Fifth IFSA World Congress (1993), 1181-1185

Automatic Generation of Fuzzy Rules using the Fuzzy-Neural Networks

Taechon Ahn*, Sungkwun Oh+, Kwangbang Woo+

*** Dept. of Control & Instrumentation Engineering, Wonkang Univ.**
+ Dept. of Electrical Engeering, Yonsei Univ.

ABSTRACT : In the paper, a new design method of rule-based fuzzy modeling is proposed for model identification of nonlinear systems. The structure indentification is carried out, utilizing fuzzy c-means clustering. Fzzy-neural networks composed back-propagation algorithm and linear fuzzy inference method, are used to identify parameters of the premise and consequence parts. To obtain optimal linguistic fuzzy implication rules, the learning rates and momentum coefficients are tuned automatically using a modified complex method.

I. Introduction

Mathematical models to express dynamic analyses of nonlinear and complex real systems do not give good results. Fuzzy model describing the static or dynamic characteristics[1,3,4] is used as a method to overcome the problems.

But because conventional fuzzy system extracts lingustic fuzzy Implication rules from the heuristic method, the method that estimate the rules from the concrete and systematic method is needed vigorusly. Recently, many reseachers have interests in the fuzzy-neuro fusion with the increasing concerns for the neural network.

Neural network system has the learning capability that can identify fuzzy implication rules and tune membership function that can not be done by the fuzzy inference system. A lot of researches on fuzzy-neuro fusion have been done, utilizing the similarity and relation of mutual compensation for the fuzzy logic and the neural network.

In the early , the technique for the fuzzy-neuro fusion introduced the neural network to the fuzzy inference. On the contrary, recently, the fuzzy inference to the neural network. A study on fuzzy-neuro fusion is variously presented in different viewpoints [1]-[8],[10]-[12].

In the paper, consider the automatic extraction of fuzzy rules from fuzzy linear inference and from configurations of fuzzy-neural network with the fusion. The identification of system is conducted after optimal fuzzy rules is extrated by the autotuning algorithm - a modified complex method.

Box and Jenkin's time series data for gas furnace [9] are used to demonstrate the feasibility and effectiveness of proposed rule-based fuzzy modeling through compari- son of identification errors with conventional fuzzy modeling.

II. Fuzzy neural network

In the paper, the structure of fuzzy-neural network is proposed as figure 1 and the membership functions like figure 2 are used.

II- 1. Premise

Consider the SCM (Soft C-Means) clustering for the identification of structure. The SCM clustering algorithm [13] produces a fuzzy c-partition of the data set $X=\{x_1, x_2, \cdots x_n\}$. The basic steps of the algorithm used in this paper are given as follows:

⟨step 1⟩

Determine the initial parameters.

① Fix the number of clusters, c ($2 \le c \le n$), where n is the number of data.

② Fix m ($1 \le m \le \infty$) and define D_i as geometrical distance.

③ Set p=1 and initialize $U^{(p-1)}$ as 1/c.

⟨step 2⟩

Calculate the cluster centers $v_i^{(p)}$ with $U^{(p-1)}$ and eqn. (1) for the i-th cluster center.

$$v_{il}^{(p)} = \frac{\sum_{k=1}^{n}(\mu_{ik})^m x_{kl}}{\sum_{k=1}^{n}(\mu_{ik})^m x_{kl}} \qquad l=1,\ldots,d \tag{1}$$

where d is dimensiion of x_k, and $\mu_{ik} = \mu_{ik}(x_k)$ is the membership grade of x_k in fuzzy set μ_i

⟨step 3⟩

Update $U^{(p)}$ fot k=1 to n.

① Calculate I_k and I_k'.

$$I_k = \{\, i \mid 1 \le i \le c,\ D_{ik} = \| x_k - v_i \| = 0 \,\} \tag{2}$$

$$I_k' = \{1, 2, \ldots, c\} - I_k \tag{3}$$

② For data k, compute new membership function values.

i) $I_k = \emptyset$ 이면

$$\mu_{ik} = \sum_{i=1}^{c} (D_{ik}/D_{jk})^{2/(m-1)}, \text{ if } \mu_{ik} < (\alpha/c),\ \mu_{ik}=0 \tag{4}$$

$$\mu_{ik} = \sum_{i=1}^{c}\mu_{ik} \tag{5}$$

ii) $I_k \ne \emptyset$ 이면

$$\mu_{ik} = 0 \text{ for all } i \in I_k, \text{ and } \sum_{i \in I_k}\mu_{ik} = 1. \tag{6}$$

③ Next k.

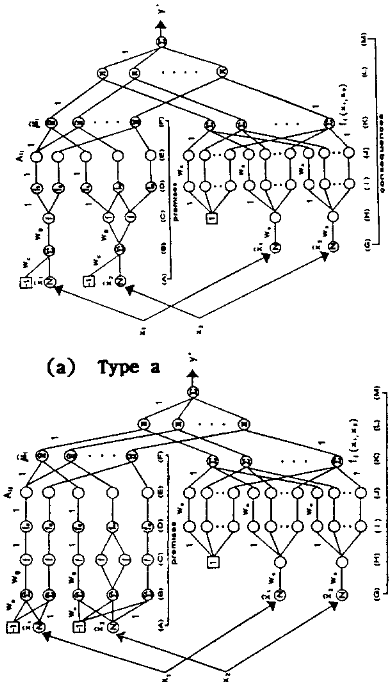
⟨step 4⟩

Compare $J_m^{(p)}$ and $J_m^{(p-1)}$. If $\left|J_m^{(p)} - J_m^{(p-1)}\right| \le \varepsilon$, stops :

$$J_m^{(p)} = \sum_{k=1}^{n}\sum_{i=1}^{c} (\mu_{ik})^m D_{ik}^2 \tag{7}$$

otherwise p = p + 1, and go to step 2.

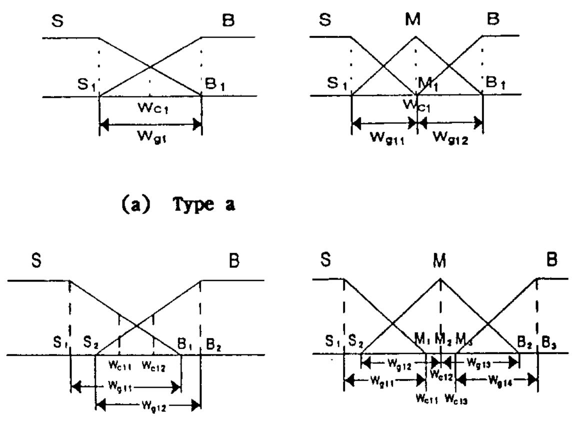
Figures 1 and 2 show 2 or 3 membership functions and 2 inputs in premise. The circles and the squares in figures represent the uints of the network. The denotations w_c, w_g, 1 and -1 between the units mean connection weights. The units with a symbol of 1 in (A)-layer are the bias units with output of unity. x_j denote normalized input variables.



(a) Type a

(b) Type b

Fig. 1. Types of fuzzy-neural network



(a) Type a

(b) Type b

Fig.2. Premise membership functions

The input-output relationships of units with symbols of N in (A)-layer, $\hat{\Sigma}$ in (B)-layer, f in (C)-layer and f_i in (D)-layer are defined as

$$N : \begin{cases} i_j^{(n)} = \sum_k w_{jk}^{(n,n-1)} o_k^{(n-1)} \\ o_j^{(n)} = i_j^{(n)} / \max(i_j^{(n)}) \end{cases} \quad (8)$$

$$\hat{\Sigma} : \begin{cases} i_j^{(n)} = \sum_k w_{jk}^{(n,n-1)} o_k^{(n-1)} \\ o_j^{(n)} = 1/i_j^{(n)} \end{cases} \quad (9)$$

$$f : \begin{cases} i_j^{(n)} = \sum_k w_{jk}^{(n,n-1)} o_k^{(n-1)} \\ o_j^{(n)} = 1/i_j^{(n)} \end{cases} \quad (10)$$

$$f_i : \begin{cases} i_j^{(n)} = \sum_k w_{jk}^{(n,n-1)} o_k^{(n-1)} \\ o_j^{(n)} = f_i(i_j^{(n)}) \quad (i=S,M,B) \end{cases} \quad (11)$$

where $i_j^{(n)}$, $o_j^{(n)}$ are the input and output of the j-th unit in (n)-layer, respectively, $w_{jk}^{(n,n-1)}$ is the connection weight between the k-th unit in (n-1)-layer and the j-th unit in (n)-layer, and $\max(i_j^{(n)})$ denotes the maximum value of $i_j^{(n-1)}$. $f_i(\cdot)$ in (D)-layer of is a triangular function given by

$$f_S(x) = -x + 1/2 \quad (S_1 < x < B_1) \quad (12)$$

$$f_B(x) = x + 1/2 \quad (S_1 < x < B_1) \quad (13)$$

$f_i(\cdot)$ in (D)-layer of figure 5 (b) is a triangular function given by

$$f_S(x) = -x \quad (S_1 < x < M_1) \quad (14)$$

$$f_M(x) = \begin{cases} x + 1 & (S_1 < x < M_1) \\ -x + 1 & (M_1 < x < B_1) \end{cases} \quad (15)$$

$$f_B(x) = x \quad (M_1 < x < B_1) \quad (16)$$

$\hat{\Sigma} - f$ relationships represent that output is reciprocal of input. The units without any symbol just deliver their inputs to succeeding layers. The outputs of the units in (D)-layer are written as

$$o_j^{(D)} = 1/w_g \cdot (\hat{x}_j - w_c) \quad (17)$$

The connection weights w_g, w_c are parameters which determine the central position and the geadient of triangular function, respectively. By approximately initializing the connection weights using the fuzzy clustering method or the maximum and minmum values of data, the membership function in the premises can be allocated on the universe of discourse. The networks initialize the membership function in the premise so that they devide the universe of discourse at the same interval and tune them by modifying their parameters w_c, w_g through the learning.

The truth value of premise in each fuzzy rule is obtained from the outputs of the uints in (F)-layer. The truth value in each fuzzy rule is given by the product of values of the membership functions.

The input-output relationships of the units with a symbol of $\hat{\Pi}$ in (F)-layer are expressed as follows;

$$\hat{\Pi} = \begin{cases} i_j^{(n)} = \prod_k w_{jk}^{(n,n-1)} o_k^{(n-1)} \\ o_j^{(n)} = i_j^{(n)} / (\sum_k i_k^{(n)}) \end{cases} \quad (18)$$

The outputs of the units in (F)-layer are the nomalized truth values by the summation of the inputs $i_j^{(F)}$. The calculation done at the units in (F)-layer can be rewritten as

$$\text{Input} \; : \mu_i = \prod_j A_{ij}(x_j) \quad (19)$$

$$\text{Output} : \hat{\mu}_i = \mu_i / (\sum_k \mu_k) \quad (20)$$

where $\hat{\mu}_i$ is the truth value of the i-th fuzzy rule and μ_i is the normalized value of μ_i. The units in (F)-layer use the center of gravity method.

Ⅱ-2. Consequence

The consequence is expressed by first-order linear equation, and uses the linear (or complex) inference given by

$$R^i : \text{If } x_1 \text{ is } A_{i1}, \ldots, \text{ and } x_k \text{ is } A_{ik}, \text{ then } y = f_i(x_1, \ldots, x_k) \quad (21)$$

$$f_i(x_1, \ldots, x_k) = a_{i0} + a_{i1}x_1 + \ldots + a_{ik}x_k$$

$$y^* = \sum_{i=1}^{n} \mu_i f_i(x_1, \ldots, x_k) \Big/ \left(\sum_{i=1}^{n} \mu_i\right) = \sum_{i=1}^{n} \hat{\mu}_i f_i(x_1, \ldots, x_k)$$

where R^i is the i-th fuzzy rule, x_j is an input variable, A_{ij} is a membership functions of fuzzy sets, a_{ij} is consequence parameters, n is the number of the fuzzy rules, y^* is the inferred value, μ_i is the truth value of R^i in the premises and $\hat{\mu}_i$ is the normalized truth value of μ_i.

(G)~(M)-layers of figure 1 are in the consquences. The connection weights w_s are the nomalizing scaling factors of the input variables in the consequences. $a_{ij}(j=0)$ is equal to the products of w_s and w_a. Normalizing factor w_s is made so that w_a may be independent on input variables. The outputs of the units in (k)-layer are the inferred values of each fuzzy rule $f_i(x_1, x_2)$.

The input-output relationships of the units with symbols of Π in (L)-layer are given by

$$\Pi : \begin{cases} i_j^{(n)} = \prod w_{jk}^{(n,n-1)} o_k^{(n-1)} \\ o_j^{(n)} = i_j^{(n)} \end{cases} \quad (22)$$

The products of μ_i and $f_i(x_1, x_2)$ are calculated in (L)-layer. The sum of the products in (M)-layer is the inferred value of the fuzzy inference. Therefore, the fuzzy inference is realized by eqn.(21).

The connection weights in the consequences w_s are initialized to be 1 as the nomalized scaling factors of input variables and are identified by modifying w_s through the learning. The connection weights w_a are initialized to be zeros and the fuzzy rules are identified by modifying w_a through the learning from the condition with no fuzzy rules.

Ⅱ- 3. Learning and Tunning Algorithms

The FNNs identify the fuzzy rules and tune the membership functions by modifying the connection weights through the learning. Their learning algorithm is based on the back-propagation (BP) method which apply the delta rule to the general multi-layer neural network. In case that the number of the input-output data is N, the cost function of output error is given by

$$E_p = \frac{1}{2} \sum_j (y_{pj} - y_{pj}^*) \quad (23)$$

$$E = \sum_{p=1}^{N} E_p \quad (24)$$

where y_{pj} is the j-th output value of the p-th input-output data (the learning data) and y_{pj}^* is the inferred value of the fuzzy-neural network for y_{pj}.

The minimization of output error E for overall input-output data is equal to the minimization of output error E_p for the p-th learning data. Then, the change $\Delta w_{ji}^{(n,n-1)}$ of connection weighs $w_{ji}^{(n,n-1)}$ is obtained using gradient decent rule as follows;

$$\Delta w_{ji}^{(n,n-1)} = \eta\left(-\frac{\partial E_p}{\partial w_{ji}^{(n,n-1)}}\right) \quad (25)$$

where η is learning rate.

Eqn.(25) can be rewritten as

$$\Delta w_{ji}^{(n,n-1)} = \eta \cdot \delta_j^{(n)} \cdot o_i^{(n-1)} \quad (26)$$

In output layer, $\delta_j^{(n)}$ is rewritten by the input-output relationships of the units a symbol of Σ as follow;

$$\delta_j^{(n)} = y_{pj} - y_{pj}^* \quad (27)$$

In hidden-layers, because the input-output relationships are combined as the type of different units, the inner function of the units is separately investigated in the case of single input or multi-inputs. The inner functions of the units with symbols of Σ , f and f_i are considered. In these units, since each input of units in succeeding layers is calculated as the sum and produt, $\delta_j^{(n)}$ is given by

$$\delta_j^{(n)} = F'(i_j^{(n)}) \sum_k \delta_k^{(n+1)} w_{kj}^{(n+1)} \cdot \left[\prod_{i \neq j} w_{ki}^{(n+1,n)} o_i^{(n)}\right] \quad (28)$$

where F means the units with symbols of f or f_i. $F'(\cdot)$ denotes the derivative of an inner function of the unit. The derivatives of linear units except units with symbols of f or f_i are to be 1 or constant. The inner functions denote the multi-inputs with symbols of Π . Each input of the units in succeeding layers is calculated as the sum and the product.

The modification of the connection weights w_c, w_g in the premise is stabilized rather than decreased by the influences of the outputs in (B)-layer and the fuzzy rules are identified by the connection weights.

$$w_{ji}^{(n,n-1)}(m+1) = w_{ji}^{(n,n-1)}(m) + \eta\delta_j^{(n)}(w_{ji}^{(n,n-1)}(m))^2 o_i^{(n-1)} + \alpha\cdot(w_{ji}^{(n,n-1)}(m) - w_{ji}^{(n,n-1)}(m-1)) \quad (29)$$

where the learning rate η and the coefficient of momentum α is autotuned by using the modified complete method and the optimal fuzzy rules are extracted at η and α.

In the parameter identification of the premises and consequences, in order to solve the optimal values of the learning rate and the coefficients of momentum we realize the algorithm to expand and unite the simplex concept - constrained optimization technique - to the complex method. The algorithm called as modified complex method is the constrained simplex method that minimizes the cost function, as follows:

$$\text{Minimize } f(X) \quad (30)$$

$$\text{Subject to } g_j(X) \leq 0, \quad j=1,2,\cdots,m$$

$$X_i^{(l)} \leq X_i \leq X_i^{(u)} \quad i=1,2,\cdots,n$$

where l denote the lower bound and u, the upper bound. The algorithm had been studied and published by authors[14].

Ⅱ. Simulation and Results

In the paper, 296 pairs of input-output data of the time series for gas furnace are used in order to do rule-based fuzzy modeling from the inference by fuzzy-neuro fusion and to evaluate the effectiveness and applications. Fuzzy rules are extracted from the input-output data, that is, the flow rate of gas and the burned carbon dioxid density. u(t-3) and y(t-1) are used as input variables for fuzzy modeling and y(t) as output variable. u(t) means the flow rate and y(t) denotes the burned carbon dioxide density.

The membership values are calculated by fuzzy clustering for structure identification of premise as figure 3.

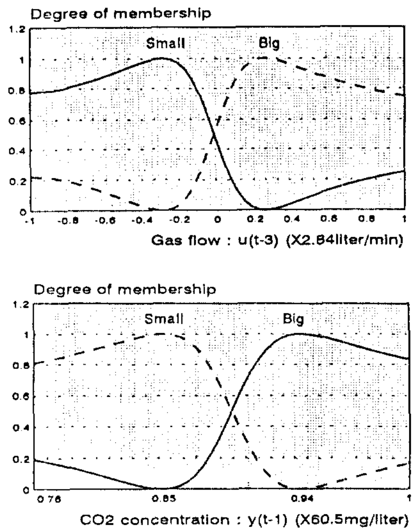


Fig. 3. Fuzzy partitioned membership values of type b

Table 1 shows initial connection weights calculated from the membership functions of input fuzzy variables on types a and b. These initial connection weights are used to identify membership functions of each input fuzzy variable through the learning of the fuzzy-neural networks.

Table 1. Initial values of premise connection weights in the number of rules 4.

Initial values			
Type a		Type b	
w_{c1}	0.0208	w_{c11}, w_{c12}	-0.0467
w_{g1}	0.9791	w_{g11}, w_{g12}	0.5302
w_{c2}	0.8787	w_{c21}, w_{c22}	0.8893
w_{g2}	0.1212	w_{g21}, w_{g22}	0.0852

Table 2 shows the initial learning rate and momentum coefficients of connection weights, and the final learning rate and momentum coefficients obtained by the extracted optimal rules.

Table 2. Initial and final learning rates and momentum coefficients of connection weights

		Initial values		Final values				
P. S.	C.W.	η	α	η	α	I.	R	PI
Type a	w_c	1.0e-5	0.01	1.7e-5	0.125	u(t-4) y(t-1)	4	.103
	w_g	1.0e-5	0.01	1.6e-5	0.125			
	w_a	0.01	0.01	0.013	0.0125			
	w_s	0.01	0.01	0.015	0.0125			
Type b	w_c	1.0e-5	0.01	1.3e-7	0.111	u(t-3) y(t-1)	4	.096
	w_g	1.0e-5	0.01	1.5e-5	0.098			
	w_a	0.01	0.01	0.014	1.13e-4			
	w_b	0.01	0.01	0.014	1.17e-4			

Briefly, consider only type b. Type b is consisted of four fuzzy rules by four fuzzy partitions using the identification data as eqn.(31). Input variables - u(t-3) and y(t-1), use Small and Big as fuzzy variable, respectively. Output vari- able y(t) uses four linear equations as follows. The identified parameters of each fuzzy rule are like table 3.

R^1:If u(t-3) is Small & y(t-1) is Small,

then $y = a_{10}+a_{11}u(t-4)+a_{12}y(t-1)$

R^2:If u(t-3) is Small & y(t-1) is Big ,

then $y = a_{20}+a_{21}u(t-4)+a_{22}y(t-1)$

R^3:If u(t-3) is Big & y(t-1) is Small,

then $y = a_{30}+a_{31}u(t-4)+a_{32}y(t-1)$

R^4:If u(t-3) is Big & y(t-1) is Big ,

then $y = a_{40}+a_{41}u(t-4)+a_{42}y(t-1)$

(31)

Table 3. Identified parameters for optimal fuzzy rules of type b

premis				consequence			
w_{c11}	0.0210	w_{c21}	1.0030	a_{10}	6.4075	a_{11}	-0.2825
w_{c12}	0.0205	w_{c22}	0.7109	a_{12}	10.0707	a_{20}	6.8748
w_{g11}	1.8843	w_{g21}	0.2397	a_{21}	-1.0137	a_{22}	8.8673
w_{g12}	1.9794	w_{g22}	0.2375	a_{30}	10.0888	a_{31}	7.9369
				a_{32}	8.5487	a_{40}	5.9818
				a_{41}	-1.0366	a_{42}	7.9289

Type b shows the best among performances of the optimal fuzzy rules extracte by the learning through the fuzzy-neural networks. The output data are compared with real data for gas furnace as figure 4.

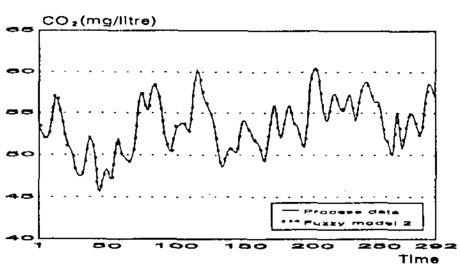


Fig.4. Comparison of original data and output data for Type b

The identifiction error (or performance index) is compared with other fuzzy modeling methods using the identification data in table 4. Evaluation criterion is like eqn. (32).

$$PI = (1/N)\cdot\sum_{k=1}^{N}(y(k)-y^{o}(k))^2 \tag{32}$$

where N is the data number, y(k) the output value of real data and $y^0(k)$ the inferred value from identified model.

Table 4. Comparison of identifiction error with other fuzzy modeling methods

Model name		Mean quence error	Number of rules
Tong's model		0.469	19
Pedrycz's model		0.776	20
Xu's model		0.328	25
Sugeno's model		0.355	6
Hwang's model		0.166	6
Our model	Type a	0.103	4
	Type b	0.096	4

Fuzzy modeling method using the proposed fuzzy-neural networks is more excellent performance than other fuzzy modeling methods.

Ⅳ. Conclusions

In the paper, efficient identification techniques are pesented that automatically extract the optimal fuzzy rules, using the fuzzy clustering method in the structure identification, the BP algorithm of the neural networks in the parameter identification and the advanced complex method which is an autotuning algorithm proposed. Here, the linear fuzzy inference is inroduced to identification methods.

As the proposed rule-based fuzzy modelings are applied to the process of gas furnace, identification errors are like 0.103 in type a and 0.096 in type b. The performance is more excellent than the conventional fuzzy modeling with the identification error 0.355 and the nuber of fuzzy rules 6. The feasibility and effec- tiveness of the new modeling is proved in data for gas furnace process.

Finally, the proposed method will be also applied to the system of sewage treatment process.

References

[1] T.C. Ahn et.al, "Rule-based fuzzy modelings", to be submitted KIEE . 1993.