ELSEVIER

Fuzzy Sets and Systems 118 (2001) 21–45

FUZZY sets and systems

www.elsevier.com/locate/fss

An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network

R.J. Kuo[a, *], C.H. Chen[b], Y.C. Hwang[c]

[a] *Department of Industrial Engineering, National Taipei University of Technology, Taipei 106, Taiwan*
[b] *Department of Finance, I-Shou University, Kaohsiung County, Taiwan 840, Taiwan*
[c] *Department of Systems Engineering, Chin-Wei Computer Company, Taipei, Taiwan*

Received January 1998; received in revised form August 1998

Abstract

The stock market, which has been investigated by various researchers, is a rather complicated environment. Most research only concerned the technical indexes (quantitative factors), instead of qualitative factors, e.g., political effect. However, the latter plays a critical role in the stock market environment. Thus, this study develops a genetic algorithm based fuzzy neural network (GFNN) to formulate the knowledge base of fuzzy inference rules which can measure the qualitative effect on the stock market. Next, the effect is further integrated with the technical indexes through the artificial neural network (ANN). An example based on the Taiwan stock market is utilized to assess the proposed intelligent system. Evaluation results indicate that the neural network considering both the quantitative and qualitative factors excels the neural network considering only the quantitative factors both in the clarity of buying-selling points and buying-selling performance. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Stock market; Forecasting; Decision support system; Artificial neural networks; Fuzzy neural networks; Genetic algorithms

Nomenclature

p	the sample number
X_p	the input vector of sample p
T_p	the target vector of sample p
O_{pk}	the output of kth output node
O_{ph}	the output of hth hidden node
W_{ih}	the connection weight from ith input node to hth hidden node
W_{hk}	the connection weight from hth hidden node to kth output node
Net_{pk}	the net internal activity level of kth output node
Net_{ph}	the net internal activity level of hth hidden node
Θ_j	the bias of jth output node
E_p	the cost function for sample p

* Corresponding author. Tel.: +886 2 27712171.
E-mail address: f10868@ntut.edu.tw (R.J. Kuo).

PII: S0165-0114(98)00399-6

E_{ps}	the cost function of s-level's α-cut set for sample p
E^{L}_{pks}	the cost function of the lower boundary for s-level's α-cut set of sample p
E^{L}_{pks}	the cost function of the upper boundary for s-level's α-cut set of sample p
$\bar{X}_p$	the fuzzy input for sample p
$\bar{O}_p$	the fuzzy output for sample p
$\bar{W}_{ih}, \bar{W}_{hk}$	the fuzzy weights
$\bar{\Theta}_h, \bar{\Theta}_k$	the fuzzy biases
η	the learning rate
α	the momentum term
$\bar{*}[\alpha]^{\mathrm{L}}$, $\bar{*}[\alpha]^{\mathrm{U}}$	the lower limit and the upper limit of the α-cut of fuzzy number

1. Introduction

The stock market is one of the most popular investments owing to its high expected profit. However, the higher the expected profit, the higher is the risk implied. Thus, numerous investigations gave rise to different decision support systems for the sake of providing the investors with an optimal prediction. Conventional research addressing this research problem have generally employed the time series analysis techniques (i.e., mixed auto regression moving average (ARMA)) [23] as well as multiple regression models. Most only consider the quantitative factors, like technical indexes. Recently, artificial intelligence techniques, like artificial neural networks (ANNs) and genetic algorithms (GAs), were applied in this area, the above mentioned concern still exists [1,30]. However, a number of qualitative factors, e.g., macro-economical or political effects, may seriously influence the stock tendency. Even the investigators' psychology could result in the index oscillation as well. Therefore, confining ourselves to some technical indexes to predict the stock tendency is not enough in such a complicated environment. It involves why the "experienced" stock experts, or stock brokers, can make more accurate decision than the common investors, since they do not only consider the technical indexes but also consider qualitative factors based on their experienced knowledge. Thus, it turns out to be significant to capture this unstructured knowledge.

Fuzzy logic has been applied in the area of control and has shown highly promising results. This concept attempts to capture experts' knowledge, which is vague. Hence, fuzzy logic appears to be a rather promising candidate for simulating the stock experts' knowledge. However, the setup of experts' knowledge is quite subjective which results from the membership functions setup of fuzzy sets. Thus, this study aims to develop a learning algorithm for fuzzy logic, consisting in a genetic algorithm based fuzzy neural network (GFNN), to capture the stock market experts' knowledge.

Basically, this study develops an intelligent stock trading decision support system based on both quantitative and qualitative factors to assist the investors to make the right decision. The proposed system consists of (1) factors identification, (2) qualitative model (GFNN), and (3) decision integration (ANN). In the initial part, the system collects the factors, no matter quantitative or qualitative, which may influence the stock market. Then, GFNN organizes the experts' knowledge, or qualitative effect on the stock market, and forms a knowledge base with fuzzy inference rules. Finally, the stock market technical indexes are integrated with the qualitative effect on the stock market from GFNN using a feedforward neural network with error backpropagation (EBP) learning algorithm to provide the final decision.

The rest of this work is organized as follows. Section 2 presents the general background. Section 3 discusses the proposed system while evaluation results and discussion based on the Taiwan stock market are summarized in Section 4. Concluding remarks are finally made in Section 5.

2. Background

This section describes the general background of ANN in stock trading system, fuzzy neural networks, and genetic algorithms which will be employed in the proposed system.

2.1. Artificial neural networks (ANNs) in stock trading

Artificial neural network (ANN) is a system derived through models of neuropshychology [9]. In general, it consists of a collection of simple nonlinear computing elements whose inputs and outputs are tied together to form a network. Recently, owing to increase of the computational speed, artificial neural networks (ANNs) have also been applied in many areas, e.g., control, image processing and forecasting [5,29].

For stock market, Schoneburg [32] analyzed the possibility of predicting stock prices on a short-term, day-to-day basis with the help of neural networks by studying three important German stocks selected at random. Baba and Kozaki [1], applied modified EBP learning algorithm to predict the Japanese stock market. The network structure consists of fifteen input nodes and one output node representing the stock market tendency. Jang et al. [13], employed dual adaptive structure neural network to predict the stock tendency for the Taiwan stock market. The simulation findings revealed that the performance of the adaptive structure network is better than the fixed structure network. Similar research using ANN in finance can refer to [3–5,8,17,36]. A new genetic-algorithm-based system was recently presented and applied to predict the future performances of individual stocks [30]. The proposed system was compared to an established neural network. The finding revealed that genetic-algorithm-based system outperforms neural network. Though most of this research declared that their systems can provide very good results, yet the qualitative factors were still not included. However, the qualitative factors sometimes have the most important effects on the stock market.

2.2. *Fuzzy neural networks (FNNs)*

An ANN, as employed for recognition purposes, generally lacks the ability to be developed for a given task within a reasonable time. On the other hand, fuzzy modeling [24,25,37], as applied to fuse the decisions from the different variables, requires an approach to learn from experience (i.e., data collected in advance). ANNs, fuzzy logic, and genetic systems constitute the three independent research fields regarding sixth generation systems (SGS). ANNs and the fuzzy model have been used in many application areas [24,25,29], each pairing its own merits and disadvantages. Therefore, how to successfully combine these two approaches, ANNs and fuzzy modeling, is a relevant concern for further studies.

Generally, the traditional fuzzy system mentioned above is based on experts' knowledge. However, it is not very objective. Besides, it is very difficult to acquire robust knowledge and find available human experts [15]. Recently, ANN's learning algorithm has been applied to improve the performance of a fuzzy system and shown to be a new and promising approach. Takagi and Hayashi [34] introduced a feedforward ANN into fuzzy inference. Each rule is represented by an ANN while all the membership functions are represented by only one ANN. The algorithm is divided into three major parts: (1) the partition of inference rules; (2) the identification of IF parts; and (3) the identification of THEN parts. Since each rule and all the membership functions are represented by different ANNs, they are trained separately. In other words, the parameters cannot be updated concurrently.

Jang [14–16] proposed a method which transforms the fuzzy inference system into a functional equivalent adaptive network, and then employs the EBP-type algorithm to update the premise parameters and least square method to identify the consequence parameters. Meanwhile, Wang and Mendel [35], Shibata et al. [33], and Fukuda and Shibata [6] also presented similar methods. Nakayama et al. [31] proposed a so-called FNN (fuzzy neural network) which has a special structure for realizing a fuzzy inference system. Each membership function consists of one or two sigmoid functions for each inference rule. Due to the lack of a membership function setup procedure, the rule determination and membership function setup are decided by so-called experts where the decision is very subjective. Lin and Lee [26,27] proposed the so-called neural-network-based fuzzy logic control system (NN-FLCS). They introduced the low-level learning power of neural networks in the fuzzy logic system and provided high-level human-understandable meaning to the normal connectionist architecture. Also, Kuo and Cohen [20] introduced a feedforward ANN into fuzzy inference represented by Takagi's fuzzy modeling and applied it to multi-sensor integration. Buckley and Hayashi [2] surveyed recent findings on learning algorithms and applications for FNNs. Furthermore, Buckley introduced several methods in the EBP learning algorithms.

The above-mentioned FNNs are only appropriate for numerical data. However, the experts' knowledge is always of a fuzzy type. Thus, some researchers have attempted to address this problem. Ishibuchi et al. [11] and Ishibuchi et al. [10], proposed learning methods of neural networks to utilize not only numerical data but also expert knowledge represented by fuzzy if–then rules. Recently, Kuo and Xue [19,22] proposed a novel fuzzy neural network whose inputs, outputs, and weights are all asymmetric Gaussian functions. The learning algorithm is EBP-type learning procedure. Lin and Lee [28] also presented a FNN, capable of handling both the fuzzy inputs and outputs.

2.3. Genetic algorithms (GAs)

Genetic algorithms (GAs) are general purpose, search algorithms for solving complex problems. Based on the mechanics of natural selection and natural genetics, GAs work by repeatedly modifying a population of artificial structures through the application of genetic operators. Fitness information, instead of Gradient information, is the only requirement for GAs. GAs can be applied in optimization or classification. The advantages of GAs over conventional parameter optimization techniques are that they are appropriate for the ill-behaved problem, highly nonlinear spaces for global optima and adaptive algorithm [7].

The parameters we are searching for are binary coded. These binary numbers are combined together as a string, or structure. It is called a chromosome and each bit of the chromosome is treated as a gene. Next, the GA starts with a population of n randomly generated structures, where each structure encodes a solution to the task at hand. The GA further processes a fixed number of generations, or until it satisfies some stopping criterion, by using three operators, selection, crossover, and mutation sequentially. The structure with the optimum, or largest, fitness value of the last population is selected.

In the GA, reproduction is implemented by a selection operator. Selection is the population improvement or "survival of the fittest" operator. It duplicates structures with higher fitness values and deletes structures with lower fitness values. The probability of being duplicated for each gene is defined as

$$PS_i = \frac{f_i}{\sum_{i=1}^{s} f_i}, \tag{1}$$

where f_i denotes the fitness function value of ith chromosome and s is the population number.

Crossover, when combined with selection, yields good components of good structures combining to yield even better structures. Crossover forms $n/2$ pairs of parents if the number of population is n. Each pair produces two offspring structures to the mutation stage. The offspring are the outcomes of cutting and splicing the parent structures at various randomly selected crossover points. The approaches for selecting crossover points are one-point crossover, two-point crossover, and uniform crossover.

Mutation creates new structures that are similar to current ones. With a small, pre-specified probability, mutation randomly alters each component of each structure. The reason for using mutation is to prevent missing some significant information during reproduction and crossover. This procedure would avoid the local minimum.

3. Methodology

This study develops an intelligent stock trading decision support system based on the viewpoint of systems integration. The proposed system (see Fig. 1) consists of three parts: (1) factors identification, (2) qualitative model, and (3) decision integration. In the following, each part is thoroughly discussed.

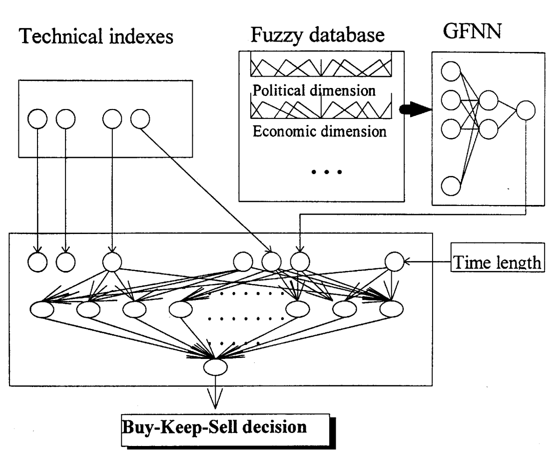

Fig. 1. The structure of the forecasting system.

3.1. Factors identification

To enhance accuracy of the decision, collecting the effective information regarding the forecasted object is highly prominent. In this aspect of the study, it is assumed that the collected factors are complete and adequate to support the decision support system model. Both the quantitative and non-quantitative, even vague, factors are identified from the related references.

3.2. Qualitative model (GFNN)

The proposed genetic-algorithm-based fuzzy neural network (GFNN) is employed herein to capture the stock experts' knowledge and forms a fuzzy database. Since both the inputs and outputs of GFNN are fuzzy numbers, the fuzzy Delphi method is employed to capture the stock experts' knowledge and transformed to the acceptable format of GFNN. For the detailed procedures of fuzzy Delphi, refer to Kuo et al. [21].

Basically, the objective of using genetic algorithm is to avoid local minima. Thus, the genetic algorithm is implemented first in order to obtain a "rough" solution. Then the FNN is employed to fine-tune the results. The derivation of GFNN is as follows.

3.2.1. GA integrated with FNN

This study founded by integrating GA with FNN is that GA can provide the initial weights for the FNN. This does not only decrease the training time but also avoid the local minimum. The procedures of GA used in this study are as follows:

Step 1. Generate n structures of population randomly and set up the number of generation and fitness function.

Step 2. Assess the fitness function value for each chromosome.

Step 3. Process the chromosome operators, selection, crossover, and mutation.

Step 4. Evaluate the fitness function for each new chromosome.

Step 5. Eliminate the chromosomes with lower fitness function values and add the new chromosomes with higher fitness function values.

Step 6. If the stop criterion is satisfied, stop; otherwise, go back to Step 3.

In this study, the fitness function is defined as

$$F = \frac{N}{\sum_{i=1}^{N} (T_i - Y_i)^2}, \tag{2}$$

where N denotes the number of the populations, T_i represents the ith desired output, and Y_i is the ith actual output. The coding method applied is the binary coding, since it is most used. For example, 12 can be represented as 00001100 for 8-digit value on the basis of 2. The number of populations is set to be 50 in this study.

3.2.2. FNN architecture

The fact that the FNN architecture is based on the fuzzy logic which possesses both the precondition, accounts for why the precondition variables represent the effective factors, meanwhile, the stock market tendency represents the consequence variable. This component intends to modify the work of Ishibuchi et al. [10]. In this work, the input, weight, and output fuzzy numbers are symmetric triangular. However, symmetric triangular membership function may cause slow convergence [19,22]. Thus, this work replaces the triangular fuzzy numbers with asymmetric Gaussian functions. The input–output relation of the proposed FNN is discussed in the following. However, the operations on fuzzy numbers are presented first.

3.2.2.1. Operations on fuzzy numbers. Before describing the FNN architecture, fuzzy numbers and fuzzy number operations are defined by the extension principle. In the proposed algorithm, real numbers and fuzzy numbers are denoted by the lowercase letters (e.g., $a, b, \ldots$) and a bar placed over uppercase letters (e.g., $\bar{A}, \bar{B}, \ldots$), respectively.

Since input vectors, connection weights and output vectors of multi-layer feedforward neural networks are fuzzified in the proposed FNN, the addition, multiplication and nonlinear mapping of fuzzy numbers are necessary for defining the proposed FNN. Thus, they are defined as follows:

$$\begin{aligned}\bar{Z}(z) &= (\bar{X} + \bar{Y})(z) \\ &= \max\{\bar{X}(x) \wedge \bar{Y}(y) \mid z = x + y\},\end{aligned} \tag{3}$$

$$\begin{aligned}\bar{Z}(z) &= (\bar{X} \cdot \bar{Y})(z) \\ &= \max\{\bar{X}(x) \wedge \bar{Y}(y) \mid z = x \cdot y\},\end{aligned} \tag{4}$$

$$\bar{Z}(z) = \max\{\overline{Net}(x) \mid z = f(x)\}, \tag{5}$$

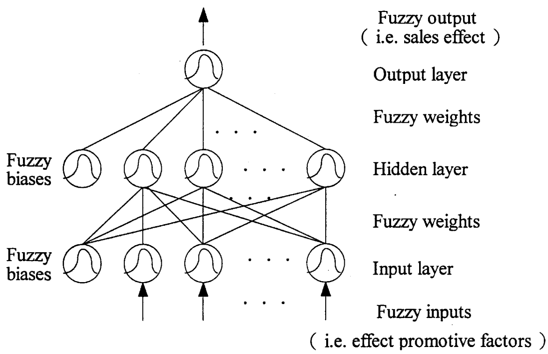

Fig. 2. The FNN architecture.

where $\bar{X}, \bar{Y}, \bar{Z}$ are fuzzy numbers, $*(a)$ denotes the membership function value of each fuzzy number as the α-cut value is a, $\wedge$ is the minimum operator, and $f(x) = (1+\exp(-x))^{-1}$ is the activation function of hidden units and output units of the proposed FNN. The α-cut of fuzzy number $\bar{X}$ is defined as

$$\bar{X}[\alpha] = \{x \mid \bar{X}(x) \geqslant \alpha,\ x \in R\} \quad \text{for } 0 < \alpha \leqslant 1,$$

where $\bar{X}[\alpha]$ represents $\bar{X}[\alpha] = [\bar{X}[\alpha]^{\mathrm{L}}, \bar{X}[\alpha]^{\mathrm{U}}]$ and $\bar{X}[\alpha]^{\mathrm{L}}$ and $\bar{X}[\alpha]^{\mathrm{U}}$ are the lower bound and the upper bound of the α-cut set, respectively.

FNN learning algorithm

The proposed FNN learning algorithm is similar to error backpropagation (EBP)-type learning algorithm. Before discussing the algorithm, some assumptions should be clarified as follows:

(1) Fuzzify a three-layer feedforward neural network with n_I input units, n_H hidden units, and n_O output units (i.e., input vectors, target vectors connection weights and thresholds are fuzzified).

(2) The input vectors are non-negative fuzzy numbers.

(3) These fuzzy numbers are asymmetric Gaussian-shaped fuzzy numbers.

The input–output relation of the proposed FNN (see Fig. 2) is defined by the extension principle [10] and can be written as follows:

Input layer:

$$\bar{O}_{pi}[\alpha] = \bar{X}_{pi}[\alpha], \quad i = 1,2,\ldots,n_I. \tag{6}$$

Hidden layer:

$$\bar{O}_{ph}[\alpha] = f(\overline{Net}_{ph}[\alpha]), \quad h = 1,2,\ldots,n_H, \tag{7}$$

$$\overline{Net}_{ph}[\alpha] = \sum_{i=1}^{n_I} \bar{W}_{hi}[\alpha] \cdot \bar{O}_{pi}[\alpha] + \bar{\Theta}_h[\alpha]. \tag{8}$$

Output layer:

$$\bar{O}_{pk}[\alpha] = f(\overline{Net}_{pk}[\alpha]), \quad k = 1,2,\ldots,n_O, \tag{9}$$

$$\overline{Net}_{pk}[\alpha] = \sum_{k=1}^{n_O} \bar{W}_{kh}[\alpha] \cdot \bar{O}_{ph}[\alpha] + \bar{\Theta}_k[\alpha]. \tag{10}$$

From Eqs. (7)–(10), the α-cut sets of the fuzzy output $\bar{O}_{pk}$ are calculated from the fuzzy inputs, fuzzy weights, and fuzzy biases. If the α-cut set of the fuzzy outputs $\bar{O}_{pk}$ is required, then the above relation can be rewritten as follows:

Input layer:

$$\begin{aligned}\bar{O}_{pi}[\alpha] &= [\bar{O}_{pi}[\alpha]^{\mathrm{L}}, \bar{O}_{pi}[\alpha]^{\mathrm{U}}] \\ &= [\bar{X}_{pi}[\alpha]^{\mathrm{L}}, \bar{X}_{pi}[\alpha]^{\mathrm{U}}], \quad i = 1,2,\ldots,n_I.\end{aligned} \tag{11}$$

Hidden layer:

$$\begin{aligned}\bar{O}_{ph}[\alpha] &= [\bar{O}_{ph}[\alpha]^{\mathrm{L}}, \bar{O}_{ph}[\alpha]^{\mathrm{U}}] \\ &= [f(\overline{Net}_{ph}[\alpha]^{\mathrm{L}}), f(\overline{Net}_{ph}[\alpha]^{\mathrm{U}})], \\ &\qquad h = 1,2,\ldots,n_H,\end{aligned} \tag{12}$$

$$\begin{aligned}\overline{Net}_{ph}[\alpha]^{\mathrm{L}} = &\sum_{\substack{i=1 \\ \bar{W}_{hi}[\alpha]^{\mathrm{L}} \geqslant 0}}^{n_I} \bar{W}_{hi}[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{L}} \\ &+ \sum_{\substack{i=1 \\ \bar{W}_{hi}[\alpha]^{\mathrm{L}} < 0}}^{n_I} \bar{W}_{hi}[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}} + \bar{\Theta}_h[\alpha]^{\mathrm{L}},\end{aligned} \tag{13}$$

$$\begin{aligned}\overline{Net}_{ph}[\alpha]^{\mathrm{U}} = &\sum_{\substack{i=1 \\ \bar{W}_{hi}[\alpha]^{\mathrm{U}} \geqslant 0}}^{n_I} \bar{W}_{hi}[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}} \\ &+ \sum_{\substack{i=1 \\ \bar{W}_{hi}[\alpha]^{\mathrm{U}} < 0}}^{n_I} \bar{W}_{hi}[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{L}} + \bar{\Theta}_h[\alpha]^{\mathrm{U}}.\end{aligned} \tag{14}$$

Output layer:

$$\begin{aligned}\bar{O}_{pk}[\alpha] &= [\bar{O}_{pk}[\alpha]^{\mathrm{L}}, \bar{O}_{pk}[\alpha]^{\mathrm{U}}] \\ &= [f(\overline{Net}_{pk}[\alpha]^{\mathrm{L}}), f(\overline{Net}_{pk}[\alpha]^{\mathrm{U}})], \\ &\qquad k = 1,2,\ldots,n_O,\end{aligned} \tag{15}$$

$$\overline{Net}_{pk}[\alpha]^{\mathrm{L}} = \sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{L}} \geqslant 0}}^{n_O} \bar{W}_{kh}[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}} + \sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{L}} < 0}}^{n_O} \bar{W}_{kh}[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} + \bar{\Theta}_k[\alpha]^{\mathrm{L}}, \tag{16}$$

$$\overline{Net}_{pk}[\alpha]^{\mathrm{U}} = \sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{U}} \geqslant 0}}^{n_O} \bar{W}_{kh}[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} + \sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{U}} < 0}}^{n_O} \bar{W}_{kh}[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}} + \bar{\Theta}_k[\alpha]^{\mathrm{U}}. \tag{17}$$

The objective is to minimize the cost function defined as

$$E_p = \sum_{\alpha} \sum_{k=1}^{n_O} \alpha(E^{\mathrm{L}}_{k(\alpha)} + E^{\mathrm{U}}_{k(\alpha)}) = \sum_{\alpha} E_{p(\alpha)}, \tag{18}$$

where

$$E_{p(\alpha)} = \sum_{k=1}^{n_0} \alpha(E^{\mathrm{L}}_{k(\alpha)} + E^{\mathrm{U}}_{k(\alpha)}), \tag{19}$$

$$E^{\mathrm{L}}_{k(\alpha)} = \tfrac{1}{2}(\bar{T}_{pk}[\alpha]^{\mathrm{L}} - \bar{O}_{pk}[\alpha]^{\mathrm{L}})^2, \tag{20}$$

$$E^{\mathrm{U}}_{k(\alpha)} = \tfrac{1}{2}(\bar{T}_{pk}[\alpha]^{\mathrm{U}} - \bar{O}_{pk}[\alpha]^{\mathrm{U}})^2, \tag{21}$$

where $E^{\mathrm{L}}_{k(\alpha)}$ and $E^{\mathrm{U}}_{k(\alpha)}$ can be viewed as the squared errors for the lower bound and the upper bound of the α-cut sets of a fuzzy weight. Other α-cut sets of a fuzzy weight are independently modified to reduce $E_{p(\alpha)}$. Otherwise, the fuzzy numbers after modifications are distorted. Therefore, each fuzzy weight is updated similar to but different from the approach of Ishibuchi et al. [10]. That is, in the proposed FNN, the membership functions are asymmetric Gaussian functions (i.e. a general shape) which is represented as

$$\bar{A}(x) = \begin{cases} \exp\left(-\dfrac{1}{2}\left(\dfrac{x-\mu}{\sigma^{\mathrm{L}}}\right)^2\right) & x < \mu, \\ 1, & x = \mu, \\ \exp\left(-\dfrac{1}{2}\left(\dfrac{x-\mu}{\sigma^{\mathrm{U}}}\right)^2\right) & \text{otherwise.} \end{cases} \tag{22}$$

Thus, the asymmetric Gaussian fuzzy weights are specified by their three parameters (i.e., center right width and left width). The gradient search method is derived for each parameter. It is the amount of adjustment for each parameter using the cost function $E_{p(\alpha)}$ as follows:

$$\Delta\mu_{kh}(t) = -\eta \frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}} + \beta\Delta\mu_{kh}(t-1), \tag{23}$$

$$\Delta\sigma^{\mathrm{R}}_{kh}(t) = -\eta \frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{R}}_{kh}} + \beta\Delta\sigma^{\mathrm{R}}_{kh}(t-1), \tag{24}$$

$$\Delta\sigma^{\mathrm{L}}_{kh}(t) = -\eta \frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{kh}} + \beta\Delta\sigma^{\mathrm{L}}_{kh}(t-1). \tag{25}$$

Appendix A provides a detailed derivation of Eqs. (23)–(25). Basically, the derivation is based on the chain rule as shown in the following:

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \mu_{kh}} + \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}}{\partial \mu_{kh}}, \tag{26}$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{R}}_{kh}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \sigma^{\mathrm{R}}_{kh}}, \tag{27}$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{kh}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}}{\partial \sigma^{\mathrm{L}}_{kh}}, \tag{28}$$

where $\bar{W}_{kh}$ is the asymmetric bell-shaped fuzzy number. Thus, $\bar{W}_{kh}[\alpha] = [\bar{W}_{kh}[\alpha]^{\mathrm{L}}, \bar{W}_{kh}[\alpha]^{\mathrm{U}}]$ and

$$\bar{W}_{kh}[\alpha]^{\mathrm{L}} = \mu_{kh} - \sigma^{\mathrm{L}}_{kh}(-2\ln\alpha)^{1/2} \quad \text{when } x \leqslant \mu,$$

$$\bar{W}_{kh}[\alpha]^{\mathrm{U}} = \mu_{kh} + \sigma^{\mathrm{R}}_{kh}(-2\ln\alpha)^{1/2} \quad \text{when } x > \mu.$$

Eqs. (26)–(28) can be written as

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}\, 1 + \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \cdot 1, \tag{29}$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{R}}_{kh}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} (-1)(-2\ln\alpha)^{1/2}, \tag{30}$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{kh}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} (-2\ln\alpha)^{1/2}. \tag{31}$$

The fuzzy weight $\bar{W}_{kh}$ is updated by the following rules:

$$\bar{W}_{kh}(t+1) \sim Gauss[c_{kh}(t+1),\ m_{kh}(t+1)], \tag{32}$$

where

$$c_{kh}(t+1) = c_{kh}(t) + \alpha \Delta c_{kh}(t), \tag{33}$$

$$m_{kh}(t+1) = m_{kh}(t) + \alpha \Delta m_{kh}(t). \tag{34}$$

The fuzzy weight $\bar{W}_{hi}$ and the fuzzy biases ($\bar{\Theta}_k$ and $\bar{\Theta}_h$) are updated in the same method as the fuzzy weight $\bar{W}_{kh}$. For the detailed derivation, refer to Kuo and Xue [22].

Lastly, assume that m patterns (i.e., $(\bar{X}_{pi}, \bar{T}_{pi})$, $i = 1,2,\ldots,n_I$, $p = 1,2,\ldots,m$) of fuzzy input vectors are given as training data, and also assume that S values of α-cut are used for the learning of the proposed FNN. In the case, the learning algorithm of the proposed FNN can be written as follows:

Learning algorithm:

Step 1: Initialize the fuzzy weights and the fuzzy biases through the GA.

Step 2: Repeat step 3 for $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ where α is the α-cut set.

Step 3: Repeat the following procedures for $p = 1,2,\ldots,m$ where p is the number of training samples.

(1) Forward calculation: Calculate the α-cut set of the fuzzy output vector $\bar{O}_p$ corresponding to the fuzzy input vector $\bar{X}_p$.

(2) Back-propagation: Adjust the fuzzy weights and the fuzzy biases using the cost function $E_{p(\alpha)}$.

Step 4: If the stop condition is not satisfied, go to step 2.

3.3. Decision integration

From the above two parts, quantitative and non-quantitative models, the general stock market tendency and special factor's effect are obtained. To obtain the final result, these two values are integrated with time effect through the other ANN. Similarly, the FNN is employed for the decision integration. Thus, the FNN has three input nodes which represent the time effect, stock market tendency without special events, and events effect, respectively. The input nodes are further connected with two hidden nodes which are connected with one output node, the stock market tendency. The investors can make a right decision based on the proposed system's support.

4. Model evaluation and discussion

The previous section has presented the proposed decision support system. This section employed the data collected from the Taiwan stock market in order to evaluate the proposed system. There are two reasons why to choose the Taiwan stock market. The first one is that it is an environment where the authors are familiar with and the second one is that the Taiwan stock market is a kind of "light-tray" stock market. If the proposed system can provide the acceptable forecast, it will also be appropriate for the other kinds of stock markets. Furthermore, the approaches with and without qualitative factors will be implemented and compared. The detailed discussion is as follows.

4.1. ANN without qualitative factors

As mentioned in Section 2, conventional research employed multiple regression models to discover the relationship between the factors and the stock index. However, owing to complexity of the model determination and linear constraint, they generally cannot provide rather acceptable findings and it is so even when the time-series analysis technique, ARMA, is applied. Thus, this study employs the ANN.

Based on the literature survey, 25 quantitative factors, or technical indexes (Appendix B) are applied to train the feedforward neural network with back propagation learning algorithm. Owing to dynamic consideration, the factors with "*" are those whose previous (one-day before) values are also included. Therefore, there are 42 input variables in total. For the output variable, two different outputs are tested. They are defined as:

$$O_1 = \frac{\mathrm{Max}(N_{t0} \sim N_{t3}) - N_{t0}}{\mathrm{Max}(N_{t0} \sim N_{t3}) - \mathrm{Min}(N_{t0} \sim N_{t3})} \tag{35}$$

Jang et al. [13] and

$$O_2 = \frac{\mathrm{Max}(N_{t0} \sim N_{t6}) - N_{t0}}{\mathrm{Max}(N_{t0} \sim N_{t6}) - \mathrm{Min}(N_{t0} \sim N_{t6})}, \tag{36}$$

where N_{t0}, N_{t3}, and N_{t6} are the indexes of current day, three days after, and six days after, respectively. $N_{t0} \sim N_{t6}$ means "from N_{t0} to N_{t6}". The training samples are from 1994 to 1995, while the testing samples are from January 1996 to April 1997. Since the network can have more than one hidden layer, all hidden layers are verified. Consequently, the number of possible testing

Table 1
Different testing network structures

Type	Input variables	Output variable	Number of hidden levels	Number of hidden nodes
1	Quantitative	O_1	1	20–110
2	Quantitative	O_1	2	20–110
3	Quantitative	O_2	1	20–110
4	Quantitative	O_2	2	20–110

Table 2
Results of ANN without qualitative effect

Type	Input variables	Output variable	Number of hidden levels	Number of hidden nodes	Forecasting MSE (training rate, momentum)
1	Quantitative	O_1	1	60	*0.09501(0.2, 0.8)
2	Quantitative	O_1	2	65	0.09512(0.5, 0.8)
3	Quantitative	O_2	1	30	0.08283(0.5, 0.8)
4	Quantitative	O_2	2	45	*0.07720(0.5, 0.8)

*The best network.

networks is four, say types 1–4, as shown in Table 1. Different combinations of training rate and momentum are also tested. The network will not stop training until 50 000 epochs. According to the computational results (Table 2), type 4 has the smallest MSE value, 0.0772, as the training rate and momentum are 0.5 and 0.8, respectively. Meanwhile, type 1 has second smallest MSE value, 0.0942, as the training rate and momentum are 0.2 and 0.8, respectively. Results of these two networks are compared with the networks while considering the qualitative factors in the next two sections.

4.2. *ANN with both quantitative and qualitative factors*

4.2.1. *Qualitative factors identification*

In the above section, only quantitative, or technical indexes, are applied to train the ANN, while both the quantitative and qualitative factors are employed in this section. Initially, the factors are divided into two categories, quantitative and qualitative, as mentioned above. All the possible qualitative events (factors), which may influence the stock market, were identified from the related economics journals, government technical reports, and newspapers from 1991 to 1997. The experienced experts, or stock brokers, eliminated the unnecessary events and then divided the other events into six dimensions (political, financial, economic, message, technical, and international) based on their characteristics to formulate the first questionnaire. The questionnaire for each event has the following format:

IF event A occurs, *THEN its effect on the stock market is from* __ *to* __.

This kind of method is named fuzzy Delphi method. For detailed procedures and discussion, refer to Ishikawa et al. [12] and Kuo et al. [21]. The major idea of fuzzy Delphi method is to capture expert's knowledge. After two runs of survey, the membership functions of all the linguistic terms have converged and are shown in Tables 3–8. These membership functions are the fuzzy inputs of the GFNN.

4.2.2. *Quantitative factors determination*

Similar to the network without considering the qualitative factors, 42 technical indexes are selected.

4.2.3. *Qualitative model (GFNN)*

Each dimension owns its linguistic terms, or fuzzy numbers. In fact, more than one event may occur in one day. Using a single GFNN, one cannot handle this condition. Therefore, a hierarchical structure is proposed. For instance, three events, say *A*, *B* and *C*, of political dimension and two events, *D* and *E*, of financial dimension occur in one day. The political

Table 3
The fuzzy numbers for political dimension

Linguistic term	Effect on stock market	Left standard deviation σ^L	μ value when membership value is 1	Right stand deviation σ^R
Possible		1.0580	−0.0330	1.0950
Low	Positive	0.4716	2.1056	0.5726
	Negative	0.5327	−2.5751	0.7520
Medium	Positive	1.1108	5.0567	0.6747
	Negative	1.3392	−5.1295	0.8810
High	Positive	1.6310	9.6950	0.1017
	Negative	0.1755	−9.4735	0.9627

Table 4
The fuzzy numbers for financial dimension

Linguistic term	Effect on stock market	Left standard deviation σ^L	μ value when membership value is 1	Right standard deviation σ^R
Possible		1.1546	0.1593	0.8685
Low	Positive	0.6092	2.5771	0.6242
	Negative	0.7563	−2.0585	0.6468
Medium	Positive	0.9167	5.2723	0.7962
	Negative	1.0173	−4.9064	0.9093
High	Positive	0.7416	9.1454	0.2849
	Negative	0.2242	−9.3273	1.5595

Table 5
The fuzzy numbers for economic dimension

Linguistic term	Effect on stock market	Left standard deviation σ^L	μ value when membership value is 1	Right standard deviation σ^R
Possible		0.7892	0.0242	0.5056
Low	Positive	0.1390	1.7692	0.1417
	Negative	0.4254	−2.0396	0.6469
Medium	Positive	0.5591	4.6465	0.5185
	Negative	1.1497	−4.8680	0.4558
High	Positive	*	*	*
	Negative	0.6567	−7.2352	0.2965

GFNN combines events A and B first and then the result is integrated with event C. It is also similar for the financial GFNN. Finally, the fuzzy effects from six-“dimensional” GFNN are integrated by using an “integration” GFNN. Intuitively, each dimensional GFNN has two input nodes and one output node. The integration GFNN possesses six input nodes (from six dimension) and one output node. The fuzzy numbers of output node is quite difficult to obtain. For the current application, the questionnaire may have more than forty thousand rules. It is not practical. Thus, the experts give each dimension a weight. The output is from the multiplication of all the dimensional effects and corresponding weights. The evaluation results for dimensional GFNNs and integration GFNN are as follows.

4.2.3.1. Dimensional GFNN. Since the training procedures for each dimensional GFNN are similar, the political dimension is used to account for explaining the whole procedures. The asymmetric Gaussian fuzzy numbers as shown in Tables 3–8 are employed to train the GFNN. The political dimension has 64

Table 6
The fuzzy numbers for international dimension

Linguistic term	Effect on stock market	Right standard deviation σ^L	μ value when membership value is 1	Right standard deviation σ^R
Possible		0.5618	-0.0841	0.7063
Low	Positive	0.5239	2.7245	0.5535
	Negative	0.4097	-1.4606	0.4869
Medium	Positive	0.4066	4.3479	0.2757
	Negative	0.7878	-4.6398	0.7100
High	Positive	0.3094	6.8896	0.2558
	Negative	0.1404	-9.5789	0.9685

Table 7
The fuzzy numbers for message dimension

Linguistic term	Effect on stock market	Left standard deviation σ^L	μ value when membership value is 1	Right standard deviation σ^R
Possible		0.1875	-0.1962	0.2651
Low	Positive	0.5032	2.8483	0.2545
	Negative	0.6192	-0.9943	0.3314
Medium	Positive	0.3906	4.5003	0.6594
	Negative	0.8842	-5.4101	0.7437
High	Positive	0.0968	8.8297	0.2796
	Negative	0.1558	-9.5327	1.3417

Table 8
The fuzzy numbers for technical dimension

Linguistic term	Effect on stock market	Left standard deviation σ^L	μ value when membership value is 1	Right standard deviation σ^R
Possible		0.5594	-0.4307	0.7689
Low	Positive	*	*	*
	Negative	0.5285	-1.2310	0.1025
Medium	Positive	0.5785	5.2712	0.3421
	Negative	1.0490	-4.3518	0.4981
High	Positive	*	*	*
	Negative	*	*	*

Table 9
The fuzzy IF–THEN rule tables for political dimension

MX	NZ	MS	MM	MM	ML	ML	ML	ML
ML	PS	NZ	MS	MS	MM	MM	ML	ML
MM	PM	PM	PS	NZ	MS	MS	MM	ML
MS	PM	PM	PS	NZ	MS	MS	MM	ML
PS	PL	PL	PM	PS	NZ	NZ	MM	MM
PM	PL	PL	PM	PM	PS	PS	MS	MM
PL	PL	PL	PL	PL	PM	PM	NZ	MS
PX	PL	PL	PL	PL	PM	PM	PS	NZ
	PX	PL	PM	PS	MS	MM	ML	MX

Table 10
The fuzzy IF–THEN rule tables for financial dimension

MX	NZ	MM	MM	MM	ML	ML	ML	ML
ML	PS	NZ	MS	MS	MM	ML	ML	ML
MM	PS	NZ	NZ	MS	MM	ML	ML	ML
MS	PM	PS	NZ	NZ	MM	MM	MM	ML
PS	PL	PM	PM	PS	NZ	MS	MS	MM
PM	PL	PL	PM	PM	NZ	NZ	MS	MM
PL	PL	PL	PL	PM	PS	NZ	NZ	MM
PX	PL	PL	PL	PL	PM	PS	PS	NZ
	PX	PL	PM	PS	MS	MM	ML	MX

Table 11
The fuzzy IF–THEN rule tables for economic dimension

ML	NZ	MS	MS	ML	ML	ML
MM	NZ	NZ	MS	MM	MM	ML
MS	PS	NZ	NZ	MM	MM	ML
PS	PM	PM	PM	NZ	MS	MS
PM	PM	PM	PM	NZ	NZ	MS
PL	PL	PM	PM	PS	NZ	NZ
	PL	PM	PS	MS	MM	ML

Table 12
The fuzzy IF–THEN rule tables for international dimension

ML	MS	MM	MM	ML	ML	ML
MM	PS	NZ	MS	MM	MM	ML
MS	PS	NZ	NZ	MM	MM	ML
PS	PL	PM	PS	NZ	MS	MM
PM	PL	PM	PM	NZ	NZ	MM
PL	PL	PL	PL	PS	PS	MS
	PL	PM	PS	MS	MM	ML

Table 13
The fuzzy IF–THEN rule tables for message dimension

MX	MS	MM	MM	MM	ML	ML	ML	ML
ML	MS	MS	MM	MM	ML	ML	ML	ML
MM	PS	NZ	MS	MS	MM	ML	ML	ML
MS	PM	PS	NZ	MS	MM	MM	ML	ML
PS	PL	PM	PM	PM	MS	MS	MM	MM
PM	PL	PL	PM	PM	NZ	MS	MM	MM
PL	PL	PL	PL	PM	PS	NZ	MS	MM
PX	PL	PL	PL	PL	PM	PS	MS	MS
	PX	PL	PM	PS	MS	MM	ML	MX

Table 14
The fuzzy IF–THEN rule tables for technical dimension

ML	MS	MS	MM	ML
MM	NZ	NZ	MM	MM
PM	PM	PM	NZ	MS
PL	PL	PM	NZ	MS
	PL	PM	MM	ML

(8×8) fuzzy rules. The fuzzy rules of each dimension are shown in Tables 9–14. Since, so far, there is no rule which can determine the hidden nodes number, different network structures are tested in order to find out the better network. In addition, some parameters must be set up for the GA. The procedures are as follows:

1. Network structures: 2-4-1 (input-hidden-output), 2-5-1, and 2-6-1.
2. GA parameters setup:
(2.1.) Generations: 1000.
(2.2.) Crossover rate: 0.2.
(2.3.) Crossover type: two-point crossover.
(2.4.) Mutation rate: 0.8
(2.5.) After 1000 generations, five groups of weights which are the five largest fitness function values (Tables 15–20) are selected.
3. Apply the five groups of weights obtained from the GA as the initial weights to train the FNN (Tables 15–20).
4. Select the best network based on the training and testing MSE (mean square error) values. (Tables 15–20).

Tables 15–20 indicate that the network using the initial weights with larger fitness function cannot guarantee the smaller MSE value. Just like political dimension, the network using the weights with the fourth largest fitness function value has the smallest MSE values both for training and testing. However, most of the network using the initial weights with larger fitness function value can yield a better estimation, e.g. financial, international, and message dimensions as summarized in Table 21.

4.2.3.2. Integration GFNN. Based on six-dimensional GFNNs defined, six fuzzy numbers are obtained for each day. These six fuzzy numbers are further input to the integration GFNN to obtain the fuzzy qualitative effect on the stock market. Thus, the integration GFNN is trying to format the fuzzy relationship among six dimensions. The fuzzy rule format can be expressed as:

IF political dimension is "positive high" AND
financial dimension is "positive medium" AND
economic dimension is "positive medium" AND
international dimension is "positive high" AND
message dimension is "possible" AND
technical is "possible"
THEN integration dimension is "positive high".

Each rule is a training sample to integrate GFNN. Thus, the network structure has six input nodes while the number of output nodes is still one. The training fuzzy rules are presented in Table 22. Similarly, the following procedures are verified to discover the optimum forecasting result:

1. Network structures: 6-6-1 (input-hidden-output), 6-9-1, and 6-12-1.
2. GA parameters setup:
2.1. Generations: 1000.
2.2. Crossover rate: 0.2.
2.3. Crossover type: two-point crossover.
2.4. Mutation rate: 0.8
2.5. After 1000 generations, four groups of weights which have the largest, second largest, middle, and smallest fitness function value (Table 23) are selected.
3. Apply the above four groups of weights obtained from the GA to train the FNN (Table 24).
4. Select the optima; network based on the training and testing MSE (mean square error) values (Table 24).

Table 24 indicates that network 6-12-1 with initial weights from largest fitness function value has the smallest MSE value, 0.006665, as both the training rate and momentum are 0.5. This network only requires 8500 epochs of training to reach the minimum. Basically, we can treat this integration GFNN as a fuzzy knowledge base for the qualitative effect

Table 15
Training and testing results for political dimension

Network type	Source of weights	Training MSE	Testing MSE	Training epochs
2-4-1	Largest fitness function value (FFV)	0.017392	0.011763	150
$\eta = 0.2$	Second FFV	0.012064	0.011537	200
$\beta = 0.8$	Third FFV	0.012040	0.011543	250
	Fourth FFV	0.011910	*0.011334	300
	Fifth FFV	0.012044	0.011543	260
2-5-1	Largest fitness function value (FFV)	0.035764	0.030959	1200
$\eta = 0.5$	Second FFV	0.028063	0.021201	460
$\beta = 0.5$	Third FFV	0.024256	0.017878	370
	Fourth FFV	0.025907	0.019253	450
	Fifth FFV	0.030459	0.023399	525
2-6-1	Largest fitness function value (FFV)	0.012881	0.022754	300
$\eta = 0.8$	Second FFV	0.018827	0.012512	50
$\beta = 0.5$	Third FFV	0.018874	0.012373	60
	Fourth FFV	0.018918	0.012459	60
	Fifth FFV	0.020782	0.013875	60

*The best network.

Table 16
Training and testing results for financial dimension

Network type	Source of weights	Training MSE	Testing MSE	Training epochs
2-4-1	First fitness function value (FFV)	0.018071	0.012691	80
$\eta = 0.2$	Second FFV	0.018075	0.012697	80
$\beta = 0.8$	Third FFV	0.022059	0.015553	80
	Fourth FFV	0.022066	0.015784	85
	Fifth FFV	0.021996	0.015694	85
2-5-1	Largest fitness function value (FFV)	0.009878	0.007245	35
$\eta = 0.5$	Second FFV	0.010014	0.009583	35
$\beta = 0.5$	Third FFV	0.010094	0.010232	35
	Fourth FFV	0.009967	0.009266	37
	Fifth FFV	0.009984	0.008675	36
2-6-1	Largest fitness function value (FFV)	0.008567	*0.006567	800
$\eta = 0.8$	Second FFV	0.009027	0.006819	1000
$\beta = 0.5$	Third FFV	0.009039	0.006823	1000
	Fourth FFV	0.008923	0.006889	1200
	Fifth FFV	0.009027	0.006883	1100

*The best network.

Table 17
Training and testing results for economic dimension

Network type	Source of weights	Training MSE	Testing MSE	Training epochs
2-4-1	Largest fitness function value (FFV)	0.018154	0.017886	830
$\eta = 0.2$	Second FFV	0.021052	0.020710	650
$\beta = 0.8$	Third FFV	0.017922	*0.017665	860
	Fourth FFV	0.018514	0.018226	880
	Fifth FFV	0.019152	0.018850	900
2-5-1	Largest fitness function value (FFV)	0.027224	0.028871	200
$\eta = 0.5$	Second FFV	0.031807	0.030226	90
$\beta = 0.5$	Third FFV	0.047687	0.056506	140
	Fourth FFV	0.032048	0.034069	90
	Fifth FFV	0.078617	0.083166	170
2-6-1	Largest fitness function value (FFV)	0.025137	0.025014	70
$\eta = 0.8$	Second FFV	0.025101	0.025038	70
$\beta = 0.5$	Third FFV	0.024985	0.024850	70
	Fourth FFV	0.025440	0.025319	70
	Fifth FFV	0.025166	0.025035	70

*The best network.

Table 18
Training and testing results for international dimension

Network type	Source of weights	Training MSE	Testing MSE	Training epochs
2-4-1	Largest fitness function value (FFV)	0.006486	0.006461	3600
$\eta = 0.2$	Second FFV	0.007714	0.007768	1700
$\beta = 0.8$	Third FFV	0.007497	0.007429	2500
	Fourth FFV	0.007370	0.007312	2750
	Fifth FFV	0.006836	0.006898	3800
2-5-1	Largest fitness function value (FFV)	0.033081	0.032987	54
$\eta = 0.5$	Second FFV	0.033560	0.033529	54
$\beta = 0.5$	Third FFV	0.033687	0.033524	55
	Fourth FFV	0.033409	0.033317	55
	Fifth FFV	0.033630	0.033580	55
2-6-1	Largest fitness function value (FFV)	0.007931	0.007612	2000
$\eta = 0.8$	Second FFV	0.008392	0.007910	3000
$\beta = 0.5$	Third FFV	0.008430	0.007943	3000
	Fourth FFV	0.008363	0.007879	3250
	Fifth FFV	0.008871	0.008639	2000

Table 19
Training and testing results for message dimension

Network type	Source of weights	Training MSE	Testing SE	Training epochs
2-4-1	Largest fitness function value (FFV)	0.008835	*0.006999	5000
$\eta = 0.2$	Second FFV	0.008837	0.007002	5000
$\beta = 0.8$	Third FFV	0.008864	0.007036	5000
	Fourth FFV	0.008849	0.007009	5000
	Fifth FFV	0.008843	0.007005	5500
2-5-1	Largest fitness function value (FFV)	0.011558	0.009160	700
$\eta = 0.5$	Second FFV	0.011360	0.009021	750
$\beta = 0.5$	Third FFV	0.011008	0.008753	700
	Fourth FFV	0.011409	0.009072	750
	Fifth FFV	0.011412	0.009073	750
2-6-1	Largest fitness function value (FFV)	0.022099	0.019773	100
$\eta = 0.8$	Second FFV	0.022105	0.020191	100
$\beta = 0.5$	Third FFV	0.022233	0.020018	100
	Fourth FFV	0.021793	0.020037	100
	Fifth FFV	0.021788	0.019799	100

*The best network.

Table 20
Training and testing results for technical dimension

Network type	Source of weights	Training MSE	Testing MSE	Training epochs
2-4-1	Largest fitness function value (FFV)	0.030062	0.029821	450
$\eta = 0.2$	Second FFV	0.029552	0.029316	500
$\beta = 0.8$	Third FFV	0.030093	0.029860	500
	Fourth FFV	0.029599	0.029941	460
	Fifth FFV	0.030020	0.029781	450
2-5-1	Largest fitness function value (FFV)	0.007725	0.007632	6300
$\eta = 0.5$	Second FFV	0.007732	0.007639	6300
$\beta = 0.5$	Third FFV	0.004760	0.004745	6000
	Fourth FFV	0.005636	0.005607	5700
	Fifth FFV	0.005027	0.004999	6000
2-6-1	Largest fitness function value (FFV)	0.003872	0.003848	3500
$\eta = 0.8$	Second FFV	0.003692	*0.003671	3250
$\beta = 0.5$	Third FFV	0.003734	0.003711	3500
	Fourth FFV	0.003726	0.003698	3500
	Fifth FFV	0.003747	0.003723	3500

*The best network.

on the stock market. It can be integrated with the quantitative factors further to determine the stock market tendency through any kinds of approaches, e.g. regression method or ANN.

Tables 15–20 and 24 show that the numbers of training epochs are very small. This indicates that using the GA result as the initial weights for GFNN is promising. For example, the network only needs 150 training epochs if the GFNN is applied, while more than 50 000 training epochs are needed if only using FNN.

Over here, one question may arise. Can final weights be determined only using GA. Therefore, the test is also conducted. The result showed that GA can find a

Table 21
The results of each dimension

Dimension	Network type	Source of weights	Training MSE	Testing MSE
Political	2-4-1	Fourth	0.011910	0.011334
Financial	2-6-1	Largest	0.009984	0.006567
Economic	2-4-1	Third	0.017922	0.017665
International	2-4-1	Largest	0.006486	0.006461
Message	2-4-1	Largest	0.008835	0.006999
Technical	2-6-1	Second	0.003692	0.003671

Table 22
Asymmetric Gaussian fuzzy numbers of integration dimension

Linguistic term	Effect on stock market	Right standard deviation σ^L	μ value when membership value is 1	Right standard deviation σ^R
Possible	Positive	0.9719	0.9175	0.6871
	Negative	0.5917	−0.1860	0.7089
Low	Positive	0.4494	2.4049	0.4293
	Negative	0.5453	−1.7265	0.4944
Medium	Positive	0.6604	4.8492	0.5445
	Negative	1.0379	−4.8843	0.6997
High	Positive	0.6947	8.6399	0.4534
	Negative	0.2643	−9.2072	1.0569

Table 23
The GA results of integration dimension

Network type	Largest fitness function value	Second fitness function value	Middle fitness function value	Smallest fitness function value
6-6-1	0.473510	0.475133	0.624284	0.903342
6-9-1	0.611257	0.618405	0.668547	0.751665
6-12-1	0.502640	0.503607	0.577028	0.951646

feasible solution after 100 000 generations. However, it is still worse than the result of integration of ANN and GA.

4.2.4. Decision integration

The integration ANN is the feedforward ANN with EBP learning algorithm. Both the quantitative and qualitative factors are inputs of the ANN, and should be normalized in [0, 1]. Forty two factors exit for the quantitative factors. Meanwhile, the number of qualitative factors is 11. The reason that the number of qualitative factors is eleven is that the fuzzy number obtained from integration GFNN is α-cut by 0.1, 0.3, 0.5, 0.7, 0.9, 1.0, respectively. Cumulatively, there are 54 input nodes for the network after including the time effect. The number of output nodes is still one. However, similarly, two different outputs, O_1 and O_2, are verified. Since the network can have more than one hidden layer, both one and two hidden layers are tested. Consequently, the number of possible testing network is four (types 5–8) as shown in Table 25. The computational results (Table 26) indicate that type 6 and type 8 are two optimum networks due to smaller MSE values. The MSE values of these two networks are 0.0386 and 0.0297, respectively.

4.3. Discussion

According to Section 4.1, two networks (types 1 and 4) have been found to be the best without considering

Table 24
Training and testing results for integration GFNN

Network type	Source of weights	Training MSE	Testing MSE	Training epochs
6-6-1	Largest fitness function value (FFV)	0.023676	0.024537	7000
$\eta = 0.5$	Second FFV	0.024707	0.024809	7500
$\beta = 0.5$	Middle FFV	0.012486	0.012612	9000
	Smallest FFV	0.036133	0.036494	8500
6-9-1	Largest fitness function value (FFV)	0.009169	0.009444	8000
$\eta = 0.5$	Second FFV	0.011750	0.012220	8000
$\beta = 0.5$	Middle FFV	0.012435	0.013057	7500
	Smallest FFV	0.014282	0.014710	8500
6-12-1	Largest fitness function value (FFV)	0.006534	*0.006665	8500
$\eta = 0.5$	Second FFV	0.007050	0.007192	8000
$\beta = 0.5$	Middle FFV	0.007501	0.075763	8500
	Smallest FFV	0.007137	0.007494	9000

*The best network.

Table 25
Testing network types

Type	Input variables	Output variables	Number of hidden levels	Number of hidden nodes
5	Composite	O_1	1	$20 \sim 110$
6	Composite	O_1	2	$20 \sim 110$
7	Composite	O_2	1	$20 \sim 110$
8	Composite	O_2	2	$20 \sim 110$

Table 26
Results of different network structures

Type	Input variables	Output variable	Number of hidden levels	Number of hidden nodes	Forecasting MSE (training rate, momentum)
5	Composite	O_1	1	65	0.04313(0.5, 0.8)
6	Composite	O_1	2	50	*0.0386(0.8, 0.5)
7	Composite	O_2	1	75	0.03114(0.2, 0.8)
8	Composite	O_2	2	65	*0.0297(0.5, 0.8)

*The best network.

the qualitative effect, while types 6 and 8 considering the qualitative effect can provide the best forecasting results in Section 4.2.4. These four networks should be further compared on the basis of the different performance measures as illustrated next.

(1) *Based on the learning efficiency*: Table 27 lists the computational results. The results indicate that two networks with composite factors always outperform the other two network with only quantitative factors. In addition, the networks with O_2 output are better than the networks with O_1 output. However, simply comparing the MSE values is insufficient. Checking the clarity of the buy–sell points and the buy–sell performance are also necessary.

(2) *Based on the clarity of the buy–sell points*: The purchasing or selling activity is determined by the buy–sell signal obtained from the proposed system. If the signal is over the upper bound, then go ahead to

Table 27
Computation results of different alternatives

	Output			
	O_1		O_2	
Network type	Type 1	Type 6	Type 4	Type 8
Network number	No. 1	No. 2	No. 3	No. 4
Input	Quantitative	Composite	Quantitative	Composite
Number of hidden layer	1	2	2	2
Number of hidden layer nodes	60	50	45	65
Training rate	0.2	0.8	0.5	0.5
Momentum	0.8	0.5	0.8	0.8
MSE	0.09424	0.03869	0.07721	0.02966

buy the stock. If the signal is below the lower bound, then sell the stock. Otherwise, keep the stock. Basically, six different conditions are as follows.

(a) When the investor already has owned the stock:

1. if the signal is below the lower bound, then SELL the stock.

2. if the signal is over the lower bound, then KEEP the stock.

(b) When the investor has not owned the stock:

3. if the signal is below the upper bound, then keep WAITING.

4. if the signal is over the upper bound, then BUY the stock.

Figs. 3 and 4 illustrate the stock index curves and the buy–sell curves for the networks No. 2. If we choose interval $[0.3, 0.7]$ as the "KEEP" instance, network No. 2 has the optimal best buy–sell clarity as presented in Table 28. Moreover, the interval $[0.2, 0.8]$ results in the largest buy–sell clarity rate, 0.5066, as shown in Table 29. This implies that the networks with qualitative factors can have the better buy–sell clarity rate. Restated, it turns out to be more sensitive. However, the higher clarity rate the higher transaction cost. Thus, further discussion is still needed. Therefore, the next comparison is based on the buy–sell performance.

(3) *Based on the buy–sell performance*: Based on the six conditions, or transaction rules, mentioned above, the buy–sell performance can be determined. Basically, the generation of the performance is as follows:

$$P = I_b - I_s, \tag{37}$$

where P denotes the performance, I_b represents the index value when the stock was bought, and I_s is the index value when the stock was sold. Table 30 lists the evaluation results. Network No. 2 has 27 transactions and its performance is 8035.6 points, while the network No. 4 has 17 times of transaction and its total performance is 6974.8 points. Clearly, the network No. 2 outperforms network No. 4 in the total buy–sell performance. In addition, it is also better than the total stock tendency, 3759.9 points. Thus, it is without doubt that network No. 2 is the best selection. Table 31 summaries those results.

5. Conclusions

The model evaluation results have revealed that the proposed intelligent decision support system can outperform the ANN only considering the quantitative factors in the learning accuracy, buy–sell clarity and the buy–sell performance. The proposed GFNN can cover all the events that have occurred from 1991 to 1997 after checking with the historical events. These events, no matter belonging to which dimension, influence the stock market more or less both in the long or short terms. Thus, the stock market is, authentically, a multiple dimension integrator.

Whenever the newly occurred event is not on the database list, the user can evaluate himself/herself. Once the fuzzy value has been well set up, the database can calculate the effect value automatically. In addition, each dimension's weight can be adjusted dynamically if needed.

Upon the determination of layer number and node number, so far, it is also an unsolved question. The optimum and the worst method is the trial and error.

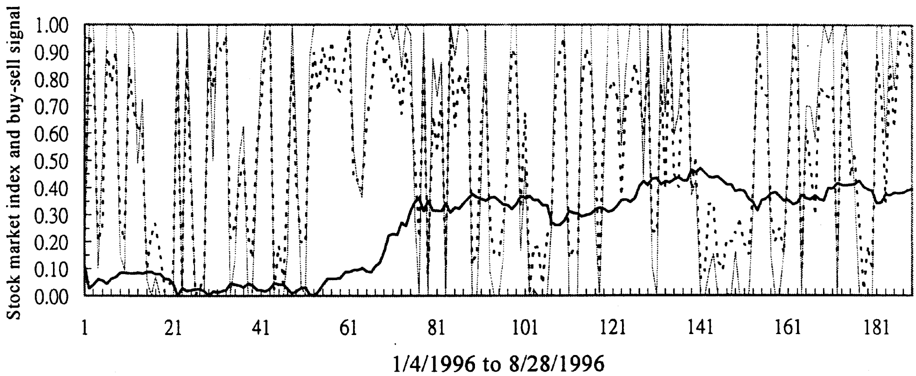

Fig. 3. The forecasting results of network No. 2 where dark line is the index curve, thin line is the 6-day actual buy–sell signal curve and dash line is the forecasting buy–sell signal curve.

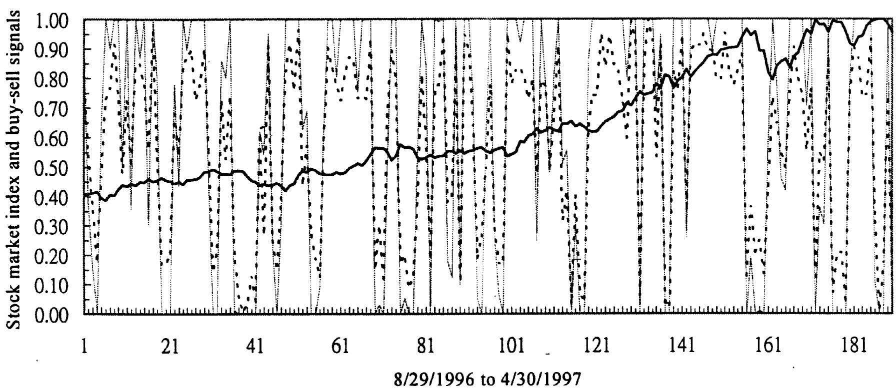

Fig. 4. The forecasting results of network No. 2 where dark line is the index curve, thin line is the 6-day actual buy–sell signal curve and dash line is the forecasting buy–sell signal curve.

Table 28
Comparison of clarity for boundary [0.3, 0.7]

	Network number			
	No. 1	No. 2	No. 3	No. 4
Number of buying signals	126	178	127	169
Number of selling signals	91	116	71	88
Total number of days	379	379	379	379
The clarity rate of buying points	0.3325	0.4696	0.3351	0.4459
The clarity rate of selling points	0.2401	0.3061	0.1873	0.2322
The clarity rate of buying–selling points	0.5725	0.7757	0.5224	0.6781

Table 29
Comparison of clarity for boundary [0.2, 0.8]

	Network number			
	No. 1	No. 2	No. 3	No. 4
Number of buying signals	72	110	80	132
Number of selling signals	62	80	37	60
Total number of days	379	379	379	379
The clarity rate of buying points	0.1890	0.2902	0.2111	0.3483
The clarity rate of selling points	0.1636	0.2111	0.0976	0.1583
The clarity rate of buying–selling points	0.3526	0.5013	0.3087	0.5066

Table 30
The results of buying–selling performance

	Network number			
	No. 1	No. 2	No. 3	No. 4
The number of transaction	20	27	15	17
The number of effective transaction (profit)	20	26	15	16
The number of ineffective transaction (lost)	0	1	0	1
The buying–selling performance	+6699.7 points	+8035.6 points	+6784.8 points	+6974.8 points
The average buying–selling performance	+335.0 points	+297.6 points	+ 452.3 points	+410.3 points
Total stock tendency	+3759.9 points			

Table 31
The summarized results of four different networks

	Network number			
	No. 1	No. 2	No. 3	No. 4
Forecasting year	Jan. 1996–Apr. 1997			
MSE	0.09424	0.03869	0.07721	0.02966
Dominant rate of buying signal	33.25%	46.96%	33.51%	44.59%
Dominant rate of selling signal	24.01%	30.61%	18.73%	23.22%
Dominant rate	57.25%	77.57%	52.24%	67.81%
Transaction times	20	27	15	17
Effective transaction times	20	26	15	16
Ineffective transaction times	0	1	0	1
Rate of effective transaction times	100%	96.30%	100%	94.12%
Performance of buying and selling	6699.7	8035.6	6784.8	6974.8
Average return of per transaction	335.0	297.6	452.3	410.3
Transaction cost	1000	1350	750	850
Net performance of buying and selling	5699.7	6685.6	6034.4	6124.8
The trade of stock index	3759.9			

In addition, the setups of training rate and momentum also waste plenty of time. The fuzzy models proposed in [18] can be applied in the future.

Though it looks that the results of GFNN is pretty good, the performance can be further improved. For instance, this study used the binary coding approach. In the future, the real-number coding approach can be applied. In addition, different ANNs can replace the feedforward neural network with EBP learning algorithm.

Acknowledgements

The authors would like to thank the National Science Council, Taiwan, ROC for partially supporting this study under Contract NSC 85-2214-213-001. The authors also thank the anonymous reviewers for their helpful comments.

Appendix A. Calculation of derivatives in the fuzzy weights determination

From input layer to hidden layer, the derivatives of $\partial E_{p(\alpha)}/\partial \mu_{kh}$, $\partial E_{p(\alpha)}/\partial \sigma^{\mathrm{L}}_{kh}$, and $\partial E_{p(\alpha)}/\partial \sigma^{\mathrm{R}}_{kh}$ for the coefficients $\mu_{kh}, \sigma^{\mathrm{L}}_{kh}$, and σ^{R}_{kh} are as follows:

(1) If $0 \leqslant \bar{W}_{kh}[\alpha]^{\mathrm{L}} \leqslant \bar{W}_{kh}[\alpha]^{\mathrm{U}}$, then

a. *for* μ_{kh}:

$$\begin{aligned}\frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}} &= \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \mu_{kh}} + \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}}{\partial \mu_{kh}} \\ &= \frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}} \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \mu_{kh}} + \frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}} \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}}{\partial \mu_{kh}}\end{aligned}$$

where $\bar{W}_{kh}[\alpha]^{\mathrm{L}}$ and $\bar{W}_{kh}[\alpha]^{\mathrm{U}}$ are

1. when $x \leqslant \mu$,

$$\alpha = \exp\left(-\frac{1}{2}\left(\frac{\bar{W}_{kh}[\alpha]^{\mathrm{L}} - \mu_{kh}}{\sigma^{\mathrm{L}}_{kh}}\right)^2\right)$$

$$\Rightarrow \ln\alpha = -\frac{1}{2}\left(\frac{\bar{W}_{kh}[\alpha]^{\mathrm{L}} - \mu_{kh}}{\sigma^{\mathrm{L}}_{kh}}\right)^2$$

So

$$\bar{W}_{kh}[\alpha]^{\mathrm{L}} = \mu_{kh} - \sigma^{\mathrm{L}}_{kh}(-2\ln\alpha)^{1/2}.$$

2. When $x > \mu$,

$$\bar{W}_{kh}[\alpha]^{\mathrm{U}} = \mu_{kh} + \sigma^{\mathrm{R}}_{kh}(-2\ln\alpha)^{1/2}.$$

Therefore,

$$\frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \mu_{kh}} = 1.$$

In addition, let

$$\begin{aligned}\delta_k[\alpha]^{\mathrm{L}} &= -\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}} \\ &= \alpha(\bar{T}_{pk}[\alpha]^{\mathrm{L}} - \bar{O}_{pk}[\alpha]^{\mathrm{L}}) \cdot \bar{O}_{pk}[\alpha]^{\mathrm{L}} \cdot (1 - \bar{O}_{pk}[\alpha]^{\mathrm{L}}),\end{aligned}$$

$$\begin{aligned}\delta_k[\alpha]^{\mathrm{U}} &= -\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}} \\ &= \alpha(\bar{T}_{pk}[\alpha]^{\mathrm{U}} - \bar{O}_{pk}[\alpha]^{\mathrm{U}}) \cdot \bar{O}_{pk}[\alpha]^{\mathrm{U}} \cdot (1 - \bar{O}_{pk}[\alpha]^{\mathrm{U}}).\end{aligned}$$

Therefore,

$$\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}} = -\delta_k[\alpha]^{\mathrm{U}}$$

and

$$\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}} = -\delta_k[\alpha]^{\mathrm{L}}.$$

So

$$\begin{aligned}\frac{\partial E_{ps}}{\partial \mu_{kh}} &= -\delta_k[\alpha]^{\mathrm{L}}\,\bar{O}_{ph}[\alpha]^{\mathrm{L}} \cdot [-(-2\ln\alpha)^{1/2}] + (-\delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}}(-2\ln\alpha)^{1/2}) \\ &= \delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}}(-2\ln\alpha)^{1/2} - \delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}}(-2\ln\alpha)^{1/2}.\end{aligned}$$

b. *For* σ^{L}_{kh}:

$$\begin{aligned}\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{kh}} &= \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \sigma^{\mathrm{L}}_{kh}} \\ &= \frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}} \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \sigma^{\mathrm{L}}_{kh}} \\ &= -\delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}}[-(-2\ln\alpha)^{1/2}] \\ &= \delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}}(-2\ln\alpha)^{1/2},\end{aligned}$$

where

$$\frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \sigma^{\mathrm{L}}_{kh}} = -(-2\ln\alpha)^{1/2}.$$

c. *For* σ_{kh}^{R}:

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{kh}^{\mathrm{R}}} = \frac{\partial E_{p(\alpha)}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}}{\partial \sigma_{kh}^{\mathrm{R}}} = \frac{\partial E_{p(\alpha)}}{\overline{Net}_{pk}[\alpha]^{\mathrm{U}}} \frac{\overline{Net}_{pk}[\alpha]^{\mathrm{U}}}{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{U}}}{\partial \sigma_{kh}^{\mathrm{R}}} = -\,\delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} (-2\ln\alpha)^{1/2},$$

where

$$\frac{\partial \bar{W}_{kh}[\alpha]^{\mathrm{L}}}{\partial \sigma_{kh}^{\mathrm{R}}} = (-2\ln\alpha)^{1/2}.$$

(2) If $\bar{W}_{kh}[\alpha]^{\mathrm{L}} \leqslant \bar{W}_{kh}[\alpha]^{U} < 0$, then similarly

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}} = -\,\delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} - \delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{kh}^{\mathrm{L}}} = \delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} (-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}^{\mathrm{R}}} = -\delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{L}} (-2\ln\alpha)^{1/2}.$$

(3) If $\bar{W}_{kh}[\alpha]^{\mathrm{L}} < 0 \leqslant \bar{W}_{kh}[\alpha]^{\mathrm{U}}$, then

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{kh}} = -\,\delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} - \delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{kh}^{\mathrm{L}}} = \delta_k[\alpha]^{\mathrm{L}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} (-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{kh}^{\mathrm{R}}} = -\delta_k[\alpha]^{\mathrm{U}} \cdot \bar{O}_{ph}[\alpha]^{\mathrm{U}} (-2\ln\alpha)^{1/2}.$$

Since the derivatives of $\partial E_{p(\alpha)}/\partial \mu_{\Theta_k}$, $\partial E_{p(\alpha)}/\partial \sigma_{\Theta_k}^{\mathrm{L}}$, and $\partial E_{p(\alpha)}/\partial \sigma_{\Theta_k}^{\mathrm{R}}$ for the learning of Θ_k can be calculated in the same manner as $\partial E_{p(\alpha)}/\partial \mu_{kh}$, $\partial E_{p(\alpha)}/\partial \sigma_{kh}^{\mathrm{L}}$, and $\partial E_{p(\alpha)}/\partial \sigma_{kh}^{\mathrm{R}}$, the derivatives are as follows:

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{\Theta_k}} = \frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}} \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}}{\partial \bar{\Theta}_k[\alpha]^{\mathrm{L}}} \frac{\partial \bar{\Theta}_k[\alpha]^{\mathrm{L}}}{\partial \mu_{\Theta_k}} + \frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}} \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}}{\partial \bar{\Theta}_k[\alpha]^{\mathrm{U}}} \frac{\partial \bar{\Theta}_k[\alpha]^{\mathrm{U}}}{\partial \mu_{\Theta_k}} = -\,\delta_k[\alpha]^{\mathrm{L}} - \delta_k[\alpha]^{\mathrm{U}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{\Theta_k}^{\mathrm{L}}} = \delta_k[\alpha]^{\mathrm{L}} (-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{\Theta_k}^{\mathrm{R}}} = \delta_k[\alpha]^{\mathrm{U}} (-2\ln\alpha)^{1/2}.$$

Thereafter, the derivatives of $\partial E_{p(\alpha)}/\partial \mu_{hi}$, $\partial E_{p(\alpha)}/\partial \sigma_{hi}^{\mathrm{L}}$, and $\partial E_{p(\alpha)}/\partial \sigma_{hi}^{\mathrm{R}}$ from hidden layer to output layer are:

(1) If $0 \leqslant \bar{W}_{hi}[\alpha]^{\mathrm{L}} \leqslant \bar{W}_{hi}[\alpha]^{\mathrm{U}}$, then

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{hi}} = -\,\delta_h[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{L}} - \delta_h[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{hi}^{\mathrm{L}}} = \delta_h[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{L}} (-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma_{hi}^{\mathrm{R}}} = -\delta_h[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}} (-2\ln\alpha)^{1/2},$$

where

$$\begin{aligned}\delta_h[\alpha]^{\mathrm{L}} &= -\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{ph}[\alpha]^{\mathrm{L}}} = -\frac{\partial E_{p(\alpha)}}{\partial \bar{O}_{ph}[\alpha]^{\mathrm{L}}} \frac{\partial \bar{O}_{ph}[\alpha]^{\mathrm{L}}}{\partial \overline{Net}_{ph}[\alpha]^{\mathrm{L}}} \\ &= -\,\bar{O}_{ph}[\alpha]^{\mathrm{L}} (1 - \bar{O}_{ph}[\alpha]^{\mathrm{L}}) \\ &\quad \times \sum_{k=1}^{n_O} \left(\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}} \cdot \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{L}}}{\partial \bar{O}_{ph}[\alpha]^{\mathrm{L}}} + \frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}} \cdot \frac{\partial \overline{Net}_{pk}[\alpha]^{\mathrm{U}}}{\partial \bar{O}_{ph}[\alpha]^{\mathrm{L}}} \right) \\ &= \bar{O}_{ph}[\alpha]^{\mathrm{L}} \cdot (1 - \bar{O}_{ph}[\alpha]^{\mathrm{L}}) \\ &\quad \times \left(\sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{L}} \geqslant 0}}^{n_O} \delta_k[\alpha]^{\mathrm{L}} \cdot \bar{W}_{kh}[\alpha]^{\mathrm{L}} + \sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{U}} < 0}}^{n_O} \delta_k[\alpha]^{\mathrm{U}} \cdot \bar{W}_{kh}[\alpha]^{\mathrm{U}} \right),\end{aligned}$$

$$\begin{aligned}\delta_h[\alpha]^{\mathrm{U}} &= -\frac{\partial E_{p(\alpha)}}{\partial \overline{Net}_{ph}[\alpha]^{\mathrm{U}}} = -\frac{\partial E_{p(\alpha)}}{\partial \bar{O}_{ph}[\alpha]^{\mathrm{U}}} \frac{\partial \bar{O}_{ph}[\alpha]^{\mathrm{U}}}{\partial \overline{Net}_{ph}[\alpha]^{\mathrm{U}}} \\ &= -\,\bar{O}_{ph}[\alpha]^{\mathrm{U}} (1 - \bar{O}_{ph}[\alpha]^{\mathrm{U}}) \\ &\quad \times \left(\sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{L}} < 0}}^{n_O} \delta_k[\alpha]^{\mathrm{L}} \cdot \bar{W}_{kh}[\alpha]^{\mathrm{L}} + \sum_{\substack{k=1 \\ \bar{W}_{kh}[\alpha]^{\mathrm{U}} \geqslant 0}}^{n_O} \delta_k[\alpha]^{\mathrm{U}} \cdot \bar{W}_{kh}[\alpha]^{\mathrm{U}} \right).\end{aligned}$$

(2) If $\bar{W}_{hi}[\alpha]^{\mathrm{L}} \leqslant \bar{W}_{hi}[\alpha]^{\mathrm{U}} < 0$, then

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{hi}} = -\,\delta_h[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}} - \delta_h[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{L}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{hi}} = \delta_h[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}}(-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{R}}_{hi}} = -\,\delta_h[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{L}}(-2\ln\alpha)^{1/2}.$$

(3) If $\bar{W}_{hi}[\alpha]^{\mathrm{L}} < 0 \leqslant \bar{W}_{hi}[\alpha]^{\mathrm{U}}$, then

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{hi}} = -\,\delta_h[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}} - \delta_h[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{hi}} = \delta_h[\alpha]^{\mathrm{L}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}}(-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{R}}_{hi}} = -\,\delta_h[\alpha]^{\mathrm{U}} \cdot \bar{O}_{pi}[\alpha]^{\mathrm{U}}(-2\ln\alpha)^{1/2}.$$

Finally, the derivatives of $\partial E_{p(\alpha)}/\partial \mu_{\Theta_h}$, $\partial E_{ps}/\partial \sigma^{\mathrm{L}}_{\Theta_h}$, and $\partial E_{ps}/\partial \sigma^{\mathrm{R}}_{\Theta_h}$ for the learning of $\bar{\Theta}_h$ are similar to $\partial E_{p(\alpha)}/\partial \mu_{hi}$, $\partial E_{p(\alpha)}/\partial \sigma^{\mathrm{L}}_{hi}$, and $\partial E_{p(\alpha)}/\partial \sigma^{\mathrm{R}}_{hi}$. The calculations are in the following:

$$\frac{\partial E_{p(\alpha)}}{\partial \mu_{\Theta_h}} = -\,\delta_h[\alpha]^{\mathrm{L}} - \delta_h[\alpha]^{\mathrm{U}},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{L}}_{\Theta_h}} = \delta_h[\alpha]^{\mathrm{L}}(-2\ln\alpha)^{1/2},$$

$$\frac{\partial E_{p(\alpha)}}{\partial \sigma^{\mathrm{R}}_{\Theta_h}} = -\,\delta_h[\alpha]^{\mathrm{U}}(-2\ln\alpha)^{1/2}.$$

Appendix B

1. Volume (∗)
2. Average volume for ten days (∗)
3. Rate of volume change (∗)
4. Index in open (∗)
5. Index in close (∗)
6. Index fluctuation (∗)
7. Rate of index change (∗)
8. Linear weighted moving (∗) average
9. Financing buying(A) (∗)
10. Financing sell(B) (∗)
11. A-B
12. Remaining quota with financing(C)
13. Short "+" & "-"(D)
14. Remaining quota with stock
15. Q/T
16. Turnover rate (∗)
17. MACD (∗)
18. RSI (∗)
19. KD (∗)
20. TAPI (∗)
21. Psychological line
22. ADL
23. BIAS
24. ADR
25. VR
26. Momentum
27. Williams overbought/oversold index

References

[1] N. Baba, M. Kozaki, An intelligent forecasting system of stock price using neural networks, Proc. IJCNN, 1992.
[2] J.J. Buckley, Y. Hayashi, Fuzzy neural networks: a survey, Fuzzy Sets and Systems 66 (1994) 1–13.
[3] D.T. Cadden, Neural networks and the mathematics of chaos – An investigation of these methodologies as accurate predictors of corporate bankruptcy, Proc. 1st Internat. Conf. on Artificial Intelligence Applications on Wall Street, IEEE Computer Society Press, 1991.
[4] S. Cheng, A neural network approach for forecasting and analyzing the price-volume relationship in the Taiwan stock market, Master Thesis, National Jow-Tung University, 1994.
[5] F.L. Fant, K.C. Pamela, A neural network approach to forecasting financial distress, The Journal of Business Forecasting (winter 1991–1992) 9–12.
[6] T. Fukuda, T. Shibata, Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural network, Proc. IJCNN'92, 1992, pp. I-269–I-274.
[7] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[8] D.D. Hawley, J.D. Johnson, D. Raina, Artificial neural systems: a new tool for financial decision-making, Financial Anal. J. 15 (1990) 63–72.
[9] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, MA, 1991.
[10] H. Ishibuchi, K. Kwon, H. Tanaka, A learning algorithm of fuzzy neural networks with triangular fuzzy weights, Fuzzy Sets and Systems 71 (1995) 277–293.
[11] H. Ishibuchi, H. Okada, R. Fujioka, H. Tanaka, Neural networks that learn from fuzzy if-then rules, IEEE Trans. Fuzzy Systems FS-1 (2) (1993) 85–97.
[12] A. Ishikawa, M. Amagasa, T. Shiga, G. Tomizawa, R. Tatsuta, H. Mieno, The max–min Delphi method and fuzzy Delphi method via fuzzy integration, Fuzzy Sets and Systems 55 (1993) 241–253.
[13] G. Jang, F. Lai, T. Parng, Intelligent stock trading decision support system using dual adaptive-structure neural networks, J. Inform. Sci. Eng. 9 (1993) 271–297.
[14] J.-S.R. Jang, Fuzzy modeling using generalized neural networks and Kalman filter algorithm, Proc. 9th National Conference on Artificial Intelligence, July 1991, pp. 762–767.

[15] J.-S.R. Jang, Fuzzy controller design without domain expert, IEEE Internat. Conf. on Fuzzy Systems, 1992, pp. 289–296.
[16] J.-S.R. Jang, C.-T. Sun, Functional equivalence between radial basic function networks and fuzzy inference systems, IEEE Trans. on Neural Networks 4 (1) (1993) 156–159.
[17] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka, Stock market prediction system with modular neural networks, Proc. IEEE Internat. Joint Conf. on Neural Networks, 1990, pp. 1–6.
[18] R.J. Kuo, Intelligent robotic die polishing via fuzzy neural networks, Comput. Industry 32 (3) (1997) 273–280.
[19] R.J. Kuo, A decision support system for sales forecasting through fuzzy neural network with asymmetric fuzzy weights, Dec. Support Systems (1999), accepted.
[20] R.J. Kuo, P.H. Cohen, Integration of artificial neural networks and fuzzy modeling for intelligent control of machining, Fuzzy Sets and Systems 98 (1) (1998) 15–31.
[21] R.J. Kuo, L.C. Lee, C.F. Lee, Intelligent stock decision support system through artificial neural networks and fuzzy Delphi, Comput. Intell. Finance 6 (2) (1998) 24–34.
[22] R.J. Kuo, K.C. Xue, Fuzzy neural network with application to sales forecasting, Fuzzy Sets and Systems (1999), accepted.
[23] S.M. Kendall, K. Ord, Time Series, 3rd ed., Oxford University Press, New York, 1990.
[24] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller – Part I, IEEE Trans. Systems Man Cybernet. 20 (2) (1990) 404–418.
[25] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller – Part II, IEEE Trans. Systems Man Cybernet. 20 (2) (1990) 419–435.
[26] C.T. Lin, C.S.G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Trans. Comput. C-40 (12) (1991) 1320–1336.
[27] C.T. Lin, C.S.G. Lee, Reinforcement structure/parameter learning for neural network based fuzzy logic control systems, IEEE Trans. Fuzzy Systems 2 (1) (1994) 46–63.
[28] C.T. Lin, C.S.G. Lee, Fuzzy adaptive learning control network with on-line neural learning, Fuzzy Sets and Systems 71 (1995) 25–45.
[29] R.P. Lippmann, An introduction to computing with neural nets, IEEE Mag. 3 (4) (1987) 4–22.
[30] S. Mahfoud, G. Mani, Financial forecasting using genetic algorithms, Appl. Artificial Intell. 10 (1996) 543–565.
[31] S. Nakayama, S. Horikawa, T. Furuhashi, Y. Uchikawa, Knowledge acquisition of strategy and tactics using fuzzy neural networks, Proc. IJCNN'92, 1992, pp. II-751–II-756.
[32] E. Schoneburg, Stock price prediction using neural networks: a project report, Neurocomputing 2 (1990) 17–27.
[33] T. Shibata, T. Fukuda, K. Kosuge, F. Arai, Skill based control by using fuzzy neural network for hierarchical intelligent control, Proc. IJCNN, 1992, pp. II-81–II-86.
[34] T. Takagi, I. Hayashi, NN-driven fuzzy reasoning, Internat. J. Approx. Reasoning 5 (1991) 191–212.
[35] L.-X. Wang, J.M. Mendel, Back-propagation fuzzy system as nonlinear dynamic system identifiers, IEEE Internat. Conf. on Fuzzy Systems, 1992, pp. 1409–1418.
[36] F.S. Wong, Time series forecasting using back-propagation neural networks, Neurocomputing 2 (1991) 147–159.
[37] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Systems Man Cybernet. SMC-3 (1) (1973) 28–44.