3

An AND-OR Fuzzy Neural Network

Jianghua Sui
Mechanical Engineering College, Dalian Ocean University, Dalian 116023, P. R. China

1. Introduction

Fuzzy neural network combines the theories of fuzzy logical and neural network, including learning, association, identification, self-adaptation and fuzzy information process. The logic neurons have received much concern all the time as the important components of neural networks. From the models designing to the algorithms studying, there are many achievements. Glorennec[1]proposed a general artificial neuron to realize Lukasiewicz logical operate. Yager[2] employed a group of OWA fuzzy Aggregation operators to form OWA neuron. Pedrycz and Rocha [3] proposed aggregation neurons and referential neurons by integrating fuzzy logic and neural network and discuss the relation about the ultimate network structure and practical problem; Pedrycz i.e. [4],[5],[6] constructed a knowledge-based network by AND, OR neurons to solve classified problem and pattern recognition. Bailey i.e. [7] extended the single hidden layer to two hidden layers for improve complex modeling problems. Pedrycz and Reformat designed fuzzy neural network constructed by AND, OR neurons to modeling the house price in Boston [8].
We consider this multi-input-single-output (MISO) fuzzy logic-driven control system based on Pedrycz. Pedrycz[8] transformed T norm and S norm into product and probability operators, formed a continuous and smooth function to be optimized by GA and BP. But there is no exactly symbolic expression for every node, because of the uncertain structure. In this paper, the AND-OR FNN is firstly named as AND-OR fuzzy neural network, The in-degree and out-degree for neuron and the connectivity for layer are defined in order to educe the symbolic expression of every layer directly employing Zadeh operators, formed a continuous and rough function. The equivalence is proved between the architecture of AND-OR FNN and the fuzzy weighted Mamdani inference in order to utilize the AND-OR FNN to auto-extract fuzzy rules. The piecewise optimization of AND-OR FNN consists two phases, first the blueprint of network is reduced by GA and PA; the second phase, the parameters are refined by ACS (Ant Colony System). Finally this approach is applied to design AND-OR FNN ship controller. Simulating results show the performance is much better than ordinary fuzzy controller.

2. Fuzzy neurons and topology of AND-OR FNN

2.1 Logic-driven AND, OR fuzzy neurons

The AND and OR fuzzy neurons were two fundamental classes of logic-driven fuzzy neurons. The basic formulas governed the functioning of these elements are constructed

with the aid of T norm and S norm (see Fig. 1, Fig. 2).Some definitions of the double fuzzy neurons show their inherent capability of reducing the input space.

Definition 1. Let $X=[x_1,x_2.....x_n]$ be input variables $W=[w_1,w_2,.....w_n]$ be adjustable connections (weights) confined to the unit interval. Then,

$$y=\underset{i=1}{\overset{n}{T}}(w_i S x_i) \tag{1}$$

is the AND fuzzy neuron, which completes a T norm and S norm composition operators like Fig.1.

Definition 2. Let $X=[x_1,x_2.....x_n]$ be input variables, $W=[w_1,w_2,.....w_n]$ be adjustable connections (weights) confined to the unit interval. Then,

$$y=\underset{i=1}{\overset{n}{S}}(w_i T x_i) \tag{2}$$

is the OR fuzzy neuron, which completes an S norm and T norm composition operators like Fig.2.

AND and OR neurons both are the mapping $[0,1]^n \to [0,1]$,the neuron expression is shown by Zadeh operators as Eq.(3)(4).

$$y=\underset{i=1}{\overset{n}{\wedge}}(w_i \vee x_i) \tag{3}$$

$$y=\underset{i=1}{\overset{n}{\vee}}(w_i \wedge x_i) \tag{4}$$

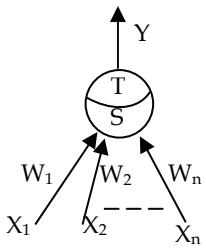


Fig. 1. AND neuron

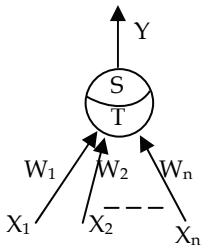


Fig. 2. OR neuron

Owing to the special compositions of neurons, for binary inputs and connections the neurons functional is equal to the standard gates in digital circuit shown in Table 1, 2. For AND neuron, the closer to 0 the connection w_i, the more important to the output the corresponding input x_i is. For OR neuron, the closer to 0 connection w_i, the more important to the output the corresponding input x_i is. Thus the values of connections become the criterion to eliminate the irrelevant input variables to reduce input space.

x_1	x_2	w_1	w_2	y
0	0	0	0	0
0	1			0
1	0			0
1	1			1
0	0	0	1	x_1
0	1			
1	0			
1	1			
0	0	1	0	x_2
0	1			
1	0			
1	1			
0	0	1	1	1
0	1			1
1	0			1
1	1			1

Table 1. AND neuron and AND gate

x_1	x_2	w_1	w_2	y
0	0	0	0	0
0	1			0
1	0			0
1	1			0
0	0	0	1	x_2
0	1			
1	0			
1	1			
0	0	1	0	x_1
0	1			
1	0			
1	1			
0	0	1	1	0
0	1			1
1	0			1
1	1			1

Table 2. OR neuron and OR gate

2.1 Several notations about AND,OR neuron

Definition 3. Let w_i be the connection value, x_i be the ith input variable. Then,

$$RD(x_i) = w_i \in [0,1] \tag{5}$$

is the relevant degree between the input x_i and the neuron's output. For AND neuron, if $RD(x_i) = 0$, then x_i is more important feature to the output; if $RD(x_i) = 1$, then x_i is more irrelevant feature to the output, it can be cancelled. For OR neuron, vice versa. Thus the RDV or RDM is the vector or matrix made up of connections, respectively , which becomes the threshold to obstacle some irrelevant input variables, also lead to reduce the input space.

Definition 4. In-degree is from directed graph in graph theory; the neural network is one of directed graph. The in-degree of the ith neuron $d^+(neuron_i)$ is the number of input variables, then the in-degree of the ith AND neuron $d^+(AND_i)$ is shown the number of his input variables; the in-degree of the jth OR neuron $d^+(OR_j)$ is shown the number of his input variables.

Definition 5. The out-degree of neuron $d^-(neuron_i)$ is the number of output variables, then the out-degree of the ith AND neuron $d^-(AND_i)$ is the number of his output variable; the out-degree of the jth OR neuron $d^-(OR_j)$ is shown the number of his output variables.

2.2 The architecture of AND-OR FNN

This feed-forward AND-OR FNN consists of five layers (Fig.3. MISO case), i.e. the input layer, fuzzification layer, double hidden layers (one consists of AND neurons, the other consists of OR neurons) and the defuzzification output layer, which corresponding to the four parts (fuzzy generator, fuzzy inference, rule base and fuzzy eliminator) of FLS (Fuzzy Logic System) respectively. Here the fuzzy inference and the fuzzy rule base are integrated into the double hidden layers. The inference mechanism behaves as the inference function of the hidden neurons. Thus the rule base can be auto-generated by training AND-OR FNN in virtue of input-output data.

Both W and V are connection matrixes, also imply relevant degree matrix (RDM) like introduce above. Vector H is the membership of consequents. The number of neurons in every layer is n, $n \times t$, m, s and 1 respectively (t is the number of fuzzy partitions.).

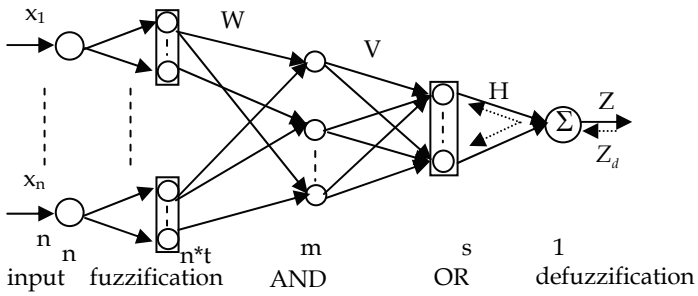


Fig. 3. The architecture of AND-OR FNN

Definition 6. The layer connectivity is the maximum in-degree of every neuron in this layer, including double hidden layer,

$$\begin{aligned} Con(AND) &= \max(d^+(AND_i)) \\ Con(OR) &= \max(d^+(OR_j)) \end{aligned} \tag{6}$$

where AND_i is the ith AND neuron, $d^+(AND_i)$ be the in-degree of the ith neuron. OR_j is the ith OR neuron, $d^+(OR_j)$ be the in-degree of the ith neuron.
Note: $Con(AND) \le n$ (n is the number of input variables). $Con(OR) \le m$ (m is the number of AND neurons).

2.3 The exact expression of every node in AND-OR FNN

According to the above definitions and the physical structure background, the AND-OR FNN model is derived as Fig. 3. The functions of the nodes in each layer are described as followers.
Layer 1 (input layer): For every node i in this layer, the input and the output are related by

7)

where O_i^1 denotes the output of ith neuron in layer 1, $i = 1,2,\cdots,n$. Here the signal only transfers to the next layer without processing.
Layer 2 (duzzification layer): In this layer, each neuron represents the membership function of linguistic variable. The most commonly used membership functions are shape of Bell and Gaussian. In this paper, Gaussian is adopted as membership function. The linguistic value (small ,$\cdots$, very large) A_j are used. The function is shown as.

$$O_{ij}^2 = \mu_{A_j}(x_i) = e^{\frac{-(x_i - m_{ij})^2}{\sigma^2}} \tag{8}$$

where $i = 1,2,...n, j = 1,2,...t$, m_{ij} and σ is the modal and spread of the jth fuzzy partition from the ith input variable.
Layer 3 (AND hidden layer): This layer is composed of AND neurons, Based on above definitions, the function is like.

$$\begin{aligned} O_k^3 &= \mathop{T}_{p_1=1}^{d^+(AND_k)}(wSO^2) \\ &= \mathop{\wedge}_{p_1=1}^{d^+(AND_k)}(w_{rk} \vee \mu_{A_{ij}}(x_i)) \end{aligned} \tag{9}$$

where $k = 1,2,\cdots,t^n \;\; j = 1,2,\cdots,t \;\; i = 1,2,\cdots,n \;\; r = (i-1)\times t + j$, $d^+(AND_k)$ is the in-degree of the kth AND neuron. t^n is the total of AND neurons in this layer.
Note: when p_1 is fixed and $d^+(AND_i) \ge 2$, q must be different. That means the input of the same AND neuron O^2 must be from the different x_i
Layer 4 (OR hidden layer): This layer is composed of OR neurons, Based on above definitions, the function is like.

$$\begin{aligned} O_l^4 &= \mathop{S}_{p_2=1}^{d^+(OR_l)}(vTO^3) \\ &= \mathop{\vee}_{p_2=1}^{d^+(OR_i)}(v_{l,k} \wedge O_k^3) \end{aligned} \tag{10}$$

where $k = 1,2,\cdots,t^n$, $l = 1,2,\cdots,s$ $d^+(OR_l)$ is the in-degree of the lth OR neuron. s is the total of OR neuron.

Layer 5 (defuzzification layer): There is only one node in this layer, but includes forward compute and backward training. Center-of-gravity method is adopted for former compute as follows.

$$z = O^5 = \frac{\sum O_l^4 h_{l,1}}{\sum O_l^4} \tag{11}$$

where $l = 1,2,\cdots,s$, H is the membership function of consequents. The latter is only imported to train data for optimization next. There is no conflict because the double directions are time-sharing. The function is like eq.(8)

$$O^5 = \mu_{B_l}(z_d) = e^{\frac{-(z_d - m_j)^2}{\sigma^2}} \tag{12}$$

3. Functional equivalent between fuzzy weighted Mamdani Inference and AND-OR FNN

This section demonstrates the functional equivalence between AND-OR FNN and fuzzy weighted Mamdani inference system, though these two models are motivated from different origins (AND-OR FNN is from physiology and fuzzy inference systems from cognitive science), thus auto-extracting the rule base by training AND-OR FNN. The functional equivalent under minor restrictions is illustrated

3.1 Fuzzy weighted Mamdani inference

The fuzzy weighted Mamdani inference system [9] utilizes local weight and global weight to avoid a serious shortcoming in that all propositions in the antecedent part are assumed to have equal importance, and that a number of rules executed in an inference path leading to a specified goal or the same rule employed in various inference paths leading to distinct final goals may have relative degrees of importance. Assume the number of the fuzzy IF-THEN rules with consequent y is B_l, then fuzzy rules can be represented as:

$$\begin{aligned} Rule: &if\ x_1 \text{ is } A_{1j} \text{and} \\ &\vdots \\ &\text{and } x_i \ is\ A_{ij} \text{ and} \\ &\vdots \\ &\text{and } x_n \ is\ A_{nj} \text{ then } y \ is\ B_l \end{aligned} \tag{13}$$

where $x_1,\cdots,x_n$ are the input variables, y is the output, A_{ij} and B_i are the fuzzy sets of input and output, respectively. w_{ij} is local weights of the antecedent part; $v_1,\cdots,v_s$ is the

class of global weight for every rules. $i=1,2,\cdots,n$, $j=1,2,\cdots,t$, $l=1,2,\cdots,s$, t is the total of antecedent fuzzy sets; s is the total of consequent fuzzy sets. The same B_i is collected to form s complex rules as follows:

$$\begin{aligned}
&Rule1: if\ x_1 \text{ is } A_{1i'} \text{and} \cdots \\
&\quad \text{and } x_n \text{ is } A_{ni''} \text{ then } y \text{ is } B_1 \\
&\quad or \cdots or \\
&\quad if\ x_1 \text{ is } A_{1j'} \text{ and} \cdots \\
&\quad \text{and } x_n \text{ is } A_{nj''} \text{ then } y \text{ is } B_1 \\
&\quad \vdots \\
&RuleS: if\ x_1 \text{ is } A_{1k'} \text{and} \cdots \\
&\quad \text{and } x_n \text{ is } A_{nk''} \text{ then } y \text{ is } B_s \\
&\quad or \cdots or \\
&\quad if\ x_1 \text{ is } A_{nl'} \text{ and} \cdots \\
&\quad \text{and } x_n \text{ is } A_{nl''} \text{ then } y \text{ is } B_s
\end{aligned} \tag{14}$$

where $i',i'',j',j'',k',k'',l',l'' \in [1,t]$

On the view of general fuzzy inference, to a single rule, the firing strength (or weight) of ith rule is usually obtained by min or multiplication operator, to the complex rule, the firing strength is the union λ_i, which is like:

$$\begin{aligned}
\lambda_l &= \bigvee_{p_2=1}^{\Delta_l} (\mu_{A_{1i'} \times \cdots \times A_{ni''}}(x_1,\cdots,x_n), \\
&\qquad \cdots, \mu_{A_{1j'} \times \cdots \times A_{nj''}}(x_1,\cdots,x_n)) \\
&= \bigvee_{p_2=1}^{\Delta_l} \begin{pmatrix} \bigwedge_{p_1=1}^{\Psi_1} \left(\mu_{A_{1i'}}(x_1),\cdots,\mu_{A_{ni''}}(x_n)\right), \\ \cdots, \bigwedge_{p_1=1}^{\Psi_{nxt}} \left(\mu_{A_{1j'}}(x_1),\cdots,\mu_{A_{nj''}}(x_n)\right) \end{pmatrix}
\end{aligned} \tag{15}$$

where Δ is the number of the same consequent B_l, Ψ the number of antecedent parts in this rule.

On the view of fuzzy weighted Mamdani inference, local weight and global weight are considered by a mode of $<\vee,\wedge>$ as follows:

$$\lambda_l = \bigvee_{p_2=1}^{\Delta_l} \begin{pmatrix} v_1 \wedge \left(\bigwedge_{p_1=1}^{\Psi_1} \left(w_{1i'} \vee \mu_{A_{1i'}}(x_1),\cdots,w_{ni''} \vee \mu_{A_{ni''}}(x_n)\right)\right), \\ \cdots, v_{\Delta_l} \wedge \left(\bigwedge_{p_1=1}^{\Psi_{nxt}} \left(w_{1j'} \vee \mu_{A_{1j'}}(x_1),\cdots,w_{nj'} \vee \mu_{A_{nj''}}(x_n)\right)\right) \end{pmatrix} \tag{16}$$

In general, center-of-gravity method is adopted for defuzzification as follows.

$$Y = \frac{\sum_{l=1}^{s} \lambda_l \mu_{B_l}(y)}{\sum_{l=1}^{s} \lambda_l} \tag{17}$$

3.2 Functional equivalence and its implication

From (11) and (17), it is obvious that the functional equivalence between an AND-OR FNN and a fuzzy weighted Mamdani inference can be established if the following is true.

1. The number of AND neurons is equal to the number of fuzzy IF-THEN rules.
2. The number of OR neurons is equal to the number of fuzzy complex IF-THEN rules.
3. The T-norm and S-norm used to compute each rule's firing strength is min and max, respectively.
4. Both the AND-OR FNN and the fuzzy inference system under consideration use the same center-of-gravity method to derive their overall outputs.

Under these conditions, when $RDM(w_{ij}) = 0$, $RDM(v_{ij}) = 1$, AND-OR FNN is completely connected, which is shown that every part of antecedent is the same important to the output, $\Psi_k = n$, $\Delta_l = n \times t$ and equal to the general fuzzy inference. But it falls short of the practice. When $w_i \in [0,1]$, $v_i \in [0,1]$, every part of antecedent has different important degree to the output and every rules has different important degree to the final output.

4. Optimization for the AND-OR FNN

In general, gradient-based optimization is often chosen for ANN, Pedrycz [8] has proved that the output of AND neuron and its derivative are heavily affected by the increasing values of "n". Meanwhile, the final output of AND-OR FNN is not a smooth function. Obviously gradient-based optimization is not preferred, A far more comprehensive optimization policy is proposed, which is a hybrid learning methodology comprising of three fundamental techniques, namely genetic optimization, pruning algorithm and ant colony system.

4.1 Structure optimization

The key part is structure design to solve the problem using network, AND-OR FNN could be over parameterized, and in this case, they produce an over trained net, which leads to worse generalization capacity. That is the reason why, in order to eliminate unnecessary weights or parameters. There are no certain methods to solve the structure optimization. In this paper, GA (Genetic Algorithm) and PA (Pruning Algorithm) are adopted to optimize the structure for avoiding local optimization and reducing the connections.

4.1.1 GA optimization

GA (Genetic Algorithm) is a very effective method to solving the structure optimization problem because of the superiority of robust, random, global and parallel process[10]. Structure optimization is transferred to the species evolution by GA to obtain the optimal solution.

4.1.1.1 Structure code mode

Parameters encoding is the first phase of GA optimization. Parameters are often transformed to unsigned binary. Here the structure parameters of two hidden layers are considered on focus by double matrixes as follows:

$$\begin{aligned} RDM(AND) &= \left[w_{i,j}^{and} \right]_{m \times (n \times t)} \\ RDM(OR) &= \left[v_{i,j}^{or} \right]_{s \times m} \end{aligned} \tag{18}$$

where i, j, m, n, t and s is the same like before. Lining up the two matrixes to a binary string as initial population represents the structure status, which is exactly satisfied to GA. In the initialization, $RDM(w_{ij}) = 0$ and $RDM(v_{ij}) = 1$ are set, the status of AND-OR FNN is completely connected. The authoritative opinion from experts can be put into initial population as seeds to reach the best initialization.

4.1.1.2 Objective Function

Proceeding with the numeric data, we carry the genetic optimization of the network. The parameters of the GA used were chosen experimentally. The fitness function maximized by the GA is expressed in the form of the sum of squared errors between target values (data) and outputs of network. This fitness has to be minimized (it stands in contrast with the classic way of using fitness functions in GA whose values are maximized; if necessary a simple transformation of taking the reciprocal of the fitness, $1/fitness$, brings in line with the way of maximizing the fitness)

$$fitness = \frac{1}{(\sum (z_d - z)^2)} \tag{19}$$

4.1.2 Pruning algorithm

The pruning of the network is a process consisting of removing unnecessary parameters and nodes during the training process of the network without losing its generalization capacity [11]. The best architecture has been sought using GA in conjunction with pruning algorithms. There are three situations as follow:

1. For AND neuron, if the connection is equal to zero, this means that the corresponding input impacts the output of this neuron, the connection can be pruned away if its value is one. For the OR neuron a reverse situation occurs: the connection equal to one implies that the specific input is essential and affects the output of the neuron. The connection can be pruned away if its value is zero.
2. For $RDM(AND)$, the number of zero in every line is shown as the in-degree of this AND neuron, the node can be pruned away if all one in the line. For $RDM(OR)$, the number of one in every line is shown as the in-degree of this OR neuron, the node can be pruned away if all zero in the line.
3. For $RDM(AND)$, if there are the same line in the matrix, that means the node is redundant, can be pruned away. For $RDM(OR)$, that means the node is conflict each other, need be pruned away.

4.2 Ant Colony Optimization as parametric refinement

As well known, most neural networks are trained using Bp algorithm, whose main drawbacks are easily getting stuck in local minim and needing longer training time. And it also requires the output function is smooth, $\vee, \wedge$ operators can't satisfy at all. ACO (Ant colony optimization) algorithm [12] was adopted to optimize the connection value of AND-OR FNN here. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony

and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants.The main features are high-precision solution, fast search speed, convergence to global optimum and greedy heuristic search. The basic idea is: assuming that there are m parameters consist of the connections from GA and PA. First, all the parameters are sequenced and set to N randomly values in [0, 1], and for the parameter $p_i \ (1 \le i \le m)$, compose a set, denoted as I_{p_i}. And then the number of ants is set to h, they start from the random connection to search the optimal solution, choosing next connection from every set I_{p_i} according to the amount of pheromones as follows:

$$Prob(\tau_j^k(I_{p_i})) = \frac{\tau_j(I_{p_i})}{\sum_{g=1}^{N} \tau_g(I_{p_i})} \tag{20}$$

where $\tau_j(I_{p_i})$ is shown the amount pheromones of the jth connection $p_j(I_{p_i})$.

When an ant finishes choosing the connection in all the sets, it arrives at the food source, and returns its nest through the paths just gone by it, and then adjust the amount of pheromones corresponding to the chosen connection as follow rules.

$$\begin{aligned} \tau_j(I_{p_i})(t+m) &= \rho\tau_j(I_{p_i})(t) + \triangle\tau_j(I_{p_i}) \\ \triangle\tau_j(I_{p_i}) &= \sum_{k=1}^{h} \triangle\tau_j^k(I_{p_i}) \end{aligned} \tag{21}$$

where $\rho(0 \le \rho \le 1)$ is shown the evaporating of pheromone, $\triangle\tau_j^k(I_{p_i})$ is the kth ant leave pheromones on the jth connection in set I_{p_i}. It is described as follows:

$$\triangle\tau_j^k(I_{p_i}) = \begin{cases} Q / e^k & \text{if } p_j(I_{p_i}) \text{ is selected} \\ 0 & otherwise \end{cases} \tag{22}$$

where Q is the constant to adjust factor. $e^k = |z_d - z|$ is shown the network output error with a group of connection value, which is selected by the kth ant. Obviously the less the error is, the more the increment of the corresponding amount of pheromone is.

Finally, until all the ants find the best optimal solution for the connections value.

5. Application to ship control

According to the principle of manual steering, it makes full use of the advantages of fuzzy logic and neural network. An AND-OR FNN controller is presented to adapt the change of navigation environment and get the information of ship maneuverability automatically. The structure and parameters of AND-OR FNN controller are learned by GA, PA and ACO. Fuzzy rules can be auto-obtained from a group of sample data, which is generated by a fuzzy control system. Fig.4 shows a block diagram of the AND-OR FNN autopilot. Test results show that an effect method has been provided for the improvement of ship steering control.

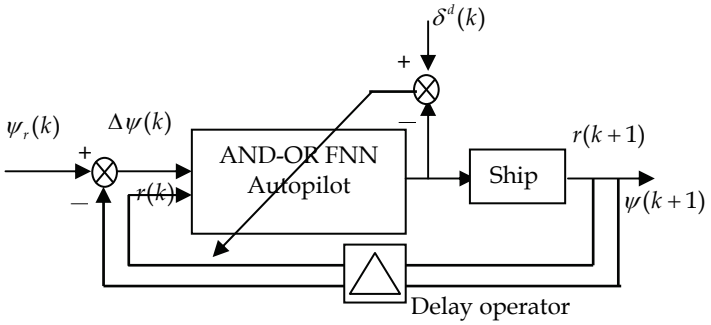


Fig. 4. AND-OR FNN autopilot for ship course-keeping

Double input variable of AND-OR FNN ship controller are heading error $\Delta\psi = \psi_k(k) - \psi(k)$ and yaw rate $\gamma(k) = \frac{d\psi}{dt}$. The control action generated by the autopilot is the command rudder angle $\delta(k)$. The range of values (universe of discourse) for a given autopilot inputs and output are $(-20^\circ, 20^\circ)$ and $(-2.5^\circ/\text{sec}, 2.5^\circ/\text{sec})$, respectively. It is usually required that the rudder should move from 35° port to 35° starboard within 30s.

5.1 The source of the experimental data set

In order to obtain integrated and exact training data, a fuzzy control system for a tanker is simulated with nonlinear Nomoto model as controlling plant. The input gains of a fuzzy controller (K=0.16sec^{-1}, T=100sec). The fuzzy controller includes double inputs, the error in the ship heading ($\Delta\psi(k)$) and the change in that error ($\gamma(k)$). Every input variable is fuzzied into 11 triangular membership functions; thus form 121 pieces of rules. The output of the fuzzy controller is the rudder input (δ). We want the tanker heading to track the reference input heading ψ_r. We assume the tanker as a continuous time system controlled by the fuzzy controller, which is implemented on a digital computer with a sampling interval of T=0.1sec. From manipulated index standpoints, it satisfied the index of 15000tdw tanker [13], a fuzzy closed-loop controlling system is designed with fuzzy rule base from experience. The positive direction angle is defined that the rudder angle goes around toward right. Center-of-gravity is adopted as defuzzification methods.

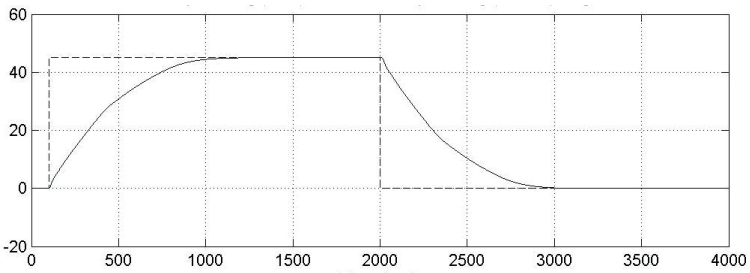


Fig. 5. Ship heading (solid) and desired ship heading (dashed)

The reference input heading ψ_r is increased 45 degree every 2000sec, rapidly. The digital simulation is completed to closed-loop with the sampling periods h=10sec, the sampling state variable $\Delta\psi$, γ and controlled variable δ , the ship heading and desired ship heading is like Fig. 5 and we obtain the change of rudder angle results utilizing conventional fuzzy control as Fig. 6.and the data sequence is $\{\Delta\psi(k),\gamma(k),\delta(k),k=1,2,...\}$.

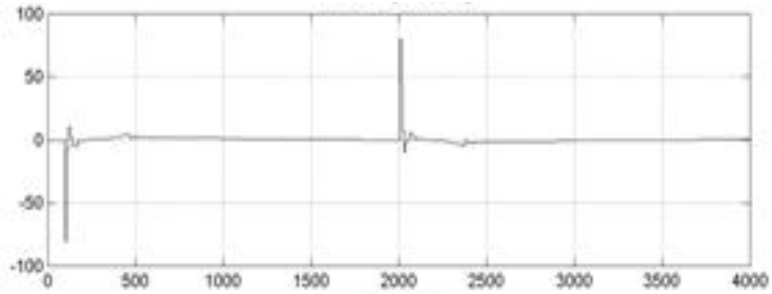


Fig. 6. Rudder angle in the conventional fuzzy control

5.2 The structure design and simulation result

In this section, an AND-OR FNN ship controller is constructed successfully. Firstly, $\Delta\psi(k)$ and $\gamma(k)$ are double input variable; $\delta(k)$ is the output variable, which are all fuzzied into 3 Gaussian membership functions with 0.5 overlap degree, thus there are 9 pieces of rules. According to analysis before, there are 9 AND neurons and 3 OR neurons in AND-OR FNN. Initialize structure parameters with all connections to form the structure string that is $RDM(w_i)=0$, $RDM(v_i)=1$. The input-output data is from $\{\Delta\psi(k),\gamma(k),\delta(k),k=1,2,...\}$ to train and test. .The structure and parameters are optimized by GA PA and ACO, some parameters is chosen experimentally, which is including population size=100, crossover rate=0.7, Mutation rate=0.01 and selection process is tournament, and the parameters of ACO are $m=19$, $p_i\in[0,1]$, $N=30$, $\rho=0.65$, $h=30$, $Q=10$ .The ultimate better structure result is transformed into the AND-OR FNN like Fig. 7. It is obviously that there are 5 AND neurons and 3 OR neurons left, that is, 5 pieces of fuzzy rules and 3 complex rules. The ultimate better performance result is compared with fuzzy control system. Test data set is

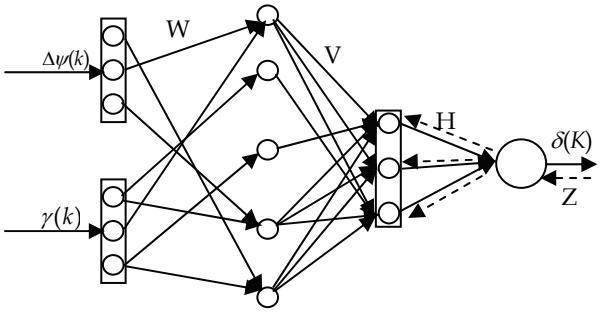


Fig. 7. The AND-OR FNN ship controller structure

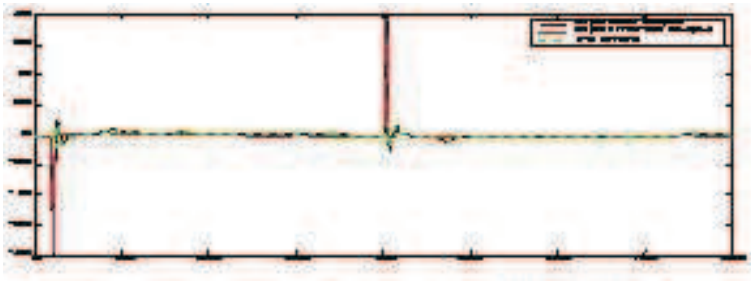


Fig. 8. The performance index compared

form the different part with training data in the same sequence introduce above. Fig. 8 shows the error comparison between test results and objective data. The simulation result illustrated the effectiveness of proposed method.

5. Conclusion

In this paper, we have proposed a novel AND-OR FNN and a piecewise optimization approach, the symbol expressions of every node in the AND-OR FNN are educed in detail, the input space is reduced by special inner structure of AND and OR. The equivalence is proved to the fuzzy weighted Mamdani inference. The fuzzy rule base auto-extracted is equal to optimize the architecture of AND-OR FNN. This novel approach has been validated using AND-OR FNN ship controller design. The simulation results illustrate the approach is practicable, simple and effective and the performance is much better than ordinary fuzzy controller.

6. Acknowledgements

This paper is supported by the Doctoral Foundation of Education Committees (2003151005) and Ministry of Communication of P. R. China (200332922505)

7. References

[1] Pierre Yves G.: Neuro-Fuzzy Logic [A].In: Proc. IEEE-FUZZ[C]. New Orleans:[s.n.], (1996).512-518.

[2] Yager R.: OWA neurons: Anew class of fuzzy neurons [A]. In: Proc. IEEE-FUZZ [c].San Diego:[s.n.],(1992).2316-2340.

[3] Pedrcy. W. and Rocha. A F.: Fuzzy-set based models of neurons and knowledge-based networks. IEEE.Trans. on Fuzzy System,1(4)(1993)254-266

[4] Hirota K, Pedrycz W.: Knowledge-based networks in classification problems. Journal, Fuzzy Sets and Systems, 59(3) (1993) 271-279

[5] Pedrycz W and Nicolino J.Pizzi.: Fuzzy adaptive logic networks Proceeding of NAFIPS 2002 June (2002) 500 -505

[6] Pedrycz W and Succi.G.: Genectic granular classifiers in modeling software quality. The journal of Systems and software, 75(2005)277-285

[7] Bailey S A, Chen Ye hwa.: ATwo Layer Network using the OR/AND Neuron [A]. In: proc. IEEE-FUZZ [C].Anchorage:[s.n.],(1998).1566-1571.

[8] Pedrycz W. Reformat M.: "Genetically optimized logic models" Fuzzy Sets and Systems 150(2) (2005)351-371

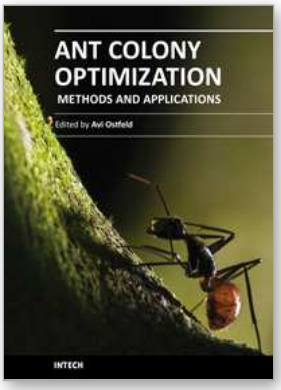
[9] Yeung D.S and Tsang E.C.C.: Weighted fuzzy production rules. Fuzzy sets and systems 88(1997)299-313

[10] Shuiling Zeng, Luanjiao Song and Weihong Xu.: Neural network structure optimization based on genetic algorithm. Journal of Jishou University Vol.26 No3 (2005)118-120

[11] Rosa Maria Garcia-Gimeno, Cesar Hervas-Martinez and Maria Isabel de Siloniz.: Improving artificial neural networks with a pruning methodology and genetic algorithms for their application in microbial growth prediction in food. International Journal of food Microbiology, 72(2002)19-30

[12] Haibin Duan.: Ant colony algorithm: theory and applications, science press (2005)390-397

[13] Guoxun Yang.: Study on ship motion hybrid intelligent control and its interacting simulation based on virtual reality. PhD Thesis. Dalian: Dalian Maritime University,(2002)



Ant Colony Optimization - Methods and Applications

Edited by Avi Ostfeld

ISBN 978-953-307-157-2
Hard cover, 342 pages
Publisher InTech
Published online 04, February, 2011
Published in print edition February, 2011

Ants communicate information by leaving pheromone tracks. A moving ant leaves, in varying quantities, some pheromone on the ground to mark its way. While an isolated ant moves essentially at random, an ant encountering a previously laid trail is able to detect it and decide with high probability to follow it, thus reinforcing the track with its own pheromone. The collective behavior that emerges is thus a positive feedback: where the more the ants following a track, the more attractive that track becomes for being followed; thus the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. This elementary ant's behavior inspired the development of ant colony optimization by Marco Dorigo in 1992, constructing a meta-heuristic stochastic combinatorial computational methodology belonging to a family of related meta-heuristic methods such as simulated annealing, Tabu search and genetic algorithms. This book covers in twenty chapters state of the art methods and applications of utilizing ant colony optimization algorithms. New methods and theory such as multi colony ant algorithm based upon a new pheromone arithmetic crossover and a repulsive operator, new findings on ant colony convergence, and a diversity of engineering and science applications from transportation, water resources, electrical and computer science disciplines are presented.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jianghua Sui (2011). An AND-OR Fuzzy Neural Network, Ant Colony Optimization - Methods and Applications, Avi Ostfeld (Ed.), ISBN: 978-953-307-157-2, InTech, Available from: http://www.intechopen.com/books/ant-colony-optimization-methods-and-applications/an-and-or-fuzzy-neural-network

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821