Neural Processing Letters **17:** 149–164, 2003.

Extracting Interpretable Fuzzy Rules from RBF Networks

YAOCHU JIN and BERNHARD SENDHOFF
Honda Research Institute Europe, 63073 Offenbach/Main, Germany.
e-mail: {yaochu.jin; bernard.sendhoff}@honda-ri.de

Abstract. Radial basis function networks and fuzzy rule systems are functionally equivalent under some mild conditions. Therefore, the learning algorithms developed in the field of artificial neural networks can be used to adapt the parameters of fuzzy systems. Unfortunately, after the neural network learning, the structure of the original fuzzy system is changed and interpretability, which is considered to be one of the most important features of fuzzy systems, is usually impaired. This Letter discusses the differences between RBF networks and interpretable fuzzy systems. Based on these discussions, a method for extracting interpretable fuzzy rules from RBF networks is suggested. Simulation examples are given to embody the idea of this paper.

1. Introduction

Jang and Sun [4] have shown that radial basis function (RBF) networks and a simplified class of fuzzy systems are functionally equivalent under some mild conditions. This functional equivalence has made it possible to combine the features of these two systems, which has been developed into a powerful type of neurofuzzy systems [5]. However, a fuzzy system that has been trained using learning algorithms may lose its interpretability or transparency, which is one of the most important features of fuzzy systems.

In this Letter, the relationship between RBF networks and fuzzy systems is re-examined. We emphasize the differences rathr than the equivalence between these two models. It is argued that the essential difference between RBF networks and fuzzy systems is the interpretability, which enables fuzzy systems to be easily comprehensible. Based on the discussions on their relationships, a method to extract interpretable fuzzy rules from trained RBF networks using regularization techniques is proposed. Simulation studies are carried out on two test problems and an example from process modeling to show how fuzzy rules with good interpretability can be extracted from RBF networks.

It should be mentioned that the method proposed in this work is quite different from the existing techniques for rule extraction from neural networks [14]. For example, a wide class of existing methods extract symbolic rules from multiplayer perceptrons [15]. Although fuzzy rule extraction has been studied in [2], the work is mainly based on a special feedforward neural network structure. In our work, fuzzy rules are

extracted from RBF networks by investigating the difference between interpretable fuzzy rules and RBF networks. Since fuzzy rules and RBF networks are mathematically equivalent, emphasis of this work has been laid on interpretability, which is most critical for the semantic meanings of fuzzy rules.

2. Relations Between RBF Networks and Fuzzy Systems

In this section, we first briefly review the functional equivalence between RBF networks and a class of fuzzy systems. A definition of interpretability of fuzzy systems is then proposed. Finally, the conditions on converting an RBF network to a fuzzy system are discussed.

2.1. FUNCTIONAL EQUIVALENCE BETWEEN RBF NETWORKS AND FUZZY SYSTEMS

Radial basis function neural networks are one of the most important models of artificial neural networks. They were proposed in [10] and [11] among others in the context of different research motivations. Generally, an RBF network with a single output can be expressed as follows:

$$y = \sum_{j=1}^{N} f_j \varphi_j \left(\frac{\| \boldsymbol{x} - \boldsymbol{\mu}_j \|}{\boldsymbol{\sigma}_j} \right) \tag{1}$$

where $\varphi_j(\cdot)$ is called the jth radial-basis function or the jth receptive field unit, $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are the center and the variance vectors of the jth basis function, and f_j is the weight or strength of the jth receptive field unit. If the basis functions of the RBF network are Gaussian functions and the output is normalized, an RBF network can be described as:

$$y = \frac{\sum_{j=1}^{N} f_j \prod_{i=1}^{m_j} \exp\left[-\left(\frac{x_i - \mu_{ij}}{\sigma_{ij}} \right)^2 \right]}{\sum_{j=1}^{N} \prod_{i=1}^{m_j} \exp\left[-\left(\frac{x_i - \mu_{ij}}{\sigma_{ij}} \right)^2 \right]} \tag{2}$$

where $1 \leqslant m_j \leqslant M$ is the dimension of the jth basis function, M is the dimension of the input space, and N is the number of hidden nodes.

Several supervised and unsupervised learning methods as well as evolutionary computation based optimization algorithms have been developed to find optimal values of the neural network parameters. Almost all of these algorithms can be applied to the neurofuzzy systems.

The theory of fuzzy sets and fuzzy inference systems [17] originated from a completely different research field. Fuzzy inference systems are composed of a set of if-then rules. A Sugeno-Takagi fuzzy model has the following form of fuzzy rules [13]:

$$\begin{aligned} R_j : &\text{ If } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \text{ and } \ldots \text{ and } x_M \text{ is } A_{Mj}, \\ &\text{Then } y = g_j(x_1, x_2, \ldots, x_M), \end{aligned} \tag{3}$$

where $g_j(\cdot)$ is a crisp function of x_i. The overall output of the fuzzy model can be obtained by:

$$y = \frac{\sum_{j=1}^{N} [g_j(\cdot) T_{i=1}^{m_j} \varphi_{ij}(x_i)]}{\sum_{j=1}^{N} T_{i=1}^{m_j} \varphi_{ij}(x_i)} \tag{4}$$

where $1 \leqslant m_j \leqslant M$ is the number of input variables that appear in the rule premise, M is the number of inputs, $\varphi_{ij}(x_i)$ is the membership function for fuzzy set A_{ij} and T is a t-norm for fuzzy conjunction. It is noticed that the RBF network expressed in Equation (2) and the fuzzy systems described by Equation (4) are mathematically equivalent provided that multiplication is used for the t-norm in fuzzy systems and both systems use Gaussian basis functions. Here, we will not re-state the restrictions proposed in [4], however, we will show that although these restrictions do result in the mathematical equivalence between RBF networks and fuzzy systems, they do not guarantee the equivalence of the two models in terms of the semantic meanings.

2.2. INTERPRETABILITY CONDITIONS FOR FUZZY SYSTEMS

The main difference between radial-basis-function networks and fuzzy systems is the interpretability. Generally speaking, neural networks are considered to be black-boxes and therefore no interpretability conditions are imposed on conventional neural systems. On the other hand, fuzzy systems are supposed to be inherently comprehensible, especially when the fuzzy rules are obtained from human experts. However, interpretability of fuzzy systems cannot be guaranteed during data based rule generation and adaptation. For this reason, interpretability of fuzzy systems has received increasing attention in the recent years [3, 6, 7, 9, 12, 16]. In the following, we propose some major conditions a fuzzy system should fulfill to be interpretable:

1. The fuzzy partitioning of all variables in the fuzzy system are both complete and well distinguishable. In addition, the number of fuzzy subsets in a fuzzy partitioning is limited.

Remarks. The completeness and distinguishability condition makes it possible to assign a clear physical meaning to each fuzzy subset in a fuzzy partitioning. Therefore, it is the most important aspect for the interpretability of fuzzy systems. Usually, this also leads to a small number of fuzzy subsets. A quantitative description of the completeness and distinguishability condition can be expressed as follows:

$$\delta_1 \leqslant S(A_i, A_{i+1}) \leqslant \delta_2 \tag{5}$$

where, A_i and A_{i+1} are two arbitrary neighboring fuzzy subsets in a fuzzy partitioning, $S(A_i, A_{i+1})$ is a fuzzy similarity measure between them, δ_1 and δ_2 are the lower and upper thresholds of the fuzzy similarity measure, where a positive δ_1 guarantees the

completeness and a δ_2 that is sufficiently smaller than one maintains good distinguishability [8].

2. Fuzzy rules in the rule base are consistent with each other and consistent with the prior knowledge, if available.

Remarks. Although the performance of fuzzy systems is believed to be insensitive to the inconsistency of the fuzzy rules to a certain degree, seriously inconsistent fuzzy rules will undoubtedly result in incomprehensible fuzzy systems. We argue that fuzzy rules are inconsistent in the following situations:

- The condition parts are the same, but the consequent parts are completely different. For example,
  R1: If $x1$ is $A1$ and $x2$ is $A2$, then y is Positive Large;
  R2: If $x1$ is $A1$ and $x2$ is $A2$, then y is Negative Large
- Although the condition parts are seemingly different, they are physically the same. However, the consequents of the rules are totally different.
  R1: If $x1$ is $A1$ and $x2$ is $A2$, then y is Positive Large;
  R2: If $x1$ is $A1$ and $x3$ is $A3$, then y is Negative Large
  Although '$x2$ is $A2$' and '$x3$ is $A3$' appear to be different conditions, they might imply the same situation in some cases. For example, for a chemical reactor, a statement 'temperature is high' may imply 'conversion rate is high'.
- The conditions in a rule premise are contradictory, e.g. 'If the sun is bright and the rain is heavy'.
- The actions in the rule consequent part are contradictory. For example, in the rule 'If x is A then y is B and z is C.' However, 'y is B and z is C' cannot happen simultaneously.

3. The number of variables that appear in the premise part of the fuzzy rules should be as small as possible. In addition, the number of fuzzy rules in the rule base should also be small. These two aspects deal with the compactness of the rule structure.

The interpretability conditions impose implicit constraints on the parameters of fuzzy systems. While interpretability is one of the most important feature of fuzzy systems, there are generally no interpretability requirements on the parameters of RBF networks. In this sense, RBF networks and fuzzy systems are not the same even if they are functionally equivalent.

2.3. CONVERSION OF AN RBFN INTO FUZZY RULES

The central point in converting an RBFN into a Sugeno fuzzy model is to ensure that the extracted fuzzy rules are interpretable, i.e. easy to understand. In order to convert an RBF network to an interpretable fuzzy rule system, the following conditions should be satisfied:

1. The basis functions of the RBF network are Gaussian functions.
2. The output of the RBF network is normalized.
3. The basis functions within each receptive field unit of the RBF network are allowed to have different variances.
4. Certain numbers of basis functions for the same input variable but within different receptive field units should share a mutual center and a mutual variance. If Mamdani fuzzy rules are to be extracted, then some of the output weights of the RBF network should also share.

Conditions 3 and 4 are necessary for good interpretability of the extracted fuzzy system. Without condition 3, the first part of condition 4 cannot be realized. As the most important condition, condition 4 requires that some weights in the RBF network should share. For the sake of simplicity, we use 'weights' to refer to both the parameters of basis functions (centers and variances) and the output weights of RBF networks in the following text. The weight sharing condition ensures a good distinguishability for the fuzzy partitioning, which is the most essential feature for the interpretability of fuzzy systems. Given a fuzzy system in Figure 1, the RBF network that is directly converted from the fuzzy system is illustrated in Figure 2. If we take a closer look at the RBF network in Figure 2, we notice that some of the basis functions are identical. On the other hand, if a fully connected RBF network with N hidden nodes is directly converted into a fuzzy system, each variable of the fuzzy system will have N sub-fuzzy sets. If N is large (e.g., $N > 10$), it will be difficult to understand the fuzzy system. However, we find it difficult to define the weight sharing condition explicitly because we cannot require that the structure of the extracted fuzzy system should be the same as its original structure, therefore, we do not know beforehand which weights should share. Additionally, the completeness

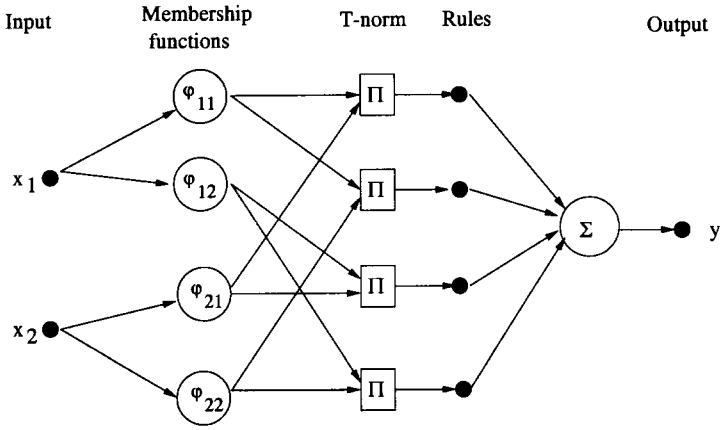

Figure 1. A fuzzy system.

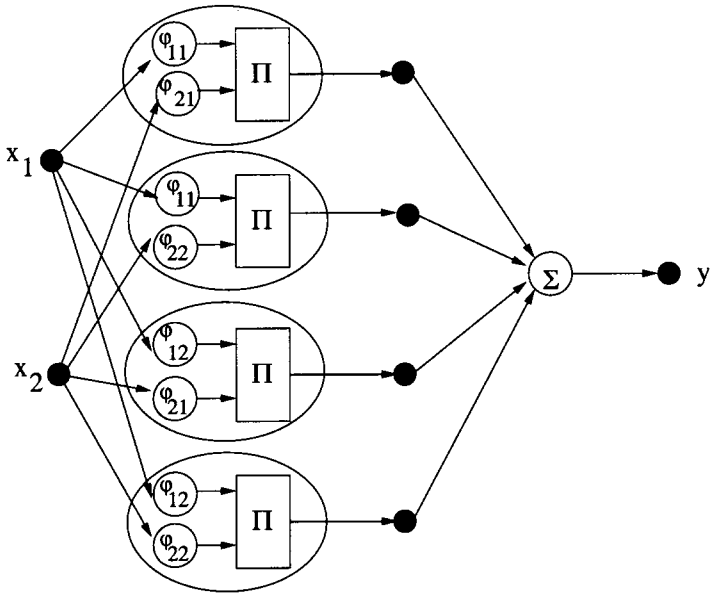

Figure 2. The RBF network converted from the fuzzy system.

of the fuzzy partitioning should be considered together with the weight sharing condition in the course of rule extraction. These problems will be treated in detail in the next section.

Note that the consistency condition is not considered here, because we suppose that it has been taken into account in generating the initial fuzzy system [8]. Nevertheless, measures can be taken to prevent the rule extraction algorithm from generating seriously inconsistent rules, namely, rules with the same premise but different consequents.

3. Fuzzy Rule Extraction from RBF Networks

As we have discussed in the last section, to extract interpretable fuzzy rules from an RBF network, the number of basis in the RBF network should be kept small and there should be no very similar basis functions, which means that some of the weights in the RBF network should share. Thus, extracting interpretable fuzzy rules from RBF networks can be treated as fine training of the RBF network with regularization [1] such that similar weights in the RBF network share the same value.

Before the regularization can be applied, it is necessary to specify, which weights should be identical before the weight sharing algorithm can be employed. Thus, the first step toward rule extraction from RBF networks is to determine which weights, including the parameters of basis functions and the output weights, should share.

3.1. SPECIFICATION OF SHARED WEIGHTS

Since the rule structure of the fuzzy system is unknown, we do not know in advance, which weights should share. However, it is straightforward to imagine that the weights that are going to share a same value should be similar before the regularization is applied. Therefore, we have to first identify similar weights using a distance measure. Currently, several distance measures (or similarity measures) are available. Among them, Euclidean distance is very simple and has been widely used. The Euclidean distance between two membership functions (basis functions) $\varphi_i(\mu_i, \sigma_i)$ and $\varphi_j(\mu_j, \sigma_j)$, where μ_i, μ_j are centers and σ_i, σ_j are the variances can be defined as:

$$\mathrm{d}(\varphi_i, \varphi_j) = \sqrt{(\mu_i - \mu_j)^2 + (\sigma_i - \sigma_j)^2}. \tag{6}$$

For a Gaussian basis function or membership function, each has two elements, namely, the center and the variance. For the output weights, each vector has only one element. Suppose that a given input variable x_i has K different basis functions $\varphi_{ij}(j = 1, 2, \ldots, K)$ with the center μ_{ij} and the variance σ_{ij}, the procedure to determine similar basis functions can be described as follows:

1. List the basis functions (φ_{ij}) in the order of increasing sequence with regard to their center values. Let U_{ik} be the kth set for x_i containing similar basis functions. Two basis functions are considered similar if the distance between them is less than d_i, where d_i is a prescribed threshold. The regularization algorithm will drive the similar basis functions in set U_{ik} to share the same parameters. Put φ_{ij} to U_{ik}, let $j, k = 1$, $\varphi_i^0 = \varphi_{i1}$.
2. If $\mathrm{d}(\varphi_i^0, \varphi_{ij+1}) < d_i$, put φ_{ij+1} to U_{ik}; else $k = k + 1$, put φ_{ij+1} to U_{ik} and let $\varphi_i^0 = \varphi_{ij+1}$.
3. $j = j + 1$, if $j \leqslant K$, go to step 2; else stop.

The prescribed distance threshold d_i is very important because it determines both the distinguishability and the completeness of the fuzzy partitions. Suppose $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ are the averaged center and variance of the basis functions in U_{ik}, then the fuzzy partition constructed by $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ should satisfy the completeness and distinguishability condition described in Equation (5).

In practice, we find that the performance of the extracted fuzzy system is not satisfactory if we simply choose $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ to be the values to share by the basis functions in U_{ik}. In other words, a direct merge of similar basis functions will degrade the performance seriously. In the following subsection, we will introduce an adaptive weight sharing method to improve the performance of the extracted fuzzy system.

3.2. ADAPTIVE WEIGHT SHARING

We do not directly require that the weights in the same set should be identical. Instead, we realize weight sharing by regularizing the RBF network. In

the following, we present the weight sharing algorithm with regard to the RBF model described in Equation (2).

Regularization of neural networks is realized by adding an extra term to the conventional cost function:

$$J = E + \lambda \cdot \Omega, \tag{7}$$

where E is the conventional cost function, λ is the regularization coefficient ($0 \leqslant \lambda < 1$), and Ω is the regularization term for weight sharing.

The cost function E is expressed as:

$$E = \frac{1}{2}(y - y^t)^2, \tag{8}$$

where, y is the output of the neural network and y^t is the target value. In the following, we assume that Sugeno fuzzy rules are to be extracted, that is to say, the output weights of the RBF network are not regularized. In this case, the regularization term Ω has the following form:

$$\Omega = \frac{1}{2}\sum_i \sum_k \sum_{\varphi_{ij} \in U_{ik}} (\mu_{ij} - \bar{\mu}_{ik})^2 + \frac{1}{2}\sum_i \sum_k \sum_{\varphi_{ij} \in U_{ik}} (\sigma_{ij} - \bar{\sigma}_{ik})^2 \tag{9}$$

where $\bar{\mu}_{ik}$ and $\bar{\sigma}_{ik}$ are the center and variance to be shared by the basis functions φ_{ij} in set U_{ik}. Empirically, the averaged center $\hat{\mu}_{ik}$ and variance $\hat{\sigma}_{ik}$ of set U_{ik} are used as the initial values for $\bar{\mu}_{ik}$ and $\bar{\sigma}_{ik}$. The gradients of J are:

$$\left.\frac{\partial J}{\partial \mu_{ij}}\right|_{\varphi_{ij} \in U_{ik}} = \frac{\partial E}{\partial \mu_{ij}} + \lambda(\mu_{ij} - \bar{\mu}_{ik}) \tag{10}$$

$$\left.\frac{\partial J}{\partial \sigma_{ij}}\right|_{\varphi_{ij} \in U_{ik}} = \frac{\partial E}{\partial \sigma_{ij}} + \lambda(\sigma_{ij} - \bar{\sigma}_{ik}) \tag{11}$$

$$\frac{\partial J}{\partial \bar{\mu}_{ik}} = -\lambda \sum_{\varphi_{ij} \in U_{ik}} (\mu_{ij} - \bar{\mu}_{ik}) \tag{12}$$

$$\frac{\partial J}{\partial \bar{\sigma}_{ik}} = -\lambda \sum_{\varphi_{ij} \in U_{ik}} (\sigma_{ij} - \bar{\sigma}_{ik}), \tag{13}$$

where

$$\frac{\partial E}{\partial \mu_{ij}} = (y - y^t)(f_j - y)(x_i - \mu_{ij}) \prod_{i=1}^{m_j} \left[\exp\left(-\frac{(x_i - \mu_{ij})^2}{\sigma_{ij}^2} \right) \right] \Big/ \sigma_{ij}^2 \tag{14}$$

$$\frac{\partial E}{\partial \sigma_{ij}} = (y - y^t)(f_j - y)(x_i - \mu_{ij})^2 \prod_{i=1}^{m_j} \left[\exp\left(-\frac{(x_i - \mu_{ij})^2}{\sigma_{ij}^2} \right) \right] \Big/ \sigma_{ij}^3 \tag{15}$$

It should be pointed out that the specification algorithm introduced in the last subsection must be applied in each iteration of the neural network learning to improve

the performance of the extracted fuzzy system. Recall that the weights of the RBF network are determined not only by the regularization term, but also by the conventional error term. That is to say, it is possible that a basis function φ_{ij} that is originally put in U_{ik} will be classified to another set after an iteration of learning. By specifying the shared weights during network learning, a more optimal rule structure will be obtained. It is also necessary to check the completeness condition on the fuzzy partitioning constructed by $(\bar{\mu}_{ik}, \bar{\sigma}_{ik})$ during learning. The incompleteness of the fuzzy partitioning can be avoided by temporarily stopping the adaptation of the shared parameters $(\bar{\mu}_{ik}, \bar{\sigma}_{ik})$.

If Mamdani fuzzy rules are to be extracted, a similar specification of the shared weights should be carried out on the output weights and an extra term should be added to Ω. Then, a regularized learning algorithm for the output weights can also be derived. The resulting weights can be explained as the centers of the membership functions for the output variable.

4. Examples

4.1. THE MACKEY–GLASS SYSTEM

In this subsection, simulation studies on the modeling of the Mackey–Glass time series are carried out to show the feasibility of the proposed method. The Mackey–Glass time series is described by:

$$\dot{x} = \frac{ax(t-\tau)}{1 + x^b(t-\tau)} - cx(t), \tag{16}$$

where $\tau = 30$, $a = 0.2$, $b = 10$, and $c = 0.1$. One thousand data samples are used in the simulation, 500 samples for training and the other 500 samples for test. The goal is to predict $x(t)$ using $x(t-1)$, $x(t-2)$ and $x(t-3)$. That is to say, the system has three inputs and one output.

An RBF neural network with 6 hidden nodes is converted from a fuzzy system that is generated using the training data [7]. After training the RBF network, the mean absolute errors on the training and test data are about 0.008. The basis functions of the $x(t-2)$ and $x(t-1)$ are shown in Figure 3. $x(t-3)$ has only one basis function, and therefore, it is not shown in the figure.

The weight specification process is then applied on the basis functions of $x(t-2)$ and $x(t-1)$. As a result, the basis functions of $x(t-2)$ are classified into two groups and those of $x(t-1)$ are divided into three groups. Using the adaptive weight sharing algorithm, five fuzzy rules are obtained. The mean absolute errors for training and test are both about 0.016, which are larger than those of the RBF network. The new basis functions of $x(t-3)$, $x(t-2)$ and $x(t-1)$, which are now called membership functions, are shown in Figure 4.

According to the distribution of the membership functions, a linguistic term is assigned to each membership function of the inputs, refer to Figure 4. For example,

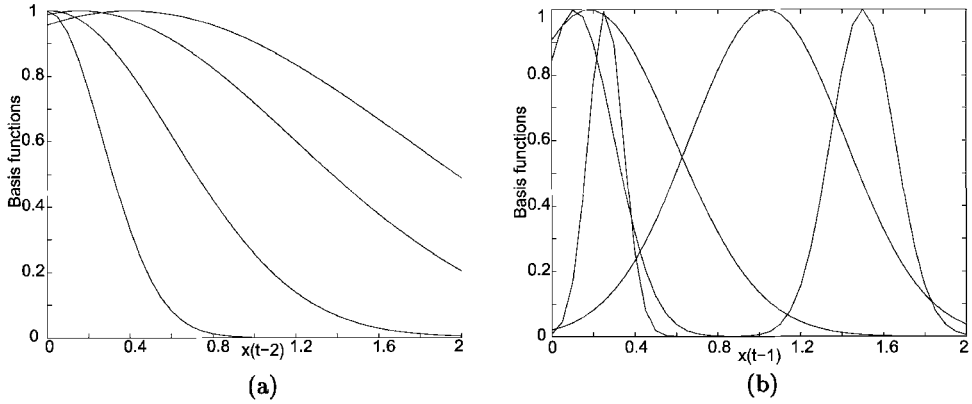

Figure 3. Basis functions of the RBF network. (a) $x(t-2)$, (b) $x(t-1)$.

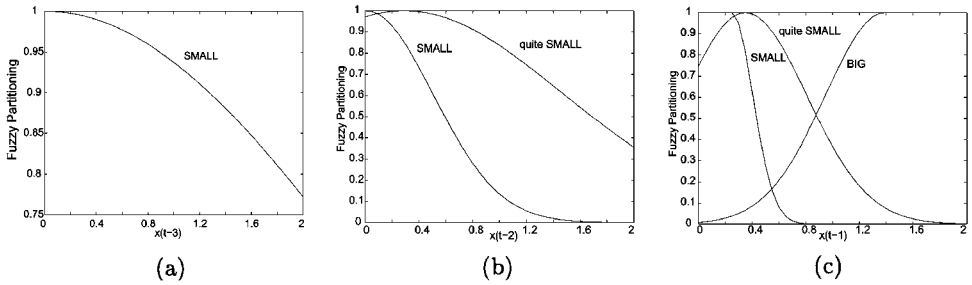

Figure 4. Membership functions of the fuzzy system. (a) $x(t-3)$, (b) $x(t-2)$, and (c) $x(t-1)$.

for $x(t-2)$, 'SMALL' can be assigned to its membership function $(0.01, 0.7)$ and 'quite SMALL' can be assigned to the membership function $(0.29, 1.68)$. In this way, the following interpretable fuzzy rules for the Mackey-Glass system can be extracted:

- If $x(t-2)$ is SMALL and $x(t-1)$ is BIG, then $x(t)$ is BIG;
- If $x(t-1)$ is quite SMALL, then $x(t)$ is SMALL;
- If $x(t-2)$ is quite SMALL and $x(t-1)$ is SMALL, then $x(t)$ is SMALL;
- If $x(t-1)$ is BIG, then $x(t)$ is BIG;
- If $x(t-3)$ is SMALL and $x(t-2)$ is SMALL, then $x(t)$ is BIG.

4.2. THE LORENZ SYSTEM

The Lorenz system studied in this paper is described by the following differential equations:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -y^2 - z^2 - a(x - F) \tag{17}$$

$$\frac{dy}{dt} = xy - bxz - y + G \tag{18}$$

$$\frac{dz}{dt} = bxy + xz - z \tag{19}$$

where $a = 0.25$, $b = 4.0$, $F = 8.0$ and $G = 1.0$. In our simulation, we predict $x(t)$ from $x(t-1)$, $y(t-1)$ and $z(t-1)$. 2000 data pairs are generated using the fourth order Runge–Kutta method with a step length of 0.05, where 1000 pairs of data are used for training and the other 1000 for test.

An initial fuzzy system is generated using the evolutionary algorithm based method. Consequently, we convert this fuzzy system to an RBF neural network and continue to train it with the conventional gradient method. During the network training, the inactive field units are deleted. Thus, after the learning algorithm converges, we obtain an RBF network with 5 receptive field units. Although the approximation performance has been improved significantly, most of its fuzzy subsets (basis functions) are hard to distinguish and such a fuzzy system is not well interpretable, see Figure 5.

The algorithm for extracting fuzzy rules is then implemented. The performance of the extracted fuzzy system is a little worse than that of the RBF network, interpretability of the extracted fuzzy system is much better: the fuzzy partitions are both complete and well distinguishable, see Figure 6, and the number of fuzzy rules is also reduced.

With an interpretable fuzzy system at hand, knowledge about the system can be acquired. In the fuzzy model, $x(t-1)$ has 4 fuzzy subsets, while $y(t-1)$ has 2 subsets and $z(t-1)$ has only one subset. It is straightforward to see that in this model, $x(t)$ depends much more on $x(t-1)$ than $y(t-1)$ or $z(t-1)$. According to the distribution of the membership functions, a proper linguistic term can be assigned to each fuzzy set (refer to Figure 6). For input $x(t-1)$, '**Negative Small** (**NS**)' can be assigned to the membership function $(-0.49, 1.69)$, '**Positive Small** (**PS**)' to (0.37, 1.24), '**Positive Middle** (**PM**)' to $(1.01, 0.80)$ and '**Positive Large** (**PL**)' to (2.01, 1.49). Similarly, for $y(t-1)$, '**Negative Small** (**NS**)' is assigned to $(-0.35, 4.11)$,

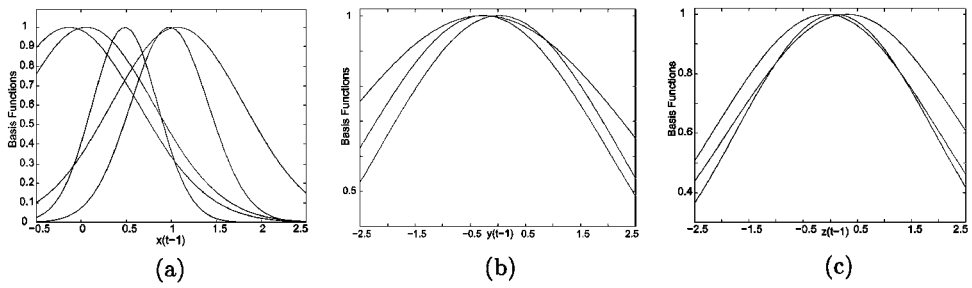

Figure 5. The basis functions of the trained RBF network: (a) $x(t-1)$, (b) $y(t-1)$ and (c) $z(t-1)$.

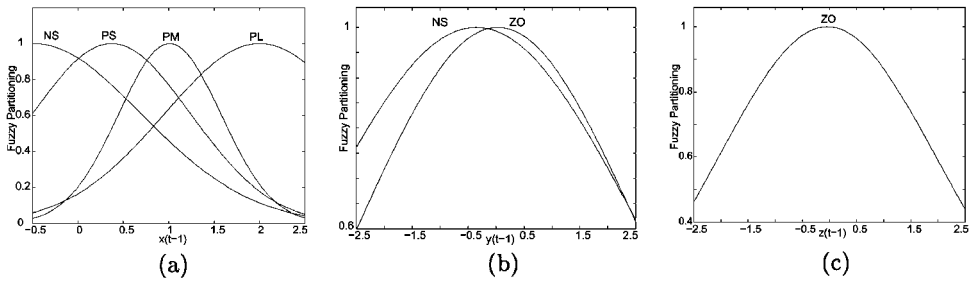

Figure 6. The membership functions of the extracted fuzzy system: (a) $x(t-1)$, (b) $y(t-1)$ and (c) $z(t-1)$.

'**Zero** (**ZO**)' to $(0.03, 3.53)$; for $z(t-1)$, '**Zero** (**ZO**)' is assigned to $(-0.04, 2.79)$. In this way, some intelligible knowledge about the Lorenz system in terms of interpretable fuzzy rules is acquired.

- If $x(t-1)$ is **Positive Small**, Then $x(t)$ is **Negative Small**
- If $x(t-1)$ is **Positive Large** and $y(t-1)$ is **Zero** and $z(t-1)$ is **Zero**, Then $x(t)$ is **Positive Large**
- If $x(t-1)$ is **Negative Small** and $z(t-1)$ is **Zero**, Then $x(t)$ is **Negative Small**
- If $x(t-1)$ is **Positive Middle** and $y(t-1)$ is **Negative Small**, Then $x(t)$ is **Positive Middle**

We mentioned that the approximation accuracy of the extracted fuzzy system is a little worse than that of the RBF neural network. This implies that better interpretability may lead to lower approximation accuracy, especially when the fuzzy partitions of the fuzzy system are required to be well distinguishable.

4.3. PROCESS MODELING

The data used in the following simulation are generated to simulate an industrial process. We use this example because it is a high-dimensional system with deliberately added biased noises. In this simulated system, there are 11 inputs and one output with $20,000$ data for training and $80,000$ data for test.

An RBF network with 27 hidden nodes is obtained using the training data. The RMS errors on training and test data are 0.189 and 0.207. Although the performance is satisfying, the RBF model is hard to understand when we take a look at the basis functions of, particularly 6 of the 11 inputs, see Figures 7 and 8.

To extract interpretable fuzzy rules from the RBF network, the proposed algorithm is employed. After training, the RMS errors of the fuzzy system on the training and test data become 0.191 and 0.213 respectively, which have slightly increased as expected. What is very encouraging is that the number of fuzzy subsets in the fuzzy partitions are significantly reduced and the distinguishability is greatly improved, see Figures 9 and 10. With these well distinguishable fuzzy partitions, it is possible to

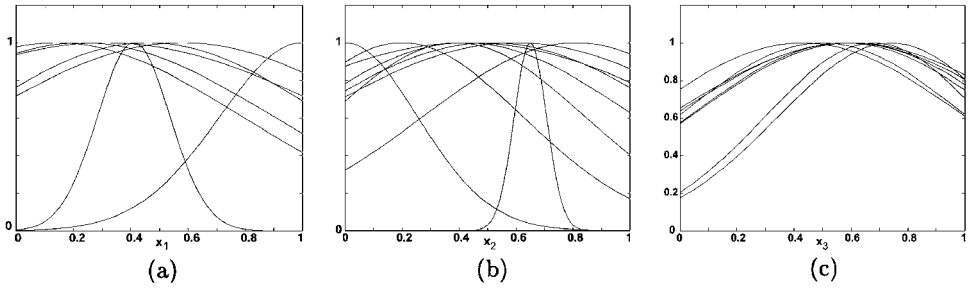

Figure 7. The basis functions of the RBF network: (a) x_1, (b) x_2 and (c) x_3.

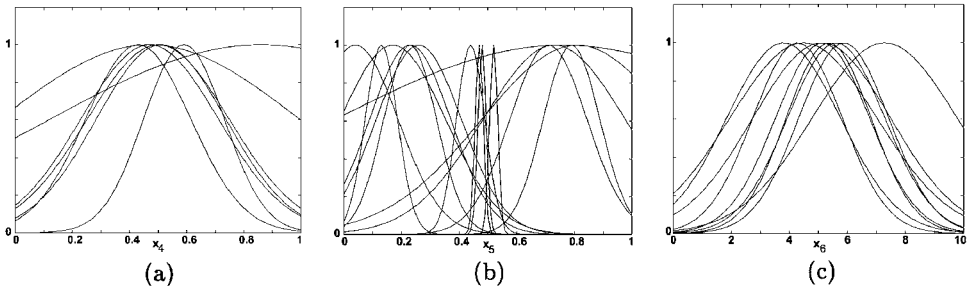

Figure 8. The basis functions of the RBF network: (a) x_4, (b) x_5 and (c) x_6.

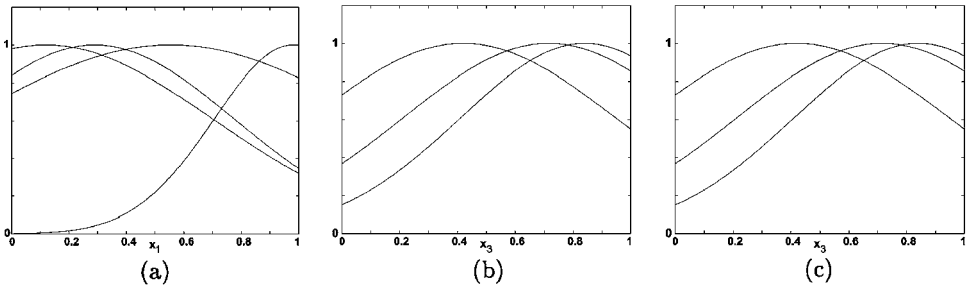

Figure 9. The membership functions of the fuzzy system: (a) x_1, (b) x_2 and (c) x_3.

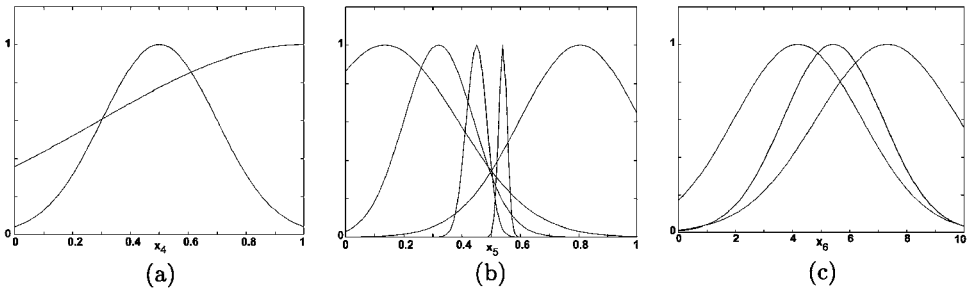

Figure 10. The membership functions of the fuzzy system: (a) x_4, (b) x_5 and (c) x_6.

associate a linguistic term to each fuzzy subset and thus interpretable fuzzy rules can be obtained. This demonstrates that the proposed algorithm is effective for high-dimensional systems.

The linguistic terms for input variables are listed as follows:

x_1: {*Very Small (VS), Small (S), Medium (M), Large (L)*};
x_2: {*Small (S), Medium (M), Large (L)*};
x_3: {*Small (S), Medium (M), Large (L)*};
x_4: {*Medium (M), Large (L)*};
x_5: {*Very Small (VS), Small (S), Medium (M), Large (L), Very Large (VL)*};
x_6: {*Small (S), Medium (M), Large (L)*};
x_7: {*Medium (M), Large (L)*};
x_8: {*Negative Large (NL), Negative Medium (NM), Negative Small (NS)*};
x_9: {*Very Small (VS), Small (S), Medium (M), Large (L), Very Large (VL)*};
x_{10}: {*Large (L)*};
x_{11}: {*Small (S), Medium (M), Large (L)*}.

Notice that the maximal number of linguistic terms for a variable is 5, and most variables have 3 to 4 linguistic terms, which is ideal for good interpretability of fuzzy systems.

The 27 fuzzy rules are:

1. IF x_1 is VS, x_2 is M, x_3 is S, x_4 is M, x_5 is VS, x_8 is NM, THEN $y = -0.434$;
2. IF x_1 is M, x_3 is M, x_5 is L, x_8 is NM, x_{10} is L, THEN $y = 4.558$;
3. IF x_3 is M, x_4 is L, x_5 is VL, x_9 is S, x_{10} is L, THEN $y = 4.012$;
4. IF x_3 is M, x_5 is VS, x_8 is NL, x_9 is M, x_{10} is L, THEN $y = 1.482$;
5. IF x_1 is S, x_2 is M, x_4 is M, x_5 is VS, x_6 is S, THEN $y = -0.497$;
6. IF x_2 is M, x_5 is M, x_6 is M, x_9 is L, x_{10} is L, THEN $y = 1.221$;
7. IF x_1 is M, x_4 is M, x_5 is M, x_8 is NM, x_{10} is L, THEN $y = 0.680$;
8. IF x_1 is M, x_3 is M, x_4 is M, x_5 is M, THEN $y = 0.268$;
9. IF x_4 is M, x_5 is VS, x_6 is M, x_9 is L, THEN $y = 0.849$;
10. IF x_2 is S, x_5 is VS, x_6 is S, x_{10} is L, THEN $y = 1.324$;
11. IF x_2 is L, x_5 is VL, x_6 is L, x_{10} is L, THEN $y = 4.428$;
12. IF x_3 is S, x_4 is M, x_5 is S, x_{10} is L, THEN $y = 1.340$;
13. IF x_5 is VS, x_6 is M, x_8 is NM, x_{10} is L, THEN $y = 2.720$;
14. IF x_1 is L, x_2 is L, x_5 is VL, x_6 is M, THEN $y = 3.893$;
15. IF x_1 is VS, x_2 is S, x_8 is NS, x_9 is VL, THEN $y = 0.770$;
16. IF x_2 is M, x_5 is VS, x_9 is VL, THEN $y = 0.546$;
17. IF x_4 is M, x_5 is VL, x_{10} is L, THEN $y = 2.102$;
18. IF x_1 is VS, x_5 is VS, x_8 is NM, THEN $y = 0.414$;
19. IF x_2 is L, x_3 is L, x_{10} is L, THEN $y = 5.114$;
20. IF x_3 is M, x_5 is L, x_8 is NS, THEN $y = 4.265$;
21. IF x_7 is L, x_9 is L, THEN $y = 2.292$;

22. IF x_5 is L, x_6 is S, THEN $y = 3.758$;
23. IF x_7 is M, x_{11} is L, THEN $y = 2.531$;
24. IF x_6 is M, x_9 is VS, THEN $y = 3.867$;
25. IF x_9 is VL, x_{11} is M, THEN $y = 2.231$;
26. IF x_9 is M, THEN $y = 2.788$;
27. IF x_{11} is S, THEN $y = 2.411$.

5. Conclusions

The relationships between RBF networks and interpretable fuzzy systems have been discussed. A definition for an interpretable fuzzy system has also been suggested. Conditions for converting RBF networks to fuzzy systems have been proposed. In order to extract interpretable fuzzy rules from an RBF network, an adaptive weight sharing algorithm has been introduced. We have shown that an RBF network and a fuzzy system are not fully equivlent in terms of their semantic meanings and that the extraction of interpretable fuzzy rules from RBF networks is both important and feasible for gaining a deeper insight into the logical structure of the system to be approximated. Simulation studies have been carried out on two test problems and one high-dimensional system to demonstrate the proposed method.

Acknowledgements

Part of this work was done when the authors were with the Institut für Neuroinformatik, Ruhr-Universität Bochum. This work was supported in part by German Ministry of the Research under the grant AENEAS.

References

1. Bishop, C. M.: *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.
2. Hayashi, Y.: A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosis, *Advances in Neural Information Processing Systems*, **3** (1990), 578–584.
3. Ishbuchi, H. Nakashima, T. and Murada, T.: Multi-objective optimization in linguistic rule extraction from numerical data, In: *Proceedings of 1st International Conference on Evolutionary Multi-criterion Optimization*, pp. 588–602, 2001.
4. Jang, J.-S. R. and Sun, C.-T.: Functional equivalence between radial basis functions and fuzzy inference systems, *IEEE Trans. on Neural Networks*, **4** (1993), 156–158.
5. Jang, J.-S. R. and Sun, C.-T.: Neuro-fuzzy modeling and control, *Proceedings of the IEEE*, **83** (1995), 378–405.
6. Jin, Y.: Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement, *IEEE Transactions on Fuzzy Systems*, **8**(2) (2000), 212–221.
7. Jin, Y., von Seelen, W. and Sendhoff, B.: An approach to rule-based knowledge extraction, In: *Proceedings of IEEE Int. Conf. on Fuzzy Systems*, pp. 1188–1193, Anchorage, AL, 1998.

8. Jin, Y., von Seelen, W. and Sendhoff, B.: On generating FC^3 fuzzy rule systems from data using evolution strategies, *IEEE Trans. on Systems, Man, and Cybernetics*, **29** (1999), 829–845.
9. Lofti, A. and Tsoi, A. C.: Interpretation preservation of adaptive fuzzy inference systems, *Int. Journal of Approximating Reasoning*, 15, 1996.
10. Moody, J. and Darken, C.: Fast learning in networks of locally-tuned processing units, *Neural Computation*, **1** (1989), 181–194.
11. Powell, M.: Radial Abasis functions for multivariable interpolation: A review. In: C. Mason and M.G. Cox, (eds.), *Algorithms for Approximation*, pp. 143–167, Oxford University Press, Oxford, UK, 1987.
12. Setnes, M., Babuska R. and Verbruggen, B.: Rule-based modeling: Precision and transparency, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, **28**(1) (1998), 165–169.
13. Takagi, T. and Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. on Systems, Man, and Cybernetics*, **15** (1985), 116–132.
14. Tickle, A., Andrews, R., Golea M. and Diederich, J.: The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks, *IEEE Transactions on Neural Networks*, **9**(6) (1998), 1057–1068.
15. Towell, G. and Shavlik, J.: Extracting refined rules from knowledge-based neural networks, *Machine Learning*, **13** (1993), 71–101.
16. Valente de Oliveira, J.: On the optimization of fuzzy systems using bio-inspired strategies, In: *IEEE Proceedings of International Conference on Fuzzy Systems*, IEEE Press, pp. 1129–1134, Anchorage, Alaska, 1998.
17. Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. on Systems, Man, and Cybernetics*, **3** (1973), 18–44.