A Learning Method of Fuzzy Inference Rules by Descent Method

Hiroyoshi NOMURA, Isao HAYASHI, Noboru WAKAMI

Central Research Laboratories,
Matsushita Electric Industrial Co., Ltd.
3-15 Yagumo-Nakamachi, Moriguchi, Osaka, 570 Japan
E-Mail nomura@g6.crl.mei.co.jp

ABSTRACT

In this paper, we propose a learning method of fuzzy inference rules by a descent method. From input-output data gathered from specialists, the inference rules expressing the input-output relation of the data are obtained automatically using the proposed method. The membership functions in antecedent part and the real number in consequent part of inference rules are tuned by means of the descent method.

The learning speed and the generalization capability of this method are higher than those of a conventional backpropagation type neural network. Furthermore, this method has a capability to express the knowledge acquired from input-output data in form of fuzzy inference rules.

Some numerical examples are described to show these advantages over the conventional neural network, and an application of this method to a mobile robot that avoids a moving obstacle and its computer simulation are reported.

1. INTRODUCTION

In order to provide fuzzy reasoning with learning function, works are being carried out to combine a fuzzy reasoning and a neural network. Under these efforts, the neural network driven fuzzy reasoning[1], the self-tuning method by Hopfield neural network[2] etc. had been proposed. But, these methods don't have a sufficient generalization capability and a expressing capability of the acquired knowledge.

We have already proposed a self-tuning method[3] of a simplified fuzzy reasoning[4][5] by a descent method[6] to solve these problems. In this paper, the usefulness of this self-tuning method are described in detail.

In this self-tuning method, triangular formed membership functions of the antecedent part and a real number of the consequent part are assumed. The center value and the width of the triangular membership function and a real number of consequent part are tuned by means of the descent method.

The learning speed of this method are higher than that of a conventional backpropagation type neural network[7] since only the inference rules matched input data are tuned. Furthermore, unlike the case of neural network, this method has a capability to express the knowledge acquired from input-output data in form of fuzzy inference rules.

In this paper, the algorithm of the proposed method is explained first, and its high speed learning capability is explained next by referring some numerical examples to which it is applied. In order to demonstrate its generalization capability and expressing capability of acquired knowledge, an application of this method to a mobile robot which is capable of avoiding a moving obstacle is reported.

0-7803-0236-2 /92 $3.00

2. SIMPLIFIED FUZZY REASONING

When the input are expressed by x_1, x_2, .., x_m, and the output is expressed by y, the inference rule of simplified fuzzy reasoning[5] can be expressed by the following.

Rule i:

If x_1 is A_{i1} and ... and x_m is A_{im} THEN y is w_i $\quad (i = 1, \ldots, n)$ (1)

where i is a rule number, A_{i1},...,A_{im} are membership functions of the antecedent part, and w_i is a real number of the consequent part.

The membership function A_{ij} of the antecedent part is expressed by an isosceles triangle shown in Fig. 1. The parameters determining the triangle are the center value a_{ij} and the width b_{ij}.

The output of fuzzy reasoning y can be derived from the equations shown below.

$$A_{ij}(x_j) = 1 - \frac{2 \cdot |x_j - a_{ij}|}{b_{ij}} \qquad (j = 1, \ldots, m) \tag{2}$$

$$\mu_i = A_{i1}(x_1) \cdot A_{i2}(x_2) \cdot \ldots \cdot A_{im}(x_m) \tag{3}$$

$$y = \frac{\sum_{i=1}^{n} \mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \tag{4}$$

where μ_i is a membership value of the antecedent part.

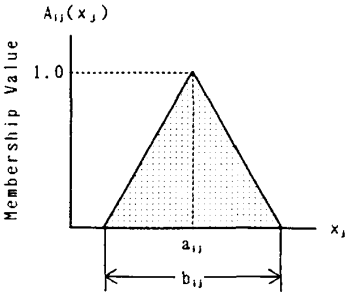

Fig. 1. Membership Function of Antecedent Part

3. ALGORITHM OF SELF-TUNING

The algorithm of self-tuning by a descent method is described below.

A descent method[6] is to seek for the vector Z which minimizes an objective function E(Z), where Z is a p-dimensional vector $Z=(z_1, z_2, \ldots, z_p)$ of the tuning parameters. In this method, the vector which decreases the value of an objective function E(Z) is expressed by $(-\partial E/\partial z_1, -\partial E/\partial z_2, \ldots, -\partial E/\partial z_p)$ and the learning rule is expressed by the following formula.

$$z_i(t+1) = z_i(t) - K \cdot \frac{\partial E(Z)}{\partial z_i} \qquad (i = 1, \ldots, p) \tag{5}$$

where t is a number of iteration of learning and K is a constant. Altering Z according to this learning rule, the value of objective function E(Z) converges into a local minimum[6].

In the present method, the inference rules are tuned so as to minimize the objective function E which is defined by the following.

$$E = \frac{1}{2}(y - y^r)^2 \tag{6}$$

where y^r is the desirable output data acquired from specialists. The objective function E means the inference error between the desirable output y^r and the output of fuzzy reasoning y.

By substituting Eqs. 3 and 4 into Eq. 6, the objective function E can be expressed by the following.

$$E = \frac{1}{2}\left(\frac{\sum_{i=1}^{n}\left(\prod_{j=1}^{m} A_{ij}\right)\cdot w_i}{\sum_{i=1}^{n}\left(\prod_{j=1}^{m} A_{ij}\right)} - y^r\right)^2 \tag{6'}$$

Since the shape of membership function A_{ij} is defined by the center value a_{ij} and the width b_{ij}, the objective function E consists of the tuning parameters a_{ij}, b_{ij}, and w_i ($i = 1,..,n$, $j = 1,..,m$). Therefore, the present method can be an application of the descent method by which the optimum vector Z to minimize the objective function E(Z) can be derived when the vector Z is defined as the follows.

$$(z_1, z_2, \ldots, z_p) = (a_{11},\ldots,a_{nm},b_{11},\ldots,b_{nm},w_1,\ldots,w_n) \quad \text{where } p = 2nm+n \tag{7}$$

From Eq. 5, the learning rules of simplified fuzzy reasoning are expressed by Eqs. 8 to 10.

$$a_{ij}(t+1) = a_{ij}(t) - K_a \cdot \frac{\partial E}{\partial a_{ij}} \tag{8}$$

$$b_{ij}(t+1) = b_{ij}(t) - K_b \cdot \frac{\partial E}{\partial b_{ij}} \tag{9}$$

$$w_i(t+1) = w_i(t) - K_w \cdot \frac{\partial E}{\partial w_i} \tag{10}$$

Eqs. 8 to 10 show respective (t+1)th values of tuning parameters. K_a, K_b and K_w are constants.

The gradients of the objective function ($-\partial E/a_{ij}$, $-\partial E/b_{ij}$, $-\partial E/\partial w_i$) in Eq. 8 to 10 can be derived from the Eqs. 2 to 4, and 6, and are shown in the following.

$$\frac{\partial E}{\partial a_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y - y^r)\cdot(w_i - y)\cdot \mathrm{sgn}(x_j - a_{ij})\cdot\frac{2}{b_{ij}\cdot A_{ij}(x_j)} \tag{11}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y - y^r)\cdot(w_i - y)\cdot\frac{1 - A_{ij}(x_j)}{A_{ij}(x_j)}\cdot\frac{1}{b_{ij}} \tag{12}$$

$$\frac{\partial E}{\partial w_i} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \tag{13}$$

where sgn(z) is a positive or negative sign of z.

Substituting Eqs. 11 to 13 into Eqs. 8 to 10, the learning rules of simplified fuzzy reasoning can be expressed by Eqs. 14 to 16 shown below.

$$a_{ij}(t+1) = a_{ij}(t) - \frac{K_a \cdot \mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i(t) - y) \cdot \mathrm{sgn}(x_j - a_{ij}(t)) \cdot \frac{2}{b_{ij}(t) \cdot A_{ij}(x_j)} \tag{14}$$

$$b_{ij}(t+1) = b_{ij}(t) - \frac{K_b \cdot \mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i(t) - y) \cdot \frac{1 - A_{ij}(x_j)}{A_{ij}(x_j)} \cdot \frac{1}{b_{ij}(t)} \tag{15}$$

$$w_i(t+1) = w_i(t) - \frac{K_w \cdot \mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \tag{16}$$

The learning rules of Eqs. 14 to 16 are to adaptively change the tuning parameters for a direction to minimize the objective function E. Thus, using the learning rules of Eqs. 14 to 16, the tuning parameters of inference rules are optimized to minimize the inference error between the desirable output y^r and the output of fuzzy reasoning y.

A typical iterative learning procedure is shown in the following.

[Step 1] An initial setting of inference rules is conducted first. The initial value of a_{ij} is so set that the domain of input x_j is divided equally. The initial value of width b_{ij} is set to allow overlaps of membership functions.

[Step 2] The input-output data (x_1,...,x_m, y^r) is inputted.

[Step 3] Fuzzy reasoning is performed for the input data (x_1,...,x_m) by using Eqs. 2 to 4. The membership value μ_i of each inference rule and the output of fuzzy reasoning y are derived.

[Step 4] Tuning of a real number w_i of the consequent part is performed by substituting the output of fuzzy reasoning y, membership value μ_i, and output data y^r into Eq. 16.

[Step 5] The fuzzy reasoning conducted in [Step 3] is repeated.

[Step 6] Tuning of the center value a_{ij} and the width b_{ij} of membership functions of the antecedent part is conducted by substituting the changed real number w_i of the consequent part in procedure of [Step 4], the output of fuzzy reasoning y, membership value μ_i, and output data y^r into Eqs. 14 and 15.

[Step 7] The inference error D(t) is calculated from Eq. 17, and [Step 3] to [Step 6] are repeated until its change D(t)-D(t-1) is less than a threshold value.

$$D(t) = \frac{1}{2} (y(t) - y^r)^2 \tag{17}$$

4. NUMERICAL EXAMPLE

In order to demonstrate the validity of the proposed method, some example systems are identified as fuzzy inference rules. Three nonlinear systems of Eqs. 18 to 20 shown below are taken up as examples. These systems have two input x_1, x_2 and one output y.

System 1:

$$y = (2 \cdot x_1 + 4 \cdot x_2^2 + 0.1)^2 \quad (18)$$

System 2:

$$y = 4 \cdot \sin(\pi \cdot x_1) + 2 \cdot \cos(\pi \cdot x_2) \quad (19)$$

System 3:

$$y = (3 \cdot \exp(3 \cdot x_1) + 2 \cdot \exp(-4 \cdot x_2))^{-0.5} \quad (20)$$

The input-output data are prepared by changing the input variables (x_1, x_2) within [-1, 1] using random number, the output data y^r is normalized within [0, 1]. Twenty data are employed each for identification and evaluation in every system. Applying the present method to the input-output data for identification, the inference rules expressing the input-output relation of the data are constructed. In this case, sixteen inference rules are used. This method stops the learning when the inference error E' for the identification data is less than 0.02, where the inference errors E' are derived by Eq. 21 in the following.

$$E' = \sum_{q=1}^{20} (y_q - y^r{}_q)^2 \quad (21)$$

where q mean a number of data.

Table 1 shows the inference errors E' for the identification data and the evaluation data, and the number of iteration for learning, when the learning are stopped.

Table 1. Comparison with Neural Network

System No.		Inference Error for Identification Data	Inference Error for Evaluation Data	Iteration of Learning
System 1	Present Method	0.0194	0.3412	19
	Neural Network	0.0200	0.4393	37268
System 2	Present Method	0.0192	0.0762	19
	Neural Network	0.0200	0.0807	29894
System 3	Present Method	0.0198	0.1422	15
	Neural Network	0.0200	0.1796	25463

Table 1 shows also the inference errors and the number of iteration for learning obtained by neural network using backpropagation[7] applied to the same data. In this case, the neural network consists of three layers including two units in the input layer, sixteen units in the hidden layer, one unit in the output layer.

Table 1 shows that the inference errors attained by the present method for the evaluation data are less than those obtained by the neural network for all of the systems. This proves the higher capability of inference rules attained by this method to express the input-output relation, and the number of iteration for learning are substantially less than those obtained by neural network also. In another word, this means the learning executed in substantially shorter period.

5. APPLICATION TO OBSTACLE AVOIDANCE

The proposed method has a generalization capability. In order to verify the capability, this method is also applied to a computer simulation of a moving obstacle avoidance. The purpose of this experiment is to let a mobile robot have a capability to judge the environment wherein it is placed, and to control its steering to arrive at a target point avoiding a moving obstacle. Fig. 2 shows input variables involved in this moving obstacle avoiding problem.

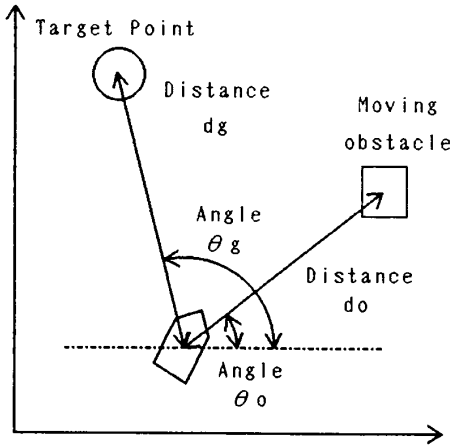

Fig. 2. Input Variables for Moving Obstacle Avoidance

The input-output variables to be considered are shown below.

Input: d_o[m] : Distance between mobile robot and obstacle.
θ_o[deg] : Angle between mobile robot and obstacle.
d_g[m] : Distance between mobile robot and target point.
θ_g[deg] : Angle between mobile robot and target point.
Output : $\Delta\theta_s$[deg] : Change of steering angle of mobile robot.

Precondition shown below for this obstacle avoiding problem are assumed in this case, i.e.,

a) The mobile robot never encounter plural obstacles at a same time.
b) Only the steering angle of robot is controlled, and the driving speed is constant.
c) No dynamics of mobile robot are considered.
d) Although the obstacle is movable, the target point is fixed.

A computer simulation is conducted by taking the following procedure.

1) An initial setting of inference rules is made first according to [Step 1]. Five membership functions are provided for each of four input variables, and 625 inference rules are provided in total. The labels of five membership functions for distance (d_g, d_o) are { VERY NEAR, NEAR, MEDIUM, FAR, VERY FAR }. The labels of five membership functions for angle (θ_g, θ_o) are { LEFT, BACK, RIGHT, FRONT, LEFT }. The domain of θ_o and θ_g is [-180, 180], the domain of d_o and d_g is normalized within [-1, 1], and the clockwise direction of steering angle θ_s is regarded positive. The width of initial membership functions for distance (d_g, d_o) and angle (θ_g, θ_o) are set at 180 degree and 1.0 respectively, all initial real number of consequent parts are set at 0.0.

2) An operator controls the steering angle of mobile robot by using a manual controller for avoiding the obstacle. The avoiding movement of mobile robot is taught for two movement patterns of the obstacle. At this time, the input values (d_g, θ_g, d_o, θ_o) and the change of steering angle ($\Delta\theta_s$) inputted by the operator are acquired as the input-output data ($x_1,\ldots,x_4.y^r$) in [Step 2]. The number of acquired input-output data is 66. The two movements of mobile robot are shown in Fig. 3, which shows the robot(○) and moving obstacle (□).
3) Tuning of fuzzy inference rules is performed by using [Step 3]~[Step 7]. In this case, the number of iteration for learning is set at 10.

In this simulation, 40 inference rules are tuned by this method. The tuned inference rules are shown in Fig. 4 wherein only five inference rules among 40 tuned inference rules are shown. The labels of membership functions in Fig. 4 are the same of the labels of initial membership functions before the tuning.

Next, letting the robot to make an automatic obstacle avoiding movement by using the 40 tuned inference rules. Several typical mobile robot movements for the moving obstacle are shown in terms of the robot tracks shown in Fig. 5.

Figs. 5(a') and 5(b') show that results obtained when the obstacle is moved along previously taught moving patterns shown by Figs. 3(a) and 3(b), the mobile robot left tracks smoother than those which are taught.

Figs. 5(c'), (d'), and (e') show results obtained when the obstacle is moved along moving patterns which are different from the taught pattern. Fig. 5(c') is a case when the obstacle is moved toward the robot by a steeper angle than that originally taught, and Fig. 5(d') shows a case when the obstacle is moved by a smoother angle on the contrary, and Fig. 5(e') shows a case when the obstacle is approached at a steeper angle from an opposite angle.

These experiments show that the mobile robot is capable to avoid the obstacle without colliding even though the obstacle is moved along patterns which are different from the taught patterns. It can be conclude this self-tuning method has a high generalization capability.

The first inference rule in Fig. 4 is understood to mean "If the target point is near to the mobile robot and located in front of mobile robot, the obstacle is near to the mobile robot and located in front of the mobile robot, Then the steering angle is increased by 39.9 degrees". For example, this inference rule can be applied to a *-marked case shown in Fig. 5(a'). The second to fifth inference rules in Fig. 4 represent the characteristics of the taught movement pattern very well.

As shown above, since this method has a capability to express the knowledge acquired from input-output data in form of inference rules, the acquired inference rules can be checked easily by a designer for correcting improper rules caused by noise contained in input-output data. Therefore, this method is, comparing with the neural network, advantageous in respect of the more clear representation of an internal structure of inference processes.

6. CONCLUSION

A learning method of fuzzy inference rules by the descent method is proposed. The membership functions in antecedent part and the real number in consequent part of simplified fuzzy inference rules are optimized by this method. Applications to some numerical examples and moving obstacle avoiding problem are reported. In these applications, high-speed learning capability, generalization capability and capability to express acquired knowledge of this method are shown.

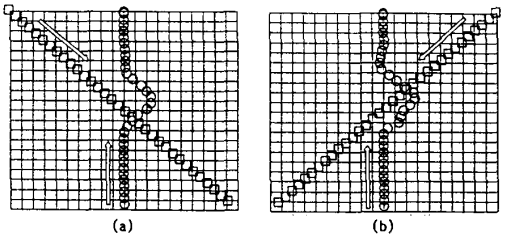

Fig. 3. Robot Tracks Taught by Operator

	Antecedent Part				Consequent Part
	Distance to Target Point dg	Angle to Target Point θg[deg]	Distance to Obstacle do	Angle to Obstacle θo[deg]	Change of Steering Angle △θs[deg]
1	-1.0 NEAR 1.0	-180 FRONT 180	-1.0 NEAR 1.0	-180 FRONT 180	39.9
2	NEAR	LEFT	VERY NEAR	LEFT	10.3
3	VERY NEAR	FRONT	FAR	BACK	5.2
4	VERY NEAR	FRONT	NEAR	BACK	-16.0
5	NEAR	FRONT	NEAR	BACK	-46.4

Fig. 4. Some Optimized Inference Rules

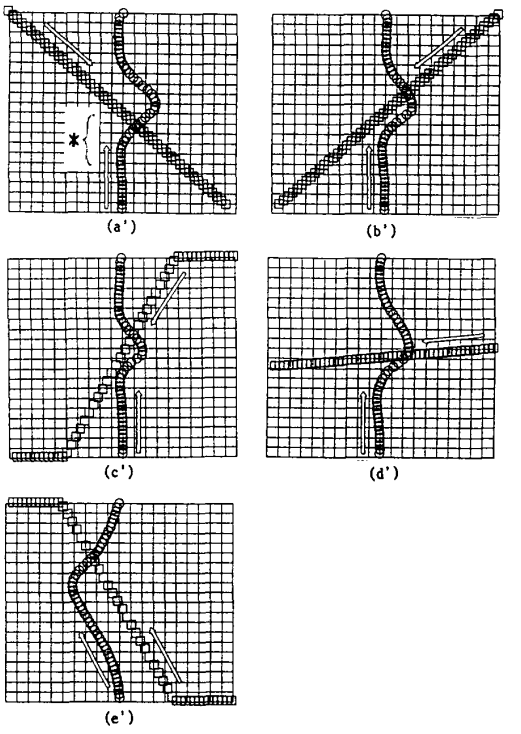

Fig. 5. Robot Tracks Controlled by Optimized Inference Rules

REFERENCES

1. I. Hayashi, H. Nomura and N. Wakami : "Artificial Neural Network Driven Fuzzy Control and Its Application to the Learning of Inverted Pendulum System", Proceedings of 3rd IFSA Congress, pp. 610-613, 1989.
2. H. Nomura, I. Hayashi, N. Wakami : "Self-Tuning Method of Fuzzy Reasoning by Hopfield Neural Network", Proceedings of 5th Fuzzy System Symposium, pp. 177-182, 1989. (in Japanese)
3. H. Nomura, I. Hayashi and N. Wakami : "A Self-Tuning Method of Fuzzy Control by Descent Method", Proceedings of 4th IFSA Congress, Engineering, pp. 155-158, 1991.
4. H. Ichihashi : "Iterative fuzzy modeling and a hierarchical network", Proceedings of 4th IFSA Congress, Engineering, pp. 49-52, 1991.
5. M. Maeda and S. Murakami : "Self-Tuning Fuzzy Logic Controller", Transactions of the Society of Instrument and Control Engineers, Vol.24, No.2, pp. 191-197, 1988. (in Japanese)
6. H. B. Curry : "The Method of Steepest Descent for Nonlinear Minimization Problems", Quart. J. Appl. Math., 2, pp. 258-261, 1944.
7. D. E. Rumelhurt, G. E. Hinton and R. J. Williams : "Learning Internal Representations by Error Propagation", Parallel Distributed Processing, pp. 318-362, 1986.