Pattern Recognizer Based on Neuro-Fuzzy Networks Using Fuzzy Relation Input Space

Keon-Jun Park[1], Jae-Hyun Kwon[1], Hyung-Kil Kang[1] and Yong-Kab Kim[1*]

[1] Department of Information and Communication Engineering,
Wonkwang University, 344-2, Shinyong-dong, Iksan-si, Chonbuk, 570-749 South Korea
bird75@wonkwang.ac.kr, martkiller@naver.com ykim@wonkwang.ac.kr

Abstract. In this paper, we introduce a pattern recognizer based on neuro-fuzzy networks using fuzzy relation input space. The premise part of the rules of the proposed networks is realized with the aid of the grid partition of the fuzzy relation input space. The partitioned spaces express the rules of the networks. The consequence part of the rules is represented by polynomial functions. The coefficients of the polynomial functions are learned by the back-propagation algorithm. To optimize the parameters of the proposed networks, we consider real-coded genetic algorithms. The proposed networks are evaluated with the use of numerical experimentation for pattern recognizer.

Keywords: Neuro-Fuzzy Networks (NFNs), Grid partition, Fuzzy Relation Input Space, Genetic Algorithms (GAs), Pattern Recognizer.

1 Introduction

Neuro-fuzzy networks (NFNs) [1, 2, 3] have emerged as one of the active areas of research in fuzzy inference systems and neural networks. These networks are predominantly designed for the integration of these two fields. Typically, NFNs are represented by fuzzy “if–then” rules, while back propagation (BP) is used to optimize the parameters.

The generation of the fuzzy rules and the adjustment of their membership functions of the NFNs were conducted by trial and error and/or on the basis of the operator’s experience. The designers find it difficult to develop adequate fuzzy rules and membership functions to reflect the essence of the data.

In this paper, we introduce the structure of neuro-fuzzy networks with multiple outputs based on fuzzy relation input space. The premise part of the rules of this network is realized with the aid of the grid partition of the input space. The consequence part of the rules is represented by polynomial functions with multiple outputs for pattern recognition. The coefficients of the polynomial functions are learned by the BP algorithm. We also optimize the parameters of the networks using real-coded genetic algorithms (GAs) [4]. The proposed network is evaluated through numerical experimentation for pattern recognition.

* Corresponding author : ykim@wonkwang.ac.kr

2 Genetic Design of the FR-based NFNs

2.1 Design of FR-based NFNs

The structure of the fuzzy relation-based NFNs (FR-based NFNs) emerges at a junction of fuzzy sets by means of the grid partition of fuzzy relation input space in the premise part and neural networks present in the consequence part of the rules. The structure of the FR-based NFNs is composed of six layers.

The proposed FR-based NFNs are implied by the fuzzy grid partition of input spaces. In this sense, each rule can be viewed as a certain rule of the following format.

$$R^j : If \ \ x_1 \ is \ A_{1c} and \cdots and \ x_d \ is \ A_{dc} \ \ Then \ \ y_{sj} = f(x_1, \ldots, x_d) \cdot \qquad (1)$$

As far as inference schemes are concerned, we distinguish these cases:

Type 1 (Simplified Inference):

$$f = w_{j0}^s$$

Type 2 (Linear Inference):

$$f = w_{j0}^s + \sum_{k=1}^{d} w_{jk}^s x_k$$

Type 3 (Modified Quadratic Inference):

$$f = w_{j0}^s + \sum_{k=1}^{d} w_{jk}^s x_k + \sum_{k=1}^{d} \sum_{i=k+1}^{d} w_{jz}^s x_k x_i \cdot$$

To be more specific, R^j is the j-th fuzzy rule, while A_{kc} denotes j-th membership function. w's are consequent parameters of the rule.

The functionality of each layer is described as follows.

[Layer 1] The nodes in this layer transfer the inputs.

[Layer 2] The nodes here are used to calculate the membership degrees for triangular membership functions.

[Layer 3] The nodes in this layer compute the firing strength for each rule.

$$f_j = \prod_{k=1}^{d} \mu_{kc}(x_k) \cdot \qquad (2)$$

[Layer 4] The nodes in this layer normalize the membership degrees

$$\hat{f}_j = f_j \Big/ \sum_{j=1}^{n} f_j \cdot \qquad (3)$$

[Layer 5] The nodes in this layer realize a certain inference process.

$$h_{sj} = \sum_{j=1}^{n} \hat{f}_j y_j \cdot \qquad (4)$$

[Layer 6] The nodes in this layer compute the outputs.

$$\hat{y}_s = \sum_{j=1}^{n} h_{sj} \cdot \qquad (5)$$

The parametric learning of the network is realized by adjusting connections of the neurons by running a standard back-propagation (BP) algorithm.

2.2 Optimization of FR-based NFNs

It has been demonstrated that genetic algorithms (GAs) [4] are useful global population-based optimizers. GAs are shown to support robust search in complex search spaces. In order to optimize the parameters of the FR-based NFNs, we determined the apexes of the membership functions of each input variable, the learning rate, and the momentum coefficient as the parameters. Each chromosome is coded using real numbers. This type of coding is helpful from the point of view of the effectiveness of the overall search process.

3 Experimental Studies

In this paper, we use the Iris dataset [5]. The Iris dataset is a collection of 150 Iris flowers of 3 kinds, with four attributes, leaf and petal width and length in cm. Three classes are the setosa, versicolor, and virginica.

For the evaluation of the performance of the network, the random sub-sampling method was applied. The random sub-sampling was performed with 5 data splits of the data set. Each split was randomly selected from the training examples and the test examples with the ratio of 7:3. We experimented with the proposed networks using the parameters outlined in Table 1 with the weight factor $\theta = 0.5$.

Table 2 shows the performance for FR-based NFNs using genetic optimization. From the table 2, we know that the optimized FR-based NFNs have the good results. We select the network that has three MFs for each input variable and linear inference engine. This network exhibits CR=99.24±0.43, PI=0.017±0.00 for training datasets and CR=99.56±0.99, PI=0.014±0.01 for testing datasets.

Table 1. Initial parameters.

	Parameter	Value
GAs	Generation	100
	Population size	50
	Crossover rate	0.65
	Mutation rate	0.1
NFNs	Apexes of MFs	2, 3
	Learning rate	$0.0 \leq \eta \leq 0.01$
	Moment coefficient	$0.0 \leq \alpha \leq 0.001$

The performance of the proposed network is compared with the performance of some other models reported in the literature; refer to Table 3. The comparison shows that the proposed network outperforms several previous developed models.

Table 2. Performance of the optimized FR-based NFNs.

No. of MFs	Inference (Type)	CR		PI	
		Training	Testing	Training	Testing
2	1	98.10±0.95	98.67±1.22	0.031±0.01	0.028±0.01
	2	99.43±0.85	98.67±1.22	0.021±0.00	0.022±0.01
	3	99.43±0.52	98.22±1.86	0.022±0.00	0.021±0.00
3	1	99.43±0.52	97.78±2.22	0.018±0.00	0.025±0.01
	2	99.24±0.43	99.56±0.99	0.017±0.00	0.014±0.01
	3	99.43±0.52	98.67±1.22	0.014±0.00	0.017±0.01

Table 3. Comparison of performance with previous models.

Model	Classification Ratio (%)
NEFCLASS / C4.5 / FID3.1	96.0 / 94.0 / 96.0
HNFB / HNFQ / HNFB-1	98.67 / 98.67 / 98.67
Our model	99.56

4 Discussion

In this paper, we introduced fuzzy relation-based neuro-fuzzy networks and discussed its optimization using real-coded genetic algorithms for pattern recognition. The input spaces of the proposed networks were divided as the grid form to generate the fuzzy relation-based rules. From this method, we have designed neuro-fuzzy networks. And genetic algorithms were also used for parametric optimization of the proposed networks. From the result in the previous section, we were able to design good networks and to achieve a balance between the approximation and generalization abilities of the resulting network using CR and PI. The proposed networks may encounter difficulties in case of high-dimension problem and theses dimensionality issues need to be tackled (e.g., through exploiting various partition strategies).

References

1. Yamakawa,T.: A Neo Fuzzy Neuron and Its Application to System Identification and Predition of the System Behavior. Proceeding of the 2nd International Conference on Fuzzy logic & Neural Networks, (1992) 447--483
2. Buckley, J.J., Hayashi, Y.: Fuzzy neural networks: A survey. Fuzzy Sets Syst. 66, 1--13, (1994)
3. Jang, J.S.R., Mizutani, E. and Sun, C.T.: Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence, Prentice Hall, NJ, 1997.
4. Golderg, D.E.: Genetic Algorithm in search, Optimization & Machine Learning, Addison wesley (1989)
5. UCI Machine Learning Repository: Data Sets, http://archive.ics.uci.edu