Using Genetic Algorithms to Support Artificial Neural Networks for the Prediction of the Korea Stock Price Index*

Kyoung-jae Kim and Ingoo Han**

Abstract

This paper compares four models of artificial neural networks (ANN) supported by genetic algorithms for the prediction of stock price index. Previous research proposed many hybrid models of ANN and genetic algorithms (GA) in order to train the network, to select the feature subsets, and to optimize the network topologies. Most these studies, however, only used GA to improve a part of architectural factors of ANN. In this paper, GA simultaneously optimizes multiple factors of ANN. Experimental results show that GA approach to simultaneous optimization for ANN (SOGANN3) outperforms the other approaches.

Key words: Simultaneous optimization; Genetic algorithms; Artificial neural networks; The prediction of stock price index

1. Introduction

For a long time, there has been much research interest in predicting the stock price index. Among them, there are many studies using data mining techniques including artificial neural networks (ANN). However, most studies showed that ANN had some limitations in learning the patterns because stock market data has tremendous noise and complex dimensionality. ANN has preeminent learning ability while it is often confronted with inconsistent and unpredictable performance for noisy data. In addition, sometimes the amount of data is so large that the learning of patterns may not work well. Many researchers in the society of data mining are interested in the reduction of dimensionality (Dash and Liu, 1997). The reduction and transformation of the irrelevant architectural factors may shorten the running time and yield more generalized results.

Feature subset, the number of processing elements in hidden layer, the number of hidden layer, and feature transformation are the architectural factors of ANN to be determined in advance (Kim and Han, 2000). These factors were determined by the method of trial and error and the subjectivity of designer. This may lead locally optimized solutions. Connection weights between layers are another architectural factors, but this factor is determined by learning process of ANN. Most research on the application of ANN used gradient descent algorithm to adjust error between predicted value by ANN and actual value. Gradient descent algorithm, however, may perform poorly even on simple problems when predicting the holdout data (Sexton et al., 1998a). Their indication stems from the fact that gradient descent algorithm is a local search algorithm and may tend to fall into a local minimum.

Many studies proposed various kinds of hybrid models for the reduction of irrelevant architectural factors. Lee et al. (1996) used the multiple discriminant analysis (MDA), ID3, and self-organizing feature map to support feature selection for ANN. Tabu search (Sexton et al., 1998a) and simulated annealing (Sexton et al., 1999) are employed as hybridizing method between ANN and heuristic search algorithms. Genetic algorithms (GA) are one of the most popular search algorithms to improve the performance of ANN. GA may relieve the problems associated with the *ad hoc* nature in the design process of ANN. In addition, GA may avoid falling into local minimum when local search method including gradient descent algorithms are used in ANN.

This paper employs four hybrid approaches of ANN and GA to predict future direction of the Korea stock price index (KOSPI). Each approach except the third approach of simultaneous optimization using GA for ANN

* This work is supported by the Korea Science and Engineering Foundation, under Grants No. 98-0102-08-01-3.

** Graduate School of Management,
Korea Advanced Institute of Science and Technology

(SOGANN3) uses GA to optimize different factors of ANN. SOGANN3 simultaneously optimizes all factors which are separately considered in other approaches. Experimental results show that SOGANN3 outperform the other approaches and conventional ANN.

The rest of the paper is organized as follows: The next section reviews research background. Section 3 proposes simultaneous optimization approach using GA for ANN and describes the benefits of the proposed approach. Section 4 describes the research data and experiments. In the fifth section, empirical results are summarized and discussed. In the final section, conclusions and the limitations of this study are presented.

2. Research background

2.1 Prior research on stock market prediction using ANN

Many studies on stock market prediction using artificial intelligence (AI) techniques were performed during the past decade. These studies used various types of ANN to predict accurately the stock index and the direction of its change.

One of the earliest studies, Kimoto et al. (1990) used several learning algorithms and prediction methods for developing the Tokyo stock exchange prices index (TOPIX) prediction system. They used the modular neural network to learn the relationships among various market factors. Kamijo and Tanikawa (1990) used the recurrent neural network and Ahmadi (1990) employed the backpropagation neural network with the generalized delta rule to predict the stock market.

Some researchers investigated the issue of predicting the stock index futures market. Trippi and DeSieno (1992) and Choi et al. (1995) predicted the daily direction of change in the S&P 500 index futures using ANN. Duke and Long (1993) executed daily predictions of the German government bond futures using the backpropagation neural network. The above studies are mainly focused on applications of ANN to the stock market prediction.

Recent research tends to hybridize several AI techniques. Hiemstra (1995) proposed fuzzy expert systems to predict stock market returns. He suggested that ANN and fuzzy logic could capture the complexities of functional mapping because they do not require the specification of the function to approximate. A more recent study of Tsaih et al. (1998) integrated the rule-based technique and ANN to predict the direction of change of the S&P 500 stock index futures on a daily basis. Some researchers tend to include novel factors to the learning process. Kohara et al. (1997) incorporated prior knowledge to improve the performance of stock market prediction.

Some of them, however, did not produce outstanding prediction accuracy partly because of the tremendous noise and non-stationary characteristics in stock market data. Lawrence et al. (1996) pointed out that when the training of ANN tends to be difficult for high noisy data then the networks fall into a naive solution such as always predicting the most common output. Another reason may be the local convergence of the gradient descent algorithms. Most ANN used in prior research employed the gradient descent algorithm to train the network.

2.2 Architectural factors for Designing ANN

Many researchers suggested various kinds of search algorithms to address the limitation of back-propagation neural networks with gradient algorithm. In addition, the optimal network architecture is critical for developing good generation ability of back-propagation learning algorithm, but back-propagation learning algorithm cannot determine the optimal hidden layer size by itself (Tsaih et al., 1998). In the modeling process of ANN, we must determine the following architectural factors:

2.2.1 *Connection weights in the ANN solutions*

Many ANN studies relied on the gradient descent algorithm to get the connection weights of the model. Sexton et al. (1998a) pointed out that the gradient descent algorithm may perform poorly even on simple problems when predicting the holdout data. Their indication stems from the fact that back-propagation is a local search algorithm and may tend to fall into a local minimum.

Sexton et al. (1998b) indicated that the use of the momentum, restarting training at many random points, restructuring the network architecture, and applying significant constraints to the permissible forms can fix it. They also suggested that one of the most promising directions is using global search algorithms to search the weight vector of the network instead of local search algorithms including the gradient descent algorithm. They employed GA to search the weight vector of ANN. The results showed that the GA-derived solution was superior to the corresponding back-propagation solution. Ignizio and Soltys (1996), and Gupta and Sexton (1999) also

suggested that the GA-derived solutions are better than the gradient descent algorithm derived solutions.

Some ANN research advocated that other global search algorithms can improve performance. Sexton et al. (1998a) used tabu search to optimize the network and tabu search-derived solutions were significantly superior to those of back-propagation solutions for test data. In another paper, Sexton et al. (1999) again incorporated the simulated annealing, one of global search algorithms, to optimize the network. They compared GA to the simulated annealing. They concluded that GA outperformed simulated annealing. On the other hand, Shin et al. (1998) concluded that back-propagation with gradient descent algorithm outperformed ANN with GA in their application on bankruptcy prediction. They concluded that GA solution cannot always guarantee better performance than ANN trained with the gradient descent algorithm.

2.2.2 Feature subset selection for ANN

Feature subset selection tries to pick a subset of features that are relevant to the target concept and remove the number of irrelevant or redundant features (Dash and Liu, 1997). Feature subset selection reduces the running time of learning and produces general results within the machine learning techniques.

Among the several methods of feature subset selection, GA is popularly used in data mining application including ANN (Ornes and Sklansky, 1997; Yang et al., 1998), inductive learning (Bala et al., 1995; Vafaie and DeJong, 1998), linear regression (Wallet, 1996). GA selects relevant feature subsets to the specific fitness function of application.

2.2.3 Number of processing elements in hidden layer

Adding additional processing elements in hidden layer may enhance the computational abilities of ANN. But, in some cases, the model does not generalize to unknown data because excessive number of processing elements in hidden layer may learn the insignificant aspects of known data (Hansen, 1998). Some researchers suggested that ANN with a parsimonious topology could yield consistent performance from unknown data. Using too few hidden processing elements, however, may starve the network of the resources it needs to solve the complex problem (Masters, 1993). The average number of input and output units and the method of trial and error have been used to determine the number of processing elements in hidden layer. But, these methods does not guarantee the effectiveness and efficiency of the network architecture (Hansen et al., 1999).

2.2.4 Feature transformation for ANN

Most studies showed that ANN had some limitations in learning the patterns because stock market data has tremendous noise and complex dimensionality. In particular, the existence of continuous data and large amount of data may pose a challenging task to explicit concepts extraction from the raw data due to the huge amount of data space determined by continuous features (Liu and Setiono, 1996).

The data mining society has been interested in feature transformation because data preprocessing is an essential step for knowledge discovery. One of the most popular preprocessing methods is feature transformation. Feature transformation is the process of creating a new set of features (Liu and Motoda, 1998b). It may be split into three categories including feature extraction, feature construction, and feature discretization. Among them, feature discretization is closely related to dimensionality reduction (Liu and Motoda, 1998a). The methods of feature discretization are classified as endogenous versus exogenous. Endogenous methods do not take into consideration the value of the dependent feature while exogenous methods do. (Dougherty et al., 1995; Scott et al., 1997; Susmaga, 1997).

The endogenous methods have the advantage of simplicity in the discretization process. However, they do not consider the association among each independent and dependent feature. The prediction performance is enhanced by the ability of discrimination not only by a single feature but also by the association among features. For this limitation, the endogenous method does not provide an effective way of forming categories (Scott et al., 1997). On the other hand, exogenous methods discretize an independent feature to maximize its association with the values of dependent and other independent features (Kim and Han, 2000).

For ANN, the method of linear transformation is popularly used. Linear transformation means the linear scaling the data to the range of 0.0 to 1.0. Linear transformation is usually used to enhance the performance of ANN because most ANN accept numeric data only in the range of 0.0 to 1.0 or −1.0 to +1.0 (Bigus, 1996). However, this method is simplistic method of feature transformation. In addition, this method is one of the endogenous methods and has the limitations mentioned above.

2.2.5 Other factors

Some researchers suggested that there are still the issues of how many hidden layers there should be and what kind of activation function should be used in ANN. In case of number of hidden layer, more than one hidden layer is almost never beneficial because training often slows dramatically when additional hidden layers are used (Masters, 1993). A few numbers of papers suggested that an activation function is the architectural factor of ANN. But the majority of back-propagation applications used a sigmoid (S-shaped curve) activation function (Hansen et al., 1999). In addition, there are few comparative studies between a sigmoid function and other activation functions in ANN.

Some studies used a binary representation of connection weight. Weight values can be either binary coded or real-valued. Binary connection weight can eliminate irrelevant weights but this may cause significant information loss. A binary representation of weight is unnecessary when the model uses a real-valued representation of weight because the latter includes the former.

3. Simultaneous optimization using GA for ANN (SOGANN)

In this paper, we use GA for optimizing ANN in four ways. GA is selected among various kinds of global search algorithms because GA has been shown to achieve superior solutions to other global search algorithms including simulated annealing and tabu search (Sexton et al., 1998b; Sexton et al., 1999). This study first uses GA to optimize connection weights of ANN. Second, GA supports to select relevant feature subset. Third, GA supports to determine the number of nodes in hidden layer. Fourth, this study adopts feature transformation based on GA. This study simultaneously optimizes above factors in ANN using GA. Although prior research used GA to optimize architectural factors in ANN, but they did not simultaneously considers the above factors. Table 1 summarizes previous studies on hybrid models between GA and ANN.

Table 1. The optimized architectural factors for ANN using GA in prior studies

	Connection weight	Processing elements in hidden layer	Feature subset	Feature transformation	Other factors
Montana and Davis (1989)	O				BW*
Maniezzo (1994)	O	O	O		HL**
Ignizio and Soltys (1996)	O	O	O		
Hansen (1997)		O	O		HL / AF***
Dorsey and Sexton (1998)	O				BW
Sexton et al. (1998b)	O				
Sexton (1998)	O		O		BW
Nissinen et al. (1999)		O			
Sexton et al. (1999)	O				
Hansen et al. (1999)		O	O		HL / AF
Gupta and Sexton (1999)	O				
Kim and Han (2000)	O			O	

*: Binary representation of connection weight / **: Number of hidden layer / ***: Activation function

To verify the effectiveness of proposed approach, we compare four hybrid approaches of ANN and GA and a conventional ANN. The first approach is named as "Conventional ANN (CNN)". This approach uses conventional approaches of back-propagation learning with gradient descent algorithm. The connection weights of this method are assigned by gradient descent algorithm and all features are incorporated as input features. In addition, the number of processing elements in hidden layer are fixed as the number of the incorporated features because back-propagation algorithm does not have the general rules to determine above architectural factors. The transformation method for each feature is linear transformation.

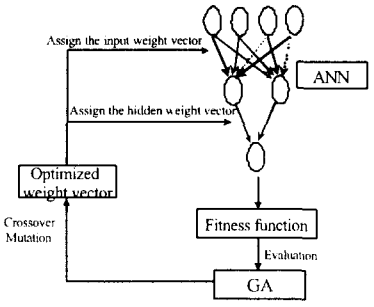
All hybrid approaches employ GA to determine connection weights between layers. In the second

approach, GA simply searches optimal or near-optimal connection weights between layers. Other factors including the method of feature subset selection, the determination method of the number of processing elements in hidden layer, and the method of feature transformation are just alike CNN. This approach is named as "Simple optimization using GA for ANN (SIGANN)". This approach was previously proposed by Montana and Davis (1989), Sexton et al. (1998b), Sexton et al. (1999), Gupta and Sexton (1999), and Kim and Han (2000).

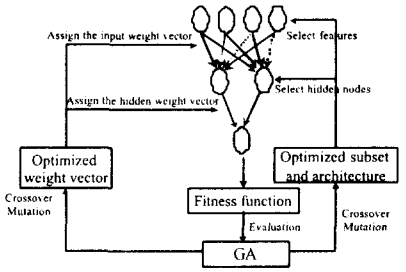
In the third approach, GA tries to optimize topological factors in ANN including feature subset, processing elements in hidden layer in addition to connection weights. This approach was suggested by Ignizio and Soltys (1996). This study names this approach "Simultaneous optimization using GA for ANN 1 (SOGANN1)".

The fourth approach uses GA to optimize feature transformation. This approach is alike SIGANN except for the method of feature transformation. This study names this approach "Simultaneous optimization using GA for ANN 2 (SOGANN2)". This approach was proposed by Kim and Han (2000).

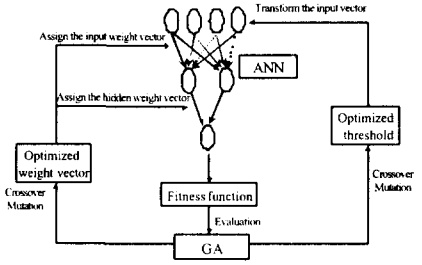
In the fourth approach, GA simultaneously searches to optimize the factors optimized in the above approaches. The fourth approach simultaneously optimizes feature subset, processing elements in hidden layers, and feature transformation in addition to connection weights. This approach is named as "Simultaneous optimization using GA for ANN 3 (SOGANN3)". The frameworks of hybrid approaches are shown in Figure 1.



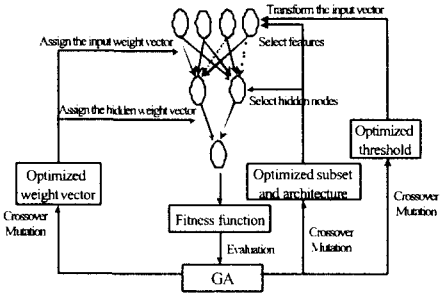
(a) SIGANN



(b) SOGANN1



(c) SOGANN2



(d) SOGANN3

Figure 1. The frameworks of the proposed approaches

Table 2 summarizes above five approaches.

Table 2. Summary of five approaches

	Assigning Connection weights	Feature subset selection	Number of P.E. in hidden layer	Feature transformation
CNN	Gradient algorithm	All available features	Number of input features	Linear scaling
SIGANN	Genetic algorithm	All available features	Number of innut features	Linear scaling
SOGANN1	Genetic algorithm	Genetic algorithm	Genetic algorithm	Linear scaling
SOGANN2	Genetic algorithm	All available features	Number of input features	Genetic algorithm
SOGANN3	Genetic algorithm	Genetic algorithm	Genetic algorithm	Genetic algorithm

Among the above three SOGANNs, this study describes the process of optimization of SOGANN3. The optimization process of other two SOGANNs and SIGANN are alike the SOGANN3 except for some architectural factors in ANN. The process of SOGANN3 consist of following three phases:

Phase 1: In the first phase, GA searches optimal or near-optimal connection weights and all architectural

factors. The populations, the connection weights and all architectural factors in ANN, are initialized into random values before the search process. The parameters for searching must be encoded on chromosomes. This study needs five sets of parameters. The first set is the set of connection weights between the input layer and the hidden layer of the network. The second set is the set of connection weights between the hidden layer and the output layer. As mentioned earlier, above two sets may mitigate the limitation of the gradient descent algorithm. The third parameter set is the selection code for feature subsets. The fourth set is the selection code for processing elements in hidden layer. The fifth parameter set is the thresholds for feature transformation of each feature.

The strings used in this study have the following encoding: This study uses twelve initial input features and employs twelve initial processing elements in the hidden layer. Each processing element in the hidden layer receives twelve signals from the input layer. The first 144 bits represent the connection weights between the input layer and the hidden layer. These bits are searched from –5 to 5. Each processing element in the output layer receives a signal from the hidden layer. The next 12 bits indicate the connection weights between the hidden layer and the output layer. These bits also varied between –5 and 5. The next 12 bits represent selection codes for feature subsets and the following 12 bits are selection codes for processing elements in hidden layer. Above 24 bits are searched between 0 and 1. The following 48 bits are the thresholds for feature transformation. Each feature is discretized into at most five categories and needs four thresholds for discretization. In addition, GA also searches the number of categories to be discretized using these bits. The thresholds are not used if the searched thresholds are more than the maximum value of each feature. The upper limit of the number of categories is five and the lower limit is one. This number is automatically determined by the searching process of GA.

The encoded chromosomes are searched to maximize the fitness function. The fitness function is specific to applications. In this study, the objectives of the model are to approximate connection weights and the optimal architectural factors for the correct solutions. These objectives can be represented by the average prediction accuracy of the training data. This study applies the average prediction accuracy of the training data to the fitness function.

The parameters to be searched use only the information about training data. In this phase, GA operates the process of crossover and mutation on initial chromosomes and iterates until the stopping conditions are satisfied. For the controlling parameters of the GA search, the population size is set to 100 organisms and the crossover and mutation rates are varied to prevent ANN from falling into a local minimum. The range of the crossover rate is set between 0.5 and 0.7 while the mutation rate ranges from 0.05 to 0.1. As the stopping condition, only 5000 trials are permitted.

Phase 2: The second phase is the process of feedforward computation in ANN. In this phase, the sigmoid function is used as the activation function. This function is a popular activation function for the back-propagation neural network because it can easily be differentiated. The linear function is used as a combination function for the feedforward computation with derived connection weights from the first phase.

Phase 3: The derived connection weights and architectural factors are applied to the holdout data. This phase is indispensable to validate the generalizability because ANN has the eminent ability of learning the known data. If this phase is not carried out the model may fall into the problem of overfitting with the training data.

4. Research data and experiments

The research data used in this study is technical indicators and the direction of change in the daily Korea stock price index (KOSPI). The total number of samples is 2,928 trading days, from January 1989 to December 1998. Table 3 gives selected features and their formulas (Achelis, 1995; Choi, 1995; Chang et al., 1996; Edwards and Magee, 1997).

Table 3. Selected features and their formulas

Names of feature	Formulas
Stochastic %K	$\frac{C_t - L_n}{H_n - L_n} \times 100$
Stochastic %D	$\frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n}$
Stochastic slow %D	$\frac{\sum_{i=0}^{n-1} \%D_{t-i}}{n}$
Momentum	$C_t - C_{t-4}$
ROC (rate of change)	$\frac{C_t}{C_{t-n}} \times 100$

LW %R (Larry William's %R)	$\frac{H_n - C_t}{H_n - L_n} \times 100$
A/D Oscillator (accumulation/distribution oscillator)	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Disparity 5 days	$\frac{C_t}{MA_5} \times 100$
Disparity 10 days	$\frac{C_t}{MA_{10}} \times 100$
OSCP (price oscillator)	$\frac{MA_5 - MA_{10}}{MA_5}$
CCI (commodity channel index)	$\frac{(M_t - SM_t)}{(0.015 \times D_t)}$
RSI (relative strength index)	$100 - \frac{100}{1 + \frac{\sum_{i=0}^{n-1} Up_{t-i} / n}{\sum_{i=0}^{n-1} Dw_{t-i} / n}}$

C: Closing price, L: Low price, H: High price, MA: Moving average of price, M_t: $\frac{(H_t + L_t + C_t)}{3}$, SM_t: $\frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$, D_t: $\frac{\sum_{i=1}^{n} |M_{t-i+1} - SM_t|}{n}$, Up / Dw: Upward / Upward price change

The direction of daily change in the stock price index are categorized as "0" or "1". "0" means that the next day's index is lower than the today's index, and "1" means that the next day's index is higher than today's index. We select twelve technical indicators as feature subsets by the review of domain experts and prior research.

About 20% of the data is used for holdout and 80% for training. The training data is used to search the optimal or near-optimal parameters and is employed to evaluate the fitness function. The holdout data is used to test the results with the data that is not utilized to develop the model. The number of cases for each set is shown in Table 4.

Table 4. Number of cases

Set	Year										Total
	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	
Training	232	233	234	236	237	237	235	235	234	234	2,347
Holdout	57	58	58	58	59	59	58	58	58	58	581
Total	289	291	292	294	296	296	293	293	292	292	2,928

5. Experimental results

Four hybrid GA and ANN models and a conventional ANN are compared. Table 5 describes the average prediction accuracy of each model.

Table 5. Average predictive performance (hit ratio: %)

Year	CNN		SIGANN		SOGANN1		SOGANN2		SOGANN3	
	Training	Holdout	Training	Holdout	Training	Holdout	Training	Holdout	Training	Holdout
1989	59.05	48.28	57.33	49.12	67.24	54.39	68.10	59.65	66.38	63.16
1990	62.23	49.15	59.23	56.90	65.24	53.45	66.95	60.34	70.82	62.07
1991	58.97	53.45	53.42	50.00	66.67	56.90	63.25	56.90	62.39	70.69
1992	61.02	51.72	60.17	44.83	69.07	46.55	66.95	58.62	65.68	56.90
1993	54.01	44.07	54.43	44.07	60.34	66.10	67.09	61.02	67.51	66.10
1994	62.45	64.41	61.18	59.32	67.09	62.71	63.29	62.71	64.14	64.41
1995	63.83	44.83	63.83	53.45	71.49	62.07	69.36	65.52	71.06	70.69
1996	61.28	60.35	61.70	50.00	65.11	70.69	64.26	67.24	62.13	77.59
1997	46.15	50.00	50.43	50.00	64.96	58.62	64.10	62.07	63.68	63.79
1998	55.98	51.72	56.84	48.28	61.11	53.45	64.53	62.07	63.68	65.52
Total	58.50 %	51.81 %	57.86 %	50.60 %	65.83%	58.52%	65.79 %	61.70 %	66.75%	66.09%

In Table 5, SOGANN3 outperforms the other approaches for the holdout data. In addition, three SOGANN approaches outperform CNN and SIGANN. It appears that simultaneous optimization approach allows better to learn noisy patterns than conventional and simple optimization approaches. SOGANN1 and SOGANN2 produces similar performance for training data but SOGANN2 outperforms SOGANN1 for holdout data. This result may be caused by the fact that the globally searched feature transformation supports the learning process and eliminates the irrelevant patterns. This prevents the network from falling into the problem of overfitting and may enhance the generalizability. This result is also supported by SOGANN3. SOGANN3 slightly outperforms SOGANN2 for training data but SOGANN3 outperforms SOGANN2 by 4.39% for holdout data. This result partly supports the fact that enhanced prediction performance can be produced through the simultaneous optimization of additional architectural factors by GA.

From Table 5, we find that the average prediction performances of CNN and SIGANN are similar. Although Sexton et al. (1998b) concluded that ANN with the genetically evolved connection weights outperforms conventional back-propagation neural networks with the gradient descent algorithm, this study does not find evidence to support their conclusions. The reasons for this result may be summarized in two points. First is that there is a generic limitation of global search algorithms. Sexton et al. (1999) reported that performance with the connection weights searched by simulated annealing, one of global search algorithms, was lower than the performance of back-propagation with the gradient descent algorithm. Their result was also supported by Shin et al. (1998). They concluded that the reason of the results came from the fact that GA may be less competent in a local search. Although a global search is more desirable than a local search for learning ANN, sometimes a local search is also needed. The other factor may be a complex dimensionality in data. GA is a global search algorithm, however, financial data including the stock market data is too complex to be searched easily. It is necessary to reduce the dimensionality of data and irrelevant factors before searching.

The McNemar tests are used to examine whether SOGANN3 significantly outperforms the other four models. This test is a nonparametric test for two related samples. This test may be used with nominal data and is particularly useful with before-after measurement of the same subjects (Cooper and Emory, 1995). Table 6 shows the results of the McNemar test to compare the performance for the holdout data.

Table 6. McNemar values for the pairwise comparison of performance among methods

	SIGANN	SOGANN1	SOGANN2	SOGANN3
CNN	0.227	7.282**	15.593**	30.096**
SIGANN		8.804**	19.267**	34.439**
SOGANN1			1.741	10.626**
SOGANN2				4.112*

(* significant at 5% level, ** significant at 1% level)

As shown in Table 6, SOGANN3 outperforms CNN, SIGANN, and SOGANN1 at 1% statistical significance level. For SOGANN2, SOGANN3 significantly outperforms SOGANN2 at 5% level. In addition, Table 6 shows that SOGANN2 outperforms CNN and SIGANN at 1% statistical significance level but does not significantly outperform SOGANN1. SOGANN1 also performs better than CNN and SIGANN at 1% statistical significance level. CNN and SIGANN do not significantly outperform each other.

6. Concluding remarks

As mentioned earlier, previous studies tried to optimize architectural factors and connect weights of ANN using global search algorithms. Their studies, however, only focus on the optimization of the limited factors of ANN. This may lead locally optimized solutions. Some of them focused on the optimization of the connection weights of ANN. Others had an interest in the optimization of the learning algorithms itself, but most studies had little interest in the dimensionality reduction and the elimination of irrelevant patterns.

This paper has proposed a new hybrid GA and ANN to mitigate the above limitations. In this paper, GA simultaneously searches for the optimal or near-optimal solutions of connection weights in the learning algorithm and architectural factors in ANN and the thresholds of feature transformation for the dimensionality reduction. We conclude that SOGANN3 simultaneously optimizes

multiple architectural factors and connection weights in ANN then enhances the prediction accuracy and the generalizability of classifier from the empirical results.

This study has some limitations. One of the limitations is that the objects for optimization are focused on only four factors of the learning process of ANN. SOGANN3 produces valid results in this study. However, GA can potentially be used to optimize other factors of learning parameters including activation function simultaneously. We also believe that there is great potential for further research with the simultaneous optimization approach using GA for other AI techniques including case-based reasoning and decision tree.

References

Achelis, S.B. (1995). *Technical analysis from A to Z.* Chicago: Probus Publishing.

Ahmadi, H. (1990). Testability of the arbitrage pricing theory by neural networks. *Proceedings of the International Conference on Neural Networks* (pp. 385-393). San Diego, California.

Bala, J., Huang, J., Vafaie, H., DeJong, K., & Wechsler, H. (1995). Hybrid learning using genetic algorithms and decision trees for pattern classification. *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 19-25).

Bigus, J.P. (1996). *Data mining with neural networks.* New York: McGraw-Hill.

Blum, A.L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*(1-2), 245-271.

Chang, J., Jung, Y., Yeon, K., Jun, J., Shin, D., & Kim, H. (1996). *Technical indicators and analysis methods.* Seoul: Jinritamgu Publishing.

Choi, J. (1995). *Technical indicators.* Seoul: Jinritamgu Publishing.

Choi, J.H., Lee, M.K., & Rhee, M.W. (1995). Trading S&P 500 stock index futures using a neural network. *Proceedings of the Third Annual International Conference on Artificial Intelligence Applications on Wall Street* (pp. 63-72). New York.

Cooper, D.R., & Emory, C.W. (1995). *Business research methods.* Chicago, Illinois: Irwin.

Dash, M., & Liu, H. (1997). Feature selection methods for classifications. *Intelligent Data Analysis-An International Journal, 1*(3), 131-156.

Duke, L.S., & Long, J.A. (1993). Neural network futures trading - A feasibility study. In Society for Worldwide Interbank Financial Telecommunications, *Adaptive intelligent systems* (pp. 121-132). Amsterdam: Elsevier Science Publishers.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 194-202). San Francisco, California.

Dorsey, R., & Sexton, R. (1998). The use of parsimonious neural networks for forecasting financial time series. *Journal of Computational Intelligence in Finance, 6*(1), 24-31.

Edwards, R.D., & Magee, J. (1997). *Technical analysis of stock trends.* Chicago, Illinois: John Magee.

Gupta, J.N.D., & Sexton, R.S. (1999). Comparing backpropagation with a genetic algorithm for neural network training. *Omega, 27*(6), 679-684.

Hansen, J.V. (1998). Comparative performance of backpropagation networks designed by genetic algorithms and heuristics. *International Journal of Intelligent Systems in Accounting, Finance and Management, 7*(2), 69-79.

Hansen, J.V., McDonald, J.B., & Nelson, R.D. (1999). Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models. *Computational Intelligence, 15*(3), 171-184.

Hiemstra, Y. (1995). Modeling structured nonlinear knowledge to predict stock market returns, In R. R. Trippi, *Chaos & nonlinear dynamics in the financial markets: theory, evidence and applications* (pp. 163-175). Chicago, Illinois: Irwin.

Ignizio, J.P., & Soltys, R. (1996). Simultaneous design and training of ontogenic neural network classifiers. *Computers and Operations Research, 23*(6), 535-546.

Kamijo, K., & Tanigawa, T. (1990). Stock price pattern recognition: A recurrent neural network approach. *Proceedings of the International Joint Conference on Neural Networks* (pp. 215-221). San Diego, California.

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 19(2), forthcoming.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural network. *Proceedings of the International Joint Conference on Neural Networks* (pp. 1-6). San Diego, California.

Kohara, K., Ishikawa, T., Fukuhara, Y., & Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. *International Journal of Intelligent Systems in Accounting, Finance and Management, 6*(1), 11-22.

Lawrence, S., Tsoi, A.C., & Giles, C.L. (1996). Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference. In Institute for Advanced computer Studies, *Technical report UMIACS-TR-96-27 and CS-TR-3625*. University of Maryland.

Lee, K.C., Han, I., & Kwon, Y. (1996). Hybrid neural network models for bankruptcy predictions. *Decision Support Systems*, 18(1), 63-72.

Liu, H., & Motoda, H. (1998a). *Feature selection for knowledge discovery and data mining.* Norwell, Massachusetts: Kluwer Academic Publishers.

Liu, H., & Motoda, H. (1998b). Feature transformation and subset selection. *IEEE Intelligent Systems and Their Applications, 13*(2), 26-28.

Liu, H., & Setiono, R. (1996). Dimensionality reduction via discretization. *Knowledge-Based Systems, 9*(1), 67-72.

Maniezzo, V. (1994). Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks, 5*(1), 39-53.

Masters, T. (1993). *Practical Neural Network Recipes in C++*. Boston: Academic Press.

Montana, D., & Davis, L. (1989). Training feedforward neural networks using genetic algorithms. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 762-767). Detroit.

Nissinen, A.S., Koivo, H.N., & Koivisto, H. (1999). Optimization of neural network topologies using genetic algorithm. *Intelligent Automation and Soft Computing, 5*(3), 211-224.

Ornes, C., & Sklanski, J. (1997). A neural network that explains as well as predicts financial market behavior. *Proceedings of the IEEE/IAFE* (pp. 43-49).

Scott, P.D., Williams, K.M., & Ho, K.M. (1997). Forming categories in exploratory data analysis and data mining. In X. Liu, P. Cohen, & M. Berthold, *Advances in intelligent data analysis* (pp. 235-246). Berlin: Springer-Verlag.

Sexton, R.S. (1998). Identifying irrelevant input variables in chaotic time series problems: Using genetic algorithm for training neural networks. *Journal of Computational Intelligence in Finance, 6*(5), 34-41.

Sexton, R.S., Alidaee, B., Dorsey, R.E., & Johnson, J.D. (1998a). Global optimization for artificial neural networks: A tabu search application. *European Journal of Operational Research, 106*(2-3), 570-584.

Sexton, R.S., Dorsey, R.E., & Johnson, J.D. (1998b). Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation. *Decision Support Systems, 22*(2), 171-185.

Sexton, R.S., Dorsey, R.E., & Johnson, J.D. (1999). Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. *European Journal of Operational Research, 114*(3), 589-601.

Shin, K., Shin, T., & Han, I. (1998). Neuro-genetic approach for bankruptcy prediction: A comparison to back-propagation algorithms. *Proceedings of the International Conference of the Korea Society of Management Information Systems* (pp. 585-597). Seoul, South Korea.

Susmaga, R. (1997). Analyzing discretizations of continuous attributes given a monotonic discrimination function. *Intelligent Data Analysis-An International Journal, 1*(3), 157-179.

Trippi, R.R., & DeSieno, D. (1992). Trading equity index futures with a neural network. *The Journal of Portfolio Management, 19*, 27-33.

Tsaih, R., Hsu, Y., & Lai, C.C. (1998). Forecasting S&P 500 stock index futures with a hybrid AI system. *Decision Support Systems, 23*(2), 161-174.

Vafaie, H., & De Jong, K. (1998). Feature space transformation using genetic algorithms. *IEEE Intelligent Systems and Their Applications, 13*(2), 57-65.

Wallet, B.C., Marchette, D.J., Solka, J.L., & Wegman, E.J. (1996). A genetic algorithm for best subset selection in linear regression. *Proceedings of the 28th Symposium on the Interface.*

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and Their Applications, 13*(2), 44-49.