인공신경망을 이용한 한국 종합주가지수의 방향성 예측

박종엽 · 한인구*

Predicting Korea Composite Stock Price Index Movement Using Artificial Neural Network

Jongyeob Park, and Ingoo Han**

ABSTRACT

This study proposes a artificial neural network method to predict the time to buy and sell the stocks listed on the *Korea Composite Stock Price Index (KOSPI)*. Four types (NN1, NN2, NN3, NN4) of independent networks were developed to predict KOSPIs up/down direction after four weeks. These networks have a difference only in the length of learning period. NN5 - arithmetic average of four networks outputs - shows an higher accuracy than other network types and Multiple Linear Regression(MLR), and buying and selling simulation using systems outputs produces higher return than buy-and-hold strategy.

Keywords : Korea composite stock price index (KOSPI), Artificial neural network, moving-period simulation, multiple linear regression(MLR), buying-and-selling simulation

1. Introduction

Worldwide stock markets have recently experienced dramatic volatility in their returns. Traditionally,

* 한국과학기술원 테크노경영대학원
** Korea Advanced Institute of Science and TechnologyGraduate School of Management

two main approaches - *time series analysis and fundamental analysis* - exist in predicting stock price. Time series analysts who use ideas from the theory of nonlinear stochastic time series believe that stock price itself reflects all information. Therefore, observing the historical movements of stock price suffices to predict the future price. Conversely, the second group - we call them fundamentalists - has rooted their idea in that stock price is affected by and give a effect to macroeconomic variables such as interest rate, gross national product, M2, export & import, etc.

Nevertheless, the results of various statistical analyses are not quite satisfactory to meet our expectation. Besides, they have some limitation of applications according to the data characteristics and also require comparatively strict assumptions on the distribution of a variables. As a result, artificial intelligence (AI) technologies are introduced in this area [Braun & Chandler, 1987].

Over the past four decades, the field of artificial intelligence (AI) has made a great progress toward computerizing human reasoning. Especially, neural network (NN) has newly received special interests. Modeling functions of NN are being applied to a widely expanding range of applications. Because of its non-linear learning and smooth interpolation capabilities, NN is supplementing or taking the place of statistical and conventional expert system (ES) approaches in many financial decision making [Trippi et al., 1993]. Stock market prediction is one of the such problems [Kimoto et al, 1990]. So, we choose NN as a means to test whether it could produce a successful model in which their generalization capabilities could be used for stock market prediction in Korea.

In this study, we want to propose a neural network method to predict the time to buy and sell the index portfolio of KOSPI. The stock market prediction studies using NN are scarce in Korea, so we will mainly focus on the applying methodology and its validity rather than concentrate on the study of the creative development of learning algorithm or optimizing technique for the neural network model.

The remaining chapters of the paper are organized as follows. In Chapter 2, we will review the literature of NN applications in stock market. In Chapter 3, the structure of experiment will be explained in detail. Chapter 4 summarizes the results of experiment. Finally, the contribution of the research and further research directions are discussed in Chapter 5.

2. Literature Review

Recently, since the late 1980s, some researchers in finance and investment began to use the AI methods. AI based trading systems were developed and used by professional investing company. It shows symptoms of wide spread uses of AI technique in worldwide security markets. In this chapter, we will review NN applications in stock market.

White (1988) tried to gain evidences against the efficient market hypothesis, and illustrated how the search for regularities in historical asset price data using neural network method might proceed, using the case of IBM daily common stock return. The training data covers trading days during the period 1974 2/4 through 1978 1/4. The evaluation periods covers 1972 2/4 through 1974 1/4 and 1978 1/4 through 1980 1/4.

As the first step, he examined the empirical evidence against the simple efficient markets hypothesis using the linear model. The linear autoregressive model of order p *(AR(p)* model) corresponds to a very simple two-layer linear feed-forward network. Given inputs $r_{t-1}, \cdots, r_{t-p}$, the network output is given as $\hat{r}_t = \hat{w}_0 + \hat{w}_1 r_{t-1} + \cdots + \hat{w}_p r_{t-p}$, where $\hat{w}_0, \hat{w}_1, \cdots, \hat{w}_p$ are the network weights arrived at by a suitable learning procedure. He designed an empirical estimate of R^2, computed in the standard way as :

$$R^2 = 1 - \hat{\text{var}}\ \varepsilon_t / \hat{\text{var}}\ r_t,$$

$$\textit{where,} \quad \hat{\text{var}}\ \varepsilon_t \equiv n^{-1} \sum_{t=1}^{n} (r_t - \hat{r}_t)^2,$$

$$\hat{\text{var}}\ r_t \equiv n^{-1} \sum_{t=1}^{n} (r_t - \bar{r})^2,$$

$$\bar{r} \equiv n^{-1} \sum_{t=1}^{n} r_t,$$

and n is the number of training observations, 1000.

The exercise was carried out for a post-sample period of 500 days, and a pre-sample period of 500 days. For the post-sample period he observed a correlation of -.0699 ; for the pre-sample period, it is .0751. Such results do not constitute convincing statistical evidence against the efficient markets hypothesis. Although the result might be disappointing, he suggested a new tool to testify *efficient markets hypothesis*.

Kimoto et al., (1990), developed several learning algorithm and prediction method for the TOPIX (Tokyo Stock Exchange Prices Indexes) prediction system.

The prediction system is made up of several neural networks (modular neural network) that learned the relationships between various technical and economical indexes and the timing for when to buy and sell stocks. The goal is to predict the best time to buy and sell for one month in the future. They developed *supplement learning*, based on the error back propagation proposed by Rumelhart, that automatically schedules pattern presentation and changes learning constants.

They also developed a prediction method called *moving simulation*. In this system, prediction periods is done by simulation while moving the objective learning and prediction periods. The TOPIX prediction system improves its prediction accuracy by averaging prediction results of modular networks that learn for different learning data items. Prediction was done for 33 months from January 1987 to September 1989.

To verify the effectiveness of the prediction system, a simulation of buying and selling of stock was done. Buying and selling according to the prediction system made a greater profit than the buying and holding. The TOPIX index of January 1987 was considered as 1.00, it was 1.67 by buy-and-hold at the end of September 1989. It was 1.98 by buying and selling operation according to the prediction system. The use of the system showed an excellent profit.

Kamijo and Tanikawa (1990) proposed a recurrent neural network model for stock pattern recognition and to develop a new method for evaluating the network. In stock trading with technical analysis, price patterns in Japanese-style stock charts, such as *triangle*, indicate an important clue to the trend of price pattern in stock price. Sixteen *triangles* were extracted by an expert, from Japanese-style *candlesticks* charts for names of corporations listed in The First Section of Tokyo Stock Exchange. The patterns were divided into two groups, 15 training patterns and one test pattern. In order to eliminate the bias due to difference in name and time span, the variation rate for the stock price average by exponential smoothing and the dissociations from the average of high and low price were utilized as normalized stock data.

The test set of triangle patterns was accurately recognized in 15 out of 16 experiments. It implied that neural network can be also used as an effective tool for technical analysis which depend upon the various types of chart analysis in stock market.

Yoon and Swales (1991) attempted to improve the forecasting accuracy of stock price performance using neural network method, and compared its predictive power with that of multiple discriminant analysis (MDA). They emphasized that both quantitative and qualitative variables help form the basis of investor expectation of stock price, and influence investment decision-making.

They could obtain the qualitative information from the firms annual report to the stockholder, and the quantitative information from the *Fortune* 500 and *Business Weeks Top 1000*. The network for the prediction of stock price performance used the following nine inputs (confidence, economic factors, growth, strategic plans, new products, anticipated loss, anticipated gains, long-term optimism, and short-term optimism) and two output (well-performing firms and poorly-performing firms) parameters.

The neural network model is structured in a four-layered network: an input layer, two hidden layers, and an output layer. The input for the network consisted of nine variables and output was a classification into two patterns: a firm whose stock price performed well and a firm whose stock price performed poorly.

Comparison of neural network technique with the MDA approach indicated that the NN approach can significantly improve the predictability of stock price performance.

Jang et al. (1991) presented one of the most carefully researched approaches to stock market predictions. A *dual adaptive structure neural network (DAS net)* that can predict short-term trends of price movements, as well as recognize reversals, was utilized to develop an intelligent profitable stock trading system for Taiwan stock market.

The input vector of the *DAS net* consisted of technical indices preferred by human experts. The output vector, on the other hand, incorporated the predictive short-term trends of price movement of the chosen stock. The calculation of retrospective input vector and predictive output vector from stock price and volume data was also discussed.

They showed that a portfolio of dual-module neural networks generalizes better than a single-module neural network. In addition, they justified the effectiveness of the proposed structure-level adaptive neural networks by comparing their generalization ability with that of fixed-structure neural networks.

Kryzanowski et al. (1993) examined the ability of an NN using historical and macroeconomic data to discriminate between stocks providing positive future returns and those providing negative future returns.

	Paradigm	Transfer function	Learning parameter	# of layer # of PEs	Input, output data	Used S/W
White (1988)	BPN			(x, x, x)		
Kimoto et al. (1990)	Supplementary learning (modular)	Sigmoid	0.4 (learning rate) 0.8(momentum)	(6, x, 1)	[0, 1] [0, 1]	
Kamijo and Tanigawa (1990)	Recurrent connection network	Sigmoid	(x, x, x, 4)	[0, 1]		
Ahmadi (1990)	BPN	Sigmoid	(x, x, x, x)	[0, 1]		NeuralWare ProII
Yoon and Swales (1991)	BPN	Sigmoid	0.1 (learning rate)	(9, x, x, 2)		C-Language
Bergerson and Wunsch II (1991)	BPN	Sigmoid			(54, 54, 1)	
Matsuba et. al., (1992)	Feedback network		0.5 (learning rate)	(14, 13, 7)	[0.1,0.9]	
Refenes	BPN		0.3 (learning rate) 0.7 (momentum)	(3, 32, 16, 1)		
Collard (1993)	BPN	Sigmoid	0.9 (momentum)	(37, 30, 1)		
Falas (1994)	line search BPN			(9, 3, 1)	[-1, 1]	
Kim and Cho (1994)	BPN	Sigmoid	0.3 (learning rate) 0.5 (momentum)	(18, 18, 2)	[0, 1] [0, 1]	NeuralWare ProII
Wu (1994)	Competitive learning			(16, 7, 16) : clusters		

Table 1 Neural Network Models in Stock Market

The NN learned the relationship between a companys stock return one year in the future and the most recent four years of financial data on the company and its industry as well as data on seven macroeconomic factors. They employed the pattern classification algorithm of the Boltzman Machine(BM), which employs a technique of stochastic optimization called simulation annealing.

Designated BM achieved a 66.4% overall accuracy in predicting whether a stocks return will be positive

Source	Statistical method	AI method	Application Domain
White (1988)			Stock return prediction (IBM)
Kimoto et al. (1990)	0.543 (Regression : teaching)	0.991 (3-layered NN : teaching), 0.527 (Integrated NN : direction)	Stock index prediction
Kamijo and Tanigawa (1990)		100 (4-layered NN : training) 93.8 (4-layered NN : testing)	Stock price pattern recognition
Ahmadi(1990)			Arbitrage pricing theory
Yoon and Swales (1991)	74 (MDA : training) 65 (MDA : testing)	91 (4-layered NN : training) 77.5 (4-layered NN : testing)	Stock price performance
Bergerson and Wunch II (1991)		760 (3-layered NN+rule-based system : testing, 2 years)	Stock Index prediction
Jang et al. (1991)	0.42 (MLR : long-term, goodness of fit) 0.99 (MLR : Short-term)	0.96 (MLR : long-term) 0.99 (Dual net : short-term)	Stock price prediction
Matsuba et al. (1992)		0.133 (3-layered NN)	Time-series data
Refenes et al. (1993)	0.138 (MLR : training, RMS) 0.123 (MLR : testing, RMS)	0.044 (4-layered NN : training, RMS) 0.112 (4-layered NN : testing, RMS)	Stock ranking
Collard (1993)		52.5 (3-layered NN : testing, rate of return for 178 days)	Long/short position
Falas et al. (1994)	65.9 (Logit : training) 62.0 (Logit : testing)	65.7 (3-layered NN : truing) 62.9 (3-layered NN : testing)	Prediction of earning
Kim and Cho (1994)	70 (MDA : learning) 60.54 (MDA : testing)	88.51 (3-layered NN : learning) 65 (3-layered NN : testing)	Stock index prediction (direction)
Wu	9. 13 (Box-Jenkins)	4.2 (3-layered NN : mean absolute error)	S&P 500

Table 2 Results of NN Models in Stock Market

or negative over the coming year. In 11 cases, the BM did not make any call (i.e., could not decide on the expected outcome). Judging the BM only on the cases for which it made a decision (described as clear calls) results in an accuracy level of 71.7%.

This research represented an initial step in the development of a model in the development of a model that can classify a companys stock price as being likely to either rise or fall over the coming year.

Kim and Cho (1994) applied multilayer feed-forward network to predict future KOSPI (Korea Composite Stock Price Index) value. Input variables for network contains 10 macroeconomic and 8 technical variables. In some cases, they might have high correlation. Used data for experiment were monthly time-series data from February 1981 to December 1993 : 1981 through 1990 for learning, 1991 through 1993 for testing.

Learned network showed 65% prediction accuracy, and higher predictability compared to 60.64% of MDAs. Developing methodology combining neural network model with statistical method was recommended for future research.

Table 1 and Table 2 show the performance of previous research, and the architecture and parameter of neural network based system, respectively.

3. Structure of Experiment

3.1. Case Selection

The data set used for learning and prediction covers the period from Jan 1990 to March 1995. The movement

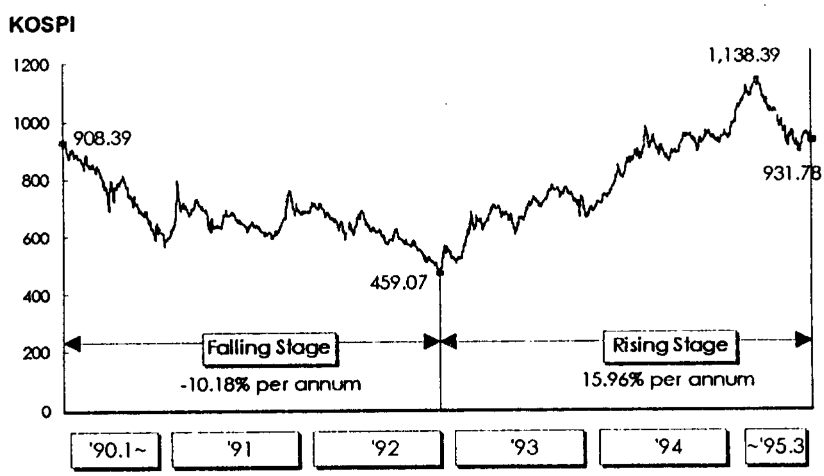

Figure 1 KOSPI Movement Through Jan 1990 - March 1995

of the Korea stock market through Jan. 1990 - March 1995 is depicted in Figure 1. The KOSPI value was 908.59 in Jan 3, 1990 and 931.78 in March 31, 1995. Per annum return during this period was 0.48%.

3.2. Variable Selection

The selection of proper variables is an important and critical phase for accurate stock market prediction model development. Stock price may be predicted by both macroeconomic data and technical data. This kind of multidimensional method of analysis is known as *synergistic market analysis* [Mendelsohn, 1994]. We chose input variables from following two steps. First, several relevant literature were reviewed (Kimoto et. al., 1990, Oh and Shin, 1990, Kim and Cho, 1994, Coryo Securitty Co., 1994, Park, 1994) to form a initial set of input variables. Secondly, we have interviewed with fund managers and model developers of investment institutions to fix the input variable set. With the aid of them, we could exclude several inconvenience variables such as real estate index, composite index of indicator, industry production price index, and M2. Theoretically, M2 should be a very important variable which affects stock market, but we excluded it because of the time interval resulted from delayed public announcement. Several technical variables such as trading volume, deposit, and foreign stock market index were also newly included from the interviews. Especially, using the Dow Jones Index was strongly recommended by most part of interviewees. As a result, we have chosen seven input variables as follows :

X_1 : *KOSPI*
X_2 : *VOLUME*
X_3 : *DEPOSIT*
X_4 : *CORPORATE BONDS YIELD*
X_5 : *EXCHANGE RATE OF Won / US\$*
X_6 : *EXCHANGE RATE OF Won / 100Yen*
X_7 : *DOW JONES INDEX (DJI)*

3.3. Neural Network Architecture

3.3.1. System Overview

This prediction system consists of four independent neural network types which are different only in the length of learning period. Each type uses the historical weekly data to learn the relationships between the various input variables and the next months KOSPI movement, up or down. The outputs produced by each type and the arithmetic average of them are evaluated to find the optimal length for learning period. Input patterns and designated neural network types will be discussed in detail.

3.3.2. Moving-period simulation

For prediction of an economic system, especially such as stock price, in which the prediction rules are changing continuously, learning and prediction must follow the changes [Kimono et al., 1990]. We used a prediction method called moving-period simulation suggested by TOPIX prediction system (Kimoto et. al., 1990). As shown in figure 2, the system learns data for the past L months, then predicts the next months and advances while repeating this. The proposed system can be evaluated with varying learning period L, and we will classify and refer our network type NN1, NN2, NN3, NN4, respectively, according to their length of learning period. Each network type learns using the pre-defined number of input patterns, and produces a single output-market return in next month. Thirtynine networks for each type could be made from our experiment data set. Each network type is explained in Table 3 in detail. In table 3, there are patterns unused. For these patterns, teaching data couldnt be generated, therefore we couldnt use these patterns for training or testing the networks.

Type	Learing Period (months)	Number of Patterns in				Number of Networks learned & predicted
		Train	Test	Unused	Output	
NN1	6	16	7	3	1	39
NN2	12	33	16	3	1	39
NN3	18	50	25	3	1	39
NN4	24	78	33	3	1	39

Table 3 Network Types and Their Learning Periods

3.3.3. Preprocessing of Data & Setting Teaching Data

Once the most appropriate raw input data have been selected, they must be preprocessed, otherwise, the *NN* may not produce accurate forecasts. The decisions made in this phase of development are critical for the performance of a network [Mendelsohn, 1994].

Transformation

We transformed daily data of each variable to weekly average data. In other words, weekly average data was used to generate input patterns for the networks. The weekly average data of input variables are then converted into three indices : *moving average, a regression coefficient* in time (12-week slope order), and the *difference* between index value and moving average-representing level, trend, and relative level of indices, respectively. We used 6-week simple moving average method with same weights. We used simple linear regression analysis to generate the regression coefficient in time by setting time as independent variable

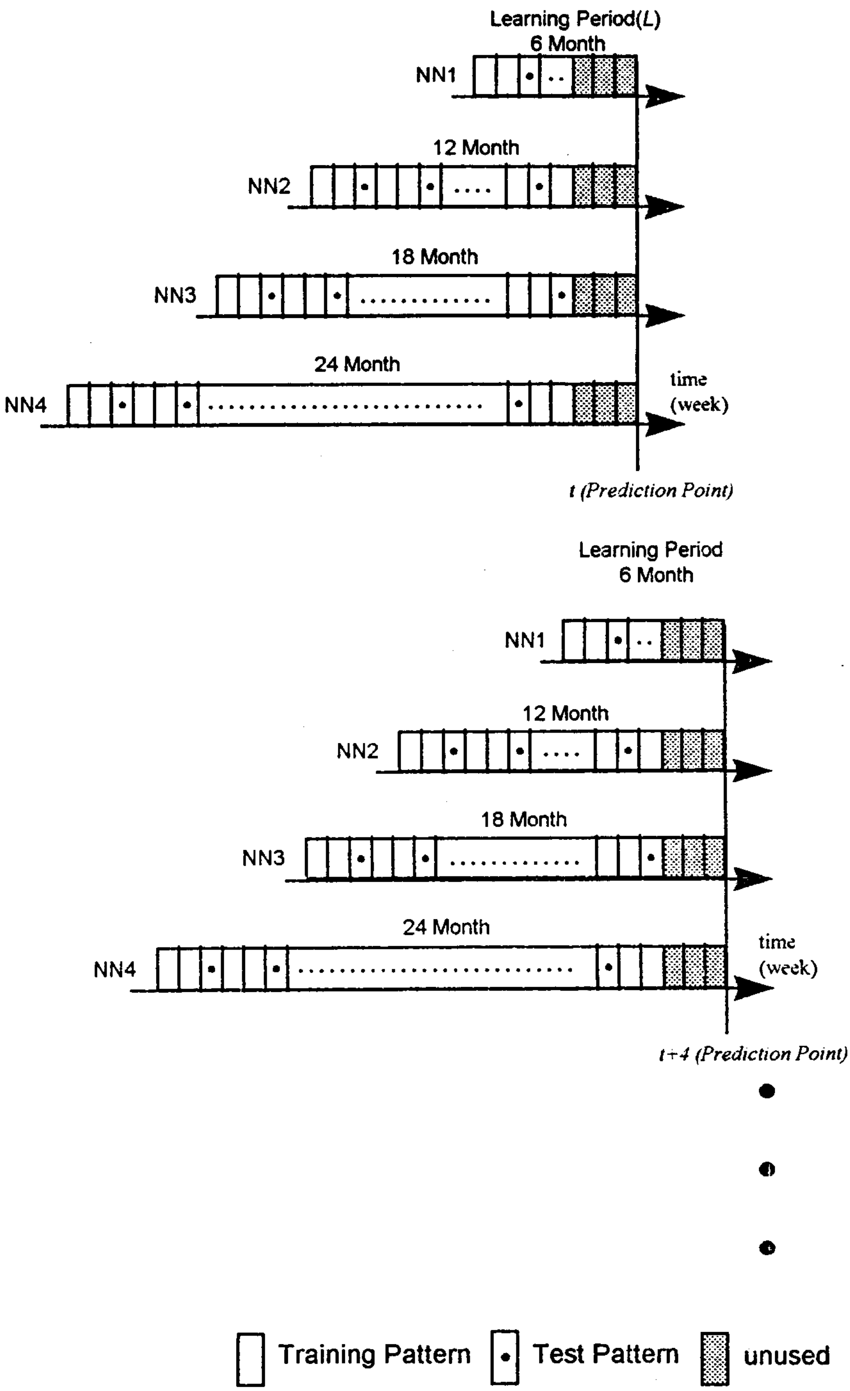

Figure 2 Moving-Period Simulation

and next months KOSPI value as dependent one. 6-week and 12-week slope derived from several times of simulations which test correlations with teaching data. These three transformed indices have been regarded as a typical forms of technical analysis. Consequently, 21-indices from X_{11} to X_{73} were included in the input pattern.

X_{k1} : moving average of *k-th* variable
X_{k2} : regression coefficient of *k-th* variable
X_{k3} : difference between value moving average of *k-th* variable and variable itself,
where $k=1, 2, \ldots\ldots, 7$.

Normalization

The main steps of data normalization are making data regular and scaling them into the range used by the input neuron, typically in the range of zero to one. We used a *log function* to adjust each variable as regular as possible. And, they are then processed by *simple linear scaling method* which normalizes into the [0, 1] section.

Setting Teaching

Data The output of neural network is an arithmetic sum of coming four weeks returns, which is equivalent to a monthly return. When the *KOSPI* weekly return at week t *is* r_t, teaching data R_t is defined as :

$$r_t = \ln \frac{KOSPI(t)}{KOSPI(t-1)},$$

$$R_t = \sum_{i=1}^{4} r_{t+i}/4$$

KOSPI(t)=KOSPI value at week t (t ∈ sample cases)

Then teaching data, R_t, should also be normalized into [0, 1] section in one output unit.

3.3.4. Topology

General structure and parameter is presented in Table 4. The basic neural network architecture consists of three layers which are completely connected to form a hierarchical network. We used *back-propagation algorithm* which was proposed by Rumelhart (1986).

Number of Input Nodes	21	
Number of Output Nodes	1	
Number of Hidden Layer	1	
Number of Nodesin Hidden Layer	NN1	16
	NN2	18
	NN3	20
	NN4	21
Paradigm	Backpropagation	
Learning rate	Modified delta weight	
Transfer Function	Sigmoid (Logistic)	

Table 4 Structure and Parameters of Neural Network Model

The experiment is performed by using *Neuroshell2* which is one of the commercial package provided by *Ward System Group, Inc.* The number of hidden neurons were also determined by the package by following formula :

$$\# of\ PE = \frac{1}{2}(input + ouput) + \sqrt{\# of}\ patterns$$

3.3.5. Learning Control

One of the important problems is when to stop training. If we train too little the network will not learn

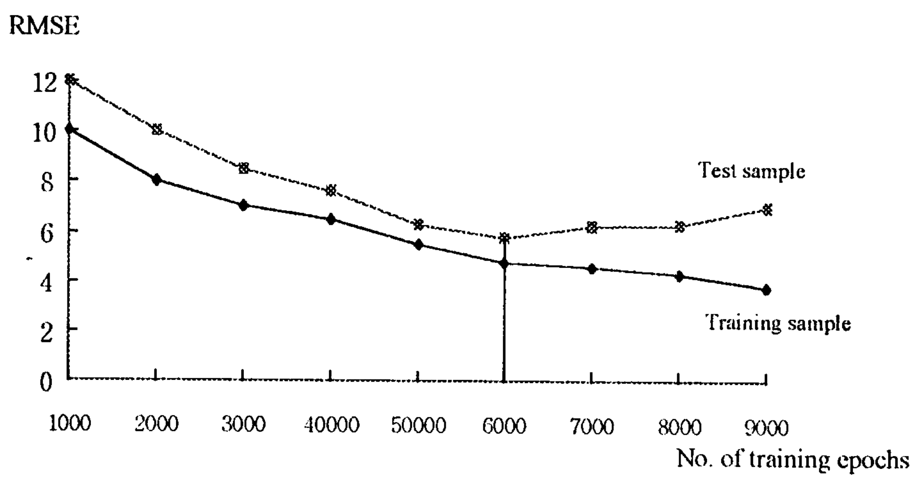

Figure 3 Neural Network Prediction Error Trace in Training and Test Data

the patterns fully. Conversely, if we train too much, the network learns the noise or memorizes the training patterns and would not generalize well with new patterns. This symptom is called *overfitting* of network.

Hecht and Nielson (1990) suggest that there should be two samples that should be used for learning. A samples that is used for the training and the other for the validate the network. The second sample is different from hold-out sample, since it is used during training to identify the point at which overtraining starts to occur. Figure 3 shows the RMSE(Root Mean Square Error) of the network that is observed generally in the training data and test data.

3.3.6. POSTPROCESSING

In this step, arithmetic average - we refer it NN5 - of the outputs of four independent network types is calculated. Then output of each network (NN1, NN2, NN3, NN4, and NN5) should be transformed into up/down direction. If the network output is greater than 0.5 it signals that market is expected to go up. Otherwise, market index is expected to go down.

4. Analysis of Results

4.1. Results of Moving-Period Simulation

Table 5 shows the learning capability of each network by two measures. Accuracy and correlation coefficient in the test set show a slightly decreasing trend as learning period increases. On the average, accuracy rate in the training set is 96.73% and in the test set 87.94%.

	Train Set		Test Set	
	Accuracy (%)	Correlation	Accuracy (%)	Correlation
NN1	96.87	0.956230	92.81	0.903229
NN2	97.05	0.959866	87.18	0.884762
NN3	96.84	0.964119	86.49	0.852875
NN4	96.15	0.959559	85.29	0.836406
Average	96.73	0.959950	87.94	0.869318

Table 5 Performance of NN in Train and Test Set

Table 6 shows the prediction capability of networks and there are no distinct differences between types in accuracy rate except NN5. NN5 dominates the others in accuracy rate. Correlation coefficient records ranged from 0.34 to 0.61. Higher correlation coefficient, however, does not necessarily guarantee higher accuracy. It may be resulted from the disturbance effect from the values of teaching data near 0.5.

	Accuracy (%)	Correlation
NN1	69.23	0.526957
NN2	71.79	0.527035
NN3	69.23	0.337551
NN4	71.79	0.608690
NN5	79.46	0.565812
Average	72.30	0.512449

Table 6 Performance of NN in the Prediction Set

4.2. Comparison with the Results of Multiple Linear Regression

For the evaluation purpose, we have compared with multiple linear regression (MLR) which most resembles our model in a view that it is also a multivariate data analysis and produces analog output. In practice, several trading systems have been compared with MLR.

	LEVEL	TREND	RELATIVE LEVEL	TOTAL
KOSPI	37*	27	53	117
VOLUME	37	28	18	83
DEPOSIT	54	28	15	97
CORPORATE BONDS YIELD	24	41	29	94
EXCHANGE RATE OF WON/US$	19	75	65	159
EXCHANGE RATE OF WON/YEN	23	22	34	79
DOW JONES INDEX	26	24	43	93
TOTAL	220	245	257	722

*frequency

TABLE 7 Selection frequency of Stepwise MLR

The same data set for learning and prediction are also applied to MLR. The selected variables for the neural networks are often highly correlated among them. First, we applied all variables in the MLR. In this case, it was good at training but very poor at prediction. Secondly, the variables for the MLR are selected by the stepwise method. In this case, it is worse than the first case in learning capability, but superior in prediction. On the average, 5.1 variables - maximum 12, minimum 1-are selected by the stepwise method. Frequencies for entering the model of variables are listed in Table 7.

The mean values of correlation coefficients of NN and stepwise MLR in training set are presented in Table 8. The values of accuracy and correlation coefficient in prediction set are presented in Figure 4 and Figure 5, respectively. These show that the NN absolutely outperforms the MLR in learning and prediction capability.

	Mean of Correlation Coefficient
NN	0.959950
MLR	0.674595

Table 8 Comparison of Learning Capability between NN and MLR in Test Set

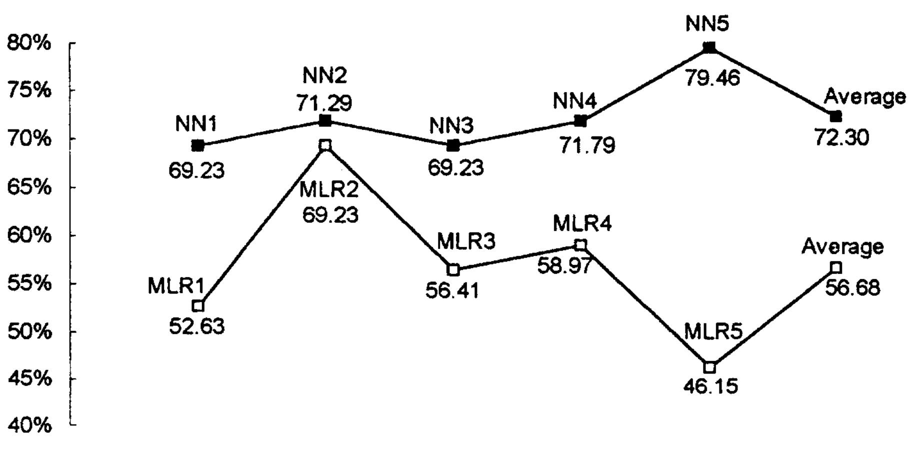

Figure 4 Comparison of Accuracy Between NN and MLR in Prediction Set

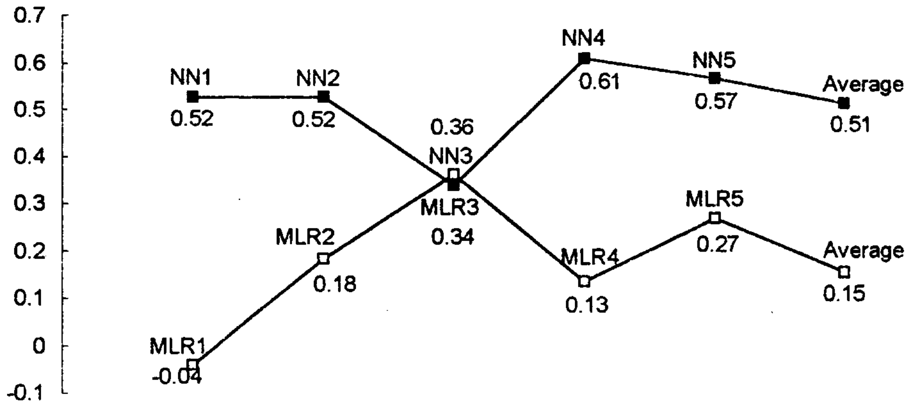

Figure 5 Comparison of Correlation Coefficient Between NN And MLR in Prediction Set

4.3. Buying-and-Selling Simulation

In this section, we attempt to use network output for trading simulation. Thereby, we can judge the practical performance of prediction model. We trade a portfolio composed of all stocks in index. According to the signals produced by the system, we buy and sell stocks under the following simple rules :

```
Rule 1
  if system output is BUY(network output >0.5)
    then {
      if BUY exist
      then HOLD STOCKS ;
    else BUY STOCKS ;
  }

Rule 2
  if system output is SELL (network output<0.5)
    then {
      if BUY exist
      then SELL STOCKS ;
    else HOLD CASHES ;
  }
```

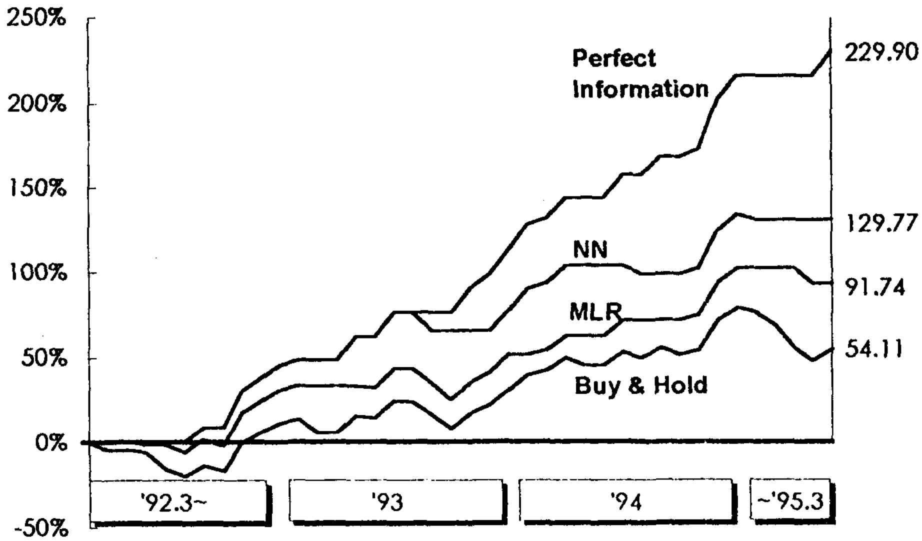

Figure 6 Results of Buying-And-Selling Simulation

We compared buy-and-hold strategy with buying-and-selling strategy through the periods from May 22, 1992 to March 19, 1995. Buying and selling actions at each point were directed from the outputs of *NN5*, *MLR3*, and perfect information, respectively. We assume that there exist no transaction costs, and no interest gains by holding cashes. The results are depicted in Figure 6 and Table 9. NN5 achieved the highest average annual return of 31.96%, and MLR3 achieved 24.23%, while Buy-and-Hold strategy resulted in 15.51%.

	NN	MLR	Buy-and-Hold	Perfect Information
Accuracy	75.00 %	75.00 %		100.00 %
When up	(15/20)	(15/20)		(20/20)
Accuracy	84.21 %	63.16 %		100.00 %
When down	(16/19)	(12/19)		(19/19)
Total	79.46%	69.23%		100.00 %
Accuracy	(31/39)	(27/39)		(39/39)
Accumulative	129.77%	91.74%	54.11%	229.90%
Rate of Return				
Yearly Average	31.96%	24.23%	15.51%	49.32%
Rate of Return				

Table 9 Results of Buying-and-Selling Simulation

5. Concluding Remarks

This paper proposed neural networks system to predict the timing when to buy and sell stocks listed on *KOSPI.* The prediction system showed an excellent performance in accuracy and Buying-and-Selling simulation. Nevertheless, this system has several limitation such as variable selection method, determining network parameters, and optimal forecasting time horizon. Hybrid systems combining neural networks with other AI technologies may release these limitations (Bergerson, 1991, Fishman, 1991, Nikolopoulos, 1994). Development of intelligent trading systems for portfolio selection and stock index futures market is suggested for the further research.

REFERENCE

1. Bergerson, K and D. C. Wunsch II, A Commodity Trading Model based on a NeuralNetwork-Expert System Hybrid, In *Proceedings of the IEEE International Conference onNeural Networks,* (1991), I 289-294.
2. Collard, J. E., Commodity Trading with A Three Year Old, In R. R. Trippi and E. Turban (Eds.), *Neural Networks in Finance and Investing,* Probus Publishing Company, Chicago, 1993, 411-420.
3. Deboeck, G. J., *Trading On the Edge,* John Wiley & Sons, Inc., 1994.
4. Falas, T., A. Charitou, and C. Charalambous, The Application of Artificial Neural Networks in the Prediction of Earning, In *Proceedings of the International Joint Conference on Neural Networks* (1994), 3629-3633.
5. Hawley, D. D., J. D. Johnson, and D. Raina, Artificial Neural Systems : A New Tool for Financial Decision-Making, *Financial Analysts Journal* (Nov.-Dec. 1990), 63-72.
6. Hecht-Nielson, R., *Neurocomputing,* Addison-Wesley Publishing Co., Inc. 1990
7. Jang, G., F. Lai, B. Jiang, and C. Pan, An Intellignet Stock Portfolio Management System Based on Short-term Trend Prediction using Dual-Module Neural Networks, *Artificial Neural Networks* (1991), 447-452.
8. Jhee, W. C. and J. K. Lee, Performance of Neural Networks in Managerial Forecasting, *Intelligent Systems in Accounting, Finance, and Management,* 2 (1993), 55-71.
9. Jo, H., J. Park, and I. Han, A Review of Artificial Intelligence Applications in Financial Market, *In Proceedings of Korea Expert Systems Society,* (1995), 147-158
10. Kamijo, K. and T. Tanigawa, Stock Price Pattern Recognition - A Reccurent Neural Network Approach-, *In Proceedings of the IEEE International Joint Conference on Neural Networks,* (1990), I 215-221.
11. Kim, K. and K. Cho, The Prediction of the Direction of Stock Index using Artificial Neural Network, Investigation Report of Daewoo Secu., Co. (Nov. 1994), 28-36 (In Korean).
12. Kimoto, T., K. Asakawa, M, Yoda, and M. Takeoka, Stock Market Prediction System with Modular Neural

Networks, In *Proceedings of the IEEE International Joint Conference on Neural Networks,* (1990), I 1-6.

13. Kwon, Y., S. Kim, H. Lee, K. Shin, Comparative Analysis of Prediction Model of Korea Composite Stock Price Index, *The Journal of Korea Security Association,* (12, 1989), 375-402 (In Korean).
14. Matsuba, I., Application of Neural Sequential Associator to Long-Term Stock Price Prediction, *IEEE* (1991), 1196-1201.
15. _______, H. Masui, and S. Hebishima, Prediction of Chaotic Time-series Data using Optimized Neural Networks, In *Proceedings of the International Joint Conference on Neural Networks,* Bejing (1992), I 340-345.
16. Mendelsohn, L., Preprocessing Data For Neural Networks, *Technical Analysis of Stock and Commodities,* (Oct 1993), 52-58.
17. Nikolopoulos, C. and P. Fellrath, A Hybrid Expert System for Investment Advising, *IEEE* (1994) 1818-1820.
18. Refenes, A. N., M. Azema-Barac, and A. D. Zapranis, Stock Ranking : Neural Networks Vs Multiple Linear Regression, *In Proceedings of IEEE International Conference on Neural Networks,* Sanfrancisco (1993), 1419-1426.
19. Trippi, R. R. and E. Turban, *Neural Networks in Finance and Investing,* Probus Publishing Company 1993.
20. White, H., Economic Prediction Using Neural Networks : The Case of IBM Daily Stock Returns, In *Proceedings of the IEEE International Conference on Neural Networks,* (1988), II 451-458.
21. Ye, Z. and L. Gu, A Fuzzy System for Trading the Shanghai Stock Market, In G. J. Deboeck (Ed.), *Trading On The Edge,* John Wiley and Sons, Inc., New York, 1994, 207-214.
22. Yoda, M., Predicting the Tokyo Stock Market, In G. J. Deboeck (Ed.), *Trading On The Edge,* John Wiley and Sons, Inc., New York, 1994, 66-79.
23. Yoon, Y. and G. Swales, Predicting Stock Price Performance : A Neural Network Approach, In *Proceedings of the IEEE 24th Annual Conference of Systems Sciences,* (1991), 156-162.