IEEE Int. Conf. Neural Networks & Signal Processing
Nanjing, China, December 14-17, 2003

GENERALIZED ADDITIVE-MULTIPLICATIVE FUZZY NEURAL NETWORK OPTIMAL PARAMETERS IDENTIFICATION BASED ON GENETIC ALGORITHM

Dong-hai ZHAI, LI Li and Fan JIN
School of Computer and Communication Engineering, Southwest Jiaotong University
Chengdu, Sichuan, 610031, P.R.China

ABSTRACT

In Additive-Multiplicative fuzzy neural networks (AMFNN), its membership functions have no adaptability and the number of fuzzy rules is determined subjectively. In this paper, a generalized Additive-Multiplicative fuzzy neural network (generalized AMFNN) is presented, and the parameters of membership functions can be adjusted. Therefore, there are many parameters to be determined. The matrix coding in genetic algorithm (GA), which combines binary coding with real number coding, is adopted to search the optimal parameters of the generalized AMFNN and determine the number of fuzzy rules. The generalized AMFNN has lower complexity and can approximate to a nonlinear system at high accuracy degree. A numerical simulation has demonstrated the validity of this approach.

1. INTRODUCTION

In reference [1], the author of this paper proposed a new fuzzy neural network ——Additive-Multiplicative Fuzzy Neural Network (AMFNN), which is an integration of additive inference and multiplicative inference. Theoretical analysis and instances verification show that AMFNN has such characteristics as high reasoning precision, wide application scope, strong generalization capability and easy implementation. Furthermore, the learning algorithm of AMFNN is very simple and can be implemented easily. However, similar to Takagi-sugeno fuzzy logic system, AMFNN has some practical difficulties shown as follows:

1) How to determine the membership function $\mu_{F_i^j}(x_i)$ (i=1, 2, … n; j=1, 2, …., m)? Does it have adaptive capability?
2) How to determine the optimal number m of fuzzy rules, which relates to the complexity of AMFNN model.

The determination of the optimal number of m fuzzy rules and membership function will directly affect whether AMFNN can be more adaptive to match real model, and much lower model complexity. Li He-sheng et al. has studied generalized Takagi-Sugeno logical system optimal parameter identification based on genetic algorithm [2]. Here, a generalized AMFNN is proposed, and the membership function of which is an adaptive generalized Gaussian function (See definition 1). In model implementation, a system optimal parameter identification method based on genetic algorithm is adopted in this paper. Global optimization and population searching strategy of GA is particularly suitable to solve complex nonlinear optimization problem which is difficult to be solved with traditional optimization method [3, 4].

2. GENERALIZED ADDITIVE-MULTIPLICATIVE FUZZY NEURAL NETWORKS

In engineering applications of fuzzy control, such functions as triangle, trapezium, Gaussian or other exponent functions are used to be membership function. The disadvantage of these membership functions is that as long as the membership function type is determined, its rough shade can't be modified. If we can find a parametric membership function, which can adaptively approximate toall functions above by changing parameters, it will have significance for the fuzzy control system. Therefore, the following are this adaptive function and corresponding fuzzy system.

Definition 1 [5]: Function $\mu(x)$ can be called generalized Gaussian function, if it has following expression:

$$\mu(x) = \exp\left(-\left|\frac{x-\beta}{\alpha}\right|^{\gamma}\right),\ \alpha > 0,\ \beta \in R,\ \gamma \geq 0. \quad (1)$$

If α=1, β=0 and appropriate value for γ, a set of generalized Gaussian membership function $\mu(x) = \exp(-|x|^{\gamma})$ can respectively approximate triangle, trapezium and Gaussian function. If α, β are changing more, we can translate, compress, expand this set of membership function to approximate triangle, trapezium and Gaussian function better. Therefor, generalized Gaussian membership function has adaptive capability.

Definition 2: A fuzzy logic system can be called generalized AMFNN, if the membership function of AMFNN is generalized Gaussian function, i.e.

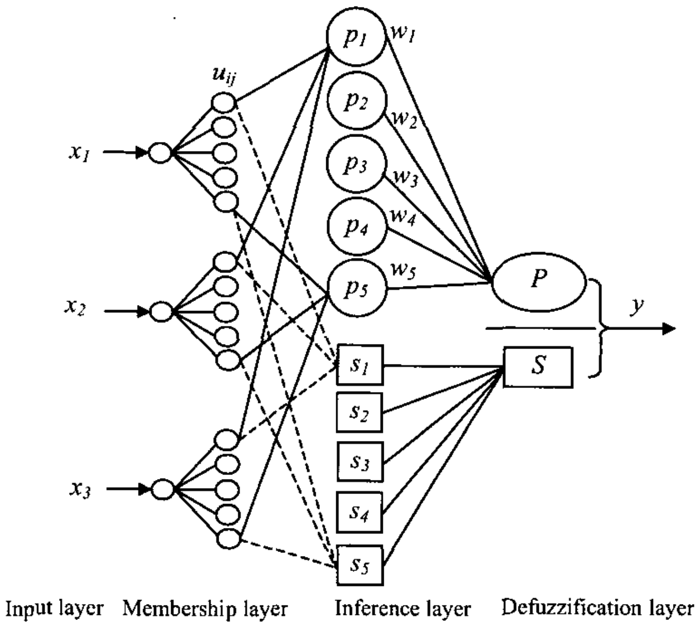

Fig.1 Architecture of AMFNN

$$\mu_{ij} = \mu_{F_i^j}(x_i) = \exp\left(-\left|\frac{x_i - \beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right) \quad (2)$$

where $\alpha_i^j > 0$, $\beta_i^j \in R$, $\gamma_i^j \geq 0$ (i=1, 2, ... n;j=1, 2, ..., m).

The generalized AMFNN with n inputs, single output and m fuzzy rules is shown as Fig.1.

The membership function is Equation (2), where α_i^j, β_i^j are parameters corresponding to each node u_{ij} at the membership generation layer. In this layer, from the top down, u_{ij} can be specifically expressed as: u_{11}, u_{12}, ..., u_{1m}; u_{21}, u_{22}, ... , u_{2m}; ... ; u_{n1}, u_{n2}, ... , u_{nm}, where n is the number of input variables, and m is the number of rules. Subscript and superscript of each α_i^j, β_i^j is similar to u_{ij}.

Multiplicative inference is paralleled to additive inference in the inference layer. The nodes of inference layer are divided into two kinds: one used in multiplicative inference, and the other used in additive inference.

The output of multiplicative inference node is an algebraic product of all its inputs:

$$p_j = u_{1j} \cdot u_{2j} \cdots\cdots u_{nj} = \prod_{i=1}^{n} u_{ij} = \prod_{i=1}^{n} \exp\left(-\left|\frac{x_i - \beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right), \quad (1 \leq j \leq m). \quad (3)$$

The output of additive inference node is an algebraic sum of all its inputs:

$$s_j = u_{1j} + u_{2j} + \cdots\cdots + u_{nj} = \sum_{i=1}^{n} u_{ij} = \sum_{i=1}^{n} \exp\left(-\left|\frac{x_i - \beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right), \quad (1 \leq j \leq m). \quad (4)$$

The final output of defuzzification layer is a ratio, the numerator of which is the weighted algebraic sum of every rule's output of multiplicative inference, and the denominator is the algebraic sum of every rule's output of additive inference.

Numerator:

$$P = w_1 p_1 + w_2 p_2 + \cdots\cdots + w_m p_m = \sum_{j=1}^{m} w_j p_j = \sum_{j=1}^{m} w_j \prod_{i=1}^{n} \exp\left(-\left|\frac{x_i - \beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right). \quad (5)$$

Denominator:

$$S = s_1 + s_2 + \cdots\cdots + s_m = \sum_{j=1}^{m} s_j = \sum_{j=1}^{m} \sum_{i=1}^{n} \exp\left(-\left|\frac{x_i - \beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right). \quad (6)$$

Ratio:

$$y=\frac{P}{S}=\frac{\sum_{j=1}^{m} w_j \prod_{i=1}^{n} \exp\left(-\left|\frac{x_i-\beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right)}{\sum_{j=1}^{m}\sum_{i=1}^{n} \exp\left(-\left|\frac{x_i-\beta_i^j}{\alpha_i^j}\right|^{\gamma_i^j}\right)} \quad (7)$$

The sufficient and necessary condition of modeling by generalized AMFNN is that this system can approximate real model at arbitrary accuracy degree, i.e. it has universal approximation. Based on Stone-Weirstrass theorem [6], the universal approximation of generalized AMFNN can be proved. The following is the theorem and its proof is omitted for brevity.

Theorem 1: Generalized AMFNN is universal approximator.

3. GENERALIZED ADDITIVE-MULTIPLICATIVE FUZZY NEURAL NETWORK OPTIMAL PARAMETER IDENTIFICATION BASED ON GENETIC ALGORITHM

The operation process of searching optimal parameters of generalized AMFNN by genetic algorithm is as follows.

1) *Encoding*. According to Equation (7), this model has $3n\times m$ variables to be determined. Especially, when n and m is large number, the number of independent variables will be very large. If we adopt binary coding, the entire code of a chromosome is so long that the searching time of GA becomes too long. In this paper, generalized AMFNN adopt matrix coding, in which, binary coding and real number coding coexist. Matrix coding of individual of generalized AMFNN is shown in Fig.2. Here, code of first rule is on the first line, and code of second rule is on the second line, ..., code of mth rule is on the mth line. Switch parameter δ_i (i=1, 2, ..., m) is one digital binary coding, where δ_i=1 means the ith rule exists, and δ_i=0 means the ith rule does not exist. Expect for δ_i (i=1, 2, ..., m), the other parameters are real number code.

$$\begin{bmatrix} \delta_1 & \alpha_1^1 & \beta_1^1 & \gamma_1^1 & \cdots & \alpha_n^1 & \beta_n^1 & \gamma_n^1 \\ \delta_2 & \alpha_1^2 & \beta_1^2 & \gamma_1^2 & \cdots & \alpha_n^2 & \beta_n^2 & \gamma_n^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \delta_m & \alpha_1^m & \beta_1^m & \gamma_1^m & \cdots & \alpha_n^m & \beta_n^m & \gamma_n^m \end{bmatrix}$$

Fig.2 the matrix coding of generalized AMFNN

2) *Population initialization*. The number T of individuals in the population is an important design value in the genetic algorithm, which can directly influence the final result and execution efficiency of genetic algorithm, but there is no widely accepted criterion to decide it. Here, let T=40, and we generate initial population by recursive cycle algorithm at random.

3) *Model evaluation*. The model can be evaluated by model accuracy and model complexity. The model accuracy is made by accumulation square error e. When e is smaller, the model accuracy is higher. Model complexity is reflected by the actual number of rules M_{TS}. When M_{TS} is smaller, model complexity is lower. The fitness value of generalized AMFNN chromosome is denoted as

$$g(t)=w_e\times\frac{1}{e}+w_M\times\frac{1}{M_{TS}} \quad (8)$$

where w_e, w_M is weight value, and the real number of rules is $M_{TS}=\sum_i \delta_i$.

4) *Reproduction*. Among various reproduction methods, we adopt the fitness-proportionate selection or the 'roulette wheel' selection to reproduce the new individuals. When p_s=1, the occurrence probability of an individual t is

$$p(t)=g(t)/\sum_t g(t) \quad (9)$$

5) *Crossover*. The crossover operation may be applied to all pairs of parents, or it may be applied only to some selected pairs. In the latter case, the selection is determined by the crossover probability ρ_c. Crossover provides a mechanism for chromosomes to mix and match attributes through random processes. First, pairs of reproduced chromosomes (referred as parents) from the current population are selection at random. Second, two minors of matrices, which are at the same position in these two individuals of parents, are selected at random. Third, the two minors of matrices are swapped between the two chromosomes in each pair, resulting in two new offsprings. The operation process is shown in Fig.3.

6) *Mutation*. Mutation is an operation on a gene. Each gene is assigned a probability p_m for mutation. When a gene is stochastically chosen for the mutation, a new gene generated at random replaces it. In this paper, adaptive multiple mutation is adopted. The range of mutation factor a_m is 0.5~1.0 obtained by Monte Carlo experiment. The mutation method is shown as follows.

I) Binary coding δ_i (i=1,2, ..., m): When it satisfies mutation probability p_m, it will mutate as follow.

$$1\to 0 \text{ and } 0\to 1 \quad (10)$$

II)Real number coding v_k: When it satisfies mutation probability p_m, it will mutate as follow.

$$\tilde{v}_k=v_k+(p_k-0.5)E(a_{\max}^t)a_m \quad (11)$$

Random number is $p_k\in[0,1]$ and a_m is mutation factor. $v_k$is real number coding in chromosome, i.e. $\alpha_i^j,\beta_i^j,\gamma_i^j$. $\tilde{v}_k$ is mutation result of v_k. $E(a_{\max}^t)$ is the accumulation

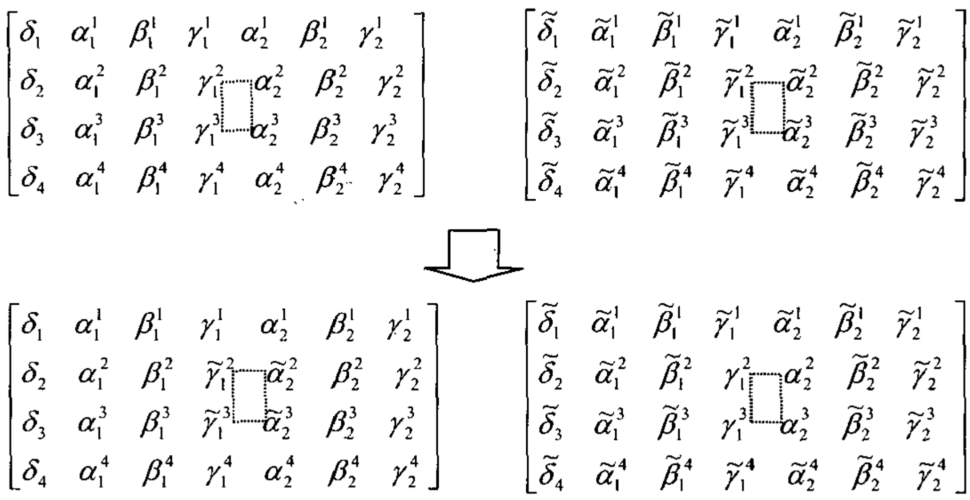

Fig.3 Crossover operation of generalized AMFNN with n=2, m=4

square error of best individual in tth generation, which has the best fitness.

7) *End condition.* The iteration halts when the iteration exceeds the fixed times or the best individual has error smaller than the threshold.

8) *System parameter determination.* Every coding value of the best individual is corresponding optimal parameter of generalized AMFNN. Switch parameter δ_i =1 means the ith rule is exist, and δ_i =0 means the ith rule is not exist. $M_{TS}=\sum_i \delta_i$ is the optimal number of rules in generalized AMFNN.

4. NUMERICAL EXAMPLES

In this section we illustrate a numerical example from [3]. The following nonlinear static system with two inputs x_1 and x_2, and a scalar output y are used:

$$y = x_1 e^{-(x_1^2+x_2^2)}, -2 \le x_1, x_2 \le 2 \qquad (12)$$

The data were generated by using independent uniformly distributed random numbers $\{(x_1(k), x_2(k)), k=1, 2, \cdots\cdots, 100\}$ in the domain $-2 \le x_1, x_2 \le 2$. The following are the parameters used in this experiment: population size T=40, selection probability ρ_s=1, crossover probability p_c=0.77, mutation probability p_m=0.01, mutation factor a_m=0.65, weight of fitness value w_e=1 and w_M=10.

The procedure of the genetic algorithm was repeated 200 times in this example. At each iteration, the best individual in the population was recorded. Not every iteration produced a better individual. Fig.4 shows the criterion g of the best individuals at each generation. There were not many differences among the best individuals after 155 iterations. Consequently the best one at the 155th iteration was regarded as the optimal solution, i.e. g_{max}=g(155)=39.9928, where

$$\max_{1\le k\le 100}\{|\, y^*(k) - y(k)\,|\} = 0.030\,31. \qquad (13)$$

Fig.5 shows the estimation error between Equation (9) and its estimation by generalized AMFNN. The original shape and approximation shape produced by generalized AMFNN are shown in Fig.6, Fig.7. To this example, the comparison with Tanaka method in [3], Li He-sheng method in [2] is show in Table 1. The result shows that this generalized AMFNN is very effective.

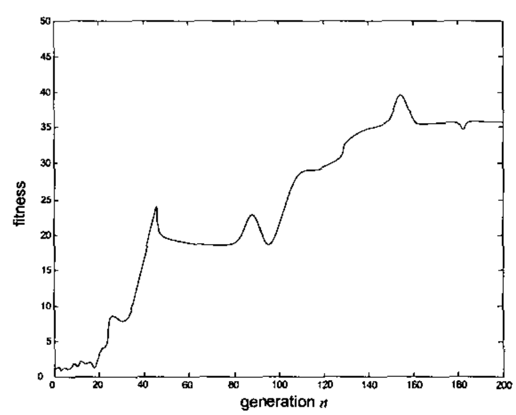

Fig.4 Fitness of best individual at each generation

Tab.1 The comparison result with reference [3], [2]

Property / Method	Partition of input space	Fuzzy domain (membership function)	GA coding	Number of rules	Accuracy (maximum error)
Tanaka method in [3]	Partition according to a certain variable	Non-adaptive Trapezium function	S-expression of LISP	22	0.090 3
Li He-sheng method in [2]	Adaptive partition according to different variables	Adaptive generalized Gaussian function	Matrix coding (binary coding combined with real number coding)	4	0.040 23
Generalized AMFNN	Adaptive partition according to different variables	Adaptive generalized Gaussian function	Matrix coding (binary coding combined with real number coding)	3	0.030 31

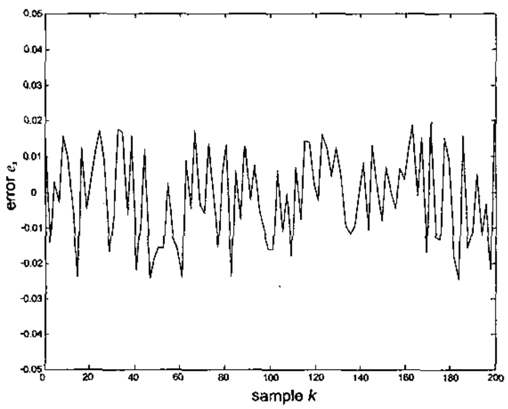

Fig.5 Estimation error

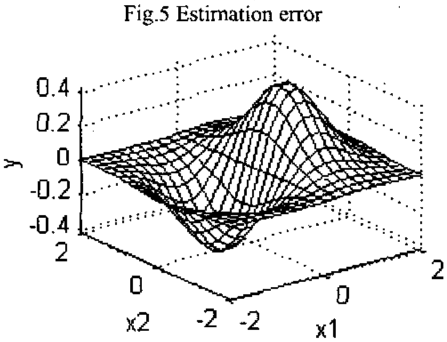

Fig.6 original shape

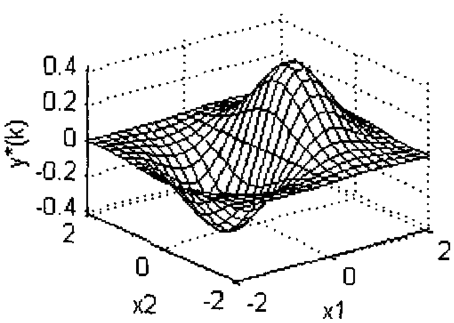

Fig.7 Approximation shape by generalized AMFNN

5. CONCLUSIONS

A generalized AMFNN is presented, and the parameters of whose membership functions can be adjusted. In real modeling, this method can approximate to practical model very well. An optimal parameter identification method based on genetic algorithm is proposed simultaneously. The simulation result show that generalized AMFNN can evaluate by GA. In some generation, we can get a set of suboptimal parameters of generalized AMFNN. This suboptimal generalized AMFNN can approximate to a high nonlinear system at high accuracy, but with has low complexity, which offers some practical value in engineering application.

6. REFERENCES

[1] Zhai Dong-hai, "A study of the realization of fuzzy inference by way of neural networks" [Master Thesis][D], Southwest Jiaotong University, pp.19-39, 2000.

[2] LI He-Sheng, Mao Jian-Qi, Dai Ji-Yang, "Generalized Takagi-Sugeno logical system optimal parameter identification based on genetic algorithm", *ACTA Automatica SINICA*, pp.581-586, 28(4), 2002.

[3] Masahiro Tanaka, Ju Ye, Tetsuzo Tanino, "Fuzzy modeling by genetic algorithm with tree-structured individuals", *International Journal of Systems Science*, pp.261-268, 27(2), 1996.

[4] Daihee Park Abraham, Kandel & Gideon Langhola, "Genetic-based new fuzzy reasoning models application to fuzzy control", *IEEE Trans. on Systems, man and Cybernetics*, pp.39-47, 24(1), 1994.

[5] Wang Shitong, Fuzzy system, fuzzy neural network and their application program designing, Shanghai literature publication house of science and technology, Shanghai, 1998.

[6] Rudin W., Principle of mathematical Analysis, mcGraw-Hill. Inc., New York,1976.

[7] Chen Guoliang, Wang Xufa, Zhuang Zhenquan, Wang Dongsheng, Genetic algorithm and its applications, Pople's postal publication house, Beijing, 1996.