

# NCCC-134

APPLIED COMMODITY PRICE ANALYSIS, FORECASTING AND MARKET RISK MANAGEMENT

## **Futures Trading with a Neural Network**

by

Lonnie Hamm, B. Wade Brorsen, and Ramesh Sharda

Suggested citation format:

Hamm, L., B. W. Brorsen, and R. Sharda. 1993. "Futures Trading with a Neural Network." Proceedings of the NCR-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management. Chicago, IL. [<http://www.farmdoc.uiuc.edu/nccc134>].

## FUTURES TRADING WITH A NEURAL NETWORK

Lonnie Hamm      B.Wade Brorsen      Ramesh Sharda\*

Much interest has been generated by a new class of computer programs called neural networks. Neural networks have been successfully applied in fields as diverse as task coordination, optical pattern recognition and sonar signal processing. Many people believe that neural networks hold great promise for predicting financial variables. A recent article in *U.S. News and World Report* states that most financial organizations are experimenting with neural networks, but only a few have actually put them to use (Egan). Much of the excitement surrounding neural networks is due to their unique ability to handle nonlinear data. Neural networks are universal approximators capable of approximating any nonlinear function (White). This means the functional form of the model need not be made explicit.

The last couple of years has seen a great deal of interest in applying neural networks to commodities trading. The popular press is filled with claims about the usefulness of neural networks in trading, but little academic research is available to support these claims. In two papers, Collard reported promising results using a neural network as a commodity trading model. A study by Bergerson and Wunsch used the hindsight of a market technician (chartist) to train a neural network when to buy and sell a commodity. They then used the network along with some traditional money management techniques and obtained good results. However, all of these studies were inconclusive because of a short sample period, as well as the absence of any statistical tests on the profits produced by the trading models.

The research reported in this paper provides a rigorous tests of whether a neural network can be used to develop a profitable commodity trading model. A trading model is built which uses a neural network to produce trading signals. The neural network combines technical indicators in a nonlinear fashion to produce the trading signals. Simulated trading returns are generated by using this model to trade the soybean contract. The null hypothesis of zero net and gross returns is tested using the bootstrap method of Monte Carlo hypothesis testing (Noreen).

## TECHNICAL ANALYSIS

Since the neural networks examined use only past prices, they are a form of technical analysis. Technical analysis is an integral part of the trading decisions of many speculators in futures markets. Brorsen and Irwin report that 80 percent of commodity investment pools use computerized technical trading systems. Two highly regarded books by Schwager (1989, 1992) give the trading testimonials of a large number of traders who have used technical analysis to produce significant trading profits over a long period of time.

In spite of the widespread use of technical analysis, the academic community has been less than receptive to its concepts. For example, Malkiel stated:

---

\*Hamm and Brorsen are M.S. degree candidate and Professor respectively in the Department of Agricultural Economics at Oklahoma State University. Sharda is Professor of Management Science and Information Systems in the Department of Management at Oklahoma State University.

Obviously, I am biased against the chartist. This is not only a personal predilection, but a professional one as well. Technical analysis is anathema to the academic world. We love to pick on it. Our bullying tactics are prompted by the two considerations: (1) the method is patently false; and (2) it's easy to pick on. And while it may seem a bit unfair to pick on such a sorry target, just remember: it is your money we are trying to save.

Traditionally, academics have pointed to the weak form of the efficient market hypothesis, proposed by Fama, which says that prices, both past and present, fully reflect all available information. If futures market are weak form efficient, then technical trading systems should not on average be profitable. However, academics are beginning to change their views of technical analysis in response to a growing body of evidence which suggests that technical analysis is of some value in predicting commodity prices.

Lukac and Brorsen simulated trading of 23 technical systems on 30 different markets and found that 22 of the systems produced statistically significant net returns. Boyd and Brorsen tested five technical trading systems on seven commodities and found that significant annual net returns were possible. Irwin, Krukemyer, and Zulauf found that returns from public commodity pools were sufficient to cover the costs and risks involved in futures trading. In addition, excess net returns significantly greater than zero were found for institutional commodity pools. The research reported in this paper will provide further empirical evidence about the usefulness of technical analysis as well as a test of the usefulness of a neural network in developing trading systems.

Disequilibrium theory, as well as noisy rational expectations theory, offers an explanation of why technical trading systems may work. Disequilibrium theory argues that markets do not immediately adjust to information shocks because of market frictions, such as the cost of acquiring information. As a result, markets may sluggishly adjust to new information. Profitable trading systems would then be possible because of the price trends that would exist as a result of the adjustment process caused by information shocks (Beja and Goldman, Nawrocki). If market participants are not heterogeneously informed of new or difficult to obtain information, then prices tend to be an imperfect aggregator of information. Using a two period noisy rational expectations model, Brown and Jennings claim that as a result of this imperfect aggregation, the current price is not a sufficient statistic for private information possessed by market participants. As a result, historical prices add information that is not available with the current price alone. That is, technical analysis provides additional information to market participants forming an expectation of future prices.

Researchers have begun to argue that asset prices exhibit nonlinear dynamics (Brockett, Hinich, and Patterson). Neftci argued that if price dynamics are nonlinear, technical analysis may be capturing information contained in higher-order moments of asset prices. This information would not be captured by traditional linear models. Brock, Lakonishok, and LeBaron examined the stock returns generated from some technical trading rules. Their results suggest that the return-generating process of stocks is more complicated than the returns suggested by linear models.

The discussion above suggests that the adjustment process in prices, as referred to in disequilibrium theory, may be nonlinear. Brown and Jennings claimed that past prices, i.e. technical analysis, are an important element in determining future prices. If the underlying process generating commodity returns is chaotic, then a neural network, using past prices, should produce statistically significant returns.

## NEURAL NETWORKS

The human brain is a marvel of nature, for many tasks it is superior to the most complex supercomputer. The human brain is especially adept at processing visual information; recognizing objects, faces, and so on. Recent developments have allowed researchers to construct artificial neural networks. An artificial neural network is a simplified model of the way a collection of brain cells, called neurons operate. Neural network is the term used to refer to a broad class of computer programs which simulate the behavior of neurons. In this way, the neural network is given the ability to learn from experience, develop rules, and recognize patterns in data. A chartist (technical analyst) looking at price charts draws from past experience and uses the brain's ability to recognize patterns. The neural network is purported to have this ability.

The neural network used here is similar to regression in that it predicts some output given a set of inputs. The neurons of a neural network are usually arranged in layers: an input layer, an output layer, and one or more hidden layers. The input neurons in a neural network correspond to the independent variables of a regression whereas the output neurons correspond to the dependent variables in a regression. A direct comparison between the hidden layer(s) in a neural network and some element of a regression is not possible. The hidden layer(s) is what allows a neural network to make generalizations and classifications.

The knowledge of a network is stored in the pattern of variable interconnection weights which connect the neurons in different layers. Learning takes place when these weights are adjusted. The adjustment process can be done in two main ways: supervised learning and unsupervised learning. In supervised learning, the weights are adjusted on the basis of a comparison between the networks output and the known correct answer. Unsupervised learning is used when the correct output is only defined in terms of the correlations of the input data or signals. Supervised learning is the most common training procedure, and is the type used in this research

Many training algorithms for supervised learning have been proposed, however, the most commonly used is back-propagation. In back-propagation, the connection weights are adjusted by an iterative training algorithm which seeks to minimize the sum of squared errors. Back-propagation minimizes the squared errors using a gradient descent algorithm. Back-propagation is the training algorithm used in this paper. The back-propagation training procedure, as well as the functions of each layer in a neural network is explained in the discussion following. Figure 1 provides a visual reference for the following discussion.

The back-propagation training algorithm involves two steps. In the first step, the network is supplied a set of inputs which correspond to a set of outputs. For a neural network using the supervised learning procedure, the first step can happen in one of two ways. In feed-forward networks the input to the network at some time  $t$ , is processed by each layer in a sequential order, beginning with the input layer, proceeding to the hidden layer(s), and finally to the output layer. Feed-back or recurrent networks, include one or more direct or indirect loops or connections, so the network has information on what it has learned in previous training runs. A feedback network can be somewhat analogous to regression models that are adjusted for autocorrelation. Feed-forward is the most popular, although feed-back networks are gaining support. Feed-forward is the architecture used in this paper.

Each neuron in the input layer contains the value, at time  $t$ , of one of the inputs to the network. The neurons in the input layer serve only as an input terminal for the inputs to the network. Each neuron  $i$  in the input layer is connected with each neuron  $j$  in the hidden layer via a weight  $w_{i,j}$ . Each hidden neuron  $j$  contains a weighted average of the values in the input layer:

$$(1) \quad h_j^t = \sum_i w_{ij} \xi_i^t$$

where  $h_j^t$  is the value of hidden neuron  $j$  at time  $t$   
 $w_{ij}$  is the connection weight between input  $i$  and hidden unit  $j$   
 $\xi_i^t$  is the value of input  $i$  at time  $t$ .

The output of each hidden neuron  $j$  at time  $t$  is

$$(2) \quad V_j^t = f(h_j^t)$$

where  $h_j^t$  is from (1)  
 and  $f$  is the nonlinear function

$$(3) \quad f(x) = \frac{1}{(1 + \exp(-x))}.$$

The function in equation (3) is referred to as the transfer function. The function scales the output from each neuron nonlinearly between 0 and 1. Each hidden neuron  $j$  is then connected to each output neuron  $k$  by the connection weight  $\eta_{jk}$ , where  $j$  is the hidden neuron, and  $k$  is the output neuron. The output of the network from output neuron  $k$  at time  $t$  is

$$(4) \quad O_k^t = f\left(\sum_j \eta_{jk} V_j^t\right)$$

where  $f$  is the function given in (3).

Note that in the case of more than one hidden layer, the output from the first hidden layer would first be processed by the next hidden layer, before being processed by the output layer. The output  $O_k^t$  from (3) is compared with the correct output at time  $t$  to compute an error term.

The second step in back-propagation is a backward pass through the network in which the error is used to adjust the connection weights  $\eta_{jk}$  and  $w_{i,j}$  using the back-propagation training algorithm. These weights are the only parameters in the network. The details of the back-propagation training algorithm are explained by Hertz, Krogh, and Palmer.

#### NEURAL NETWORK TRADING MODEL

A trading model is built which uses a neural network to produce trading signals. Simulated trading returns are generated using this model to trade the soybean contract. The neural network is a feed-forward type network using the back-propagation training algorithm. The inputs to the neural network consist of moving averages, oscillators, and a trend indicator. Specifically, the trading model includes 10 technical inputs: 6 moving averages, 2 oscillator indicators, and 2 trend indicators.

Typically, moving averages perform best in trending markets while oscillators perform best in a trading range market. The neural network uses the trend indicator input to give more emphasis to the moving averages or oscillators depending on the trending behavior of the market.

The moving average inputs consist of six moving averages: 5-day, 10-day, 15-day, 20-day, 30-day, and 60-day. A moving average of length  $n$  is

$$(5) \quad M_t(n) = \frac{\sum_{i=0}^n P_{t-i}}{n}$$

where  $M_t(n)$  = moving average of length  $n$  at time  $t$   
 $P_t$  = closing price at time  $t$ .

Buy or sell signals from traditional moving average systems are given when a shorter term moving average crosses over a longer term moving average, and/or the price of the commodity crosses over some moving average of the price. To represent this type of information to the network, the moving averages will be preprocessed. The raw moving average series is transformed into a ratio of the current closing price to the moving average

$$(6) \quad I_{n,t} = \frac{P_t}{M_t(n)}$$

where  $P_t$  = closing price at time  $t$   
 $M_t(n)$  = the moving average of length  $n$  at time  $t$ .

The oscillator type indicators measure the relative position of the closing price within the daily range. These type of oscillators are commonly referred to as stochastics. It is claimed that when the market is at a turning point, the intraday highs will be higher than the previous high, but the closing price will settle nearer the low (Kaufman). The two stochastics used for the trading model are the %K-slow and %D-slow. The initial %K is given as

$$(7) \quad \%K_t = 100 * \frac{C_t - L_t(5)}{R_t(5)}$$

where  $C_t$  is the closing price at time  $t$   
 $L_t(5)$  is the lowest intraday low of the last 5 days  
 $R_t$  is the highest intraday high minus the lowest intraday low of the last 5 days.

the %K-slow is a three day moving average of the %K and the %D-slow is a three day moving average of the %K-slow. These indicators can range from a low of 0 to a high of 100. Typically a buy (sell) signal is given when the %D reaches an extreme high (low) such as 80 (20). The buy or sell signal is confirmed when the %K line crosses the %D line.

The trend indicator used in the trading model is the average directional movement index (ADX). The ADX varies from a low of 0 to a high of 100. The higher the number the stronger the trend. Murphy suggests that a reading below 20 indicates a nontrending environment and therefore an oscillator type indicator, as opposed to moving averages, would perform better in this environment. Whether the ADX is rising or falling is also important, therefore, the 5-day slope of the ADX line is also included as an input to the neural network. The details of the ADX are explained by Kaufman.

Most neural networks are designed to perform best when inputs are in the range of 0 to 1. The reason for this is related to the nonlinear transfer function given in (3), which bounds the output from each hidden neuron to the range of 0 to 1. Before the inputs are used to train the neural network, they will be normalized or mapped into the range of 0 to 1. Many neural network software packages normalize input data automatically, which is the case with the software used here.

The training pattern outputs of the neural network consist of a 0 or 1, 0 representing a sell signal and 1 representing a buy signal. The buy and sell signals are based upon the current price relative to the price 5 trading days in the future. The trading signal  $S_t$  at time  $t$  is

$$(8) \quad S_t = \begin{cases} 1 & \text{if } P_{t+5} \geq P_t \\ 0 & \text{if } P_{t+5} < P_t \end{cases}$$

where  $P_t$  is the closing price at time  $t$ .

One concern with neural networks is their tendency to memorize or overfit the data. If the network memorizes the data it will fail to make generalizations about the relationships between the inputs and output signal. The network would then perform poorly in out-of-sample tests. Presenting training patterns in chronological order to the network may encourage memorization of the facts. Markets contain trends, therefore if training patterns are presented to the network chronologically, any particular training pattern will not be independent of the preceding training pattern. Randomizing the order of the training patterns prior to training easily solves this problem.

Neural networks will also tend to memorize the facts if the network is trained too long and/or there are too many hidden neurons relative to the number of inputs and training patterns. To prevent this problem, 15% of the training sample is randomly selected to form a preliminary test sample. The network is only trained on 85% of the training sample and 15% is left out for preliminary testing. Therefore, the preliminary test sample is drawn from the training sample prior to training the network. As the network is training, it is periodically tested on the preliminary test sample to check if the network is generalizing, or testing well on a pattern the network has not been trained on. The training run which performs the best on the preliminary testing sample is then selected to simulate trading on the out-of-sample data sample.

Most neural network programs allow the user to select some convergence criterion. Generally, this criterion is set so that the network stops training when most of the outputs from the network is within a certain percentage of the correct output. This percentage is referred to as the training tolerance. For this research, the training tolerance is set at .1. Since the output is binary, a training tolerance of .1 implies an output is "good" if it is greater than or equal to .9 for a buy signal, and less than or equal to .1 for a sell signal. This results in a stochastic model. Since training is continued only as long as the network is generalizing well, the majority of the training samples will not meet this tolerance. The neural network was chosen to have one hidden layer and fifteen neurons in the hidden layer. The neural network program used is Brainmaker, from California Scientific Software.

#### DATA

One difficulty in dealing with futures price series is that they are discontinuous. Each commodity is represented by several different futures prices representing different months of delivery. Because of liquidity costs, most commodity pool operators hold the majority of their positions in the "nearby" contract (Brorsen and Irwin). Therefore, it has become common practice to construct a price series using the contract closest to expiration. The process of constructing a continuous price series for this research is described below.

The price series is constructed using daily changes in the price of the nearby contract. The daily changes are then added to the beginning price of the series to form an "artificial" price series. The absolute level of the artificial prices may bear no resemblance to the actual price level, but the artificial price series is accurate in representing the daily price changes of the nearby contract. A position is held in the nearby contract until the 20th of the month before expiration of the nearby contract. This is called the rollover date. If the rollover date falls on a weekend or holiday, the next trading day will be used as the rollover date. The daily changes of the old and new contracts, on the rollover date, are spliced together in such a way so as to adjust for the discontinuity in price between the contract month being rolled out of and the contract month rolled into. The daily price change of the new contract on the rollover date is used in place of the daily change of the old contract. The price changes surrounding the rollover date  $t$  are

$$(9) \quad \{ \dots, P_{t-2}^O - P_{t-1}^O, P_{t-1}^O - P_{t-2}^O, P_t^N - P_{t-1}^N, P_{t+1}^N - P_t^N, \dots \}$$

where  $P^O$  = the price of the old contract month  
 $P^N$  = the price of the new contract month.

The data used for this study consists of the open, high, low and close of the soybean contract from January 1975 through December 1989. The source of the data is the data vendor Technical Tools. The neural network is trained on two separate training samples, each of which has a corresponding testing sample. The network is first trained on a training set consisting of data from January 1975 through December 1981. Out-of-sample simulated trading is performed using a test sample which consists of the time period from January 1982 through December 1985. The neural network is then retrained on a training set consisting of data from January 1979 through December 1985. Out-of-sample simulated trading is then performed with the time period from January 1986 through December 1989.

#### TRADING RULES AND ASSUMPTIONS

Since the network produces output in the range of 0 to 1, Buy and sell rules are used whereby the trading model will go long (short) when the output is greater (less) than the buy (sell) rule number. Five pair of numbers representing buy (sell) rules for five trading models are: model 1, .5 (.499); model 2, .6 (.4); model 3, .7 (.3); model 4, .8 (.2); model 5, .9 (.1). Theoretically, the output from the network should be making some probability statement about the likelihood of a winning trade. Therefore, it would be expected that buying (selling) when the network's output is high (low) would result in more profitable trades. A long position is liquidated when the networks output falls below .5 and a short position is liquidated when the networks output rises above .5. Neutral positions are possible using the trading rules. Since the closing prices are used to create the input to the neural network, the opening price of the day following the trading signal is used as the entry or exit price.



In this study, transaction costs are composed of the commission costs and the skid error, or bid-ask spread, of each trade. Trades can be executed through any one of a number of discount futures brokers for no more than \$35 per round-turn trade. An additional \$65 is added to the transaction cost to account for skid error. This leaves transaction costs of \$100 per round-turn trade. Transaction costs will also be incurred on rollover dates when a trade is rolled over into the next contract month.

#### EVALUATION PROCEDURE

Mean daily Returns (MDR) are computed for the five trading models. Since the trading models are in the market only on selective trading days, MDR is the mean only of days when a position is held. The daily returns are the actual dollar returns, assuming one contract is being traded. The bootstrap method of Monte Carlo Hypothesis testing is used to test the hypotheses

$$\begin{aligned} H_0: \text{MDR} &\leq 0 \\ H_a: \text{MDR} &> 0 \end{aligned}$$

and

$$\begin{aligned} H_0: \text{MDR} &\leq \text{average transaction costs} \\ H_a: \text{MDR} &> \text{average transaction costs} \end{aligned}$$

where MDR is the mean daily returns without transaction costs and average transaction costs is defined as

$$(10) \quad \text{ATC} = \frac{\text{total transaction costs}}{\text{number of days in the market}}$$

The first hypothesis concerns gross daily returns, while the second hypothesis concerns net daily returns.

A bootstrap sample is drawn which results in a sample size equal to the number of days the trading model is in the market. Long and short positions are randomly assigned to the bootstrap sample. The long and short positions are equal to the number of long and short positions produced by the trading model<sup>1</sup>. The bootstrap mean daily return is computed by replicating the above procedure 1000 times. The test statistic for the null hypothesis of zero gross daily returns is given by (Noreen p. 69)

$$(11) \quad p^g = \frac{[\text{bootstrap means} > \text{neural network mean}] + 1}{\text{bootstrap samples} + 1}$$

where  $p^g$  = the significance level.

The test statistic for the null hypothesis of zero net daily returns is

$$(12) \quad p^n = \frac{[\text{bootstrap means} > (\text{neural network mean} - \text{ATC})] + 1}{\text{bootstrap samples} + 1}$$

where  $p^n$  = the significance level.

#### RESULTS

<sup>1</sup>Choosing long and short positions in this ratio is a more conservative test of trading performance than making the long and short positions equal.

The results of the trading simulation are presented in table 1. The results show that models 1 through 4 generated significantly positive gross returns. Models 1, 2, and 4 also generated significantly positive net returns. Model 3 was marginal with a p value of .053. In the case of model 3, lower transaction costs would result in a significant p value. Model 5 had the lowest mean daily returns. It appears that the network is unable to extract enough information from the inputs to make such a strong probability statement (i.e. .9 and .1) about the direction of the market. This seems reasonable due to the stochastic nature of the neural network.

Training times for the neural networks were long for both training samples. The training time required for the training sample consisting of data from 1975-1981 was approximately 25 hours, while training on the data from 1979-1985 took approximately 35 hours. The neural networks were trained on a 386Dx-25mhz processor with a math coprocessor.

## CONCLUSIONS

Research suggests the underlying process generating commodity returns is nonlinear and possibly chaotic. Neural networks possess the ability to model chaotic processes. The objective of this research was to test whether a useful neural network trading system could be developed to exploit the nonlinear behavior of commodity prices. A neural network trading model was developed which uses information contained in past prices to produce trading signals. These trading signals were used to simulate trading of the soybean futures contract from January 1982 through December of 1989. The bootstrap method of Monte Carlo Hypothesis testing was used to test the following two hypotheses: mean daily returns are greater than zero, and mean daily returns are greater than transaction costs. Trading simulation showed 3 of the 5 trading models produced statistically significant daily returns at the .05 significance level with one trading model marginal with a p value of .053.

The neural network trading model used technical indicators to make predictions of the markets direction, therefore, this research is a test of the efficient market hypothesis. Consistent with much past research, the results here would suggest some alternative model to describe futures prices. Disequilibrium and noisy rational expectations theory offer an alternative to the efficient market hypothesis.

Much evidence exists which suggests that commodity prices exhibit nonlinear dynamics. The neural networks unique ability to model nonlinear processes may be the reason for the statistically positive returns. Many difficulties exist in training a neural network. A neural network with different inputs or different structure, i.e. recurrent, may produce better results, but it appears that a useful neural network trading model has been developed.

## References:

- Beja, Avraham and M. Barry Goldman. "On the Dynamic Behavior of Prices in Disequilibrium." *Journal of Finance* 34(May 1980):235-247.
- Bergerson, Karl and Donald C. Wunsch II. "A Commodity Trading Model based on a Neural Network-Expert System Hybrid." *IEEE International Conference on Neural Networks*, 1991.
- Boyd, Milton S. and B. Wade Brorsen. "Factors Related to Futures Market Disequilibrium." *Canadian Journal of Agricultural Economics* 39(Dec 1991):769-778.
- Brock, William, Joseph Lakonishok and Blake LeBaron. "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns." *The Journal of Finance* 47(Dec 1992):1731-1764.
- Brockett, P.L., M.L. Hinich and D. Patterson. "Bispectral Based Tests for the Detection of Gaussianity and Linearity In Time Series." unpublished manuscript, University of Texas at Austin, 1985.
- Brorsen, B. Wade and Scott H. Irwin. "Futures Funds and Price Volatility." *Review of Futures Markets* 6(2)(1987):118-135.
- Brown, David P. and Robert H. Jennings. "On Technical Analysis." *The Review of Financial Studies* 2(Dec 1989):527-551.
- California Scientific Software. *Brainmaker*. 10024 Newton Road, Nevada City, California 95959.
- Collard, J.E. "Commodity Trading with a Two Year Old." *ORSA/TIMS Meeting* Nashville, 1991.
- Collard, J.E. "A B-P Commodity Trader." *IEEE Conference on Neural Networks*, 1990.
- Egan, Jack. "Artificially intelligent investing." *U.S. News & World Report* 56(March 15, 1993):73
- Fama, Eugene F., "Efficient Capital Markets: A Review of Theory and Empirical Work." *Journal of Finance* 25(May 1970):383-417.
- Grossman, Stanford J. and Joseph E. Stiglitz. "On the impossibility of Informationally Efficient Markets." *The American Economic Review* 70(June 1980):393-408.
- Hertz, John, Anders Krogh and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Redwood City, California: Addison-Wesley Publishing Co., 1991.
- Irwin, Scott H., Terry R. Krukemyer, and Carl R. Zulauf. "Investment performance of Public Commodity Pools: 1979-1990." *Journal of Futures Markets*, forthcoming.
- Kaufman, Perry J. *The New Commodity Trading Systems and Methods*. New York: John Wiley & Sons, Inc., 1991.

- Lukac, Louis P., and B. Wade Brorsen. "A Comprehensive Test of Futures Market Disequilibrium" *Financial Review* 25(Nov 1990):593-622.
- Malkiel, Burton G. *A Random Walk Down Wall Street*. New York: Norton, 1981.
- Murphy, John D. *Technical Analysis of The Futures Market*. New York: New York Institute of Finance., 1986.
- Neftci, Salih N. "Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of 'Technical Analysis'." *Journal of Business* 64(Oct 1991):549-571.
- Noreen, Eric W. *Computer Intensive Methods for Testing Hypothesis: An Introduction*. New York: John Wiley & Sons, 1989.
- Nawrocki, David. "Adaptive Trading Rules and Dynamic Market Disequilibrium." *Applied Economics* 16(Feb 1984):1-14.
- Schwager, Jack D. *Market Wizards*. New York: Harper & Row Publishers Inc., 1989.
- Schwager, Jack D. *The New Market Wizards*. New York: HarperCollins Publishers Inc., 1992.
- Technical Tools. *Continuous Contractor*. 334 State Street, Suite 201, Los Altos California 94022.
- White, Halbert. "Learning In Neural Networks: A Statistical Perspective." *Neural Computation* 4(Aug 1989):425-464.

**Table 1. Performance of Neural Network Trading Model**

Model	Trading Rule <sup>a</sup>		MDR <sup>b</sup>	ATC <sup>c</sup>	NDR <sup>d</sup>	Bootstrap Significance	
	buy	(sell)				(p <sup>g</sup> ) <sup>e</sup>	(p <sup>h</sup> ) <sup>f</sup>
1	.5	(.499)	28.16	7.89	20.27	.011	.048
2	.6	(.4)	28.37	7.49	20.88	.012	.049
3	.7	(.3)	28.09	6.14	21.95	.018	.053
4	.8	(.2)	34.98	5.72	29.26	.007	.035
5	.9	(.9)	27.56	4.60	22.97	.080	.157

<sup>a</sup> If neural network output is above (below) buy (sell) rule then trading model goes long (short). All long (short) positions are liquidated when output falls (rises) below (above) .5.

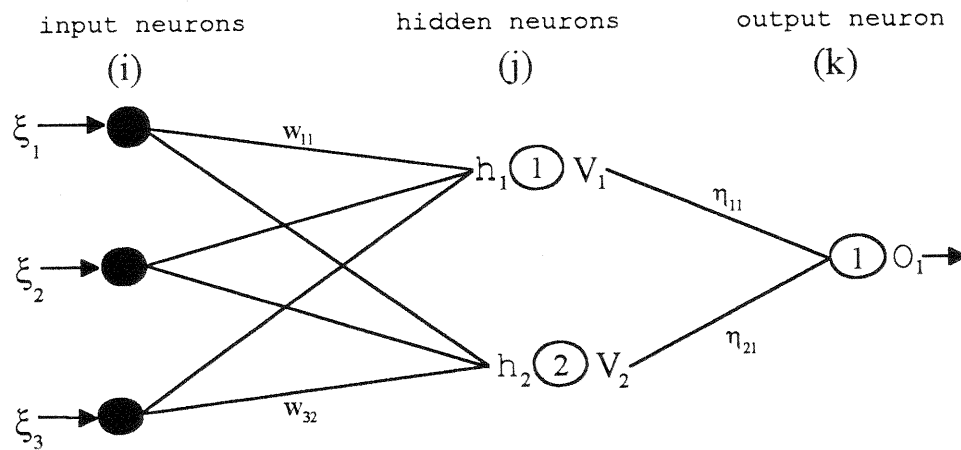
<sup>b</sup> MDR is mean daily returns when the model is in the market.

<sup>c</sup> ATC is average transaction costs.

<sup>d</sup> MDR-ATC.

<sup>e</sup> Bootstrap significance level for  $H_0: \text{MDR} \leq 0$ .

<sup>f</sup> Bootstrap significance level for  $H_0: (\text{MDR}-\text{ATC}) \leq 0$ .

**Figure 1. A Feed-Forward Neural Network With One Hidden Layer**

where 
$$h_j = \sum_i w_{ij} \xi_i$$

$$V_j = f(h_j)$$

$$O_k = f\left(\sum_j \eta_{jk} V_j\right)$$

and 
$$f(x) = \frac{1}{(1 + \exp(-x))}$$