Two-Stages Support Vector Regression for Fuzzy Neural Networks with Outliers

Chen-Chia Chuang, Jin-Tsong Jeng, and C.W. Tao

Abstract

In this study, two-stages support vector regression (TSSVR) approach is proposed to deal with training data set with outliers for fuzzy neural networks (FNNs). The proposed approach in the stage I, called as data preprocessing, the support vector machines for regression (SVMR) approach is used to filter out the outliers in the training data set and determine the number of fuzzy rule. Due to the outliers in the training data set are removed, the concept of robust statistic theory have no need to reduce the outlier’s effect. Then, the training data set except for outliers, called as the reduced training data set, is directly used to training the sparse least squares support vector machines for regression (LS-SVMR) in the stage II. Consequently, the learning mechanism of the proposed approach for fuzzy neural network does not need iterated learning for simplified fuzzy inference systems. Based on the simulation results, the performance of the proposed approach is superior to the robust LS-SVMR approach when the outliers are existed.

Keywords: Outliers, Support vector machines for regression, Least squares support vector machines for regression, Fuzzy neural network.

1. Introduction

In this paper, two-stage support vector regression (TSSVR) approach is proposed to improve the simplified fuzzy inference systems for the fuzzy neural networks. In general, the support vector machine (SVM) is very suitable to approximate a high dimensionality space and ill-posed problem in function approximation [1-2]. That is, the support vector algorithm consists of a quadratic programming problem that can be solved efficiently and guaranteed to find a global extremism. Another, the lease square (LS) versions of SVM, called as LS-SVM, is also investigated for classification [3] and regression (LS-SVMR) [4]. In these LS-SVM formulations on works with equality instead of inequality constraints and a sum squared error cost function as it is frequently used in the training of traditional neural networks. This reformulation greatly simplifies the problem in such a way that the solution is characterized by a linear system, more precisely a KKT (Karush-Kuhn-Tucker) system [5], which takes a similar form as the linear system that one solves in every iteration step by the interior point method for the standard SVM [6]. This linear system can be efficiently solved by iterative methods such as conjugate gradient [7]. Then, a LS-SVM is computationally more effective than a SVM [8]. While a LS-SVM incorporates all training data in the network to produce the result, the traditional SVM selects some of SVs that are important in the regression. This sparseness of the traditional SVM can also be reached with the LS-SVM by applying a pruning method [8]. Besides, a radial basis function network is one of the SVM and it can be found in [2]. That is, the support vector algorithm can automatically determines cents, weights, and threshold that minimize an upper bound on the expected test error for the radial basis function network.

In general, a fuzzy inference system consists of four important parts; namely, the fuzzification interface, knowledge base, decision-making unit, and defuzzification [9]. Besides, fuzzy inference systems and simplified fuzzy inference systems have been proved to be a university approximator [10-12]. That is, it is capable of approximating any real continuous function on a compact set to an arbitrary accuracy. Hence, it has many applications for the control and identification of nonlinear systems. However, a fuzzy system has not much learning capability; it is difficult for a human operator to tune the fuzzy rule and membership function from the training data set. On the other hand, the internal layers of neural networks are always opaque to the user; the mapping rules in the network are not visible and difficult to understand. Hence, for a fuzzy neural network, neural network is essentially low-level computational structure and algorithm that offer better performance in dealing with sensory data, while fuzzy logic techniques often deal with issues such as reasoning.
on a higher -level than pure neural networks. That is, a fuzzy neural network has better performance than a fuzzy system and a neural network. Therefore, there are many researches in mixes neural network and fuzzy logic to be a fuzzy neural network [13-15]. However, the number of simplified fuzzy inference system rule and the initial weights and structure of fuzzy neural networks are difficult to determine at present.

In real applications, the obtained data may be subject to outliers. The outliers occur for various reasons, such as erroneous measurements or noisy data from the heavy-tails of noise distribution functions [16-17]. To overcome the outlier’s effects, the robust LS-SVMR with outliers, called as the weighted LS-SVMMR, is proposed [18]. Additionally, the robust support vector regression networks (RSVRNs) based on the support vector machine for regression (SVMR) approach and robust learning algorithm is also proposed to overcome the outlier’s effects [19]. Due to the training data that includes outliers are aggregatedly considered in the training process, the concepts of the robust statistic theory are used to reduce the outlier’s effects. When the parameters in those approaches are improperly selected, those approaches cannot provide a satisfactory performance when the training data with outliers. In order to overcome the above drawbacks, the TSSVR is proposed to improve the simplified fuzzy inference systems for the fuzzy neural networks. Firstly, the SVMR approach is used to filter out the outliers in the training data set. That is, to filter out the outliers relies on the standard deviation of training error with $\varepsilon = 0$ in the SVMR, the concepts of repeated SVMR approach is used [19]. Hence, the outliers in the training data set are removed in the stage I of proposed approach. At the same time, the concept of robust statistic theory has no need to reduce the outlier’s effect in the stage II of proposed approach. Secondly, the training data set except for outliers, called as the reduced training data set, is directly used to training the sparse LS-SVMR in the stage II. That is, the learning mechanism of the proposed approach in the stage II for the FNNs does not need iterated learning. At the same time, the proposed sparse LS-SVMR is applied to determine the number of rule and the weights for the FNNs with the reduced training data set. Finally, simulation results are provided to show the validity and applicability of the developed method.

The remaining part of the paper is outlined as follows. Section 2 simply describes the fuzzy neural networks. In section 3, hybrid support vector machines for fuzzy neural networks with outliers are proposed. In section 4, the simulation results show the superiority of the proposed approach for the fuzzy neural networks. Concluding remarks are presented in section 5.

2. Preliminary on Fuzzy Neural Networks

In the fuzzy inference systems, fuzzy logic principles are used to combine the fuzz IF-THEN rules in fuzzy rule base into a mapping from fuzzy sets in $U = U_1 \times \cdots \times U_n$ to fuzzy sets in $V$. This class of fuzzy inference systems has been proved to be a universal approximator [11-12]; that is, they are capable of approximating any real continuous function on a compact set to an arbitrary accuracy, provided sufficient fuzzy logic rules are available. The fuzzy rule base comprises the following fuzzy IF-THEN rules:

Rule i: IF $x_1$ is $A^i_1$ and ... and $x_n$ is $A^i_n$, THEN $y$ is $B^i$,           (1)

where $A^i_j$ and $B^i_j$ are fuzzy sets in $U_i \subset R$ and $V \subset R$, respectively, and $\bar{x} \in R^n$ and $y \in V$ are the input and output variables of the fuzzy system, respectively.

Given a pair $(\bar{x}_i, y_i)$, the final output of the fuzzy system is inferred as follows:

\[
\hat{y}_i = \frac{\sum_{j=1}^m \beta_i \prod_{k=1}^n \mu_{A^i_j}(x_{ij})}{\sum_{j=1}^m \prod_{k=1}^n \mu_{A^i_j}(x_{ij})}, \quad s = 1, \ldots, p. \tag{2}
\]

Let $\psi_j(\bar{x})$ be functions of input $x = (x_1, x_2, \cdots, x_n)$:

\[
\psi_j(\bar{x}) = \frac{\prod_{k=1}^n \mu_{A^i_j}(x_{kj})}{\sum_{j=1}^m \prod_{k=1}^n \mu_{A^i_j}(x_{kj})}. \tag{3}
\]

We can define the class of fuzzy inference system as a family of function $\hat{y}_s : R^n \to R$ in the form of

\[
\hat{y}_s(\bar{x}) = \sum_{i=1}^m \beta_i \psi_j, \tag{4}
\]

for $\bar{x} \in R^n$ and $\beta_j \in R$, and $m$ is a finite number of fuzzy rule.

In this paper, the proposed fuzzy neural networks consist of $m$ rules in the form of Eq. (1) and that membership functions are

\[
\mu_{A^i_j}(x_{ij}) = \exp \left( -\frac{(x_{ij} - \bar{x}^j)^2}{2\sigma^j} \right), \tag{5}
\]

where $\bar{x}^j_i$ and $\sigma^j_i$ are the constant parameters, $j = 1, 2, \cdots, n$ and $i = 1, 2, \cdots, m$. That is, the membership functions are all Gaussian function in this
Let $\sigma_1 = \sigma_2 = \cdots = \sigma_n = \sigma_i$, 
\[ \prod_{j=1}^{n} \exp \left( \frac{(x_j - \bar{x}_j)^2}{2\sigma_j^2} \right) = \exp \left( \frac{-\|\bar{x} - \bar{y}\|^2}{2\sigma_i^2} \right), \quad (6) \]
where $\bar{x}_i = (\bar{x}_{i1}, \bar{x}_{i2}, \cdots, \bar{x}_{in})$. Substituting Eq. (6) into Eq. (3)
\[ \hat{y}_s = \frac{\sum_{i=1}^{m} \beta_i \exp \left( \frac{-\|\bar{x} - \bar{y}\|^2}{2\sigma_i^2} \right)}{\sum_{i=1}^{m} \exp \left( \frac{-\|\bar{x} - \bar{y}\|^2}{2\sigma_i^2} \right)} . \]
When $\Phi_i = \exp \left( \frac{\|\bar{x} - \bar{y}\|^2}{2\sigma_i^2} \right)$ and $W_i = \frac{\beta_i}{\sum_{i=1}^{m} \exp \left( \frac{-\|\bar{x} - \bar{y}\|^2}{2\sigma_i^2} \right)}$, the Eq. (7) can be rewrite as
\[ \hat{y}_s (\bar{x}) = \sum_{i=1}^{m} W_i \Phi_i = \sum_{i=1}^{m} W_i \exp \left( \frac{-\|\bar{x} - \bar{y}\|^2}{2\sigma_i^2} \right) . \quad (8) \]
It is shown that fuzzy neural networks in Eq. (8) can be represented as the functional link networks that are based on Gaussian function. Besides, the proposed fuzzy neural networks are shown in Fig. 1, which are comprised by the input, membership, rule inference system, and output layer.

3. The Hybrid Support Vector Machines for Fuzzy Neural Networks with Outliers
In the proposed approach, two stages are included in this study. Fig. 2 shows the proposed two-stages structure. In the stage I, the repeated SVMR approach is used to filter out the outliers in the training data points. That is, the repeated SVMR approach uses twice SVMR approaches. The process for the repeat SVMR approach is used to filter out the outliers. In regression formulation, the goal is to estimate an unknown continuous-valued function based on a finite number set of noisy samples. Assumed statistical model for data generation has the following form:
\[ y = r(\bar{x}) + \delta , \quad (9) \]
where $r(\bar{x})$ is unknown target function (regression) and $\delta$ can be represented as noise or outliers.

![Fig. 2. The procedure of TSSVR in this study is shown.](image)

In the SVMR, the input $\bar{x}$ is the first mapped onto an $m$-dimensional feature space using some fixed (nonlinear) mapping and then a linear model is constructed in this feature space. Using mathematical notation, the linear model (in the feature space) $f(\bar{x}, \tilde{w})$ is given by
\[ f(\bar{x}, \tilde{w}) = \sum_{j=1}^{m} w_j g_j (\bar{x}) + b, \quad (10) \]
where $\tilde{w} = (w_1, \ldots, w_m)$ is the parameter vector that needs to identified. $g_j (\bar{x})$ denotes a set of nonlinear transformations, and $b$ is the “bias” term. Often the data are assumed to be zero mean (this can be achieved by preprocessing), so the bias term in Eq. (10) is dropped.

The quality of estimation is measured by the loss function $L(y, f(x, \omega))$. The traditional SVMR approach uses a new type of loss function called $\varepsilon$-insensitive
loss function that proposed by Vapnik [2]:

\[
L_{\varepsilon}(y, f(\tilde{x}, \tilde{w})) = \begin{cases} 
0 & \text{if } |y - f(\tilde{x}, \tilde{w})| \leq \varepsilon, \\
|y - f(\tilde{x}, \tilde{w})| - \varepsilon & \text{otherwise.}
\end{cases} 
\] (11)

An $\varepsilon$ zone is defined as that if the value of $\varepsilon$ is within the zone, the loss is zero. Otherwise, the loss is the magnitude of the difference between the absolute value of $\varepsilon$ and the $\varepsilon$ zone. The empirical risk function is estimated as

\[
R_{\text{emp}}(\tilde{w}) = \frac{1}{N} \sum_{i=1}^{N} L_{\varepsilon}(y_i, f(\tilde{x}_i, \tilde{w})). 
\] (12)

Note that the $\varepsilon$-insensitive loss coincides with least-modulus loss and with a special case of Huber’s robust loss function when $\varepsilon = 0$. The SVMR approach performs a linear regression in the high-dimensional feature space using the $\varepsilon$-insensitive loss and tries to reduce model complexity by minimizing $\|\tilde{w}\|^2$. This can be described by introducing (non-negative) slack variables $\xi_i, \xi_i^*$ for $i = 1, ..., N$, to measure the deviation of training samples outside $\varepsilon$-insensitive zone. Thus the SVMR is formulated as minimization of the following functional:

\[
\begin{align*}
\min & \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*), \\
\text{s.t.} & \quad y_i - f(\tilde{x}_i, \tilde{w}) \leq \varepsilon + \xi_i^*, \\
& \quad f(\tilde{x}_i, \tilde{w}) - y_i \leq \varepsilon + \xi_i, \\
& \quad \xi_i, \xi_i^* \geq 0, \quad i = 1, ..., N.
\end{align*} 
\] (13)

This optimization problem can be transformed into the dual problem, and its solution is given by

\[
f(\tilde{x}) = \sum_{i=1}^{n_{SV1}} (\alpha_i^* - \alpha_i) K(\tilde{x}_i, \tilde{x}), \\
\text{s.t.} \quad 0 \leq \alpha_i^* \leq C, 0 \leq \alpha_i \leq C,
\] (14)

where $n_{SV1}$ is the number of SVs for the original training data, $C$ is a regularization constant and the kernel function is defined as

\[
K(\tilde{x}_i, \tilde{x}) = \sum_{j=1}^{m} g_j(\tilde{x}_i) g_j(\tilde{x}). 
\] (15)

In this study, Gaussian kernel function is used as kernel function. Then, Eq. (14) can be represented as

\[
f(\tilde{x}, \alpha, \alpha^*) = \sum_{i=1}^{n_{SV1}} (\alpha_i^* - \alpha_i) \exp\left[-\frac{|\tilde{x} - \tilde{x}_i|^2}{2\sigma^2}\right] + b, 
\] (16)

where $\sigma$ is termed as the Gaussian kernel parameter (GKP). Figure 3 shows that the relational among the output of SVMR, $\varepsilon$ and slack variables.

In general, the data points A, A’, B and B’ are SVs in Figure 3. Due to the data points A’ and B’ are far from the majority of training data, the data points A’ and B’ are regarded as either the outliers or the SVs. To determine which the SVs are regarded as the outlier, the criterion is build as follows rule

\[
\begin{align*}
CR = \begin{cases} \text{Outliers} & \text{if } (\xi_i, \xi_i^*) \geq 3 \times \text{std}(\text{slack variables}), \\
\text{SVs,} & \text{otherwise},
\end{cases}
\end{align*}
\] (17)

where $\text{std}(\text{slack variables})$ represented as the standard deviation of slack variables $\xi_i$ and $\xi_i^*$ for $i = 1, 2, ..., n_{SV1}$.

In the proposed approach, the hyper-parameters (i.e. a Gaussian kernel parameter GKP, epsilon and a regularization constant C) in the SVMR must carefully choose. In general, the regularization constant is suggested as larger as possible [19]. Hence, the regularization constant is chosen as 1000 in this study. In the literature, Jeng’s approach [20] is suitable method for selecting those parameters with outliers. Hence, Jeng’s approach is adopted for the proposed approach in this study. Additionally, the GKP is suggested as $(0.2 \sim 0.5) \times \text{range}(x)$ [21]. It is well known that the value of $\varepsilon$ should be proportional to the input noise level $\psi$ [21]. In this study, the parameter $\varepsilon$ is adopted as

\[
\varepsilon = 0.3 \times \psi \times \sqrt{\frac{\ln N}{N}}. 
\] (18)

Due to the input noise level $\psi$ unknown, the concepts of repeated SVMR approach is used [19]. According to these concepts, the input noise level $\psi$ is estimated as

**Step 1:** The SVMR with $\varepsilon = 0$ is applied to obtain the estimated output results. Consequently, the actual and expected output corresponding to the same input data were used to calculate the errors.

**Step 2:** According to the above errors, the noise level $\psi$...
easily obtained by \( std(\text{errors}) \), where \( std \) represented as the standard deviation.

The procedure for the filter out of the outliers using the SVMR approach is stated as follows. 

**Step 1:** Set kernel parameter GKP and regularization constant \( C \) in the SVMR approach by Jeng’s approach.

**Step 2:** Using the repeated SVMR approach to estimate the input noise level \( \psi \). Then, the parameter \( \varepsilon \) is calculated by Eq. (18).

**Step 3:** The SVs and slack variables are obtained by the SVMR approach with the above epsilon value.

**Step 4:** Using CR in Eq. (17) to discriminate the outliers from SVs.

**Step 5:** Remove the outliers from the training data set to obtain the number of reduced training data.

For the learning mechanism of the LS-SVMR approach, the training data set of \( N_D \) points \( \{x_i^D, y_i^D\}_{i=1}^{N_D} \) with input data \( x_i^D \in R^k \) and output data \( y_i^D \in R \) are considered. Then, the following optimization problem in primal weight space is considered as

\[
\min_{\tilde{w}, \tilde{b}, \tilde{\varepsilon}} J(\tilde{w}, \tilde{\varepsilon}) = \frac{1}{2} x^T \tilde{w}^2 + \frac{1}{2} \sum_{i=1}^{N_D} \tilde{\varepsilon}_i^2,
\]

where \( \tilde{\varepsilon} \) is the error variables and \( \tilde{b} \) is the bias term. The relative importance of these terms is determined by the positive real constant \( \gamma \). In the case of noisy data one avoids overfitting by taking a smaller \( \gamma \) value. In primal weight space, one has the model

\[
y^D(x^D) = \tilde{w}^T \tilde{\phi}(x^D) + \tilde{b}.
\]

The weight vector \( \tilde{w} \) can be infinite dimensional, which makes a calculation of \( \tilde{w} \) from Eq. (19) impossible in general. Therefore, one computes the model in the dual space instead of the primal weight space. One defines the Lagrangian as

\[
L(\tilde{w}, \tilde{b}, \tilde{\varepsilon} ; \alpha) = J(\tilde{w}, \tilde{\varepsilon}) - \sum_{i=1}^{N_D} \alpha_i \left[ \tilde{w}^T \tilde{\phi}(x^D) + \tilde{b} + \tilde{\varepsilon}_i - y_i^D \right]
\]

where \( \alpha \) and \( \alpha_i \) are Lagrange multipliers (called support values). The conditions for optimality are given by

\[
\begin{align*}
\frac{\partial L}{\partial \tilde{w}} = 0 & \implies \tilde{w} = \sum_{i=1}^{N_D} \alpha_i \tilde{\phi}(x_i^D), \\
\frac{\partial L}{\partial \tilde{b}} = 0 & \implies \sum_{i=1}^{N_D} \alpha_i = 0, \\
\frac{\partial L}{\partial \tilde{\varepsilon}_i} = 0 & \implies \alpha_i = \gamma \cdot \tilde{\varepsilon}_i, \quad i = 1, \ldots, N_D, \\
\frac{\partial L}{\partial \alpha_i} = 0 & \implies \tilde{w}^T \tilde{\phi}(x_i^D) + \tilde{b} + \tilde{\varepsilon}_i - y_i^D = 0, \quad i = 1, \ldots, N_D.
\end{align*}
\]

After elimination of \( \tilde{w} \) and \( \tilde{\varepsilon} \), the solution can be represented as

\[
\begin{bmatrix}
0 \\
1_v \Omega + \frac{1}{\gamma} \cdot \frac{1}{\gamma}
\end{bmatrix}
\begin{bmatrix}
\tilde{b} \\
\alpha
\end{bmatrix}
= \begin{bmatrix}
y \\
0
\end{bmatrix},
\]

where \( y = [y_1, \ldots, y_{N_P}] \), \( 1_v = [1, \ldots, 1] \), \( \tilde{\alpha} = [\tilde{\alpha}_1, \ldots, \tilde{\alpha}_{N_P}] \) and \( \Omega_{il} = \tilde{\phi}(x_i^D)^T \tilde{\phi}(x_l^D) \) for \( i, l = 1, \ldots, N_D \). According to Mercer’s condition, a kernel \( K(.) \) is represented as

\[
K(x_i^D, x_l^D) = \tilde{\phi}(x_i^D)^T \tilde{\phi}(x_l^D), \quad i, l = 1, \ldots, N_D.
\]

As a result, the output of the LS-SVMR approach for regression can be represented as

\[
y^D(x^D) = \sum_{i=1}^{N_D} \tilde{\alpha}_i K(x_i^D, x^D) + \tilde{b},
\]

where \( \tilde{\alpha}_i \) and \( \tilde{b} \) are the solution to Eq. (22). When a Gaussian function is used as a kernel function, the Eq. (24) can be written as

\[
y^D(x^D) = \sum_{i=1}^{N_D} \tilde{\alpha}_i \exp \left[ -\frac{\|x^D - x_i^D\|^2}{2\sigma^2} \right] + \tilde{b},
\]

where \( \sigma \) is termed as the GKP for the LS-SVMR. It notes that one drawback of LS-SVMR in comparison with standard SVMR is the lack of sparseness in the solution vector. Hence, in order to reduce the number of rule, the sparse LS-SVMR is adopted for the proposed approach. Besides, the sparse LS-SVMR algorithm can be found in [8]. And let \( \tilde{\alpha}_i = \tilde{b} / \exp \left[ -\frac{\|x^D - x_i^D\|^2}{2\sigma^2} \right] \) in Eq. (25), then Eq. (25) is represented as

\[
y^D(x^D) = \sum_{i=1}^{N_D+1} \tilde{\alpha}_i \exp \left[ -\frac{\|x^D - x_i^D\|^2}{2\sigma^2} \right],
\]

Because the \( \sigma_i \) in the SVMR can be fixed for all kernel function, we need to assume \( \sigma_1 = \sigma_2 = \cdots = \sigma_n = \sigma \) in the proposed fuzzy neural networks on stage I. Besides, from Eq. (26) and Eq. (8), the parameters \( \tilde{\alpha}_i \), \( N_D+1 \), \( 2\sigma^2 \) are equal to the parameters \( w_i \), \( l \), \( 2\sigma^2 \), respectively.
respectively. Hence, the weights $W_i$ and the number of rule $m$ of proposed fuzzy neural networks in Fig. 1 can be determined via the TSSVR.

4. Simulations

In this section, two examples are used to verify the effectiveness of the proposed approach. The index used for evaluating the performance is the root mean square error (RMSE) defined as

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (y_j - \hat{y}_j)^2}, \quad (27)$$

where $N$ denotes the number of the test data, $y_j$ represents the actual output and $\hat{y}_j$ is the output of the SVMR approach for the $j$-th training data.

Example 1: The training data sets are generated by [22]:

$$y = \frac{\sin x}{x}, \quad x \in [-10, 10]. \quad (28)$$

Assume the number of sampling data is 601, and the membership function is Gaussian function. Besides, we add the Gaussian noise and three outliers to this nonlinear function. The regularization constant $C$ is chosen as 1024 in the SVMR approach. The Gaussian kernel parameter $\sigma$ is suggested as $(0.2 \sim 0.5) \times \text{range}(x)$ in the SVMR approach. Then, the GKp is obtained as 0.4. For the hyperparameters (the regularization constant $\gamma$ and the Gaussian kernel parameter $\sigma$) of the robust LS-SVMR and sparse LS-SVMR, are obtained by 10 fold CV (cross validation). To verify the validation of the proposed approach, the sparse LS-SVMR and the robust+sparse LS-SVMR approach are considered to independently compare with proposed approach. The original training data set is shown in Fig. 4.

Fig. 5 shows results with the robust+sparse LS-SVMR approach for original training data set. From Fig. 5, it is obviously that the robust+sparse LS-SVMR approach is affected via outliers. If the original training data set directs to use the LS-SVMR approach, it needs 601 rules. At the same time, the outlier is also affected the SVMR approach. Hence, the SVMR approach applies to filter out the outliers and the sparse LS-SVMR under the results with the SVMR approach can reduce the number of rule in the proposed. After the proposed approach, the total number of rule is 47 and weights are determined in the proposed Sparse LS-SVMR. Besides, the comparison results among the sparse LS-SVMR approach, the robust+sparse LS-SVMR approach and the proposed approach for original training data set are shown in Fig. 6. Fig. 7 shows another case with 32 rules among above approaches. From Figs. 6 and 7, it is obviously that the performance of the proposed approach is superior to the robust LS-SVMR approach when the outliers are existed.

$$\gamma = 12817, \quad \sigma^2 = 0.8812$$

$$x(k) \text{ is the input sequence generated by equally sampling the interval } [-1.1] \text{ for } k = 0, 1, 2, \ldots, 50. \quad (29)$$

$$y(x[k]) = \sin(5x[k])\cos^{-1}(x[k])\cos(3x[k]-1), \quad \text{for } k = 0, 1, 2, \ldots, 50.$$
The regularization constant $C$ is chosen as 1063 and the Gaussian kernel parameter $\sigma$ is 0.8745. For the hyperparameters (the regularization constant $\gamma$ and the Gaussian kernel parameter $\tilde{\sigma}$) of the robust LS-SVMR and sparse LS-SVMR, are obtained by 10 fold CV (cross validation). To verify the validation of the proposed approach, the sparse LS-SVMR and the robust+sparse LS-SVMR approach are considered to independently compare with proposed approach. The original training data set is shown in Fig. 8.

Fig. 9 shows results with the robust+sparse LS-SVMR approach for original training data set. From Fig. 9, it is obviously that the robust+sparse LS-SVMR approach is affected via outliers. If the original training data set directs to use the LS-SVMR approach, it needs 401 rules.

At the same time, the outlier is also affected the LS-SVMR approach. Hence, the SVMR approach applies to filter out the outliers and the sparse LS-SVMR under the results with the SVMR approach can reduce the number of rule in the proposed. After the proposed approach, the total number of rule is 42 and weights are determined in the proposed Sparse LS-SVMR. Besides, the comparison results among the sparse LS-SVMR approach, the robust+sparse LS-SVMR approach and the proposed approach for original training data set are shown in Fig. 10. Fig. 11 shows another case with 22 rules among above approachs. From Figs. 10 and 11, it is obviously that the performance of the proposed approach is superior to the robust LS-SVMR approach when the outliers are existed. On the contrary, when this function uses FNNs with ARLA under random weights to get Figs. 6~7 and 10~11, it needs greater than 3500 epochs. That
is, the learning mechanism of the proposed approach for fuzzy neural network does not need iterated learning on weights for simplified fuzzy inference systems. Hence, the proposed method can fast determine the number of rule and weights for the FNNs without the robust learning algorithm.

5. Conclusions

In this study, the TSSVR approach is proposed to deal with training data set with outliers for the FNNs. There are two-stages strategies in the proposed approach that stage I and stage II use the SVMR and the sparse LS-SVMR, respectively. Besides, the proposed approach applies the tuning hyperparameters to replace learning mechanism for the FNNs. Consequently, the learning mechanism of the proposed approach for fuzzy neural network does not need iterated learning for the weight of simplified fuzzy inference systems. At the same time, the number of rule can be directly determined via the proposed approach for the FNNs.

Acknowledgements

This work was supported by National Science Council Under Grant NSC 96-2221-E-150-070-MY3.

References

Chen-Chia Chuang received the B.S. and M.S. degrees in Electrical Engineering from National Taiwan Institute of Technology, Taipei, Taiwan, in 1991 and 1993, respectively. He received Ph.D. degree in the Department Electrical Engineering at the National Taiwan University of Science and Technology, Taipei, Taiwan in 2000. He is currently an Associate Professor with the Department of Electrical Engineering, National Ilan University. His current research interests are neural networks, statistics learning theory, robust learning algorithm, and signal processing.

Jin-Tsong Jeng was born in Taiwan, R.O.C., in 1967. He received the B.S.E.E., M.S.E.E. and Ph. D. degrees all in Electrical Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1991, 1993, and 1997, respectively. He is currently a Professor in the Department of Computer Science and Information Engineering, National Formosa University, Huwei Jen, Yunlin, Taiwan. His primary research interests include neural networks, fuzzy system, intelligent control, support vector regression, magnetic bearing system, bio-informatics, non-holonomic control system and Microarray.

C. W. Tao received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1984, and the M.S. and Ph.D. degrees in electrical engineering from New Mexico State University, Las Cruces, in 1989 and 1992, respectively. He is currently a Professor with the Department of Electrical Engineering, National I-Lan University, I-Lan, Taiwan. His research interests are on the fuzzy neural systems including fuzzy control systems and fuzzy neural image processing.

Dr. Tao is an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics.