Forecasting S&P and Gold Futures Prices: An Application of Neural Networks

Gary Grudnitski
Larry Osburn

INTRODUCTION

Forecasting futures prices is an integral component of a profitable futures trading strategy. Current research suggests the standard random walk assumption of futures prices may actually be only a veil of randomness that shrouds a noisy nonlinear process [see, for example, Savit (1990) and Tvede (1992) for an explanation of the apparent randomness in these price time series, and Frank and Stenzos (1989), Blank (1991), and DeCoster et al. (1992) for evidence of the existence of nonlinear structures of rates of return]. What is called for to remove this veil and thereby make futures price forecasting possible is the application of nonlinear models. Further, some believe that nonlinear prediction can be enhanced by incorporating information about the sentiments of major trading groups of these futures [e.g., Leuthold et al. (1989) and Turner et al. (1992)]. Accordingly, this research examines the feasibility of employing neural networks to forecast price changes of Standard & Poor's 500 Stock Index and gold futures based on past price changes[1] and historical open interest patterns that are held to represent the beliefs of a majority of the traders in these markets.[2]

This article is organized as follows. In the next section, the rationale for the factors selected to forecast price changes of these futures are provided. The third section

We would like to acknowledge N. Marovac and R. Swiniarski of the Department of Mathematical Sciences at San Diego State University for introducing us to neural networks. Also, the helpful comments of E. Omberg and N. Varaiya of the Department of Finance, San Diego State University, and two anonymous referees are gratefully acknowledged. Last, we would like to thank the School of Accountancy, San Diego State University for providing financial support for this project.

[1]White (1988) used neural networks in an attempt to predict daily IBM stock prices. Harston (1990, p. 396) made the following comments on White's efforts:

> Unfortunately, the results were disappointing. In some ways, this result could have been expected. After all, neural networks can only process information, make data transformations and detect patterns. They cannot make up something from nothing. Where no information exists, neural networks cannot magically find meaning.

[2]Presumed here is that traders' beliefs lead to actions and market prices reflect those actions. Therefore, by analyzing patterns of traders' beliefs, discernible trends in prices of specific futures can be forecast. Inclusion of variables that reflect beliefs or expectations in a neural network forecasting model of trading behavior is warranted by the recent works of Zaremba (1990) and Collard (1991).

Gary Grudnitski is a Professor of Accountancy at San Diego State University.
Larry Osburn is a member of the Technical Staff at TRW Mead Company.

The Journal of Futures Markets, Vol. 13, No. 6, 631–643 (1993)

CCC 0270-7314/93/060631-13

discusses the appropriateness of a neural network approach to the prediction of changes in an economic series. The fourth section describes the data on which the forecasts are produced, and the fifth section reports the results. A summary is given in the last section.

FORECASTING PRICE CHANGES

This study presumes that two factors other than specific price trends are related to price movements of futures: namely, general economic conditions, and traders' expectations about what will happen in the market for these futures. The first factor stems from the literature [Stein (1989)], which suggests that market prices of many futures are affected by certain underlying economic conditions. For example, the price of crude oil, although subject to the machinations of OPEC, is still a function of the value of the dollar because the world pays for its oil with dollars [Cousin (1990)]. This influencing tendency may manifest itself as a seasonal (or longer) futures-specific, cycle-driven positioning of traders. What results is an exaggeration of or a reduction in the short-term patterns of price behavior in the futures market, and, may be accounted for by including a variable that evidences high correlation with the futures' price movements.

A number of studies focusing on the futures markets [Teweles and Jones (1987); Hadady (1987); Kaufman (1987); Hadady et al. (1987); Band (1986); and Yoo and Maddala (1991)] indicate that over time small traders lose money and large traders, particularly large speculators, make money. To capture the trading strategy of both of these trading groups, data are gathered from a publication[3] of the Commodity Futures Trading Commission (CFTC). These data, which are held as representing traders' expectations about what will happen, comprise the second factor hypothesized to influence price changes of Standard and Poor's Stock Index and gold futures.

FORECASTING WITH NEURAL NETWORKS

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 [Eberhart and Dobbins (1990, p. 7)]. The Appendix contains a technical description of the computational procedure used by neural networks to derive these maps.

Hecht-Nielsen [(1990, p. 120)] makes the following case for the predictive superiority of neural networks over general linear models[4]:

> 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 summary, enough experimental evidence has now been gathered to state with some confidence that mapping networks are, in general, different than statistical regression approaches. The function approximations that arise from properly applied

[3]This monthly report first published in 1983, lists the size and nature of the open interest positions for large and small traders.

[4]Sharda and Patil (1990) also found that neural networks forecast about as well or better than the Box–Jenkins technique.

mapping networks (at least in instances where sufficient training data are available) are usually better than those provided by regression techniques.

To date two studies have used neural networks to determine prices in futures markets. Zaremba (1990) focused on the training facility of neural networks to Standard & Poor's 500 Stock Index, Gold, and Treasury-bond futures prices. Once these networks were trained, however, Zaremba stopped short of systematically testing their forecasting ability. An effort more oriented at determining the ability of neural networks to produce accurate futures price forecasts was made by Mendelsohn and Stein (1991). They trained a neural network on three years of daily Deutche mark futures prices to generate buy and sell trading signals. Over a one-year test period half of the 24 trades advised by their network were correct, resulting in a profit of $13,252 net of transaction costs.

Neural networks can also learn associations between input patterns and output classes.[5] Once they learn to associate patterns to classes, they can generalize the responses they make to novel inputs that are similar to the input patterns on which they have been trained [McClelland and Rumelhart (1988, p. 83)].[6]

DATA

Network models of the International Monetary Market's Standard & Poor's 500 Stock Index futures market (S&P) and the Commodity Exchange (COMEX) Incorporated's Gold futures market (Gold) are constructed for the period December, 1982, to September, 1990. These markets are chosen because they are indicative of the general direction of the stock market and inflation, and are relatively insulated from such natural phenomena as floods, droughts, and pestilence, which affect the prices of many other futures.

Inputs

The types of inputs to the neural networks are as follows:

1. The monthly growth rate of the aggregate supply of money, M-1. M-1 is intended to represent an underlying economic factor that influences both the S&P and Gold markets [Cousin (1990)]. These data are from *Barron's*.
2. The change in and volatility of S&P and Gold futures prices. The price inputs must coincide with the availability of the commitments information (described next), which is made public approximately two and a half weeks after month end. This suggests the derivation of a mean price that is centered around month end. Accordingly, using *Barron's* weekend closing prices for the most widely held futures contract, the derived month end price, P_m, includes the weekend closing prices two weeks before and two weeks after the end of the month (including an adjustment for the situation where the last weekend of the month does not fall on the last trading day of the month). Specifically, P_m is calculated as follows:

$$P_m = \sum P_i/5$$

where $i = -2$ to $+2$, and P_i is the weekend closing price measured i weeks from the end of month m.

[5]Evidence of the performance of neural networks versus statistical pattern classification techniques such as maximum likelihood and Bayesian estimation is given in Shadmehr and D'Argenio (1990).

[6]As discussed later, utilization of the associative capability of neural networks allows the construction of similarity measures for each forecast.

First differences in monthly prices are then calculated and are graphed in Figures 1 and 2. Additionally, the volatility of futures prices is represented by using the centered mean and weekend closing prices around the centered mean to compute a standard deviation of prices[7] for each month.

3. End-of-month net percentage commitments[8] of large speculators, large hedgers, and small traders. As mentioned previously, these data, which first became available in January, 1983, for the two markets studied, is from monthly volumes of the *Commitments of Traders in Futures Reports,* published by the CFTC.[9]

Outputs

For forecasting purposes, the networks have one output. This output is the change of the monthly centered price mean for the forecasted month. To assess the quality of the forecast, an output matrix is employed by the associative networks. The rows of this output matrix consist of a one and all zeros, where the one's location corresponds to an individual input pattern. As mentioned previously, one of the strengths of a neural network lies in its ability to discriminate one pattern from another, and then use that discriminating power to tell something about a pattern that has not been seen before. To illustrate this fact consider the following situation where there are two patterns, PT_1 and PT_2. A neural network can be trained to recognize these patterns consistently by telling it that every time it sees PT_1 to relate it to the values 1 and 0 of its two output nodes, and every time it sees PT_2 to relate it to the values 0 and 1 of these nodes. After the network is trained, assume a third pattern, PT_3, is introduced. The neural network can

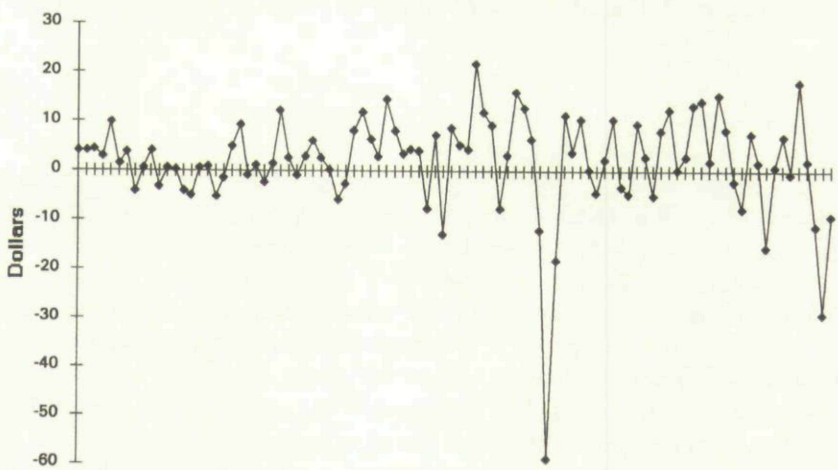

Figure 1
S&P futures price changes—January, 1983 to September, 1990.

[7]Representing volatility by the standard deviation of prices is in accord with the research of Wilson et al. (1988); Herbst and Maberly (1990); and Pliska and Shalen (1991).

[8]Five hundred and 300 contracts are the reporting levels for the S&P and Gold markets, respectively. Net percentage commitments are the long minus short positions of the trading groups divided by the total open interest in the future.

[9]In the *Commitments of Traders in Futures Report,* commitments of large speculators are listed under the heading: "Reportable Positions/Non-Commercial/Long or Short Only/Long-Short." Commitments of large hedgers are listed under the heading: "Reportable Positions/Commercial/Long-Short." Commitments of small traders are identified under the heading: "Nonreportable Positions/Long-Short."

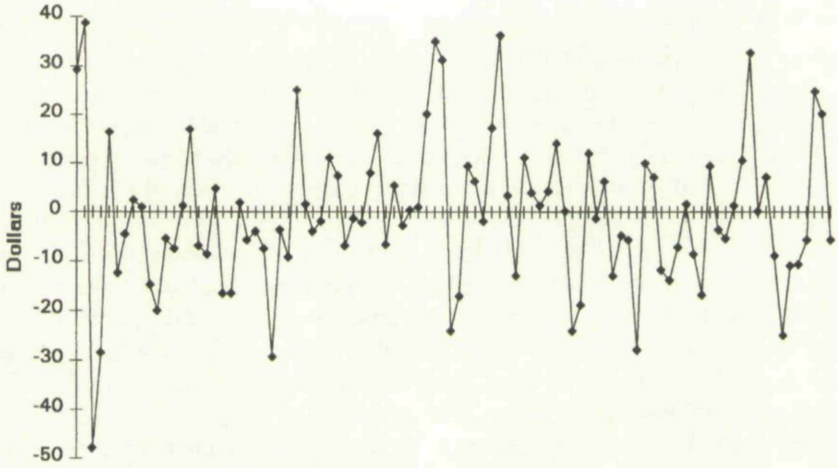

Figure 2
Gold futures price changes—January, 1983 to September, 1990.

determine how similar this pattern is to the two it has learned. Using the derived weights from training it will produce a value between 1 (completely similar) and 0 (completely dissimilar) for each of the output nodes, which can be interpreted as the third pattern's similarity to PT_1 and PT_2, respectively.

Network Description

The forecast and similarity networks have 24 input nodes. These input nodes represent six inputs per month (i.e., price change, volatility, three sentiment percentages, and M-1) presented four months at a time. For example, the August, 1989, similarity value and forecast are produced from the April, May, June, and July, 1989, input patterns.

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 $I_0 \ldots I_n$, is somehow related to the following sequence of output values $O_0 \ldots O_n$, and is defined entirely within the data set. More formally, the compositions of the patterns are as follows:

$$t_0\colon\ I_0 = \{V_0, \ldots, V_3\} \longrightarrow O_0 = \{V_4\}$$

$$t_1\colon\ I_1 = \{V_1, \ldots, V_4\} \longrightarrow O_1 = \{V_5\}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$t_j\colon\ I_j = \{V_j, \ldots, V_{j+3}\} \longrightarrow O_j = \{V_{j+4}\}$$

The forecast networks have two hidden layers and one output node. The similarity networks have one hidden layer and 15 outputs nodes. The networks are designed with 24 nodes in their first hidden layer[10] (and the only hidden layer of the reliability networks), and eight nodes in their second hidden layer.

[10]Although the number of hidden nodes for a single, hidden layer network is initially set at $2i + 1$, where i is the number of input nodes, training of the networks occurs only when the number of nodes in the hidden layer is reduced to 24.

Network Training

Following the framework of Lukac and Brorsen (1989) and Zaremba (1990), it is first necessary to establish the duration of the training period. Here the goal is to balance the requirement of providing enough training patterns so that adequate learning can take place against the competing desire to test the ability of the networks to generalize during bullish, bearish, and trendless market periods of a complete business cycle.

Toward this goal, the 90 periods of data are initially divided into training sets of 30, 45, and 60 months duration. Holdout patterns in the subsequent 60, 45, and 30 months, respectively, are evaluated then by the networks as to their similarity to the input patterns in the training sets. For these partitionings of the data, no holdout pattern is found to be similar[11] to a training pattern that occurs earlier than the 15 preceding months. Accordingly, the networks are adaptively trained from the patterns representing the most recent 15 months of data.

Training consists of presenting patterns from the 15-month training sets to the networks. On each iteration through the complete training set, the parameters of the networks are modified to minimize the average sum-squared error between the target values and the calculated values of the training set.

RESULTS

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. Here, 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.

To initiate a trade in any period then, the following two conditions must be met:

1. The test input pattern must exhibit a similarity of more than 0.5 to one or more of its 15 training input patterns. For the S&P and Gold networks, 45 and 51 of the 75 test patterns show this degree of similarity to their training input patterns.
2. When a test pattern shows more than a 0.5 level of similarity to more than one of its 15 training patterns, the sign of the actual price change of all similar training patterns must agree. This further restriction reduces the number of trading periods to 41 for both markets.

Where a trade is simulated the forecast network predicts the correct sign (i.e., whether the price change is positive or negative) of the next month's centered mean price change of a S&P contract 75% of the time, and a Gold contract 61% of the time.[12] This relative difference in forecast accuracy in the two markets is significant, and likely attributable

[11]On a scale of 0–1, where 0 represents complete dissimilarity and 1 represents perfect similarity, test patterns that have a value of greater than 0.5 to any training pattern are assessed as being similar.

[12]To establish a comparison benchmark, a forecast based on a simple four-period moving average is made also during these 41 periods. The moving average method predicts the correct price change of a S&P contract 59% of the time, and a Gold contract 51% of the time.

to Gold price changes being more saw-toothed than S&P price changes, which show a persistent, long-term upward drift.[13]

To evaluate forecast accuracy further, Weiss and Kulikowski (1991) and Eberhart et al. (1990) suggest a contingency table approach. The results of Table I exhibit a significant difference in forecast performance conditioned on the direction of change in the monthly price of a S&P futures contract. Specifically, when a positive price change occurs, the network forecasts the price change correctly 89% of the time. When the price change of a S&P contract is negative, however, the network is able to forecast the decrease correctly less than half the time. This significant difference in forecast performance relative to the sign of the monthly price change is not evident from the Gold contingency results in Table II (i.e., 10/17 positive changes versus 15/24 negative changes).

To determine the gains achieved by the neural networks, a simple trading strategy of fixed transaction volumes[14] is followed. Based on the networks' forecasts, total and average per trading period gains for a single S&P futures contract are $55,055 and $1,343, and $17,525 and $427 for a single Gold futures contract.[15] To compute a return on investment per trading period, R_i, an initial margin[16] of 0.05 of the opening futures contract price is assumed. It is further assumed that transaction costs per contract traded consist of a roundturn commission cost of $50 and a market impact cost of $50.

The 41 trades of a S&P and Gold futures contract results in an average per period (cumulative) return on investment of 17.04% (698%) and 16.36% (670%),[17] respectively. For these markets, the return per trading period[18] is depicted by bars and the cumulative return is represented by a line in Figures 3 and 4.

Table I
S&P FORECAST ACCURACY

	Predicted Price Change		
	Positive	**Negative**	
Actual price change			
Positive	25	3	28
Negative	7	6	13
	32	9	41

[13]There is no way to adjust for drift because of the shortness of the data series (i.e., a neural network is being trained to make individual forecasts of price changes on four months of data over 15 patterns), and the fact that the network has no way of knowing one pattern is related (in time) to any other pattern it sees. Said another way, the same neural network forecasts would be obtained if the patterns representing the most recent 15 months had been presented to the network in random instead of chronological order.

[14]An alternative is to adopt a variable transaction volume, which might be proportional to the size of the forecasted price change.

[15]Comparatively, forecasts based on a simple four-period moving average result in average and total per-trading periods gains (losses) for a single S&P futures contract of ($150.12) and ($6155), and $162.50 and $6662.60 for a single Gold futures contract.

[16]It is assumed as in Lukac et al. (1988) and Turner et al. (1992) that the initial margin consists of Treasury securities. Trades are simulated for the same contract and maintenance margins are not considered because all positions are liquidated completely each month.

[17]The comparable simple moving average forecaster for Gold results in an average per-period and cumulative return on investment of 2.88% and 118.13%.

[18]Note the extremely large loss in period 21 of Exhibit 3 associated with the October, 1987, "meltdown."

Table II
GOLD FORECAST ACCURACY

	Predicted Price Change		
	Positive	**Negative**	
Actual price change			
Positive	10	7	17
Negative	9	15	24
	19	22	41

To adjust the returns achieved by the neural networks for risk, the following model from Lukac et al. (1988) is used:

$$E(R_i) = a + BR'_m + e_i$$

where $E(R_i)$ is the return from a trade of a S&P and Gold future in period i, R'_m equals $E(R_m - R_f)$, where R_m represents the returns to the S&P Stock Index as computed by Ibbotson Associates (1991), and R_f is the reported 30-day U.S. Treasury-bill return. Following the procedure of Turner et al. (1992), who assumed that the capital invested in margins and held for margin calls could have been simultaneously invested in 30-day U.S. Treasury-bills, a risk-free rate is not subtracted from $E(R_i)$, and, hence, R'_m in the above equation is represented by R_m.

If returns above risk are earned from trades initiated by the neural networks, the regression intercept term should differ significantly from zero (i.e., $a > 0$). As can be seen from Table III, while trades of both S&P and Gold futures yield positive returns above risk, only the Gold intercept term is found to be statistically significant based on a one-tailed "t" test. Thus it appears that when the Jensen (1968) test is applied to neural network trades, only the returns from Gold futures are significantly above a return to risk.

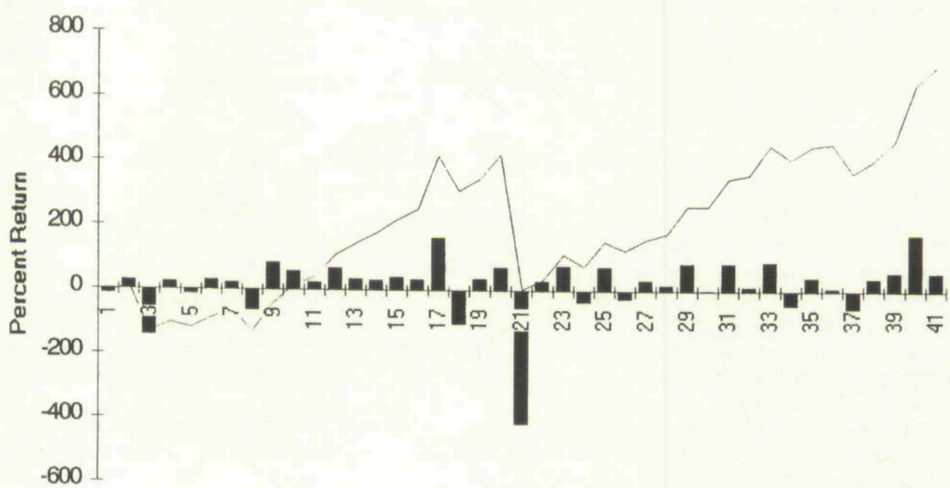

Figure 3
Return on investment of S&P futures trades.

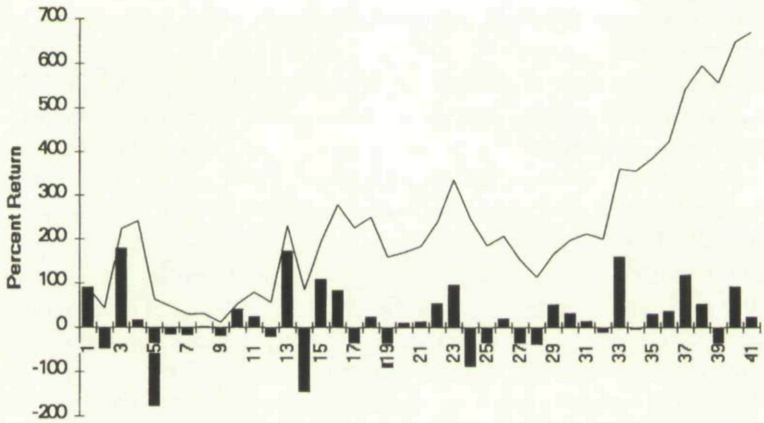

Figure 4
Return on investment of Gold futures trades.

SUMMARY

The research examines the feasibility of employing neural networks to forecast monthly price changes of S&P and Gold futures. Input to the networks consist of monthly price changes and measures of price volatility of S&P and Gold futures, the aggregate supply of money, and commitments of major trading groups in each of these futures markets. The input covers a period of 90 months.

On the basis of a preliminary analysis of the data, it is determined that forecast parameters for the networks can be derived by relying on a relatively short history of 15 months of patterns. This means that the forecast performance of the networks could be tested and evaluated over a period of 75 months.

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. Trading gains during the 75-month period, exclusive of transaction costs, are $55,055 for a single S&P futures contract and $17,525 for a single Gold futures contract. 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. Moreover, while neural network trades of both S&P and Gold futures yield positive

Table III
TEST OF EXCESS OVER MARKET S&P AND GOLD RETURNS

	Intercept (*a*) (Std. Error)	**Slope (*B*) (Std. Error)**	R^2
S&P	6.861	9.487[a]	0.323
	(12.361)	(2.201)	
Gold	16.671[b]	1.558	0.014
	(12.177)	(2.061)	

[a]Significant at the 0.001 level.
[b]Significant at the 0.10 level.

returns above risk, only the Gold returns from neural network trades are statistically above a return to risk.

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. First, derivation of the parameters of the networks relies heavily on published commitments data as a surrogate for the expectations of major trading groups. It is noted that this reliance limits the application of the methodology to the 50 or so futures for which the CFTC publishes data.

A second important factor is the selection of futures whose prices are relatively insulated from natural phenomena such as floods and pestilence. If other futures, such as orange juice and heating oil, had been chosen for study, lower rates of forecast accuracy of their price change would likely have resulted.

Preliminary work in determining an appropriate length for the training period is the third important factor that contributes to the success of the research. This avoids the potential danger of deriving network parameters based on patterns that are no longer relevant in representing market behavior. It is found that a relatively short training period is able to capture changes over time, both in the composition and conduct of the major trading groups.

Last, the research is aided by trading selectively rather than in every period. Selective trading is facilitated by employing the classification ability of the networks, and linking the decision to trade to the similarity exhibited by the most recent forecast pattern to previous data patterns.

Appendix

An artificial neural network is a computer-based, highly interconnected computational model with many simple individual processing elements or nodes arranged in layers. Nodes in one layer are fully connected to all nodes in an adjacent layer.

In general, a node contains a summation function, which totals, through the application of weights, incoming signals from other connected nodes. A node also has a transfer function, which determines the strength with which the summed signal will be transmitted to other nodes in the next connected layer.

A simple three layer neural network is illustrated in Figure A-1. Each node is represented by a circle and each interconnection, with its associated weight, by an arrow. The nodes labeled b are bias nodes, which produce constant values, and enable the construction of intercept weights.

Nodes in the input layer receive input, and nodes in the output layer provide output. Nodes in the middle layer receive signals from the input layer nodes and pass signals to output layer nodes. Because the output of these nodes is not directly observable, their layer can be thought of as hidden.

Let the subscripts i, j, and l refer to the input, hidden, and output layers, respectively. Further, let i represent input, o represent output, and w stand for a connection weight. Signals in the network feed forward (i.e., flow from left to right). The signal presented to a hidden layer node in the network is the output value of the input node times the value of a hidden layer node's connection weight. The net input to the hidden node is equal to the sum of all connections into it. This value is described in eq. (A-1).

$$i_j = \sum_i w_{ji} o_i \tag{A-1}$$

The output of a hidden node is transformed according to a sigmoid function so that its value falls between 0 and 1. Eq. (A-2) shows this transformation.

$$o_j = 1/(1 + \exp(-i_j)) \tag{A-2}$$

Once the outputs of all hidden layer nodes are calculated, the net input to output layer nodes is calculated in an analogous manner, as described in eq. (A-3). Similarly, the output of each output layer node is transformed according to eq. (A-4).

$$i_l = \sum_i w_{lj} o_j \tag{A-3}$$

$$o_l = 1/(1 + \exp(-i_l)) \tag{A-4}$$

To summarize, the feedforward process entails performing two operations per hidden and output layer nodes. The first operation sums the output of the previous layer nodes times the interconnecting weights, and the second operation functionally transforms the output.

Instances of the input called patterns, p, usually are divided into training and testing sets. Generally, training a neural network involves presenting the inputs and correct outputs or target values, t_l, of each training pattern over and over again to the network. The network adapts its weights so that its outputs mirror the target values of the training patterns. The neural-network term for this adaptation process is learning and is discussed next.

An error term measures how well a network is trained. The term represents the difference between an output target value and the value a node actually has as a result of feedforward calculations. The error term is calculated for a given pattern and summed over all output nodes for that pattern. Then the grand total is divided by the number of patterns to give the following average sum-squared value:

$$E_p = \sum (t_{pl} - o_{pl})^2 \tag{A-5}$$

The goal of training is to minimize the average sum-squared error over all training patterns. The process of adapting the node weights to achieve this objective is called backpropagation.

Recall that by eq. (A-4) the output of a node in the output layer is a function of its input, or $o_l = f(i_l)$. The first derivative of the function $f'(i_l)$ is called the error signal

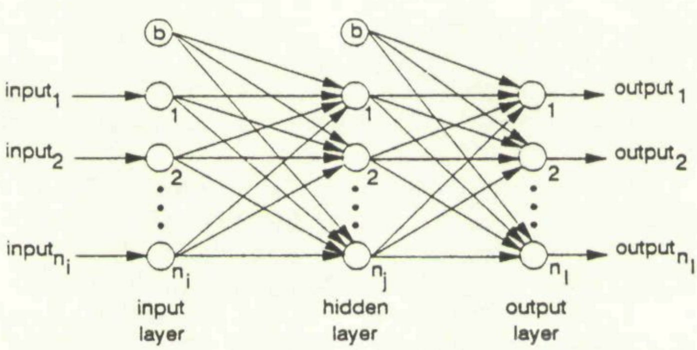

Figure A-1
A three-layer neural network.

δ_l and defined by eq. (A-6) for output layer nodes.

$$\delta_l = f'(i_l)(t_l - o_l) \tag{A-6}$$

The first derivative of the sigmoid transfer function of eq. (A-4) is $o_l(1 - o_l)$. Eq. (A-7) represents the output layer error signal, calculated for each node.

$$\delta_l = (t_l - o_l)o_l(1 - o_l) \tag{A-7}$$

This error value must be propagated back through the output layer nodes and the network must perform appropriate adjustments to weights of the nodes. Specifically, using a learning coefficient η and a momentum factor α, eq. (A-8) describes how the weights feeding the output layer nodes are updated.

$$w_{lj}(\text{new}) = w_{lj}(\text{old}) + \eta\delta_l o_j + \alpha[\Delta w_{lj}(\text{old})] \tag{A-8}$$

where Δw(old) represents the weight change on the previous training iteration.

What is being applied in eq. (A-8) is weight adaptation according to the gradient descent method. The idea of gradient descent is to make a change in the weight that is proportional to the negative of the derivative of the error, as measured on the current pattern, with respect to each weight.

For the hidden layer nodes, the error term is calculated as

$$\delta_j = f'(i_j)\sum w_{lj}\delta_l \tag{A-9}$$

and eqs. (A-10) and (A-11) are equivalent to eqs. (A-7) and (A-8):

$$\delta_j = o_j(1 - o_j)\sum w_{lj}\delta_l \tag{A-10}$$

$$w_{ji}(\text{new}) = w_{ji}(\text{old}) + \eta\delta_j o_i + \alpha[\Delta w_{ji}(\text{old})] \tag{A-11}$$

In summary, for each pattern in the training set the network first calculates the error terms for each output layer node using eq. (A-7) and then for each hidden layer node using eq. (A-10). It then sums the error terms, and, after all patterns have been presented once, adjusts the weights of the nodes according to eqs. (A-8) and (A-11). Learning is complete and the weights in the neural network are no longer modified once the sum of the error terms reaches zero.

Bibliography

Band, R. E. (1986): *Contrary Investing.* New York: Viking Penguin.

Blank, S. C. (1991): "'Chaos' in Futures Market? A Nonlinear Dynamical Analysis," *The Journal of Futures Markets,* 11:711–728.

Collard, J. E. (1991): "A B-P ANN Commodity Trader," in *Advances in Neural Information Processing Systems 3,* Lippmann, R. P., Moody, J. E., and Touretzky, D. S. (eds.). San Mateo, CA: Morgan Kaufman Publishers.

Cousin, E. J. (1990): *Investment Strategy and the Money Connection.* New York: Wiley.

DeCoster, G. P., Labys, W. C., and Mitchell, D. W. (1992): "Evidence of Chaos in Commodity Futures Prices," *The Journal of Futures Markets,* 12:291–305.

Eberhart, R. C., and Dobbins, R. W. (1990): "Introduction," in *Neural Network PC Tools,* Eberhart, R. C., and Dobbins, R. W. (eds.). San Diego, CA: Academic Press.

Eberhart, R. C., Dobbins, R. W., and Hutton, L. V. (1990): "Performance Metrics," in *Neural Network PC Tools,* Eberhart, R. C., and Dobbins, R. W. (eds). San Diego, CA: Academic Press.

Frank, M., and Stengos, T. (1989): "Measuring the Strangeness of Gold and Silver Rates of Return," *Review of Economic Studies,* 56:553–567.

Hadady, R. E. (1987): "Using Open Interest to Analyze Price Trends," *Futures,* 16(7):59–61.

Hadady, R. E., Finberg, I. L., and Rahfeldt, D. (1987): *Winning with the Insiders.* West Palm Beach, FL: Weiss Research.

Harston, C. T. (1990): "Business with Neural Networks," *Handbook of Neural Computing Applications,* Maren, A. J., Harston, C. T., and Pap, R. M. (eds.). San Diego, CA: Academic Press.

Hecht-Nielson, R. (1990): *Neurocomputing.* Reading, MA: Addison-Wesley.

Herbst, A. F., and Maberly, E. D. (1990): "Stock Index Futures, Expiration Day Volatility, and the 'Special' Friday Opening: A Note," *The Journal of Futures Markets,* 10:323–325.

Jensen, M. L. (1968): "The Performance of Mutual Funds in the Period 1945–64," *Journal of Finance,* 23:389–416.

Kaufman, P. J. (1987): *The New Commodity Trading Systems and Methods.* New York: Wiley.

Ibbotson Associates (1991): *Stocks, Bonds, Bills and Inflation.* Chicago: Capital Market Research Center.

Leuthold, R. M., Junkus, J. C., and Cordier, J. E. (1989): *The Theory and Practice of Futures Markets.* Lexington, MA: Lexington Books.

Lukac, L. P., and Brorsen, B. W. (1989): "The Usefulness of Historical Data in Selecting Parameters for Technical Trading Systems," *The Journal of Futures Markets,* 9:55–65.

Lukac, L. P., Brorsen, B. W., and Irwin, S. H. (1988): "A Test of Futures Markets Disequilibrium using Twelve Different Technical Trading Systems," *Applied Economics,* 20:623–639.

McClelland, J. L., and Rumelhart, D. E. (1988): *Explorations in Parallel Distributed Processing.* Cambridge, MA: MIT Press.

Mendelsohn, L., and Stein, J. (1991): "Fundamental Analysis Meets the Neural Network," *Futures,* 20(10):22–24.

Pliska, S. R., and Shalen, C. T. (1991): "The Effects of Regulations on Trading Activity and Return Volatility in Futures Markets," *The Journal of Futures Markets,* 11:135–151.

Shadmehr, R., and D'Argenio, D. Z. (1990): "A Comparison of a Neural Network Based Estimator and Two Statistical Estimators in a Sparse and Noisy Environment," *International Joint Conference on Neural Networks,* II:289–292.

Sharda, R., and Patil, R. B. (1990): "Neural Networks as Forecasting Experts: An Empirical Test," *International Joint Conference on Neural Networks,* II:491–494.

Stein, J. (1989): "How Commodity Markets Fit Business Cycle Patterns," *Futures,* 18(13):34–38.

Teweles, R. J., and Jones, F. J. (1987): *The Futures Game* (2nd ed.). New York: McGraw-Hill.

Turner, S. C., Houston, J. E., and Shepard, T. L. (1992): "Markov Processes in Soybean Futures Trading," *The Journal of Futures Markets,* 12:61–74.

Tvede, L. (1992): "What 'Chaos' Really Means in Financial Markets," *Futures,* 21(2):34–36.

Weiss, S. M., and Kulikowski, C. A. (1991): *Computer Systems that Learn.* San Mateo, CA: Morgan Kaufmann Publishers.

White, H. (1988): "Economic Prediction using Neural Networks: The Case of IBM Daily Stock Returns," *IEEE International Conference on Neural Networks,* II:451–458.

Wilson, W. W., Fung, H. G., and Ricks, M. (1988): "Option Price Behavior on Grain Future Markets," *The Journal of Futures Markets,* 8:47–65.

Zaremba, T. (1990): "Technology in Search of a Buck," in *Neural Network PC Tools,* Eberhart, R. C., and Dobbins, R. W. (eds.). San Diego, CA: Academic Press.