GenSoFNN: A Generic Self-Organizing Fuzzy Neural Network

W. L. Tung and C. Quek, *Member, IEEE*

***Abstract*—Existing *neural fuzzy* (*neuro-fuzzy*) networks proposed in the literature can be broadly classified into two groups. The first group is essentially fuzzy systems with self-tuning capabilities and requires an initial rule base to be specified prior to training. The second group of neural fuzzy networks, on the other hand, is able to automatically formulate the fuzzy rules from the numerical training data. No initial rule base needs to be specified prior to training. A cluster analysis is first performed on the training data and the fuzzy rules are subsequently derived through the proper connections of these computed clusters. However, most existing neural fuzzy systems (whether they belong to the first or second group) encountered one or more of the following major problems. They are 1) *inconsistent* rule-base; 2) heuristically defined node operations; 3) susceptibility to noisy training data and the *stability–plasticity* dilemma; and 4) needs for *prior* knowledge such as the number of clusters to be computed. Hence, a novel neural fuzzy system that is immune to the above-mentioned deficiencies is proposed in this paper. This new neural fuzzy system is named the generic self-organizing fuzzy neural network (GenSoFNN). The GenSoFNN network has strong noise tolerance capability by employing a new clustering technique known as discrete incremental clustering (DIC). The fuzzy rule base of the GenSoFNN network is consistent and compact as GenSoFNN has built-in mechanisms to identify and prune redundant and/or obsolete rules. Extensive simulations were conducted using the proposed GenSoFNN network and its performance is encouraging when benchmarked against other neural and neural fuzzy systems.**

***Index Terms*—Backpropagation (BP), compact and consistent rule-base, compositional rule inference (CRI), generic self-organizing fuzzy neural network (GenSoFNN), laser data, learning vector quantization (LVQ), noise tolerance, one-pass learning, rule pruning, traffic modeling and prediction, 2-spiral.**

I. INTRODUCTION

NEURAL fuzzy networks are the realizations of the functionality of fuzzy systems using neural networks [21]. The main advantage of a neural fuzzy network is its ability to model a problem domain using a linguistic model instead of complex mathematical models. The linguistic model is essentially a fuzzy rule base consisting of a set of IF–THEN fuzzy rules that are highly intuitive and easily comprehended by the human users. In addition, the black-box nature of the neural-network paradigm is resolved, as the connectionist structure of a neural fuzzy network essentially defines the IF–THEN fuzzy rules. Moreover, a neural fuzzy network can self-adjust the parameter of the fuzzy rules using neural-network-based learning algorithms.

Manuscript received May 9, 2001; revised October 26, 2001.

The authors are with the Intelligent Systems Laboratory, Nanyang Technological University, School of Computer Engineering, Singapore 639798, Singapore (e-mail: ashcquek@ntu.edu.sg).

Publisher Item Identifier S 1045-9227(02)05561-3.

Existing neural fuzzy systems proposed in the literature can be broadly classified into two groups. The first group is essentially fuzzy systems with self-tuning capabilities and requires an initial rule base to be specified prior to training [3], [13]. The second group of neural fuzzy networks, on the other hand, is able to automatically formulate the fuzzy rules from the numerical training data [18], [22], [23]. No initial rule base needs to be specified prior to training. The main advantage that the latter group of neural fuzzy systems has over the former is that they are not subjected to the Achilles' heel of traditional fuzzy systems. This is because the first group of neural fuzzy systems may have difficulty in obtaining the initial rule base. That is, it may be difficult to verbalize the knowledge of human experts or formalize them into IF–THEN fuzzy rules if the system is complex. However, most existing neural fuzzy systems (whether they belong to the first or second group) encountered one or more of the following major problems. They are 1) inconsistent rule-base; 2) heuristically defined node operations; 3) susceptibility to noisy training data and the *stability–plasticity* dilemma [17]; and 4) needs for *prior* knowledge such as the number of clusters to be computed.

A consistent rule base [21] is especially important for the knowledge interpretation of a neural fuzzy system. The fuzzy rules extracted from the neural fuzzy network will be meaningless and/or obscure if a fuzzy label can be represented by more than one fuzzy set and these fuzzy sets are allowed to evolve differently during the training phase. In addition, the operations of the neural fuzzy network needs to be clearly defined and mapped to formal fuzzy inference schemes such as the compositional rule of inference (CRI) [35], approximate analogous reasoning schema (AARS) [32], or the truth value restriction (TVR) [19]. If not, the inference steps of the neural fuzzy network become logically heuristic and mathematically unclear.

The choice of clustering techniques in neural fuzzy networks is also an important consideration. The established pseudo outer-product based fuzzy neural network (POPFNN) family of networks [22], [23] has weak resistance to noisy/spurious training data. This is due to the use of *partition-based* clustering techniques [7] such as fuzzy C-means (FCM) [4], linear vector quantization (LVQ) [15] and LVQ-inspired techniques such as modified LVQ, fuzzy Kohonen partitioning (FKP) and pseudo FKP [1] to perform the cluster analysis. Such clustering techniques require *prior* knowledge such as the number of clusters C present in a data set and are not sufficiently flexible to handle nonpartitionable problems such as the XOR dilemma and the 2-spiral problem [16]. Generally, neural fuzzy networks that employ partition-based clustering techniques also lack

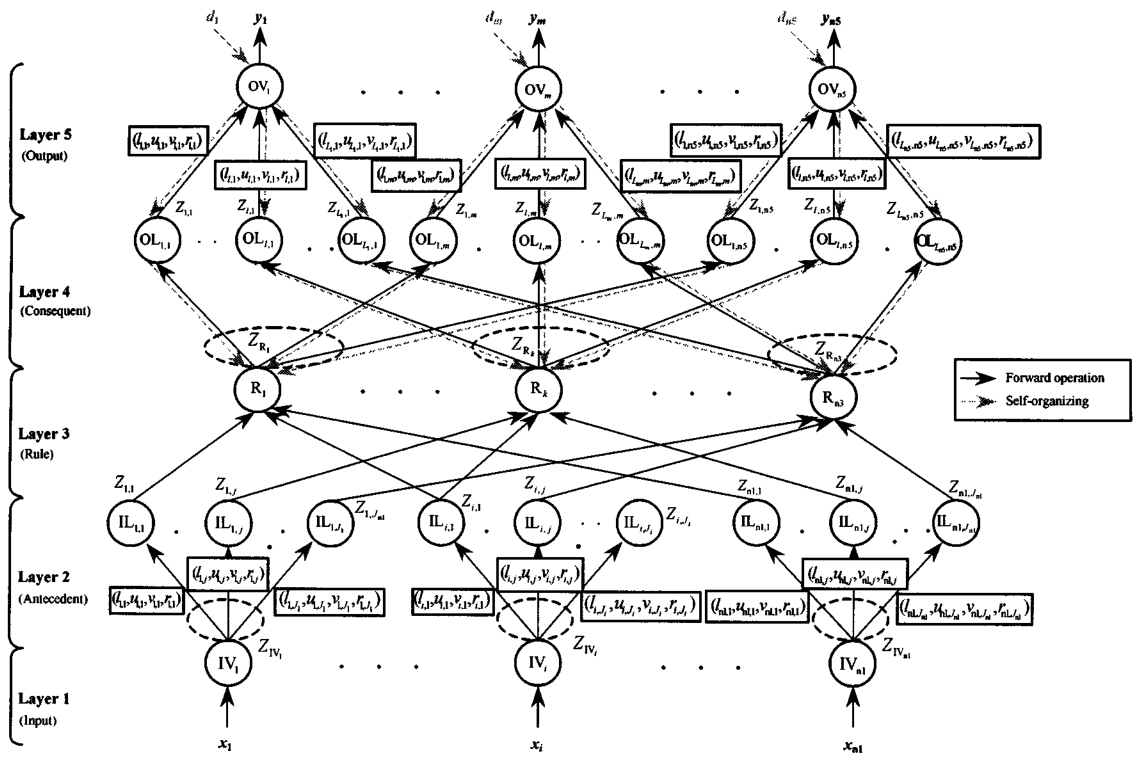

Fig. 1. Structure of GenSoFNN.

the flexibility to incorporate new clusters of data after the training has completed. This is known as the *stability–plasticity* dilemma [17].

Hence, a novel neural fuzzy system that is immune to the above-mentioned deficiencies is proposed in this paper. The new neural fuzzy system is named generic self-organizing fuzzy neural network (GenSoFNN). The GenSoFNN network automatically formulates the fuzzy rules from the numerical training data as compared against the ANFIS [13] and ARIC [3] models and maintains a consistent rule base. Each fuzzy label in the input–output dimensions is uniquely represented by only one cluster (fuzzy set). The GenSoFNN network employs a new clustering technique known as discrete incremental clustering (DIC) to enhance its noise tolerance capability. DIC creates separate clusters for noisy/spurious data that have poor correlation to the genuine or valid data and does not require *prior* knowledge of the number of clusters present in the training data set. In addition, the proposed GenSoFNN network does not require the predefinition of the number of fuzzy rules, as the rule formulation process is entirely data-driven. GenSoFNN is suitable for on-line applications as its training cycle takes place in a single pass of the training data.

This paper is organized as follows. Section II describes the general structure of the GenSoFNN and its on-line training cycle. Section III presents the GenSoFNN-CRI(S) network that is developed by mapping the CRI inference scheme onto the GenSoFNN structure. In Section IV, the GenSoFNN-CRI(S) network is evaluated using three different simulations and its performances are benchmarked against other neural and neural fuzzy systems. Section V concludes this paper.

II. GenSoFNN

The training cycle of the GenSoFNN network (Fig. 1) consists of three phases: *self-organizing, rule formulation*, and *parameter learning*. These are performed sequentially with a single pass of the training data. The DIC clustering technique is developed and incorporated into the GenSoFNN network to automatically compute the input–output clusters from the numerical training data. The fuzzy rules are subsequently formulated by connecting the appropriate input and output clusters during the rule-mapping phase of the training cycle. Consequently, the popular backpropagation (BP) [26] learning algorithm based on negative gradient descent is employed to tune the parameters of the GenSoFNN network.

A. Structure of the GenSoFNN

The GenSoFNN network consists of five layers of nodes. Each input node $\mathrm{IV}_i, i \in \{1, \ldots, n1\}$, has a single input. The vector $X = [x_1, \ldots, x_i, \ldots, x_{n1}]^T$ represents the inputs to the GenSoFNN. Each output node OV_m, where $m \in \{1, \ldots, n5\}$, computes a single output denoted by y_m. The vector $Y = [y_1, \ldots, y_m, \ldots, y_{n5}]^T$ denotes the outputs of the GenSoFNN network with respect to the input stimulus X. In addition the vector $D = [d_1, \ldots, d_m, \ldots, d_{n5}]^T$ represents the desired network outputs required during the parameter learning phase of the training cycle. The trainable weights of the GenSoFNN network are found in layers 2 and 5 (enclosed in rectangular boxes in Fig. 1). Layer 2 links contain the parameters of the input fuzzy sets while layer 5 links contain the parameters of the output fuzzy sets. The weights of the remaining connections are unity. The trainable weights (parameters) are interpreted as the corners of the trapezoidal-shaped fuzzy sets computed by the GenSoFNN network. They are denoted as l and r (left and right support points), and u and v (left and right kernel points). The subscripts denote the presynaptic and postsynaptic nodes, respectively. For clarity in subsequent discussions, the variables i, j, k, l, m are used to refer to arbitrary nodes in layers 1, 2, 3, 4 and 5, respectively. The output of a node is denoted as Z and the subscripts specify its origin.

Each input node IV_i may have different number of input terms J_i. Hence, number of layer 2 nodes is "$n2$," where $n2 = \sum_{i=1}^{n1} J_i$. Layer 3 consists of the rule nodes R_k, where $k = \{1, \ldots, n3\}$. At layer 4, an output term node $\mathrm{OL}_{l,m}$ may have more than one fuzzy rule attached to it. Each output node OV_m in layer 5 can have different number of output terms L_m. Hence, number of layer 4 nodes is "$n4$," where $n4 = \sum_{m=1}^{n5} L_m$. In Fig. 1, the black solid arrows denote the links that are used during the feedforward operation of the GenSoFNN network. The dashed, grey arrows denote the backward links used during the self-organizing phase of the training cycle of the GenSoFNN. The GenSoFNN network adopts the Mamdani's fuzzy model [21] and the kth fuzzy rule R_k has the form as defined in (1)

$$\begin{aligned} R_k\text{: } &\textbf{If } x_1 \text{ is } \mathrm{IL}_{(1,j)_k} \ldots \text{ and } x_i \text{ is } \mathrm{IL}_{(i,j)_k} \ldots \text{ and} \\ &x_{n1} \text{ is } \mathrm{IL}_{(n1,j)_k} \textbf{ then } y_1 \text{ is } \mathrm{OL}_{(l,1)_k} \ldots \text{ and } y_m \\ &\text{is } \mathrm{OL}_{(l,m)_k} \ldots \text{ and } y_{n5} \text{ is } \mathrm{OL}_{(l,n5)_k} \end{aligned} \tag{1}$$

where

$\mathrm{IL}_{(i,j)_k}$ the jth fuzzy label of the ith input that is connected to R_k; and

$\mathrm{OL}_{(l,m)_k}$ the lth fuzzy label of the mth output to which R_k is connected.

Two motivations drive the development of the GenSoFNN network. The first is to define a systematic way of crafting the linguistic model required in neural fuzzy systems and avoids the above-mentioned deficiencies faced by many of the existing neural fuzzy networks. The second motivation is to create a generalized network architecture whereby different fuzzy inference schemes such as CRI can be mapped onto such a network with ease. This closely relates to our definition of what a neural fuzzy network is. That is, a neural fuzzy network is the integration of fuzzy system and neural network, whereby the operations of the hybrid system should be functionally equivalent to a similar standalone fuzzy system. Hence, the operations and outputs of the various nodes in the GenSoFNN network are defined by the fuzzy inference scheme adopted by the network. However, the generic operations of the proposed GenSoFNN can be defined as follows. The forward-based aggregation and activation functions of each layer I are denoted as $f^{(I)}$ and $a^{(I)}$, respectively, where $I \in \{1 \ldots 5\}$. In addition, the label Net defines the aggregated input to an arbitrary node.

Layer 1:

$$\mathrm{Net}_{\mathrm{IV}_i} = f^{(1)}(x_i); \quad Z_{\mathrm{IV}_i} = a^{(1)}\left(\mathrm{Net}_{\mathrm{IV}_i}\right). \tag{2}$$

Layer 2:

$$\mathrm{Net}_{i,j} = f^{(2)}\left(Z_{\mathrm{IV}_i}\right); \quad Z_{i,j} = a^{(2)}(\mathrm{Net}_{i,j}). \tag{3}$$

Layer 3:
The feedforward operation is defined as follows:

$$\begin{aligned} \mathrm{Net}_{R_k} &= f^{(3)}\left(Z_{(1,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}\right) \\ Z_{R_k} &= a^{(3)}\left(\mathrm{Net}_{R_k}\right) \end{aligned} \tag{4}$$

where

$Z_{(i,j)_k}$ = output of fuzzy label $\mathrm{IL}_{i,j}$ connected to rule R_k.

The backward operation (for the self-organizing phase) is defined as follows:

$$\begin{aligned} \mathrm{Net}_{R_k}^{(\mathrm{backward})} &= f_{\mathrm{backward}}^{(3)}\left(Z_{(1,l)_k}, \ldots, Z_{(m,l)_k}, \ldots, Z_{(n5,l)_k}\right) \\ Z_{R_k}^{(\mathrm{backward})} &= a_{\mathrm{backward}}^{(3)}\left(\mathrm{Net}_{R_k}^{(\mathrm{backward})}\right) \end{aligned} \tag{5}$$

where

$Z_{(m,l)_k}$ = backward-based output of fuzzy label $\mathrm{OL}_{l,m}$ that is connected to rule R_k (the subscripts are reversed to denote the backward flow of data).

Layer 4:
The forward operation is defined as follows:

$$\begin{aligned} \mathrm{Net}_{l,m} &= f^{(4)}\left(Z_{R_1}^{(l,m)}, \ldots, Z_{R_k}^{(l,m)}, \ldots, Z_{R_\Omega}^{(l,m)}\right) \\ Z_{l,m} &= a^{(4)}(\mathrm{Net}_{l,m}) \end{aligned} \tag{6}$$

where

$Z_{R_1}^{(l,m)}$ = first rule in GenSoFNN with $\mathrm{OL}_{l,m}$ as part of its consequent;

$Z_{R_k}^{(l,m)}$ = kth rule in GenSoFNN with $\mathrm{OL}_{l,m}$ as part of its consequent; and

$Z_{R_\Omega}^{(l,m)}$ = last rule in GenSoFNN with $\mathrm{OL}_{l,m}$ as part of its consequent.

The backward operation (for the self-organizing phase) is defined as follows. The order of the subscripts has been reversed to reflect the backward operation

$$\begin{aligned} \mathrm{Net}_{m,l} &= f_{\mathrm{backward}}^{(4)}\left(Z_m^{(\mathrm{backward})}\right) \\ Z_{m,l} &= a_{\mathrm{backward}}^{(4)}(\mathrm{Net}_{m,l}) \end{aligned} \tag{7}$$

where

$Z_m^{(\mathrm{backward})}$ = backward output of node OV_m.

Layer 5:
The forward operation is defined as follows:

$$\begin{aligned} \mathrm{Net}_{\mathrm{OV}_m} &= f^{(5)}\left(Z_{1,m}, \ldots, Z_{l,m}, \ldots, Z_{L_m,m}\right) \\ y_m &= a^{(5)}\left(\mathrm{Net}_{\mathrm{OV}_m}\right) \end{aligned} \tag{8}$$

where

$$Z_{l,m} = \text{output of node } \mathrm{OL}_{l,m} \text{ in layer 4.}$$

The backward operation (for the self-organizing phase) is defined as follows:

$$\begin{aligned} \mathrm{Net}_m^{(\text{backward})} &= f_{\text{backward}}^{(5)}(d_m); \\ Z_m^{(\text{backward})} &= a_{\text{backward}}^{(5)}\left(\mathrm{Net}_m^{(\text{backward})}\right) \end{aligned} \tag{9}$$

where

$$d_m = \text{the } m\text{th desired output for the GenSoFNN network.}$$

The detailed node operations are defined by the fuzzy inference system adopted by the GenSoFNN network. For instance, when the CRI [35] inference scheme utilizing the Mamdani's implication rule [20] is mapped onto the GenSoFNN network as in [31], the generic forward operation of rule node R_k as specified by (4) are defined as

$$\begin{aligned} \mathrm{Net}_{R_k} &= f^{(3)}\left(Z_{(1,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}\right) \\ &= \left\{Z_{(1,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}\right\} \end{aligned} \tag{10}$$

$$\begin{aligned} Z_{R_k} &= a^{(3)}\left(\mathrm{Net}_{R_k}\right) \\ &= \min_{i \in \{1 \ldots n1\}} \left\{Z_{(1,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}\right\}. \end{aligned} \tag{11}$$

B. Self-Organization (Clustering) of GenSoFNN

The proposed GenSoFNN network models a problem domain by first performing a cluster analysis of the numerical training data and subsequently deriving the fuzzy rule base from the computed clusters. Generally, clustering techniques may be classified into *hierarchical*-based and *partition*-based techniques. Hierarchical-based clustering techniques included *single link* [9] and *complete link* [2], [14]. The main drawback of hierarchical clustering is that the clustering is static, and points committed to a given cluster in the early stages cannot move to a different cluster. This violates our vision of a dynamic neural fuzzy system where the network can self-organize and self-adapt with changing environments. Prototype-based partition clustering techniques, on the other hand, are dynamic and the data points can move from one cluster to another under varying conditions. However, partition-based clustering techniques require *prior* knowledge such as the number of classes C in the training data. Such information may be unknown and is difficult to estimate in some data set such as traffic flow data [29]. For classification tasks such as the XOR and 2-spiral problems, computing a predefined number of clusters C may not be good enough to satisfactorily solve the problems. Moreover, partition-based clustering techniques suffer from the *stability–plasticity* dilemma [17] where new information cannot be learned without running the risk of eroding old (previously learned) but valid knowledge. Hence, such deficiencies serve as the main motivations behind the development of the discrete incremental clustering (DIC) technique. This new clustering technique is not limited by the need to have *prior* knowledge of the number of clusters and is able to robustly handle noisy/spurious data, as well as preserving the dynamism of partition-based clustering techniques.

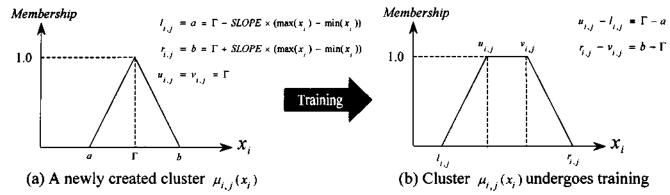

Fig. 2. New cluster (fuzzy set) in DIC with respect to the ith input dimension, $\max(x_i) = \text{maximum input}$ and $\min(x_i) = \text{minimum input}$.

1) DIC: The proposed DIC technique uses raw numerical values of a training data set with no preprocessing. In the current implementation, DIC computes trapezoidal-shaped fuzzy sets and each fuzzy label (fuzzy set) belonging to the same input–output dimension has little or no overlapping of kernel with its immediate neighbors. This is similar to but does not have the same restrictions as a *pseudopartition* [4] of the data space. The DIC technique maintains a consistent representation of the fuzzy sets (fuzzy labels) by performing clustering on a local basis. That is, the number of fuzzy sets for each input–output dimension need not be the same. This is similar to the ART [10] concept. However, unlike ART, if the fuzzy label (fuzzy set) for a particular input–output dimension already exists, then it is not "recreated." Hence, DIC ensures that a fuzzy label is uniquely defined by a fuzzy set and this serves as a basis to formulate a consistent rule base in the GenSoFNN network. The proposed DIC technique has five parameters: a plasticity parameter β, a tendency parameter TD, an input threshold IT, an output threshold OT, and a fuzzy set support parameter SLOPE.

a) Fuzzy Set Support Parameter SLOPE: Each new cluster in DIC begins as a triangular fuzzy set as shown in Fig. 2(a). The kernel of a new cluster (fuzzy set) takes the value of the data point (Γ) that triggers its creation and its support is defined by the parameter SLOPE. As training continues, the cluster "grows" to include more points, but maintains the same amount of buffer regions on both sides of the kernel [Fig. 2(b)]. The same applies for the output clusters.

b) Plasticity Parameter β: A cluster "grows" by expanding its kernel. This expansion is controlled by the plasticity parameter β. A cluster expands its kernel when it is the best-fit cluster (has the highest membership value) to a data point and this point has not yet appear within its kernel. The plasticity parameter β determines the amount a cluster (fuzzy set) expands its kernel to include the new data point. To satisfy the *stability–plasticity* dilemma [17], the initial value of β for all newly formed input–output clusters is preset to 0.5. The value of its β parameter decreases as the cluster expands its kernel. The first quadrant of a cosine waveform (Fig. 3) is used to model the change of β in a cluster.

The parameter θ in Fig. 3 is intuitively interpreted as the maximum expansion a cluster (fuzzy set) can have and a parameter STEP controls the increment of θ from 0 to 1.57 rad. Hence, the amount of expansion a cluster can adopt decreases with the number of expansions.

c) Tendency Parameter TD: The tendency parameter TD is analogous to a cluster's willingness to "grow" when it is the

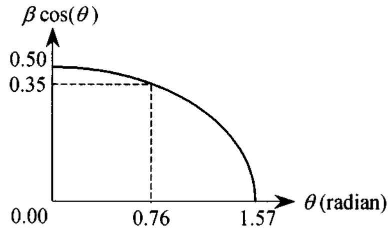

Fig. 3. Modeling of the plasticity parameter β.

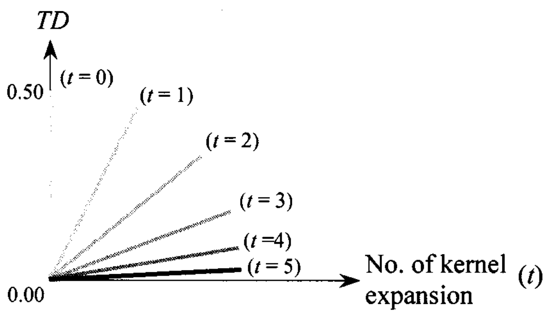

Fig. 4. Dynamics of the tendency parameter TD.

best fit cluster to a data point that falls outside its kernel. The parameter TD complements the use of the plasticity parameter β. This is because β only decreases with the number of times a cluster expand its kernel. The parameter TD maintains the relevance of a cluster and prevents it from incorporating too many data points that has low "fitness" or membership values to the cluster. Otherwise, the kernel of a cluster may become overly large and the semantic meaning of the fuzzy label the cluster represents may become obscure and poorly defined. The initial value of TD of a newly created cluster is preset at 0.5 and the cluster stops expanding its kernel when TD reaches zero. The rate of decrease depends on the "fitness" or membership values of the data points that the cluster incorporates as shown in (12). With respect to node $\mathrm{IL}_{i,j}$ in Fig. 1

$$\mathrm{TD}_{i,j}^{\mathrm{new}} = \mathrm{TD}_{i,j}^{\mathrm{old}} + \left(A - \mathrm{TD}_{i,j}^{\mathrm{old}}\right) \times (1 - \mu_{i,j}(x_i))^2 \quad (12)$$

where $A = -0.5$ and $\mu_{i,j}(\cdot) =$ membership function of the node $\mathrm{IL}_{i,j}$. When TD is less than or equal to zero, the cluster stops "growing" and sets its plasticity parameter β to zero. It must be noted here that A has to be less than zero, otherwise TD can never reach or exceed zero. This is because the value of the term $(1 - \mu_{i,j}(x_i))^2$ is in the range [0, 1) (The case when $\mu_{i,j}(x_i) = 0$ is not valid as the point is then not relevant to the expanding cluster). Here, A is defined as -0.5. The dynamics of the parameter TD is illustrated by Fig. 4.

Hence, the less relevant the data points (with small membership values) a cluster tries to incorporate or absorb; the faster its TD decreases and vice versa. Thus, the tendency parameter TD and the plasticity parameter β works together to maintain the integrity of the input clusters and the fuzzy labels they represent. The same applies for the output clusters.

d) Thresholds (IT and OT): The input (output) threshold, IT (OT), specifies the minimum "fitness" or membership value an input (output) data point must have before it is considered as relevant to any existing input (output) clusters or fuzzy sets. If the membership value of the input (output) data point with respect to the existing best fit input (output) cluster falls below the predefined IT (OT), then a new cluster is created based on that data point. In addition, IT (OT) determines the degree of overlapping of an input (output) cluster with its immediate neighbors (see Fig. 5).

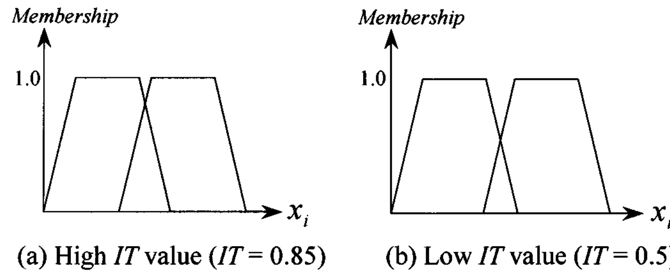

Fig. 5. Effects of IT on clusters for the ith input. (The same applies for OT and the output clusters.)

Hence, the larger the preset value of IT (OT), the closer are the computed input (output) clusters. In order to prevent excessive overlapping of the input (output) clusters (whereby the fuzzy labels become obscure or poorly defined), IT (OT) is predefined at 0.5. The following algorithm performs clustering of the input space. (The same applies to clustering of the output space.) More details on the DIC technique is reported in [30].

Algorithm DIC

Assume data set $\bar{X} = \{X^{(1)}, \ldots, X^{(p)}, \ldots, X^{(P)}\}$, where P is the number of training vectors.
  Vector $X^{(p)} = \{x_1^{(p)}, \ldots, x_i^{(p)}, \ldots, x_{n1}^{(p)}\}$ represents the pth input training vector to the GenSoFNN network.
  Initialize STEP, and $\mathrm{SLOPE} \in (0, 0.5]$.
  $\mathrm{IT} = \beta = \mathrm{TD} = 0.5$.
  For all training vector $p \in \{1 \ldots P\}$ do {
    For all input dimensions $i \in \{1 \ldots n1\}$ do {
      If there are no fuzzy labels (clusters) in the ith input dimension $(J_i = 0)$
        Create a new cluster using $x_i^{(p)}$
      Else do {
        Find the best-fit cluster *Winner* for $x_i^{(p)}$ using (13).

$$\mathrm{Winner} = \arg \max_{j \in \{1 \ldots J_i\}} \left\{ \mu_{i,j}\left(x_i^{(p)}\right) \right\} \quad (13)$$

        Where $\mu_{i,j}(\cdot)$ = membership function of fuzzy label $\mathrm{IL}_{i,j}$.
        If $\mu_{i,\mathrm{Winner}}(x_i^{(p)}) > \mathrm{IT}$ /* Membership value greater than input threshold */
          Update kernel of *Winner*/* grows cluster *Winner**/
        Else
          Create a new cluster using $x_i^{(p)}$
      } End If-Else
    } End For all $i \in \{1 \ldots n1\}$
  } End For all $p \in \{1 \ldots P\}$

End DIC

The parameters used in the DIC technique are constants except for two: the STEP and SLOPE parameters. In the current implementation, the selection of the parameters STEP and SLOPE is heuristic and varies with different tasks. However, there are several guidelines to help in selecting suitable values for these two parameters. A small STEP value results in "fat" fuzzy sets with large kernels and vice versa. On the other hand, a small SLOPE value results in steep slopes (nearly crisp fuzzy sets) and the fuzziness of the fuzzy sets (input and output clusters) increases as the value of SLOPE increases.

C. Rule Formulation of GenSoFNN

The fuzzy rules of the GenSoFNN network are formulated using a *rule mapping* process RuleMAP. Under the GenSoFNN framework, "*input space partition of rule* k" (ISP_k) is the collective term for all the input fuzzy labels (layer 2 nodes) that contribute to the antecedent of rule node R_k (refer to Fig. 1). Similarly, "*output space partition of rule* k" (OSP_k) refers to all the output fuzzy labels (layer 4 nodes) that form the consequent of rule node R_k. During the rule mapping process, each rule $R_k, k \in \{1 \ldots n3\}$, activates its ISP and OSP. For the ISPs, it means a firing of layers 1 and 2 of the GenSoFNN with the input stimulus X feeding into layer 1. To activate the OSPs, layers 4 and 5 are fired with the desired outputs (denoted by the vector D) feeding backward from the output nodes of layer 5. The backward links depicted by the dashed, gray arrows in Fig. 1 are used for the activation of the OSPs. For rule R_k, the aggregated input due to the activation of its ISP_k is denoted as F_{ISP_k} (where $F_{\mathrm{ISP}_k} = Z_{R_k}$ from (4)) and the aggregated input due to the activation of its OSP_k is denoted as F_{OSP_k} [where $F_{\mathrm{OSP}_k} = Z_{R_k}^{(\mathrm{backward})}$ from (5)]. Two user-defined parameters, $\mathrm{Thres}_{\mathrm{ISP}}$ and $\mathrm{Thres}_{\mathrm{OSP}}$, govern the updating of the fuzzy rules in GenSoFNN. When a fuzzy rule is updated in GenSoFNN, the labels (fuzzy sets) in its ISP and OSP "grow" to incorporate the input vector X and the desired output vector D, respectively. An existing rule R_k must satisfy (14) to qualify for update

$$F_{\mathrm{ISP}_k} \geq \mathrm{Thres}_{\mathrm{ISP}} \quad \text{and} \quad F_{\mathrm{OSP}_k} \geq \mathrm{Thres}_{\mathrm{OSP}}. \qquad (14)$$

The flowchart of the rule mapping process RuleMAP with the embedded self-organizing and parameter learning phases is presented as Fig. 6.

The function *EstLink* identifies the proper connections between the input fuzzy labels (layer 2 nodes), the fuzzy rules (layer 3 nodes) and the output fuzzy labels (layer 4 nodes). Overlapping input–output labels are annexed and their respective rules are combined if necessary to maintain a consistent rule base. A new rule R_{new} and a new input space partition $\mathrm{ISP}_{\mathrm{new}}$ are created in tandem to the creation of a new output space partition $\mathrm{OSP}_{\mathrm{new}}$. This is to prevent the crafting of ambiguous rules where an ISP maps to two or more OSPs (that is, the same condition maps to different consequent). This same reason prompts the creation of a new $\mathrm{ISP}_{\mathrm{new}}$ when both $\mathrm{Best}_{\mathrm{ISP}}$ and $\mathrm{Best}_{\mathrm{OSP}}$ are connected to different rules. Details on the rule mapping process RuleMAP is described in [30]. The process RuleMAP is responsible for the structural learning of the GenSoFNN network. The crafted rule base is consistent but not compact, as there may be numerous redundant and/or obsolete rules. Redundant and obsolete rules are the results of the dynamic training of the GenSoFNN where the fuzzy sets of the fuzzy rules are constantly tuned by the backpropagation algorithm. To maintain the integrity and accuracy as well as the compactness of the rule base, these redundant rules have to be deleted. The deletion of obsolete and redundant rules is performed at the end of each training epoch.

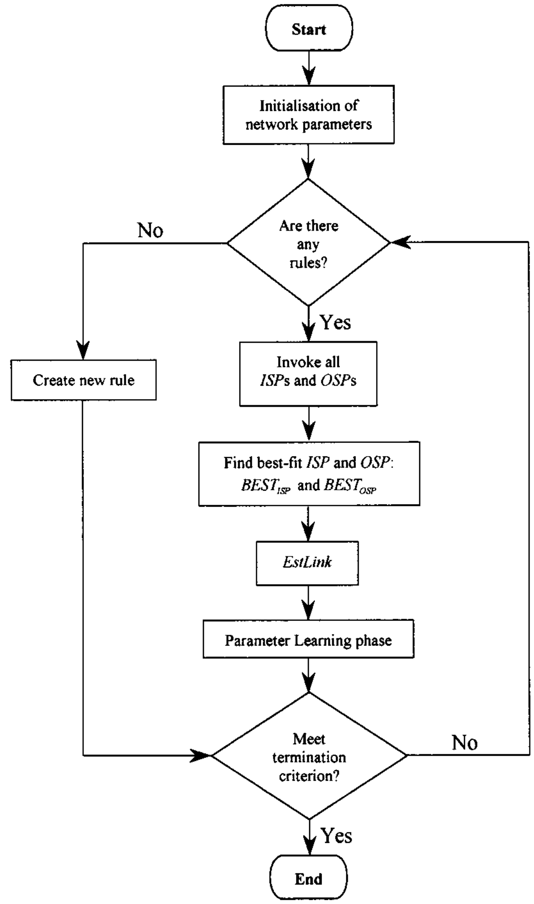

Fig. 6. Flowchart of RuleMAP.

1) Deletion of Redundant and/or Obsolete Rules: Each rule node R_k is time-stamped with a training epoch number T during its creation. The training epoch number T is initialized at zero prior to the start of the training cycle and increases with the iteration of the training data set. Whenever a rule R_k is updated, its time-stamp reflects the current training epoch number. Rules with time-stamp that is more than a training epoch old are considered as obsolete/redundant rules and are deleted at the end of the current training epoch.

D. Parameter Learning of GenSoFNN

The backpropagation learning equations for the parameter learning phase depends on the fuzzy inference scheme adopted by the GenSoFNN network. In Section III, the GenSoFNN-CRI(S) network is presented. GenSoFNN-CRI(S) is created by mapping the CRI [35] inference scheme together with the Mamdani implication rule [20] onto the generic structure of the GenSoFNN network. Singleton fuzzifiers are implemented in layer 1 of the GenSoFNN-CRI(S) network in order to logically map the operations of the network to the CRI inference scheme. Hence, the "(S)" in the network's name refers to the singleton fuzzifiers.

III. GenSoFNN-CRI(S)

Section II described the basic structure and training cycle of the GenSoFNN network. However, the operation and the output of the various nodes in the GenSoFNN network have yet to be defined. This is resolved by mapping an inference scheme onto the basic GenSoFNN architecture. Subsequently, the equations describing the learning operations (of the backpropagation algorithm) for the parameter learning phase in the training cycle of the GenSoFNN network can be derived. These equations are used to tune the fuzzy sets of the term nodes in layers 2 and 4. The GenSoFNN-CRI(S) network results from mapping the CRI [35] reasoning scheme (with Mamdani's implication rule [20]) onto the basic GenSoFNN architecture. The GenSoFNN-CRI(S) network has the same structure as the basic GenSoFNN network (Fig. 1). The CRI inference scheme provides a strong fuzzy logic theoretical foundation for the operations of the GenSoFNN-CRI(S) network. This ensures the operations of the GenSoFNN-CRI(S) network imitate that of the human cognitive process. Please refer to [31] for details on how the mappings are performed.

IV. Simulation Results and Analysis

The GenSoFNN-CRI(S) network is evaluated using three different simulations: 1) 2-Spiral classification, 2) traffic prediction; and 3) time series prediction of a set of laser data. The background of the data sets and the objectives of the simulations are given in the respective sections. For all the simulations, the parameters specified in Table I are used.

A. 2-Spiral Classification

The 2-spiral classification problem is a complex neural-network benchmark task developed by Lang [16]. The task involves learning to correctly classify the points of two intertwined spirals (denoted here as Class 0 and Class 1 spirals, respectively). The two spirals each make three complete turns in a two-dimensional (2-D) plane, with 32 points per turn plus an endpoint, totaling 97 points per spiral (Fig. 7).

Lang *et al.* [16] reported that this problem cannot be solved using a conventional feedforward neural network based on the BP learning algorithm. Instead, they proposed a special network with a 2-5-5-5-1 structure that has 138 trainable weights. In [5], the fuzzy ARTMAP system is trained using the standard 2-spiral data set consisting of 194 points [16]. Evaluation of the fuzzy ARTMAP is performed using the training set as well as a test set that consists of two dense spirals, each with 385 points. For the evaluation of the proposed GenSoFNN-CRI(S) network, the training set is the standard 2-spiral data set consisting of 194 points. The test set consists of two dense spirals with 385 points each (as in [5]) and is generated using (15) to (18). For $n = \{1, \ldots, 385\}$

$$x_1^{(2n)} = 1 - x_1^{(2n-1)} \quad \text{and} \quad x_1^{(2n-1)} = r_n \sin\alpha_n + 0.5 \tag{15}$$

$$x_2^{(2n)} = 1 - x_2^{(2n-1)} \quad \text{and} \quad x_2^{(2n-1)} = r_n \cos\alpha_n + 0.5 \tag{16}$$

TABLE I
Predefined GenSoFNN-CRI(S) Network Parameters

GenSoFNN-CRI(S) network parameters	
Plasticity parameter (DIC), $\beta = 0.5$	Learning constant (back-propagation), $\eta = 0.005$
Tendency parameter (DIC), $TD = 0.5$	Target error $\varepsilon_{\max} = 0.00005$
Input threshold (Self-organisation phase), $IT = 0.5$	$Thres_{ISP} = Thres_{OSP} = 0.6$
Output threshold (Self-organisation phase), $OT = 0.5$	Maximum number of epoch to train, $Epoch_{\max} = 50$

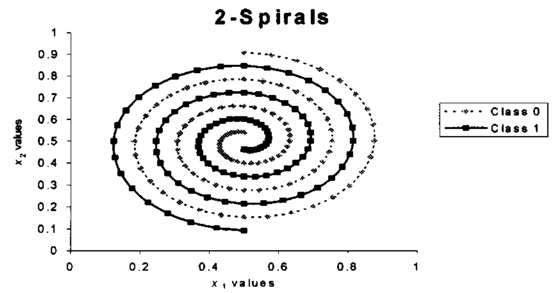

Fig. 7. 2-spiral problem.

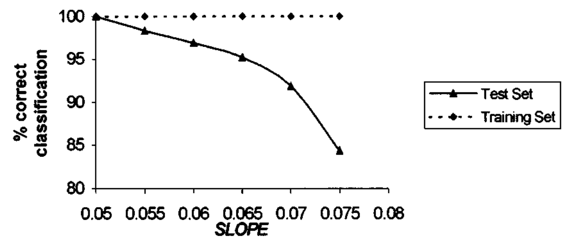

Fig. 8. 2-spiral results of GenSoFNN-CRI(S) versus SLOPE with $\mathrm{STEP} = 0.01$.

$$r_n = 0.4\left(\frac{417 - n}{416}\right) \quad \text{and} \quad \alpha_n = \frac{\pi(n-1)}{64} \tag{17}$$

$$2n - 1 = \text{Class 0} \quad \text{and} \quad 2n = \text{Class 1}. \tag{18}$$

There are two inputs and a single output. During a training epoch, the outermost Class 0 point is presented first followed by the outermost Class 1 point and the sequence continues, alternating between the two spirals and moving toward the center of each spiral. Fig. 8 is drawn to show the effect of the parameter SLOPE on the classification rate of GenSoFNN-CRI(S) for the 2-spiral task. Both the training and test sets are used in the evaluation.

It is seen that the classification rate of the training set is not affected by the change in the parameter SLOPE that varies from 0.05 to 0.075 and maintains at 100%. That is, all the 194 points are correctly classified. However, the classification rate of the test set decreases rapidly from 100% at $\mathrm{SLOPE} = 0.05$ to 84.3% at $\mathrm{SLOPE} = 0.075$. It is probably due to the increased fuzziness of the clusters (fuzzy sets) that result from a larger SLOPE. As fuzziness of the clusters increases, more uncertainty and ambiguity arises between the fuzzy sets (due to gentler slopes). Hence, the test set, which contains a higher density of points packed into two spirals, gives a poorer classification rate with increasing SLOPE as points appearing between the fuzzy (uncertain) regions of clusters are subjected to high probability of wrong classification. Table II shows the best classification results for the 2-spiral task using GenSoFNN-CRI(S), Fuzzy ARTMAP [5] and Lang's proposed neural structure [16].

TABLE II
BEST CLASSIFICATION RESULTS IN 2-SPIRAL SIMULATIONS

Architecture	Test set (770 points)	Training set (194 points)
Lang's 2-5-5-5-1 structure	92.8%	100%
Fuzzy ARTMAP	100%	100%
GenSoFNN-CRI(S)	**100%**	**100%**

Lang *et al.* considered the task as completed when each of the 194 points in the two spirals used for training produces an output within 0.4 of its target output value. On the other hand, Carpenter *et al.* used the most stringent criteria to train the fuzzy ARTMAP system using the standard 2-spiral data set in order to obtain 100% classification for the dense spirals [5]. As a result, the fuzzy ARTMAP system creates 194 ART categories for the standard 2-spiral data set that contains 194 points. In comparison, GenSoFNN-CRI(S) achieves 100% classification for both the standard 2-spiral data set as well as the dense spirals with only 23 fuzzy sets in each of the two input dimensions. This occurs when SLOPE is 0.05 and a total of 156 rules are created (as compared to fuzzy ARTMAP's 194 categories). Moreover, the output responses of the GenSoFNN-CRI(S) network to the 194 points in the standard 2-spiral data set are all within 0.01 of the desired value, as compared to the value of 0.4 specified by Lang.

B. *Traffic Prediction*

This simulation is conducted to evaluate the effectiveness of the GenSoFNN-CRI(S) network in universal approximation and data modeling using a set of traffic flow data. The raw traffic flow data for the simulation was obtained from [29]. The data were collected at a site (Site 29) located at exit 15 along the east-bound Pan Island Expressway (PIE) in Singapore (see Appendix A) using loop detectors embedded beneath the road surface. There are a total of five lanes at the site, two exit lanes and three straight lanes for the main traffic. For this experiment, only the traffic flow data for the three straight lanes were considered. The traffic data set has four input attributes. The four attributes are time and the traffic density of the three lanes. The purpose of this simulation is to model the traffic flow trend at the site using the GenSoFNN-CRI(S) network. The trained GenSoFNN-CRI(S) network is then used to obtain prediction for the traffic density of a particular lane at a time $t+\tau$, where $\tau = 5, 15, 30, 45$ and 60 min. Fig. 9 shows a plot of the traffic flow density data for the three straight lanes spanning a period of six days from 5th to 10th September 1996.

For the simulation, three cross-validation groups of training and test sets are used. They are CV1, CV2, and CV3. The training windows are labeled as such in Fig. 9. The square of the *Pearson product-moment correlation value* (denoted as R^2) [8] is used to compute the accuracy of the predicted traffic trends obtained using the GenSoFNN-CRI(S) network. The predictions made by the GenSoFNN-CRI(S) network for $\tau = 5$ to 60 minutes for lane 1 traffic density (using CV1 as training set) are shown in Fig. 10. The squared errors of each prediction are also included for analysis.

When prediction is made at 5-min intervals, the predicted trend follows very closely to the actual traffic density trend,

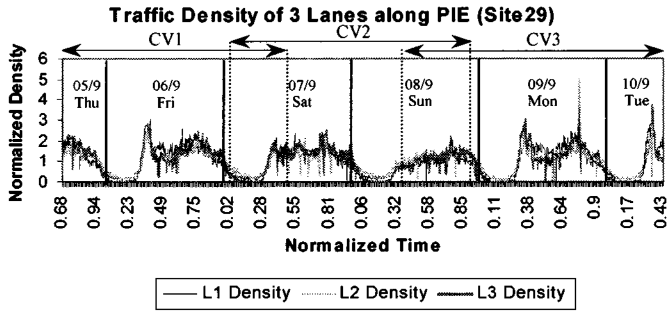

Fig. 9. Traffic density of three straight lanes along PIE.

hence a high accuracy index (R^2) of 0.740 678 [Fig. 10(a)]. The GenSoFNN-CRI(S) is able to predict the peaks and the troughs of the traffic density of lane 1 accurately. However, as the time interval (τ) increases, more errors are observed in the subsequent predicted trends. This is illustrated by Fig. 10(b)–(e). The increased errors are particularly more obvious at the peaks and troughs of the predictions for $\tau = 30, 45,$ and 60 min. The increased errors for longer time intervals are expected as more uncertainties set in for larger values of τ. The squared errors of all the predictions also show that more errors are expected at the peaks than the troughs. This is probably due to the sharp transitions and oscillations in the traffic density characterizing the peaks. The mean squared errors (MSEs) for the different predictions in Fig. 10(a)–(e) at $\tau = 5$ to 60 min are shown by Fig. 10(f). Hence, Fig. 10(f) shows that the accuracy of the predictions decreases as the time interval τ increases.

The same set of experiment is repeated using the multilayer perceptron (MLP) [17] network with four input nodes, ten hidden nodes and one output node. The structure of the MLP is decided after several experiments. The bipolar sigmoidal function with an output range of $[-1, 1]$ is used as the activation function for the hidden and output nodes. The traffic density data set has to be normalized to a range of [0, 1] to fit the data points into the output range of the bipolar sigmoidal function. The input nodes simply relay the input signals to the hidden nodes. The MLP network is trained with the backpropagation algorithm. During the prediction phase, the network functions in a feedforward mode. The average accuracy of the predictions (denoted as $\mathrm{Avg}\, R^2$) by GenSoFNN-CRI(S) and the MLP across the three cross-validation groups (CV1, CV2 and CV3) as τ increases from 5 to 60 min for all the three lanes are shown in a plot as Fig. 11.

Comparing against the results of the GenSoFNN-CRI(S), the MLP achieves better predictions initially (when $\tau = 5$ and 15 mins). However, for predictions when $\tau = 30$ to 60 mins, the MLP network experienced a drastic drop in its accuracy. This drop in the accuracy of the predictions is obvious among all the three lanes. In addition, the trained MLP is a black box and the linguistic rules defining the traffic flow pattern cannot be extracted from it. Hence, the GenSoFNN-CRI(S) network may not be as accurate as the MLP in the initial predictions, but it offers a better overall performance from $\tau = 5$ to 60 mins. In addition, an intuitive set of fuzzy rules can be extracted from its trained structure to describe the dynamics of the traffic conditions. This contrast in prediction accuracy for different time intervals also

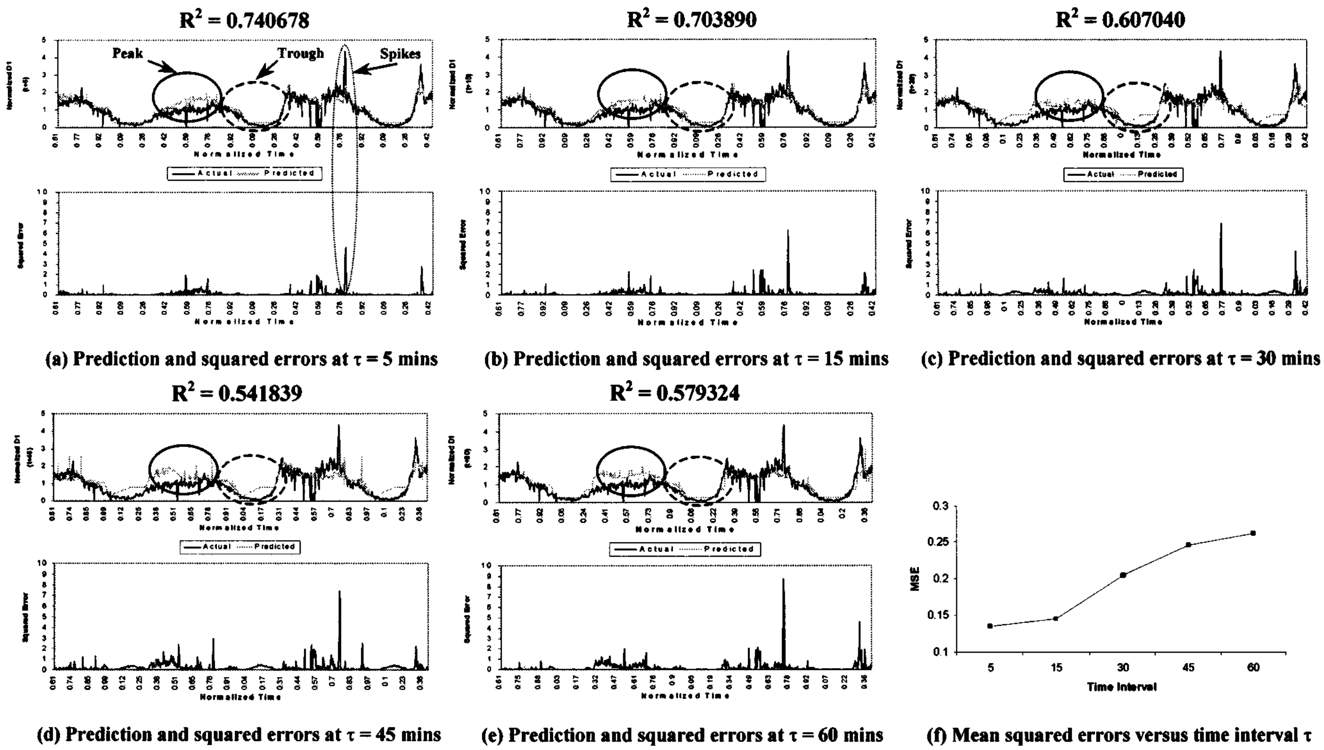

Fig. 10. Prediction of lane 1 density using GenSoFNN-CRI(S) (Training set is CV1).

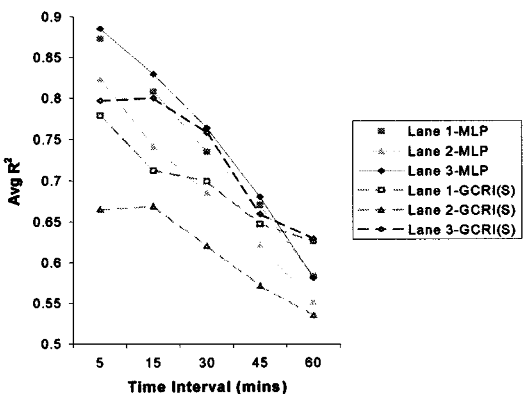

Fig. 11. Avg R^2 versus time interval τ for GenSoFNN-CRI(S) and MLP (4-10-1).

TABLE III
BENCHMARKING OF SIMULATION RESULTS OF TRAFFIC PREDICTION

Network	Lane 1 Var (%)	Lane 2 Var(%)	Lane 3 Var (%)	Avg Var (%)
Falcon-FCM(CL)	24.17	9.32	30.47	21.32
Falcon-MLVQ(CL)	36.41	25.94	30.21	30.85
Falcon-FKP(CL)	23.87	22.09	35.19	27.05
Falcon-PFKP(CL)	27.81	21.05	28.25	25.70
Falcon-MART	20.78	15.47	20.58	18.94
GAMFFRC	24.76	22.48	24.52	23.92
MLP (4-10-1)	33.24	33.10	34.38	33.57
GenSoFNN-CRI(S)	**19.64**	**19.58**	**21.09**	**20.10**

reveals that the proposed DIC technique has a better noise handling capability than the MLP network. The results of the GenSoFNN-CRI(S) network are subsequently benchmarked against that of other neural and neural fuzzy systems in Table III.

Two indicators are used for the benchmarking of the various systems. The first indictor is "Var" (the change in $\mathrm{Avg}\, R^2$ value from $\tau = 5$ min to $\tau = 60$ min expressed as a percentage of the former) and the second indictor is "Avg Var," the mean "Var" values across all three lanes. These two indicators reflect the consistency of the predictions made by the benchmarked systems over the time interval when τ changes from 5 to 60 min across the three lanes. Table III shows that the performance of the GenSoFNN-CRI(S) network compares favorably against the Falcon-class of networks [25] and the GAMFFRC system [6], a GA (Genetic algorithm) [17], [28] based system that is able to automatically construct fuzzy membership functions and fuzzy rules from numerical data. However, an analysis of the detailed results in [6] and [25] reveal that the accuracy of the predictions is poor as compared to that of the GenSoFNN-CRI(S) network. Moreover, the results of the GAMFFRC system cannot be easily reproduced, as GA is a search paradigm that derives its strength from randomness. In addition, the numbers of fuzzy rules and fuzzy labels have to be predefined in the GAMFFRC system and the fuzzy rules derived by the GAMFFRC system may not be consistent, as there is no control over the evolution of the fuzzy labels. Table III shows that GenSoFNN-CRI(S) has superior performance to all the networks except Falcon-MART [24]. However, Falcon-MART uses more than 200 rules for the simulation as compared to the 120–130 fuzzy rules derived by the GenSoFNN-CRI(S) network.

C. Time Series Prediction Using Laser Data

The third simulation uses a set of laser data [11] from the Santa Fe Institute (hereby denoted as SFI) time series prediction and analysis competition [34]. The laser data is publicly available at SFI [27]. The original laser data set from the SFI competition consists of 1000 observations of the fluctuations in

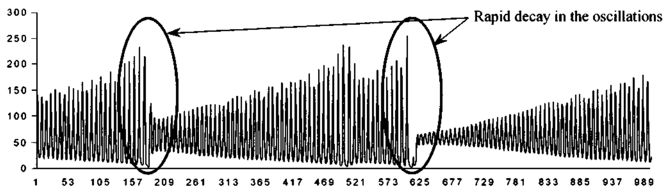

Fig. 12. Laser data used in the SFI competition.

a far-infrared (FIR) laser. The task is to use these 1000 points and try to predict the continuation of the series for the next 100 points (i.e., observations at 1001st–1100th time instants). The plot of the first 1000 observations in the laser data set is shown as Fig. 12.

The laser data time series has relatively simple oscillations (though gradually increasing) but has global events that are hard to predict (the rapid decay of the oscillations). In order to train the GenSoFNN-CRI(S) network for the prediction task, a training set is generated using the 1000 points in the laser data set. This training set consists of five inputs and one output. The data points in the training set are governed by

$$x(t) = F(x(t-1), x(t-2), x(t-3), x(t-4), x(t-5)) \tag{19}$$

where $x(t)$ is the next output to be predicted, $\{x(t-1), \ldots, x(t-5)\}$ is the set of the five past observations, and F is the nonlinear function relating the next observation to the five past observations which the GenSoFNN-CRI(S) tries to model. Hence, $x(t)$ acts as the output in the training set while $\{x(t-1), \ldots, x(t-5)\}$ forms the inputs, and $t \in \{6 \ldots 1000\}$.

The predicted continuation of the time series by GenSoFNN-CRI(S) for the laser data for the next 100 time instants is given in Fig. 13. The true continuation to the laser series is shown as the darker of the two plots.

The predicted continuation of the laser series by the GenSoFNN-CRI(S) network is accurate except at the point of rapid decay of the oscillations. However, the general trend of the decay (as seen by the similar transitions) is still captured by GenSoFNN-CRI(S) and the continuation of the prediction thereafter is fairly accurate as shown by Fig. 13. A measure of prediction accuracy is given by the normalized mean-squared error (NMSE) [34]

$$\mathrm{NMSE} = \frac{1}{\sigma^2 N} \sum_{i=1}^{N} (x_i - \widehat{x}_i)^2 \tag{20}$$

where

- x_i true value of the ith point of the series of length N;
- $\widehat{x}_i$ predicted value;
- σ^2 variance of the true time series during the prediction interval N.

A value of $\mathrm{NMSE} = 1$ corresponds to simply predicting the average of the time series. The NMSE value is used as an indicator to benchmark against the results reported in [34].

As can be seen from Table IV, the prediction by GenSoFNN-CRI(S) is more accurate than all the feedforward networks of various structures but inferior to the systems

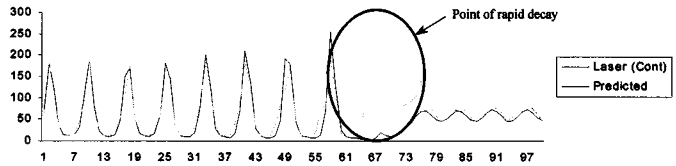

Fig. 13. Continuation of the laser series of length 100 by GenSoFNN-CRI(S).

TABLE IV
BENCHMARKING OF SIMULATION RESULTS OF LASER SERIES

System	NMSE
Delay Coordinate Embedding (DCE)	0.080
Network with Internal Delay Lines (IDL)	0.028
Feedforward (200-100-1)	0.770
Feedforward (50-20-1)	1.000
Feedforward (50-350-50-50)	0.380
GenSoFNN-CRI(S)	**0.244**

proposed by Sauer (DCE) and Wan (IDL) [34]. A major factor contributing to the large NMSE value of 0.244 as compared against DCEs 0.080 and IDLs 0.028 is the failure by the GenSoFNN-CRI(S) to closely approximate the characteristic of the laser series at the point of rapid decay in the oscillations. This is probably because the DCE and IDL structures are better equipped to capture distinct temporal information, which is very significant in prediction task with sudden, varying global events as in the case of the rapid decay in oscillations for the laser data time series.

V. CONCLUSION

In this paper, a novel neural fuzzy architecture named GenSoFNN is proposed. The CRI fuzzy inference scheme [35] using Mamdani's implication [20] is subsequently incorporated into the general structure of the proposed GenSoFNN network to create the GenSoFNN-CRI(S) network. In this way, the operations and outputs of the various nodes in the GenSoFNN-CRI(S) network are clearly defined and have a strong fuzzy logic foundation. This ensures that the functionality of the GenSoFNN-CRI(S) network is similar to that of the human cognitive process.

The key strength of the GenSoFNN-CRI(S) network is that an intuitive and consistent fuzzy rule base describing the dynamics (behavior) of the problem domain can be extracted from its trained structure. The GenSoFNN-CRI(S) network maintains a consistent fuzzy rule base by ensuring that each fuzzy label in the rule base is represented by only one cluster (fuzzy set). The fuzzy rule base formulated by the GenSoFNN-CRI(S) network is consistent as well as compact. That is because redundant or obsolete fuzzy rules are pruned off at the end of each training epoch. This maintains the integrity of the derived fuzzy rule base and ensures that the dynamics (behavior) of the problem domain is properly modeled. The GenSoFNN-CRI(S) network employs a new clustering technique called DIC to compute the trapezoidal-shaped fuzzy sets during its self-organization phase. DIC is superior to partition-based clustering techniques such as LVQ [15] and FCM [4]. It does not require *prior* knowledge of

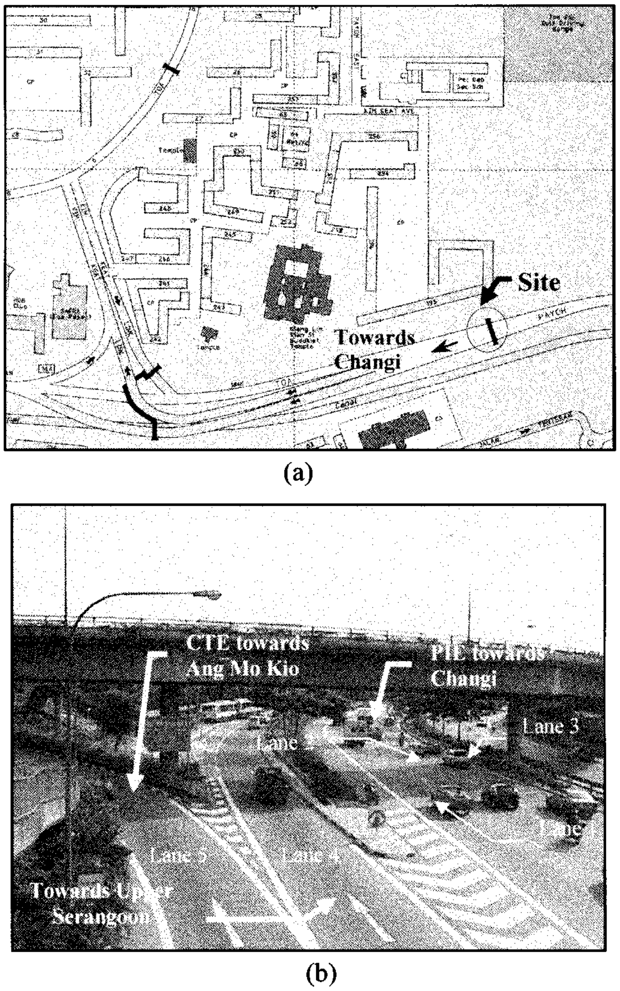

Fig. 14. (a) Location of Site 29 along PIE (Singapore). (b) Actual site at exit 15.

the number of classes in the data set and has good noise-tolerance capability.

In addition, the GenSoFNN-CRI(S) network does not require an initial rule base to be specified prior to training, which is a prerequisite for systems like ANFIS [13] and ARIC [3]. This removes the tedious task of having to translate subconscious knowledge of a problem domain into IF–THEN fuzzy rules. The GenSoFNN-CRI(S) network has great flexibility to include new training data and new fuzzy rules even after training terminates. This is a great contrast to the POPFNN [22], [23] class of networks where the rule base is fixed after training. In the latter, incorporating new fuzzy rules or new clusters of data usually mean a retraining for the entire system.

The performance of the GenSoFNN-CRI(S) network is evaluated using three simulations: 1) 2-Spiral classification; 2) traffic prediction; and 3) time series prediction using a set of laser data. The results of the GenSoFNN-CRI(S) network have been encouraging when benchmarked against other neural fuzzy systems and traditional systems such as the MLP network.

APPENDIX A

The site location (Site 29) at which traffic flow data for the second experiment is collected is as shown in Fig 14. The arrows show the direction of traffic flow.

REFERENCES

[1] K. K. Ang, H. C. Quek, and M. Pasquiera, "POPFNN-CRI(S)Pseudo Outer Product-Based Fuzzy Neural Network Using the Compositional Rule of Inference and Singleton Fuzzifier," IEEE Trans. Syst., Man, Cybern. B, 2002, to be published.

[2] H. Benkirane, M. A. Kbir, K. Maalmi, and R. Benslimane, "Hierarchical fuzzy partition for pattern classification with fuzzy if-then rules," *Pattern Recognition Lett.*, vol. 21, pp. 503–509, 2000.

[3] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks*, vol. 3, pp. 724–740, Sept. 1992.

[4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[5] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 698–713, Sept. 1992.

[6] W. J. Chen, "GA-Optimized Fuzzy Systems," M.Eng. Thesis, School Comput. Eng., Nanyang Technol. Univ., Singapore, 2000.

[7] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 450–465, May 1999.

[8] R. N. Goldman and J. S. Weinberg, *Statistics: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[9] J. C. Gower and G. J. S. Ross, "Minimum spanning trees and single linkage cluster analysis," *Appl. Statist.*, vol. 18, pp. 54–64, 1969.

[10] S. Grossberg, "Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions," *Biol. Cybern.*, vol. 23, pp. 187–202, 1976.

[11] U. Hubner, N. B. Abraham, and C. O. Weiss, "Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared Nh3 laser," *Phys. Rev. A*, vol. 40, p. 6354, 1989.

[12] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 260–270, Aug. 1995.

[13] J. S. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665–685, Sept. 1993.

[14] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 323, pp. 241–254, 1967.

[15] T. K. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.

[16] K. J. Lang and M. J. Witbrock, "Learning to tell two spirals apart," in *Proc. 1988 Connectionist Models Summer School*, 1988, pp. 52–59.

[17] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[18] C. J. Lin and C. T. Lin, "An ART-based fuzzy adaptive learning control network," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 477–496, Aug. 1997.

[19] R. L. Mantaras, *Approximate Reasoning Models*. Chichester, West Essex, U.K.: Ellis Horwood Limited, 1990.

[20] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Trans. Comput.*, vol. C-26, pp. 1182–1191, Nov. 1977.

[21] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. New York, : Wiley, 1997.

[22] H. C. Quek and R. W. Zhou, "POPFNN: A pseudo outer-product based fuzzy neural network," *Neural Networks*, vol. 9, no. 9, pp. 1569–1581, 1996.

[23] ——, "POPFNN-AARS(S): A pseudo outer-product based fuzzy neural network," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 859–870, Nov. 1999.

[24] H. C. Quek and W. L. Tung, "A novel approach to the derivation of fuzzy membership functions using the falcon-MART architecture," *Pattern Recognition Lett.*, vol. 22, no. 9, pp. 941–958, 2000.

[25] ——, "PACL-FNNS: A novel class of falcon-like fuzzy neural networks based on positive and negative exemplars," in *Intelligent System Techniques and Applications*. Boca Raton, FL: CRC, 2000.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams *et al.*, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, ch. 8.

[27] Repository for the laser data. [Online]. Available: ftp.santafe.edu

[28] S. C. Shin and S. B. Park, "GA-based predictive control for nonlinear processes," *Electron. Lett.*, vol. 34, no. 20, pp. 1980–1981, 1998.

[29] G. K. Tan, "Feasibility of Predicting Congestion States with Neural Networks," Final Year Project Thesis, School Civil Structural Eng., Nanyang Technol. Univ., Singapore, 1997.

[30] W. L. Tung, "A Generalized Platform for Fuzzy Neural Network," School Comput. Eng, Nanyang Technol. Univ., Singapore, Tech. Rep., ISL-TR-01/01, 2001.

[31] W. L. Tung and C. Quek, "Derivation of GenSoFNN-CRI(S) From CRI-Based Fuzzy System," School Comput. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep., ISL-TR-04/01, 2001.

[32] I. B. Turksen and Z. Zhong, "An approximate analogical reasoning scheme based on similarity measures and interval valued fuzzy sets," *Fuzzy Sets Syst.*, vol. 34, pp. 323–346, 1990.

[33] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Systs., Man, Cybern.*, vol. 22, pp. 1414–1427, Nov. 1992.

[34] A. S. Weigend and N. A. Gershenfeld, "Time series prediction: Forecasting the future and understanding the past," presented at the Nat. Advanced Res. Workshop Comparative Time Series Anal., A. S. Weigend and N. A. Gershenfeld, Eds., Santa Fe, New Mexico, 1992.

[35] L. A. Zadeh, "Calculus of fuzzy restrictions," in *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. New York: Academic, 1975, pp. 1–39.

W. L. Tung received the B.A.Sc. degree in computer engineering from Nanyang Technological University, Singapore, Singapore, in 2000. He is currently pursuing the Ph.D. degree at the same university.

His research interests include neural networks, fuzzy systems, fuzzy rule-based systems, and genetic algorithms.

C. Quek (M'83) received the B.Sc. degree in electrical and electronics engineering and the Ph.D. degree in intelligent control from Heriot Watt University, Edinburgh, U.K.

He is an Associate Professor and a Member of the Intelligent Systems Laboratory, School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include intelligent control, intelligent architectures, artificial intelligence in education, neural networks, fuzzy systems, fuzzy rule-based systems, and genetic algorithms.