Journal of the Korean
OR/ MS Society
Vol. 19, No. 2, August 1994

A Knowledge-Based System Using a Neural Network for Management Evaluation and Its Support

Soung Hie Kim*, Kyung Sam Park**, Kuen Chae Jeong**

Abstract

Recently, Decision Support Systems (DSS) research has seen a more to combine Artificial Intelligence (AI) including neural network techniques with traditional DSS concepts and technologies to build an intelligent DSS or a knowledge-based DSS. This article proposes a Management Evaluation and its Support System (MESS) as a knowledge-based DSS. The management evaluation of a firm means the performance of all managerial operations is appraised by considering the situations of the firm. A neural network is used to represent the management evaluation structure as a suitable means of management knowledge representation. Finally a case study in a telecommunication corporation is presented.

1. Introduction

An interest in the application of AI, especially expert systems or knowledge-based systems, to management decision making has existed for some time. Despite this interest, very few management-directed AI applications exist. Perhaps the reason is arisen from a lack of management supporting systems based on traditional AI excluding artificial neural network (ANN) that is due to the difficulties associated with representing unstructured relationships such as those found in many management decision domains.

ANN has attracted a considerable amount of interest in recent years largely due to the growing recognition of the potentials of those networks in performing various works. These works cover a wide range of areas such as knowledge representation, speech processing and pattern recognition [7, 9, 24, 28]. A neural network is theoretically capable of producing a proper out-

* Department of Management Information Systems, KAIST
** Department of Industrial Engineering, KAIST

put to a given problem even when the information is incomplete, confused, and/or unstructured, thus providing the elements of intuition and judgment that are necessary for problem solving [11, 29].

The information system (IS) community has recognized the importance of AI (including ANN) applications in the development of DSS. From a conceptual perspective, DSS and AI seem to differ in focus : AI has focused on expert systems or knowledge-based systems that replace the decision maker in a specific task, while DSS has emphasized systems that supply tools to support a decision maker in large problem domains. A merging of the ideas of the DSS and AI disciplines may result in a more effective management support system [12].

From this point of view, DSS research has seen a more to combine AI-based techniques with traditional DSS concepts and technologies to create intelligent DSS or knowledge-based DSS [1, 3, 4, 10, 16, 17]. Bonczek et al. [2] define a DSS as consisting of three components, a language system, a knowledge base and a problem-processing system. They further state that unless the system contains some knowledge about the decision-maker's problem domain, a DSS is likely to be of little practical value.

In building knowledge-based DSS, a principal issue to be addressed in providing the domain-specific knowledge is how to represent knowledge in a way that is suitable for a particular management domain. Various technologies have existed for this issue such as production rules, frames, semantic nets, neural nets, and other techniques. The guidelines for the choice of a knowledge representation technique should be efficiency and sufficiency, where sufficiency means that the precision of the predictions we obtain meets our requirements, and efficiency refers to their practical applicability. ANN can capture a large number of cases quickly to provide acceptably accurate responses, whereas rule-based system implementation can be a lengthy process depending on the size of the domain and the range of cases that must be considered [14, 22]. The objective of this article is to develop a management evaluation and it's support system based on a neural net knowledge representation approach.

The management evaluation of a firm or corporation means that the performance of all managerial operations is analyzed and evaluated by considering the situations and circumstances of the firm [19]. In one sense, the term management evaluation has been used as financial analysis or performance appraisal [13]. The managers attempt to improve the managerial operations of their firm. For the improvement it is strongly needed to carry out accurate evaluations of managerial performances. Up to the present, these processes have been performed by related managers and experts with their own domain knowledge that contains individual tasks and those hierarchical relationships, and the performance appraisal methods of the tasks.

Some limitations are revealed in these processes that humans have performed, because of the expensive cost in hiring human experts and the time and effort in performing manually. Many decisions are made by the humans without considering all cause−effect relations of the managerial operations, thus a subjective bias is easily intervened. In addition, the human's evaluation process is difficult to be modelled in a traditional AI technique. The recent advance in ANN makes it possible to efficiently represent the human's processes.

In this article, we propose a management evaluation model in which to have hierarchical structure, and use some performance appraisal methods to evaluate the tasks of a firm. The hierarchical structure of management evaluation is represented by a feedforward neural network. With the model, we develop a management evaluation support system, and implement a knowledge−based system that performs the management evaluation of a telecommunication corporation.

2. Artificial Neural Networks

ANN models are specified by the network topology, node characteristics, and training or learning rules. In the topological view, neural nets consist of many nodes linked to each other with connection weights. Then the nodes are usually organized into a sequence of layers with full or random connections between successive layers. The nodes are characterized by how to mathematically operate the input and what kinds of activation functions (such as linear, sigmoid, hyperbolic tangent, etc) are used [20]. The learning rules require an initial set of weights and indicate how the weights should be adapted during use to improve performance of networks.

Generally, ANN has some advantages as follows [27]:

a) Learning : ANN can modify their behavior in response to their environment. This factor, more than any other, is responsible for the interest they have received. Shown a set of inputs (with desired outputs), they self−adjust to produce consistent responses.

b) Generalization : Once trained, a network's response can be, to some degree, insensitive to minor variations in its input. Overcoming the limitation of the traditional computing, it produces a system that can deal with imperfect, ill−structured and error−prone information.

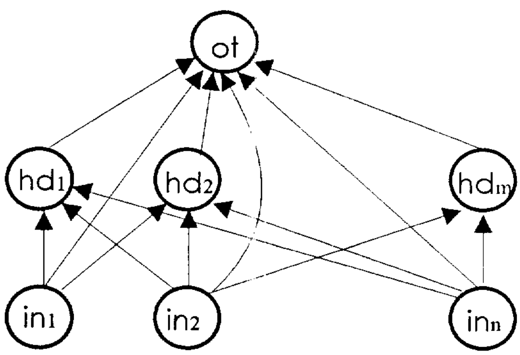
c) Abstraction : Some neural nets are capable of abstracting the essence of a set of inputs. In one sense, it has learned to produce something that it has never seen before.

d) Applicability : ANN can perform many works that conventional computations do poorly or not at all. It typically provides a high degree of robustness or fault tolerance, because knowledge is not contained in one place but is distributed throughout the system. Fault tolerance is that the behavior of the network as a whole is only slightly altered when some nodes or links are destroyed or altered slightly.

There are many kinds of neural networks, for example, feedforward net, Kohonen's self-organization net, Hopfield net (see [7, 27]). Among these, feedforward neural net is composed with one or more layers between the input and output nodes. This network has been used in a stock market forecasting [6], a cancer diagnosis [21], a multiple criteria decision making [26], and many other applications.

An efficient training algorithm known as backpropagation learning rule is proposed by Rumelhart et al. [23]. It is the most popular method based on supervised learning strategy. In the training process with a set of input/output paired samples, the neural net adjusts its internal parameters, i.e., connection weights and thresholds, according to a learning rule to accommodate the training samples. After a number of epoches (or iterations) starting with an arbitrary parameter configuration, the network could then discover the mapping mechanism from the input space to the output one in the sense that an input/output relation is established. It has been proven the feedforward nets are universal approximators for any mapping [8, 18]. It has also been discovered the mappings are often fault tolerant and represent generalizations of examples in the learning process.

This article uses a feedforward network that has an configuration as shown in Figure 1. The input nodes of this network are directly connected to the output nodes as well as to all hidden nodes. It is reported these direct connections are helpful in both training and generalization [8, 18]. Since proposed neural net has the capability of approximating arbitrary mappings, it is greatly efficient this network is used to represent the management evaluation knowledge of the experts.



[Figure 1] A feedforward neural network configuration

The state equations of the neural net are defined as follows :

$$ot = f\left(\sum_{j=1}^{m} w_j^{ho}\, hd_j + \sum_{i=1}^{n} w_i^{io}\, in_i + t^{ot}\right),$$

$$hd_j = f\left(\sum_{i=1}^{n} w_{ij}^{ih}\, in_i + t_j^{hd}\right),\ j = 1, 2, \cdots, m.$$

where, ot = state of output, hd_j = state of activation level of the jth hidden node, in_i = state of the ith input, w_j^{ho} = weight between the jth hidden node and the output node, w_i^{io} = weight between the ith input node and the output node, w_{ij}^{ih} = weight between the ith input node and the jth hidden node, t^{ot} = threshold value of the output node, t_j^{hd} = threshold of the jth hidden node, and $f(u) = 1/(1+e^{-u})$, as an activation function.

The weights and thresholds are updated at each time epoch on the basis of error between the actual output and the target or desired output. The training procedure with error backpropagation algorithm is described as : First, let tot and η respectively be to the target output and the training coefficient, then the following terms are calculated :

$$\delta = ot(1-ot)(tot - ot), \qquad \delta_j = w_j^{ho}\,\delta,$$

$$\Delta w_j^{ho} = \eta\,\delta \cdot hd_j, \qquad \Delta w_i^{io} = \eta\,\delta \cdot in_i, \qquad \Delta w_{ij}^{ih} = \eta\,\delta_j \cdot in_i,$$

$$\Delta t^{ot} = \eta\,\delta, \qquad \Delta t^{hd} = \eta\,\delta_j.$$

Finally, the weights and the thresholds are updated as :

$$w_j^{ho} = w_j^{ho} + \Delta w_j^{ho}, \qquad w_i^{io} = w_i^{io} + \Delta w_i^{io},$$

$$w_{ij}^{ih} = w_{ij}^{ih} + \Delta w_{ij}^{ih}, \qquad w^{hd} = w^{hd} + \Delta w^{hd},$$

$$t^{ot} = t^{ot} + \Delta t^{ot}.$$

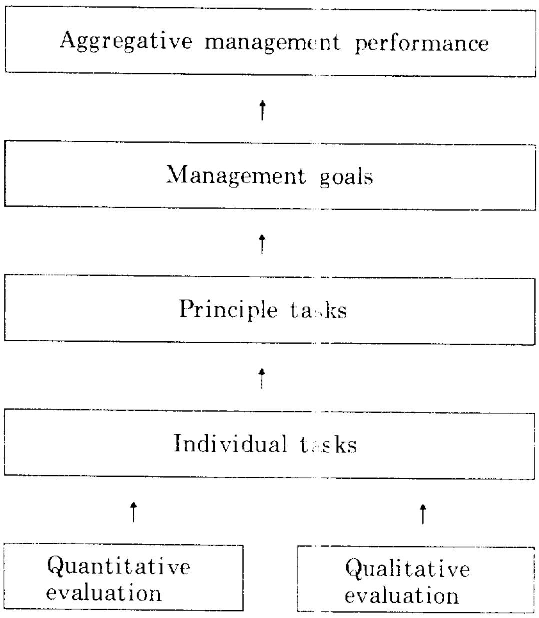
Incorporation of information in addition to that given by training instances into supervised learning, or utilization of prior knowledge of the trainer, is an important aspect of the inductive inference. One way to achieve representability is to ensure that the functional behaviors of the configured feedforward nets comply with monotonicity property for the evaluation functions that will be defined in Section 4. The monotonicity property has been presented by [26].

The backpropagation algorithm does not always find global minimum but may stop at a local minimum (see [7, 27]). However, in most cases the system can usually be driven to the global minimum or as the desired accuracy with appropriate choice of the number of hidden nodes [25]. As the addition of more hidden nodes improves the accuracy of training, the ability of generalization gets worse. Because too many hidden nodes leave too much freedom for the connection weights to adapt during the training. Therefore, the number of hidden nodes must be large enough to form a decision region that can be as complex as required by the given problem, and on the other hand, to be so small that the generalization ability can remain good. Hence, we start with a small number of hidden nodes and increase the number until it becomes possible to drive the learning error to a desired accuracy.

3. Management Evaluations

3.1 The overview

In a firm, there are various types of tasks or jobs. The tasks are categorized into some principle tasks that are the collections or sets of functionally similar individual tasks. A firm also has several goals in operational or managerial level, for example, the improvement of customer service or efficient operation of facilities. As shown in Figure 2, the relationship among goals, principle tasks and individual tasks can be represented as a hierarchical structure.



[Figure 2] Hierarchical evaluation structure

The aggregative managerial performance as a unique measure can be computed by combining the degree of goals' achievements that are calculated by the aggregation of the principle tasks' performances. The performance of principle tasks is also obtained by measuring the performance of individual tasks. The appraisal of individual tasks is carried out in two types of evaluations typically, i.e., quantitative and qualitative evaluation. The management evaluation in a firm is usually performed in a fixed period such as quarter or year. The manager of a firm can map out a strategy or tactics toward an efficient management with the results of the management evaluation. For instance, on a principle task or an individual task that is found out with a low performance, the manager and the humans related with the task can make progress the task's performance by understanding the causes of the low performance of the task.

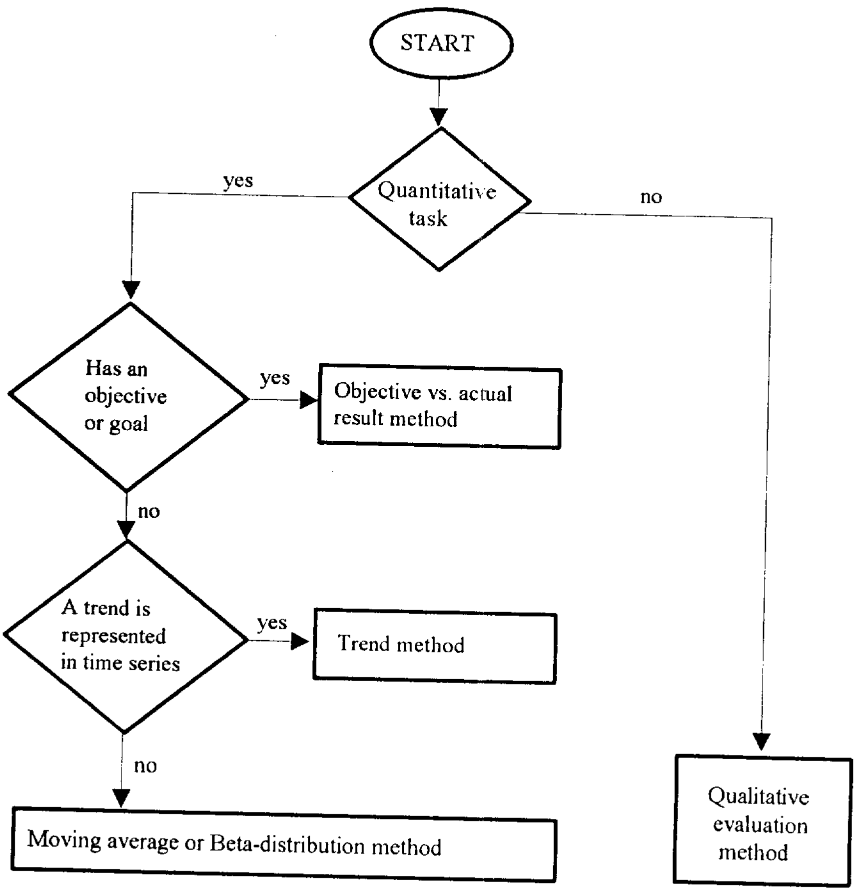
For the management evaluation, it is required to identify the evaluation methods of individual tasks, which is described in the next subsection. It is also needed to develop the aggregation method that in successive hierarchical levels calculates the evaluation score of the above level on the basis of the evaluation scores of the below level, which is described in Section 4.

3.2 Evaluations of individual tasks

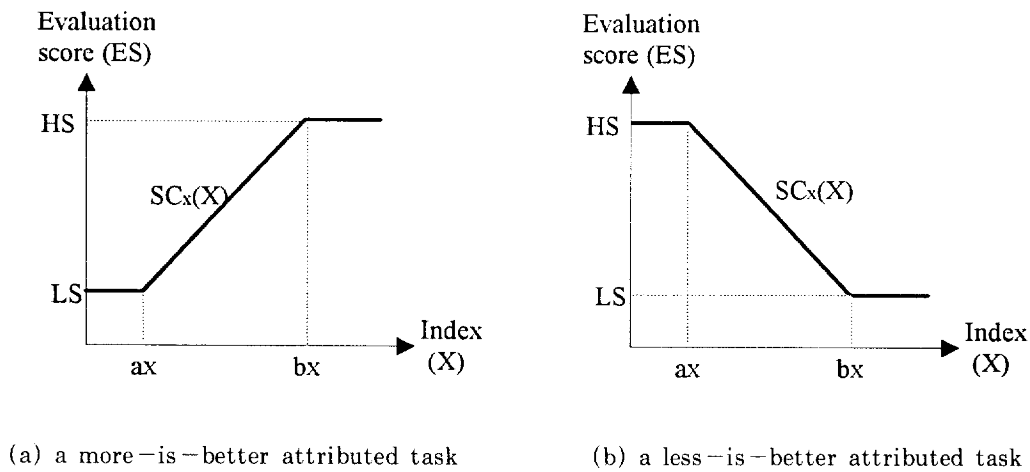
The evaluation results of individual tasks are used in evaluating all managerial performances of a firm. One or more evaluation methods are needed for evaluating the performances of the tasks. These evaluation methods should be developed by considering the inherent properties of each task and the circumstances of the firm. Some guidelines for the development of the methods have been proposed [19] :

a) acceptability : a method must be accepted by the human who performs or related with the task.
b) measurability : a method has to be measurable as a quantitative manner objectively or a qualitative manner subjectively.
c) sufficiency: the performance of a task should be sufficiently evaluated by the corresponding evaluation method.
d) improvability : a method should be developed in a direction that the task performance is improved positively in the future.

Based on the guidelines, we develop a number of evaluation methods, and their selection flow in evaluating individual tasks as shown in Figure 3. Herein it is noted that proposed may be no complete one that can cover all tasks existed in any corporation. An evaluation method could be developed additionally according to the characteristics or attributes of a task. Shown in Figure 3 is developed for a case study of this paper that is described in Section 5. In the quantitative cases, the evaluation methods are consisted of *Objective vs. actual result*, *Trend*, *Moving average or Beta-distribution* method, which are identified as equation types. Otherwise, the evaluation of a task is carried out based on the *evaluation indicators* that are main causes or factors to determine the performance of the task. The evaluation methods are illustrated in detail below.



[Figure 3] Selection flow of an evaluation method



[Figure 4] An example of transformation functions

3.2.1 Objective vs. actual result evaluation

This method is applied to the evaluation of an individual task that has its objective (or goal). It uses two indices in evaluating a task :

$$\text{Objective－achievement rate}(X_1) = \frac{\text{current atual result}}{\text{the stated objective level}},$$

$$\text{Growth rate}(X_2) = \frac{\text{current atual result} - \text{last actual result}}{\text{last actual result}}.$$

These two rates are translated into the corresponding evaluation scores respectively by using a transformation function. An example of this function is shown in Figure 4. Let HS and LS to be the highest score and the lowest score respectively. Then each evaluation score of all tasks in a firm is a value in the interval $[LS, HS]$ that is determined by the managers arbitrarily such as [0, 1], [10, 100], etc. Then, the evaluation scores of a task are calculated by a function, $SC_x(x)$ for $x = X_1$ or X_2, as follows : If a task is a more－is－better attribute,

$$SC_x(x) = \begin{cases} LS, & x < a_x \\ LS + (HS - LS)(x - a_x) / (b_x - a_x), & a_x \le x \le b_x \\ HS, & x > b_x, \end{cases}$$

else if less－is－better attribute,

$$SC_x(x) = \begin{cases} HS, & x < a_x \\ HS - (HS - LS)(x - a_x)/(b_x - a_x), & a_x \leq x \leq b_x \\ LS, & x > b_x. \end{cases}$$

Though the criteria a_x and b_x could be determined by the managers and/or experts according to the situations of a firm, these are obtained from some computations with the data of x for the last few or some years (or quarters). Let $\bar{X}$ be the sample mean, S^2 be the sample variance and μ be the population mean for x, then $\bar{X} = \sum_{i=1}^{n} x_i/n$, $S^2 = \sum_{i=1}^{n} (x_i - \bar{X})^2/(n-1)$ where x_1 is a data in current year, x_2 a data in last year, etc. Let us now consider $100(1-\alpha)\%$ confidence interval of μ, where α is a confidence level such as 0.1, 0.2, etc. Then

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1),$$

where $t(n-1)$ is a t-distribution with parameter $n-1$ (see [15]). When given α, $100(1-\alpha)\%$ confidence interval of μ, $[C_L, C_R]$ is given by

$$\left[\bar{X} - t(n-1, \alpha/2)\frac{S}{\sqrt{n}},\ \bar{X} + t(n-1, \alpha/2)\frac{S}{\sqrt{n}}\right].$$

Herein we use as $a_x = C_L$ and $b_x = C_R$.

After two evaluation scores of a task are obtained respectively, the evaluation score of the individual task ($ESIT$) is calculated by a weighted additive rule :

$$ESIT = \{w_1 SC_{x1}(X_1) + w_2 SC_{x2}(X_2)\} / (w_1 + w_2),$$

where w_i is a weight indicative of the relative importance of the ith evaluation score. The above equation is somewhat subjective rather than objective because the selection of w_i is often based on expert judgment in a firm.

In Objective vs. actual result evaluation method, a reason of using growth rate (X_2) is as follows : In practice, an appropriate objective (or goal) setting is difficult, thus the stated goal level may be too high or too low in the situation that a firm faced. Additionally if it is known that a task is evaluated with only X_1, then an objective would be set to low level. Because very high $ESIT$ value can be derived. In this case, by using growth rate index simultaneously, a more reasonable or acceptable $ESIT$ value can be obtained.

3.2.2 Moving average or Beta−distribution evaluation

This method is used to evaluate a task that has no objective and no trend or pattern in time series. The actual results for the last some time such as three or five years (or quarters) are used for estimating the sample mean ($\bar{X}$) and sample variance (S^2). For computing $\bar{X}$ and S^2, two techniques can be used, simple moving average and Beta−distribution method. Let x_i be the actual result data of a task, then by Moving average technique

$$\bar{X} = \sum_{i=1}^{n} x_i / n, \; S^2 = \sum_{i=1}^{n} (x_i - \bar{X})^2 / (n-2).$$

where x_1 is a data in current year, x_2 a data in last year, etc, or by Beta−distribution technique

$$\bar{X} = \frac{a+4m+b}{6}, \; S^2 = \frac{(b-a)^2}{36},$$

where $a = \max(x_1,\cdots,x_n)$, $b = \min(x_1,\cdots,x_n)$, and m = the arithmetic mean of the x_i except for a and b.

Now an index T is calculated as $T = \sqrt{n}(x_1 - \bar{X})/S$, which follows a t−distribution with parameter $n-1$, $t(n-1)$. Then $ESIT$ is computed by a function $SC(T)$ in the similar way of Objective vs. actual result method : If a task is a more−is−better attribute,

$$SC(T) = \begin{cases} LS, & T < C_L(\alpha) \\ LS + (HS - LS)(T - C_L(\alpha)) / (C_R(\alpha) - C_L(\alpha)), & C_L(\alpha) \le T \le C_R(\alpha) \\ HS, & T > C_R(\alpha), \end{cases}$$

else if it is a less−is−better attribute,

$$SC(T) = \begin{cases} HS, & T < C_L(\alpha) \\ HS - (HS - LS)(T - C_L(\alpha)) / (C_R(\alpha) - C_L(\alpha)), & C_L(\alpha) \le T \le C_R(\alpha) \\ LS, & T > C_R(\alpha), \end{cases}$$

where $C_L(\alpha) = \bar{X} - t(n-1, \alpha/2)S/\sqrt{n}$, and $C_R(\alpha) = \bar{X} + t(n-1, \alpha/2)S/\sqrt{n}$.

3.2.3 Trend evaluation

This method is applied to evaluate an individual task that has no objective but a trend in time series, t. A simple linear regression model, $Y_t=\beta_0+\beta_1 t$, or a curvilinear regression model, $Y_t=\beta_0+\beta_1 t+\beta_2 t^2+,\cdots,\beta_n t^n$, can be used to represent a trend. For the sake of simplicity, we use simple linear (or straight-line) model. The regression coefficients, β_0 and β_1, are estimated from the past results for some time (see [5]), and then the expectation ($\bar{X}$) of current result and the sample variation (S^2) is computed as:

$$\bar{X} = \beta_0 + \beta_1 t_1,$$

$$S^2 = \sum_{t=1}^{n}(Y_t-\beta_0-\beta_1 t_1)^2 \;/\; (n-2),$$

where $t_1=1$ is current time that management evaluation is performed. An index T is calculated as $T=\sqrt{n}(x_1-\bar{X})/S$, where x_1 is the actual result in current time. Then $ESIT$ is obtained by the way of Moving average or Beta-distribution method.

3.2.4 Qualitative evaluation method

This method is used to evaluate the performance of a task that is difficult to be measured in quantity. In fact, there are several (or many) tasks in a firm that do not have quantitative attributes, for example, organization administration or effectiveness of advertisement. Thus the performances of qualitative-attributed tasks ought to be evaluated by experts and/or managers' subjective judgment. Although subjective evaluation, more acceptable or useful evaluation results in overall standpoint of a corporation can be obtained with the consideration of the qualitative-attributed tasks.

The evaluation of a qualitative task is carried out based on the evaluated scores of the indicators, SI that are main causes to determine the performance of the task. Let X be an index for evaluating the task performance, then

$$X = F(SI_1, SI_2,\cdots,SI_n),$$

where n is the number of indicators, and F is typically characterized by one of the following equations :

$$X = \sum_{i=1}^{n} SI_i/n, \quad or \quad \sum_{i=1}^{n} w_i SI_i/n,$$

$$X = (\prod_{i=1}^{n} SI_i)^{1/n}, \quad or \quad (\prod_{i=1}^{n} w_i SI_i)^{1/n},$$

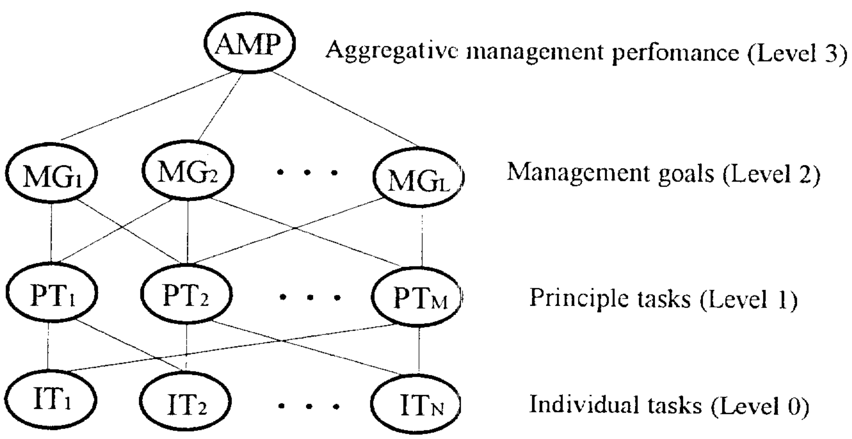
where w_i is a weight indicative of the relative importance of the ith indicator, which has sum−to−one property. Then, $ESIT$ is computed by $SC(X)$, where function SC is similar with Objective vs. actual result method.

The determination of SI is based on the subjective judgment of appraisers. The judgment is often first described in linguistic terms. For example, these terms might be expressed as being in the set {bad, some, good, excellent}, or any other appropriate set. These linguistic terms can be assigned values, for example, {1, 2, 3, 4}.

4. Representation of Management Knowledge

4.1 Representation of hierarchical structure

The management evaluation structure is composed of multi−level hierarchies as shown in Figure 2. In an adjacent level, the evaluation score of a target (that is a management goal or principle task) in the upper level is computed on the basis of evaluation scores in the lower level. An example of this hierarchical structure is shown in Figure 5.



[Figure 5] An example of management evaluation structure

Let us now consider to represent the hierarchical relationships. These can be formally expressed by some functional forms below.

$$ESAMP = f(ESMG_1, \cdots, ESMG_L),$$
$$ESMG_l = g_l(ESPT_1, \cdots, ESPT_M), \text{ for } l = 1, \cdots, L,$$
$$ESPT_m = h_m(ESIT_1, \cdots, ESIT_N), \text{ for } m = 1, \cdots, M,$$

where $ESAMP$ = evaluation score of aggregative management performance,

$ESMG_l$ = evaluation score of the lth management goal,

$ESPT_m$ = evaluation score of the mth principle task,

$ESIT_i$ = evaluation score of the ith individual task, and L, M and N is the number of management goals, principle tasks, and individual tasks, respectively.

The functions f, g_l and h_m are called by *evaluation functions* in this article. The identification of the evaluation functions is difficult in an acceptable manner. Generally most humans would use a weighted average form as an easy approach, for example, $f = \sum_{l=1}^{L} w_l \, ESMG_l$. A difficulty of this approach lies in assessing an appropriate weight w_l.

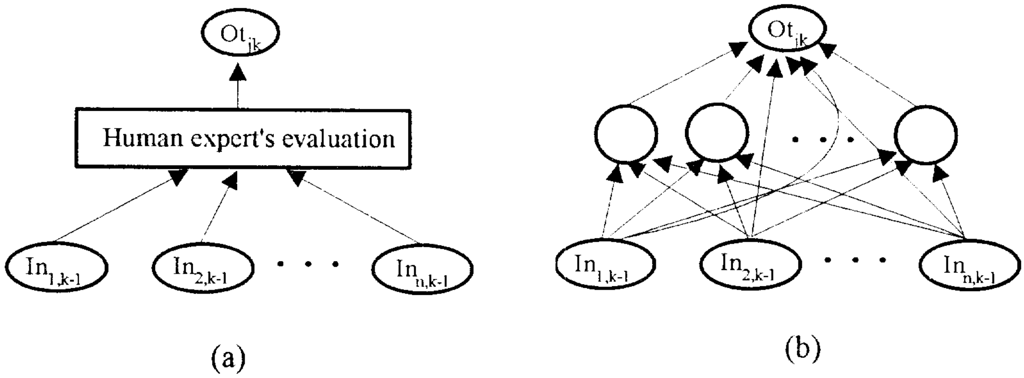
In practice, the evaluation functions are usually characterized by a panel of domain experts and/or managers whether these functions are weighted sum or not. Accordingly the evaluation functions should be determined by gathering and aggregating the knowledge or information of multiple experts. The aggregated management functions are presumably more reliable and useful than an individual's functions because its knowledge is derived from multiple sources. Thus, it is important to develop a suitable means of representing by combining the preliminary considerations of individuals. This article uses a feedforward neural net so as to address this problem.

4.2 Representation based on feedforward neural nets

Let EF be an evaluation function in the set of $\{f, g_l, h_m\}$. The human expert's role shown in Figure 6.a can be expressed by the following evaluation function EF:

$$Ot_{j,k} = EF(In_{1,k-1}, In_{2,k-1}, \cdots, In_{m,k-1}),$$

where $Ot_{j,k}$=evaluation score of the jth target in level k, $In_{i,k-1}$=evaluation score of the ith target in level $k-1$, for k=1,2,3. If $EF=f$, then $j=1$, k=3 and $n=L$, else if $EF=g_l$, then j=1,$\cdots$,L, k=2 and $n=M$, else then j=1,$\cdots$,N, k=1 and $n=N$ (see Figure 5).



[Figure 6] (a) Evaluation process of the expert, (b) Neural network configuration

Feedforward neural nets can represent the aggregative management performance (AMP), management goals (MGs), principle tasks (PTs), individual tasks (ITs), and their relationships (i.e., EF). Each $In_{j,k-1}$ corresponds to each node in input layer of a neural net, and each $Ot_{j,k}$ corresponds to each node in output layer of the neural net. One hidden layer is inserted between the input and output layer for a good representative capability (see Figure 6. b). From our experiments in training neural nets of this study, the number of nodes in the hidden layer, NHN, is recommended as: $\max\{\lceil (NIN + NON + 1)/2 \rceil, 2\}$, where $\lceil u \rceil$ is a minimum of the integers which are greater than or equal to u, NIN is the number of input layer's nodes, and NON the number of nodes in the output layer (According to an accuracy required the number of hidden nodes can be adjusted in flexible).

These nodes are interconnected by the connection weights and have their own thresholds. The weights and thresholds identify the evaluation function EF, after learning the neural nets by training sets. A training set is composed of several training pairs, which can be extracted from a panel of experts.

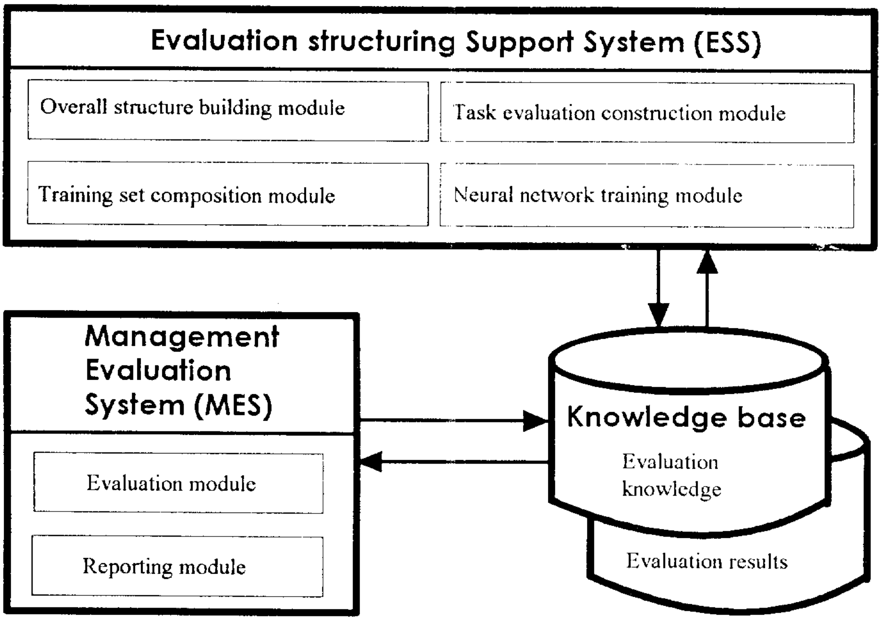
Let a set of training data pairs $\{(A_v, B_v) \mid v = 1,2,\cdots, V\}$, where A_v is an input vector, B_v the desired output vector of expert v, and V the number of vectors (or experts) in the training set. Then $A_v = (In_{1,k-1}(v), In_{2,k-1}(v), \cdots, In_{n,k-1}(v))$ and $B_v = Ot_{jk}(v)$. When given A_v to expert v, then we assume the expert can reply the evaluation score B_v. This paper uses a generalized-delta rule (described in Section 2) for training the neural nets. The trained neural nets ought to be satisfied the monotone nondecreasing property, that is, the output of any evaluation functions is not decreasing when one of the inputs is increased.

In building the knowledge base for management evaluation, the number of neural networks used in this paper is $(1+L+M)$, where L is the number of function g_l and M the number of h_m. Hence, if some changes are occurred in the knowledge base, the knowledge base is able to adapt to changes more easily by adjusting some small parts of the knowledge base, that is, only re-training the neural nets which are affected by those changes.

5. System Implementation and a Case Study

5.1 Overview of MESS

Developed is a Management Evaluation and its Support System (MESS) based on the methodologies presented earlier, which is coded in C (Turbo C) in IBM-PC/AT compatible. MESS is devided into two subsystems, Evaluation structuring Support System (ESS) and Management Evaluation System (MES). The relationship among these subsystems and knowledge base is shown in Figure 7. With ESS, the user can construct the management evaluation structure of his firm in the initial stage, and revise it when the evaluation situation is changed. After constructing the evaluation knowledge base, the management evaluation is carried out by MES.



[Figure 7] System architecture of MESS

There are four modules in ESS. Overall structure building module supports it's user by a natural language-styled query for representing AMP, MGs, PTs, ITs, and their relationships. This module determines the neural network structures and stores the structures in knowledge base, automatically. In task evaluation construction module, knowledge about the evaluation methods of individual tasks is required. The user extracts the training sets with the help of training set composition module. Finally the trained networks by neural network training module are stored in knowledge base for further uses.

5.2 Experiments in training neural nets

Neural network training module in MESS is a simulator based on the backpropagation algorithm. In this paper, initial weights and thresholds are used small random numbers of range [-0.2, 0.2], although any other range could be used. The learning coefficients of the generalized-delta rule are selected such that all of the step sizes with respect to weights and thresholds are 0.9 because they are reported to yield fast learning [23].

We train 8 ($1+L+M=1+4+3=8$, see Table 2) neural networks respectively. We have collected 20 data pairs with the help of 5 experts for each neural net. Among this data pairs, 10 samples are used for training and the remaining samples are used to test the learning performance. We use $NHN_i=\{[(NIN_i+NON_i+1)/2], 2\}$, for $i=1,2,\cdots,8$. The training runs are numbers between 500 and 2000 epoches and the mean square errors are 0.61×10^{-5} through 0.32×10^{-2}, for all neural nets.

Shown in Table 1 is an example of a sampled data in a training and testing set, and the actual output generated by the learned $4(NIN)-3(NHN)-1(NON)$ neural network. The data numbered from 1 to 5 are a part of 10 training samples and the others are a part of 10 testing samples. The training run is 1300 epoches and the mean square error is 0.59×10^{-3}, thus the performance would be quite good. In Table 1, the notations are : PT_3 is a principle task, *technology function*, and IT_i's are individual tasks as : IT_6=*call management*, IT_7=*failure management*, IT_{14}=*improving call quality*, and IT_{15}=*facility management*(see Table 2). Most experts' preferences are as : $IT_{14}\geq IT_7\geq IT_6\geq IT_{15}$.

〈Table 1〉 Data in training and testing samples, and the actual output

No.	Input				Desired output	Actual output
	IT_6	IT_7	IT_{14}	IT_{15}	PT_3	PT_3
1	0.8	0.3	0.2	0.4	0.30	0.298
2	0.8	0.4	0.3	0.5	0.40	0.401
3	0.3	0.6	0.4	0.7	0.50	0.497
4	0.5	0.6	0.8	0.2	0.70	0.700
5	0.9	0.9	0.7	0.8	0.80	0.802
6	0.4	0.2	0.2	0.2	0.20	0.202
7	0.6	0.3	0.4	0.8	0.45	0.448
8	0.8	0.4	0.5	0.7	0.50	0.500
9	0.7	0.6	0.6	0.4	0.60	0.603
10	0.8	0.9	0.9	0.5	0.86	0.868

5.3 An application

The authors performed a management evaluation project of a telecommunication corporation several years ago [19]. The corporation mainly produces mobile telecommunication services such as mobile telephones and pagers. The corporation has four management goals, three principle tasks, and fifteen individual tasks as shown in Table 2. This table would also shown the hierarchical relationships and the evaluation methods for each individual task.

The overall process of evaluating the managerial operations in the corporation is 1) collect the data of the past results, 2) appraise based on the data, and 3) after reporting, announce and feedback the evaluated results. In the past, the appraisal process was performed by gathering and aggregating the knowledge of multiple experts, who consisted of employed experts in external (e.g., members of a consulting company), and experts or managers internally. Time spent in each stage was about 2 weeks, 3 weeks, and 1 week, respectively. The reason of the most time taken in appraising would be caused by tedious opinion pooling of multi-experts, and the reason that one week consumed in reporting was the time of word-processing.

MESS based upon neural net models can reduce expenses in cost and time that are necessary to perform the management evaluation. Particularly the appraisal can be performed in several days because opinion pooling is replaced by neural net training, and the word-processing time is

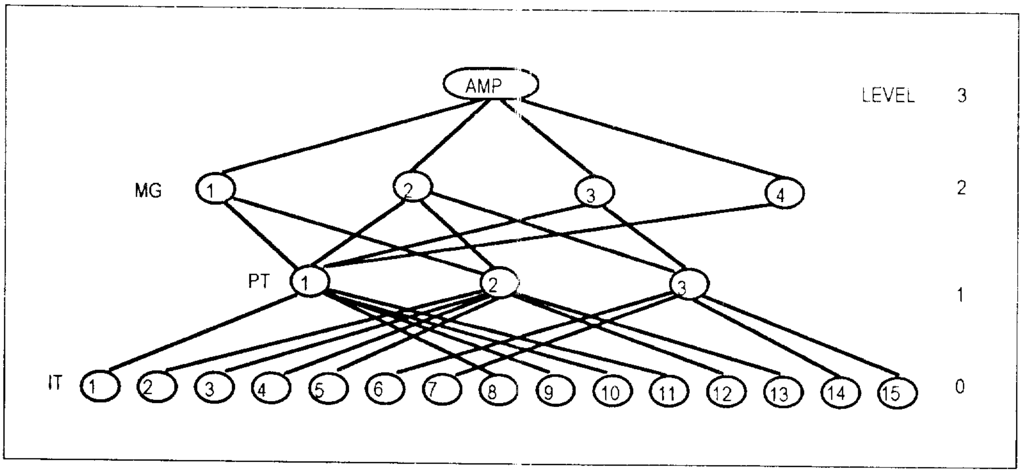
〈Table 2〉 Management evaluation structure of a telecommunication corporation

Level	Notation	Name	Evaluation method	Evaluation indicators
3	AMP	Aggregative management performance	f function	MG_1, MG_2, MG_3, MG_4
2	MG_1	Reinforcement of business	g_1 function	PT_1, PT_2
	MG_2	Improvement of customer service	g_2 function	PT_1, PT_2, PT_3
	MG_3	Efficient operation of facilities	g_3 function	PT_1, PT_3
	MG_4	Efficient management	g_4 function	PT_1
1	PT_1	Mangement function	h_1 function	IT_1, IT_8, IT_9, IT_{10}, IT_{11}
	PT_2	Business function	h_2 function	IT_2, IT_3, IT_4, IT_5, IT_{12}, IT_{13}
	PT_3	Technology function	h_3 function	IT_6, IT_7, IT_{14}, IT_{15}
0	IT_1	General management	Obj. vs. rlt.	–
	IT_2	Business operation	Obj. vs. rlt.	–
	IT_3	Selling operation	Obj. vs. rlt.	–
	IT_4	Market research	Obj. vs. rlt.	–
	IT_5	Tariff management	Obj. vs. rlt.	–
	IT_6	Call management	Obj. vs. rlt.	–
	IT_7	Failure management	Beta – dist.	–
	IT_8	Budget operation	Moving aver.	–
	IT_9	Cost management	Qaulitative	Saving in energy, Economy in expenditures
	IT_{10}	Organization administration	Qaulitative	Coorperation between capital & labor, Education & training
	IT_{11}	Material management	Qaulitative	Requirement planning, Inventory control, Material flow process
	IT_{12}	Business planning	Qaulitative	Forecasting, Advertizment, automation level
	IT_{13}	Customer management	Qaulitative	Kindness & attendance, Replection of customers' opinions
	IT_{14}	Improving call quality	Linear trend	-
	IT_{15}	Facility management	Qaulitative	Maintenance & repair, Reliability

not needed. In addition hiring external experts is not required before the management evaluation structure is largely changed. MESS is able to easily adapt to changes by only re-training the neural networks which are affected by those changes. It can also use to carry out a consistent evaluation without the intervention of appraisers' biases.

Shown in Figure 8 is a screen generated by overall structure building module in ESS. The reporting module in MES shows the evaluation scores for all hierarchical levels as graphical or textual mode (see Figure 9).

Overall Structure Building Module



〈Principle Tasks Level〉

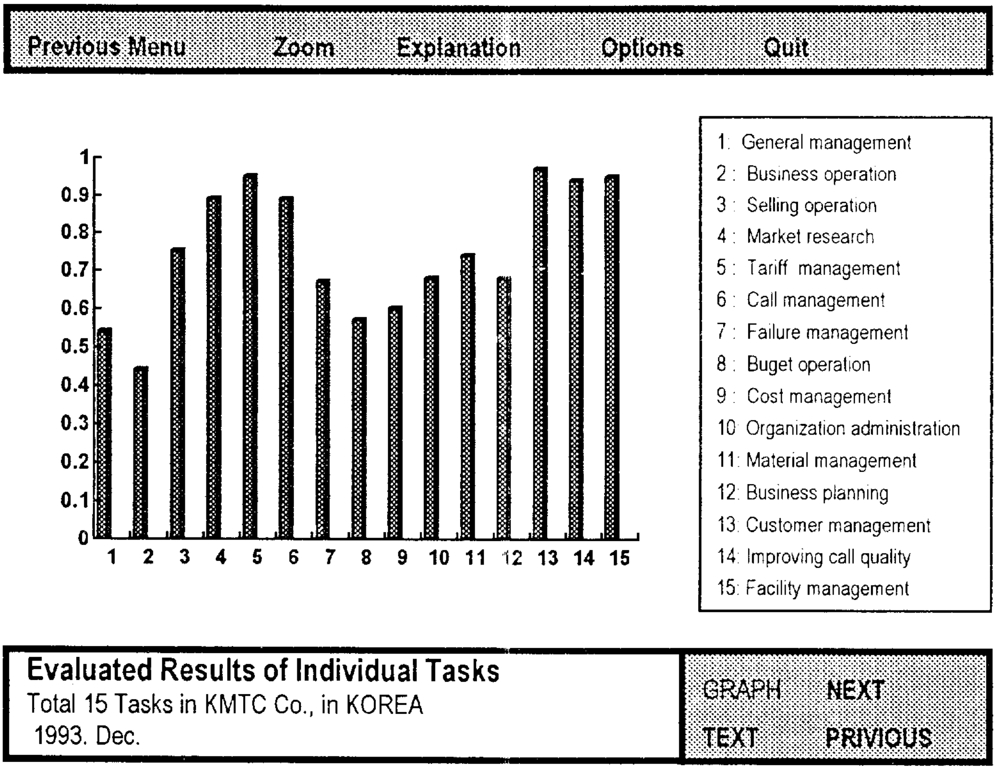
Input name of the 3th Principle Task : Technology function

Input the numbers related Individual Tasks : 6,7,14,15

REVISE

SAVE

[Figure 8] An example in Overall structure building module



[Figure 9] An example of evaluation result in reporting module

6. Concluding Remarks

In this paper, we presented a management evaluation model in which to be a hierarchical structure. The hierarchical structure was represented by a feedforward neural network. The experts' knowledge was easily represented and the evaluation process of the experts was nicely imitated in this model. Based on this model, we developed a management evaluation support system (MESS), and implement a knowledge-based system that performs the management evaluation of a telecommunication corporation. MESS reduced expenses in cost and time that are necessary to perform the management evaluation.

One of the most difficulty in the use of neural nets for management evaluation would be the composition of training sets and the network training that require much efforts and experiments.

Thus much works to address such problem, i.e., systematical approaches based on the experiments, should be conducted in further researches. Additionally, some extensions of MESS are needed for applying management evaluations in any corporations.

References

[1] Arinze, B., Igbaria, M., and Young, L. F., "A Knowledge Based Decision Support System for Computer Performance Management," *Decision Support Systems*, 8 (1992), 501-515.

[2] Bonczek, R. H., Holsapple, C. W., and Whinston, A. B., *Foundations of Decision Support Systems*, New York, Academic Press, 1981.

[3] Courtney, J. F., Paradice, D. B., and Mohammed, A., "A Knowledge-Based DSS for Managerial Problem Diagnosis," *Decision Sciences*, *18*, 3(1987), 373-399.

[4] Dalal, N. P. and Yadav, S. B., "The Design of a Knowledge-Based Decision Support System to Support the Information Analyst in Determining Requirements," *Decision Sciences*, 23 (1992), 1373-1388.

[5] Draper, N. R. and Smith, H., *Applied Regression Analysis*, John Wiley & Sons, 1981.

[6] Felsen, J., "Learning Pattern Recognition Techniques Applied to Stock Market Forecasting," *IEEE Trans. Syst., Man and Cybern.*, *5*, 6(1975), 583-594.

[7] Freeman, J. A. and Skapura, D. M., *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, 19[illegible].

[8] Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, *2* (1989), 183-192.

[9] Gallant, S. I., "Connectionist Expert Systems," *Communications of the ACM*, *31*, *2* (1988), 152-169.

[10] Goul, M., Shane, B., and Tonge, F., "Using a Knowledge-Based Decision Support System in Strategic Planning Decisions: An Empirical Study," *Jr. of MIS*, *2*, 4 (1986), 70-84.

[11] Hawley, D. D., Johnson, J. D., and Raina, D., "Artificial Neural Systems: A New Tool for Financial Decision-Making," *Financial Analysis Journal*, (1990), 63-72.

[12] Henderson, J. C., "Finding Synergy between Decision Support Systems and Expert Systems Research," *Decision Sciences*, *18*, 3 (1987), 333-349.

[13] Henderson, R., *Performance Appraisal Theory to Practice*, Verginia, Restone Publishing Company, 1980.

[14] Hillman, D. V., "Integrating Neural Nets and Expert Systems," *AI Expert*, June (1990),

54-59.

[15] Hogg, R. V. and Craig, A. T., *Introduction to Mathematical Statistics*, Macmillan Publishing Co., 1978.

[16] Hollnagel, E., "Commentary: Issue in Knowledge-Based Decision Support," *Int. Jr. of Man-Machine Studies*, *27* (1987), 743-751.

[17] Holsapple, C. W. and Whinston, A. B., "Management Support through Artificial Intelligence," *Human Systems Management*, *5*(1985), 137-149.

[18] Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforword Neural Networks are Universal Approximators," *Neural Networks*, *2* (1989), 359-366.

[19] Kim, S. H., Park, K. S., Joeng, K. C., Chung, T. Y., and Choi, S. H., *A Study on the Management Evaluation Systems in KMTC* (in Korean), Research Report, Dept. of Industrial Engineering, KAIST in Korea, 1991.

[20] Lippmann, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine, April* (1987), 4-22.

[21] Moallemi, C., "Classifying Cells for Cancer Diagnosis Using Neural Networks," *IEEE Expert, December* (1991), 8-12.

[22] Park, K. S., Kim, S. H., and Kim, J. K., "An Artificial Neural Network Approach to the Decision Class Analysis," *Proceedings of 93Korea/Japan Joint Conference on Expert Systems*, (1993), 717-729.

[23] Rumelhart, D. and McClelland, J., *Parallel Distributed Processing*, Vol. I, Cambridge, MIT Press, 1986.

[24] Shastri, L., "A Connectionist Approach to Knowledge Representation and Limited Inference," *Cognitive Science*, *12* (1988), 331--392.

[25] Sorsa, T., Koivo, H., and Koivisto, H., "Neural Networks in Process Fault Diagnosis," *IEEE Trans. Syst., Man and Cybern.*, *21*, 3 (1991), 815-824.

[26] Wang, J. and Malakooti, B., "A Feedforward Neural Network for Multiple Criteria Decision Making," *Compt. & Operns Res.*, *19*, 2 (1992), 151-167.

[27] Wasserman, P. D., *Neural Computing: Theory and Practice*, New York, Van Nostrand Reinhold, 1989.

[28] Zeidenberg, M., *Neural Network Models in Artificial Intelligence*, London, Ellis Horwood Limited, 1990.

[29] Zahedi, F., "An Introduction to Neural Networks and a Comparison with Artificial Intelligence and Expert Systems," *Interfaces*, *21*, 2 (1991), 25-38.