ELSEVIER

Pattern Recognition Letters 22 (2001) 941–958

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

A novel approach to the derivation of fuzzy membership functions using the Falcon-MART architecture

C. Quek *, W.L. Tung

Intelligent Systems Laboratory, Nanyang Technological University, School of Computer Engineering, Blk N4, #2a-32 Nanyang Avenue, Singapore 639798, Singapore

Received 27 April 2000; received in revised form 7 September 2000

Abstract

A fuzzy neural network, *Falcon-MART*, is proposed in this paper. This is a modification of the original *Falcon-ART* architecture. Both Falcon-ART and Falcon-MART are fuzzy neural networks that can be used as fuzzy controllers or applied to areas such as forgery detection, pattern recognition and data analysis. They constitute a group of hybrid systems that incorporate fuzzy logic into neural networks. In this way, the structure of these hybrid networks become transparent as high level IF-THEN human-like reasoning is used to interpret the network connections. In addition, the hybrid networks automatically derive the fuzzy rules (knowledge base) of the problem domain using neural network techniques and hence avoid the pitfalls of traditional fuzzy systems. The main problem in designing a fuzzy neural network is how to formulate the fuzzy rule base. Most proposed fuzzy neural networks in the literature could be classified into two categories. The first group assumes the existence of a preliminary rule base and uses neural techniques to tune the parameters to obtain the final set of fuzzy rules. The second group assumes no knowledge of any fuzzy rules and performs a cluster analysis on the numerical training data before formulating the rules from the computed clusters. Falcon-ART attempts to overcome the constraints faced by these two groups of fuzzy neural networks by using the fuzzy ART technique to partition the training data set. However, there are several shortcomings in the Falcon-ART network. They are:

1. Poor network performances when the classes of input data are closely similar to each other;
2. Weak resistance to noisy/spurious training data;
3. Termination of network training process depends heavily on a preset error parameter; and
4. Learning efficiency may deteriorate as a result of using complementary coded training data.

Falcon-MART has been developed to address these shortcomings. To evaluate the effectiveness of Falcon-MART, three different sets of experiments are conducted. The first experiment demonstrates the efficiency of Falcon-MART over Falcon-ART using the *Fisher's Iris data* set. The second experiment evaluates the modeling capability of Falcon-MART against the classical multi-layered perceptron (MLP) network using a set of traffic flow data. The last experiment uses a set of phoneme data to demonstrate the clustering ability of Falcon-MART against the traditional *K-nearest-neighbor* (*K*-NN) classifier. The results obtained are encouraging.

Keywords: Falcon-ART; Falcon-MART; Noisy data; Complementary coding; Fuzzy rules; Classification; Fuzzy neural networks; Adaptive resonance theory

* Corresponding author. Tel.: +65-790-4926; fax: +65-792-6559.
E-mail address: ashcquek@ntu.edu.sg (C. Quek).

0167-8655/01/$ - see front matter
PII: S 0 1 6 7 - 8 6 5 5 (0 1) 0 0 0 3 3 - 2

1. Introduction

Traditional fuzzy systems are flexible and intuitive in their modeling of the problem domain. However, the main drawback of traditional fuzzy systems is the process of formulating the fuzzy rule base, which is often manual, subjective and sometimes inconsistent. In addition, experts may find it difficult to explicitly express their knowledge of the problem domain in IF-THEN linguistic rules. Neural networks, on the other hand, are able to efficiently model the systems that they are presented with. However, the trained networks are opaque and become a black box to the users. It is difficult to interpret and relate the weight matrix of a trained neural network to the dynamics of the problem domain. Artificial fuzzy neural networks are hybrid networks that incorporate fuzzy logic into neural network structures. In this way, the connectionist structures of these hybrid networks become transparent as high level IF-THEN human-like reasoning can be applied to interpret the network connections. At the same time, the hybrid networks automatically derive the fuzzy rules and the associated parameters, hence avoiding the pitfalls of traditional fuzzy systems.

However, a new set of problems besieges the design of fuzzy neural networks. The main problem is how to formulate the fuzzy rule base to accurately reflect the dynamics of the problem domain. Most proposed fuzzy neural networks in the literature could be classified into two categories. The first group assumes the existence of a preliminary set of fuzzy rules and their parameters are tuned using neural techniques to obtain the final fuzzy rule base. This group of hybrid networks faces the same constraint as the traditional fuzzy systems, that is, the derivation of a consistent set of preliminary fuzzy rules. The second group assumes no knowledge of any available fuzzy rules and performs a cluster analysis on the numerical training data set before formulating the fuzzy rule base using the computed clusters. However, this group of fuzzy neural networks is prone to the problems encountered by the clustering techniques they used. This includes the need of a prior knowledge such as the number of clusters for a given data set and the need to determine the optimal values for the parameters in the clustering algorithm since they vary with data sets.

The *Falcon-ART* (Lin and Lee, 1991) architecture attempts to overcome the constraints faced by these two groups of fuzzy neural systems. Falcon-ART applies fuzzy adaptive resonance theory (ART) (Carpenter and Grossberg, 1987a,b, 1988, 1990, 1991; Grossberg, 1976) to obtain the trapezoidal fuzzy partitions of the input and output data spaces. There is no need to specify the number of clusters and most of the parameters used are application independent. Moreover, fuzzy ART computes the fuzzy partitions and formulate the associated fuzzy rules in a single pass of the data set, making Falcon-ART suitable for on-line learning. However, there are several shortcomings in the network. They are:

1. Poor network performances when the classes of input data are closely similar to each other;
2. Weak resistance to noisy/spurious training data;
3. Termination of network training process depends heavily on a preset error parameter; and
4. Learning efficiency may deteriorate as a result of using complementary coded training data.

In this paper, a modified version of the Falcon-ART network is proposed to address these deficiencies. This improved network is named *Falcon-Modified ART* (*Falcon-MART*). As compared to Falcon-ART, Falcon-MART is able to differentiate data points from similar classes and is more resilient to noisy/spurious inputs. This is achieved by using weighted averaging in determining the firing strengths of the fuzzy rules. In addition, the training of Falcon-MART terminates when the difference in cost errors between two successive training epoch is small enough. This eliminates the need to determine the optimal value of the targeted cost error as in Falcon-ART. By using absolute-valued training data, Falcon-MART formulates a fuzzy rule base that is more intuitive than the one derived by Falcon-ART, and it is able to handle application with a single input/output. To illustrate the concepts behind Falcon-MART, the paper is organized as follows. In Section 2, the original Falcon-ART architecture is introduced and experimental results using *Fisher's Iris data* set (Fisher, 1936) in Section 3 are presented to highlight the shortcomings. Section 4 introduces the proposed Falcon-MART architecture and the performance of the Falcon-MART network is validated by results from three different experiments in Section 5. Section 6 presents the conclusions and highlight additional work undertaken.

2. Dynamics of Falcon-ART

The Falcon-ART architecture developed by Lin and Lee (1991) is a highly autonomous system. It has five layers as shown in Fig. 1 and generates fuzzy rules of the form in Eq. (1).

$$\begin{aligned}&\textbf{If } \textit{input } 1 \textit{ is } L_{1\phi} \ldots \textit{ and input } p \textit{ is } L_{p\phi} \ldots \\ &\quad \textit{and input } n \textit{ is } L_{n\phi} \\ &\textbf{Then } \textit{output } 1 \textit{ is } L'_{1\theta} \ldots \textit{ and output } t \textit{ is } L'_{t\theta} \ldots \\ &\quad \textit{and output } m \textit{ is } L'_{m\theta}\end{aligned} \tag{1}$$

The fuzzy rule in Eq. (1) has five elements, namely the input linguistic variables and terms, the output linguistic variables and terms and the IF-THEN rule construct. The labels *input* 1 ... *input n* denotes the input linguistic variables and the labels *output* 1 ... *output m* denote the output linguistic variables, where n and m are the number of input and output, respectively. They represent entities such as height, speed and weight. The labels $L_{1\phi}, \ldots, L_{n\phi}$ denote the input linguistic terms and the labels $L'_{1\theta}, \ldots, L'_{m\theta}$ denote the output linguistic terms. In Falcon-ART, the input and output

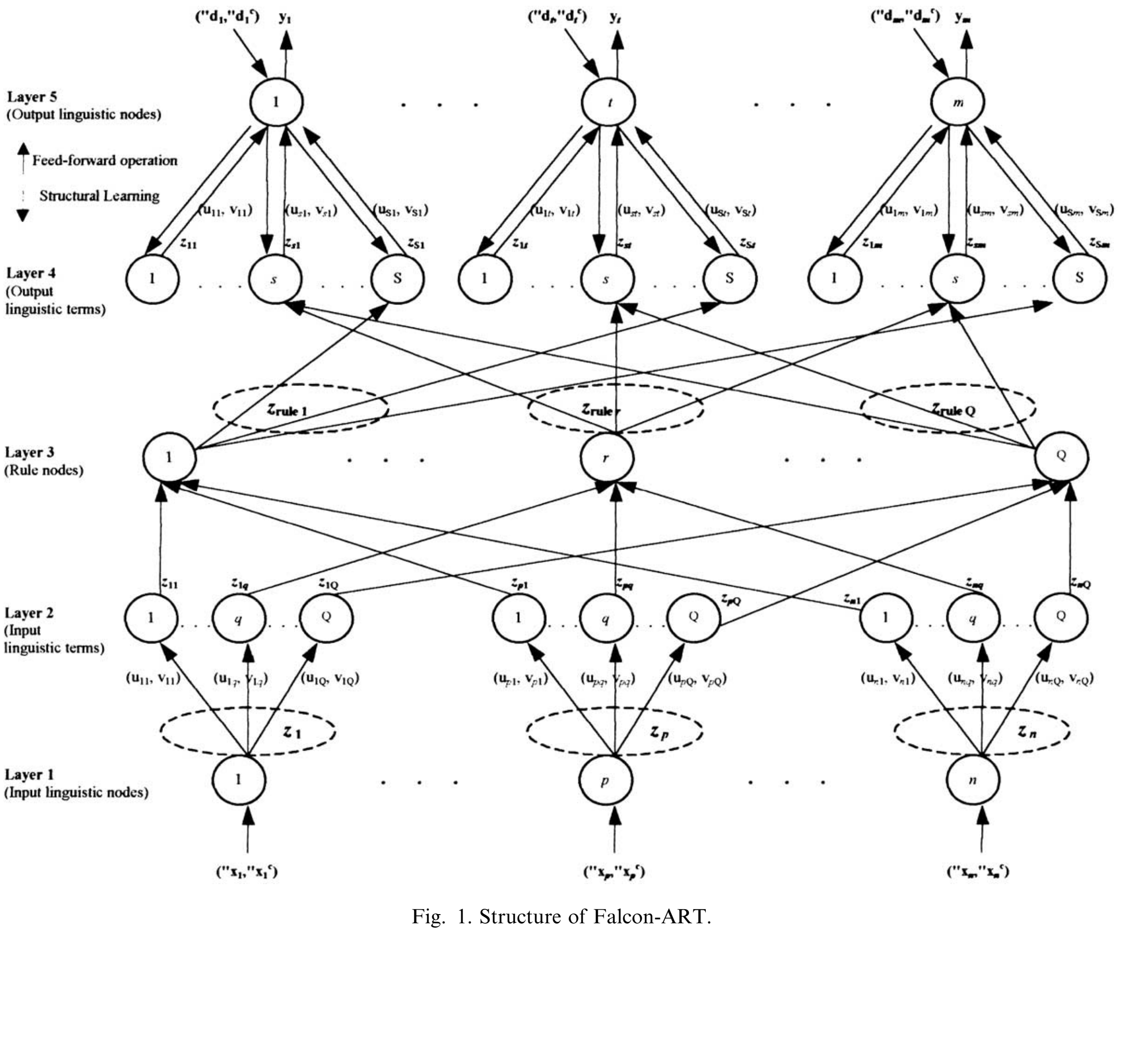

Fig. 1. Structure of Falcon-ART.

linguistic terms are represented as trapezoidal fuzzy sets. The linguistic terms represent fuzzy concepts such as tall, short, medium, fat and thin. The input linguistic variables and terms constitute the antecedent (condition) section of a fuzzy rule while the output linguistic variables and terms made up the consequent section of the rule. The IF-THEN construct is used to join the condition section to the consequent section. Each of the five layers in Falcon-ART is mapped to the respective elements of the fuzzy rule as shown in Fig. 1.

Prior to training, Falcon-ART has only the input and output layer to represent the input and output linguistic variables, respectively. There are n inputs and m outputs. The hidden layers for the input and output term nodes and the fuzzy rules are created and begin to grow as the learning cycle progresses. Falcon-ART dynamically partitions the input–output data spaces into trapezoidal fuzzy sets, tunes the trapezoidal membership functions representing the linguistic terms, and determines the proper network connections (fuzzy rules) in a single pass of the training data set. The fuzzy ART is used to perform the fuzzy clustering of the input–output spaces into fuzzy hyper-boxes (hyper-cubes). Falcon-ART then dynamically determines the proper fuzzy rules by connecting the appropriate input and output clusters (input and output hyper-boxes) through a mapping process. The *back-propagation* learning scheme is subsequently used to tune the input–output membership functions. Thus, Falcon-ART effectively combines the fuzzy ART algorithm for structural learning (formulation of the fuzzy rules) and the back-propagation algorithm for parameter learning (tuning of the membership functions).

With reference to Fig. 1, layer 1 nodes represent the input linguistic variables and each node in layer 2 acts as a one-dimensional trapezoidal fuzzy set for an input node. The label Q denotes the number of term nodes for each input variable. In Falcon-ART, each input node has the same number of term nodes. The nodes in layer 3 are the rule nodes and they form the fuzzy rule base of the Falcon-ART network. Layer 5 represents the output linguistic variables and layer 4 the respective output term nodes. The label S denotes the number of term nodes for each output node. After training, the number of rule nodes in layer 3 is determined by the number of term nodes for each input node. That is, there will be Q rules if there are Q terms for each input variable. This is because in Falcon-ART, each input term node is connected to only one rule node and only one term from each input node contributes to the antecedent of a fuzzy rule.

The output of the nodes in each layer is denoted by z. That is, the label z_p denotes the output from the pth input node in layer 1 and z_{pq} denotes the output of the qth term of the pth input node. The weights connecting each layer are unity unless otherwise shown. The trapezoidal membership functions of the term nodes in layer 2 and layer 4 are represented by a tuple consisting of the left (u) and right (v) flat points of the kernel. The tuple is shown as the weights of the links in layers 2 and 4. The membership function of the qth term node of the pth input node is denoted by (u_{pq}, v_{pq}). For the sth term node of the tth output node, its membership function is denoted by (u_{st}, v_{st}). During training, the inputs to Falcon-ART are the complementarily coded input vector $\boldsymbol{x}' \equiv (''x_1, ''x_1^c, ''x_2, ''x_2^c, \ldots, ''x_n, ''x_n^c)^{\mathrm{T}}$ and the complementarily coded desired output vector $\boldsymbol{d}' \equiv (''d_1, ''d_1^c, ''d_2, ''d_2^c, \ldots, ''d_n, ''d_n^c)^{\mathrm{T}}$. The output of Falcon-ART is denoted as $\boldsymbol{y} \equiv (y_1, y_2, \ldots, y_m)^{\mathrm{T}}$.

Based on the five layers of the Falcon-ART model shown in Fig. 1, Lin has developed an on-line learning algorithm **FALCON_ART** (Lin and Lee, 1991) to train the network.

3. Performance of Falcon-ART

To illustrate the shortcomings of the Falcon-ART network, a simple classification experiment using the Fisher's Iris data (Fisher, 1936) set is conducted. The data set is partitioned into a training set and a test set. The training set consists of one-third of the data points, (approx. 35%, i.e., 17 data points for each class) while the test set contains the remaining 65%. The results are cross-validated by using three different groups of training and test sets. They are denoted as CV1, CV2 and CV3. For the experiment, the following Falcon-ART parameters are used.

- Learning constant in the back-propagation algorithm $(\eta) = 0.005$.
- In-vigilance parameter in the fuzzy ART algorithm $(\rho_{in}) = 0.80$.
- Out-vigilance parameter in the fuzzy ART algorithm (ρ_{out}) is 0.80.
- Targeted cost error $E_{max} = 0.00005$.
- Sensitivity parameter of the trapezoidal membership function $(\gamma) = 30.00$.
- Training set = 35%.
- Test set = 65%.
- Number of input (input linguistic variables) = 4.
- Number of output (output linguistic variables) = 3.
- Maximum number of training iterations = 1000.

The parameter 'maximum number of training iterations' is used to ensure that the training cycle stops if the targeted cost error is unrealized. When it is realizable, then the training may stop before the specified 1000 iterations. The hardware configuration on which the experiment for Falcon-ART is conducted is listed.

- CPU = Intel Pentium III 450 MHz.
- Operating system = Microsoft Windows 95 (4.00.950 B).
- Memory available = 128 Mbytes.
- Hard disk used = Seagate 4.0 Gbytes.
- File system = Fat 32.
- Virtual memory = 32 bit.
- Disk compression = Not installed.

Training set of CV1 is shown in the Appendix A. Three outputs are used to represent the three classes of irises for the experiment. They are namely, Class 1-Setosa, Class 2-Virginica and Class 3-Versicolor. A cost function E defined in Eq. (2) is used to measure the convergence of the back-propagation learning algorithm of Falcon-ART during training.

$$E = 1/2 \sum_{i=1}^{m} (d_i^{(b)} - y_i^{(b)})^2 \text{ for } b = 1, \ldots, \Omega, \tag{2}$$

where m is the number of outputs from the Falcon-ART network, d_i is the desired output for the ith output node, y_i is the actual network output for the ith output node and, Ω is the number of training data in the training set.

Cost function E is computed for each training data in the training set and the total error (TE) for one training epoch is obtained using Eq. (3).

$$\mathrm{TE} = \sum_{b=1}^{\Omega} E^{(b)}, \tag{3}$$

where TE is the total error for one training epoch; Ω is the number of training data in the training set; and $E^{(b)}$ is the cost function for the bth training data in the training set.

Training stops when TE is less than the user preset E_{max} or the number of training cycle exceeds the maximum number of iterations specified. Fig. 2 shows the convergence of the back-propagation algorithm during training (CV1).

TE remains unchanged throughout the 1000 training iterations except for the initial rise. This is far from the preset targeted cost error of 0.00005. This problem can be explained when one examines the fuzzy sets that are derived using Falcon-ART in Fig. 3. The result shows that Virginica and Versicolor are very much alike from the large region of overlap in their respective fuzzy sets. In each of the numeric attribute, there is only a slight difference between the two classes. In Fig. 2, the TE does not converge because the fuzzy ART algorithm is unable to distinguish minute differences between fuzzy sets of different classes as it examines the membership values of the fuzzy sets together as a class. For example, a class 2 training data has membership values of {0.9, 0.8, 0.9, 0.9} for the class 2 fuzzy sets. Hence it gives a resemblance of 0.875 (0.9/4 + 0.8/4 + 0.9/4 + 0.9/4) to class 2. However, the same training data has

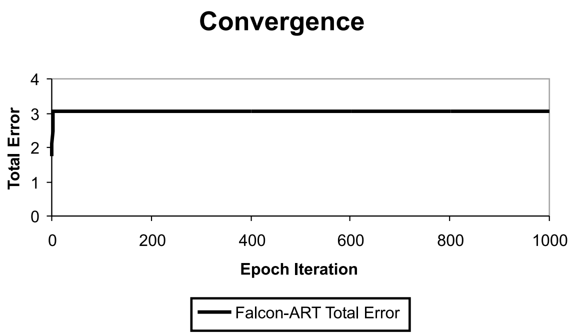

Fig. 2. A plot of TE against the number of training iterations.

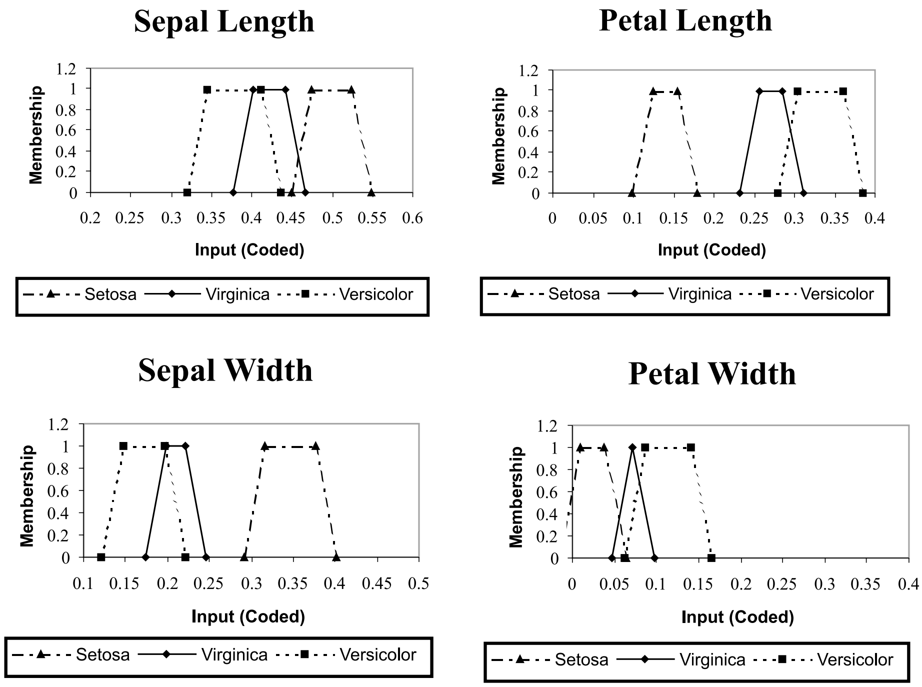

Fig. 3. Fuzzy sets derived using the Falcon-ART network.

membership values of {1.0, 0.6, 1.0, 0.6} for class 3 fuzzy sets. This gives a resemblance of 0.8 to class 3 and gives rise to a large error at the outputs.

Learning by back-propagation causes the corners of the membership functions to oscillate. This occurs when classes 2 and 3 training data are fed into the network. The network initially tunes itself towards class 2 outputs. When a class 3 training data is presented, the correction is reversed. Therefore, the TE remains constant. The initial rise of TE occurs between the first epoch and the second epoch when the fuzzy sets have not been completely formed. Hence, the generated error was much lesser than when all three classes of fuzzy sets had been formed.

This assessment reveals significant drawbacks of the Falcon-ART architecture. Firstly, the network is able to provide a satisfactory performance only when the classes of the input data are very different from each other (for example, between Setosa and Virginica). If they only differ by a slight difference (like in the case of Virginica and Versicolor), then this difference is likely to be missed by the network because the input iris data on the whole closely resembles both classes and hence, gives a similar firing strength for both classes. Because of this drawback, the classification results using the test sets are poor. Each test set consists of 65% of the original iris data set. There are 99 data points in a test set. The classification results for CV1, CV2 and CV3 are summarized in Table 1.

The learning and classification performance of the Falcon-ART is poor because class 2-Virginica and class 3-Versicolor are still inseparable even

Table 1
Iris classification using Falcon-ART[a]

Result	Group		
	CV1	CV2	CV3
C	69	83	73
M	30	16	26
U	0	0	0
C rate[b] (%)	69.70	83.84	73.74

[a] C = Classified; M = Misclassified; U = Unclassified; C rate = Classification rate.
[b] Mean = 75.76%, S.D. = 7.54%.

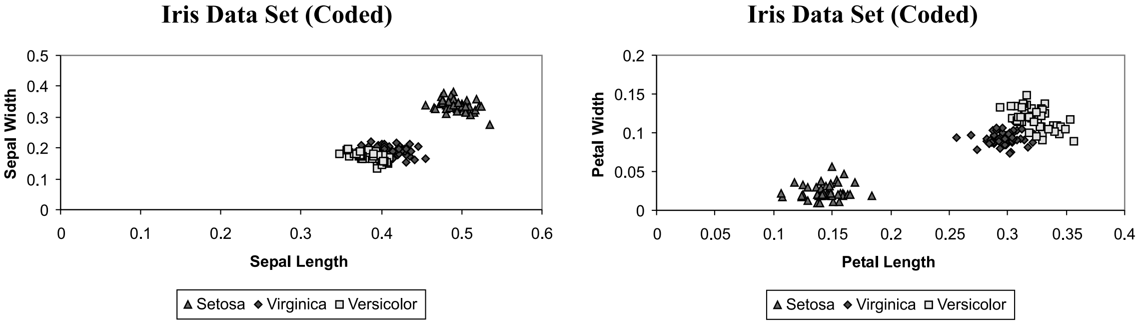

Fig. 4. Complementary coded iris data set.

with the complementary coding of the input data. Fig. 4 shows the plot for all 150 instances of irises in the data set after complementary coding.

It is clear that class 1-Setosa is linearly separable from the other two classes and classes 2 and 3 are inseparable. When one examines the sepal length attribute, both classes 2 and 3 have large overlaps. This also occurs in the sepal width, petal width and petal length attributes. Fig. 4 shows why back-propagation in Falcon-ART is unable to converge for the iris data set used and the poor classification performances of the network for the test groups. This is not limited only to Fisher's Iris data set but also for any data set with clusters (classes) that are inseparable.

Another problem of Falcon-ART is its weak resistance to noisy/spurious training data points in the training sets. The fuzzy ART algorithm incorporates a new training data point into an existing category as long as that category is the closest to the training data point and passes the resonance test. However, this training data may be far from most of the data in the category and is incorporated because the resonance parameter is set small enough to accommodate such an inclusion. Inclusion is done through the *fast learning* scheme in the structural learning step of the Falcon-ART architecture.

The learning cycle in Falcon-ART very much depends on the setting of the targeted cost error. If this is set to a large value, then learning will terminate quickly. On the other hand, if it is set to too small a value, then learning may not cease. This is likely to occur for data sets like Fisher's Iris data since there are classes that are inseparable from one another. This makes comparison of results difficult since the optimal value of $E_{\max}$ is often obtained through trial-and-error and is different for different data sets. This makes Falcon-ART application dependent. An alternative is to terminate the learning process when the difference in the TE for consecutive training epochs is sufficiently small, as in the case of Kohonen's competitive learning rule (Kohonen, 1989).

In addition, the use of complementary coding in Falcon-ART can cause the learning efficiency of the network to deteriorate. Consider when two different training data points with magnitudes that are proportional to each other but belonging to different classes enter the network. Complementary coding under this situation will yield the same input for the two data points. In addition, Falcon-ART cannot handle single input/output application because of the need for complementary coding. Moreover, the fuzzy sets that are derived using Falcon-ART cannot be directly interpreted because all the values used are normalized. Hence, the fuzzy rules that are generated are non-intuitive to our understanding of the dynamics of the problem domain. The next section describes the new architecture called Falcon-MART that attempts to overcome the shortcomings highlighted.

4. Dynamics of Falcon-MART

Falcon-MART is developed to address the inherent deficiencies of the Falcon-ART network.

The deficiencies are: (1) unsatisfactory classification performance when the classes of input data are very similar to each other; (2) susceptible to noisy training data and outlier; (3) use of preset error parameter to terminate the learning process and (4) deterioration of learning efficiency. The performance of Falcon-MART is evaluated in Section 5 using three different experiments.

4.1. Poor performances for similar classes of input pattern

The Falcon-ART architecture compares an input pattern against its stored clusters as a whole and averages out the incompatibility against each of the respective numeric attributes (four numeric attributes as in the iris data set). As a result, an input that is close to 2 or more stored clusters will give similar firing for those rules representing the clusters. To overcome this drawback, a method is proposed to enlarge/magnify any difference between the respective elements in two given sets of membership values. This method is implemented in Falcon-MART and is outlined in the algorithm *Magnify*.

Algorithm Magnify:

Variables: Sum = 0, Weight = 0, array *value* of size m. Assume m-dimensional stored clusters in the Falcon-ART network.

Step 1 (*Sorting*). Store the respective membership values for a cluster due to the present input pattern in the array value of m-dimension and sort the array in descending order, with the largest value at element 0 and the smallest value at element $m-1$. That is, value $[0] \geqslant$ value $[1] \geqslant \cdots \geqslant$ value $[m-1]$.

Step 2 (*Weighted averaging*). Multiply each element e_i in the array with a weight w_i that is derived using Eq. (4) and add this product to Sum. Add w_i to Weight.

$$w_i = 2^{3i}, \tag{4}$$

$$\mathrm{Sum} = \sum_{i=0}^{m-1} e_i w_i,$$

$$\mathrm{Weight} = \sum_{i=0}^{m-1} w_i,$$

where i is the index position of the element e_i in the array value starting with index 0; and m is the number of elements in array value.

Step 3 (*Firing*). Obtain the firing strength F of the cluster due to the present input pattern using Eq. (5).

$$F = \mathrm{Sum}/\mathrm{Weight}, \tag{5}$$

where n is the number of numeric attributes in the data set.

This algorithm gives the largest weight to the smallest element in the array value. Hence, any small differences between the membership values of two clusters are magnified by a large factor.

End Magnify

An advantage of the proposed algorithm is its ability to magnify the difference in membership value between the same attribute of different membership sets and hence producing a very different firing strength for membership sets that have the same cardinality. For example, consider two fuzzy membership sets $A\{0.8, 0.8, 0.8, 0.8\}$ and $B\{1, 0.8, 1, 0.4\}$ that has the same cardinality. Under Falcon-ART, both will have a firing strength of 0.8. However, using Falcon-MART, set A will give a firing strength of 0.8 but set B will give a firing strength of 0.45. Moreover, it preserves the firing strength of a membership set in which all the elements are the same. For example, set A would give a firing strength of 0.8 because each of its elements has 80% compatibility.

4.2. Weak resistance to noisy training data

The Falcon-ART architecture uses the fast-learn rule of the fuzzy ART algorithm to include an input pattern into a chosen fuzzy rule. This method performs well if all the data has strong correlation to one another; that is, they cluster closely to each other. The network is then able to quickly form the correct fuzzy hyper-boxes. However, when noisy data exists among those used for training, the learning efficiency drops because such data upsets the equilibrium of the system. The data is included because the resonance parameter is small. Hence, it becomes easy to pass the resonance test. A more stable and progressive learning rule is adopted by Falcon-MART to minimize the effects of noisy data. This is shown in Eq. (6).

$$\boldsymbol{w}(t+1) = \beta(\boldsymbol{I} \wedge \boldsymbol{w}(t)) + (1-\beta)\boldsymbol{w}(t), \tag{6}$$

where $\boldsymbol{w}(t)$ is the weight vector of the selected cluster at learning step t; $\boldsymbol{I}$ is the input to the network; and $0 < \beta < 1$.

By setting β very close to 1 approximates the original fast-learning rule since the input pattern is heavily absorbed into the chosen fuzzy rule and hence may cause instability. On the other hand, setting β very close to 0 reduces the *plasticity* of the network. Hence, β is set at 0.5 to achieve a compromise for the *stability–plasticity dilemma*.

4.3. Termination of learning process depends heavily on a preset error parameter

Experimental results for the Falcon-ART in Section 3 have shown that the back-propagation algorithm fails to converge to the targeted cost error when the network is used to learn and classify data that are close to one another. The TE is constant throughout the training iterations. Hence, the learning process can in fact terminate when the difference between TE of two consecutive iterations is sufficiently small. Therefore, in Falcon-MART, a parameter ε is introduced to allow the network to terminate the learning process when the change in TE between two consecutive epochs is smaller than ε. Now ε replaces the targeted cost error as the termination criterion. This pseudo termination of the back-propagation learning algorithm is similar to the forced termination in the Kohonen's rule of competitive learning. It eliminates the need to determine the optimal value of $E_{\max}$ as in Falcon-ART through trial-and-error.

4.4. Complementary coding

The problem of complementary coding in Falcon-ART has been illustrated in Section 3. To overcome this problem, absolute-valued data is used in Falcon-MART instead of complementarily coded data. In Falcon-ART, complementary coding is used to prevent category proliferation. The number of category to be formed is determined by the vigilance parameters ρ_{in} and ρ_{out} and the resonance test performed by the fuzzy ART algorithm. However, the resonance test and the computation of the choice function to determine the best-fit cluster for the present input require that the inputs be normalized to unity. Hence, changing the form of the input data implies modifications to these two operations. Firstly, to determine the cluster that is a best fit to the present input using the choice function is essentially selecting the cluster that has the strongest rule firing due to the input pattern.

The computation of the choice function in Falcon-ART determines the firing strength of the stored clusters for a given input. Hence, in Falcon-MART, a firing of layers 1, 2 and 3 can be activated and the best-fit cluster is determined from the outputs of the rule nodes in layer 3 as the one with the largest rule firing strength. This is expressed in Eq. (7) for the best-fit cluster R.

$$R = \arg\max_{r}(z_{\text{rule}\,1}, z_{\text{rule}\,2}, \ldots, z_{\text{rule}\,r}, \ldots, z_{\text{rule}\,Q}). \tag{7}$$

Eq. (8) describes the input resonance test in Falcon-ART. A cluster is said to pass the test when Eq. (8) evaluates as true. When a cluster passes the resonance test, the input training data is incorporated into the cluster. The output resonance test is similarly defined.

$$\frac{|I \wedge W_R|}{|I|} \geqslant \rho_{\text{in}}, \quad \rho_{\text{in}} \in [0,1], \tag{8}$$

where ρ_{in} is the input vigilance parameter set by user; I is the input vector to the network; and W_R is the weight vector representing the best-fit cluster.

The cardinality of the complementary coded input vector I is n, where n is the number of attributes in the data set. To pass the resonance test in Falcon-ART, minimal changes to the kernels of the trapezoidal fuzzy sets of a cluster is to be made so that the average kernel of the updated cluster (after the input training data is incorporated) must not exceed $1 - \rho_{\text{in}}$.

In Falcon-MART, to obtain the functionally equivalent input resonance test of Falcon-ART, the maximum value of each of the input attributes and training data of absolute values are used instead of the complementarily coded data. Now, the corners of the trapezoidal functions are of absolute values. Therefore, to pass the resonance test and have the present input included into its kernel, the best-fit cluster R must satisfy the condition defined in Eq. (9).

$$\sum_{i=1}^{n} K_{Ri} \leqslant (1-\rho_{\text{in}})\text{MaxSum} \tag{9}$$

and $\text{MaxSum} = \sum_{i=1}^{n}$ maximum value of input attribute i, where n is the number of input attributes; K_{Ri} is the range of kernel of attribute i of best-fit cluster R; and ρ_{in} is the user preset input vigilance parameter.

5. Performance of Falcon-MART

The performance of Falcon-MART is evaluated using three different experiments. The first experiment demonstrates the efficiency of Falcon-MART over Falcon-ART using the Fisher's Iris data set. The second experiment evaluates the modeling capability of Falcon-MART against the classical *multi-layered perceptron* (MLP) network using a set of traffic flow data. The last experiment uses a set of phoneme data to demonstrate the clustering ability of Falcon-MART against the traditional *K-nearest-neighbor* (*K*-NN) classifier. All the experiments are carried out on the same hardware configuration as listed in Section 3.

5.1. Experiment 1: classification of Iris data

To benchmark against the performance of Falcon-ART, the Fisher's Iris data set with the same training and test sets are used. The network parameters common to both architectures are kept constant with the exception of the sensitivity parameter, γ. The parameter γ is the gradient of the slope of the trapezoidal membership function. This is set to 2.0 in Falcon-MART to provide a buffer of 0.5 on either side of the kernel of the trapezoidal function. The following Falcon-MART parameters are used in the assessment.

- Learning constant in the back-propagation algorithm $(\eta) = 0.005$.
- In-vigilance parameter in the fuzzy ART algorithm $(\rho_{\text{in}}) = 0.80$.
- Out-vigilance parameter in the fuzzy ART algorithm (ρ_{out}) is 0.80.
- Termination criterion $\varepsilon = 0.00005$.
- Sensitivity parameter of the trapezoidal membership function $(\gamma) = 2.00$.
- Training set = 35%.
- Test set = 65%.
- Number of input (input linguistic variables) = 4.
- Number of output (output linguistic variables) = 3.
- Maximum number of training iterations = 1000.

Fig. 5 shows the convergence of the back-propagation algorithm in Falcon-MART for the training set of CV1.

Falcon-MART completes the learning cycle in 11 iterations. Moreover, comparing against the Falcon-ART results in Fig. 2, the TE has converged and is much lower than that of the original Falcon-ART architecture. The fuzzy sets derived using the Falcon-MART network is shown in Fig. 6.

From Fig. 6, it can be seen that Class 2-Virginica and Class 3-Versicolor are still very much alike. In the fuzzy sets for the sepal width and petal width attributes, the two classes are almost identical. Hence, the distinguishing properties of these two classes lie in the sepal length and petal length attributes. The classification results using the training and test sets of CV1, CV2 and CV3 are summarized in Table 2.

Comparing against the results of Falcon-ART in Table 1, Falcon-MART offers a much-improved

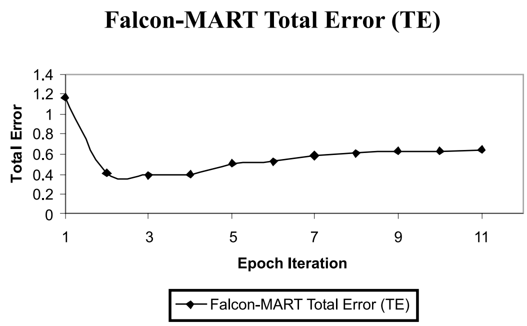

Fig. 5. TE against number of training iteration.

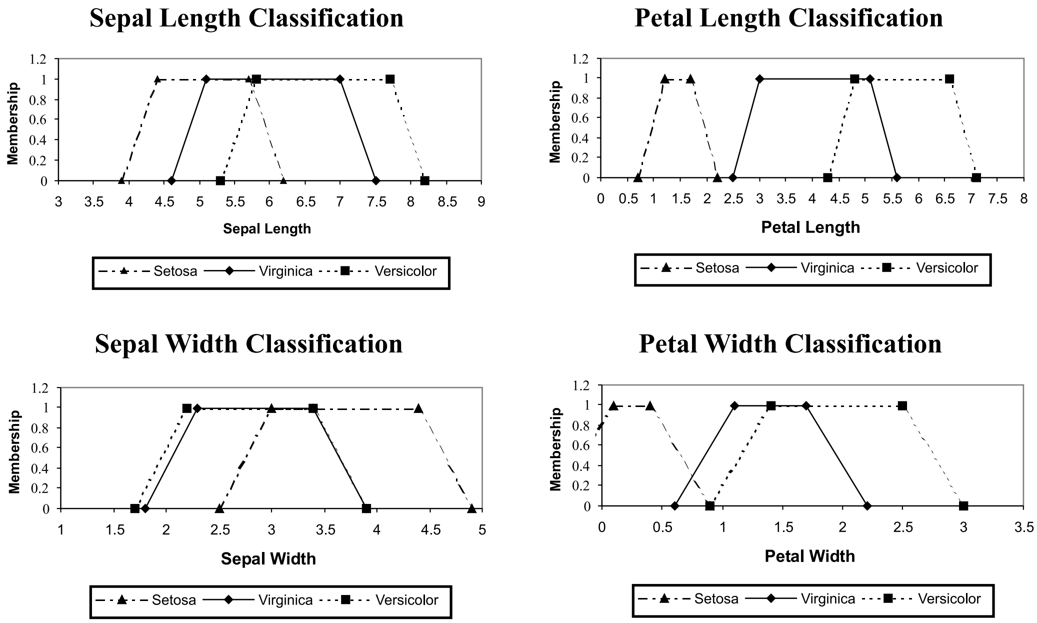

Fig. 6. Fuzzy sets of the three classes of irises derived using Falcon-MART.

Table 2
Iris classification using Falcon-MART[a]

Result	Group		
	CV1	CV2	CV3
C	98	95	89
M	1	4	10
U	0	0	0
C rate[b] (%)	98.99	95.96	89.90

[a] C = Classified; M = Misclassified; U = Unclassified; C rate = Classification rate.
[b] Mean = 94.95%, S.D. = 4.64%.

performance for the classification of the iris data in the test sets. In addition to the smaller number of training epochs required, it also has a higher mean classification rate of 94.95% for correct classification as compared to the Falcon-ART of 75.76%. It also has a higher noise tolerance than Falcon-ART as shown by the smaller value in the S.D. of the classification rate.

In addition to the classification rate, three other benchmark criteria are used to compare the two architectures. They are the training time of the network, recall time and memory usage. The CPU timing for training and recall and the memory usage for the two network architectures are shown in Table 3. The variable *h* denotes the size of a *long double* variable type. Here, only the memory space used by the networks to store the fuzzy sets of the input space partition is used in the comparison.

The timings shown in Table 3 are obtained by simulating the experiments 10 times each for CV1, CV2 and CV3. The mean and the S.D. of the timings are shown in the table. Falcon-ART requires a longer CPU training time because of its failure to converge to the preset termination criterion ($E_{max} = 0.00005$) and hence training continued to the preset limit of 1000 cycles before terminating. On the other hand, Falcon-MART

Table 3
CPU timings and memory usage of various networks

Network (parameter)	CPU time (s)	Recall time (s)	Memory usage (h)
Falcon-ART ($\eta = 0.005$)	98.75 ± 8.57	0.07 ± 0.051	24
Falcon-MART ($\eta = 0.005$)	1.46 ± 0.39	0.07 ± 0.045	24

requires a mean CPU training time of just 1.46 s. This is less than 2% of the average training time for Falcon-ART. The memory usage of both networks is 24 h. They are used essentially for the storage of the two corners of the kernel of the membership functions in the input term nodes of hidden layer 2. The iris data set has four numeric inputs and both networks derived three fuzzy rules (therefore three terms per input node). Hence, the total memory requirement is 4 inputs $\times$ 3 input terms $\times$ 2 corners $\times$ size of long double variable type $=$ 24 h. The recall time for both architectures are similar.

By assigning semantics (labels) to the fuzzy sets of each numeric attribute; namely: short (S), medium (M) and long (L), the fuzzy rules derived by Falcon-ART in the classification of the Fisher's Iris data set (from CV1) can be expressed as follows:

- Rule 1a: **If** sepal length is L and sepal width is L and petal length is S and petal width is Ss, **then** iris is Setosa.
- Rule 1b: **If** sepal length is M and sepal width is M and petal length is M and petal width is M, **then** iris is Virginica.
- Rule 1c: **If** sepal length is S and sepal width is S and petal length is L and petal width is L, **then** iris is Versicolor.

Examining the data distributions of the training set of CV1 in the Appendix A, the fuzzy rules derived by Falcon-ART do not directly represent the training set because complementary coded data are used for training. Such representation is non-intuitive, as we tend to classify the irises by examining their numeric attributes and making comparisons within the same attribute. As a result, the fuzzy rules derived by Falcon-ART do not match the ones deduced from the data distributions found in the Appendix A.

On the other hand, the fuzzy rules derived by Falcon-MART for CV1 (Fig. 6) is as followed:

- Rule 2a: **If** sepal length is S and sepal width is L and petal length is S and petal width is S, **then** iris is Setosa.
- Rule 2b: **If** sepal length is M and sepal width is M and petal length is M and petal width is M, **then** iris is Virginica.
- Rule 2c: **If** sepal length is L and sepal width is S and petal length is L and petal width is L, **then** iris is Versicolor.

Comparing this set of fuzzy rules with the one found in the Appendix A, it is observed that both sets of rules are similar except for the definition of the sepal width attribute in the classification of the irises. Falcon-MART derived that Virginica and Versicolor should have medium and short sepal widths respectively whereas the data distributions show the opposite. This is unsurprising as a closer examination of the data distributions in the Appendix A reveals that except for the sepal width attribute, the other three attributes can be clearly classified as shown by the data distributions. However, for the sepal width attribute, both Virginica and Versicolor are very similar in their data distributions. With respect to Fig. 6, the fuzzy sets derived by Falcon-MART also suggest the same problem. Hence, due to this great similarity between Virginica and Versicolor in the sepal width attribute, it is unsurprising that Falcon-MART fails to differentiate the two classes of irises. This is due to the clustering characteristic of the fuzzy ART algorithm in Falcon-MART. Although the original learning rule has been modified to make Falcon-MART more resistive to noisy data or outliers, it does not entirely remove the deficiency. And from the data distributions in the Appendix A, it is clear that Versicolor has a wider distribution than Virginica for the sepal width attribute. This resulted in the slight inaccuracy in the fuzzy rules derived by Falcon-MART.

5.2. Experiment 2: modeling of traffic flow data

This experiment is conducted to evaluate the effectiveness of the Falcon-MART network in universal approximation and data modeling using traffic flow data. The raw traffic flow data for the experiment was obtained from Tan (1997), courtesy of School of Civil and Structural Engineering (CSE), NTU, Singapore. 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 B) using loop detectors embedded beneath the road surface. These inductive loop detectors were pre-installed by the Land Transport

Authority (LTA) of Singapore in 1996 along major roads to facilitate traffic flow data collection. 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 purpose of this experiment is to model the traffic flow trend at the site using 10 inputs, namely time and traffic volume, speed and traffic density of the three lanes using Falcon-MART. The trained network is then used to obtain prediction for traffic density of a particular lane at a time $t + \tau$, where $\tau = 5$, 15, 30, 45 and 60 min. Fig. 7 shows a plot of the traffic flow density data for the three straight lanes spanning a period of 6 days from 5–10 September 1996.

The following Falcon-MART parameters are used in the experiment.

- Learning constant in the back-propagation algorithm $(\eta) = 0.005$.
- In-vigilance parameter in the fuzzy ART algorithm $(\rho_{\text{in}}) = 0.70$.
- Out-vigilance parameter in the fuzzy ART algorithm (ρ_{out}) is 0.95.
- Termination criterion $\varepsilon = 0.005$.
- Sensitivity parameter of the trapezoidal membership function $(\gamma) = 5.00$.
- Training set = 40%.
- Test set = 60%.
- Number of input (input linguistic variables) = 10.
- Number of output (output linguistic variables) = 1.
- Maximum number of training iterations = 1000.

For the experiment, three groups of training and test sets are used. They are CV1, CV2 and CV3. The training windows are labeled as such in Fig. 7. To compute the accuracy of the predictions by Falcon-MART, the square of the Pearson product–moment correlation value (R^2) is used. The prediction made by Falcon-MART for $\tau =$ 60 min for lane 1 traffic density is shown in Fig. 8. The variation of R^2 against τ is shown in Fig. 9 and the results of the experiment are summarized in Table 4.

As expected, the correlation of the data points decreases as τ increases. Hence, the accuracy of the prediction by the Falcon-MART network is expected to fall. This is verified by the decrease in R^2 value as τ increases. The prediction results for all the three lanes for various τ values are given in Table 4.

The same set of experiment is repeated using the MLP network with 10 input nodes, five hidden nodes and one output node. The bipolar sigmoidal function with an output range of $[-1, 1]$ is used as the activation function for the hidden and output nodes. The input nodes simply relay the input signals to the hidden nodes. The MLP network is trained with the back-propagation algorithm. During the prediction phase, the network functions in a feed-forward mode. The results for the

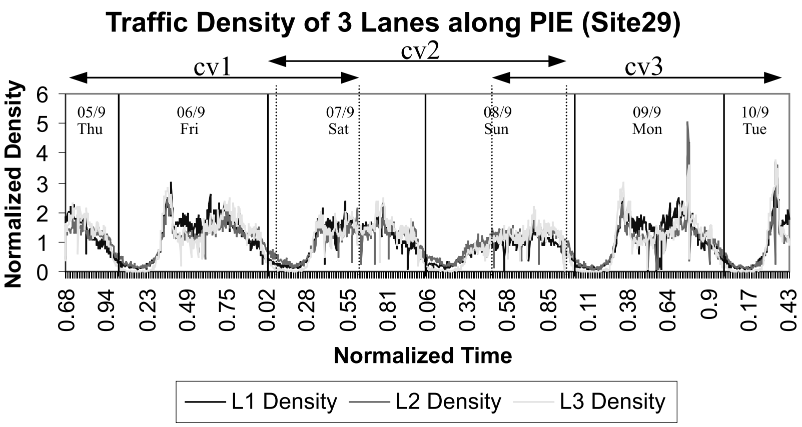

Fig. 7. Traffic density of three straight lanes along PIE.

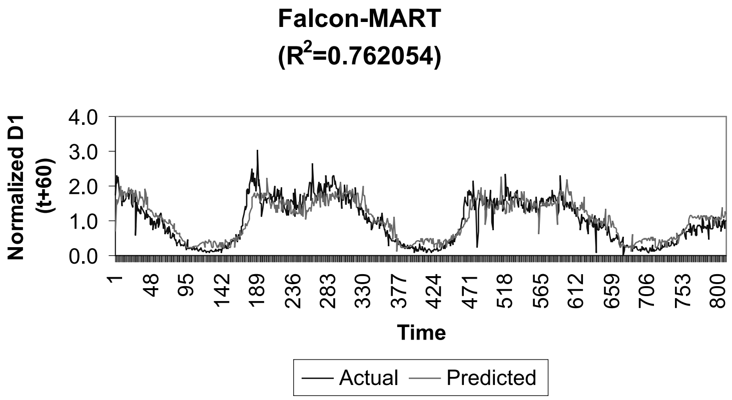

Fig. 8. Prediction of lane 1 density for $\tau = 60$ min using Falcon-MART.

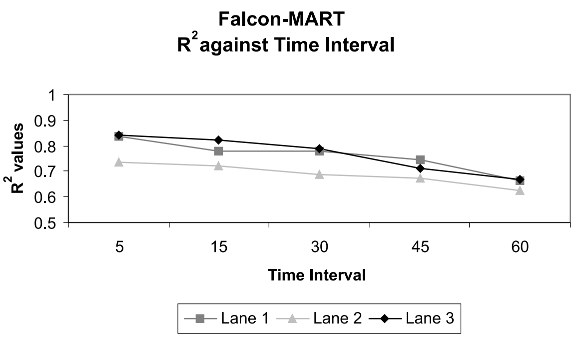

Fig. 9. Avg R^2 variations against τ.

prediction of lane 1 traffic flow density for $\tau = 5$ min are given in Table 5.

Comparing against the results of Falcon-MART in Table 4, the MLP can achieve better results. However, the trained MLP is a black-box and the linguistic rules defining the traffic flow pattern cannot be extracted from the MLP network. Hence, Falcon-MART may not be as efficient an architecture as MLP in data modeling and universal approximation, but its advantage lies in its ability to formulate a set of understandable fuzzy rules to describe the problem domain.

5.3. Experiment 3: clustering of phoneme data

This experiment is conducted to determine the clustering capability of Falcon-MART against a traditional classifier such as the *K*-NN classifier. The data set used is a set of phoneme data that is available on-line at the ELENA website.

The aim of the phoneme database is to distinguish between nasal and oral vowels. There are thus two different classes, namely Class 0-Nasals and Class 1-Orals.

This database contains vowels coming from 1809 isolated syllables (for example: pa, ta, pan, etc.). Five different attributes were chosen to characterize each vowel. They are the amplitudes of the five first harmonics, normalized by the total energy (integrated on all the frequencies). Each harmonic is signed: positive when it corresponds to a local maximum of the spectrum and negative otherwise.

Three observation moments have been kept for each vowel to obtain 5427 different instances:

– Observation corresponding to the maximum total energy.

– Observations taken 8 ms before and 8 ms after the observation corresponding to this maximum total energy.

From these 5427 initial values, 23 instances for which the amplitude of the five first harmonics was zero were removed, leading to the 5404 instances of the present database. The patterns are presented in a random order. Due to the size of the data set, it is partitioned into two sets: 20% for the training set and the remaining 80% for the test set. A total of five groups of training and test sets are used for the experiment. They are labeled as CV1, CV2, CV3, CV4 and CV5. The data for the training sets

Table 4
Prediction results of traffic flow density by Falcon-MART

τ	CV1_R^2	CV2_R^2	CV3_R^2	Avg R^2
Lane 3				
5	0.818516	0.830194	0.875139	0.841283
15	0.796118	0.830353	0.844567	0.823679
30	0.763694	0.78257	0.817728	0.787997
45	0.656274	0.720153	0.757044	0.711157
60	0.601274	0.682837	0.720392	0.668168
Lane 2				
5	0.715375	0.675857	0.820863	0.737365
15	0.6848	0.706844	0.778582	0.723409
30	0.667232	0.610182	0.781189	0.686201
45	0.6222	0.64312	0.758077	0.674466
60	0.572623	0.604237	0.692918	0.623259
Lane 1				
5	0.792404	0.828861	0.884448	0.835238
15	0.708169	0.801472	0.822332	0.777324
30	0.73429	0.789983	0.807692	0.777322
45	0.682927	0.735286	0.812296	0.743503
60	0.601775	0.621251	0.762054	0.661693

Table 5
Prediction of traffic flow density for lane 1 at $\tau = 5$ min by MLP

Lane 1				
τ	CV1_R^2	CV2_R^2	CV3_R^2	Avg R^2
5	0.834817	0.847881	0.881109	0.854602

do not repeat among the five groups. For this experiment, the Falcon-MART network has five inputs and one output. The following Falcon-MART parameters are used.

- Learning constant in the back-propagation algorithm $(\eta) = 0.005$.
- In-vigilance parameter in the fuzzy ART algorithm $(\rho_{\text{in}}) = 0.70$.
- Out-vigilance parameter in the fuzzy ART algorithm (ρ_{out}) is 0.70.
- Termination criterion $\varepsilon = 0.0005$.
- Sensitivity parameter of the trapezoidal membership function $(\gamma) = 8.00$.
- Training set = 20%.
- Test set = 80%.
- Number of input (input linguistic variables) = 5.
- Number of output (output linguistic variables) = 1.
- Maximum number of training iterations = 1000.

The classification results using Falcon-MART is summarized in Table 6.

In comparison, the confusion matrix obtained with the *K*-NN classifier (result obtained from ELENA and tested with the leave-one-out method with $k = 20$) is shown in Table 7. In this case, the TE rate is 14.2%.

Comparing the results in Tables 6 and 7, *K*-NN seems to have the better clustering capability. However, this is achieved with the whole data set as the training set with the exception of one data point that is used to perform the classification test. The classification test for *K*-NN is performed a total of 5404 times. On the other hand, Falcon-MART is able to achieve a reasonable error rate by using just 20% of the data set for training. Moreover, IF-THEN fuzzy rules can be extracted from the Falcon-MART network to describe the clustering behavior of the data set. This cannot be done using the *K*-NN classifier.

6. Conclusions

The Falcon-ART architecture is a fuzzy neural network that is developed with the aim of quickly

Table 6
Phoneme classification results using Falcon-MART[a]

Group	Results		
	Confusion matrix		
CV1		Class 0	Class 1
	Class 0	94.37%	5.63%
	Class 1	54.77%	45.23%
	Error rate = 20.05%		
CV2		Class 0	Class 1
	Class 0	94.36%	5.64%
	Class 1	63.50%	36.50%
	Error rate = 22.62%		
CV3		Class 0	Class 1
	Class 0	93.75%	6.25%
	Class 1	59.84%	40.16%
	Error rate = 21.98%		
CV4		Class 0	Class 1
	Class 0	90.33%	9.67%
	Class 1	47.55%	52.45%
	Error rate = 20.79%		
CV5		Class 0	Class 1
	Class 0	93.05%	6.95%
	Class 1	55.38%	44.62%
	Error rate = 21.16%		

[a] Mean error rate = 21.32%; S.D. = 1.006%. Class 0 – Mean classification rate = 93.17%; S.D. = 1.68%. Class 1 – Mean classification rate = 43.79%; S.D. = 7.24%.

Table 7
Phoneme classification result using K-NN ($k = 20$) and leave-one-out method

K-NN ($k = 20$)		
Confusion matrix		
	Class 0	Class 1
Class 0	91.40%	8.60%
Class 1	27.80%	72.20%
Error rate = 14.20%		

deriving the fuzzy sets from training data and to formulate fuzzy rules that accurately reflects the dynamics of the problem domain in a single pass of the training data. However, there are several shortcomings in the architecture as illustrated by the results of the experiments carried out. Falcon-MART is proposed in this paper to overcome these shortcomings. Both Falcon-ART and Falcon-MART identify trapezoidal membership functions that are convex and normal from the training data. Experimental results have shown that Falcon-MART produces superior performance to those derived from Falcon-ART. In addition, the training cycle for Falcon-MART is significantly reduced. Moreover, the fuzzy rules derived using Falcon-MART provide a closer representation to the training data than those derived using Falcon-ART. However, due to the clustering nature of the fuzzy ART algorithm used in Falcon-MART, there is a slight inaccuracy in the derived fuzzy rules. As this inaccuracy is inherent in the clustering technique used, efforts have been made to substitute another clustering technique into the basic *Falcon* architecture to replace fuzzy ART to identify the membership functions in the fuzzy rule base. Some of the clustering techniques under consideration are fuzzy C-means (FCM) (Bezdek et al., 1987), modified learning vector quantization (MLVQ) (Kohonen, 1989; Quek et al., 1998), fuzzy Kohonen partitioning (FKP) and pseudo fuzzy Kohonen partitioning (PFKP) (Quek and Ang, 1999).

Appendix A

To compare the fuzzy rules derived using Falcon-ART and Falcon-MART, the data distributions for the training set of CV1 according to each of the four numeric attribute are shown as histograms in Fig. 10. The data samples are grouped according to the ranges specified and the frequencies are tabulated.

Since the four numeric attributes are physical measurements of length, the semantics of short (S), medium (M) and long (L) can be used to describe the data distribution of the irises according to a particular attribute. From the distributions of the data samples in the training set, the following set of fuzzy rules on the classification of the irises is deduced

- Rule 3a: **If** sepal length is S and sepal width is L and petal length is S and petal width is S, **then** iris is Setosa.
- Rule 3b: **If** sepal length is M and sepal width is S and petal length is M and petal width is M, **then** iris is Virginica.
- Rule 3c: **If** sepal length is L and sepal width is M and petal length is L and petal width is L, **then** iris is Versicolor.

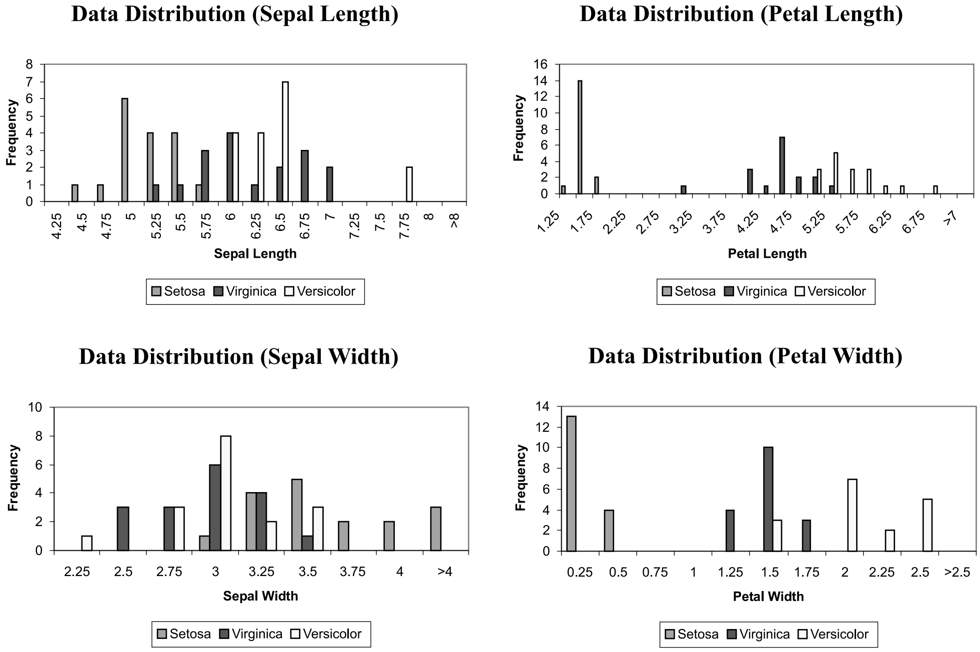

Fig. 10. Data distributions according to numeric attribute.

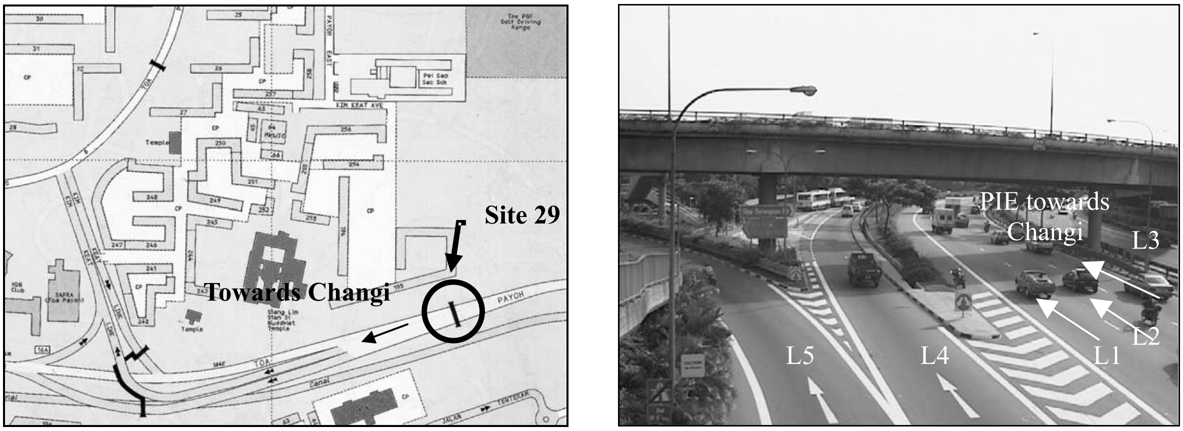

Fig. 11. Location of site 29 along PIE (Singapore).

This set of fuzzy rules is used to benchmark those derived using Falcon-ART and Falcon-MART.

Appendix B

The site location (site 29) at which traffic flow data for the second experiment is collected is shown in the map (Fig. 11). The arrows show the direction of traffic flow.

References

Bezdek, J.C., Hathaway, R.J., Tucker, W.T., 1987. An improved convergence theory for the fuzzy C-means clustering algorithms. In: Analysis of Fuzzy Information, Vol. 3. CRC Press, Boca Raton, pp. 123–129 (Chapter 8).

Carpenter, G.A., Grossberg, S., 1987a. A massively parallel architecture for a self-organizing neural pattern recognition machine. Comput. Vision Graphics Image Process. 37, 54–115.

Carpenter, G.A., Grossberg, S., 1987b. ART2: Self-organization of stable category recognition codes for analog input patterns. Appl. Opt. 26, 4919–4930.

Carpenter, G.A., Grossberg, S., 1988. The ART of adaptive pattern recognition by a self-organizational neural network. Computer 21 (3), 77–88.

Carpenter, G.A., Grossberg, S., 1990. ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. Neural Networks 3 (2), 129–152.

Carpenter, G.A., Grossberg, S., Rosen, D.B., 1991. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771.

ELENA website: http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7, Part II. pp. 179–188.

Grossberg, S., 1976. Adaptive pattern classification and universal recoding: II Feedback, expectation, olfaction, illusions. Bio. Cybern. 23, 187–202.

Kohonen, T.K. 1989. Self-Organization and Associative Memory, third ed. Springer, New York.

Lin, C.T., Lee, C.S.G., 1991. Neural-network-based fuzzy logic control and decision system. IEEE Trans. Comput. 40 (12), 1320–1336.

Quek, H.C., Ang, K.K., 1998. MLVQ: Modified learning vector quantization with initial partitioning and error descent. NTU Singapore.

Quek, H.C., Ang, K.K., 1999. FKP: An automatic derivation of fuzzy membership functions using novel fuzzy Kohenen partitioning algorithms. NTU Singapore.

Tan, G.K., 1997. Feasibility of predicting congestion states with neural networks, Final Year Project, Nanyang Technological University, CSE.