A more methodical choice of target patterns is a good area for future work, perhaps with the aid of a financial expert.

This work also indicates that there is enough information to train networks successfully in merely the daily closing prices of the ten markets under consideration. The previous successes of Kamijo [11] and Trippi [12] were based heavily on additional factors like daily high and low prices. Because of this data, Trippi was able to make trades (and profits) on a daily basis, instead of the approximate four month trading cycle seen here. The longer term forces analyzed by these networks might not be quite as profitable, but might someday provide more insight into medium scale economic trends.

Acknowledgements

The authors' thanks go out to Ravi Aggarwal who contributed the financial market data and invaluable insight for this experiment. Thanks also to Stan Kwasny, James Hu, and Ian Flanagan who provided comments on early drafts.

References

[1] Bosarge, W.E. Jr. (1991). Adaptive Processes to Exploit the Nonlinear Structure of Financial Markets. *Neural Networks and Pattern Recognition in Forecasting Financial Markets* (February 1991), Santa Fe Institute of Complexity Conference.

[2] Collard, J.E. (1992). Commodity Trading with a Three Year Old. *Neural Networks in Finance and Investment*, ed. R. Trippi and E. Turban, 411-420. Chicago: Probus Publishing Co.

[3] Elman, J.L. (1990). Finding Structure in Time. *Cognitive Science* **14**, 179-211.

[4] Hoptroff, R.G., Bramson, M.J., and Hall, T.J. (1991). Forecasting Economic Turning Points with Neural Nets. *Proceedings of the IEEE International Joint Conference on Neural Networks* (Seattle 1991), vol. I, 347-352.

[5] Kalman, B.L., and Kwasny, Stan C. (1993). TRAINREC: A System for Training Feedforward and Simple Recurrent Networks Efficiently and Correctly. Technical Report WUCS-93-26, St. Louis: Department of Computer Science, Washington University.

[6] Kalman, B.L., Kwasny, Stan C., and Abella, A. (1993). Decomposing input patterns to facilitate training. *Proceedings of the World Congress on Neural Networks*, vol. III, 503-506. Hillsdale, NJ: Lawrence Erlbaum Associates.

[7] Kalman, B.L., and Kwasny, Stan C. (1992). Why Tanh: Choosing a Sigmoidal Function. *Proceedings of the IEEE International Joint Conference on Neural Networks* (Baltimore 1992), vol. IV, 578-581. New York: IEEE.

[8] Kalman, B.L., and Kwasny, Stan C. (1991). A Superior Error Function for Training Neural Nets. *Proceedings of the IEEE International Joint Conference on Neural Networks* (Seattle 1991), vol. II, 49-52. New York: IEEE.

[9] Marquez, L., et. al. (1991). Neural Network Models as an Alternative to Regression. *Proceedings of the IEEE 24th Annual Hawaii International Conference on Systems Sciences* (Hawaii, 1991), vol. VI, 129-135. Hawaii: IEEE Computer Society Press.

[10] Odom, M.D., and Sharda, R. (1990). A Neural Network Model for Bankruptcy Prediction. *Proceedings of the IEEE International Joint Conference on Neural Networks* (San Diego 1990), vol. II, 163-168. New York: IEEE.

[11] Kamijo, K., and Tanigawa, T. (1992). Stock Price Pattern Recognition: A Recurrent Neural Network Approach. *Proceedings of the IEEE International Joint Conference on Neural Networks* (San Diego 1990), vol. I, 215-221. New York: IEEE.

[12] Trippi, R., and DeSieno, D. (1992). Trading Equity Index Futures with a Neural Network. *Journal of Portfolio Management* **19** (Fall 1992), 27-33.

[13] Trippi, R., and E. Turban (eds.) (1992). *Neural Networks in Finance and Investment: Applying Artificial Intelligence to Improve Real-World Performance.* Chicago: Probus Publishing Co.

[14] White, H. (1988). Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns. *Proceedings of the IEEE International Conference on Neural Networks.* (July 1988), vol. II, 451-458.