S & P 500 Stock Index' Futures Trading with Neural Networks

Jae Hwa Choi*

신경망을 이용한 S & P 500 주가지수 선물거래

최 재 화*

ABSTRACT

Financial markets are operating 24 hours a day throughout the world and interrelated in increasingly complex ways. Telecommunications and computer networks tie together markets in the form of electronic entities. Financial practitioners are inundated with an ever larger stream of data, produced by the rise of sophisticated database technologies, on the rising number of market instruments. As conventional analytic techniques reach their limit in recognizing data patterns, financial firms and institutions find neural network techniques to solve this complex task. Neural networks have found an important niche in financial applications. We apply neural networks to Standard and Poor's (S & P) 500 stock index futures trading to predict the futures market behavior. The results through experiments with a commercial neural network software do support future use of neural networks in S & P 500 stock index futures trading.

Keywords : neural network, classification/prediction, stock index futures trading.

* Department of Management Dankook University

1. INTRODUCTION

Financial markets are operating 24 hours a day throughout the world and interrelated in increasingly complex ways. Telecommunications and computer networks tie together markets in the form of electronic entities. Financial practitioners are inundated with an ever larger stream of data, produced by the rise of sophisticated database technologies, on the rising number of market instruments. To cope with this information explosion, intelligent systems with quantitative analyses are considered to be the best tool for financial professionals and traders.

As conventional analytic techniques reach their limit in recognizing data patterns, financial firms and institutions find neural network techniques to solve this complex task. Neural networks have found an important niche in financial applications and have recently been applied to finance and investment domain issues (Trippi and Turban 1993, Freedman, Klein, and Lederman 1995, Refenes 1995). Banks and investment firms are deploying neural networks in financial trading arena. Neural network's ability to learn from the past and generalize a model to forecast future prices, adaptability to changing market conditions and ability to incorporate fundamental and technical analysis into a forecasting model are some of the main reasons for its popularity.

Recent advances in neural network technology have facilitated end users in applications that involve prediction and classification schemes (Widrow, Rumelhart, and Lehr 1994). Classifying data has been one of the most widely used capabilities of neural network. Examples include credit approval (Klimasauskas 1991), bankruptcy prediction (Tam and Kiang 1992), stock picking (Yoon, Guimares, and Swales 1994), stock market prediction (Kimoto, Asakawa, Yoda, and Takeoka 1990, Park and Han 1995), and automated trading (Trippi and Desieno 1992). These studies have reported neural networks' superior performance over other statistical techniques. Tam and Kiang (1992) and Salchenberger, Cinar and Lash (1992) compared neural network to discriminant analysis technique. Their results have reported neural network's superior performance over discriminant analysis. Neural networks are being applied to a number of 'live' systems in financial engineering and have shown promising results (Loofbourrow and Loofbourrow 1995). To date, however, only one published work by Trippi and Desieno (1992) has attempted to aid traders in stock index futures trading.

To provide financial analysts with an intelligent system for the futures trading market, we are in the process of developing an intelligent futures trading system. The system integrates market data with modeling and analytical tools to support trader strategies. Modeling and analytical tools include simple analytical techniques, statistical models and neural network models. The integration will be implemented through the rule-based expert system approach with the object technology. Neural network models of the system are discussed in the current paper.

In this paper neural network models are developed for Standard & Poor's (S & P) 500 stock index futures contracts. The neural network developed in this study is similar to that of Trippi and Desieno (1992). Trippi and Desieno (1992) develops six differently configured neural networks using the historical data and combines neural network results with a set of rules to generate a composite recommendation for the current day's position. However, the strategy to mixing neural network results shown in Trippi and Desieno (1992) seems somewhat arbitrary. The purpose of this study is to show that neural networks alone can provide professional traders with useful decision aids.

The rest of this paper is organized as follows. Section 2 describes the process of developing neural network models to predict the market behavior of S & P 500 stock index futures trading. Results of simulated trading with the neural network model are analyzed in Section 3. Conclusions and future research issues are discussed in Section 4.

II. A NEURAL NETWORK MODEL FOR S & P STOCK INDEX FUTURES TRADING

Neural networks are essentially statistical devices for performing inductive inference and are analogous to nonparametric, nonlinear regression models. Nonlinear modeling techniques are the subject of increasing interest from practitioners in quantitative asset management, with neural networks assuming a prominent role. One of the reasons for the great potential for classification and prediction by multilayer neural networks is that neural networks do not require rigid assumptions like normal distribution in the data which is often made in statistical techniques. This is particularly useful in financial engineering applications, where much is assumed and little is known about the nature of the processes determining asset prices(Refenes 1995). We build a neural network prediction and categorization model by finding the behavioral pattern of historical data. The following describes the process of developing a neural network and its performance analysis.

1. Selection of Variables

The choice of input variables is clearly influenced by one's perspective on the market. The selection of variables in this study is primarily determined by a group of professional traders. This selection is completely tilted to technical analysis which suggests the use of only single-market technical data as input. Fundamental data is not used in this research because one of the objectives of this research is to determine the possibility of using neural networks in predicting the future prices based on past prices alone. The problem of finding fundamental data that matches the price data in the correct time sequence is another reason for not considering it.

Inputs to the network are price information, statistics derived from price information, and one subjective market indicator. Price information includes Open, High, Low, Close price data. Statistical indicators include Moving Average (MA), Rate of Change (ROC), and Relative Strength Index (RSI) which are derived from the past price information for one or two week period prior to the trading day. For example, RSI is computed by dividing the sum of all price increases by the sum of all price increases and decreases for the past fourteen days. We also include the Market Breakdown (MB) which classifies the trading day's market into one of three categories. Market Breakdown values represent Long(Buy), Short(Sell), or Hold market for the day which is recommended by the professional traders using the perfect hindsight information. Table 1 categorizes input variables to develop neural networks into three types.

Table 1. Input Variables

Type	Variable
Price	Open
	High
	Low
	Close
Statistics	Moving Average (MA)
	Rate of Change (ROC)
	Relative Strength Index (RSI)
Indicator	Market Breakdown (MB)

A limited selection of variables makes possible to compare the neural network model against that of Trippi and DeSieno(1992). Trippi and DeSieno(1992) also uses the similar price variables and statistical variables whose specifics are not disclosed. When the selection of input variables between this study and Trippi and DeSieno(1992) is compared, major difference lies in the use of market indicator in this study and the use of trading day's opening price and the price of fifteen minutes after the market opening in Trippi and DeSieno (1992).

Neural networks typically work with inputs in the range 0 to 1 or -1 to $+1$. When input data are loaded into a neural network, it must be scaled into a numeric range that is comparable with the neural network algorithm. In this study, the input data scaling is performed via the linear scaling function which converts a range of values into $[-1, 1]$. The neural network then produces an output of value between 0 and 1. Desired outputs, Long(or Buy) or Short(or Sell) decisions for each trading day, were found by observing the actual market behavior in retrospect.

To validate the learning model, four year data of 1,013 trading days is divided into a learning and a test sample. The data for the year 1991 through 1993 is used as a learning sample to construct neural network models. The test sample, the data for the year 1994, is used for validation of the neural network. The composition of sample data is summarized in Table 2.

Table 2. Composition of Sample

	Sample Data (Trading Days)
Learning Sample	760
Test Sample	253
Total Sample	1,013

2. Configuring Network

To identify the network architecture an experiment with three-layer backpropagation networks is performed. The Sigmoid logistic function is used as the transfer function for each node since it is known that this function is particularly effective when the outputs are categories (NeuroShell 1993). The process of developing

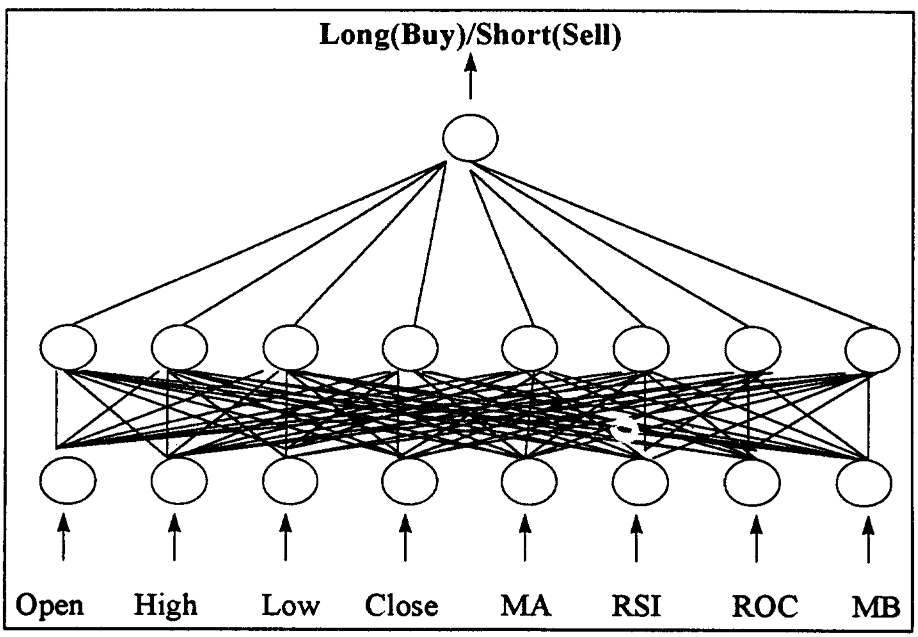

Fig. 1. A Neural Network for S & P Stock Index Futures

a neural network consists of trying several configurations to see which has the least error. After evaluating results from the experiment with different numbers of hidden nodes, from 30 to 2, a three-layer network with 8 input nodes, 8 hidden nodes and one output node was selected. Figure 1 shows the topology of the neural network used in the study.

3. Learning

Learning consists of presenting the learning data set to the network so that the weights can be adjusted to produce the target output for each input data. During a training epoch, the cases were presented in a random order. Each cycle of presentation of all cases is called an epoch. Weights and biases were initialized to random numbers. Weights are adjusted by the backpropogation algorithm after each input vector is presented. Typically, a large number of iterations of the learning data are required to produce a stable set of weights that can properly categorize the learning sample. The neural network was trained over 760 trading days from 1991 to 1993 for S & P500 stock index futures market.

The proper setting of the learning parameters is part of the art of neural networks. Learning ceases to make any progress if the learning rate and momentum are too high or the network has too few hidden nodes. A neural network continually works to improve the learning model's categorization of the learning sample inputs. Generally, learning improvement is continuous, but eventually there will be a point where the forward progress is too slow to be practical or observable. The experiment examined each neural network architecture with different values for these parameters and stopped with the learning rate and momentum factors of 0.1 and 0.1 respectively.

During the learning process it is important to train the neural network such that just the right amount of learning is applied to the learning sample and no more. Otherwise, over-learning tends to diminish generalization. One way to determine optimal learning is by setting a stopping criteria. The package allows the model developer to stop training when one of the following conditions is true about the training set (NeuroShell 1993) ;

- average error $<\lambda$
- epochs since minimum average error$>\lambda$
- largest error$<\lambda$
- learning epochs$>\lambda$

λ represents a value for the given criteria to stop training and is set by the model developer. λ with 500 for learning epochs is used in the experiment. It is equivalent to 380,000 events. Since the error distances do not continuously decrease during training, a record is maintained of those weights that yield the minimum error distance encountered during training. The weights at that time are used as the solution.

The relative contribution factor is a rough measure of the importance of a variable in predicting the neural network's output, relative to the other input variables in the same network. The contribution factor is the sum of the absolute values of the weights leading from the particular variable. The higher the number, the more the variable is contributing to the prediction or classification. Table 3 shows each variable's relative contribution factor in the order of ranking.

Table 3. Contribution Factor

Variable	Low	High	Open	Close	ROC	RSI	MA	MB
Relative Contribution Factor	6.08596	6.68809	7.25304	8.10350	9.43863	10.76891	11.05111	11.98944

From Table 3 we find market breakdown (MB) to be the most important variable. Next come statistical indicator variables. Price information variables are the least important ones. This confirms that the expertise of professional traders and derived information are more important than simple price information.

4. Testing the Model

Once the neural network is trained by the learning sample, the learned weights on the connections between nodes are kept constant during the testing phase. The neural network is tested with 240 trading events

Table 4. Cut-off Point Comparison

Cut-off Point	Test sample	Learn sample
0.48	57.71%	60.40%
0.50	58.50%	61.18%
0.52	61.66%	62.63%
0.54	62.45%	62.37%
0.56	61.27%	63.42%
0.57	62.45%	63.82%
0.58	61.66%	63.68%
0.60	60.87%	63.55%
0.62	60.08%	63.15%
0.64	59.68%	62.76%
0.66	60.08%	60.66%

of the year 1994. The predicted outcome of each trading day has been examined to find a cutting point for classifying either Long or Short. Table 4 shows the accuracy rate of the trained network at various cut-off points.

The accuracy rate of the trained network came out to be 62.5%, that is, the neural network made 158 right out of 253 trading decisions in the test data set when predicted outcomes were classified with 0.57 cut-off point. With the learning data set the trained network has 63.8% of accuracy rate (485 out of 760 trading decisions) at 0.57 cut-off point. However, at 0.50 cut-off point which is often used as a threshold in other studies, the network generated 58.5% of accuracy rate with the test data set. At 0.50 cut-off point it generated 61.2% of accuracy rate with the learning data set. The final decision rule for the output is set to 0.57 and can be stated as :

output unit $>$ 0.57 $\rightarrow$ Long(Buy)
output unit $\leq$0.57 $\rightarrow$ Short(Sell).

III. SIMULATED TRADING RESULTS

The ultimate criteria for adopting a technique as a trading aid is profitability. The output from the neural network must be turned into investment actions. Buy-and-hold test is a good benchmark against which to quantify excess return. Thus, we performed a simulated futures trading with the results of neural networks.

Profitability and other measurements are calculated with the following trading rules. A simulated futures trading is designed such that a buy order is issued if the closing price of each trading day is expected to be greater than the open price. And the opposite is the case when the relationship of the two prices is reversed. Trading each day is set to take place in the morning immediately after observing the open price. This differs from the trading rule of Trippi and DeSieno (1992) which executes the trading using the price fifteen minutes after the market opening.

The simulated trading assumed that order placement be executed at market closing time and the slippage will be 1 tick (5 points) for each transaction. In S & P 500 stock index future market 5 points correspond to 5\$. The commission was assumed to be \$5.50 per side. The trading simulation was first performed with the learning sample of 760 trading days from 1991 to 1993. The ex post near optimized network generated the simulation performance as in Table 5.

Table 5. Summary of Simulated Trading with the Learning Sample

Net Profit	% Gain	Max. Drawdown	Profit/Risk	Average Gain	Average Loss
$49,280	72	$1,943	22.79	$395.97	$259.08

From the simulation result with the learning sample, positive net profit represents net gains after deducting total losses from total gains during the trading period. 72 % of the trading generated positive profits. The average of all profits was $395.97 and the average of all losses was $259.08. Investors are not only interested in net profit, but also interested in risk-reward ratios. The biggest loss(maximum drawdown) was $1,943. Profit/Risk is computed by dividing the net profit by maximum exposure which is the maximum amount of money to risk or to invest in the market.

Next, the trained neural network model has been checked over the test period which covers from January 3, 1994 through December 30, 1994. Figure 2 shows the Long/Short positions for 253 trading days in 1994. The top box indicates the time to buy and its holding period and the bottom box indicates the time to sell and its holding period.

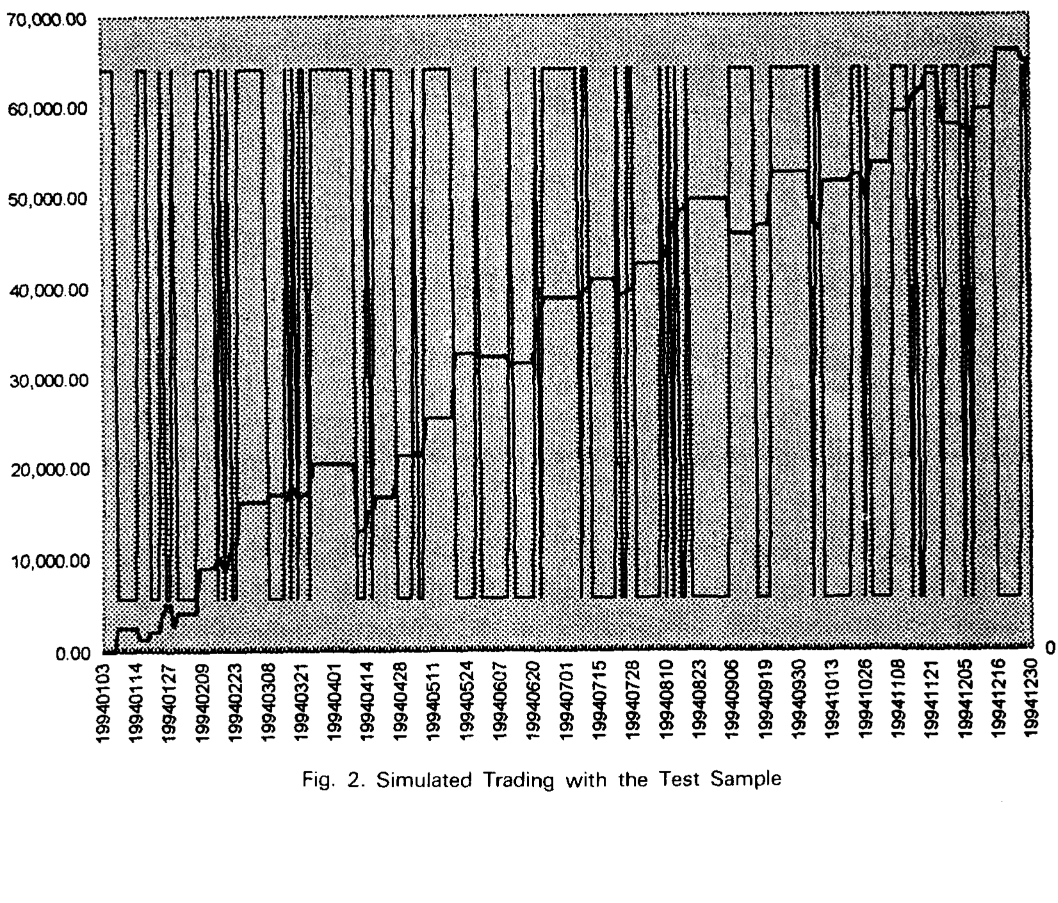

Fig. 2. Simulated Trading with the Test Sample

Figure 2 also shows the ex ante trading performance resulting from these positions. The account value line(or equity curve) shows strong upward slope very consistently. The account value line is simply the total equity of an account (plotted on the vertical axis) over a series of trades (plotted on the horizontal axis). The ex ante performance shows a final account value of $63,308. For the seventy-eight days that trading took place, there were gains on fifty-two days, or 67.5% of the time. This trading generates a net profit of $12,831. Other measurements are shown in Table 6.

Finally, the performance of the neural network model with the test sample has been compared with the perfect hindsight information, and five-day and ten-day moving average system. Table 6 shows the performance comparison. The net profit of neural network trading is 22% of that of the perfect information trading. The moving average trading shows a net loss of $12,008. The trading with the moving average technique shows a large maximum drawdown of $12,153. This comparison reveals that neural network outperforms moving average.

Table 6. Performance Comparison

Model	Net Profit(Loss)	% Gain	Max. Drawdown	Profit/Risk	Average Gain	Average Loss
Perfect Information	$58,088	98	0	infinite	$433.41	0.00
Neural Network	$12,831	68	$1,475	4.90	$405.50	330.64
Moving Average	($12,008)	34	$12,153	0.98	$288.69	657.17

Ⅳ. CONCLUSIONS AND FUTURE RESEARCH

This article describes the development and performance of neural networks in trading S & P 500 stock index futures contracts. It supports the use of neural networks in a practical intelligent trading system. The intelligent system should employ techniques to integrate various information from different sources and provide quick recommendation to the decision maker. Each information source may be a subsystem of the intelligent system. The subsystem may be a conventional program or a modeling system. The best way to connect and integrate individual subsystems into the intelligent system is through the knowledge-based systems approach.

Neural networks can be used as a knowledge acquisition tool. Neural networks can generate profitable trading rules from a profitable trade data set. Trading rules and price forecasting models need to be incorporated into the expert system which is a key component of the intelligent trading system. The expert system is required to implement money management constraints such as the maximum amount of drawdown(or loss) that the trader can tolerate and limits on trading position size. The trading system should require to exit if the position taken is no longer favorable even if it means taking a loss and to ride on the profits if the market starts trending in a favorable direction (not take profit too early).

The current work has some limitations and lays ground for future extensions. More extensive input data including fundamental information are to be used to examine the possibility of performance improvements, to show how much improvement can be made with other variables, and to find out major dominant information affecting the stock index futures market. The simulated trading is set to occur at a specified time during each trading day to facilitate performance evaluation. Trading volume is also fixed in the simulation for the sake of simplicity. The performance in actual trading would be improved by varying the timing of the trade and its volume.

The robustness of the performance of our model could also be checked by splitting up the data sample into many different training and testing periods. For example, a moving simulation approach (Kimoto, Asakawa, Yoda, and Takeoka 1990) is performed at various lengths of periods. This is the approach requiring multiple repetitive simulations of learning and prediction exercises as time advances. Training and testing periods can even be switched to assess the degree of the stationarity of the index futures prices.

More realistic assumption and strategies of trading simulation should be devised for testing system performance and designing real-time trading systems. We plan to look into the possibility of a pseudo-arbitrage opportunity in the stock index futures. It is well known in the finance literature that arbitrage profits are possible when the actual stock index futures prices differ from the so-called cost-of-carry fair prices by more than transactions costs. We could feed in the fair prices to the neural network and conduct this experiment.

REFERENCES

1. Chester, M., *Neural Networks : A Tutorial*, Prentice Hall, 1993.
2. J.E. Collard, "Commodity Trading with a Three Years Old," in *Neural Networks in Finance and Investment : Using Artificial Intelligence to Improve Real-World Performance*, Trippi, R., and E. Turban (eds.), Probus Publishing Co., 1993.
3. Feedman, R. S., Klein, R. A., and Lederman (eds.), J., *Artificial Intelligence in the Capital Markets*, Probus Publishing, 1995.
4. Gallant, S. I., *Neural Network Learning and Expert Systems*, MIT Press, 1993, Chapter 11.

5. Hush, D. R. and B. G. Horne, "Progress in Supervised Neural Networks : What's New Since Lippmann," *IEEE Signal Processing Magazine,* January 1993, pp.8-39.
6. Kimoto T., K. Asakawa, M., Yoda, and M. Takeoka, "Stock Market Prediction System with Modular Neural Networks," in *Neural Networks in Finance and Investment : Using Artificial Intelligence to Improve Real-World Performance,* Trippi, R., and E. Turban (eds.), Probus Publishing Co., 1990.
7. Klimasauskas, C. C., "Applying Neural Networks," *PCAI,* January/February 1991 : 30-33 ; March/April 1991 : 27-34 ; May/June 1991 : 20-24.
8. Lippman, R. P., "An Introduction to Computing with Neural Nets," *IEEE Acoustics, Speech and Signal Processing Magazine,* April 1987, pp.4-22.
9. Lo, A., and A. MacKinlay, "Stock Prices Do Not Follow Random Walks : Evidence From a Simple Specification Test," *Review of Financial Studies,* 1988.
10. Lo, A., and A. MacKinlay, "When Are Contrarian Profits Due To Stock Market Overreaction ? ," *Review of Financial Studies,* 3, 1990, pp.175-205.
11. Loofbourrow, J. W. and Loofbourrow, T. H., "What Artificial Intelligence Brings to Trading and Protfolio Management," in *Artificial Intelligence in the Capital Market,* Probus Publishing, 1995
12. *Neuro Shell 2 User's Manual,* Ward Systems Group, Inc., 1993.
13. Park, J. and Han, I., "Predicting Korea Composite Stock Price Index Movement Using Artificial Neural Network," Korean Journal of Expert Systems, 1995.
14. Refenes, A. (ed.), *Neural Networks in the Capital Markets,* John Wiley & Sons, 1995.
15. Rumelhart, D. E., B. Widrow, and M. A. Lehr, "The Basic Ideas in Neural Networks," *Communications of the ACM,* March 1994, pp.87-92.
16. Salchenberger, L. M., E. M. Cinar, and N. A. Lash, "Neural Networks : A New Tool for Predicting Thrift Failures," *Decision Sciences,* July/August 1992, pp. 899-916.
17. Tam, K. Y. and M. Y. Kiang, "Managerial Applications of Neural Networks : The Case of Bank Failure Predictions," *Management Science,* July 1992, pp. 926-947.
18. Trippi, R., and D. DeSieno, "Trading Equity Index Futures with a Neural Network," *The Journal of Portfolio Management,* 1992, pp.27-33.
19. Trippi, R., and E. Turban (eds.), *Neural Networks in Finance and Investment : Using Artificial Intelligence to Improve Real-World Performance,* Probus Publishing Co., 1993.
20. Widrow, B., D. E. Rumelhart, and M. A. Lehr, "Neural Networks : Applications in Industry, Business and Science," *Communications of the ACM,* March 1994, pp.93-105.
21. Yoon, Y., T. Guimaraes, and G. Swales, "Integrating artificial neural networks with rule-based expert systems," *Decision Support Systems,* Vol.11, 1994, pp. 497-507.