The Third AFSS(1998), 391－394

Notes on Conventional Neuro-Fuzzy Learning Algorithms

Yan SHI[1] Masaharu MIZUMOTO[2]

[1]School of Engineering, Kyushu Tokai University
9-1-1, Toroku, Kumamoto 862-8652, Japan
Tel: +81-96-386-2666 Fax: +81-96-381-7956 E-mail: shi@ktmail.ktokai-u.ac.jp
[2]Division of Information and Computer Sciences, Osaka Electro-Communication University
18-8, Hatsu-cho, Neyagawa, Osaka 572-8530, Japan
Tel: +81-720-20-4569 Fax: +81-720-24-0014 E-mail: mizumoto@mzlab.osakac.ac.jp

Abstract

In this paper, we try to analyze two kinds of conventional neuro-fuzzy learning algorithms, which are widely used in recent fuzzy applications for tuning fuzzy rules, and give a summarization of their properties. Some of these properties show that uses of the conventional neuro-fuzzy learning algorithms are sometimes difficult or inconvenient for constructing an optimal fuzzy system model in practical fuzzy applications.

Keywords: Neuro-fuzzy learning algorithms, Tuning fuzzy rules, Simplified fuzzy reasoning method

1. Introduction

In recent fuzzy applications, it is getting more important to consider how to design optimal fuzzy rules from training data, in order to construct a reasonable and suitable fuzzy system model for identifying the corresponding practical systems. However, when a fuzzy system model is designed, it is sometimes too hard or impossible for human beings to give desired fuzzy rules or membership functions, due to the ambiguity, uncertainty or complexity of the identifying system. Due to the above reasons, it is natural and necessary to generate or tune fuzzy rules by some learning technique. By means of the back-propagation algorithm of the neural networks [5], so-called "*neuro-fuzzy learning algorithms*", which are widely used in recent fuzzy applications for generating or tuning optimal fuzzy system models [8,9], have been proposed by Ichihashi [1], Nomura et al. [4], Wang and Mendel [7], independently.

In this paper, we try to analyze two kinds of neuro-fuzzy learning algorithms and give a summarization of the properties. Some of their properties show that uses of the conventional neuro-fuzzy learning algorithms are difficult or inconvenient sometimes for learning a fuzzy system model in practical fuzzy applications.

2. Conventional neuro-fuzzy learning algorithms

In what follows, we shall explain briefly two kinds of approaches of neuro-fuzzy learning algorithms.

2.1 *Two Forms of Fuzzy Inference Models*

Let $x_1, x_2, \ldots, x_m$ be variables on the input space $X = X_1 \times X_2 \times \ldots \times X_m$, and y be a variable on the output space Y, then two forms of fuzzy inference rules by the fuzzy "*If...then...*" rule model can be described as follows.

Product-sum-gravity fuzzy reasoning method:

Rule i: If x_1 is A_{1i}, and x_2 is A_{2i} and ... x_m is A_{mi}
then y is B_i (1)

where A_{ji} (j=1,2,...,m; i=1,2,...,n) and B_i are fuzzy subsets of X_j and Y, respectively and the subscript i corresponds to the i-th fuzzy rule.

When an observation $(x_1, x_2, \ldots, x_m)$ is given, a fuzzy inference consequence y can be obtained by using *Product-sum-gravity fuzzy reasoning method* [3] as follows:

$$y = \frac{\sum_{i=1}^{n} h_i s_i y_i}{\sum_{i=1}^{n} h_i s_i} \quad (2)$$

where $h_i = A_{1i}(x_1)A_{2i}(x_2)\ldots A_{mi}(x_m)$ (i=1,2,...,n) is the agreement of the antecedent of the i-th fuzzy rule, s_i is the area of B_i, and y_i is the center of B_i.

Simplified fuzzy reasoning method:

Rule i: If x_1 is A_{1i} and x_2 is A_{2i} and ... x_m is A_{mi}
then y is y_i (3)

where A_{ji} (j=1,2,...,m; i=1,2,...,n) is a fuzzy subset of X_j, and y_i is a real number on Y.

When an observation $(x_1, x_2, \ldots, x_m)$ is given, a fuzzy inference consequence y can be obtained by using *Simplified fuzzy reasoning method* [2] as follows:

$$y = \frac{\sum_{i=1}^{n} h_i y_i}{\sum_{i=1}^{n} h_i} \quad (4)$$

where $h_i = A_{1i}(x_1)A_{2i}(x_2)...A_{mi}(x_m)$ (i=1,2,...,n) is the agreement of the antecedent of the i-th fuzzy rule as in (2), y_i is a real number on Y.

2.2 Conventional Neuro-Fuzzy Learning Algorithms

On the basis of the above two forms of fuzzy inference rules, we shall state two kinds of conventional neuro-fuzzy learning algorithms.

First, two kinds of membership functions are defined as follows:

Gaussian-type membership function:

$$A_{ji}(x_j) = exp(-(x_j-a_{ji})^2/2\sigma_{ji}^2) \quad (5)$$

$$B_i(y) = exp(-(y-y_i)^2/2\sigma_i^2) \quad (6)$$

where a_{ji} and σ_{ji} (j=1,2,...,m; i=1,2,...,n) are the center and width of A_{ji} respectively, as shown in Fig. 1. y_i and σ_i (i=1,2,...,n) are the center and width of B_i, respectively.

Triangular-type membership function:

$$A_{ji} = \begin{cases} 1 - \dfrac{2|x_j - a_{ji}|}{b_{ji}}, & a_{ji} - b_{ji}/2 \le x_j \le a_{ji} + b_{ji}/2; \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where a_{ji} and b_{ji} (j=1,2,...,m; i=1,2,...,n) are the center and width of A_{ji} respectively, as shown in Fig. 2.

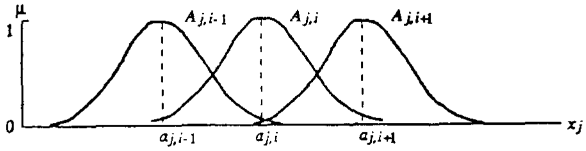

Fig. 1 Gaussian-type membership functions for x_j

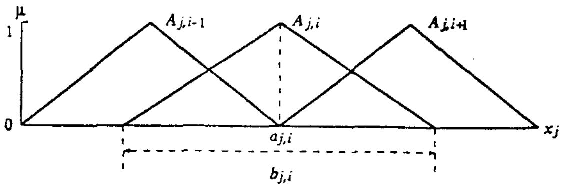

Fig. 2. Triangular-type membership functions for x_j

When the training input-output data $(x_1, x_2, ..., x_m; y^*)$ are given for a fuzzy system model, it is well-known to use the following objective function E for evaluating an error between y^* and y, which can be regarded as an optimum problem:

$$E = (y^* - y)^2 / 2 \quad (8)$$

where y^* is the desired output value, and y is the corresponding fuzzy inference result.

In order to minimize the objective function E, two kinds of neuro-fuzzy learning algorithms for tuning the parameters of the fuzzy rules have been proposed, based on the gradient descent method [5], which are described as follows.

Gaussian-type neuro-fuzzy method:

In (1) the fuzzy subsets A_{ji} (j=1,2,...,m; i=1,2,...,n) on the antecedent parts and B_i on the consequent parts are of Gaussian-type as (5) and (6), respectively. And, in (6) let $\sigma_i = \sigma$ be a constant. Then, (2) can be rewritten as

$$y = \frac{\sum_{i=1}^{n} h_i y_i}{\sum_{i=1}^{n} h_i} \quad (9)$$

which is equivalent to (4), where y_i (i=1,2,...,n) stands for the center of fuzzy subsets B_i.

By using the back-propagation algorithm, Wang and Mendel formulate the following neuro-fuzzy learning algorithm for updating the center a_{ji} and width σ_{ji} of A_{ji} (j=1,2,...,m; i=1,2,...,n), and the center y_i of B_i [7]:

$$a_{ji}(t+1) = a_{ji}(t) - \alpha\, \partial E / \partial a_{ji}(t)$$

$$= a_{ji}(t) - \alpha(y^* - y)\frac{\sum_{i=1}^{n}(y_i - y)h_i(x_j - a_{ji})}{\sigma_{ji}^2 \sum_{i=1}^{n} h_i} \quad (10)$$

$$\sigma_{ji}(t+1) = \sigma_{ji}(t) - \beta\, \partial E / \partial \sigma_{ji}(t)$$

$$= \sigma_{ji}(t) - \beta(y^* - y)\frac{\sum_{i=1}^{n}(y_i - y)h_i(x_j - a_{ji})^2}{\sigma_{ji}^3 \sum_{i=1}^{n} h_i} \quad (11)$$

$$y_i(t+1) = y_i(t) - \gamma\, \partial E / \partial y_i(t)$$

$$= y_i(t) - \gamma(y^* - y)\frac{h_i}{\sum_{i=1}^{n} h_i} \quad (12)$$

where α, β and γ are the learning rates which are regarded as the constants in the learning process, and t means the learning iteration.

Triangular-type neuro-fuzzy method:

In (3) fuzzy subset A_{ji} is defined as the triangular membership function of (7), and y_i is a real number of consequent parts. Then, by (4) Nomura et al. formulate the following neuro-fuzzy learning algorithm based on the back-propagation algorithm for updating the center a_{ji} and width b_{ji} of A_{ji}, and the real number y_i [4]:

$$a_{ji}(t+1) = a_{ji}(t) - \alpha\, \partial E/ \partial a_{ji}(t)$$

$$= a_{ji}(t) - 2\alpha(y^* - y)(y_i - y)\frac{\operatorname{sgn}(x_j - a_{ji})\prod_{k \neq j}^{n} A_{ki}(x_k)}{b_{ji}\sum_{i=1}^{n} h_i} \quad (13)$$

$$b_{ji}(t+1) = b_{ji}(t) - \beta\, \partial E/ \partial b_{ji}(t)$$

$$= b_{ji}(t) - 2\beta(y^* - y)(y_i - y)\frac{|x_j - a_{ji}|\prod_{k \neq j}^{n} A_{ki}(x_k)}{b_{ji}^2\sum_{i=1}^{n} h_i} \quad (14)$$

$$y_i(t+1) = y_i(t) - \gamma\, \partial E/ \partial y_i(t)$$

$$= y_i(t) - \gamma(y^* - y)\frac{h_i}{\sum_{i=1}^{n} h_i} \quad (15)$$

where in (13) and (14), $k \neq j$ implies $k = 1,\ldots,j-1,j+1,\ldots,m$.

For the above two kinds of neuro-fuzzy approaches, there exists a common construction of neural networks of the fuzzy system model as shown in Fig. 3.

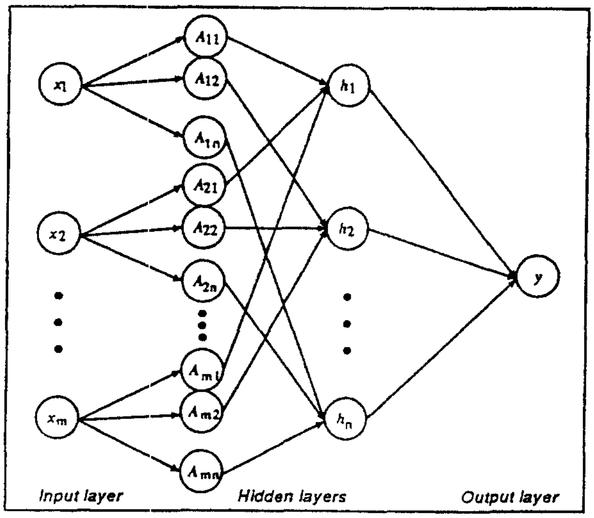

Fig. 3 Neural networks of the fuzzy system model

3. Some of properties on conventional neuro-fuzzy learning algorithms

We now turn to the discussion of some of common characters on the conventional neuro-fuzzy learning algorithms.

Property 1: *Membership functions on antecedent parts are independent each other.*

For the input variable x_j ($j = 1,2,\ldots,m$), all of the membership functions $A_{j1}, A_{j2}, \ldots, A_{jn}$ in (1) and in (3) are independent each other, that is, for $i \neq k$ ($i,k \in \{1,\ldots,n\}$) the following inequality holds, in general [6]:

$$A_{ji} \neq A_{jk} \quad (16)$$

Property 1 implies that the linguistic variable A_{ji} ($j=1,2,\ldots,m$; $i=1,2,\ldots,n$) is strongly restricted and used only one time corresponding to the i-th fuzzy rule under the conventional neuro-fuzzy learning algorithms.

Property 2: *Fitting to training data is fast and finer.*

For given suitable training data, the fitting to the data is fast by the neuro-fuzzy learning algorithms because of the independence of individual membership function by Property 1, if the number of training data is not too much over the number of fuzzy rules. This implies that neuro-fuzzy learning algorithms have high generalization capability.

Note that Property 2 will become difficult if the number of the training data is much bigger than the number of the fuzzy rules, since each membership function is related with a number of training data, so the freedom of individual membership function is strongly restricted.

Property 3: *Formulations are simple.*

From algorithms (10) - (15) we can see that the formulations of the neuro-fuzzy learning algorithms are not so complex for performing the computation.

Property 4: *The number of membership functions on individual input space is the same, which is also equal to the number of the fuzzy rules.*

The partition number by membership functions on each input space must be the same. According to the arrangements (1), (3) and the independence of membership functions by Property 1, the number of membership functions for individual input linguistic variable is the same, which is also equal to the number of the fuzzy rules. This implies that the restricting to the number of fuzzy partitions is too strong under the conventional neuro-fuzzy learning algorithms.

As is well known in fuzzy applications, it is not necessary to restrict all of the partition numbers for input variables to be the same. When we deal with a multiple-inputs system, it is reasonable and suitable to partition individual input space by the appropriate number of membership functions according to the interrelation between the input variable and the output variable.

Property 5: *The number of tuning parameters for a multiple-input system is large.*

To deal with a complex and multiple-input system, the use of many of fuzzy inference rules for constructing the corresponding fuzzy system model is necessary and reasonable. By Property 4 the number of the membership functions for each input variable is equal to the number of fuzzy inference rules, which leads to the great number of tuning parameters when the number of input variables increases. That is, the number of the membership functions for each input variable increases rapidly, as the number of fuzzy inference rules increases.

Property 6: *Representation of the form of fuzzy rule table is impossible.*

In the neuro-fuzzy learning algorithms, the expression of fuzzy inference rules in the form of fuzzy rule table is hard or impossible because of the independence of membership functions. For example, in the case of triangular-type membership functions, if we take $m = 2$ and $n = 9$ in (3), then as seen in Fig. 4, the fuzzy rule table generated by (13) - (15) is different from the usual fuzzy rule table used widely in fuzzy control applications.

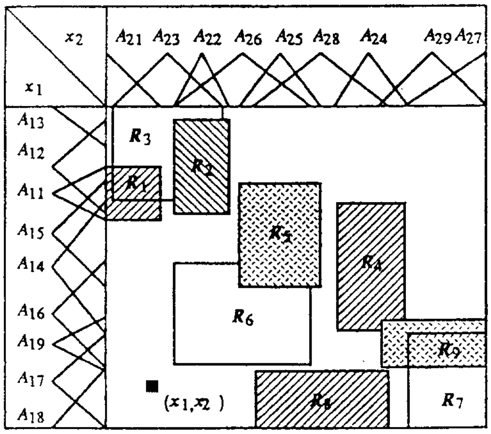

Fig. 4 Fuzzy rule table under triangular-type membership functions

Property 7 : *Non-firing state or weak-firing state exist.*

As discussed in Property 2, the fitting to training data is fast by the neuro-fuzzy learning algorithms because of the freedom of individual membership function. However, it is likely to occur a weak-firing case under Gaussian-type membership functions or non-firing case under triangular-type membership functions after the learning because of the independence of membership functions. For the given training data, it is possible that all of membership functions are changed into another forms from their initial states as shown in Fig. 4 under the neuro-fuzzy learning algorithms. In this case, there exist a non-firing state in the blank place as in Fig. 4, so the fuzzy inference result must be effected. In other words, one may consider that the approximation of the system model is not good.

4. Conclusions

We have analyzed some of basic properties on two kinds of neuro-fuzzy learning algorithms, and illustrated some of advantages and shortages in these approaches. As was discussed, it is sometimes limited to apply the neuro-fuzzy learning algorithms to the construction of a multiple-input or a large scale fuzzy system model, in general. Therefore, it is necessary and important to develop a new approach of neuro-fuzzy learning algorithm to cope with the above problems.

References

1 H. Ichihashi, "Iterative fuzzy modeling and a hierarchical network", Proc. of the 4th IFSA Congress, 49-52, Brussels, 1991.

2 M. Maeda and S. Murakami, "An automobile tracking control with a fuzzy logic", Proc. of the 3rd Fuzzy System Symposium, 61-66, Osaka, 1987 (in Japanese).

3 M. Mizumoto, "Fuzzy controls by product-sum-gravity method", in Advancement of Fuzzy Theory and Systems in China and Japan (eds. X. H. Liu and M. Mizumoto), Int. Academic Publishers, c1.1-c1.4, 1990.

4 H. Nomura, I. Hayashi and N. Wakami, "A self-tuning method of fuzzy control by descent method", Proc. of the IEEE International Conference on Fuzzy Systems, 203-210, San Diego, 1992.

5 D.E. Rumelhart, J.L. McClelland and the PDP Research Group, Parallel Distributed Processing, MA: MIT Press, Cambridge, 1986.

6 Y. Shi, M. Mizumoto, N. Yubazaki and M. Otani, "A learning algorithm for tuning fuzzy rules based on the gradient descent method", Proc. of the 5th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96), 55-61, New Orleans, 1996.

7 L.X. Wang and J.M. Mendel, "Back-Propagation fuzzy system as nonlinear dynamic system identifiers", Proc. of the IEEE International Conference on Fuzzy Systems, 1409-1416, San Diego, 1992.

8 R.R. Yager and D.P. Filev, Essentials of Fuzzy Modeling and Control, John Wiley & Sons, 1994.

9 R.R. Yager and D.P. Filev, "Generation of fuzzy rules by mountain clustering", Journal of Intelligent & Fuzzy Systems, 2(3), 209-219, 1994.