

The Bounded Capacity of Fuzzy Neural Networks (FNNs) Via a New Fully Connected Neural Fuzzy Inference System (F-CONFIS) With Its Applications

Jing Wang, Chi-Hsu Wang, *Fellow, IEEE*, and C. L. Philip Chen, *Fellow, IEEE*

Abstract—In this paper, a fuzzy neural network (FNN) is transformed into an equivalent three-layer fully connected neural inference system (F-CONFIS). This F-CONFIS is a new type of a neural network whose links are with dependent and repeated weights between the input layer and hidden layer. For these special dependent repeated links of the F-CONFIS, some special properties are revealed. A new learning algorithm with these special properties is proposed in this paper for the F-CONFIS. The F-CONFIS is therefore applied for finding the capacity of the FNN. The lower bound and upper bound of the capacity of the FNN can be found from a new theorem proposed in this paper. Several examples are illustrated with satisfactory simulation results for the capacity of the F-CONFIS (or the FNN). These include “within capacity training of the FNN,” “over capacity training of the FNN,” “training by increasing the capacity of the FNN,” and “impact of the capacity of the FNN in clustering Iris Data.” It is noted that the finding of the capacity of the F-CONFIS, or FNN, has its emerging values in all engineering applications using fuzzy neural networks. This is to say that all engineering applications using FNN should not exceed the capacity of the FNN to avoid unexpected results. The clustering of Iris data using FNN illustrated in this paper is one of the most relevant engineering applications in this regards.

Index Terms—Capacity of neural networks, fuzzy neural networks (FNNs), fuzzy system, Iris data, neural networks.

I. INTRODUCTION

IN the past decade, fuzzy neural networks (FNNs) have been widely used in many kinds of subject areas and engineering applications for problem solving, such as pattern recognition, intelligent adaptive control, regression or density estimation, and so on [1]–[6]. The FNN possesses the characteristics of linguistic information and the learning of a neural

network (NN) [7]–[12]. If the FNN is properly constructed, then it follows universal approximate theorem (UAT), i.e., the properly constructed FNN can approximate any nonlinear functions [13]–[16]. However, the universal approximate theorem does not show us how to properly construct and tune the FNN. This is to say that the FNN designed for certain applications by human expert must have constraints, such as the maximum number of input and output samples it can approximate or memorize. Similar to the discussions of the capacity of multilayer NNs [17], the capacity of the FNN is thus defined as the maximum number of arbitrary distinct input samples that can be mapped to desired output samples with a zero error. The overcapacity may lead training process to diverge in the FNN. The training samples should be independent. During the past decade, the capacity of an associative memory and multilayer perceptron (MLP) has been derived, assuming a fully connected NN [18]–[24].

However, it is challenging to find the capacity of the FNN because the links between the membership function layer and fuzzy rule layer are not fully connected, and the operators of the fuzzy rule layer of the FNN are the product form rather than the adding form in an NN. For we can find the capacity of the FNN by the method for a fully connected NN, the four-layer FNN is converted into an equivalent three-layer fully connected NN, namely, fully connected neural fuzzy inference System (F-CONFIS) in [25]. This F-CONFIS is a new type of NN, whose links are with dependent and repeated weights of membership functions in the premise part of the FNN. For these special properties of the F-CONFIS, its learning algorithm should be different from that of the conventional NNs with independent and different link weights. The initial discussion of the BP training for the F-CONFIS without a proper algorithm to find the backward path has been proposed in [26]. It is, thus, very important for this new type of the FNN (F-CONFIS) to develop a new complete training algorithm with special properties. Therefore, for these special dependent repeated links of the F-CONFIS, some special properties are revealed. A new and complete gradient descent training algorithm is thus proposed with these special properties in this paper. By using this new F-CONFIS in this paper, the lower bound and upper bound of the capacity of the FNN can be found. Several examples are illustrated to validate the theoretical capacity bounds and new gradient descent algorithm for the F-CONFIS, or the FNN. These include “within capacity training of the FNN,” “over capacity training of the FNN,” “increasing the capacity of the FNN to let the training process converge,” and “impact of the capacity of the FNN in Clustering Iris Data.” The clustering of Iris flower dataset using the FNN

Manuscript received December 4, 2012; revised April 23, 2013; accepted October 22, 2013. Date of publication November 26, 2013; date of current version November 25, 2014. This work was supported in part by the National 973 Basic Research Program of China under Grant 2011CB302801, Multiyear Research Grants, Macau Science and Technology Development Fund under Grant 008/2010/A1, and the National Science Council of Taiwan under Grant 102-2221-E-009-110.

J. Wang and C. L. P. Chen are with the Faculty of Science and Technology, University of Macau, Macau, China (e-mail: elizawangj@gmail.com; philip.chen@ieee.org).

C.-H. Wang is with the Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, and also an Adjunct Professor with Northeastern University at Qinhuangdao, Qinhuangdao, China (e-mail: cwang@cn.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TFUZZ.2013.2292972

illustrated in this paper is one of the most relevant engineering applications in this regard [27]–[29]. Excellent agreements have been achieved between the training simulations and theoretical bounds. The capacity using the Vapnik–Chervonenkis (VC) dimension (more precisely, the growth functions) is the maximum number of dichotomies that can be induced by a model. Both capacities are the measure of the complexity of a model. However, a VC dimension applies to only binary-valued functions. If the FNN has only one output node and the output belongs to one of two exclusive states, or binary states, then both capacities are the same. The pseudodimension is an extension of the VC-dimension to real-valued functions [30]. The relationship between the VC dimension or pseudodimension and the mapping capacity needs to be further studied.

In this paper, the major contributions are as follows: 1) After transforming the FNN into the F-CONFIS, some special properties of the F-CONFIS, which is a new type of the NN, were first developed and formulated. 2) We propose a new complete gradient descent training algorithm of the FNN via the F-CONFIS, which can handle not only the weight of the consequent part but also the repeated weights between the input layer and hidden layer of the premise part for FNN or F-CONFIS. 3) We find the lower bound and upper bound of the capacity of the FNN. 4) With the help of the theoretical capacity bounds in this paper, we provide a practical and effective method to adjusting the capacity of the F-CONFIS (or FNNs) in order to guarantee that the training process converges successfully.

The paper is organized as follows. In Section II, the equivalent three-layer F-CONFIS is briefly introduced, then a gradient descent learning algorithm of the F-CONFIS with special properties is proposed. The capacity of the F-CONFIS (or the FNN) is derived and formulated in Section III. In the next section, some effective methods of adjusting the capacity for the FNN are discussed. In Section V, examples to find the capacities for different configurations of the F-CONFIS are illustrated. The last section contains some conclusions.

II. FULLY CONNECTED NEURAL FUZZY INFERENCE SYSTEM (F-CONFIS) AND ITS SPECIAL LEARNING ALGORITHM

A. Fully Connected Neural Fuzzy Inference System (F-CONFIS)

Since the FNN is not a fully connected graph, discussions on finding its capacity were hindered. For we can find the capacity of the FNN by the method for a fully connected NN, the FNN has to be converted, that is, into an equivalent three-layer fully connected NN (or F-CONFIS) for the FNN as proposed briefly in [25]. For the convenience of a more detailed discussion of the capacity and new gradient descent training algorithm of the FNN, we have to summarize the derivation of the F-CONFIS in this section. The configuration of the FNN is shown in Fig. 1, four layers comprised. The first layer is an input layer, while its nodes correspond to input variables. The input vectors are $\{x_i^*, i = 1, 2, 3, \dots, N\}$. Layer II is a membership function layer, in that the value of a node quantifies the degree of the membership function of the input linguist variable. Layer III is a fuzzy rule layer. Therein each node represents a fuzzy rule. The last layer is an output layer. The output vectors are $\{y_k, k = 1, 2, \dots, M\}$.

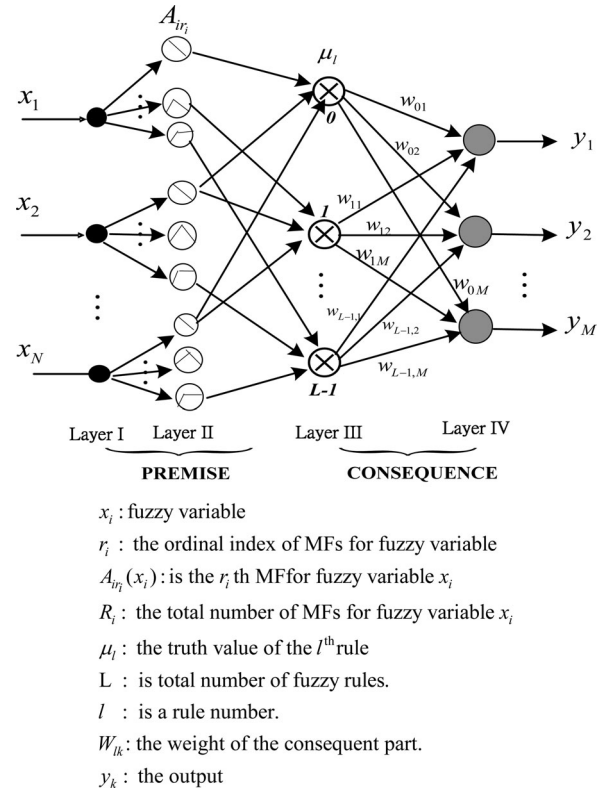


Fig. 1. Configuration of the FNN.

The consequent part is a fully connected graph, but apparently, the premise part is not, because not every node in a membership function layer connects to all nodes in the fuzzy rule layer.

The fuzzy Mamdani model is represented as [31] given next.

Rule l : IF x_1 is A_{1r_1} and \dots and x_N is A_{Nr_N} THEN y_1 is B_{l1} and \dots and y_M is B_{lM}

where $A_{1r_1}, \dots, A_{Nr_N}$ and B_{l1}, \dots, B_{lM} are the standard fuzzy sets. When the input variables are given, the truth value μ_l of the l th rule is defined by

$$\mu_l = A_{1r_1}(x_1) \times \dots \times A_{Nr_N}(x_N) \times \{l = 0, 1, 2, \dots, L - 1; r_i = 0, 1, 2, \dots, R_i - 1\}. \quad (1)$$

From the previous discussion, converting the FNN into a fully connected NN is necessary, however, will encounter two difficulties. One is that, in the FNN, the links between the membership function layer and fuzzy rule layer are not fully connected. The other is that the operators of the fuzzy rule layer of the FNN are the product form rather than the adding form.

To begin with, in order to overcome the first difficulty, in Fig. 1, the membership function layer of the FNN can be withdrawn, and substituted for new links between Layers I and III, so that we can have a fully connected three-layer NN as shown in Fig. 2. The new link weight $\{V_{ij} (i = 1, 2, \dots, n; j = 1, 2, \dots, L)\}$ in Fig. 2 is represented as between the i th node of the input layer and the j th node of the fuzzy rule layer. The equivalent solid line links substituted for the withdrawal of dash line links.

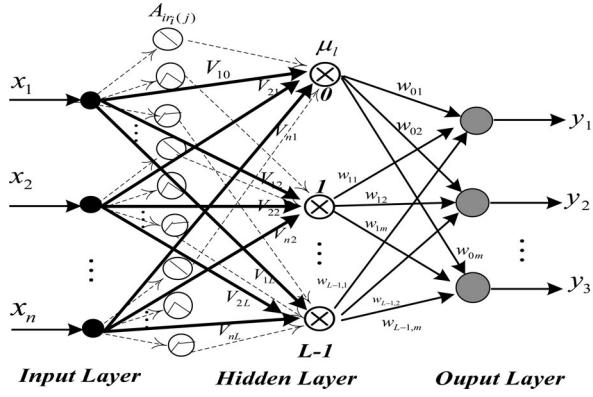


Fig. 2. Withdraw Layer II (MF layer) in the original FNN and substitute it for new links (solid line) between Layer I and Layer III.

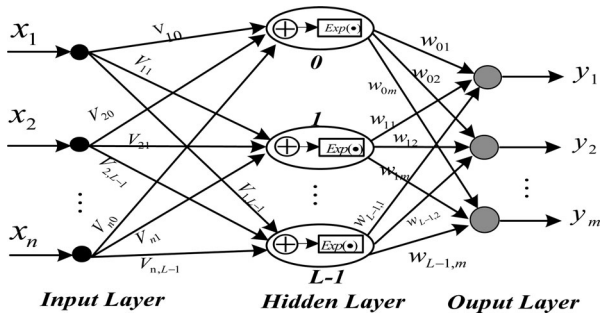


Fig. 3. Complete fully connected FNN, or F-CONFIS.

The next task is to overcome the second difficulty. The product form of the fuzzy rule layer will be converted into an equivalent adding form in Fig. 1. To do so, the new link weight V_{ij} ($i = 1, 2, \dots, N; j = 0, 1, \dots, L-1$) is defined as

$$V_{ij} = \text{Ln}(A_{ir_i(j)}(x_i)) \quad (2)$$

where the ordinal indices $r_i(l)$ ($i = 1, 2, \dots, N$) of membership functions correspond to rule l . If the exponential function was taken as activation function σ , then the output of the fuzzy rule layer is

$$\mu_j = \sigma \left(\sum_{i=1}^N \text{Ln}(A_{ir_i(j)}(x_i)) \right) = \sigma \left(\text{Ln} \left(\prod_{i=1}^N A_{ir_i(j)}(x_i) \right) \right) \quad (3)$$

Fig. 3 depicts the complete F-CONFIS for the FNN, and apparently it is equivalent to the original FNN in Fig. 1. The F-CONFIS has three layers as shown in Fig. 3, in which the hidden layer is also a fuzzy rule layer. Moreover, the three-layer fully connected NN (F-CONFIS) will be the foundation based on which we develop training algorithm and find the capacity of the FNN.

B. Special Learning Algorithm for the Fuzzy Neural Network Via a Fully Connected Neural Fuzzy Inference System

The neuro-fuzzy system is usually represented as special 4 or 5 layers multilayer feed-forward NN. There are different training algorithms with different emphasis [11], [32]–[34]. The gradient descent method is the most commonly used in

the learning algorithm for feed-forward NNs and fuzzy systems [35]–[41]. As aforementioned, the network architecture of the F-CONFIS is a standard fully connected three layers feed-forward NN except that there has repeated link weights between the input and hidden layer. This special property does not exist in common NN. The essential difference between the traditional gradient descent training algorithm for the FNN and the proposed one for the F-CONFIS lies in that the dependent repeated link weights between the input layer and hidden layer need special treatment.

The output layer of the F-CONFIS adopts center average defuzzification method as

$$y_k = \frac{\sum_{j=0}^{L-1} \mu_j w_{jk}}{\sum_{j=0}^{L-1} \mu_j}, (k = 1, 2, \dots, M) \quad (4)$$

where the number of outputs is M , the output vectors are $\{y_k, k = 1, 2, \dots, M\}$, μ_l is the truth value of the l th rule, and w_{jk} is the weighting vector between the fuzzy rule layer and output layer. The number of training patterns is P , the total training error [42] is given as

$$E(W) = \frac{1}{2P} \sum_{p=1}^P \sum_{k=1}^M (y_k^p - d_k^p)^2 \quad (5)$$

where y_k^p is the k th actual output, d_k^p is the k th desired outputs. In the F-CONFIS, the Gaussian membership function is adopted:

$$A_{ij}(x) = \exp \left[\frac{-(x - m_{ij})^2}{2\sigma_{ij}^2} \right], i = 1, \dots, n; j = 0, \dots, R_i \quad (6)$$

where m_{ij} and σ_{ij} denote the center and width of the membership functions, respectively.

In the F-CONFIS (or FNN), there are two types of parameters needed to be tuned. One is premise parameters, namely the parameters of MFs (i.e., the means and the width of the Gaussian function). The other is consequent parameters. Therefore, weighting vector W is defined as

$$W = [m_{ij} \quad \sigma_{ij} \quad w_{jk}] \quad i = 1, \dots, n; \quad j = 1, \dots, R_i; \quad k = 1, \dots, M \quad (7)$$

where w_{jk} is the link weights between the fuzzy rule layer and output layer. The gradient of $E(W)$ with respect to W is

$$g = \left[\frac{\partial E}{\partial m_{ij}} \quad \frac{\partial E}{\partial \sigma_{ij}} \quad \frac{\partial E}{\partial w_{jk}} \right] \quad i = 1, \dots, n; \quad j = 1, \dots, R_i; \quad k = 1, \dots, M. \quad (8)$$

The training process is to update W in $-g$ direction to minimize total squared error $E(W)$. The weighting vector W can be tuned as

$$W(\alpha, \beta) = W - \left[\alpha \frac{\partial E}{\partial m_{iq}}, \alpha \frac{\partial E}{\partial \sigma_{iq}}, \beta \frac{\partial E}{\partial w_{jk}} \right] \times (i = 1, \dots, N; q = 0, \dots, R_i - 1; k = 1, \dots, M) \quad (9)$$

where α and β are the learning rate of the premise part and the consequent part in the F-CONFIS, respectively. From (4) and

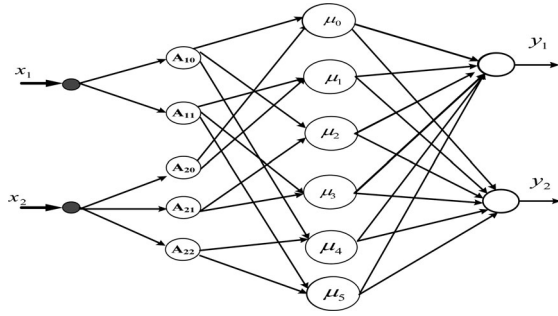


Fig. 4. FNN with two input variables, two output variables.

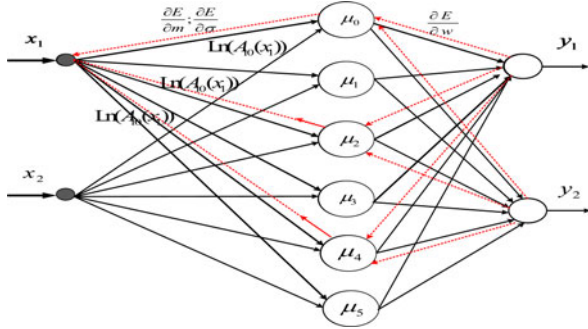


Fig. 5. Gradient descent flow (dotted lines) for the first MF of x_1 of F-CONFIS with repeated link eights.

(5), by the chain rule, we have

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k^p} \frac{\partial y_k^p}{\partial w_{jk}} = \frac{1}{P} \sum_{p=1}^P (y_k^p - d_k^p) \mu_l \Big/ \sum_{j=1}^L \mu_l$$

$$j = 0, \dots, L-1; \quad k = 1, \dots, M. \quad (10)$$

From (1), (2), and (4)–(6), by a series of the chain rule, we have

$$\begin{cases} \frac{\partial E}{\partial m_{iq}} = \frac{1}{P} \sum_{p=1}^P \sum_{k=1}^M \frac{(y_k^p - d_k^p)(w_{jk} - y_k^p) \mu_l(x_i^p - m_{iq})}{\sum_{j=0}^{L-1} \mu_j \sigma_{iq}^2} \\ \frac{\partial E}{\partial \sigma_{iq}} = \frac{1}{P} \sum_{p=1}^P \sum_{k=1}^M \frac{(y_k^p - d_k^p)(w_{jk} - y_k^p) \mu_j(x_i^p - m_{iq})^2}{\sum_{j=0}^{L-1} \mu_j \sigma_{iq}^3} \\ i = 1, \dots, N; q = 0, \dots, R_i - 1. \end{cases} \quad (11)$$

For a normal multilayer network, every weight is updated only once to reduce the error in the gradient descent process of each epoch. The configuration of the FNN is shown in Fig. 4 and the equivalent F-CONFIS is shown in Fig. 5. The configuration of the FNN (or F-CONFIS) has two input variables and two output variable, two MFs $\{A_{10}(x_1), A_{11}(x_1)\}$ for x_1 and three MFs $\{A_{20}(x_2), A_{21}(x_2), A_{22}(x_2)\}$ for x_2 . In Fig. 5, the dotted line is the gradient descent path for the first MF of x_1 , namely, $Ln(A_{10}(x_1))$. From Fig. 5, it is shown that the control parameters of $Ln(A_{10}(x_1))$ will be updated three times, the first time when the gradient of the control parameters propagated form μ_0 to x_1 , the second time when the gradient of the control parameters propagated form μ_2 to x_1 , and the third time when the gradient of the control parameters propagated form μ_4 to x_1 . $Ln(A_{10}(x_1))$

are also handled three times. Each centers ($m_{iq}, q = 0, \dots, R_i - 1$) and spreads ($\sigma_{iq}, q = 0, \dots, R_i - 1$) of Gaussian MFs will update the number of repeated links $NRL(i)$ for each fuzzy variable x_i . This multiple updates should be processed carefully. From the previous illustration, it is important to find the number of repeated links $NRL(i)$ for each fuzzy variable x_i in premise part so that the gradient descent training algorithm can properly be carried out for FNN, or F-CONFIS. The following Theorem 1 will show a precise formula of finding the number of repeated links $NRL(i)$ in the F-CONFIS (or FNN).

Theorem 1: In the FNN (or F-CONFIS), let R_i be the number of MFs for fuzzy variable x_i , then each fuzzy variable x_i will have $NRL(i)$ repeated link weights between the input layer and hidden layer with repeated link weights equal to $\{A_{iq} | q = 0, 1, 2, \dots, R_i - 1\}$, where

$$NRL(i) = L/R_i \quad (i = 1, \dots, N). \quad (12)$$

Proof: From (1), the total number of fuzzy rules is

$$L = \prod_{i=1}^n R_i \quad (13)$$

where R_i being the number of MFs for fuzzy variable x_i . Since the F-CONFIS is an equivalent fully connected network, there are L links from every node of the input layer to all of nodes of the fuzzy rule layer.

However, there are only different R_i MFs for x_i

$$\{A_{iq} | q = 0, 1, 2, \dots, R_i - 1\}.$$

Therefore, there must have links come with repeated weights. In the FNN, or F-CONFIS, it is quite obvious that for each fuzzy variable x_i , the number of repeated link weights with repeated link weights equal to $\{A_{iq} | q = 0, 1, 2, \dots, R_i - 1\}$.

$$NRL(i) = L/R_i.$$

Q.E.D.

To be compatible with l defined in [42], the ordinal index of fuzzy rules should start from zero ($0 \leq l \leq \prod_{i=1}^N R_i - 1$). Therefore, ordinal index l of the first fuzzy rule is

$$l = r_1 + \sum_{i=2}^N \left(\prod_{j=1}^{i-1} R_j \right) r_i \quad (14)$$

where r_i ($0 \leq r_i \leq R_i - 1$) is the ordinal index for the r_i th MF for fuzzy variable x_i . The range of l is $0 \leq l \leq (\prod_{i=1}^N R_i) - 1$. From (14), all the r_i 's for a specific rule number l are represented as

$$r_1(l) = l \% R_1, \quad r_i(l) = \left(l / \prod_{k=1}^{i-1} R_k \right) \% R_i, \quad i = 2, \dots, N \quad (15)$$

where $A \% B$ denotes the remainder and A/B denotes the quotient of the division of A over B .

For the j th hidden node, by (15), we can get the corresponding n ordinal index $r_i(j)$ ($i = 1, 2, \dots, N; j = 0, 1, \dots, L-1$) of

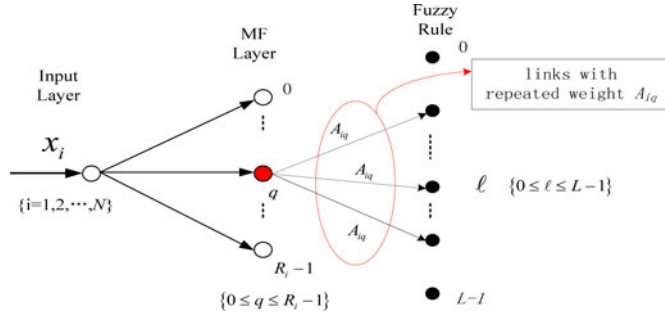


Fig. 6. Finding the links with repeated weights.

MFs for x_i . Therefore, (11) is revised as follows:

$$\begin{cases} \frac{\partial E}{\partial m_{ir_i(j)}} = \frac{1}{P} \sum_{p=1}^P \sum_{k=1}^M \frac{(y_k^p - d_k^p)(w_{jk} - y_k^p) \mu_j(x_i^p - m_{ir_i(j)})}{\sum_{j=0}^{L-1} \mu_j \sigma_{ir_i(j)}^2} \\ \frac{\partial E}{\partial \sigma_{ir_i(j)}} = \frac{1}{P} \sum_{p=1}^P \sum_{k=1}^M \frac{(y_k^p - d_k^p)(w_{jk} - y_k^p) \mu_j(x_i^p - m_{ir_i(j)})^2}{\sum_{j=0}^{L-1} \mu_j \sigma_{ir_i(j)}^3} \end{cases} \quad (16)$$

$i = 1, \dots, N; j = 0, \dots, L-1.$

It is important to find the links with repeated weight from the input node to hidden nodes so that we can properly carry out the gradient vector of algorithm for the F-CONFIS, or the FNN. Fig. 6 illustrates this scenario. From Fig. 6, we know that the repeated weight for the q th MF of x_i is equal to $A_{iq} = A_{ir_i(\ell)}$. Therefore, we have

$$q = r_i(\ell_k) \Rightarrow \{\ell_k = h_k(i, q); k = 1, \dots, \text{NRL}(i)\} \quad (17)$$

where $\{\ell_k = h_k(i, q); k = 1, \dots, \text{NRL}(i)\}$ in (17) are the links with repeated weight A_{iq} for the q th MF of x_i .

Theorem 2: The gradient components of the F-CONFIS in the premise part should be summed $\text{NRL}(i)$ times for fuzzy variable x_i . The gradient components $\partial E/\partial m_{iq}$ and $\partial E/\partial \sigma_{iq}$ of g in (11) have the following forms:

$$\begin{cases} \frac{\partial E}{\partial m_{iq}} = \sum_{k=1}^{\text{NRL}(i)} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial m_{iq}} \\ \frac{\partial E}{\partial \sigma_{iq}} = \sum_{k=1}^{\text{NRL}(i)} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial \sigma_{iq}} \end{cases} \quad (18)$$

$i = 1, \dots, N; q = 1, \dots, R_i.$

Proof: By Theorem 1, we know that each fuzzy variable x_i will have $\text{NRL}(i)$ repeated links between the input layer and the hidden layer. Let $A_{iq} (q \in [1, R_i-1])$ be the q th MF of the fuzzy variable x_i , and the weights of the repeated links are $V_{ih_k(i,q)} (k = 1, \dots, \text{NRL}(i))$. Let m_{iq} and σ_{iq} be the parameters of A_{iq} . In the error gradient decent process, error will propagate along interconnection $V_{ih_k(i,q)} (k = 1, \dots, \text{NRL}(i))$ to x_i to get error's partial derivative with respect to m_{iq} and σ_{iq} . From (1)–(4), by chain rule, we can get these partial derivate as

$$\begin{cases} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial m_{iq}} \\ \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial \sigma_{iq}} \end{cases}, k = 1, \dots, \text{NRL}(i)$$

All these intermediate partial derivatives should be summed up to give the final update for m_{iq} and σ_{iq} as the following equations:

$$\begin{cases} \frac{\partial E}{\partial m_{iq}} = \sum_{k=1}^{\text{NRL}(i)} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial m_{iq}} \\ \frac{\partial E}{\partial \sigma_{iq}} = \sum_{k=1}^{\text{NRL}(i)} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial \sigma_{iq}} \end{cases} \quad i = 1, \dots, N; q = 1, \dots, R_i$$

Q.E.D.

The components of W are finally updated by the following equations:

$$\begin{cases} m_{iq}(t+1) = m_{iq}(t) - \alpha \sum_{k=1}^{\text{NRL}(i)} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial m_{iq}} \\ \sigma_{iq}(t+1) = \sigma_{iq}(t) - \alpha \sum_{k=1}^{\text{NRL}(i)} \frac{\partial E}{\partial \mu_{h_k(i,q)}} \frac{\partial \mu_{h_k(i,q)}}{\partial A_{iq}} \frac{\partial A_{iq}}{\partial \sigma_{iq}} \\ w_{jk}(t+1) = w_{jk}(t) - \beta \frac{\partial E(W)}{\partial w_{jk}} \end{cases} \quad (19)$$

$i = 1, \dots, N; q = 0, \dots, R_i-1;$
 $j = 0, \dots, L-1, k = 1, \dots, M$

where t is the number of iterations. The following Algorithm I describes the complete gradient descent algorithm for the F-CONFIS.

Algorithm I: A new gradient descent algorithm for FNN via F-CONFIS

Step 1: Initializing the weight vector, learning rates. Let t be iteration count, $t = 1$.

Step 2: Calculate truth value μ_l by (1), output y_k by (4), and ordinal index l of all fuzzy rule by (14).

Step 3: calculates the update changes of the parameters in the premise part, i.e. the overall update changes of the centers and spreads of the Gaussian MFs.

- 1) Let hidden node number $l = 0$.
- 2) For training input data $i = 1, \dots, N$
 - a) By (15), get the ordinal index $r_i(l)$ of membership functions for the i^{th} fuzzy variable.
 - b) Find the gradient components $\partial E/\partial m_{ir_i(l)}$ and $\partial E/\partial \sigma_{ir_i(l)}$ in premise part via (16).
 - c) From Theorem 1, find the number of repeated link weights $\text{NRL}(i)$. Since every repeated A_{iq} weight will be repeated for $\text{NRL}(i)$ times, we have to sum the update changes of $\partial E/\partial m_{iq}$ and $\partial E/\partial \sigma_{iq}$ with the same subscripts for i^{th} fuzzy variable x_i for $\text{NRL}(i)$ times from Theorem 2.
- 3) If l less than to the total number of fuzzy rules L , then $l = l + 1$, go to 2), else go to next step 4

Step 4: Calculate the update changes of consequent part, i.e., $\partial E(W)/\partial w_{jk}$ by (10).

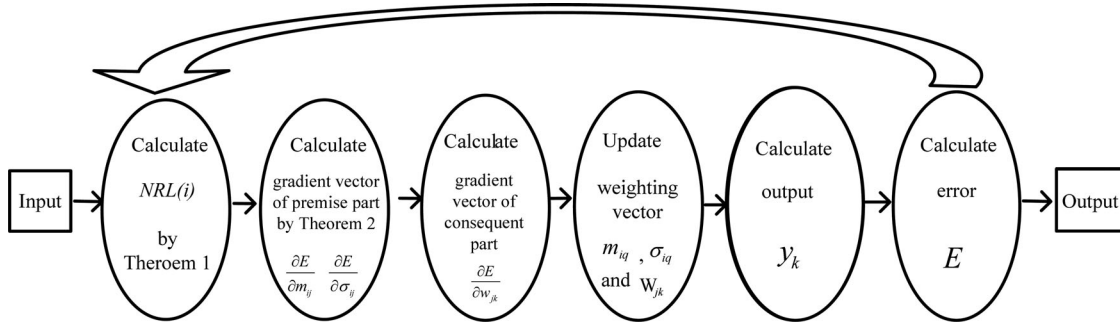


Fig. 7. Gradient descent training process of the FNN via F-CONFIS.

Step 5: Update the components of weighting vectors w_{jk} , m_{iq} and σ_{iq} by (19).

Step 6: Calculate error E in (5)

Step 7: If error E is less than error tolerance or iteration t reaches maximum iterations, go to step 8, else iteration $t = t + 1$; go to step 2.

Step 8: Stop.

Fig. 7 shows the complete gradient decent training process

III. CAPACITY OF FUZZY NEURAL NETWORK OR FULLY CONNECTED NEURAL FUZZY INFERENCE SYSTEM

The capacity of the FNN is thus defined as the maximum number of arbitrary distinct input samples that can be mapped to desired output samples with a zero error. All engineering applications using the FNN should not exceed the capacity of the FNN to avoid unexpected results. For we can find the capacity of the FNN by the method for a fully connected NN, in the Section II, the FNN has been converted into an equivalent three-layer fully connected NN (or F-CONFIS) and hence both have the same capacity. The capacity of the three-layer F-CONFIS with exponential activation functions in the hidden layer will be discussed in this section. As follows, both upper and lower bound of the three-layer F-CONFIS, or FNN, can be determined by Theorem 3.

Theorem 3: For the FNN defined in Fig. 1, where there are N inputs, M outputs, and L neurons in the fuzzy layer (or the hidden layer), each fuzzy variable x_i has R_i MFs, each MFs has N_{cp} parameters, then the capacity P of the FNN can be bounded by the following inequality:

$$L \leq P \leq \left(\sum_{i=1}^N R_i N_{cp} + LM \right) / M. \quad (20)$$

Proof: In [24], it has been shown that standard single-hidden layer feed-forward networks (SLFNs) with at most L hidden neurons and with any bounded nonlinear activation function which has limit in either direction can learn L distinct samples with zero error. This implies that the SLFN with L hidden units can memorize at least L distinct patterns. For the F-CONFIS, the exponential activation function in the hidden layer satisfies the condition of activation function in [24], and therefore, the lower bound on the capacity P is the number of hidden nodes L , i.e.,

$$P \geq L. \quad (21)$$

Furthermore, it has been shown in [17] that the upper bound for the capacity of feed-forward networks with arbitrary hidden unit activation functions can be found. For a feed-forward network (without bias) with N inputs, M outputs, L hidden units, and arbitrary hidden activation functions, the number of patterns P that can be memorized with no error is less than or equal to $L(N + M)$, divided by the number of outputs M . Then, the upper bound for the capacity P of the SLFN is

$$P \leq L(N + M)/M. \quad (22)$$

Following (22), we have $P \leq L(N + M)/M = (LN + LM)/M$, in which LN denotes the amount of independent links between the input and hidden layer, while LM denotes the amount of independent links between hidden and output layer. However, the link weights between input and hidden layer in the F-CONFIS are dependent on input variables. Therefore, we should find the total number of free parameters between the input and hidden layer to find the upper bound of the capacity of the F-CONFIS. Assume N_{cp} parameters are identical for every MF in the FNN, e.g. Gaussian MFs ($= \text{Exp}(-(x - \mu)^2/2\sigma^2)$), where parameters are mean and width (or variance), thus N_{cp} is 2 for this case. Then, the number of independent parameters (NICP) between the input and hidden layer is

$$\text{NICP} = \sum_{i=1}^N R_i N_{cp}. \quad (23)$$

Replacing NL in (23) with NICP, we have the following upper bound for the capacity P of F-CONFIS:

$$P \leq \left(\sum_{i=1}^N R_i N_{cp} + LM \right) / M. \quad (24)$$

From (21) and (24), we can conclude that

$$L \leq P \leq \left(\sum_{i=1}^N R_i N_{cp} + LM \right) / M.$$

Q.E.D.

From the previous Theorem 3, we can find the lower and upper bound of the capacity of the F-CONFIS, or the FNN. It is important to find the capacity of the FNN so that engineering applications via the FNN can be achieved without unexpected results.

IV. SOME APPLICATION ISSUES FOR THE CAPACITY OF THE FUZZY NEURAL NETWORK

For a NNs, it is impossible to map an unlimited number of input/output vectors. This issue is also valid for the FNN. In this paper, boundary of the capacity of the F-CONFIS, or the FNN, is determined by Theorem 3. In practical applications, there are several issues in training process related with F-CONFIS (FNN) capacity: 1) training when samples size is less than the lower bound; 2) training when the samples size is larger than the upper bound; 3) how to change the capacity of the FNN, when the training process fails to converge. For a FNN, when the number of training patterns is less than the lower bound, the training process will converge successfully, while the training patterns are larger than upper bound, the training process will diverge. It should be noted that the capacity is the basic property of a FNN, which has nothing to do with the type of training patterns. This is identical with the fact that the stability of a control system is unrelated with the type of input signals. In order to fully check the validity of the theoretical results, it is better that the set of training data be uncorrelated.

In the practical application of FNNs, we may encounter the case that training process cannot converge successfully, regardless of increasing the number of training iterations. This is due to the training samples size is larger than the upper bound of the capacity of the FNN. To guarantee the training convergence, it is better to increase the capacity of the FNN. According to Theorem 3, there are four ways to increase the capacity of the FNN: 1) Increase the number of input fuzzy variables; 2) increase the numbers of MFs for input fuzzy variables; 3) increase the number of parameters of MFs; 4) decrease the number of output variables. It is obvious that the second way to increase the numbers of MFs of input fuzzy variables is the most efficient and feasible way. All the other ways are limited by the constraints of physical applications, which cannot be altered at will to increase the capacity of the FNN.

For example, assuming that there are 150 independent training patterns and the configuration of the FNN (or F-CONFIS) has four input variables and one output variable, i.e., $N = 4$, $M = 1$. Let every linguistic variable have two MFs, namely, $R_1 = R_2 = R_3 = R_4 = 2$. Therefore, there are $2^4 = 16$ rules, i.e., $L = 16$. We adopt Gaussian functions as the membership functions, so that the number of parameters is 2, i.e., $N_{cp} = 2$. Boundary of the capacity P of the FNN is determined by Theorem 3 ($L \leq P \leq (\sum_{i=1}^n R_i N_{cp} + LM)/M$), that is, $16 \leq P \leq 32$. Since the number of training pattern is 150, which is larger than the capacity. In such a case, training process cannot converge, regardless of increasing the number of training iterations. The training process will thus be failed. To guarantee the training convergence, it is better to increase the capacity of the FNN. As stated in the above discussion, the most efficient and feasible way is the second way of increasing the capacity of the FNN, which is to increase the numbers of MFs of input fuzzy variables. Therefore, in a new architecture of the FNN (or F-CONFIS), we let $R_1 = R_2 = 3$, $R_3 = R_4 = 4$. Therefore, there are $3^2 4^2 = 144$ rules. Boundary of the capacity P of the FNN is determined by Theorem 3, that is, $144 \leq P \leq 172$. Therefore, the number

of training samples (150) is within the capacity bounds which will yield a successful training process.

It should be pointed out that there are two kinds of capability for learning machine, i.e., learning and generalization. The learning concerns the error rate on training data, while the generalization concerns the error rate on test data [44], [45]. In this paper, we discuss only the learning capability. According to the result in this paper, by increasing the capacity of the FNN, we can always get small training error. But small training error does not necessarily mean small test error. In statistical learning theory, the VC dimension [46], [47] is a measure of the capacity of a learning machine. With VC dimension, we can predict a probabilistic upper bound on the test error of a classification model. The expected risk $R(\alpha)$ with probability $1 - \eta$, the following bound holds [41]:

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log(2l/h) + 1 - \log(\eta/4))}{l}} \quad (25)$$

where $R_{\text{emp}}(\alpha)$ is an empirical risk, l is number of samples, and h is the VC dimension. A system with low empirical risk and high VC dimension may encounter overfitting problem. In our paper, given a training set, we should design the architecture of the FNN to let the number of samples falls within the lower and upper bound given by Theorem 3. If the upper bound is less than the number of training samples, the empirical risk may be large; if the lower bound is larger than the number of training samples, although the empirical risk may converge to zero, the generalization error will be deteriorated. This scheme is also in accordance with the principle of Occam's razor [48], [49].

In this paper, the capacity of the FNN is defined as the maximum number of arbitrary distinct input samples that can be mapped to desired output samples with zero error. The capacity using VC dimension (more precisely, the growth functions) is the maximum number of dichotomies that can be induced by a model. Both capacities are the measure of the complexity of a model. However, the VC dimension applies to only binary-valued functions. If the FNN has only one output node and the output belongs to one of two exclusive states, or binary states, then both capacities are the same. The pseudo-dimension is an extension of the VC-dimension to real-valued functions [30]. The relationship between VC dimension or pseudodimension and the mapping capacity needs to be further studied.

V. ILLUSTRATED EXAMPLES

In Section IV, we have found the lower and upper bound of the capacity of the F-CONFIS. This means that the number of training samples has significant impact on training convergence. When the number of training sample is less than the lower bound, training process is guaranteed to converge successfully. On the other hand, if the number of training sample size is larger than the upper bound, then the training process will fail to converge. All applications using the FNN should not exceed the capacity of the FNN to have unexpected results. Four examples will be to further discuss this phenomenon. Example 1 will perform the validation of "Within Capacity training of F-CONFIS," in which the training sample size is less than the

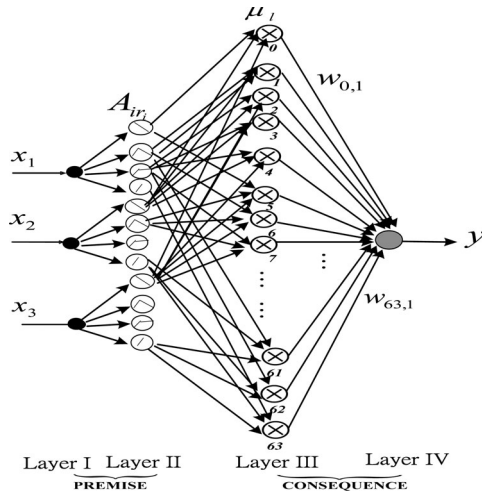


Fig. 8. Original configuration of the FNN has three input variables, where each variable has four MFs.

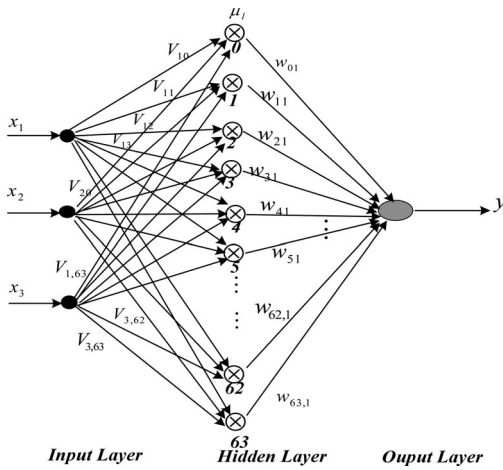


Fig. 9. Transformed complete fully connected F-CONFIS has three input variables, where each variable has four MFs.

lower bound of the FNN capacity. Example 2 will illustrate the “Overcapacity training of F-CONFIS,” in which the training sample size is larger than the upper bound of the FNN capacity. Example 3 will demonstrate a method to deal with overcapacity issue by increasing the capacity of the F-CONFIS. Example 4 will further illustrate the influence of the capacity of the FNN in clustering the Iris flower dataset [27].

In order to fully check the validity of the theoretical results, it is better that the set of training data be uncorrelated. Examples 1–4 share the same random (uncorrelated) dataset and FNN configuration.

A. Example 1: Within Capacity Training of Fuzzy Neural Network—The training Process Convergent Successfully.

In this example, the pattern number is less than the lower bound of the capacity of the FNN. The configuration of the FNN is shown in Fig. 8, by which the equivalent F-CONFIS shown in Fig. 9. There are three input variables ($N = 3$) and one output variable ($M = 1$). Each linguistic variable has four

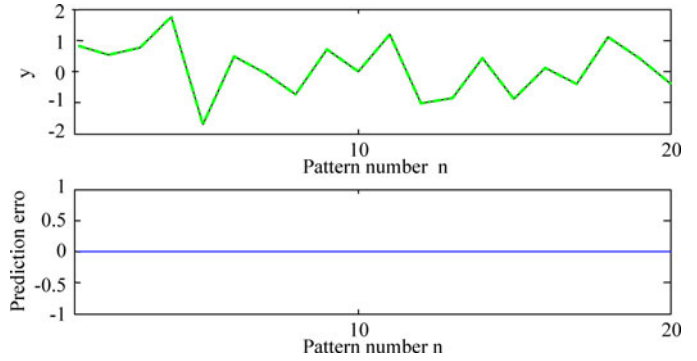


Fig. 10. Within capacity training of the FNN in Example 1, total pattern number = 20.

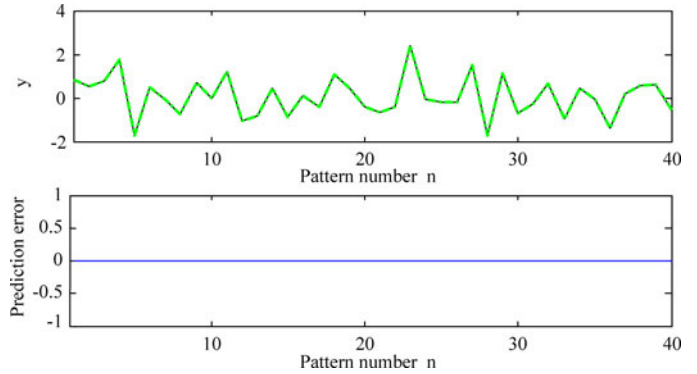


Fig. 11. Within capacity training of the FNN in Example 1; total pattern number = 40.

membership functions. Therefore, hidden nodes of the fuzzy rule layer are 64 corresponding to 64 fuzzy rules ($L = 64$). The membership functions are the Gaussian functions, the output node is a linear summing unit. We apply algorithm I to conduct gradient descent training algorithm of the FNN, or the F-CONFIS in the following examples. The transformed complete fully connected F-CONFIS has three input variables, each variable with four MFs. Both the input (x_1, x_2, x_3) and output (y) training data are random data generated by MATLAB function *normrnd()*. Since both the input and output training data are random data, there is no relationship between the input and output. We cannot expect the F-CONFIS model to learn any underlying relationship from the training dataset. We make this on purpose to prove that if the lower bound of the FNN capacity is larger than samples size n , then the FNN can map any n pairs of data, even if they are totally uncorrelated. The original FNN is shown in Fig. 8. Fig. 9 shows the equivalent F-CONFIS. In this case, $n = 3$, $L = 64$, $R_1 = R_2 = R_3 = 4$, $N_{cp} = 2$, and $M = 1$. Boundary of the capacity P of the FNN is determined by Theorem 3 ($L \leq P \leq (\sum_{i=1}^n R_i N_{cp} + LM)/M$), that is, $64 \leq P \leq 88$. When the number of training pairs is less than the lower bound, the training error will converge successfully. Let the number of training patterns be 20, 40, and 60. Figs. 10–12 show the experimental results. On each figure, the top curve is the F-CONFIS (or FNN) output (green dashed line) over training data (black solid line) and the bottom curve is the prediction process between the F-CONFIS output and original training

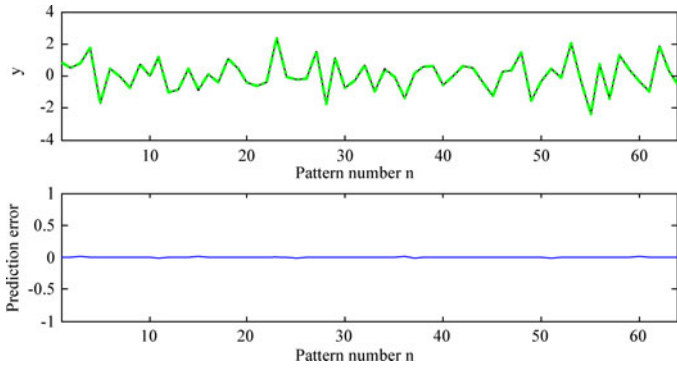


Fig. 12. Within capacity training of FNN in Example 1; total pattern number = 64.

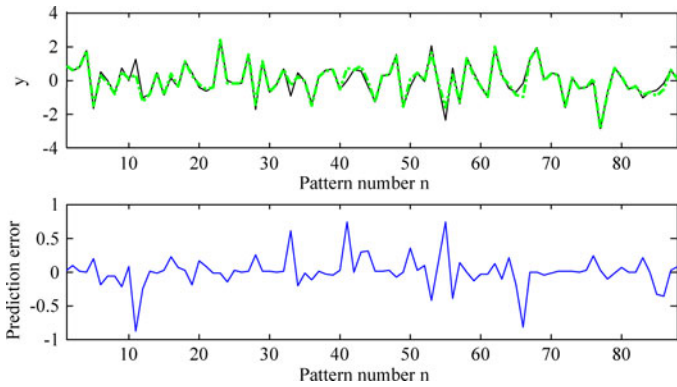


Fig. 13. Overcapacity training of the FNN in Example 2; total pattern number = 88 \geq 88.

data. From Figs. 10–12, it is shown that if the pattern number is less than the lower bound ($=64$), the prediction error can be made as close as possible to zero. The difference between the original training data and the output of the F-CONFIS is trivial, thus, we can only see one curve in the top parts of Figs. 10–12.

From Example 1, it is shown that when the numbers of training patterns is less than the lower bound, the training error converges to zero and training process is successful.

B. Example 2: Overcapacity Trainings of the Fuzzy Neural Network—Training Process Converge Unsuccessfully

In this case, the pattern number is great than the upper bound of the capacity of the FNN. This example is to illustrate the overcapacity training of the FNN. The upper bound of the F-CONFIS, or the FNN, can be determined by Theorem 3 in Fig. 8, that is, 88. This is to say, when the samples size is larger than the upper bound ($= 88$) and training samples are uncorrelated, the training error will not converge to zero successfully. Let the number of training patterns be 88, 100, and 110. Figs. 13–15 show the experimental results.

From Figs. 13–15, it is shown that all the four training processes failed to converge. Table I summarizes the result of MSE (mean square error) versus the number of training patterns. It is obvious that the MSE increases as the number of training patterns is increased. The training process converges unsuccessfully when the number of training patterns is over the capacity of

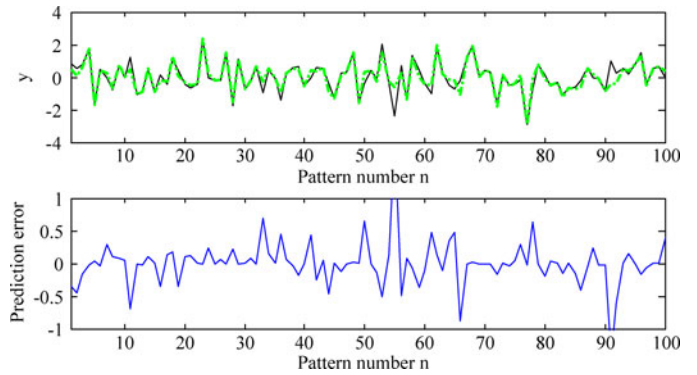


Fig. 14. Overcapacity training of the FNN in Example 2; total pattern number = 100 \geq 88.

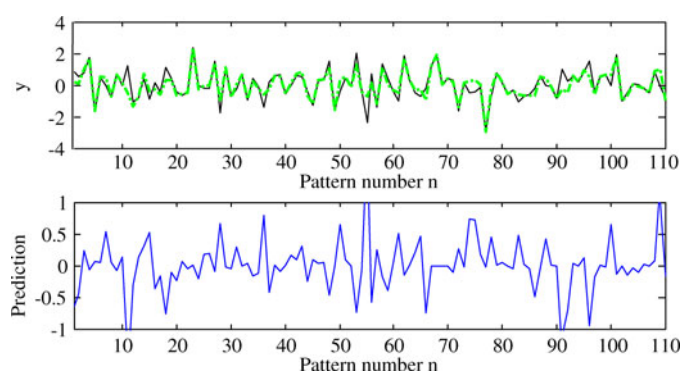


Fig. 15. Overcapacity training of the FNN in Example 2; total pattern number = 110 \geq 88.

TABLE I
RELATIONSHIP BETWEEN THE MSE AND PN FOR OVERCAPACITY TRAINING OF F-CONFIS

Pattern Number (PN)	MSE
PN=88	0.025924
PN=100	0.058892
PN=110	0.08591

the FNN. Therefore, all engineering applications using the FNN should not exceed the capacity of the FNN to avoid unexpected results.

C. Example 3: Increasing the Capacity of the Fuzzy Neural Network to Let the Training Process Convergent Successfully.

In Example 2, it is shown that when the number of training pattern is larger than the upper bound ($= 88$) of the FNN capacity, the training process will not converge. In this case, increasing the number of training pattern is helpless and on the contrary, will increase the training error. With the help of Theorem 3, we can handle this issue by increasing the capacity of the FNN.

To make the lower bound larger than 88, let each linguistic variable has five MFs, then there are 125 nodes in the hidden layer ($L = 125$). The boundary of the capacity P of the FNN is determined by Theorem 3, that is, $125 \leq P \leq 155$. The lower

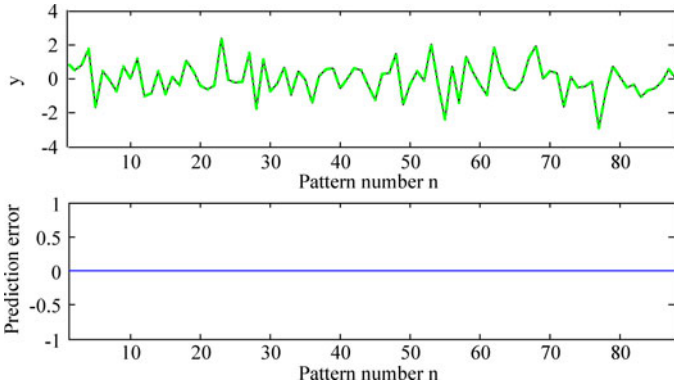


Fig. 16. Increasing the capacity of the FNN in Example 3; total pattern number = 88.

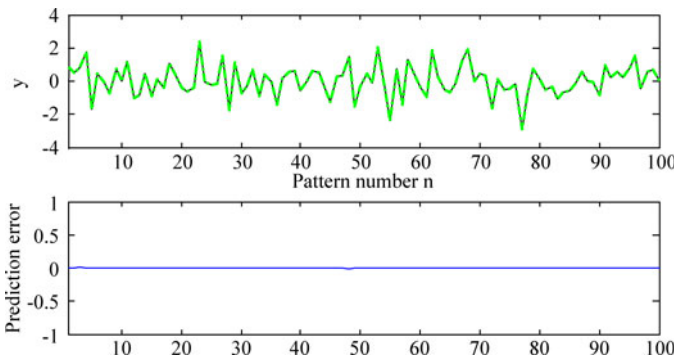


Fig. 17. Increasing the capacity of the FNN in Example 3; pattern number = 100.

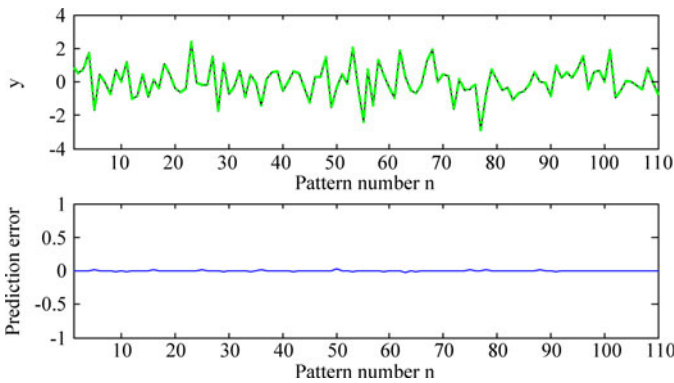


Fig. 18. Increasing the capacity of the FNN total pattern in Example 3; pattern number = 110.

bound of the FNN will now become 125. With this new FNN, we reconstruct the experiments in Example 2. Let the number of training patterns be 88, 100, and 110 again. Figs. 16–18 show the results.

It can be seen that the prediction error in Figs. 16–18 is as close as possible to zero. Table II summarizes the result of MSE versus the number of training patterns. The training process can not converge successfully in Example 2 when the number of training patterns is over the capacity of the FNN. From Example 3, we increased the capacity of the FNN so that the number of training patterns is within the capacity of the FNN. Then, the

TABLE II
RELATIONSHIP BETWEEN THE MSE AND PN AFTER INCREASING THE CAPACITY OF AN FNN

Pattern Number (PN)	MSE
PN=88	7.6911e-008
PN=100	3.7116e-006
PN=110	2.013e-005



Fig. 19. Iris setosa (left), Iris virginica (middle), and Iris versicolor (right) of Iris data.

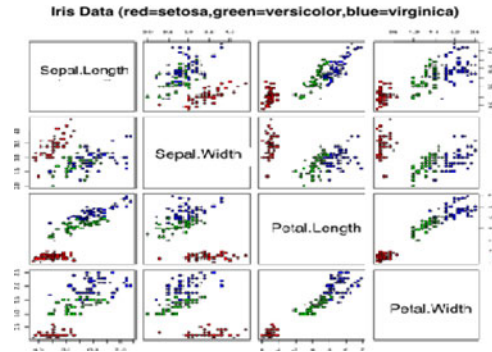


Fig. 20. Iris data.

training process can be converged successfully again. Thus, the improvement is effective and satisfactory.

D. Example 4: Impact of the Capacity of the Fuzzy Neural Network in Clustering Iris Data

This example will adopt a well-known Iris flower dataset to be clustered in pattern recognition. An Iris flower dataset is a multivariate dataset [27]–[29]. It consists of 150 samples and from three species of Iris, i.e., Iris setosa, Iris virginica, and Iris versicolor as shown in Fig. 19. Each species consists of 50 samples. Four features were measured from each sample: the Sepal length (SL), the Sepal width (SW), Petal length (PL), and the Petal width (PW) in centimeters, as in Fig. 20.

Corresponding to the four features as input and one species class as output, the input variable x_i has four features as follows:

$$\text{Input variable } x_i = \{x_{i\text{Sepal length}}, x_{i\text{Sepal width}}, x_{i\text{Petal length}}, x_{i\text{Petal width}}\} \quad (1 \leq i \leq 150).$$

The output y_i is the species of flower as follows:

$$y_i \in \{\text{Iris setosa}, \text{Iris virginica}, \text{Iris versicolor}\}.$$

In each training pattern, we let **1** stands for Iris setosa, **2** stands for Iris virginica, and **3** stands for Iris versicolor. The membership functions are the Gaussian functions, the output node is a linear summing unit. The threshold error value

TABLE III
INITIAL VALUE OF THE CENTER AND WIDTH FOR 14 MFs

MFs number	width	center	MFs number	width	center
1	0.7644	4.3	8	0.8352	2.967
2	0.7644	6.1	9	0.8352	4.933
3	0.7644	7.9	10	0.8352	6.9
4	0.5096	2.0	11	0.3397	0.1
5	0.5096	3.2	12	0.3397	0.9
6	0.5096	4.4	13	0.3397	1.7
7	0.8352	1.0	14	0.3397	2.5

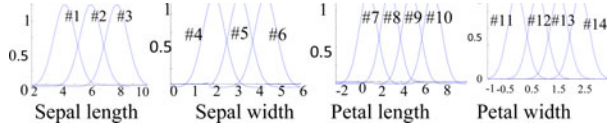


Fig. 21. Fourteen MFs before training.

TABLE IV
FINAL VALUES OF THE CENTER AND WIDTH FOR 14 MFs

MFs number	width	center	MFs number	width	center
1	0.96	4.557	8	0.9187	3.103
2	0.7073	6.235	9	0.7495	4.865
3	0.7937	7.822	10	0.9352	6.695
4	0.2945	1.942	11	0.3779	0.12
5	0.3538	3.097	12	0.09288	1.101
6	0.5859	4.258	13	0.0207	1.789
7	0.913	1.034	14	0.1515	2.386

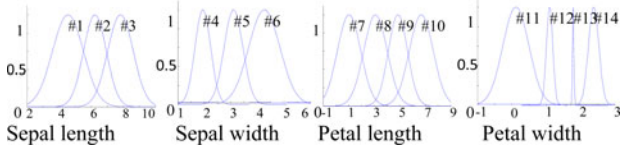


Fig. 22. Fourteen MFs after training.

(= ε) for successful classification is set as $\varepsilon = 0.05$. We apply Algorithm I to conduct gradient descent training algorithm of the FNN, or F-CONFIS. Therefore, we will take all the 150 samples of training pattern. We will use two examples to show the influence of the FNN capacity on the training process.

1) *Example 4.1: Pattern Number Is Less Than the Capacity of the Fuzzy Neural Network—Training Process Converges Successfully:* In this example, let the first two linguistic variables to have three MFs, and the next two linguistic variables to have four MFs, namely, $R_1 = R_2 = 3$, and $R_3 = R_4 = 4$. The initial values of the centers and widths for all the MFs are shown in Table III and the corresponding MFs are shown in Fig. 21. The boundary of the capacity P of the FNN is determined by Theorem 3 ($L \leq P \leq (\sum_{i=1}^n R_i N_{cp} + LM)/M$), that is, $144 \leq P \leq 172$. The pattern number is 150, which is smaller than the capacity upper bound (= 172). After applying the BP training algorithm in Section III (see Algorithm I), the final values of the center and width for all the MFs are shown in Table IV and the corresponding MFs are shown in Fig. 22.

Fig. 23 shows the experimental result when the number of iterations reaches 200. It can be seen that all the training errors

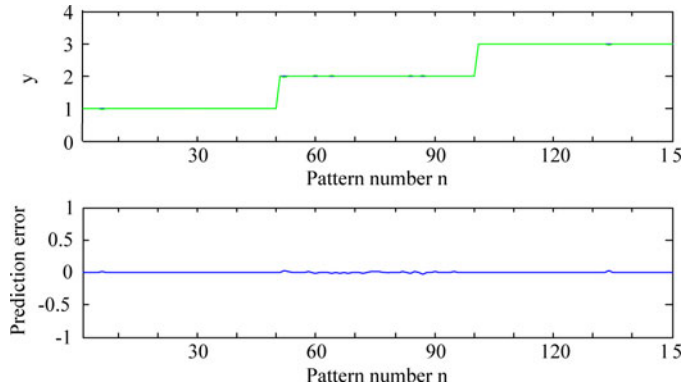


Fig. 23. Pattern number is less than the capacity of FNN in Example 4.

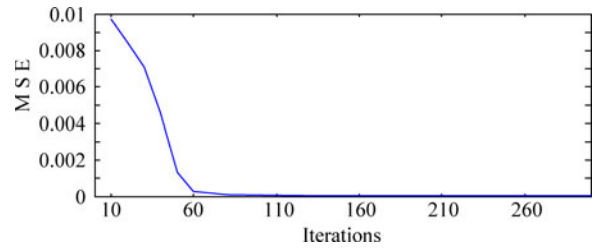


Fig. 24. Relationship between the training error and iterations.

TABLE V
CENTER AND WIDTH FOR EIGHT MFs

The initial values of the center and width for eight MFs			The final values of the centers and widths for eight MFs		
MFs number	width	center	MFs number	width	center
1	1.529	4.3	1	1.63	4.353
2	1.529	7.9	2	1.552	7.912
3	1.019	2.0	3	0.9699	1.989
4	1.019	4.4	4	0.9358	4.443
5	2.505	1.0	5	2.624	1.086
6	2.505	6.9	6	2.674	6.859
7	1.019	0.1	7	0.3251	0.06315
8	1.019	2.5	8	0.2117	2.82

approach zeros in all three classes. Fig. 24 shows the relationship between the training error and iterations. It is shown that when the iteration is larger than 115, the MSE value converges to zero successfully. From this example, it is shown that when the number of training patterns is less than the capacity of the FNN, the training will converge successfully.

2) *Example 4.2: Pattern Number Is Larger than the Capacity of the Fuzzy Neural Network—Overcapacity Training of the Fuzzy Neural Network:* In this case, let $R_1 = R_2 = R_3 = R_4 = 2$. The initial values of the centers and widths for all the MFs are shown in Table V and the corresponding MFs are shown in Fig. 25. The boundary of the capacity P of the FNN is determined by Theorem 3 ($L \leq P \leq (\sum_{i=1}^n R_i N_{cp} + LM)/M$), that is, $16 \leq P \leq 32$. The pattern number is 150, which is larger than the capacity upper bound (=32). The final values of the centers and widths for all the MFs are shown in Table V, and the corresponding MFs are shown in Fig. 26.

Fig. 27 shows the experimental result when iteration reaches 200. It can be seen that some of the training errors cannot

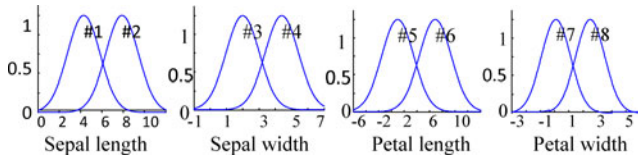


Fig. 25. Eight MFs before training.

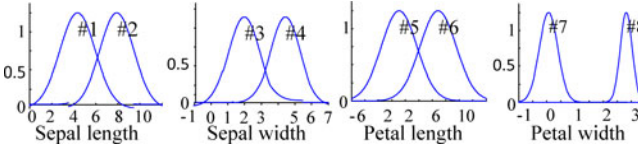


Fig. 26. Eight MFs after training.

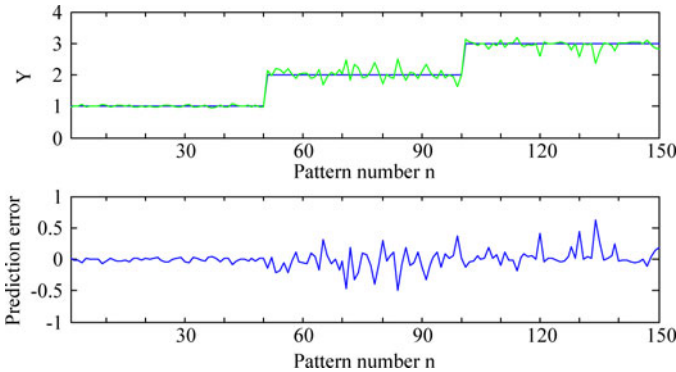


Fig. 27. Pattern number is larger than the capacity of the FNN in Example 4.

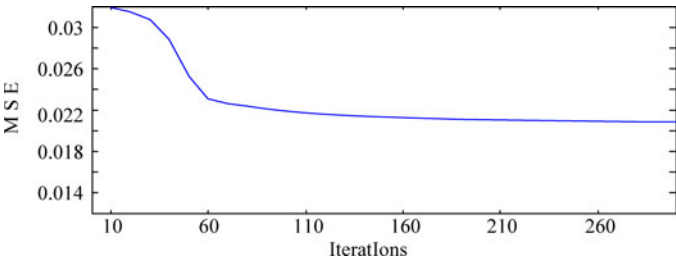


Fig. 28. Relationship between the training error and iterations.

converge to zeros, especially in Class 2 (Iris virginica) and Class 3 (Iris versicolor). The pattern number (150) is larger than the upper bound of the capacity in this example.

The classification accuracy in this case is only 58.000%. Fig. 28 shows the relationship between the training error and iterations. It can be seen that when the iteration is larger than 200, the MSE values will hardly be decreased. This is to say that if the number of training sample is larger than the upper bound of the capacity of the FNN, the training process will not converge to zero and there are the unexpected results in the application, regardless of the increasing of the number iterations. To guarantee the training convergence, it is better to increase the capacity of the FNN, which has been discussed in Section V.

Table VI shows the classification accuracy rates in different cases. The threshold error value (ϵ) for successful classification is set as $\epsilon = 0.05$. Case 1 is the result of Example 4.1, whose pattern number (150) is less than the upper bound of the

TABLE VI
CLASSIFICATION ACCURACIES IN DIFFERENT CASES

Case	R_i	Rule	P	AC	ACAR	Status
1	$R_1=3, R_2=3$ $R_3=4, R_4=4$	144	$144 \leq P \leq 172$	150	100.000%	In
2	$R_1=3, R_2=3$ $R_3=3, R_4=3$	81	$81 \leq P \leq 105$	131	87.333%	Over
3	$R_1=2, R_2=2$ $R_3=3, R_4=3$	36	$36 \leq P \leq 56$	128	85.333%	Over
4	$R_1=2, R_2=2$ $R_3=2, R_4=2$	16	$16 \leq P \leq 32$	87	58.000%	Over
5	$R_1=1, R_2=1$ $R_3=2, R_4=2$	4	$4 \leq P \leq 16$	79	52.667%	Over

P-Capacity AC-Accuracy value In-In capacity
ACAR-Average Classification Accuracy Rate Over-Overcapacity

FNN capacity ($= 172$). Therefore, the classification accuracy is 100%. The training will converge successfully. Cases 2–5 show that pattern numbers are larger than the capacity of the FNN. Case 4 is actually the result from Example 4.2 with its classification accuracy dropped to 58.000%. Therefore, excellent agreements have been achieved between the training simulations and theoretical bounds. All engineering applications using the FNN should not exceed the capacity of the FNN to avoid unexpected results.

VI. CONCLUSION

By transforming the conventional four-layer FNN into an equivalent fully connected three layer feed-forward NN, or F-CONFIS, this paper proposes a new approach to find the lower and upper bounds of the capacity of the FNN. The proposed F-CONFIS is a new architecture of the FNN with dependent and repeated link weights, in which a new gradient descent training algorithm is also developed in this paper. Several examples are illustrated to demonstrate the validity of theoretical bounds for the capacity of the F-CONFIS (or FNN), such as the famous classification of Iris flower dataset. Satisfactory results have been obtained between the theoretical bounds and simulation results. The impacts of this paper are as follows: 1) All engineering applications via FNN (or F-CONFIS) can now adopt the proposed F-CONFIS architecture with its new gradient descent training algorithm to tune its parameters. 2) All engineering applications via the FNN should have the capacity of their imperfect FNN (or F-CONFIS) in mind to complete their successful engineering applications.

REFERENCES

- [1] G. D. Wu, Z. W. Zhu, and P. H. Huang, "A TS-Type maximizing-discriminability-based recurrent fuzzy network for classification problems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 339–352, Apr. 2011.
- [2] C. H. Wang, W. Y. Wang, T. T. Lee, and P. S. Tseng, "Fuzzy B-spline membership function and its applications in fuzzy-neural control," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 841–851, May 1995.
- [3] F. L. Chung, Z. H. Deng, and S. T. Wang, "From minimum enclosing ball to fast fuzzy inference system training on large datasets," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 1, pp. 173–184, Feb. 2009.
- [4] C. F. Juang, T. C. Chen, and W. Y. Cheng, "Speedup of implementing fuzzy neural networks with high-dimensional inputs through parallel processing on graphic processing units," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 4, pp. 717–727, Aug. 2011.

- [5] C. T. Lin, W. C. Cheng, and S. F. Liang, "An on-line ICA-mixture-model-based fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, vol. 52, no. 1, pp. 207–221, Jan. 2005.
- [6] C. L. P. Chen and Y. Lu, "FUZZ: A fuzzy-based concept formation system that integrates human categorization and numerical clustering," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 27, no. 1, pp. 79–94, Feb. 1997.
- [7] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Menlo Park, CA: Addison-Wesley, 1998.
- [8] C. L. P. Chen and Y.-H. Pao, "An integration of neural network and rule-based systems for design and planning of mechanical assemblies," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 5, pp. 1359–1371, Sep./Oct., 1993.
- [9] J. L. Castro, C. J. Mantas, and J. M. Benitez, "Interpretation of artificial neural networks by means of fuzzy rules," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 101–116, Jan. 2002.
- [10] E. Tron and M. Margaliot, "Mathematical modeling of observed natural behavior: A fuzzy logic approach," *Fuzzy Sets Syst.*, vol. 146, pp. 437–450, 2004.
- [11] J. S. R. Jang, "ANFIS: Adaptive-Network-Based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May 1993.
- [12] H. X. Li and C. L. P. Chen, "The equivalence between fuzzy logic systems and feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 356–365, Mar. 2000.
- [13] J. R. Castro, O. Castillo, P. Melin, and A. Rodríguez-Díaz, "A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks," *Inform. Sci.*, vol. 179, no. 13, pp. 2175–2193, 2009.
- [14] V. Kreinovich, H. T. Nguyen, and Y. Yam, "Fuzzy systems are universal approximators for a smooth function and its derivatives," *Int. J. Intell. Syst.*, vol. 15, no. 6, pp. 565–574, Jun. 2000.
- [15] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least squares learning," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 807–814, Sep. 1992.
- [16] R. Hassine, F. Karray, A. M. Alimi, and M. Selmi, "Approximation properties of fuzzy systems for smooth functions and their first-order derivative," *IEEE Trans. Syst., Man, Cybern., A, Syst., Humans*, vol. 33, no. 2, pp. 160–168, Mar., 2003.
- [17] P. L. Narasimha, M. T. Manry, and F. Maldonado, "Upper bound on pattern storage in feedforward networks," in *Proc. Int. Joint Conf. Neural Netw.*, Orlando, FL, USA, Aug. 2007, pp. 1714–1719.
- [18] R. J. McEliece, "The capacity of the hopfield associative memory," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 4, pp. 461–482, Jul., 1987.
- [19] A. Kuh and B. W. Dickson, "Information capacity of associative memories," *IEEE Trans. Inform. Theory*, vol. 35, no. 1, pp. 59–68, Jan. 1989.
- [20] T. M. Cover, "Capacity problems for linear machines," in *Pattern Recognition*, L. Kanal, Ed. Oklahoma City, OK, USA: Thompson, 1968, pp. 283–289.
- [21] S. Tamura, "Capabilities of a three-layer feedforward neural network," in *Proc. Int. Joint Conf. Neural Netw.*, Nov. 1991, pp. 2757–2762.
- [22] S. C. Huang and Y. Huang, "Bounds on the number of hidden neurons in multilayer perceptrons," *IEEE Trans. Neural Netw.*, vol. 2, no. 1, pp. 47–55, Jan. 1991.
- [23] M. A. Sartori and P. J. Antsaklis, "A simple method to derive bounds on the size and to train multilayer neural networks," *IEEE Tran. Neural Netw.*, vol. 2, no. 4, pp. 467–471, Jul. 1991.
- [24] G. B. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation function," *IEEE Trans. Neural Netw.*, vol. 9, no. 1, pp. 224–229, Jan. 1998.
- [25] J. Wang, C. H. Wang, and C. L. P. Chen, "Finding the capacity of Fuzzy Neural Networks (FNNs) via its equivalent fully connected neural networks (FFNNs)," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2011, pp. 2193–2198.
- [26] J. Wang, C. H. Wang, and C. L. P. Chen, "On the BP training algorithm of Fuzzy Neural Networks (FNNs) via its equivalent fully connected neural networks(FFNNs)," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011, pp. 1376–1381.
- [27] C. T. Lin, C. M. Yeh, S. F. Liang, and J. F. Chung, "Support-vector-based fuzzy neural network for pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 31–41, Feb. 2006.
- [28] A. Singh, C. Quek, and S. Y. Cho, "DCT-Yager FNN: A novel yager-based fuzzy neural network with the discrete clustering technique," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 625–644, Apr. 2008.
- [29] W. C. Wong, S. Y. Cho, and C. Quek, "R-POPTVR: A novel reinforcement-based POPTVR fuzzy neural network for pattern classification," *IEEE Trans. Neural Netw.*, vol. 20, no. 11, pp. 31–41, Nov. 2009.
- [30] B. Hammer, "Generalization ability of folding networks," *IEEE Trans. Knowledge Data Eng.*, vol. 13, no. 2, pp. 196–206, Mar./Apr. 2001.
- [31] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585–1588, 1976.
- [32] Z. H. Deng, K. S. Choi, F. L. Chung, and S. T. Wang, "Scalable TSK fuzzy modeling for very large datasets using minimal-enclosing-ball approximation," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 210–226, Apr. 2011.
- [33] C. F. Juang and C. D. Hsieh, "A fuzzy system constructed by rule generation and iterative linear SVR for antecedent and consequent parameter optimization," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 2, pp. 372–384, Apr. 2012.
- [34] C. F. Juang and Y. C. Chang, "Evolutionary group-based particle swarm-optimized fuzzy controller with application to mobile robot navigation in unknown environments," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 379–392, Apr. 2011.
- [35] A. K. Palit and G. Doeding, "Backpropagation based training algorithm for Takagi–Sugeno type MIMO neuro-fuzzy network to forecast electrical load time series," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2002, vol. 1, pp. 86–91.
- [36] S. Tong and H. X. Li, "Fuzzy adaptive sliding-mode control for MIMO nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 3, pp. 354–360, Jun., 2003.
- [37] S. Islam and P. X. Liu, "Robust adaptive fuzzy output feedback control system for robot manipulators," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 2, pp. 288–296, Apr. 2011.
- [38] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 801–806, Sep. 1992.
- [39] C. L. P. Chen, S. R. LeClair, and Y. H. Pao, "An incremental adaptive implementation of functional-link processing for function approximation, time-series prediction, and system identification," *Neurocomput.*, vol. 18, no. 1–3, pp. 11–31, Jan. 1998.
- [40] Y. J. Liu, W. Wang, S. C. Tong, and Y. S. Liu, "Robust adaptive tracking control for nonlinear systems based on bounds of fuzzy approximation parameters," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 170–184, Jan., 2010.
- [41] M. N. H. Siddique and M. O. Tokhi, "Training neural networks: Back-propagation vs. genetic algorithms," in *Proc. Int. Joint Conf. Neural Netw.*, 2001, vol. 4, pp. 2673–2678.
- [42] C. H. Wang, H. L. Liu, and C. T. Lin, "Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 31, no. 3, pp. 467–475, Jun. 2001.
- [43] C. H. Wang and J. S. Wen, "On the equivalence of a table lookup (TL) technique and fuzzy neural network (FNN) with block pulse membership functions (BPMFs) and its application to water injection control of an automobile," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 4, pp. 574–580, Jul. 2008.
- [44] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.
- [45] C. S. Chen, "SRobust self-organizing neural-fuzzy control with uncertainty observer for MIMO nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 4, pp. 694–706, Aug. 2011.
- [46] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
- [47] J. M. Leski, "TSK-fuzzy modeling based on ϵ -insensitive learning," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 2, pp. 181–193, Apr. 2005.
- [48] W. M. Thorburn, "Occam's razor," *Mind*, vol. 24, pp. 287–288, 1915.
- [49] J. Yu and M. S. Yang, "A generalized fuzzy clustering regularization model with optimality tests and model complexity analysis," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 5, pp. 904–915, Oct. 2007.



Jing Wang received the M.S. degree in computer science from Guangdong University of Technology, Guangzhou, China, in 2004. She is currently working toward the Ph.D. degree in software engineering with the Faculty of Science and Technology, University of Macau, Macau, China.

Her current research interests include computational intelligent, systems, and cybernetics.



Chi-Hsu Wang (M'92–SM'93–F'08) was born in Tainan, Taiwan, in 1954. He received the B.S. degree in control engineering from National Chiao Tung University, Hsinchu, Taiwan, the M.S. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin, Madison, WI, USA, in 1976, 1978, and 1986, respectively.

He was an Associate Professor in 1986, and a Professor in 1990, with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. He is currently a Professor with the Department of Electrical Engineering, National Chiao Tung University. His current research interests and publications are in the areas of digital control, fuzzy-neural-networks, intelligent control, adaptive control, and robotics.

Dr. Wang is currently serving as an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS and as a Webmaster of IEEE Systems, Man, and Cybernetics Society.



C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 1985 and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1988, all in electrical engineering.

After having worked at US for 23 years as a Tenured Professor, the Department Head, and an Associate Dean in two different universities, he is currently a Chair Professor with the Department of Computer and Information Science and the Dean with the Faculty of Science and Technology, University of

Macau, Macau, China.

Dr. Chen is a Fellow of American Association for the Advancement of Science and HKIE. He is currently the President of the IEEE Systems, Man, and Cybernetics Society. His research interests include computational intelligence, systems, and cybernetics.