UDC 612.8:681.513.7

Knowledge-Based Neural Network – Using Fuzzy Logic to Initialize a Multilayerd Neural Network and Interpret Postlearning Results

● Hiroyuki Okada ● Ryusuke Masuoka ● Akira Kawamura

(Manuscript received February 26, 1993)

This paper proposes a neural network system based on fuzzy logic which enables easy conversion between neural networks and fuzzy systems.
In the proposed system, the network is initialized based on existing knowledge, and the network extracts knowledge acquired through learning. The system also implements the time-consuming rules of fuzzy systems automatically and tunes membership functions. This paper also formulates a procedure for building structured neural networks based on a fuzzy inference system. The usefulness of the networks was verified by applying them to bond-rating problems. The percentage of correct answers involving unknown data was 96 % with our networks, compared to 80 % with conventional three-layer neural networks. Also, the time required for learning was reduced by a factor of 40.

1. Introduction

Information processing using neural networks is characterized by knowledge acquisition based on learning. However, before this type of processing can be put into practical use, the problems of traps into local minima that depend on the initial state and difficulties in represnting acquired knowledge must be overcome.

Researchers have tried to combine neural networks and fuzzy theory to solve these problems. Towell's neural network[1], which is based on if-then rules, can acquire new rules through learning. However, using his methods, a network with sufficient accuracy tends to be too large for practical purposes. Horikawa et al. propose a fuzzy modeling technique[2] for simultaneously identifying control rules and tuning membership functions. This technique begins without control rules, and then identifies new control rules through learning. However, we believe that such a process will cause instability in the input-output map during learning, thus making it unsuitable for learning in an actual plant and limiting it to a very narrow range of applications.

The neurofuzzy system we propose converts between fuzzy and neural network models while maximizing the advantages of both[3), 4)]. Neurofuzzy systems convert expert knowledge based on fuzzy rules into neural networks, thus acquiring new knowledge through learning. They then analyze the internal representation of the postlearning neural networks within the framework of fuzzy theory.

Our fuzzy-logic-based neural networks initially obtain knowledge from experts and then acquire additional knowledge through learning.

This paper formulates the procedure for building these neural networks and discusses their application to bond-rating problems. Since our networks are initialized by expert knowledge instead of random numbers, they need a smaller number of learning steps and generalize better than conventional layered neural networks. Also, since the knowledge source obtained is readily identifiable, internal representation can be analyzed using methods similar to the method used in fuzzy inference. Rules and membership functions are automatically tuned using the neural networks' learning facilities, in contrast to the conventional, time-consuming process.

2. Symbolism and connectionism

Since the development of the first widely acknowledged electronic computer (ENIAC) in 1946, people in the scientific community have dreamed of expanding their knowledge by using computers. Symbolism and connectionism are two, often contrasting, approaches to artificial intelligence and are both paradigms of the information processing done by humans.

Symbolism assumes that all human intellectual information processing, including problem solving and inference, is conducted by manipulating symbols. High-level information processing, for example, induction, analogy, and inference from hypotheses, is thus achieved through sequential logic processing based on this manipulation. The symbolism used in the early days of artificial intelligence was therefore considered the most powerful means to implement AI. One reason for this belief was that sequential logic processing of symbols could be easily carried out on machines based on the von Neumann architecture. Fuzzy theory came from the fuzzy set theory proposed by Lotfi Zadeh[5] in 1965. It describes logical knowledge by if-then production rules, and represents subjects and ambiguities by using membership functions. It handles both objective logic and subjective intuition, and is especially useful for implementing intellectual information processing. Using fuzzy theory, we can construct knowledge models representing subjective thinking and judgment, which although vague and ambiguous, can be understood empirically by experts.

Since Mamdani put fuzzy theory into practical use for steam engine control in 1974[6], this field has grown rapidly and now covers a number of control applications. Two examples of these applications are control of a cement kiln[7] and chemical injection control in water purification plants[8].

Connectionism is a massively parallel and highly distributed processing paradigm which is designed to achieve flexible information processing based on intuition or incomplete data. Its basic element is the information processing unit, which exchanges simple messages with other information processing units. Complex processing is represented by a network combining many basic elements. Knowledge is described by the characteristics of each unit and the structure of the network. Connectionist models are flexible systems that perform pattern recognition, and process sense-based information and other intellectual information that is difficult to process with conventional symbolic processing. Neural networks are used to implement connectionism, and are composed of processing units called neurons which have simple functions and exchange information with each other. Parallel and distributed information processing is achieved using the competition-and-cooperation principle. Neural networks also have a training rule whereby the weights of connections are adjusted on the basis of presented examples. This enables the neural network to adapt to changes in the environment. Typical neural networks are interlined networks consisting of one layer with a feedback connection and other layers having multiple players with no connection between the neurons within each layer. Layered neural networks learn using error back-propagation[9], which is a technique used for robot control[10] and image recognition[11].

There are however several problems associated with fuzzy theory:

1) It is difficult to describe pattern information other than symbols.
2) Knowledge from experts is specialized and difficult to generalize.
3) The system must be tuned manually, which requires much time.

One of the main factors limiting the application of fuzzy-logic is the lack of established methods for tuning inference rules and membership functions. Up to now, most practical applications have been in home appliances.

Neural networks also have problems:

1) Logical rules from experts cannot always be directly implemented.
2) Knowledge acquired through learning is distributed over the network.
3) There is no function to explain inference processes.

The inability to extract knowledge acquired through learning in a form that is understandable to humans greatly hinders the practical application of neural networks. This is especially true for plant fault-diagnosis, consulting, and other jobs requiring such an extraction.

3. Building fuzzy-logic-based neural networks

3.1 Describing existing knowledge

Mamdani and other researchers have proposed various techniques for fuzzy inference. Our system uses a multiple fuzzy inference format based on if-then production rules.

Given n rules, the i-th rule is written as:

$$Rule^{i}: \text{if } x_1 \text{ is } A_1^{i^{x_1}} \text{ and } x_2 \text{ is } A_2^{i^{x_2}} \text{ and } \cdots \text{ and } x_m \text{ is } A_m^{i^{x_m}} \text{ then } y \text{ is } B^{i^{y}} \text{ with } w^{i} \quad (i = 1, 2, \cdots, n), \qquad \cdots\cdots\cdots(1)$$

where x_j $(j = 1, 2, \cdots, m)$ is an input variable, y is the output variable, A_j $(j = 1, 2, \cdots, m)$ and B are fuzzy sets defined by membership functions $\mu_{A_j}(x_j)$ and $\mu_B(y)$, i^{x_j} and i^{y} are numbers assigned to fuzzy sets corresponding to input variable x_j and output variable y and indicate the membership functions used by these variables, and w^i is the fuzzy importance assigned to the rule.

Given the input value $x = (x_1^0, x_2^0, x_3^0, \cdots, x_m^0)$, the antecedent fitness of the i-th rule is obtained as:

$$f^{i} = \mu_{A_1^{i^{x_1}}}(x_1^0) \times \mu_{A_2^{i^{x_2}}}(x_2^0) \times \cdots \times \mu_{A_m^{i^{x_m}}}(x_m^0), \qquad \cdots\cdots\cdots(2)$$

where $\times$ indicates a fuzzy logical product operation.

The fitness obtained is used to determine the inference result $\mu_{\bar{B}}{}^{i}(y)$ of the rule as follows:

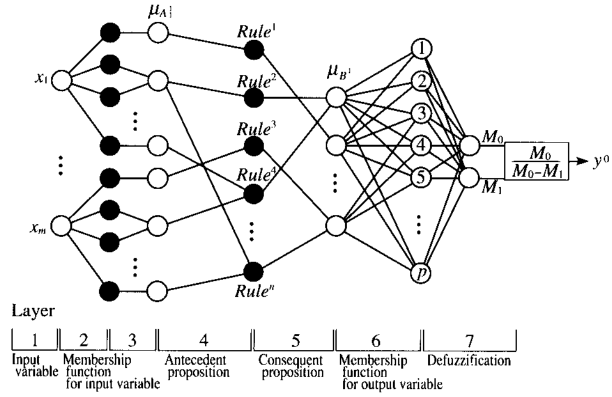


Fig. 1 – Structured neural network similar to fuzzy inference system.

$$\mu_{\bar{B}}{}^{i}(y) = f^{i} \cdot \mu_{B}{}^{i^{y}}(y) \cdot w^{i}, \qquad \cdots\cdots\cdots(3)$$

where $\cdot$ indicates multiplication.

The final inference result $\mu_B(y)$ is obtained by adding the fuzzy sets resulting from inferences based on n rules,

$$\mu_{\bar{B}}(y) = \mu_{\bar{B}}{}^{1}(y) + \mu_{\bar{B}}{}^{2}(y) + \cdots + \mu_{\bar{B}}{}^{i}(y). \qquad \cdots\cdots\cdots(4)$$

The final result of fuzzy inference represented by a membership function is not suitable for robot control or other practical applications, which is why defuzzification is introduced to obtain representative output values. Using the center of gravity method, we obtain the representative value y^0 by calculating the centroid of the final inference result $\mu_B(y)$ as follows:

$$y^0 = \frac{\int y \mu_{\bar{B}}(y)\,dy}{\int \mu_{\bar{B}}(y)\,dy}. \qquad \cdots\cdots\cdots(5)$$

3.2 Conversion to neural networks

The seven-layer structured neural network operates as a fuzzy inference system (see Fig. 1). The number k_{x_i} given to membership function $\mu_{A_i}{}^{k_{x_i}}$ $(i = 1, 2, 3, \cdots, m)$ in layer 3 is the total number of membership functions used by the i-th input variable. The number k_y given to membership function $\mu_B{}^{i}$ $(i = 1, 2, 3, \cdots, k_y)$ is the total number of membership functions used by the output variable. The black circles are sigmoidal-function neurons, and the white circles are linear-function neurons. The input-output activities of these neurons is given by Equation (6) below (see Fig. 2).

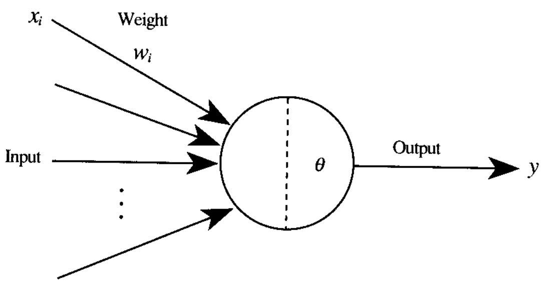


Fig. 2 – Neuron model.

$$S = \sum_i x_i w_i - \theta,$$

$$y = \begin{cases} 1/\{1+\exp(-S)\} & \textit{Sigmoidal} \\ S & \textit{Linear} \end{cases} \quad \cdots\cdots\cdots(6)$$

where x_i is the input to the unit, w_i is the weight of the connections between units, θ is the unit threshold, and y is the output of the unit.

Our proposed network is related to fuzzy inference as follows:

1) Neurons in layers 2 and 3 provide antecedent membership functions.
2) Layer 4 handles antecedent proposition. Fuzzy logical operation is achieved by sigmoidal-function units.
3) The connections with layer 5 represent fuzzy rules. The weight of a connection represents the importance of the corresponding rule.
4) Neurons in layer 5 handle consequent propositions.
5) The weights of connections with layer 6 represent consequent membership functions.
6) Layer 7 and beyond calculate the centroid used in defuzzification.

3.2.1 Antecedent membership functions

The neurons in layers 2 and 3 of the network provide antecedent membership functions.

Fuzzy systems generally use monotonic increasing membership functions. We also use monotonic increasing sigmoidal functions for approximated membership functions, for example, Equation (6). Approximation errors are evaluated by using L^1-norm, which gives the mean of errors; L^2-norm, which gives the sum of the squares of errors; L^∞-norm, which gives the maximum error; and other quantities. A suitable L^2-norm is used to evaluate errors in error back-propagation learning.

Consider the following monotonically increasing membership function:

$$y = \mu_{\text{large}}(x) = \begin{cases} 0 & (x < b) \\ \frac{1}{\mathrm{b-a}} x & (a \le x < b) \\ 1 & (b \le x) \end{cases} . \quad \cdots\cdots\cdots(7)$$

For this membership function, we define the following quantity (L^2-norm):

$$A(w,\theta) = \int_{-\infty}^{\infty} |1/[1+\exp\{-(wx-\theta)\}] - \mu_{\text{large}}(x)|^2 dx . \quad \cdots\cdots\cdots(8)$$

The values of w and θ that minimize $A(w,\theta)$ are $w \simeq -5.3012/(a-b)$ and $\theta \simeq -2.6506(a+b)/(a-b)$.

Also, consider the following monotonic decreasing membership function:

$$y = \mu_{\text{small}}(x) = \begin{cases} 1 & (x < a) \\ \frac{1}{\mathrm{b-a}} x & (a \le x < b) \\ 0 & (b \le x) \end{cases} . \quad \cdots\cdots\cdots(9)$$

For this membership function, we define the following quantity (L^2- norm):

$$B(w, \theta) = \int_{-\infty}^{\infty} |1/[1+\exp\{-(wx-\theta)\}] - \mu_{\text{small}}(x)|^2 dx . \quad \cdots\cdots\cdots(10)$$

The values of w and θ that minimize $B(w,\theta)$ are $w \simeq 5.3012(a-b)$ and $\theta \simeq 2.6506(a+b)/(a-b)$.

Sigmoidal functions approximate triangular and trapezoidal membership functions by combining sigmoidal functions which approximate monotonically decreasing and increasing membership functions.

3.2.2 Fuzzy logical operation

The neurons in layer 4 implement fuzzy logical product operation. In this paper we talk about the bounded product, which is a t-norm (see Fig. 3). A t-norm is a fuzzy logical product operation which is an extend AND operation in crisp set logic. The algebraic product and drastic product are also in general use. Suppose that one neuron approximates the bounded product of input m, and connection weights

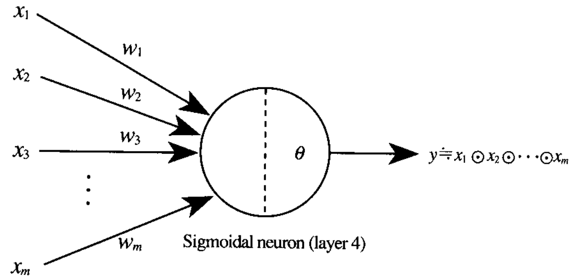


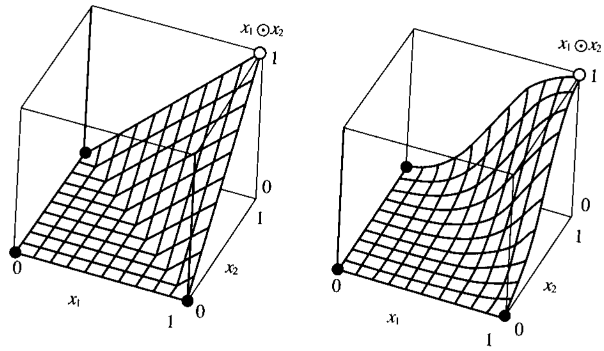
Fig. 3 – Neuron approximating the bounded product.

w_i (i = 1, 2, ···, m) are given an identical value w. Then the neuron's function is governed by $y = 1/[1 + \exp\{-(w\Sigma_{i=1}^{m} x_i - \theta)\}]$ {see Equation (6)}.

Then, the following quantity can be defined to obtain the values of w and θ that minimize $C(w, \theta)$:

$$C(w,\ \theta) = \iint\cdots\int_0^1 |1/[1+\exp\{-(x_1+x_2+\cdots+x_m)-\theta)\}] - (x_1 \odot x_2 \odot \cdots \odot x_m)|^2 dx_1\, dx_2 \cdots dx_m\,. \quad \cdots\cdots\cdots(11)$$

These values depend on the value of m. From the above, we obt... $w_1 = w_2 \simeq 7.0$ and $\theta \simeq 10.5$. Figure 4a) shows the input-output relationship of the bounded product, and Fig. 4b) shows that of the bounded product approximated by a neuron.



a) Bounded product
b) Bounded product approximated by a neuron

Fig. 4 – Fuzzy logical operation.

3.2.3 Consequent membership functions

Consequent membership functions are written by using the weights of the connections between layers 5 and 6, and by assuming that the k-th consequent membership function is defined in the interval [0, 1]. This definition interval simplifies subsequent discussion, but a general interval [a, b] may also be used.

The p neurons in layer 6 correspond to distinct p points $\{y_i \in [0, 1] \mid i = 1, \cdots, p\}$. Function $\mu_{B^k}(y)$ is implemented by the weights of the connections of the k-th neuron in layer 5 with the p neurons in layer 6 {*i.e.* $\mu_{B^k}(y)$ is the input to the i-th neuron in layer 6}, and its values at points other than y_i are determined through appropriate interpolation from the function value at y_i.

3.2.4 Defuzzification

We developed a defuzzification scheme which is based on the calculation of the centroid and is free of difficult division operations. This scheme also maintains the neural network structures and back-propagates errors.

In layer 6, the fuzzy membership function $\mu_{\bar{B}}(y)$, which is the inference result, is represented so that the p neurons in layer 6 correspond to points $\{y_i \in [0, 1] \mid i = 1, \cdots, p\}$ which are equally spaced in the definition interval. The membership function $\mu_{\bar{B}}(y)$ is represented by p neuron outputs.

Layer 7 of the neural network, which is the defuzzification layer, consists of two linear neurons that calculate the signed moments of rotation at $0 \in [0, 1]$ and $1 \in [0, 1]$. That is, the neuron corresponding to 0 calculates $M_0 = \Sigma_i \mu_{\bar{B}}(y_i) \times (y_i - 0)$, and the neuron corresponding to 1 calculates $M_1 = \Sigma_i \mu_{\bar{B}}(y_i) \times (y_i - 1)$.

The centroid is calculated from these outputs using:

$$y^0 = \frac{M_0}{M_0 - M_1}\,. \quad \cdots\cdots\cdots(12)$$

3.2.5 Teaching signals

Teaching signals are given to the structured neural network in one of two ways:

1) An appropriate constant $\alpha > 0$ is taken, and $c \in [0, 1]$ is the centroid that the neural network should learn. When the neuron outputs of layer 7 are M_0 and M_1, the neural network is given $\alpha(0-c)$ and $\alpha(1-c)$ as teaching signals. The error signals are $\alpha(0-c) - M_0$ and $\alpha(1-c) - M_1$ meaning that the teaching signals used here are given by

the y coordinates at $x=0$ and $x=1$ of the line with a slope of α.

2) For M_0 and M_1, which are the outputs of neurons in layer 7, $\alpha = M_1 - M_0$ is taken adaptively, and the neural network is given $\alpha(0-c)$ and $\alpha(1-c)$ as teaching signals and $\alpha(0-c)-M_0$ and $\alpha(1-c)-M_1$ as error signals. That is, a line passes through c that is parallel to the line $[(0, M_0), (1, M_1)]$, and the y coordinates at $x = 0$ and $x = 1$ of that line are treated as teaching signals.

4. Application to bond-rating problems

Bond rating, which indicates the degree of certainty of bond redemption and interest payment as simple symbolic investment information, is used by many investors to help them make decisions. The Japan Bond Research Institute (JBRI), a typical Japanese bond-rating organization, defines its bond-rating definition for bond issuing companies as shown in Table 1.

Bonds are rated according to financial indicators, for example, capital, ordinary income, and debt, by experienced specialists. When rating a bond, experts not only consider quantitative data such as financial indices but also qualitative data items. Therefore, the modeling of bond rating is a very difficult problem for conventional artificial intelligence systems.

4.1 Initialization with existing knowledge

Bond-rating knowledge was used as existing knowledge to initialize the neural networks. The five financial indicators in Table 2 were used to specify simple fuzzy rules serving as initial knowledge.

4.1.1 Describing fuzzy rules

We built a rough bond-rating fuzzy model using the financial indicators in Table 2. Bond rating in Japan tends to judge investment safety based on company size. We used ordinary profit when writing basic rules because this financial indicator is related to company size, and is considered to be closely associated with bond

Table 1. JBRI bond-rating definitions

Rating	Definition
AAA	Indicates the highest degree of protection of principal and interest in the overall judgment of major component factors.
AA	Indicates a high degree of security but falls slightly behind AAA in some component factors.
A	Indicates an acceptably high level overall and excellence in some specific component factors.
BBB	Indicates the medium degree, with promise of security in the future, but issuer requires constant watching.
BB	Indicates uncertainty as to the degree of security when future prospects are taken into consideration.
B	Indicates that credit standing is extremely low and future improvement is considered difficult. Future security is unascertainable at present.

Table 2. Financial indicators

Item	Formula
Ordinary profit	Operating profit + Nonoperating profit − Nonoperating expenses
Owned capital	Capital + Stock issue costs + Legal reserve + Other reserves
Owned capital ratio	$\frac{\text{Owned capital}}{\text{Current assets + Fixed assets + Deferred charges}} \times 100$
Interest coverage ratio	$\frac{\text{Operating income + Interest received + Dividends}}{\text{Interest paid + Discounts}} \times 100$
Long-term debt ratio	$\frac{\text{Fixed liabilities}}{\text{Fixed liablities + Capital}} \times 100$

Table 3. Fuzzy rules

	Rule	Importance
Rule[1]	Rating is high if ordinary profit is high.	1.0
Rule[2]	Rating is medium if ordinary profit is medium.	1.0
Rule[3]	Rating is low if ordinary profit is low.	1.0
Rule[4]	Rating is high if owned capital is high.	0.2
Rule[5]	Rating is low if owned capital is low.	0.2
Rule[6]	Rating is low if owned capital ratio is low.	0.2
Rule[7]	Rating is high if interest coverage ratio is high.	0.2
Rule[8]	Rating is high if interest coverage ratio is low.	0.2
Rule[9]	Rating is high if long-term loan ratio is low.	0.2
Rule[10]	Rating is low if lorfg-term loan ratio is high.	0.2

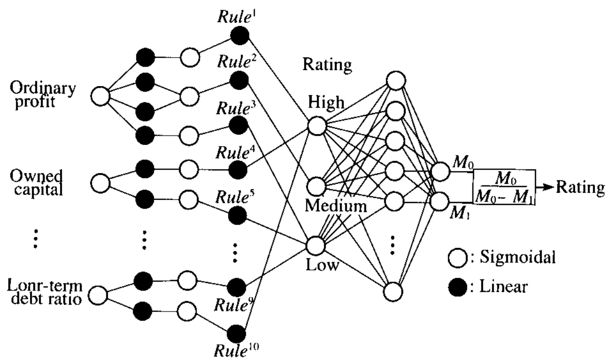


Fig. 5 – Neural network initialized with bond-rating knowledge.

rating. We also wrote auxiliary rules using four other financial indicators. The basic rules related to ordinary profit were assigned an importance of 1.0, and the auxiliary rules were assigned an importance of 0.2. Table 3 shows the fuzzy rules we generated.

4.1.2 Conversion to neural networks

We converted the fuzzy rules that were written as explained in section 4.1.1, "Describing fuzzy rules" into structured neural networks using the method discussed in chapter 3. By using the technique proposed here, knowledge about bond-rating as represented by fuzzy rules can be used to initialize the neural networks.

Figure 5 shows the fuzzy-logic-based structured neural network. The network, which consists of neurons arranged in seven layers, was initialized with bond-rating knowledge. There are 5 neurons in layer 1, 11 in layer 2, 10 in layers 3 and 4, 3 in layer 5, 10 in layer 6, and 2 in layer 7.

4.2 Knowledge acquisition through learning and analysis of acquired knowledge

When the network is taught with financial data about companies whose bonds have already been rated by bond-rating organizations, it notes the differences and connections between the primitive bond-rating data given by us and the data from expert bond-rating organizations. We then return the acquired knowledge to the fuzzy inference framework and analyze the network's internal representation.

4.2.1 Learning

We generated teaching data by referencing bond-rating data published by the JBRI in 1989. We selected 81 Japanese companies in the electric and machinery fields, used the data of 56 of these companies that randomly selected, and evaluated bonds for the 25 remaining companies.

Corporate financial indicators normalized to a [0, 1] interval were then given to the network. Since ordinary profit, owned capital, and the interest coverage ratio had greatly deviated data distributions, we applied abnormal-value elimination and/or logarithmic transformation to change them to appropriate distributions before normalizing them. Table 4 shows the ratings of the companies whose data was used to teach the neural network. The output rating values were normalized to a [0, 1] interval to be used as teaching data.

We used error back-propagation learning. The network was taught the connection weights and thresholds of layer 2 to provide the antecedent membership functions, the connection weights of layer 5 to indicate rule importance, and the connection weights of layer 6 to provide the consequent membership functions. We did not teach the network the connection weights or thresholds of any other layer.

For comparison, the same data was used to teach an ordinary three-layer neural network. This network had five neurons in the input layer and one in the output layer. We also experimented by varying the number of units in the hidden layer and the random numbers used for initialization.

4.2.2 Membership function changes

The membership function indicating a high ordinary profit shifted toward a higher ordinary profit (see Fig. 6). The medium ordinary profit and low ordinary profit membership functions approached each other, and their shapes after learning changed so that the entire area was divided into two at about 50 billion yen. The shape of the low owned-capital ratio membership function became sharper.

By analyzing the changes in the shapes of the membership functions, we learned the following about the relationship between bond rating and ordinary profit and between bond rating and the owned capital ratio:

1) An ordinary profit of about 50 billion yen is the dividing line.
2) The owned capital ratio influences bond rating significantly when it is low.

4.2.3 Changes in rule importance

Rules 1, 5, and 6 became much more important after learning (see Table 5), which means that ordinary profit and owned capital are the most important factors in bond rating.

Rules whose values became negative are considered to have the opposite meaning. Rule 10 changed to an apparent conflict in which the bond rating is high even though the liability is high. When one considers the changes in the importance of Rules 7, 8, and 9 together with Rule 10 after learning, the network seems to have learned that leading enterprises with higher ranking bonds tend to have larger liabilities relative to the business size. This is a noteworthy point peculiar to Japanese company management.

Table 4. Output assignment

Rating	Output
AAA	0.9
AA	0.7
A	0.5
BBB	0.3
BB	0.1

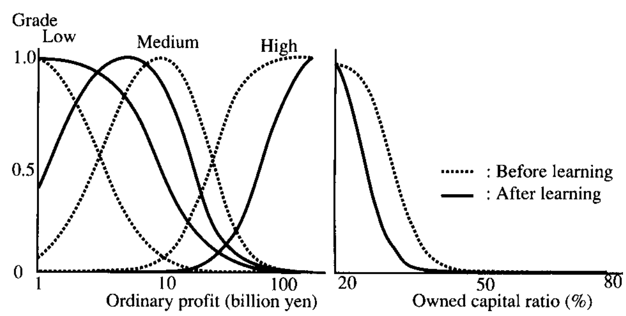


Fig. 6 – Changes in membership functions.

Table 5. Changes in rule importance

	Rule	Before learning	After learning
Rule 1	Rating is high if ordinary profit is high.	1.0	2.5
Rule 2	Rating is medium if ordinary profit is medium.	1.0	0.7
Rule 3	Rating is low if ordinary profit is low.	1.0	0.6
Rule 4	Rating is high if owned capital is high.	0.2	0.8
Rule 5	Rating is low if owned capital is low.	0.2	2.2
Rule 6	Rating is low if owned capital ratio is low.	0.2	2.2
Rule 7	Rating is high if interest coverage ratio is high.	0.2	0.7
Rule 8	Rating is low if interest coverage ratio is low.	0.2	−0.2
Rule 9	Rating is high if long-term loan ratio is low.	0.2	0.0
Rule 10	Rating is low if long-term loan ratio is high.	0.2	−0.1

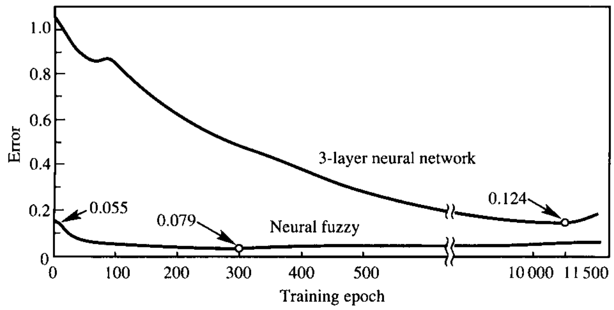


Fig. 7 – Learning.

Table 6. Results of comparison

	Neural fuzzy network	Three-layer neural network	Fuzzy system
Correct answers	96 %	80 %	84 %
Optimum number of learning epochs	300	11 500	–
Sum squared error	0.079	0.124	0.155

4.3 Comparison

We compared the bond rating results of our network with those obtained from a three-layer neural network (see Fig. 7). Our network was structured and initialized according to the fuzzy system. We verified that, before learning, our network achieved the same accuracy as the fuzzy system by using the conversion method proposed in this paper. We assumed that the optimum number of learning epochs minimized the error in evaluation data and that learning had converged when this value was reached.

Table 6 shows the percentages of correct answers that our system and the three-layer neural network gave after the optimum number of learning epochs in response to evaluation data. According to the results of experiments conducted for comparison with the three-layer neural network, the error relating to evaluation data after the optimum number of learning epochs was smallest when the hidden layer had five units.

Our system required a much lower number of learning epochs than the three-layer neural network because bond-rating knowledge is directly implemented in the form of fuzzy rules, so the system begins learning with the accuracy of the fuzzy system (see Table 6).

It can therefore learn much quicker than the neural network, which is initialized by random numbers. In addition, considering the percentage of correct answers relating to data that the system has not learned, our system seems to be superior in terms of generalization capabilities.

The fuzzy system used for initial knowledge can be returned to the accuracy of the neural network. That is, the learning facilities of the neural network can be used to automatically tune rules and membership functions. In contrast, it takes much time to perform such a tuning on conventional fuzzy systems.

5. Conclusion

Structured neural networks, which in terms of performance are equivalent to fuzzy inference systems, have been proposed, and the procedure for building them has been formulated.

We demonstrated that it is possible to initialize a structured neural network with existing knowledge, and explained the internal representation after learning for applications in bond rating. Our network required fewer learning epochs and generalized better than a conventional neural network.

We also showed that our network automatically tunes fuzzy membership functions and rules using learning.

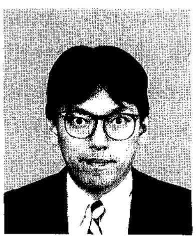
One of the remaining problems is how to extract rules automatically through learning for problems that involve expert knowledge that is difficult to describe using fuzzy rules. The network was initialized using if-then rules described using fuzzy logic, but initialization using a different knowledge format, for example, numerical expressions, could also be considered.

References

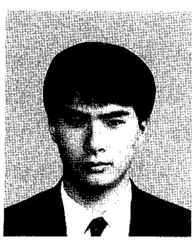
1) Towell, G.G.: "Constructive Induction in Knowledge-Based Neural Networks". Machine Learning, California, Morgan Kaufmann Publishers, 1991, pp. 213-217.
2) Horikawa, S., Furuhashi, T., Uchikawa, Y., and Tagawa, T.: A STUDY OF FUZZY MODEL-

ING USING FUZZY NEURAL NETWORKS. Proc. Int. Fuzzy Eng. Symp. '91, pp. 562-573.

3) Watanabe, N., Kawamura, A., Masuoka, R., Owada, Y., and Asakawa, K.: Fuzzy Control Using Neural Network. Proc. 40th Natl. Conv. IPSJ, 1990, pp. 148-149.
4) Masuoka, R., Watanabe, N., Kawamura, A., Owada, Y., and Asakawa, K.: Neurofuzzy System-Fuzzy Inference Using a Structured Neural Network. Proc. Int. Conf. Fuzzy Logic & Neural Networks, 1990, pp. 173-177.
5) Zadeh, L. A.: Fuzzy Sets. *Inf. Control*, **8**, pp. 338-353(1965).
6) Mamdani, E. H.: Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. *Proc. Inst. Electr. Eng.*, **121**, 12, pp. 1585-1588 (1974)
7) Holmblad, L. P., and Ostergaad, J. J.,: "Control of a Cement Kiln by Fuzzy Logic". *Fuzzy Information and Decision Processes*, Gupta, M. M., and Sanchez, E. eds, North-Holland, 1982, pp. 389-399.
8) Yanagishita, O., Ito, O., and Sugeno, M.: Application of Fuzzy Reasoning to the Water Purification Process. *System and Control*, **28**, 10, pp. 597-604 (1984)
9) Runmelhart, D. E., and McClelland, J. L.: Parallel Distributed Processing. 1st ed., London, MIT Press, 1986, 547p.
10) Nagata, S., Kimoto, T., and Asakawa, K.: Neurocomputer and its Application to Robot Control. (in Japanese), *FUJITSU*, **39**, 3 (Special Issue: FUJITSU TECHNOLOGY '88) pp. 175-184 (1988).
11) Cottrell, G. W., Murano, P., and Zipser, D.: Image compression by Back Propagation – An Example of Extensional Programming –. ICS Report 8702, University of California San Diego, 1987, 24p.



Hiroyuki Okada received the B.S. degree and the M.S. degree in agricultural chemistry from Meiji University, Kanagawa, Japan, in 1984 and 1986. He joined Fujitsu Ltd., Kawasaki, in 1986 and has been engaged in research and development of artificial neural networks.
He is a member of the Information Processing Society of Japan and the Japanese Society for Fuzzy Theory and Systems.



Ryusuke Masuoka received the B.S. degree and the M.S. degree in mathematics from Tokyo University, Tokyo, Japan, in 1985 and 1987.
He joined Fujitsu Laboratories Ltd., Kawasaki, in 1988 and has been engaged in research and development of artificial neural networks.
He is a member of the Information Processing Society of Japan, the International Neural Network Society, the Institute of Electrical and Electronics Engineers, and the Mathematical Society of Japan.

Akira Kawamura received the B.S. degree and the M.S. degree in Information engineering from Nagoya University, Aichi, Japan, in 1985 and 1987.
He joined Fujitsu Laboratories Ltd., Kawasaki, in 1987 and has been engaged in research and development of artificial neural networks.
He is a member of the Information Processing Society of Japan and the Japanese Neural Network Society.