Institut für Neuroinformatik

Ruhr-Universität Bochum

Internal Report 2000–02

Extracting Interpretable Fuzzy Rules from RBF Neural Networks

by

Yaochu Jin, Werner von Seelen, Bernhard Sendhoff

Ruhr-Universität Bochum
Institut für Neuroinformatik
44780 Bochum

IR-INI 2000–02
January 2000
ISSN 0943-2752

Extracting Interpretable Fuzzy Rules from RBF Neural Networks

Yaochu Jin[1,*], Werner von Seelen[2], Bernhard Sendhoff [1,2]
[1]Future Technology Research, Honda R&D Europe (Germany), Germany
[2]Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany

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 damaged and interpretability, which is considered to be the most important feature of fuzzy systems, is usually impaired. This paper investigates the differences between RBF neural networks and well interpretable fuzzy systems and discusses the convertibility conditions between them. Inspired by these discussions, a method to extract well interpretable fuzzy rules from trained RBF networks is proposed by introducing an adaptive weight sharing algorithm. We show that the extraction of interpretable fuzzy rules from RBF neural networks provides an effective way to gain a deeper insight into the system. Simulation studies are carried out on the Lorenz system to embody the idea of this paper.

1 Introduction

It is found in [1] 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 models systematically and has initiated a new model called neurofuzzy system [2]. However, we believe that such neurofuzzy models will be more interesting if they have the main features of both neural networks and fuzzy inference systems. That is to say, they should have good learning capability, strong representational power as well as good interpretability (transparency). Among these features, learning capability and representational power are always guaranteed since they are the inherent features of neurofuzzy systems. On the contrary, the interpretability of neurofuzzy systems is not inherent and relatively little attention has been paid to it. In fact, the interpretability of neurofuzzy systems is usually impaired due to an excessive endeavor for the flexibility of neurofuzzy systems.

In this paper, the equivalence between RBF networks and fuzzy systems is re-examined. We find that the conditions for converting an RBF neural network to a fuzzy system are different from those for converting a fuzzy system to an RBF neural network. In other words, the conditions for such conversions should be considered separately. We emphasize the differences rather than the equivalence between these two models. It is argued that the essential difference between RBF neural networks and fuzzy systems is the interpretability, which enables fuzzy systems to be easily comprehensible. After a detailed discussion on their relationships, a method to extract well interpretable fuzzy rules from trained RBF neural networks based on a neural network regularization technique is proposed. Simulation studies are carried out on the Lorenz system to show how a neurofuzzy model with good interpretability can be obtained.

Rule extraction from artificial neural networks is considered to be important due to the following reasons:

*email yaochu.jin@hre-ftr.f.rd.honda.co.jp

1. Rule extraction provides artificial neural networks with an explanation capability, which makes it possible for the user to check on the internal logic of the system.
2. Rule extraction helps to discover previously unknown dependencies in data sets and thus new knowledge about the system can be acquired.
3. It is believed that a rule system with good interpretability improves the generalization ability of neural networks where training data are insufficient.

In this sense, the method developed in this work can be seen as a general approach to rule extraction from RBF networks. Currently, techniques for extracting symbolic [3,4] and fuzzy rules [5,6] have been proposed. However, no sufficient attention has been paid to the interpretability of the extracted fuzzy systems until recently.

2 Relationships between RBF neural networks and fuzzy systems

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

A. Functional Equivalence Between RBF Neural Networks and Fuzzy Systems

Radial basis function neural networks are one of the most important models of artificial neural networks. They were proposed by Moody and Darken [7,8] and Powell [9] among others in the context of different research efforts. Generally, an RBF neural network with a single output can be expressed as follows:

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

where $\varphi_j(\cdot)$ is called the j-th radial-basis function or the j-th receptive field unit, $\vec{\mu}_j$ and $\vec{\sigma}_j$ are the center and the variance vectors of the j-th basis function, and f_j is the weight or strength of the j-th receptive field unit. The variance vector $\vec{\sigma}_j$ is understood as shown in equation (2). If the basis functions of the RBF network are Gaussian functions, we get:

$$\begin{aligned} y &= \sum_{j=1}^{N} f_j \mathrm{exp} \left[- \frac{\| \vec{x} - \vec{\mu}_j \|}{\vec{\sigma}_j} \right] \\ &= \sum_{j=1}^{N} f_j \mathrm{exp} \left[- \sum_{i=1}^{m_j} \left(\frac{x_i - \mu_{ij}}{\sigma_{ij}} \right)^2 \right] \\ &= \sum_{j=1}^{N} f_j \prod_{i=1}^{m_j} \exp \left[- \left(\frac{x_i - \mu_{ij}}{\sigma_{ij}} \right)^2 \right] \end{aligned} \tag{2}$$

where $1 \leq m_j \leq M$ is the dimension of the j-th basis function, M is the dimension of the input space. A two-input single output RBF neural network with Gaussian basis functions is schematically depicted in Fig. 1. Alternatively, the output of the RBF network can also be normalized [7]:

$$y = \frac{\sum_{j=1}^{N} f_j \varphi_j (\| \vec{x} - \vec{\mu}_j \| / \vec{\sigma}_j)}{\sum_{j=1}^{N} \varphi_j (\| \vec{x} - \vec{\mu}_j \| / \vec{\sigma}_j)} \tag{3}$$

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 [10] and fuzzy inference systems [11] originated from a completely different research field. Fuzzy inference systems are composed of a set of if-then rules. Generally, there are two fuzzy models, namely, the Mamdani fuzzy model [12] and the Sugeno-Takagi fuzzy model [13]. Consider a fuzzy model with M inputs and one output, the Mamdani fuzzy rules have the following form:

$$R_j: \text{ If } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \text{ and ... and } x_M \text{ is } A_{Mj}, \text{ Then } y \text{ is } B_j \tag{4}$$

where $A_{ij}, i = 1, 2, ..., M$ and B_j are fuzzy subsets defined on the universe of discourse of the input and output variables. A fuzzy set is specified by its fuzzy membership function and all the membership functions for the same variable define its fuzzy partitioning. Several types of functions, such as triangular, trapezoid, Gaussian, Cauchy, Laplace and spline functions can serve as membership functions. Nevertheless, not all of them satisfy the interpretability conditions of fuzzy systems.

It is noticed that the output of the Mamdani fuzzy rules is also a fuzzy linguistic variable. In practice, it should be defuzzified. A number of defuzzification strategies are available, among which a simplified defuzzification method is as follows:

$$y = \frac{\sum_{j=1}^{N}(b_j w_j)}{\sum_{j=1}^{N} w_j} \tag{5}$$

where, b_j is the center value of the membership function B_j, w_j is called the truth value of the j-th rule, which is calculated by:

$$w_j = T_{i=1}^{M} A_{ij}(x_i) \tag{6}$$

where T is a fuzzy AND (T-norm) operator.

The Sugeno-Takagi fuzzy model (also TSK fuzzy model) was proposed by Sugeno and his colleagues. A Sugeno-Takagi fuzzy model has the following form of fuzzy rules:

$$R_j: \text{ If } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \text{ and ... and } x_M \text{ is } A_{Mj}, \text{ Then } y = g_j(x_1, x_2, ..., x_M) \tag{7}$$

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

$$y = \frac{\sum_{j=1}^{N}(g_j w_j)}{\sum_{j=1}^{N} w_j} \tag{8}$$

where w_j is the truth value of the rule and can be obtained using equation (6). A schematic description of a fuzzy system with two input variables is given in Fig. 2, where each variable has two fuzzy subsets and there are four fuzzy rules in total.

It is noticed that the RBF neural network expressed in (3) and the fuzzy systems described by equation (5) or equation (8) are mathematically equivalent provided that multiplication is used for the T-norm in fuzzy systems and both systems use Gaussian basis functions (refer also to Fig. 3). We will not re-state the restrictions proposed in [1], 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 physical meaning. In fact, when we talk about extracting fuzzy rules from neural networks, there must be some differences between these two systems.

B. 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 believed to be inherently intelligible, especially when the fuzzy rules are obtained from human experts. Therefore, the interpretability of fuzzy systems has not been studied until recently [14,15]. However, interpretability of fuzzy systems will be lost if no special attention is paid during data based rule generation and adaptation. In the following, we try to provide some main conditions for a fuzzy system that can be said to be well interpretable:

1. The fuzzy partitionings 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:

$$\delta_1 \leq S(A_i, A_{i+1}) \leq \delta_2 \tag{9}$$

   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. The fuzzy similarity measure is a distance measure in the following form:

$$S(A_i, A_{i+1}) = \frac{M(A_i \cap A_{i+1})}{M(A_i) + M(A_{i+1}) - M(A_i \cap A_{i+1})} \tag{10}$$

   where $M(A)$ is the size of the fuzzy set A calculated by:

$$M(A(x)) = \int_{x \in U} A(x)dx \tag{11}$$

   where, U is the universe of discourse of variable x.

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 performance deterioration and make the fuzzy system inexplicable. Therefore, the consistency of fuzzy systems has received increasing attention recently [16-18].

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.

It is noticed that some of the interpretability conditions are imposed because of the limited cognitive capacity of human beings. This is reasonable in that the comprehensibility of knowledge is also dependent on the cognitive capacity of human beings.

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 neural networks. In this sense, RBF neural networks and fuzzy systems are not the same even if they are functionally equivalent.

C. Convertibility between RBF Neural Networks and Fuzzy Systems

Since RBF networks and fuzzy systems are not identical if we impose interpretability conditions on fuzzy systems, it is sensible to consider the convertibility between these two systems. It has been mentioned above that the conditions for converting fuzzy systems to RBF networks are not the same as those for converting RBF neural networks to well interpretable fuzzy systems. In this subsection, we discuss these conditions separately and show what needs to be done to extract interpretable fuzzy rules from RBF neural networks.

Let us first discuss which conditions should be satisfied to convert a fuzzy system to an RBF neural network. It is noticed that a fuzzy system can be converted to an RBF neural network if the following conditions are satisfied:

1. The fuzzy system is either a zero-order Takagi-Sugeno model or a Mamdani model. If a Mamdani model is involved, the defuzzification method should use equation (5).

2. The membership functions of the fuzzy system are Gaussian functions.

3. The fuzzy T-norm is a product.

4. The basis functions within each receptive field unit of the RBF neural network are allowed to have different variances.

5. The output of the RBF network is normalized (see equation (3)).

In reference [1], it is required that the membership functions within each fuzzy rule should have the same variance. Such a restriction is proposed to ensure that the variances of the basis functions within a receptive field unit are the same. However, it gives rise to an undesired situation, where either all the membership functions have the same variance or each membership function can be used only once. In the former case, the flexibility of fuzzy systems is seriously reduced. In the latter case, on the other hand, the number of fuzzy subsets needed for each input variable will be as large as the number of times it appears in all fuzzy rules. This obviously harms the interpretability of fuzzy systems because the number of times a variable appears in a rule base is usually large. In a standard fuzzy system, where each variable appears in the premise part of all the rules, the number of fuzzy sets needed for each input will equal the number of fuzzy rules. With the introduction of condition 4, this situation can be avoided. Note that condition 5 makes it possible to convert Mamdani fuzzy rules as well as Sugeno fuzzy rules to RBF neural networks.

Conversion of fuzzy systems to RBF neural networks is of great importance due to the following reasons. After a fuzzy system is converted to an RBF neural network, all the learning strategies developed in the field of neural networks can be applied to the fuzzy system. We will show that for fuzzy systems, an adaptation in terms of neural network learning is much more than a pure parameter optimization, in other words, the structure of the fuzzy system will also be changed. Such a structural change damages the interpretability of the original fuzzy system if no extra conditions are imposed. On the other hand, a novel, usually a more optimal rule structure will be found if well interpretable fuzzy rules are again extracted from the trained RBF network. Therefore, this provides a new approach to structure optimization for fuzzy rule systems. In addition, the conversion of fuzzy rules to neural networks paves a way for structure determination and knowledge integration for artificial neural networks.

Conversion of RBF neural networks back to fuzzy systems is no less significant than conversion of fuzzy systems to neural networks. However, we will show that converting RBF neural networks to fuzzy systems is quite different from converting fuzzy systems to RBF neural networks. The key work of such a conversion is to ensure that the extracted fuzzy rule is well interpretable, i.e. easy to understand. In order to convert an RBF network to a well interpretable fuzzy rule system, the following conditions should be satisfied:

1. The basis functions of the RBF neural network are Gaussian functions.

2. The output of the RBF neural network is normalized (equation(3)).

3. The basis functions within each receptive field unit of the RBF neural 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 can not be realized. As the most important condition, condition 4 requires that some weights in the RBF neural 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 neural networks in the following text. The weight sharing condition ensures a good distinguishability for the fuzzy partitionings, which is the most essential feature for the interpretability of fuzzy systems. If we have a closer look at the RBF network that is directly converted from a fuzzy system (see Fig. 3), we notice that some of the basis functions are identical, for example, $\varphi_{11} = \varphi_{12}$, $\varphi_{21} = \varphi_{23}$ and so on. However, we find it difficult to define the weight sharing condition explicitly because we can not 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 of the fuzzy partitionings 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.

We also notice that the consistency condition is not considered here, because we suppose that it has been taken into account in generating the initial fuzzy system [18]. 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 neural networks

As we have discussed in the last section, to extract well interpretable fuzzy rules from an RBF neural network, some weights should share. That is to say, there exist some inherent constraints among the parameters of a fuzzy system. From the neural network point of view, for an interpretable fuzzy system and an RBF neural network that are mathematically equivalent, the effective number of free parameters in the fuzzy system is smaller than that in the neural network. Thus, extracting interpretable fuzzy rules from RBF neural networks can be regarded in a way as reducing the effective number of free parameters in the neural network. To achieve this, neural network regularization methods can be introduced.

The original goal of regularization was to improve the generalization ability of neural networks. One basic idea of regularization is to reduce the effective number of free parameters in the neural network. Weight sharing [19] is one of the methods inspired by this idea, which try to let one single value be shared among many weights. Therefore, it is necessary to specify, which weights should be identical before the weight sharing algorithm can be applied. Thus, the first step toward rule extraction from RBF neural networks is to determine which weights, including the parameters of basis functions and the output weights, should share.

A. Specification of Shared Weights

As we have mentioned above, we can not expect that the structure of the extracted fuzzy system is the same as its original structure. Therefore, we are not aware in advance, which weights should be identical. However, it is straightforward to imagine that the distance between the weights that will share a common value should be small. Currently, several distance measures (or similarity measures) are available. Among them, Euclidean distance is very simple and has been widely used. For two arbitrary n dimensional vectors $V_1 = (v_{11}, v_{12}, \cdots, v_{1n})^T$ and $V_2 = (v_{21}, v_{22}, \cdots, v_{2n})^T$, the Euclidean distance between these two vectors is calculated by:

$$d(V_1, V_2) = \sqrt{\sum_{k=1}^{n} (v_{1k} - v_{2k})^2} \tag{12}$$

For the basis functions (or membership functions), each vector has two elements, namely, the center and the variance. For the output weights, each vector has only one element. As an example, the procedure to specify which basis functions are expected to have the same parameters is demonstrated in Fig. 4, where we suppose that the input variable x_i has M different basis functions $\varphi_{ij} (j = 1, 2, \cdots, M)$ with center μ_{ij} and variance σ_{ij}.

Remarks: Since the basis functions in U_{ik} will share a mutual center and a mutual variance, the prescribed distance threshold d_i is very important. It determines not only the distinguishability of the fuzzy partitioning, but also the completeness of the fuzzy partitioning. Suppose $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ are the averaged center and variance of the basis functions in U_{ik}, then the fuzzy partitioning constructed by $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ should satisfy the completeness and distinguishability condition described in equation (9).

In practice, we find that the performance of the extracted fuzzy system is not satisfactory if we simply select $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ as the values to be shared by the basis functions in U_{ik}. In other words, it is imperative to adapt the parameters that are to be shared. In the following subsection, we will introduce an adaptive weight sharing method to improve the performance of the extracted fuzzy system.

B. Adaptive Weight Sharing

We do not directly require that the weights in the same set should be identical. Alternatively, we realize weight sharing by regularizing the RBF neural network. In order to describe the regularization algorithm conveniently, we assume that the RBF network has the following form:

$$y = \frac{\sum_{j=1}^{N}[w_j \Pi_{i=1}^{m_j} \exp(-\frac{(x_i-\mu_{ij})^2}{\sigma_{ij}^2})]}{\sum_{j=1}^{N} \Pi_{i=1}^{m_j} \exp(-\frac{(x_i-\mu_{ij})^2}{\sigma_{ij}^2})} \tag{13}$$

where, N is the number of the receptive field units (the number of fuzzy rules), m_j is the number of basis functions within the j-th receptive field unit (the number of variables appearing in the premise of the j-th rule), μ_{ij} and σ_{ij} are the center and variance of the basis function (fuzzy membership function), and w_j is the output weight of the j-th receptive field unit. For the definition of m_j, please refer to equation (2). It is noticed that we have adopted a non-standard rule structure for the fuzzy system, because it is found that a fuzzy system with a non-standard structure has usually better performance [20]. In addition, we intend to optimize both the parameters and the structure of the fuzzy system with the help of neural network learning. Therefore, such a non-standard rule structure is necessary.

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

$$J = E + \lambda \cdot \Omega \tag{14}$$

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

$$E = \frac{1}{2}(y - y^t)^2 \tag{15}$$

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

$$\Omega = \frac{1}{2}\sum_i \sum_k \sum_{\varphi_{ij} \in U_{ik}} (\mu_{ij} - \bar{\mu}_{ik})^2 + \frac{1}{2}\sum_i \sum_k \sum_{\varphi_{ij} \in U_{ik}} (\sigma_{ij} - \bar{\sigma}_{ik})^2 \tag{16}$$

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 $\bar{\sigma}_{ik}$. According to the gradient method, we have

$$\begin{aligned}
\frac{\partial J}{\partial \mu_{ij}} |_{\varphi_{ij} \in U_{ik}} &= \frac{\partial E}{\partial \mu_{ij}} + \lambda(\mu_{ij} - \bar{\mu}_{ik}) &(17)\\
\frac{\partial J}{\partial \sigma_{ij}} |_{\varphi_{ij} \in U_{ik}} &= \frac{\partial E}{\partial \sigma_{ij}} + \lambda(\sigma_{ij} - \bar{\sigma}_{ik}) &(18)\\
\frac{\partial J}{\partial \bar{\mu}_{ik}} &= -\lambda \sum_{\varphi_{ij} \in U_{ik}} (\mu_{ij} - \bar{\mu}_{ik}) &(19)\\
\frac{\partial J}{\partial \bar{\sigma}_{ik}} &= -\lambda \sum_{\varphi_{ij} \in U_{ik}} (\sigma_{ij} - \bar{\sigma}_{ik}) &(20)
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial E}{\partial \mu_{ij}} &= (y - y^t)(w_j - y)(x_i - \mu_{ij}) \prod_{i=1}^{m_j} \left[\exp\left(-\frac{(x_i - \mu_{ij})^2}{\sigma_{ij}^2}\right)\right] / \sigma_{ij}^2 &(21)\\
\frac{\partial E}{\partial \sigma_{ij}} &= (y - y^t)(w_j - y)(x_i - \mu_{ij})^2 \prod_{i=1}^{m_j} \left[\exp\left(-\frac{(x_i - \mu_{ij})^2}{\sigma_{ij}^2}\right)\right] / \sigma_{ij}^3 &(22)
\end{aligned}$$

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. Remind 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 set to 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 partitionings 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 (w_j) 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. Without loss of generality, we assume a general value one for the variance of the membership functions for the output.

4 Results from the Lorenz System

This section describes the simulation study on the Lorenz system. The goal of this simulation study is to show how a fuzzy system can be further optimized by converting it to an RBF neural network and how a well interpretable fuzzy system can again be extracted. Through such a process, we mainly achieve the following two goals:

1. To develop an effective method for optimizing fuzzy systems with a systematical combination of evolutionary computation and artificial neural networks. In this way, the power of the existing soft computing techniques can be exploited as fully as possible.

2. To find a new way to discover unknown dependencies between the input and output of the system and thus to acquire well intelligible knowledge about it.

The Lorenz system studied in this paper is described by the following differential equations [21]:

$$\frac{dx}{dt} = -y^2 - z^2 - a(x - F) \tag{23}$$

$$\frac{dy}{dt} = xy - bxz - y + G \tag{24}$$

$$\frac{dz}{dt} = bxy + xz - z \tag{25}$$

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 sets 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 model for this system is generated with the help of evolution strategies, which is provided in Table 1, where a '*' means 'don't care'. The fuzzy system has 13 fuzzy rules, and its approximation results are given in Fig. 5. We stop the evolution process when no significant performance improvement is observed. 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, it is found that the activation level of some receptive field units (fuzzy rules) becomes zero, because centers of some basis functions within these units have moved out of their corresponding working ranges. Such units are deleted because they are inactive rules from the fuzzy system point of view. Thus, after the learning algorithm converges, we obtain an RBF network with 5 receptive field units, as is demonstrated in Table 2. Fig. 6 shows the approximation results of the RBF neural network. From Fig. 6, it can be seen that the performance improvement is significant. Moreover, during the neural network learning, not only the parameters, but also the structure of the fuzzy system have been changed. That is to say, it is impossible to convert it back to the original fuzzy system. Of course, an RBF network can always be treated as a fuzzy system without considering the interpretability conditions. However, most of its fuzzy subsets (basis functions) are hard to distinguish (see Fig. 7) and such a fuzzy system is not well interpretable.

To extract a well interpretable fuzzy system from the RBF network, the algorithm for extracting fuzzy rules is implemented. The algorithm consists of two parts: specification of shared weights and network training with regularization. The extracted fuzzy rules are listed in Table 3, while the approximation results are shown in Fig. 8. Although the performance of the extracted fuzzy system is a little worse than that of the RBF network, it is much better than the initial fuzzy system. More importantly, the interpretability of the extracted fuzzy system is good: the fuzzy partitionings are both complete and well distinguishable (see Fig. 9) and the number of fuzzy rules is significantly reduced.

Furthermore, new 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 meaning can be assigned to each fuzzy set (refer to Fig. 9). 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 $(-0.35, 4.11)$, " **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 can be 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 precision 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, especially when the fuzzy partitionings of the fuzzy system are required to be well distinguishable. However, some other aspects of the interpretability, for example, higher consistency of the fuzzy rules and the completeness of fuzzy partitionings may improve the generalization capability of the fuzzy model when training data are insufficient [18].

5 Conclusions

The relationships between an RBF neural network and an interpretable fuzzy system have been discussed. A definition for a well interpretable fuzzy system is also suggested. Conditions for converting RBF neural networks to fuzzy systems and vice versa are proposed. In order to extract well interpretable fuzzy rules from an RBF neural network, an adaptive weight sharing algorithm is introduced. We show that an RBF neural network and a fuzzy system are not fully equivalent in terms of their physical meaning and the extraction of well interpretable fuzzy rules from RBF neural networks is both important and feasible for gaining a deeper insight into the logical structure of the system to be approximated. The simulation studies on the Lorenz system are carried out to demonstrate our ideas and methods.

6 References

[1] J.-S. R. Jang and C.-T. Sun, " Functional equivalence between radial basis functions and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, pp. 156-158, 1993.

[2] J.-S. R. Jang and C.-T. Sun, " Neuro-fuzzy modeling and control," *Proc. of the IEEE*, vol. 83, pp.378-405, 1995.
[3] G. Towell and J. Shavlik, " Extracting refined fuzzy rules from knowledge-based neural networks," *Machine Learning*, vol. 13, pp. 71-101, 1993.
[4] L. M. Fu, " Rule generation from neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 1114-1124, 1994.
[5] Y. Hayashi, " A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosis," *Advances in Neural Information Processing Systems*, vol. 3, pp. 578-584, 1990.
[6] A. Lozowski, T. J. Cholevo and J. M. Zurada, " Crisp rule extraction from perceptron network classifiers," In: Proc. IEEE Int. Conf. on Neural Networks, pp. 94-99, Washington, D.C., 1996
[7] J. Moody and C. Darken, " Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281-194, 1989.
[8] J. Moody, " Fast learning in multi-resolution hierarchies," *Advances in Neural Information Processing Systems*, vol. 1, pp. 29-39, 1989.
[9] M. Powell, " Radial basis functions for multivariable interpolation: A review," *Algorithms for Approximation*, C. Mason and M.G. Cox (eds.), Oxford, UK: Oxford University Press, 1987, pp. 143-167.
[10] L.A. Zadeh, " Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
[11] L.A. Zadeh, " Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, Vol. 3, pp. 28-44, 1973.
[12] E. H. Mamdani and S. Assilian, " An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Machine Studies*, vol. 7, pp. 1-13, 1975.
[13] T. Takagi and M. Sugeno, " Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp. 116-132, 1985.
[14] A. Lofti and A.C. Tsoi, " Interpretation preservation of adaptive fuzzy inference systems," *Int. J. Approximating Reasoning*, vol.15, 1996
[15] Y. C. Jin, W. von Seelen and B. Sendhoff, " An approach to rule-based knowledge extraction," in: IEEE Int. Conf. on Fuzzy Systems, Anchorage, AL, pp. 1188-1193, 1998
[16] D. Dubois, H. Prade and L. Ughetto, " Checking the coherence and redundancy of fuzzy knowledge bases," *IEEE Trans. Fuzzy Systems*, vol. 5, pp. 398-417, 1997.
[17] S. Gottwald, U. Petri, " An algorithmic approach toward consistency checking systems of fuzzy control rules," in: Proc. 3rd European Congress on Intelli. Tech. Soft Computing," Aachen, Germany, 1995, pp. 682-687.
[18] Y. C. Jin, W. von Seelen and B. Sendhoff, " Generating FC^3 fuzzy rule systems from data using evolution strategies," *IEEE Trans. Syst., Man, Cybern.*, Vol. 29, No. 6, pp. 829-845, 1999.
[19] Y. LeCun, " Generalization and network design strategies," *Technical Report*, University of Toronto, 1989.
[20] Y. C. Jin, " Decentralized adaptive fuzzy control of robot maniputaltors," *IEEE Trans. Syst., Man, Cybern.*, Vol. 28B, No. 1, pp. 47-57, 1998
[21] E. N. Lorenz, " Irregularity: A fundamental property of the atmosphere," *Tellus*, vol. A, no. 36, pp. 98, 1984.

If			Then
$x(t-1)$	$y(t-1)$	$z(t-1)$	$x(t)$
(1.27, 1.92)	(-1.73, 3.52)	(-1.54, 0.08)	2.50
(-0.13, 0.97)	$*$	$*$	-0.20
(0.67, 0.93)	(-0.44, 3.01)	(-1.54, 0.08)	-0.20
(-0.13, 0.97)	(-1.73, 3.52)	(0.01, 2.87)	-0.20
$*$	$*$	(-1.54, 0.08)	0.70
(-0.13, 0.97)	$*$	(0.01, 2.87)	-0.20
(-0.13, 0.97)	(-0.44, 3.09)	(-1.54, 0.08)	-0.20
(-0.13, 0.97)	(-1.73, 3.52)	(-1.54, 0.08)	2.50
$*$	(0.44, 3.01)	(-1.54, 0.08)	0.70
(0.67, 0.93)	(0.44, 3.01)	(0.01, 2.87)	2.50
(-0.13, 0.97)	(-0.44, 3.09)	$*$	-0.20
(-0.13, 0.97)	$*$	(-1.54, 0.08)	-0.20
(1.27, 1.92)	$*$	(-1.54, 0.08)	2.50

Table 1: The initial fuzzy system.

Basis Functions			Output Weights
$x(t-1)$	$y(t-1)$	$z(t-1)$	w_j
(-0.14, 1.09)	$*$	$*$	-0.49
(0.98, 0.61)	(-0.28, 4.21)	(-0.08, 2.94)	1.10
(0.06, 1.06)	$*$	*(0.31, 3.10)	-0.49
(1.07, 1.04)	(0.01, 3.14)	(0.09, 2.58)	2.48
(0.49, 0.51)	(-0.27, 3.25)	$*$	0.53

Table 2: The RBF neural network.

If			Then
$x(t-1)$	$y(t-1)$	$z(t-1)$	$x(t)$
(0.37, 1.24)	$*$	$*$	-0.50
(2.01, 1.49)	(0.03, 3.53)	(-0.04, 2.79)	2.50
(-0.49, 1.69)	$*$	(-0.04, 2.79)	-0.50
(2.01, 1.49)	(0.03, 3.53)	(-0.04, 2.79)	2.50
(1.01, 0.80)	(-0.35, 4.11)	$*$	1.18

Table 3: The extracted fuzzy system

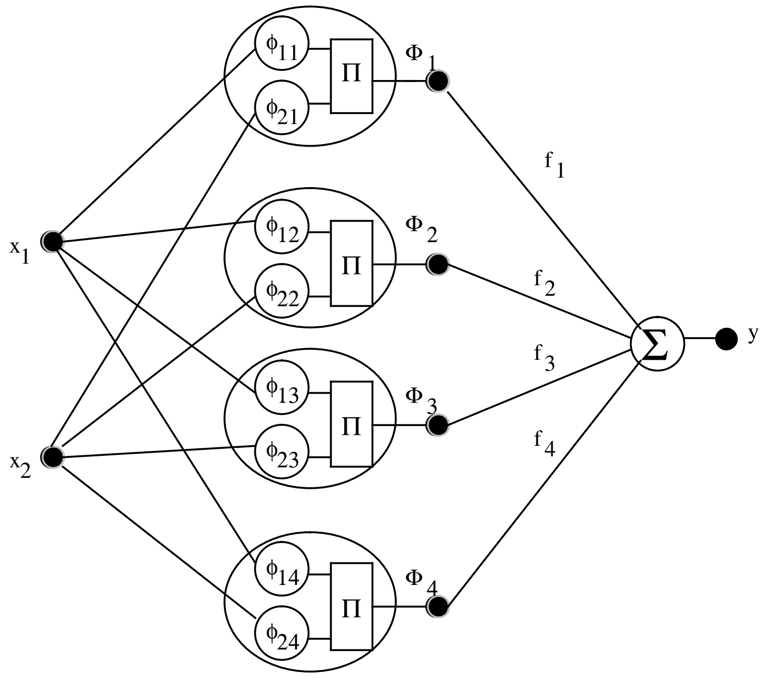

Figure 1: An RBF Neural Network with Gaussian Basis Functions.

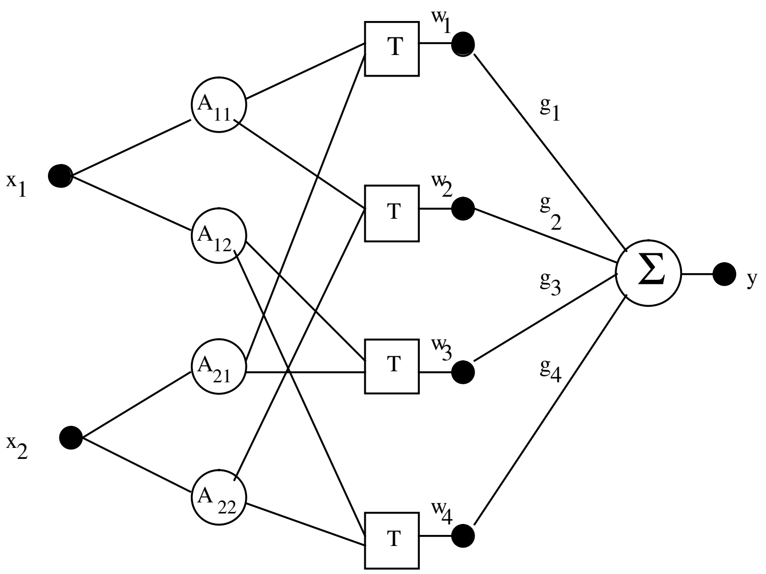

Figure 2: A Fuzzy System.

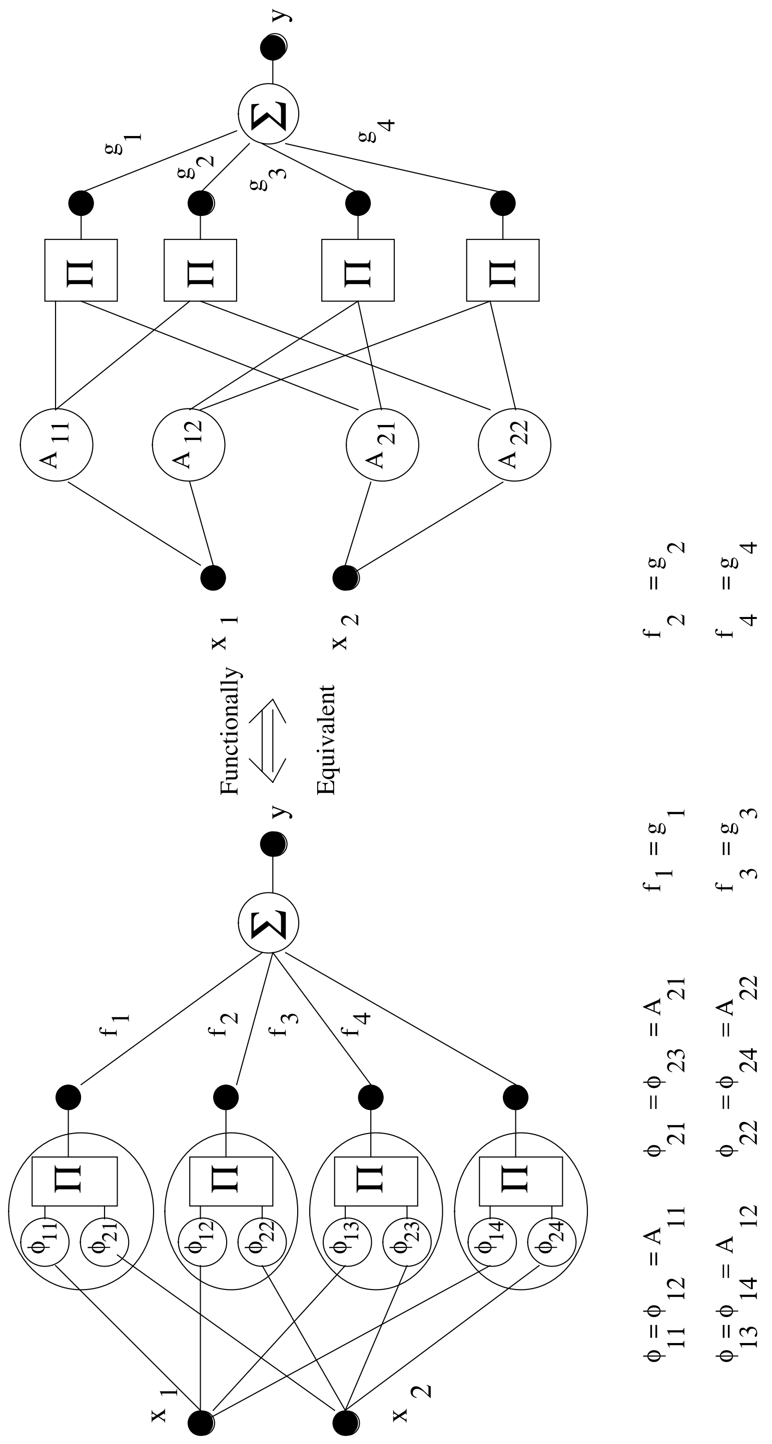

Figure 3: Functional Equivalence between RBF Networks and Fuzzy Systems.

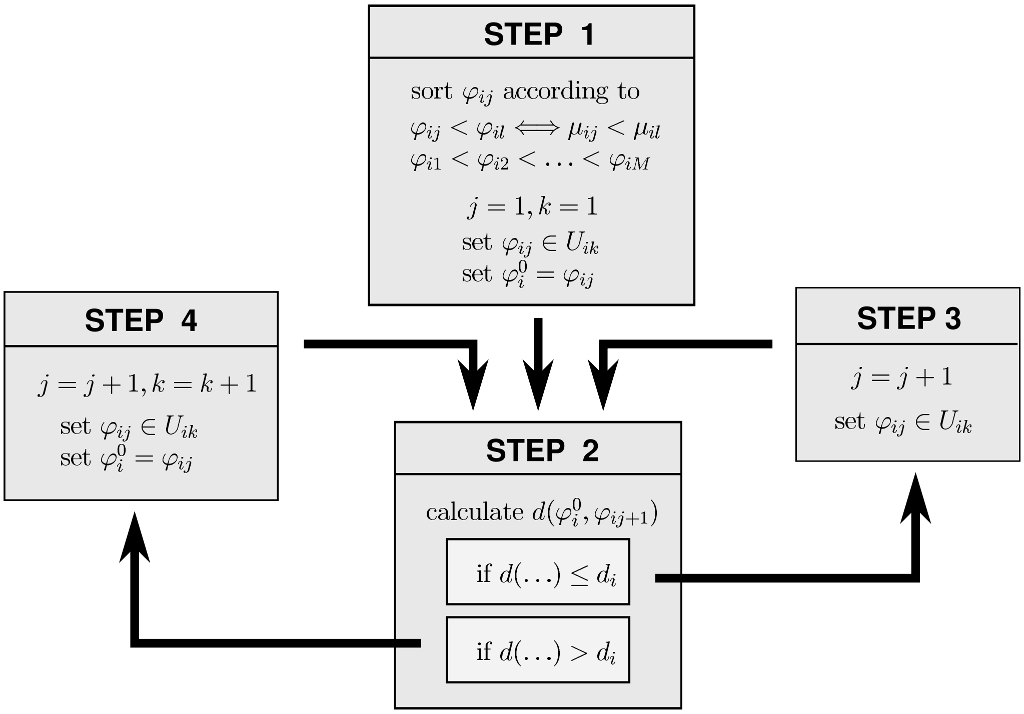

Figure 4: Flow Chart for Specification of Shared Weights.

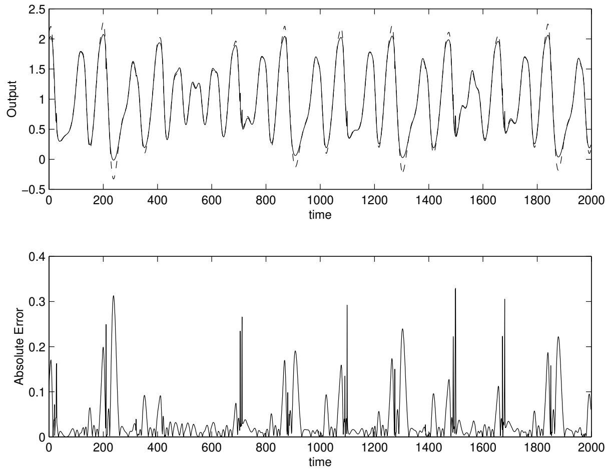

Figure 5: Approximation Results of the Initial Fuzzy System.

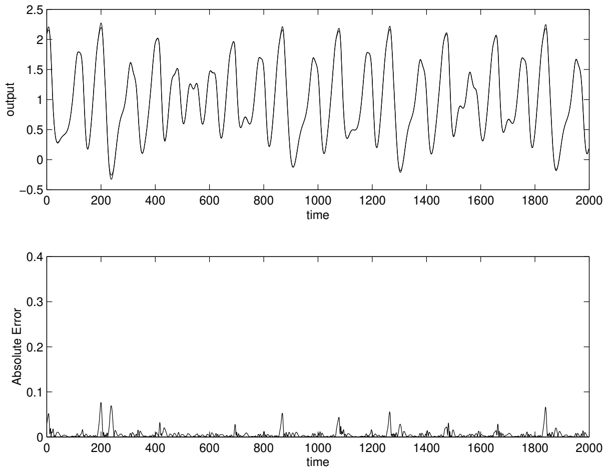

Figure 6: Approximation Results of the RBF Neural Network.

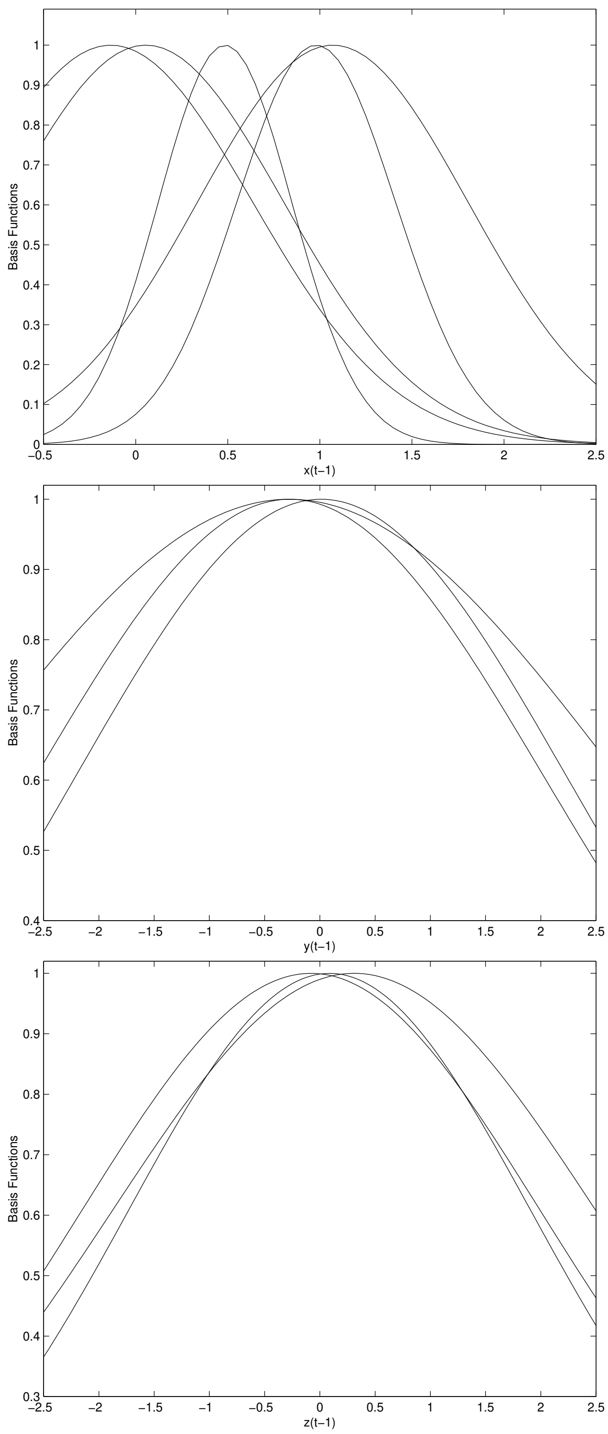

Figure 7: Basis Functions of the RBF Neural Network: a) x(t-1) b) y(t-1) c) z(t-1).

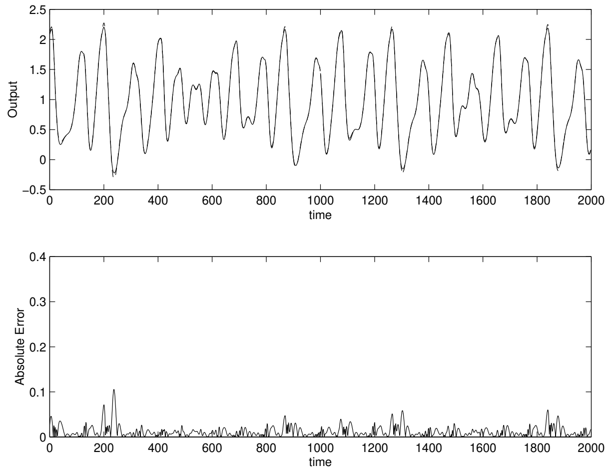

Figure 8: Approximation Results of the Extracted Fuzzy System.

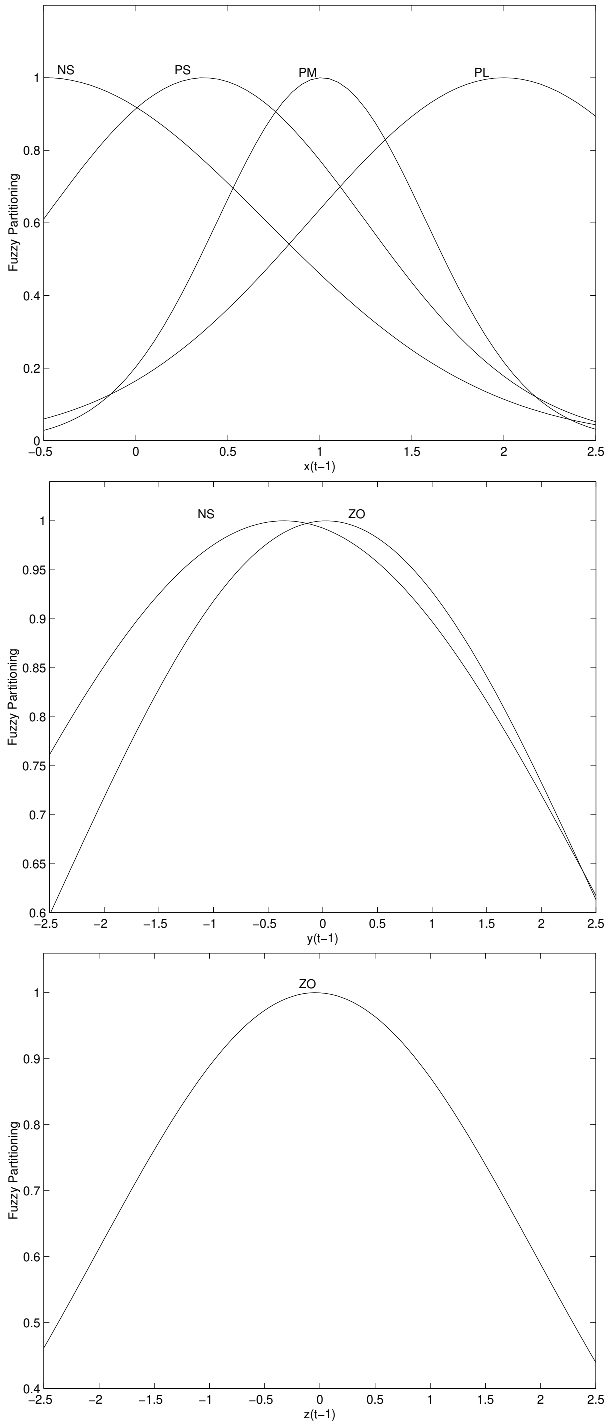

Figure 9: Fuzzy Partitionings of the Extracted Fuzzy System: a) x(t-1) b) y(t-1) c) z(t-1).