A New Neural Fuzzy System Using Fuzzy Linguistic Input-output Training Samples

Jing Lu , Blayne Mayfield , and Jia Liu

Computer Science Department, Oklahoma State University , Stillwater, OK, USA

Abstract - *This paper proposes a new fuzzy neural system that deals with fuzzy training samples. That is, the proposed learning system includes training of a neural network using fuzzy training samples, and it outputs fuzzy value according to a set of given rules. The learning algorithm used in the system is composed of two phases. The first phase transforms crisp samples into fuzzy ones; the second phase trains the network according to the fuzzy samples through a new neural network structure proposed in this paper. Finally, two error evaluation methods and comparison between them are given.*

Keywords: Fuzzy logic, neural network, mean solution, gragh solution.

1 Introduction

Fuzzy logic [1] and neural networks [2] play an important role in the modeling of intelligent control in complex systems. Their combination is widely used in solving problems such as classification, identification, pattern recognition and so on.

There are three common ways to combine these techniques. The first one is a fuzzy neural network [3] (FNN), which increases the efficiency of the neural network and its velocity of convergence; the second one is a neural-fuzzy system[4], the activation function of which is related to fuzzy relations or fuzzy operators; the third one is a fuzzy neural hybrid system[5], where fuzzy logic and neural networks perform separately to attain a common goal.

In our previous studies, the first and second types used crisp input-output samples to train the neural network and to output crisp values. Using these approaches, FNNs embed fuzzy logic into a neural network and use that fuzzy logic in order to reach some goal; however, the input and output values of these networks are crisp values. In this paper, by contrast, the neural network uses crisp logic, but the input and output values are fuzzy values. In other words, the previous approaches and the new approach operate in opposite ways.

The new neural fuzzy system can deal with fuzzy input-output training samples, but also with crisp training samples. And, it can output fuzzy values according to fuzzy rules given in advance. The new approach adds two extra layers in front of the network to generate fuzzy training samples. These additional layers generate crisp training samples and then fuzzify them for use as input to the network.

The organization of this paper is as follows. In section 2, we briefly introduce the preliminaries of the paper: fuzzy logic theory and neural network theory. Section 3 presents the main structure of the system and algorithm used in the new network. Section 4 is an example of the system. Finally, the experiment of the example is shown and the conclusion is summarized.

2 Preliminaries

2.1 Fuzzy Logic Theory

Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic theory, where binary sets have two-valued logic, fuzzy logic variables may have a true value that ranges in degree between 0-1. Fuzzy logic is used to handle the concept of partial truth, in which the value may range between completely true and completely false. Furthermore, when linguistic variables[6] are used, these degrees may be managed by specific functions called membership function, which is a function from a universal set U to the interval [0,1]. A fuzzy set A is defined by its membership function μ_A over U.

2.2 Neural Network Theory

Artificial neural networks refer to computer algorithms and structures that mimic the function of the biological neutrons. Similarly, it is composed of artificial neurons. The connections of the biological neuron are modeled as weights. A negative weight reflects an inhibitory connection, while positive values mean excitatory connections. The following components of the model represent the actual activity of the neuron cell. All inputs are modified by a weight and summed altogether. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1.

Mathematically, this process is described in Figure 1.

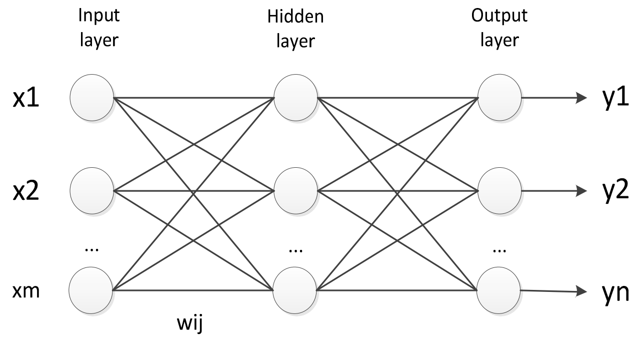

Figure 1. Neural network structure.

This network has an input layer (on the left) with m neurons, a hidden layer (in the middle) and an output layer (on the right) with n neurons.

There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*- 1 neurons are used to represent the *N* categories of the variable.

Input Layer: A vector of predictor variable values from x1 to xm is presented to the input layer. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the bias that is fed to each of the hidden layers; the bias is multiplied by a weight wij and added to the sum going into the neuron.

Hidden Layer: Arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight, and the resulting weighted values are added together producing a combined value, which is fed into a transfer function, which outputs a value. The outputs from the hidden layer are distributed to the output layer.

Output Layer: The y values are the outputs of the network. If a regression analysis[7] is being performed with a continuous target variable, then there is a single neuron in the output layer, and it generates a single y value. For classification problems with categorical target variables, there are *N* neurons in the output layer producing *N* values, one for each of the *N* categories of the target variable.

3 Neural fuzzy system with crisp input and fuzzy output

Herein, we consider a neural fuzzy system with a rule base of R rules, e.g., n-input m-output. The jth control rule is described as following form:

Rj: IF x1 is A1 AND x2 is A2 AND …AND xn is An, THEN y1 is B1 AND y2 is B2 AND … AND ym is Bm. Where j is a rule number, n is the total number of input variables and m is the total number of output variables, A1-An denotes the fuzzy set over the universe of x(X) and maybe not identical to each other, B1-Bm denotes the fuzzy set over the universe of y(Y) and maybe not identical to each other.

However, we cannot use fuzzy words to train the neural network and the conception of fuzzy logic can be introduced to solve such problem. In fact, it is necessary to use fuzzy values which depict the degree to which a crisp value belongs to a fuzzy set (fuzzy words) to train the neural network. By transferring the crisp samples into fuzzy ones, we could train the neural network using the fuzzy samples and output fuzzy values.

The whole structure of the neural fuzzy system with fuzzy output is show as Figure 2.

We can see from the picture that the whole neural network has several parts: Input layer, Fuzzification layer, Neural training layer and Output layer.

There may be many dashed parts. The dashed area is similar to the solid part but differ from each other corresponding to different rules. That is, the number of the solid parts and dashed parts depend on the number of the rules. We will give the reason why there is dashed part later.

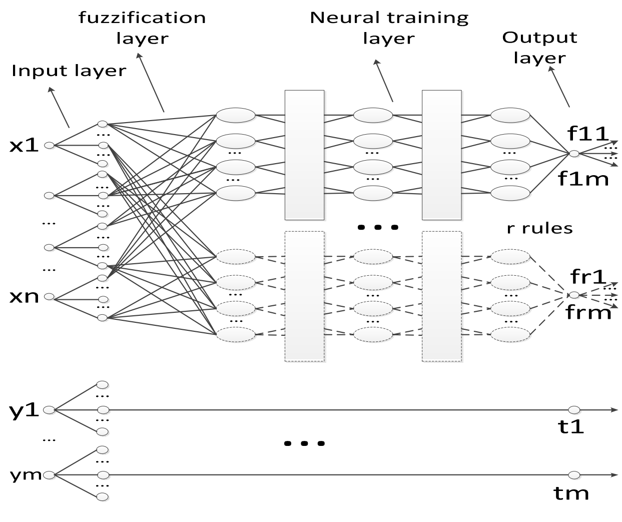

Figure 2. Fuzzy neural system.

Take a simple rule as an example:

(R1) IF x1 is Small AND x2 is Small, THEN y is Small; (R2) IF x1 is Small AND x2 is Large, THEN y is Middle; (R3) IF x1 is Large AND x2 is Small, THEN y is Middle; (R4) IF x1 is Large AND x2 is Large, THEN y is large.

There are 4 rules in the example, where n is 2 and m is 1, A1 (Small) and A2 (Small) denotes the fuzzy set (fuzzy words) over the universe of x, which are not identical to each other in rule 1 but identical in rule 2, since A1 is Small and A2 is Large, B1denotes the fuzzy set (fuzzy words) “Small” over the universe of y, B2 denotes the fuzzy set (fuzzy words) “Middle” over the universe of y, B3 denotes the fuzzy set “Large” over the universe of y.

The network structure of this example is show as Figure 3.

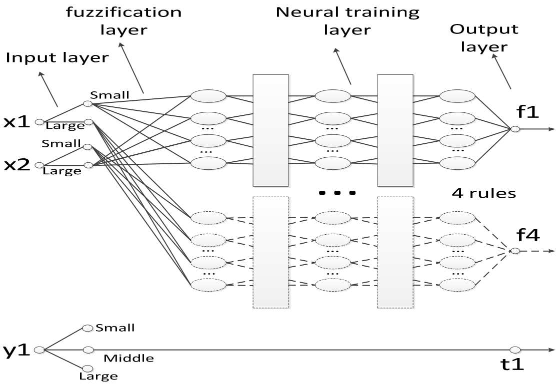

Figure 3. Fuzzy neural system with two inputs and one output.

Specification:

Input layer: If the neural fuzzy system is given suitable fuzzy training samples containing fuzzy values, this layer isn’t a necessary part. But, what is the suitable fuzzy training sample? Just as the name implies, a suitable fuzzy training sample is a training sample whose members are fuzzy values corresponding to some fuzzy set (word). Besides, “suitable” means that the training sample must obey the rules we set in advance.

The main goal of this layer is to generate some crisp training samples randomly, each of which contains independent variable values and expected values of induced variable.

Using the example above, obviously, there are 2 independent variable values (x1, x2) and 1 expected value y1. Specifically, we should generate members of a training sample in the same universe of discourse for x and do the same for y.

Fuzzification layer: This layer is also redundant if the neural fuzzy system is given suitable fuzzy training samples containing fuzzy values.

If necessary, it transfers the training samples into fuzzy ones according to fuzzy logic theory. For each value in a sample, firstly, the system gets several membership function values corresponding to the current value being checked within its universal; secondly, compare membership function values and get the biggest one as a member of the fuzzy sample, at the same time, records the fuzzy word with the biggest membership function value. That is the maximum membership degree principle[8].

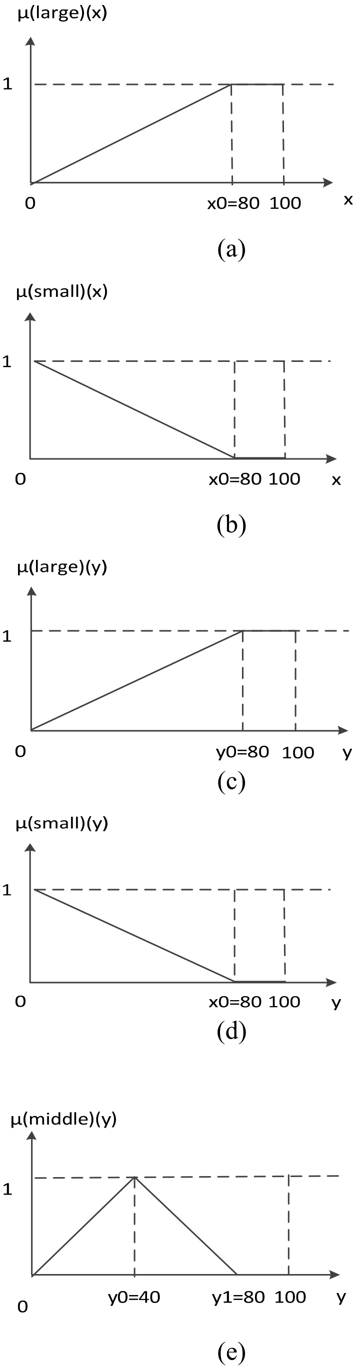

Figure 4. Membership fuction curve of fuzzy set (a) “large” with universal x, (b) “small” with universal x, (c) “large” with universal y, (d) “small” with universal y, (e) “middle” with universal x.

Assuming there is a training sample <24, 42, 90>.Firstly, the system gets 2 values for "24" according to the curve in graph(a) and graph(b); Secondly, compare them, get the biggest one --- max(μ_{large}(24), μ_{small}(24)) and the fuzzy words "small" since max(μ_{large} (24), μ_{small} (24)) is μ_{small}(24).

Sometimes, we generate some samples which are not accordance with the fuzzy rules we care, at this time, we discard these useless samples.

In the given example, we should notice (from the membership function graph of x and y) that the fuzzy sets (fuzzy words) over x and over y we have defined are not always the same but it could be.

Neural training layer: The function of this layer is similar to existing neural network, but the difference between them is that the system proposed in this paper use different structure related to the different rule to train the neural network in order to output expected value in keeping with the rule. In addition, we may make use of the same neural training layer working area, instead of open up a new working area. That is why there is dashed part in Figure 2 and 3.

Output layer: This layer outputs fuzzy values labeled by output fuzzy words. It is easy to see that one output has two meanings. One is the fuzzy set (word), the other is the fuzzy value corresponding to this fuzzy set. The point is how to decide for a fuzzy output which fuzzy set it belongs to. Deciding the fuzzy set of the output according to the given rules and the record of the fuzzy words of the inputs is a direct method.

4 Evaluation methodology

4.1 Graph solution

Here, we introduce a new evaluation methodology called "graph solution" to evaluate the performance of the neural network. This method can only be applied when the membership functions are given.

Step1: Draw the membership function of different fuzzy sets given the same universal in one graph.

For example, Figure 5 denotes the graph solution for y within the universal 0-100.

Step2: Find the biggest values for each interval. For example, [0-27]: μ_{small}(y) has the biggest values;[27-53]: μ_{middle}(y) has the biggest values;[53-100]:(y) has the biggest values.

Step3: Within each interval, find out the difference of the biggest membership function value and the smallest membership function value for the fuzzy word with the biggest values. Obviously, the example above has the same differences, which are "μ_{small}(0) - μ_{small}(27)", "μ_{middle}(40) - μ_{middle}(27)" and "μ_{large}(80) - μ_{large}(53)".

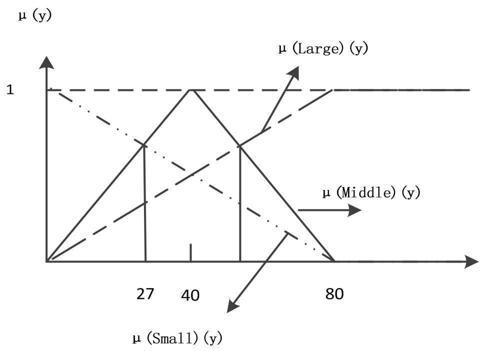

Figure 5. Graph solution.

If the error is between "0" to "0.47", the training of the neural network is reasonable.

We should notice that this method is useless if we do not know the membership function.

4.2 Mean solution

If there is a one-many relationship between the inputs and targets in the training data, then it is not possible for any mapping of the form to perform perfectly. It is straightforward to show that if a probability density P(Y|X) describes the data, then the minimum of error measure is attained by the map taking **X** to the average target

$$\int d\boldsymbol{Y}\, \mathrm{P}(\mathbf{Y}|\mathbf{X})\mathbf{Y}$$

Any given network might or not be able to approximate this mapping well, but when trained as well as possible it will form its best possible approximation to this mean.

For the fuzzy linguistic inputs and outputs, one input set may match along with many outputs. We calculate the average target based on P(**Y**|**X**) for each rule.

5 Experiment

Rule specification:

(R1) IF x1is Small AND x2 is Small, THEN y is Small; (R2) IF x1 is Small AND x2 is Large, THEN y is Middle; (R3) IF x1 is Large AND x2 is Small, THEN y is Middle; (R4) IF x1 is Large AND x2 is Large, THEN y is large.

Original training sample:

Universal of x (X) and Universal of y (Y) are [0,100]. We just generate randomly these samples such as (x1, x2, y), each of which ranges from 0 to 100, inclusively.

Algorithm to train the neural network:

BP Algorithm.

Neural network structure:

4 layers having 3, 3, 3 and 1 neutrons respectively.

Experiment environment:

Matlab.

5.1 Experiment using graph solution

The system generates 100000 samples randomly, where 28793 samples are satisfied with the 4 rules in the example. Using these training samples to train the neural network, we get error values below (Figure 6) and the error is reasonable since it fall within the interval [0-0.47] according to the "graph solution" mentioned above.

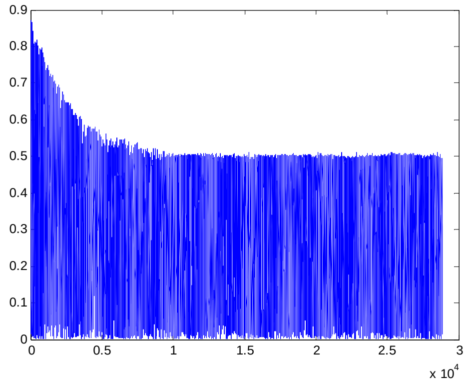

Figure 6. Training error using graph solution.

Using 14428 testing samples on this network, we get error values shown as Figure 7. The result is approximately between 0-0.47 and reasonable.

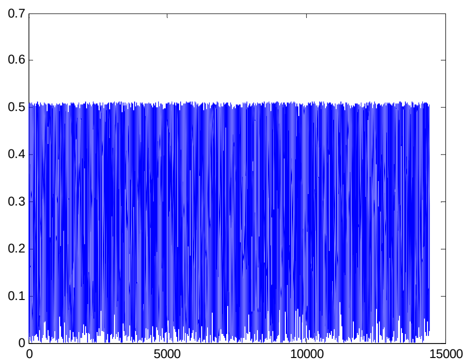

Figure 7. Testing error using graph solution.

5.2 Experiment using mean solution

Assuming P(**Y**|**X**) of each rule is an average distribution, we easily calculate the average target for four rules and they are 0.8313, 0.8375,0.8375,0.8313 separately. We get error values below (Figure 8).

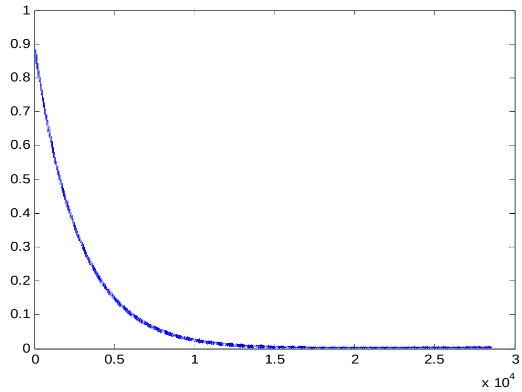

Figure 8. Training error using mean solution.

Using 14455 testing data samples on this network, we get error values shown as Figure 9. The result is between 0.015-0.04 and reasonable.

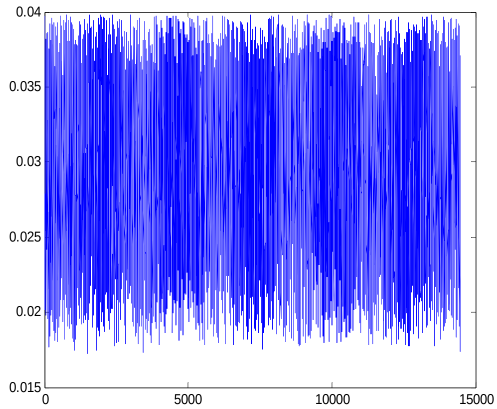

Figure 9. Testing error using mean solution.

Experiments show that the graph solution goes a litter faster than the mean solution to get to a stable state. However, mean solution could deal problems which have background knowledge P(**Y**|**X**). They are determined by properties of probability mathematics and the practical needs. So, we could realize that mean solution make use of known conditions more sufficiently. On the other hand, graph solution would be chosen if we do not want to fix fuzzy output and there is no background knowledge at all.

6 Conclusions

This neural fuzzy system could deal with problem of fuzzy rules.

Obviously, this system can input crisp values and output fuzzy values, which is more convenient compare to previous fuzzy neural networks with crisp inputs and outputs.

In addition, this system is also suitable for fuzzy reference problems. That is, if we are given the membership function value of X, what the membership function value of Y should be? It is already known that there are different fuzzy reasoning rules defined by human-beings in the fuzzy logic area, such as Mamdanian Larsen fuzzy reference[9], Zadeh fuzzy reference[1], and so on. We may choose different methods doing fuzzy reference. But which method is better? It depends on needs. For complex reference problem, fuzzy neural network system could be used to get the relationship between fuzzy variable X and Y---R(X, Y), which is a good method to get a suitable rule according to our needs.

7 References

[1] L. A. Zadeh. “Fuzzy Sets”. *Information and Control*, 1986.

[2] J. M. Zurada. “*Introduction to Artificial Neural Systems*”. *West Publishing Company*, pp. 175-181, 1992.

[3] C. T. Lin and C. S. G. Lee. “Neural-network-based Fuzzy Logic Control and Decision system”, *IEEE Transactions on Computers*, vol. 40, pp. 1320-1336, Dec. 1991.

[4] E. Soria-Olivas and J.D. Martin-Guerrero. “A Low Complexity Fuzzy Activation Function for Artificial Neural Networks”, *IEEE Transactions on Neural Networks*, vol. 14, pp.1576-1579, Nov 2003.

[5] W. Li. “A Method for Design of a Hybrid Neuro-Fuzzy Control System Based on Behavior Modeling”, *IEEE Transaction Fuzzy System*, pp. 462-477, 1997.

[6] L. A. ZADEH. “The Concept of a Linguistic Variable and its Application to Approximate Resoning-I”, *Information Science*, pp. 199-249, 1975.

[7] D.V. Lindley. “Regression and correlation analysis”, New Palgrave: A Dictionary of Economics, pp. 120-23., 1987.

[8] QiuWen Zhang and Ming Zhong. “Using Multi-level Fuzzy Comprehensive Evaluation to Assess Reservoir Induced Seismic Risk”, Journal of Computers, pp. 1670-1677, 2011.

[9] W. J. M Kickert and E. H. Mamdani. “Analysis of a Fuzzy Logic Controller”, *Fuzzy Sets and Systems,* pp. 29-44, 1978.