Engineering Letters, 14:1, EL_14_1_29 (Advance online publication: 12 February 2007)

A General Reflex Fuzzy Min-Max Neural Network

A.V. Nandedkar, *Member, IAENG*, P.K. Biswas, *Member, IAENG*

Abstract—**"A General Reflex Fuzzy Min-Max Neural Network" (GRFMN) is presented. GRFMN is capable to extract the underlying structure of the data by means of supervised, unsupervised and partially supervised learning. Learning under partial supervision is of high importance for the practical implementation of pattern recognition systems, as it may not be always feasible to get a fully labeled dataset for training or cost of labeling all samples is not affordable. GRFMN applies the acquired knowledge to learn a mixture of labeled and unlabeled data. It uses aggregation of hyperbox fuzzy sets to represent a class or cluster. A novel reflex mechanism inspired from human brain is used to solve the problem of class overlaps. The reflex mechanism consists of compensatory neurons which become active if the test data belongs to an overlap region of two or more hyperboxes representing different classes. These neurons help to approximate the complex topology of data in a better way. The proposed new learning approach to deal with partially labeled data and inclusion of compensatory neurons, has improved the performance of GRFMN significantly. The advantages of GRFMN are its ability to learn the data in a single pass through and no requirement of retraining while adding a new class or deleting an existing class. The performance of GRFMN is compared with "General Fuzzy Min-Max Neural Network" proposed by Gabrys and Bargiela. The experimental results on real datasets show a better performance of GRFMN.**

***Index Terms*—Fuzzy Min Max Neural Network, Reflex Mechanism, Compensatory Neurons.**

I. Introduction

Pattern recognition deals with the problem of identifying the underlying structure in the data or in other words, it is "a search for the structure in the data" [1]. Pattern recognition has gained significant importance due to number of reasons such as the need to add intelligence to machines so as to automate work with flavor of human skills, heavy demand of data processing, searching and the need of easy interfaces to communicate with computing machines. In 1965, Fuzzy set theory proposed by Zadeh [2] revolutionized the field of pattern recognition due to its capability to handle the imprecision. Interpretation of the data structure using fuzzy techniques is an efficient and natural choice where precise data partitions are not known. Since fuzzy logic and neural network are complimentary rather than competitive [3], lots of neuro-fuzzy approaches have been developed. The evolution of neuro-fuzzy approach and its need for pattern recognition is adequately elaborated in [4]. A novel approach called as "Fuzzy Min-Max Neural Network" (FMNN) using aggregation of hyperbox fuzzy sets was proposed by Simpson [5], [6]. A hyperbox [7] is a simple geometrical structure, shown in Figures 1(a),(b). An n-dimensional hyperbox can be formed by just defining its min point and max point.

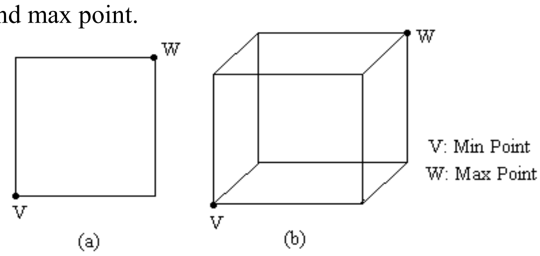

Fig.1 (a) Two Dimensional (b) Three Dimensional Hyperbox

FMNN algorithm is an example of merge of fuzzy logic and ART1 neural network [8] for pattern clustering, similar to the method proposed by Carpenter et al in [9]. This is a powerful method to classify linearly non-separable multidimensional data (e.g. Figure 2), in a single pass through. A recursive algorithm called as "General Fuzzy Min-Max Neural Network" (GFMN) [10] proposed by Gabrys and Bargiela is a fusion of FMNN classification [5] and FMNN clustering [6] algorithms. The main advantage of GFMN is that the hybrid approach enabled handling of labeled and unlabeled data simultaneously (i.e. partial supervision). Learning with partial supervision is an important issue in the practical implementation of pattern recognition systems. Sometimes it is not feasible to get a fully labeled dataset for training, or the cost of labeling all the data is not affordable. In such situations efficient learning with partial supervision is essential. The use of additional unlabelled data has been shown to offer improvements in comparison to the classifiers generated only on the basis of limited labeled dataset [11]. The approach of partial supervision or semi-supervised learning is found to be very useful in many practical applications [12]-[14].

The present work proposes "A General Reflex Fuzzy Min-max Neural network" (GRFMN) which deals with supervised, unsupervised and semi-supervised pattern

Manuscript received April 30, 2006.

A.V.NAndedkar is with E&ECE Dept, Indian Institute of Technology, Kharagpur, India.- 721302 as a Research scholar. Phone: +91-3222-281491, fax: +91-3222-255303; (e-mail: avn@ece.iitkgp.ernet.in).

P.K.Biswas, Professor, E&ECE Dept., Indian Institute of Technology, Kharagpur, India.- 721302 (e-mail: pkb@ece.iitkgp.ernet.in).

recognition problems.

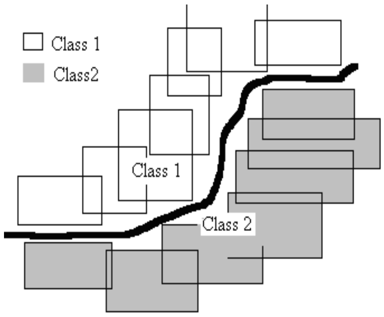

Fig.2 A Two dimensional example of classification of linearly non-separable data using Hyperboxes.

FMNN and GFMN learn the data by creation and expansion of hyperboxes. It contracts the hyperboxes belonging to different classes, facing the problem of overlap (Figure 3(a)) or containment (Figure 4(a), 5(a)). The contraction process avoids membership ambiguity in the overlapped region. The error analysis for FMNN and GFMN algorithms on labeled data show that contraction process causes classification errors in the learning phase [15][16]. In hybrid learning mode (partial supervision), knowledge acquired from labeled samples is applied to label the unlabeled data. Thus, a better approach to deal with labeled data is needed to improve the performance in hybrid mode. Moreover, performance of FMNN and GFMN are highly dependant on the choice of expansion coefficient i.e. maximum hyperbox expansion allowed. There are many approaches reported in the literature to optimize the performance of FMNN classifier such as [10],[18]-[21]. However, the optimization in these methods is achieved at the cost of recursive nature of the algorithms i.e. single pass through learning capability is lost.

The proposed GRFMN classifier is a merge of FMNN classification [5], clustering [6] and concept of reflex mechanism [16][17]. It handles labeled, unlabeled and partly labeled dataset more efficiently. GRFMN avoids the use of contraction process for the labeled data and hence errors due to it. It uses a reflex mechanism to overcome the hyperbox overlap and containment problems. This mechanism is inspired from the reflex system of human brain which takes over the control in hazardous conditions unconsciously [22]. The reflex mechanism consists of compensatory neurons which are helpful to approximate the complex topology of the data in a better way. Compensatory neurons are added dynamically to the reflex section of the network during training. These neurons maintain the hyperbox dimensions and control the membership in the overlapped region. The paper also proposes a modified learning algorithm for *labeling* the *unlabeled* data during training. A comparison of FMNN, GFMN and GRFMN in different modes is shown in section II. The main advantage of GRFMN over GFMN is its better capability to identify underlying structure in the data. Moreover, the training of GRFMN is on-line, single pass through compared to the recursive training of GFMN. Results show a significant improvement with a reduced classification error for GRFMN.

Rest of the paper is organized as follows. Section II describes error analysis of FMNN and GFMN learning algorithms and modifications proposed in GRFMN learning algorithm. The architecture details of GRFMN are given in section III. Section IV elaborates the learning mechanism of GRFMN. GRFMN is tested on several real datasets and results are compared with GFMN. The experimental details are given in section V. The final section concludes with summery.

II. Analysis of FMNN, GFMN Learning Algorithms

The analysis of FMNN and GFMN learning is done for the three learning modes i.e. i) classification, ii) hybrid and iii) clustering mode. FMNN and GFMN learning algorithms constitute three steps,

1) Hyperbox Expansion,
2) Overlap Test and
3) Hyperbox Contraction.

When a training sample is presented to the network, it tries to accommodate the sample in one of the existing hyperboxes of that class, provided the hyperbox size is not exceeding a specified maximum limit. Otherwise, a new hyperbox is added to the network. To avoid an ambiguity in the classes, overlap test is carried out. If overlap exists then it is removed by contraction process.

A. Classification Mode

Here we discuss the problems faced by FMNN and GFMN while dealing with labeled training data. Figure 3 depicts one such situation. When the data shown in Figure 3(a) is used for training FMNN or GFMN, two hyperboxes are created with an overlap. To remove overlap, the hyperboxes are contracted. Figure 3(b) shows the result after the contraction process. Note that the data points B and C are contained in the hyperboxes of class1 and class2 respectively, after contraction. Thus, point C gets full membership of class 2 and partial membership of class1 i.e. $\{\mu_2(C)=1\}$ and $\{0 \le \mu_1(C)<1\}$. However, point C actually belongs to class1 and its membership grade should be $\{\mu_1(C)=1\}$ and $\{0 \le \mu_2(C)<1\}$. Instead, it is classified with less membership grade in class 1. Similar is the case for the data point B, instead of membership grades $\{\mu_2(B)=1\}$ and $\{0 \le \mu_1(B)<1\}$, contraction process causes membership to be $\mu_1(B)=1$ and $\{0 \le \mu_2(B)<1\}$ and leads to training error. In general, information about the non-ambiguous area (such as gray area in Figure 3(b)) is also tampered by this contraction process.

This is handled more accurately in proposed GRFMN by adding overlap compensation neurons (OCNs) to the network. OCNs are added to the network whenever such overlap occurs during training.

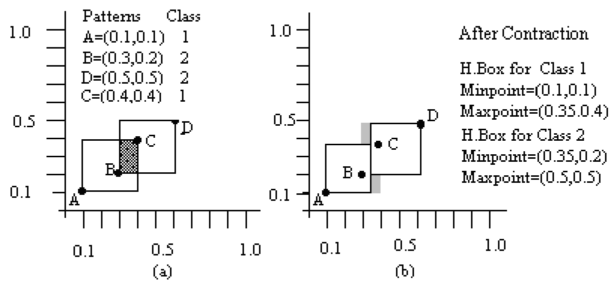

Fig. 3. Overlap removing process in FMNN and GFMN.

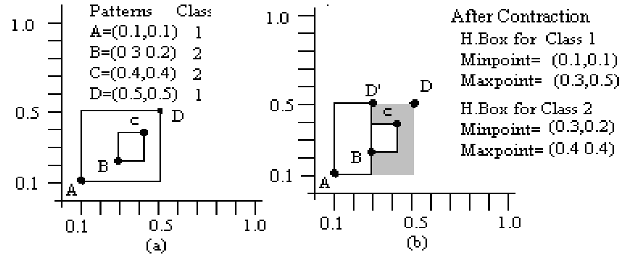

Fig.4 Full Containment Problem.

Figure 4(a) depicts a problem of hyperbox containment. Here a hyperbox of class 2 is completely contained by the hyperbox of class 1. FMNN and GFMN solve this problem by hyperbox contraction as shown in the Figure 4(b). It is clear that after contraction the membership of point D for class1 becomes $\{0 \le \mu_1(D) < 1\}$ even though actual membership is $\{\mu_1(D) = 1\}$.

Therefore, this causes a gradation error due to contraction (here, gradation error for a sample means the difference between desired and actual grade for certain class). If μ_2(D)>μ_1(D) (after contraction) then it causes a classification error as well. Moreover, information about the non-ambiguous area (gray area in Figure 4(b)) is tampered as well.

Figure 5 depicts a partial containment problem and the way it is eliminated by FMNN and GFMN. The max point of a hyperbox partially containing other class hyperbox is changed from D to D' and will not get a full membership in its original class. Moreover, membership grade of the point B remains ambiguous and learned non-ambiguous information (gray area Figure 5(b)) is also tampered. The problems shown in Figures 4(b), 5(b) are solved in GRFMN by adding containment compensation neuron (CCN) to the network.

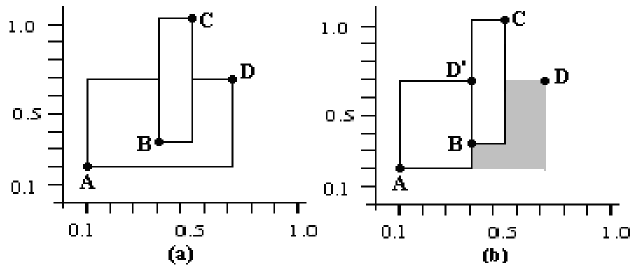

Fig. 5 Partial Containment Problem.

B. Hybrid Mode

This mode is used if training dataset consists of mixture of labeled and unlabeled samples. This mode does not exist in FMNN algorithm. In hybrid mode of learning, GFMN tries to label the unlabeled data with the help of acquired knowledge from the labeled training data. The above error analysis is equally valid for the labeled training samples in hybrid mode. Thus, contraction process in GFMN introduces training errors for the labeled data, which in turn hampers its performance in hybrid mode. We introduce the use of compensatory neurons in GRFMN to handle labeled data efficiently so as to improve the performance in hybrid mode.

GFMN learning algorithm assigns a label to an unlabeled hyperbox when a single labeled sample falls in it, but on the other hand the overlap between labeled and unlabeled hyperboxes is not allowed. Rather, it removes the overlap of unlabeled hyperboxes with all other hyperboxes. Here we introduce a change in GRFMN and allows the unlabeled hyperbox to overlap with a labeled hyperboxes. Moreover, we assign its class label to the unlabeled hyperbox. GRFMN undergoes a hyperbox contraction process if there is an overlap between two unlabeled hyperboxes.

C. Clustering Mode

In clustering mode, all training samples are unlabeled. There exists a difference in the contraction process of FMNN clustering algorithm [6] and GFMN algorithms. FMNN contracts the hyperboxes along all the dimensions if two hyperboxes are found to be overlapping. Whereas, GFMN contracts overlapped hyperboxes along a dimension with minimal overlap. Gabrys and Bargiela have reported in [10] that using this approach GFMN works better than FMNN with creation of less number of hyperboxes. Hence, GRFMN uses same contraction approach for the overlap amongst unlabeled hyperboxes.

There is a change in the hyperbox expansion policy in GRFMN compared to GFMN. GRFMN allows expansion of a hyperbox to accommodate a training point, if its membership to the hyperbox is greater than zero and average expansion is less than a user specified maximum limit. Whereas, GFMN expands the hyperbox if expansion is not crossing a user specified limit along any dimension of the hyperbox. Experimental results show that the approach used in GRFMN gives more correct results with less number of hyperboxes. Note that compensatory neurons are not created for GRFMN in pure clustering mode.

III. A General Reflex Fuzzy Min-Max Neural Network

General Reflex Fuzzy Min-Max Neural Network (GRFMN) is a hyperbox fuzzy set based algorithm for supervised, unsupervised and semi-supervised learning. The learning is of incremental type i.e. data samples are applied in a sequence. The network tries to learn the applied labeled samples by accommodating them in one of the existing hyperboxes of same class or label. This process is called as hyperbox expansion. If hyperbox expansion is not possible, a new hyperbox is added to the network. The aggregation of such hyperboxes (subsets) belonging to a class forms a complete class.

When an unlabeled sample is applied for the training, network tries to accommodate it in one of the existing labeled or unlabeled hyperboxes otherwise creates a hyperbox without label. An attempt is always made during training to label the unlabeled hyperboxes or samples, using the acquired knowledge. The details of learning algorithm are discussed section IV.

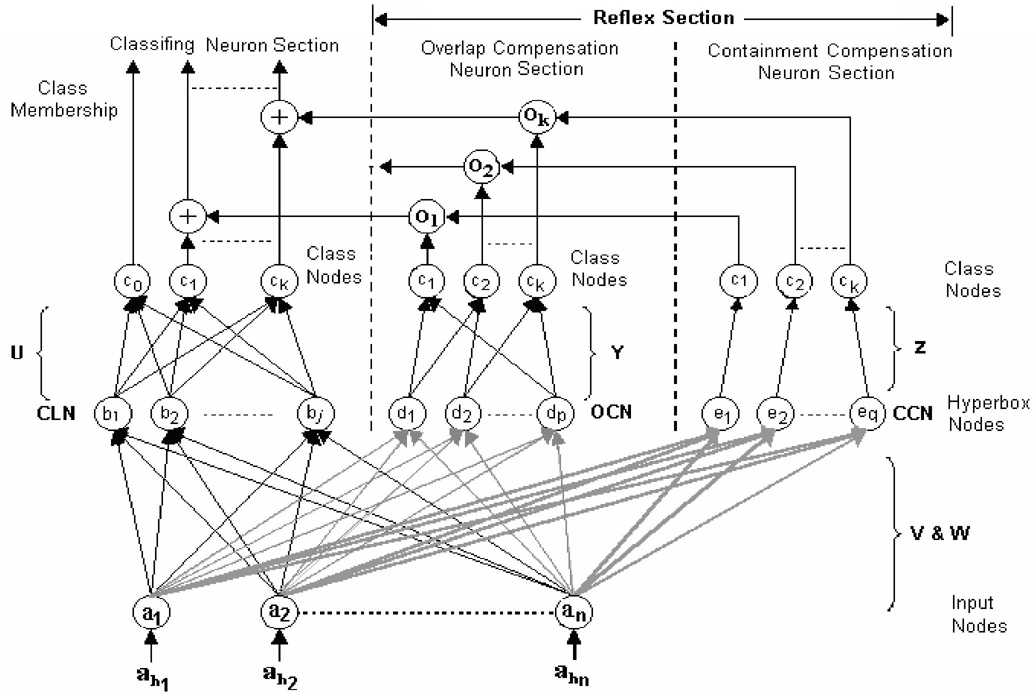

Fig. 6 GRFMN Architecture.

where a_{h1}-a_{hn}: Input data sample.
a_1-a_n : Input nodes.
b_1-b_j : Classification hyperbox nodes.
C_0-C_k : Class nodes.
d_1-d_p : Overlap compensation hyperbox nodes.
e_1-e_q : Containment compensation hyperbox nodes.
o_0-o_k : Overall compensation nodes.

During the training process hyperbox overlaps or containments (Figures 3(a), 4(a), 5(a)) do occur and this creates an ambiguity to decide the class for a test sample belonging to overlap region. FMNN [5] and GMNN [10] use a hyperbox contraction process to solve this problem. As discussed in section II this contraction causes the training errors for FMNN and GFMN algorithms. Here we deal with this problem using a novel concept of reflex mechanism which consists of overlap compensation and containment compensation neurons [16], [17]. As a result, the underlying data structure is captured more correctly.

A. GRFMN Architecture Overview

The proposed architecture of GRFMN is shown in Figure 6. Three different types of neurons are used in the architecture of GRFMN. The classifying neurons (CLNs) are the backbone of GRFMN. It helps to classify the data into number of classes learned. The reflex section consists of overlap compensation neurons (OCNs) and containment compensation neurons (CCNs). These are membership compensation neurons, used to decide the membership in the overlap zone of classifying neurons. The topology of network grows to meet the need of the problem during the training phase. All the input features are scaled between [0-1] for simplicity. The input layer receives a vector representing an input sample $\boldsymbol{a_h} \in I^n$, where I^n is an n-dimensional unit cubic space. The number of nodes in the input layer is equal to the dimension of applied data sample. The middle layer neurons and output layer nodes are partitioned into two sections- Classifying Neuron (CLN) section, and Reflex section. The reflex section is subdivided into Overlap Compensation Neuron (OCN) section and Containment Compensation Neuron (CCN) section.

A neuron in the middle layer represents an n-dimensional hyperbox fuzzy set. A connection between an input and a hyperbox node in the middle layer represent min and max points (***V*** & ***W***) respectively. The transfer functions of neurons in the middle layer are elaborated in section (III.2). Hyperbox nodes in the reflex section represent overlaps and containments occurring in CLN section. All middle layer neurons are created dynamically during the training process.

A hyperbox node in the CLN section is created if training sample belongs to a class which has not been encountered so far or existing hyperboxes belonging to that class can not be expanded further to accommodate it. The connections between hyperbox and class nodes in CLN section are represented by matrix ***U***. A connection between a hyperbox node b_j to a class node C_i is given by,

$$u_{ij} = \begin{cases} 1 & \text{if} \quad \{b_j \in C_i\} \\ 0 & \text{if} \quad \{b_j \notin C_i\} \end{cases} \qquad (1)$$

Note that all *unlabeled* hyperboxes created during training are connected to node C_0. Here onwards terminology "class nodes" is specifically used for labeled hyperboxes.

A hyperbox node in the middle layer of reflex section is created whenever the network faces the problem of hyperbox set overlap or containment. OCN section takes care of the overlap problem illustrated in Figures 3(a),8(b). The connections between hyperbox nodes and class nodes in the

OCN section are represented by matrix **Y**. The connection weights from a neuron d_p (representing overlap between i^{th} and j^{th} class hyperboxes) to i^{th} and j^{th} class nodes in OCN sub-section are given by,

$$y_{ip} \text{ and } y_{jp} = \begin{cases} 1 & \text{if } \{d_p \in C_i \cap C_j, i \neq j \text{ and } i,j>0\} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Similarly, a hyperbox node in CCN section is created whenever a hyperbox of one class is contained fully or partially within a hyperbox of another class, as depicted in Figures 4(a), 5(a), 10(b). The connections between the hyperbox nodes and class nodes in the CCN section are represented by **Z** matrix. The connection weights from a neuron e_q to a class node C_i in CCN section are given by,

$$z_{iq} = \begin{cases} 1 & \text{if } C_j \text{ is contained fully or partially by } C_i, i \neq j \text{ and } i,j>0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Note that the overlap or containment between a labeled hyperbox $(B_j \in C_i, \forall\, i>0)$ and unlabeled hyperbox $(B_k \in C_i, \forall i=0)$ does not create any OCN or CCN in reflex section as this overlap is allowed and is used to label the unlabeled hyperboxes.

The output layer consists of (k+1) nodes, if k classes are learned along with unlabeled data. One extra node (C_0 in Figure 6) represents the unlabeled hyperboxes created during training. The number of output layer nodes in CLN section is same as the number of classes learned. The number of class nodes in CCN, OCN sections depends on the nature of overlap the network encountered during the training process. The final membership for i^{th} class node is given by,

$$\mu_i = C_i + O_i \tag{4}$$

where C_i is the membership of i^{th} class in CLN section given by,

$$C_i = \max_{m=1..j} (b_m u_{mi}) \tag{5}$$

O_i is the compensation for the i^{th} class given by,

$$O_i = min\Big(\min_{j=1,..p}(d_j y_{ji}), \min_{j=1,..q}(e_j z_{ji})\Big) \tag{6}$$

Equation 4 is designed to handle the multiple class overlaps. Equation 5 finds a class membership by finding the maximum value amongst all the membership grades offered by different hyperboxes belonging to that class and Equation 6 finds compensation for a class. Thus, μ_i gives a maximum grade to a class from the available grades considering its compensation.

B. Classifying Neurons and Reflex Mechanism

The details of classifying neurons and reflex mechanism are as follows:

1) Classifying Neuron (CLN)

A neuron b_j in CLN section represents a hyperbox fuzzy set B_j, as defined in [5] ,

$$B_j = \{A_h, V_j, W_j, f(A_h, V_j, W_j)\} \;\forall\, A_h \in I^n \tag{7}$$

where $V_j = (v_{j1}, v_{j2}, ...v_{jn})$, $W_j = (w_{j1}, w_{j2}, ...w_{jn})$ are min and max points of the hyperbox.

Gabrys and Bargiela [10], have discussed the disadvantages of activation function used in FMNN [5],[6] and proposed a new activation function for a neuron b_j. Hence, the activation function used for the neuron b_j is given by [10],

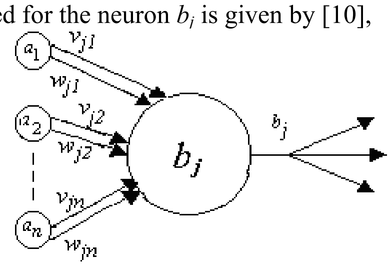

Fig. 7 Classifying Neuron.

$$b_j(a_h, V_j, W_j) = \min_{i=1..n} \left(min \begin{bmatrix} (1 - f(a_{hi} - W_{ji}, \gamma)), \\ (1 - f(V_{ji} - a_{hi}, \gamma)) \end{bmatrix} \right) \tag{8}$$

where V, W are min-max point of hyperbox B_j, $f(x,\gamma)$ is a two parameter ramp threshold function

$$f(x,\gamma) = \begin{cases} 1 & \text{if } x\gamma > 1 \\ x\gamma & \text{if } 0 \le x\gamma \le 1 \\ 0 & \text{if } x\gamma < 0 \end{cases} \tag{9}$$

γ : Fuzziness control parameter.

This activation function assigns membership value one if the test sample falls within the hyperbox. If it is outside hyperbox, membership value is calculated based on its distance from the min-max points. Figure 7 shows detailed diagram of a hyperbox node.

2) Overlap Compensation Neuron (OCN)

Figure 8(a) shows an overlap compensating neuron. Figure 8(b) depicts a general situation where two hyperboxes belonging to different classes are overlapping each other. OCN represents a hyperbox of size equal to the overlapped region between two hyperboxes belonging to different classes. OCN produces two compensation outputs, one each for the two overlapping classes. The OCN is active only when the test data is within the overlap region. The activation function is given by Equation 10. The activation function of this neuron protects the class of the min-max points of overlapping hyperboxes and offers a graded membership within the overlapped region.

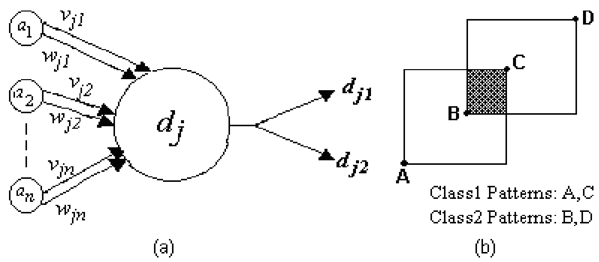

Fig. 8 Overlap Compensation Neuron

The activation function is as follows:

$$d_{jp} = T\big(b_j(a_h, v_j, w_j) - 1\big) \times \left(-1 + \frac{1}{n}\sum_{i=1}^{n} max\left(\frac{a_{hi}}{w_{p_{ji}}}, \frac{v_{p_{ji}}}{a_{hi}} \right) \right) \tag{10}$$

where d_{jp} : p=1 and 2, outputs for class 1 and class 2 respectively.

$\boldsymbol{a_h}$: input data.

$\boldsymbol{V_j, W_j}$:are min-max point of OCN.

V_1, W_1, V_2, W_2 : are min-max point of the overlapping hyperboxes.

T(x) : A Threshold function

$$T(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (11)$$

The first bracket consisting of $T(x)$, avoids neuron to be active if the test data is outside the overlapped region. This is possible because

$$T(b_j(x)-1) = \begin{cases} 1 & \text{if } b_j = 1, \text{ i.e. } a_h \text{ Contained within } d_p \\ 0 & \text{Otherwise} \end{cases} \quad (12)$$

The second bracket calculates the class membership with reference to the min-max points of overlapped hyperboxes. As a result, OCN takes over the control from CLN section and assign membership to the test sample if it falls in the overlapped region. This is very similar to the reflex action in human brain which takes over the control unconsciously in hazardous conditions. If the test sample $\boldsymbol{a}_h$ belongs to the overlapped region i.e. $\{ a_h \in C_i \cap C_j \; i \neq j \}$, the classifier section will allocate membership grade '1' for both the classes. In such a case, the activation function of compensatory neuron compensates the dispute and adds the respective membership grade.

Figure 9 shows the membership plot after adding the compensation output to the class 1 for the overlap region as depicted in Figure 3(a), 8(b). Note that the membership grade for class1 gradually decreases from point C to B and membership grade for class 2 is exactly opposite. It protects the class membership of min and max points of the overlapping hyperboxes, thus protecting the learned samples accommodated as min point and max points while expanding the hyperboxes. As no contraction is done, non-ambiguous region (as shown in Figure 3(a)) are also protected by OCN.

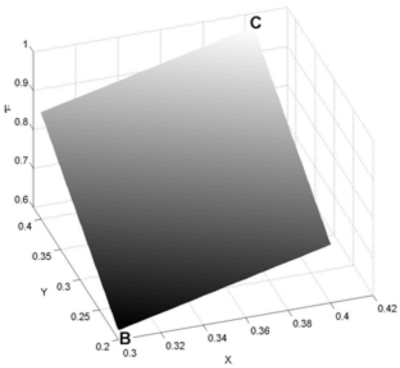

Fig. 9 Membership for class 1, after adding the compensating output O_1.

3) Containment Compensation Neuron (CCN)

The containment compensating neuron is shown in Figure 10. CCN is trained to handle the partial and full containment problems. CCN represents a hyperbox of size equal to the overlapping region between two classes as shown in the Figure 10(b). This neuron is also active only when the test sample falls inside the overlapped region. The output of this neuron is connected to the class that contains the hyperbox of other class.

Referring to the Figure 10(b), for the point P, both neurons representing the hyperboxes for the class1 and class 2 will

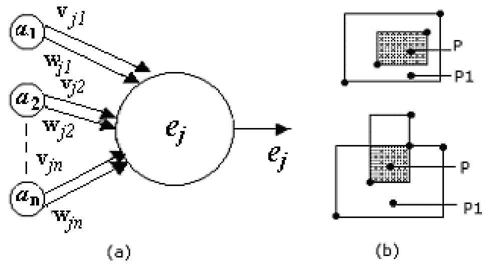

Fig.10 (a) Containment Compensation Neuron (b) The hyperbox represented by CCN (Shaded area)

produce membership grade '1'. Actually, the class 2, which is surrounded, by the class 1 contains the point P. Therefore, this compensating neuron subtracts 1 from the grade allocated to class 1 and hence compensates the output. For the point P1 grade given by the neuron representing hyperbox of class1 is '1'. The neuron representing the hyperbox of class 2 gives the membership grade depending on the distance of the point P1 from its min-max points. The output of compensating neuron is zero as P1 falls outside the overlapping region. The activation function of CCN is given by,

$$e_j = -1 \times T(b_j(\boldsymbol{a}_h, V, W) - 1) \quad (13)$$

where e_j : output of OCN.
$\boldsymbol{V}, \boldsymbol{W}$: min-max point of CCN,
$\boldsymbol{a}_h$: input data
T(x) : threshold function equation (12).

Note that as no hyperbox contraction is done, non-ambiguous regions as depicted in Figures 4(b), 5(b) are also protected.

IV. LEARNING AND RECALL ALGORITHMS

GRFMN learning algorithm creates and expands hyperboxes depending on the need of the problem. It tries to label the unlabeled data using currently learned structure. If there is any overlap, containment created (between hyperboxes of different classes) during expansion of labeled hyperboxes; respective compensatory neuron is added to the network. Contraction procedure [5] is executed if there is any overlap between unlabeled hyperboxes. Following learning algorithm can train GRFMN in three different modes i.e. classification, clustering and Hybrid modes (Partial Supervision).

A. Learning Algorithm

Learning algorithm consists of mainly two steps namely Data Adaptation and Overlap Test.

Assume that $\{\boldsymbol{a}_h, C_i\}$ is a training data sample, $\{b_j, C_j\}$ a hyperbox for class C_j and Θ_{max} is a maximum hyperbox size allowed during expansion. Initialize the network with b_1 having $\boldsymbol{V_1}=\boldsymbol{W_1}=\boldsymbol{a_h}$ and class label as C_i for an ordered pair of data $\{\boldsymbol{a}_h, C_i\}$. Unlabeled data or hyperboxes are represented by a label C_0 for simplicity.

Repeat the following steps 1 and 2 for the all-training samples.

STEP 1: Data Adaptation

Take a new training data point $\{a_h, C_i\}$

a) Find a hyperbox $\{b_j, C_j\}$ such that $C_j=C_i$ or $C_j= C_0$ offering largest membership and satisfying the conditions,

(1) $\Theta_{max} \geq \frac{1}{n}\sum_{i=1}^{n}(max(w_{ji}, a_{hi}) - min(v_{ji}, a_{hi}))$,   (14)

(2) b_j is not associated with any OCN or CCN, and

(3) If $C_i = C_0$ *or* $C_j = C_0$ then $\mu_j > 0$, where μ_j is membership with hyperbox b_j.

Adjust min-max points of hyperbox *bj* as,

$$V_{ji}^{new} = \min(V_{ji}^{old}, a_{hi})$$
$$W_{ji}^{new} = \max(W_{ji}^{old}, a_{hi}) \text{ where } i = 1,2\ldots n \quad (15)$$

and If $C_j = C_0$ *and* $C_i \neq C_0$ *then* $C_j = C_i$.

b) If no suitable b_j is found create a new hyperbox with $V_j = W_j = a_h$ and class C_i. **Go to step 1**

STEP 2: Overlap Test

Assume that hyperbox b_j, expanded in previous step is compared with all other hyperboxes b_k.

If C_j and $C_k = C_0$ then do the overlap and contraction test as explained in **Test 2** [5].

Else do the following **Test 1**.

Test 1:

a) Isolation Test:

If $(V_{ki} < W_{ki} < V_{ji} < W_{ji})$ *or* $(V_{ji} < W_{ji} < V_{ki} < W_{ki})$ is true for any *i*, $(i \in 1\ldots n)$ then (b_k, b_j) are isolated Go to Step1.

Else go for Containment test.

b) Containment Test:

If $(V_{ki} < V_{ji} < W_{ji} < W_{ki})$ *or* $(V_{ji} < V_{ki} < W_{ki} < W_{ji})$ is true for any *i*, $(i \in 1,2..\, n)$ then

Case1: if $C_j = C_0$ then assign $C_j = C_k$ go to step 1.

Case2: if $C_k = C_0$ then assign $C_k = C_j$ go to step 1.

Case3: if C_j and $C_k \neq C_0$ create a CCN with hyperbox min-max co-ordinates given by

$$V_{ci} = \max(V_{ki}, V_{ji}),\ W_{ci} = \min(W_{ki}, W_{ji}) \text{ for } i = 1,2\ldots n \quad (16)$$

Go to Step 1

Else hyperboxes are not facing containment problem go to step (c)

c*) Overlap compensation neuron creation:*

Case1: if $C_j = C_0$ then assign $C_j = C_k$ go to step 1.

Case2: if $C_k = C_0$ then assign $C_k = C_j$ go to step 1.

Case3: if C_j and $C_k \neq C_0$ create a OCN with hyperbox min-max co-ordinates given by

$$V_{oci} = \max(V_{ki}, V_{li}),\ W_{oci} = \min(W_{ki}, W_{li}) \text{ for } i = 1,2\ldots n \quad (17)$$

Go to Step 1

Test 2:

This follows the principle of minimal disturbance[5]. It finds the dimension with minimum overlap and contracts it. Let *d* represents the overlap along a dimension.

a) Overlap Test:

Assume that overlap is $d^{old} = 1$.

Check for the following cases for $i = 1, 2\ldots n$.

Case 1: $(v_{ji} < v_{ki} < w_{ji} < w_{ki})$

$d^{new} = min(w_{ji} - v_{ki}, d^{old})$

Case 2: $(v_{ki} < v_{ji} < w_{ki} < w_{ji})$

$d^{new} = min(w_{ki} - v_{ji}, d^{old})$

Case3: $(v_{ki} < v_{ji} \leq w_{ji} < w_{ki})$ *or* $(v_{ji} < v_{ki} \leq w_{ki} < w_{ji})$

$d^{new} = min(\, min(w_{ki} - v_{ji}, w_{ji} - v_{ki}),\, d^{old})$

If overlap exists (i.e. one of the above condition is true) and $(d^{new} - d^{old}) > 0$, then $\Delta = i$ else $\Delta = -1$.

If no overlap found then **Go to step 1.**

b) Contraction:

If overlap exists and is minimum along Δ dimension, then contract the hyperboxes using the following conditions:

Case 1: $(v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta})$

$$v_{k\Delta}^{new} = w_{j\Delta}^{new} = \frac{v_{k\Delta}^{old} + w_{j\Delta}^{old}}{2} \quad (18)$$

Case 2: $(v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta})$

$$v_{k\Delta}^{new} = w_{j\Delta}^{new} = \frac{v_{j\Delta}^{old} + w_{k\Delta}^{old}}{2} \quad (19)$$

Case 3: $(v_{k\Delta} < v_{j\Delta} \leq w_{j\Delta} < w_{k\Delta})$

$$\text{If } (w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta}) \text{ then } v_{j\Delta}^{new} = w_{k\Delta}^{old} \text{ else } w_{j\Delta}^{new} = v_{k\Delta}^{old} \quad (20)$$

Case 4: $(v_{j\Delta} < v_{k\Delta} \leq w_{k\Delta} < w_{j\Delta})$

$$\text{If } (w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta}) \text{ then } w_{j\Delta}^{new} = v_{k\Delta}^{old} \text{ else } v_{j\Delta}^{new} = w_{k\Delta}^{old} \quad (21)$$

Go to step 1.

B. Recall Procedure

Recall procedure is very simple. Class nodes in the each section calculate the class memberships and respective compensations. The summing node in the classifying neuron does the final grade calculation. The membership grade is computed according to equation 4.

V. RESULTS

GRFMN algorithm was tested on various datasets. GRFMN and GFMN algorithms can operate in three different modes that is classification, clustering and hybrid mode. The performance of GRFMN was compared with that of GFMN in all the three modes. Various experiments were carried out with different aims as follows.

1. To verify the performance improvement of GRFMN by addition of compensatory neurons.
2. To study the effect of variation of expansion coefficient (Θ) on the performance of GRFMN and compare it with FMNN and GFMN.
3. To compare performance of GRFMN with GFMN in hybrid and clustering mode of learning

Real dataset such as Iris, Wine, Ionosphere, Glass, Thyroid [23] and Breast Cancer [24] were used for the experiments.

A. Example to demonstrate the Overlap Compensation Neuron working

Referring to the problem discussed in Figure 2(a), after the contraction of hyperboxes, classification result for the training data is shown in the Table II. Table III shows the result for Crisp or Binary classification based on the maximum membership grade of a data point in a certain class. This problem is solved by GRFMN with creation of an OCN with min point as (0.3, 0.2) and max point (0.4, 0.4). The results shown in Tables II and III indicate that training data B and C are misclassified by both FMNN and GFMN due to hyperbox contraction. However, by introducing overlap compensation, original classes of B and C are retained.

TABLE I

Expansion Coefficient θ =0.4, sensitivity Parameter γ=4.

Sr	Data	Class (C)	Membership Grade					
			FMNN		GFMN		GRFMN	
			C_1	C_2	C_1	C_2	C_1	C_2
1	A (0.1,0.1)	C_1	1	0.65	1	0.0	1	0.2
2	B(0.3,0.2)	C_2	1	0.95	1	0.8	0.63	1
3	C(0.4,0.4)	C_1	0.95	1	0.8	1	1	0.8
4	D(0.5,0.5)	C_2	0.75	1	0.4	1	0.6	1

TABLE II

CRISP CLASSIFICATION OF THE TRAINING DATA.

Sr	Data	Class (C)	Crisp Classification					
			FMNN		GFMN		GRFMN	
			C_1	C_2	C_1	C_2	C_1	C_2
1	A (0.1,0.1)	C_1	1	0	1	0	1	0
2	B (0.3,0.2)	C_2	1	0	1	0	0	1
3	C (0.4,0.4)	C_1	0	1	0	1	1	0
4	D (0.5,0.5)	C_2	0	1	0	1	0	1

B. Classification Mode

In this mode of operation, all the training samples used are labeled. Results of GRFMN, GFMN and FMNN are compared with respect to the variations in expansion coefficient.

1) Complete data set for training and testing

We tested GRFMN algorithm on widely used datasets such as Iris, Wine, Ionosphere and Thyroid. Table III shows a comparison of results for GRFMN on Iris, Wine and Ionosphere datasets with the results of GFMN and FMNN reported in [10]. All the samples are used for training and testing. Expansion coefficient is varied from 0 to 1 in step of 0.01 and maximum and minimum errors are reported in Table III. Results on Thyroid dataset for GRFMN, GFMN and FMNN algorithms are also added in Table III. The minimum correct classification on the Iris data with both FMNN and GFMN is 92%, where as GRFMN gives minimum correct classification of 98%. Similar improvement is also observed on Wine, Ionosphere and Thyroid datasets.

2) Fifty percent Iris data for training and testing

Here GRFMN is trained with 50% randomly selected iris data and testing was carried out on remaining data. No training and testing error are observed for GRFMN with expansion coefficient 0.038. So appending the table of comparison in [5],[10] we can compare GRFMN performance with other conventional classifiers on the Iris data set as shown in Table IV. Note that GRFMN classifies all the data correctly.

3) Performance on Expansion coefficient variation

In this experiment, the effect of expansion coefficient on performance of GRFMN, FMNN and GFMN is studied. Sixty percent data sample from each class of Iris dataset are selected randomly for the training purpose and entire Iris dataset is used for Testing. The expansion coefficient was varied from 1.0 to 0.02 in steps of 0.02. The results obtained during training and testing are shown graphically in Figure 11, 12 respectively.

TABLE III

PERCENT RECOGNITION OF REAL DATA

Data	Iris		Wine		Ionosphere		Thyroid	
	Max	Min	Max	Max	Max	Min	Max	Min
FMNN	97.3	92	100	94.3	98.0	84.8	100	94.4
GFMN	100	92	100	88.6	98.7	90.1	100	94.4
GRFMN	100	98	100	100	100	96.0	100	98.1

TABLE IV

COMPARISON WITH OTHER TRADITIONAL CLASSIFIERS

Technique	Misclassifications
Bayes classifier [1]	2
K nearest neighborhood [1]	4
Fuzzy *K*-NN [2]	4
Fisher ratios [1]	3
Ho-Kashyap [1]	2
Perceptron [3]	3
Fuzzy Perceptron [3]	2
FMNN [1]	2
GFMN [1]	1/0
GFMN [3]	0
GRFMN [1]	0
GRFMN [3]	0

where

[1] Training set is of 75 data points (25 from each class) and Test set consists of remaining data points.

[2] Training data is of 36 data points (12 from each class) and test set consist of 36 data points results are then scaled up for 150 points.

[3.] Training and testing data were same.

In both the cases, performance of GRFMN is almost independent of the expansion coefficient and maximum error is less than 2% during training and less than 3.05% during testing. On the other hand, the errors for FMNN and GFMN are significantly higher.

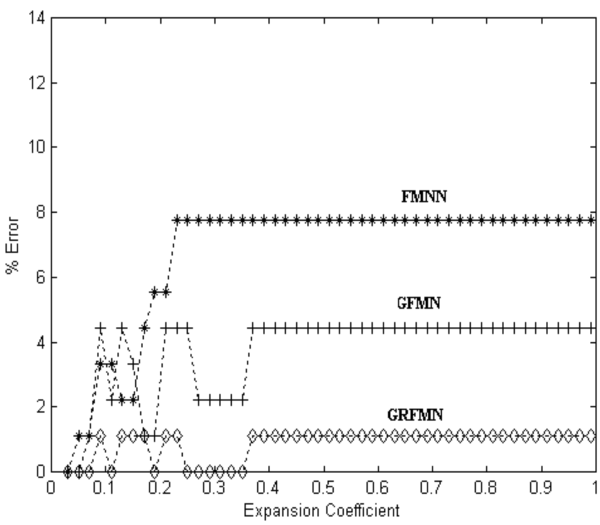

Fig. 11 Recognition of Iris Training Dataset

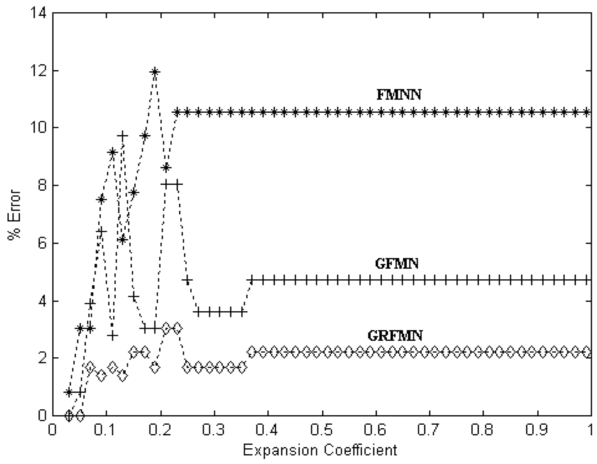

Fig. 12 Recognition for Iris Dataset

To study performance dependence on expansion coefficient more rigorously, we tested GRFMN, GFMN and FMNN algorithms on other real datasets such as Thyroid, Glass [23] and Breast Cancer [24]. Training and test datasets are prepared by randomly dividing each class of a dataset into two parts with 60-40% proportion. For Breast Cancer dataset, two equal parts were made for training and testing. Table V shows a performance comparison for these algorithms. Note that, GRFMN has the lowest minimum-learning (LE_{min}) and minimum-test errors (TE_{min}). Number of hyperboxes required to achieve this minimum learning and test error is also less compared to GFMN and FMNN. The number of hyperboxes for GRFMN is a sum of hyperboxes representing classification and compensatory neurons. Similarly, average learning and test errors (LE_{avg}, TE_{avg}) are also lowest for GRFMN when expansion coefficient is varied from 1 to 0.05 in step of 0.05. Moreover, the error deviation (δ) indicates a reduced dependency on expansion coefficient. This indicates that GRFMN is capable to extract the underlying structure in the data more correctly, which in turn helps to perform in a better way while dealing with mixture of labeled and unlabeled dataset. This can be observed from GRFMN results in the hybrid mode.

C. *Hybrid Mode*

To verify the performance improvement of GRFMN in hybrid mode, two experiments are conducted. The details are as follows.

1) *IRIS and Thyroid Datasets*

In this experiment, performance dependence on expansion coefficient is studied. Training data set is generated by randomly selecting 60% samples from each class of Iris and Thyroid datasets and 50% of the selected data was unlabeled for each class. Complete Iris and Thyroid datasets are used for testing so that we can evaluate the performance in hybrid mode. From Figures 13, 14 it is clear that GRFMN performance is better on training and test data even though Θ was varied from maximum to minimum value. Due to the inclusion of CNs and the new learning approach, the unlabeled data is handled more efficiently and is labeled more correctly. Similar results were obtained on Thyroid dataset (Figures 15, 16).

TABLE V
Effect of Expansion Coefficient
(Θ= 0.05 TO 1.0 IN STEP OF 0.05, γ =3)

Dataset	Method	LE_{min}	TE_{min}	HB	LE_{avg}	δ_{LE}	TE_{avg}	δ_{TE}
IRIS	GRFMN	0.00	0.00	42	1.94	0.7	4.58	1.06
	GFMN	0.00	1.33	45	6.83	3.12	8.42	2.39
	FMNN	0.00	5.00	42	8.33	3.52	10.33	2.79
Thyroid	GRFMN	0.00	2.32	29	0.03	0.17	2.96	1.27
	GFMN	0.00	3.49	34	3.06	2.03	4.76	0.64
	FMNN	0.00	3.49	88	3.57	1.95	5.17	1.33
Glass	GRFMN	0.00	7.77	15	0.00	0.00	8.89	0.72
	GFMN	0.00	7.77	23	0.00	0.00	9.78	1.27
	FMNN	0.00	8.89	29	0.00	0.00	10.94	0.83
Breast Cancer	GRFMN	0.00	2.05	45	0.27	0.26	4.92	3.65
	GFMN	0.00	2.64	65	1.53	1.80	7.89	4.66
	FMNN	0.00	3.22	97	2.16	2.58	6.61	3.05

HB: Number of hyperboxes, LE : Learning Error, TE : Testing Error, δ : Standard deviation.

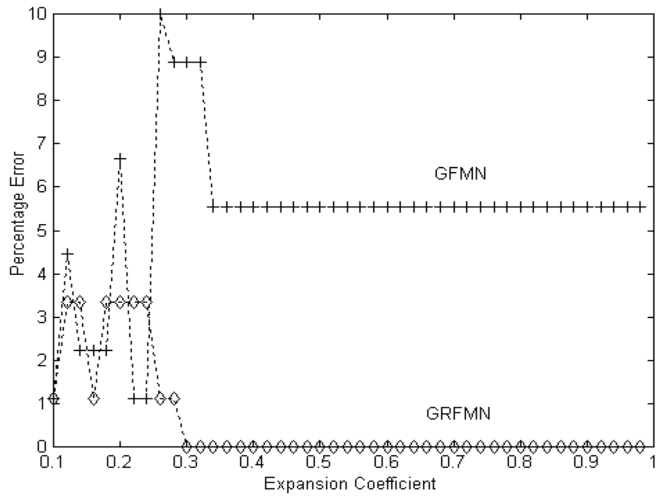

Fig 13 Error for Iris Training Data.

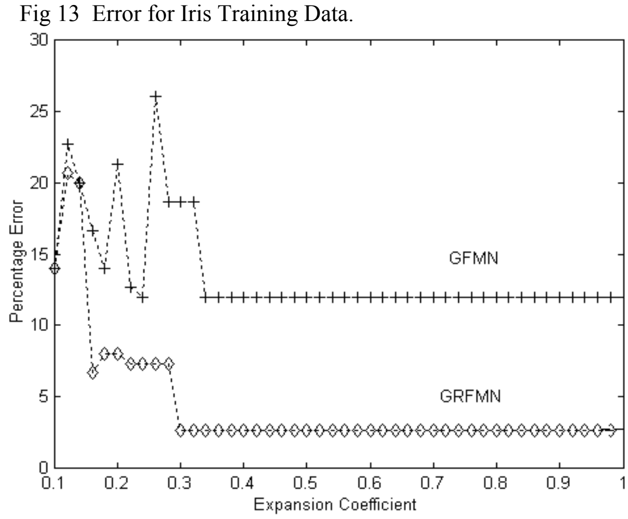

Fig. 14 Recognition for Iris dataset

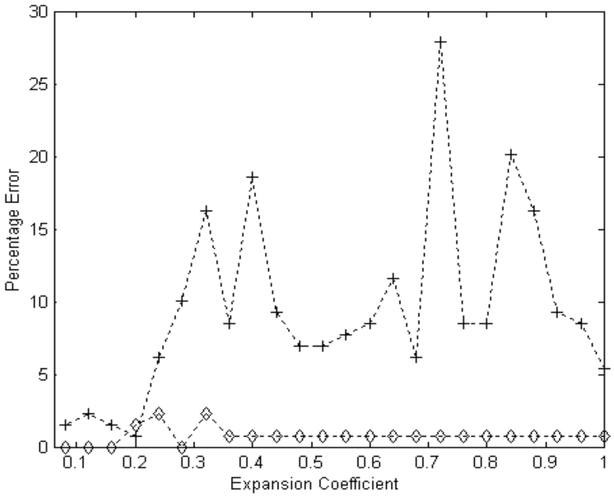

Fig. 15 Error on Thyroid training Data

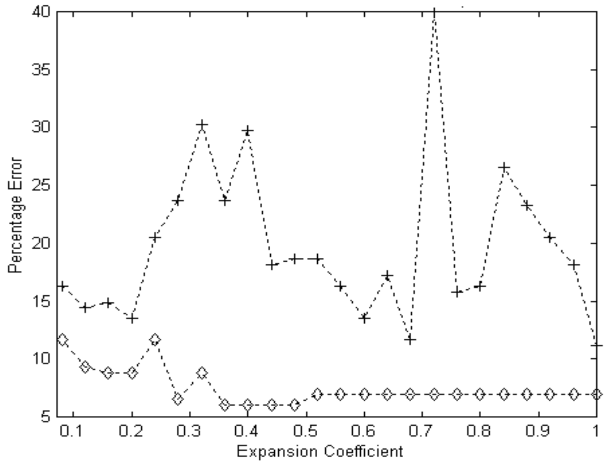

Fig. 16 Recognition for Thyroid Dataset

2) Performance on training dataset with variation in unlabeled samples

In this experiment, we fixed the value of expansion coefficient to 0.1 and varied the percentage of unlabeled training samples. Experiment is repeated 10 times for each value of percentage unlabeled data. Average of results on Iris dataset are reported in Figure 17. Training dataset is prepared by selecting and un-labeling the samples randomly from each class. The proportion of label and unlabeled samples is same for the all classes. The proportion of unlabeled samples is varied from 30-90%. The results shown in Figure 17 indicate that GRFMN performance is significantly improved compared to GFMN. Performance of GFMN is drastically affected when the number of unlabeled samples is increasing. As GRFMN can extract the data structure more correctly, its performance is better even though the number labeled samples are less in the training data set.

D. Cluster Mode:

In this experiment fully unlabeled Iris and Glass datasets were used for training. GRFMN and GFMN were trained with different expansion coefficient values. After training, the

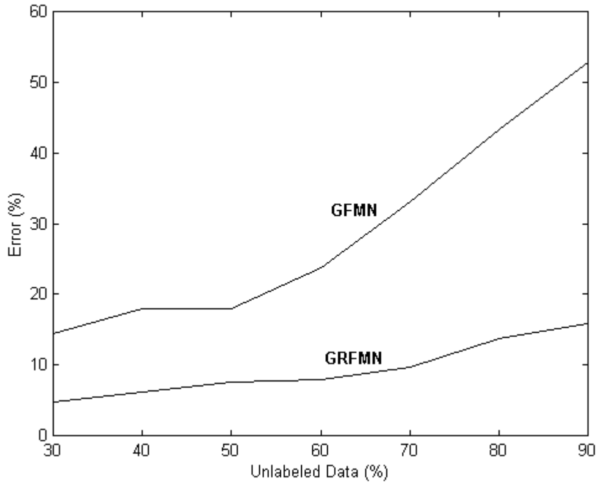

Fig. 17 Effect of variation in proportion of unlabeled data samples

created hyperboxes were labeled considering majority class samples contained by them. Results on Iris dataset for GFMN with Θ=0.12, 0.06, 0.03 are reported in [10]. We compared results for GRFMN keeping Θ same. A reading with $\Theta = 0.18$ is added in table VI. The results in Table VI show that GRFMN can cluster data with better performance and requires less hyperboxes compared to GFMN. A similar result on Glass dataset is reported in Table VII.

1) Iris Dataset

TABLE VI

Θ	GRFMN		GFMN	
	Error (%)	Hyper boxes	Error (%)	Hyper boxes
0.03	0	74	0	87
0.06	1.33	37	2.0	44
0.12	4.00	11	4.67	13
0.18	6.00	04	12.00	07

2) Glass Dataset

TABLE VII

Θ	GRFMN		GFMN	
	Error (%)	Hyper boxes	Error (%)	Hyper boxes
0.06	0.00	152	0.00	167
0.1	2.34	99	2.80	122
0.15	8.41	55	12.62	81

VI. Conclusion

A neural algorithm with reflex mechanism for classification and clustering called as "General Reflex Fuzzy Min-max Neural" has been presented here. Error analysis shows that contraction process in the learning algorithm causes classification errors for FMNN and GFMN. As far as possible GRFMN avoids the use of contraction process, and hence reduces errors caused due to it. The compensatory neurons can handle the overlaps and containments of the hyperboxes more efficiently. Inclusion of these neurons helps GRFMN to approximate complex topology of data in a better way. As performance of GRFMN on labeled data is better than GFMN, its performance in hybrid mode (where mixture of labeled and unlabeled data used for training) is also better. From the experiments, we conclude that GRFMN can avoid the dependency of the network performance on Θ to a larger extent hence single pass through learning is possible. Moreover, due to its new learning policy to label unlabeled data and hyperbox expansion in hybrid and clustering mode; its performance is significantly improved in all three modes of operations, compared to GFMN

REFERENCES

[1] Duda R.O, Hart P.E., Stock D.G., "Pattern Classification", 2nd ed., John Wiley & Sons, Singapore, 2002.

[2] L.A.Zadeh , "Fuzzy Sets" , Information and control, Vol. 8, 1965, 338-353.

[3] L.A Zadeh., "Fuzzy logic, neural network and soft computing", *Communication of ACM*, Vol.37, 1994, pp.77-84.

[4] J.C.Bezdek, S.K. Pal, "Fuzzy Models for Pattern Recognition", IEEE Press, Piscataway, N.Y., 1992.

[5] P.K.Simpson, "Fuzzy Min-Max Neural Network – Part I: classification" ,*IEEE Tran. Neural Networks*, vol.3, no.5, 1992, pp.776-786.

[6] P.K.Simpson, " Fuzzy Min-Max Neural Network – Part II: Clustering" , *IEEE Tran. Fuzzy System*, Feb. (1993), Vol.1, no.1, 32-45.

[7] B.Alpern , L.Carter "The hyperbox*", proceedings IEEE cnf. on Visualization*,(California), Oct 1991, pp. 133- 139.

[8] G.Carpenter, S. Grossberg, "A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine", *Computer Vision, Graphics & Image Understanding*, Vol. 37, 1987, pp. 54-115.

[9] G.Carpenter, S. Grossberg, D.B. Rosen, " Fuzzy ART: An Adaptive Resonance Algorithm for Rapid, Stable Classification of Analog Patterns", *Proc. Int. joint Conf. Neural Networks*, IJCNN-91(Seattle), Vol.2, 1991,pp. 411–416.

[10] B.Gabrys, A. Bargiela, "General Fuzzy Min-Max Neural Network for clustering and Classification" , IEEE Tran. Neural Network, May (2000). Vol.11, pp. 769-783.

[11] B.Gabrys, L.Petrakieva, "Combining Labeled and Unlabelled Data in The Design of Pattern Classification Systems", *Int. Journal of Approximate Reasoning*, Vol. 35, 2004, pp.251-273.

[12] A. M. Bensaid, L.O. Hall,J.C. Bezdek and L.P. Clarke, "Partially Supervised Clustering For Image Segmentation", *Pattern Recognition,* Vol. 29, No. 5, 1996, pp. 859 871.

[13] R.Ghani, "Combining Labeled and Unlabeled Data for Multi-Class Text Categorization", , *Proc. of 19th Int. Cnf. on Machine Learning*, ICML, Sydney, Australia, 2002, pp.187-194

[14] M. R. Amini, P. Gallinari, "The use of unlabeled data to improve supervised learning for text summarization*", Proc. of the 25th ACM SIGIR Cnf.* Tampere, Finland , 2002, pp.105-112.

[15] A. Bargiela, W. Pedrycz and M. Tanaka, "An exclusion/inclusion fuzzy hyperbox classifier", *Int. Journal of Knowledge-based and Intelligent Engineering Systems*, Vol. 8, No. 2, 2004, pp. 91-98.

[16] A.V. Nandedkar, P.K. Biswas, "A Fuzzy Min-Max Neural Network Classifier with Compensatory Neuron Architecture", *17th Int. Cnf. on Pattern Recognition* (ICPR2004) Cambridge UK, Vol. 4, 2004, 553-556.

[17] Nandedkar A.V., Biswas P.K., "A General Fuzzy Min Max Neural Network with Compensatory Neuron Architecture", *Proceedings of Int. Cnf. On Knowledge Based Knowledge Based Intelligent Engg. Systems, KES2005*, Australia, Vol.3, 2005, pp.1160-1167.

[18] C.Xi, J.Dongming, L. Zhijian : Recursive Training for Multi-resolution Fuzzy Min-Max Neural Network Classifier, *Proc. of 6th Int. Cnf. Solid-State and Integrated Circuit Technology*, (Shanghai), 2001, pp.131-134.

[19] A.Rizzi, M. Panella, F.M. FrattaleMascioli: Adaptive Resolution Min-Max Classifiers, I*EEE Trans. on Neural Networks*, Vol.13, 2002, pp. 402-414.

[20] S.Abe and M.S. Lan : A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern classification , *IEEE Trans. on Fuzzy Systems*, Vol 3, No.1, 1995, pp.18-28.

[21] F.M.Frattale Mascioli, G.Martinelli, "A Constructive Approach to Neuro-Fuzzy Networks", *Signal Processing*, Vol.64, No.3, 1998, pp.347-358.

[22] G.A.Baitsell : Human Biology , second edition, Mc-Graw Hill Book co. inc. NY, 1950.

[23] C.Blake,E.Keogh,C.J.Merz., UCI repository of machine learning databases. University of California, Irvine., 1998, Available : http://www.ics.uci.edu/~mlearn/MLRepository.html

[24] W.H.Wolberg, O.L.Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", *Proceedings of the National Academy of Sciences*, U.S.A., Vol. 87, 1990, pp. 9193-9196.

Abhijeet V. Nandedkar (M'06) received his B.E. (Electronics) and M.E.(Instrumentation) degrees from Shri Guru Govind Singhji Institute of Engineering and Technology, Nanded, Maharashtra, India in the years 1995 and 1997 respectively. He is working toward the Ph.D. degree at the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, India.

Since 1998 he has been working as a Faculty member in the Department of Electronics and Tele-Communication Engineering, Shri Guru Govind Singhji Institute of Engineering and Technology, Nanded, Maharashtra, India where he is presently Lecturer. His areas of interest are Pattern Recognition, Neural Networks, Fuzzy logic, Image Processing. He is a member of IAENG and Student member IEEE.

P. K. Biswas(M'06) completed his B. Tech. (Hons), M. Tech. and Ph. D. from the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, India in the years 1985, 1989 and 1991 respectively. From 1985 to 1987 he was with Bharat Electronics Ltd. Ghaziabad as a Deputy Engineer.

Since 1991 he has been working as a Faculty member in the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, where he is presently holding the position of Professor. Dr. Biswas visited University of Kaiserslautern, Germany under Alexander von Humboldt Research Fellowship during March 2002 to February 2003. Dr. Biswas has more than seventy research publications in International and National Journals and Conferences and has filed seven international patents. His areas of interest are Image Processing, Pattern Recognition, Computer Vision, Video Compression, Parallel and Distributed Processing and Computer Networks. He is a member of IEEE and IAENG.