Design of Neuro-Fuzzy Networks Based on Respective Input Space for Pattern Recognition

Keon-Jun Park[1], Jun-Myung Lee[1], Jung-Won Choi[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, cjw03871@naver.com ykim@wonkwang.ac.kr

Abstract. The design of neuro-fuzzy networks based on fuzzy respective input space for pattern recognition is introduced in this paper. The premise part of the rules of the proposed networks is realized by partitioning of the fuzzy respective input space. The respectively partitioned spaces express the rules of the networks. The consequence part of the rules is represented by polynomial functions. The coefficients of consequence part of the rules are learned by the back-propagation algorithm. And the proposed networks are optimized using real-coded genetic algorithms. A numerical example for pattern recognition is given to evaluate the validity of the proposed networks.

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

1 Introduction

Neuro-fuzzy networks (NFNs) [1, 2, 3] are predominantly designed for the integration of the fuzzy inference systems and neural networks. 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. Some enhancements to the networks have been studied by many researchers, yet the problem of finding good parameters of the fuzzy sets and of partitioning spaces in the rules remains open.

In this paper, we introduce the structure of neuro-fuzzy networks with multiple outputs based on respective input space. The premise part of the rules of this network is realized by partitioning the respective 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 Design of the Respective Space-based NFNs

2.1 The structure of the Respective Space-based NFNs

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

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

$$R^{kc}: If \;\; x_k \; is \; A_{kc} \;\; Then \;\; y_{kc}^s = f(x_1, \ldots, x_d) \cdot \tag{1}$$

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

Type 1 (Simplified Inference):

$$f = w_{kc}^{s0}$$

Type 2 (Linear Inference):

$$f = w_{kc}^{s0} + \sum_{i=1}^{d} w_{kc}^{si} x_k$$

Type 3 (Modified Quadratic Inference):

$$f = w_{kc}^{s0} + \sum_{i=1}^{d} w_{kc}^{si} x_k + \sum_{i=1}^{d} \sum_{j=i+1}^{d} w_{kc}^{sz} x_k x_i \cdot$$

To be more specific, R^{kc} is the k, c-th fuzzy rule, while A_{kc} denotes k, c-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 to the respective inputs.

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

[Layer 3] The nodes in this layer normalize the membership degrees

$$\hat{f}_{kc} = \mu_{kc} \Big/ \sum_{c=1}^{g} \mu_{kc} = \mu_{kc}, \quad \sum_{c=1}^{g} \mu_{kc} = 1 \cdot \tag{2}$$

[Layer 3] The nodes in this layer compute the local output of each rule.

$$a_{kc}^s = \hat{f}_{kc} y_{kc}^s \cdot \tag{3}$$

[Layer 5] The nodes in this layer calculate the local output of each input.

$$h_{sk} = \sum_{c=1}^{g} a_{kc}^s \cdot \tag{4}$$

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

$$\hat{y}_s = \sum_{k=1}^{d} h_{sk} \cdot \tag{5}$$

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

2.2 Genetic Optimization

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. Given their stochastic character, such methods are less likely to get trapped in local minima (which becomes quite a common problem in case of gradient-descent techniques). In order to optimize the parameters of the RS-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 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, 5
	Learning rate	$0.0 \le \eta \le 0.01$
	Moment coefficient	$0.0 \le \alpha \le 0.001$

Table 2 shows the performance for RS-based NFNs using genetic optimization. From the table 2, we know that the optimized RS-based NFNs have the good results. We select the network that has two MFs for each input variable and modified quadratic inference engine. This network exhibits CR=99.05±0.67, PI=0.023±0.00 for training datasets and CR=99.56±0.99, PI=0.025±0.01 for testing datasets.

The proposed network is contrasted with some previously developed models; refer to Table 3. We note that the performance of the proposed model is better than several previous developed models.

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

No. of MFs	Inference (Type)	CR		PI	
		Training	Testing	Training	Testing
2	1	97.33±1.56	96.89±3.37	0.030±0.00	0.032±0.01
	2	99.43±0.52	98.67±1.22	0.023±0.00	0.025±0.00
	3	99.05±0.67	99.56±0.99	0.023±0.00	0.025±0.01
3	1	99.43±0.52	99.11±1.22	0.021±0.00	0.018±0.01
	2	99.43±0.52	99.11±1.22	0.018±0.00	0.020±0.00
	3	99.43±0.52	99.11±1.22	0.019±0.00	0.015±0.00
5	1	99.43±0.52	98.22±0.99	0.017±0.00	0.018±0.00
	2	99.05±0.67	99.11±1.22	0.016±0.00	0.019±0.00
	3	99.81±0.43	96.89±1.22	0.015±0.00	0.021±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 Conclusions

This paper introduced neuro-fuzzy networks based on fuzzy respective input space 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 respective space-based rules. 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. Finally it could be possible to apply to many fields.

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