FRBF: A Fuzzy Radial Basis Function Network

Sushmita Mitra and Jayanta Basak

Machine Intelligence Unit, Indian Statistical Institute, Calcutta, India

The FRBF network is designed by integrating the principles of a radial basis function network and the fuzzy c-means algorithm. The architecture of the network is suitably modified at the hidden layer to realise a novel neural implementation of the fuzzy clustering algorithm. Fuzzy set-theoretic concepts are incorporated at the input, output and hidden layers, enabling the model to handle both linguistic and numeric inputs, and providing a soft output decision. The effectiveness of the model is demonstrated on a speech recognition problem.

Keywords: Fuzzy clustering; Neuro-fuzzy computing; Pattern classification; Radial basis function network

1. Introduction

There have been several attempts [1–4] by researchers in making a funsion of the merits of fuzzy set theory [5] and Artificial Neural Networks (ANN) [6] under the heading of *neuro-fuzzy computing*, for improving performance in decision making systems. The integration promises to provide, to a great extent, more intelligent systems (in terms of parallelism, fault tolerance, adaptivity and uncertainty management) to handle real life recognition/decision making problems. We first provide a brief survey of some of the existing neuro-fuzzy models used for supervised/unsupervised classification.

Huntsberger and Ajjimarangsee [1] modified Kohonen's network for generating the fuzzy self-organising feature map. Fuzziness was also incorporated into the learning process by replacing the learning rate with fuzzy membership of the nodes in each class. Further modifications on the rate of learning, shrinking of the neighbourhood and termination conditions of the algorithm were reported by Bezdek et al. [7] in the FLVQ algorithm. A relationship between the fuzzy version of Kohonen's algorithm and the fuzzy c-means algorithm [8] was also established.

Carpenter et al. [2,9] developed a fuzzy version of ART by designing a neural network structure which realises a new min-max learning rule, that minimises predictive error and improves generalisation. The model performs online learning of inputs. A supervised neural network classifier that utilises min-max hyperboxes as fuzzy sets (which are aggregated into fuzzy set classes) was introduced by Simpson [3]. The network has a three layered architecture consisting of the input, hidden and output layers. Each hidden layer neuron represents a hyperbox fuzzy set having two types of connections from the input layer, representing the *min* and *max* points of the inputs. Learning is a single pass procedure. The model is capable of finding reasonable decision boundaries in overlapping classes, and for learning highly nonlinear relations.

The fuzzy multilayer perception (MLP) [10,11] incorporates fuzzy set-theoretic concepts at the input and output levels, and during learning. The input is modelled in terms of the linguistic properties *low, medium* and *high*. The output is represented as class membership values. The fuzzy MLP is found to be more efficient than the conventional MLP for classification and rule generation. Jang and Sun [12] have shown that fuzzy systems are functionally equivalent to a class of Radial Basis Function (RBF) networks, based on the similarity between the local receptive fields of the network and the membership

Correspondence and offprint requests to: S. Mitra, Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India. Email: sushmita@isical.ac.in

functions of the fuzzy system. Conditional fuzzy clustering has been used by Pedrycz [13] in the preprocessing phase of the design of an RBF.

In this paper, we propose a fuzzy version of the RBF network. The RBF is a three-layered network, typically used for supervised classification. The hidden layer performs crisp clustering using Gaussian basis function at the nodes. The output layer performs a linear combination of the weighted activations from the hidden layer.

The Fuzzy RBF (FRBF) is designed by integrating the principles of the RBF network and the fuzzy c-means algorithm. It incorporates fuzzy set-theoretic concepts at the input, output and hidden layers. The model can handle both linguistic and numeric inputs, and provides a soft decision in the case of overlapping pattern classes at the output. The use of fuzzy c-means in the hidden layer allows the network to provide a more accurate representation of real-life situations, where a pattern can have finite non-zero membership of two or more classes. To realise the neural implementation of the fuzzy clustering algorithm, the architecture of the network is suitably modified. The classification capability of the new model is demonstrated on a speech recognition problem.

2. Preliminary Concepts

In this section we describe the Radial Basis Function (RBF) network and the fuzzy c-means algorithm. These are the essential ingredients of the proposed fuzzy RBF, which is described in the following section.

2.1. RBF Network

A Radial Basis Function (RBF) network [14,15] consists of three layers. The connection weight vectors between the input-hidden and hidden-output layers are denoted as $\vec{v}$ and $\vec{W}$, respectively. The basis (or kernel) functions in the hidden layer produce a localised response to the input stimulus. The output nodes form a weighted linear combination of the basis functions computed by the hidden nodes.

The input and output nodes correspond to the input features and output classes, while the hidden nodes represent the number of clusters (specified by the user) that partition the input space. Let $\vec{X} = (X_1, \ldots, X_i, \ldots, X_n) \in R^n$ and $\vec{y} = (y_1, \ldots, y_i, \ldots, y_l) \in R^l$ be the input and output (respectively), and c be the number of hidden nodes.

The output h_j of the jth hidden nodes, using the *Gaussian kernel* function as a a basis, is given by

$$h_j = \exp\left[-\frac{(\vec{X} - \vec{v}_j)^T(\vec{X} - \vec{v}_j)}{2\sigma_j^2}\right], \quad j = 1, 2, \ldots, c \tag{1}$$

where $\vec{X}$ is the input pattern, $\vec{v}_j$ is its input weight vector (i.e. the centre of the Gaussian for node j), and σ_j^2 is the variance determining the sensitivity of the Gaussian to off-centre input, such that $0 \leq h_j \leq 1$.

The output y_j of the jth output node is

$$y_j = \vec{W}_j^T \vec{h}, \quad j = 1, 2, \ldots, l \tag{2}$$

where $\vec{W}_j$ is the weight vector for this node, and $\vec{h}$ is the vector of outputs from the hidden layer. The network performs a linear combination of the non-linear basis functions of Eq. (1).

The problem is to minimise the error

$$E = \frac{1}{2}\sum_{p=1}^{N}\sum_{j=1}^{l}(y_j^{(p)} - {}^*y_j^{(p)})^2 \tag{3}$$

where ${}^*y_j^{(p)}$ and $y_j^{(p)}$ are desired and computed output at the jth node for the pth pattern, N is the size of the data set, and l is the number of output nodes.

Learning in RBF networks can, in general, be performed by two different strategies [6]. A fixed set of cluster centres is first formed by a clustering algorithm (e.g. the c-means algorithm [16]). The associations of the cluster centres with the output are then learned by squared error minimisation (i.e. minimisation of E). Alternatively, the cluster centres can also be learned along with the weights from the hidden layer to the output layer using a gradient descent technique. However, learning the centres along with weights may lead to some locally fixed points in the error space, leading to a deviation from the desired result.

Here a fixed set of cluster centres is formed by the c-means algorithm [16]. Let the cluster centres, so determined, be denoted as $\vec{v}_j$, $j = 1, \ldots, m$. The parameter σ_j represents a measure of the spread of data associated with each node.

Learning in the output layer is performed after the parameters of the basis functions have been determined. The weights are typically trained using the Least Mean Squares algorithm, given by

$$\Delta\vec{W}_j^{(p)} = -\eta e_j^{(p)}\vec{h}^{(p)} \tag{4}$$

where $e_j^{(p)} = y_j^{(p)} - {}^*y_j^{(p)}$ and η is the learning rate. The SVD algorithm can also be used.

In an RBF network, the clustering of the input data is represented by crisp partitions, and the clusters are modelled by the Gaussian distribution. The degree of belongingness (i.e. membership) of an input to any cluster may not always follow the Gaussian structure. The learning algorithm of the conventional RBF network essentially adjusts the weights of the links from the hidden to the output layer, depending on the mean and variances of the Gaussian distribution in each hidden node. At this stage, however, any input vector can fire one or more hidden nodes, to some extent.

It may be more natural from the fuzzy set-theoretic point of view to determine the membership value of each data point to different clusters using the fuzzy c-means algorithm [8]. Here, instead of considering the Gaussian structure, the membership value of a point to different clusters is determined based on the relative closeness of the point to the different cluster centres. In the following section, we describe the fuzzy c-means algorithm.

2.2. Fuzzy C-Means

The Fuzzy C-Means (FCM) clustering algorithm [8] is a set-partitioning method based on Picard iteration through necessary conditions for optimising a weighted sum of squared errors objective function (J_m). Let $c \geq 2$ be an integer; let $X = (X_1, \ldots, X_N) \subset R^s$ be a finite data set containing at least $c < N$ distinct points; and let R^{cN} denote the set of all real $c \times N$ matrices. A non-degenerate fuzzy c-partition of X is conveniently represented by a matrix $U = [u_{ik}] \in R^{cN}$, the entries of which satisfy:

$$u_{ik} \in [0,1], \quad 1 \leq i \leq c, 1 \leq k \leq N \tag{5}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \quad 1 \leq k \leq N \tag{6}$$

$$\sum_{k=1}^{N} u_{ik} > 0, \quad 1 \leq i \leq c \tag{7}$$

The set of all matrices in R^{cN} satisfying Eqs (5)–(7) is denoted by M_{fcN}. A matrix $U \in M_{fcN}$ can be used to describe the cluster structure of X by interpreting u_{ik} as the grade of membership of X_k in the ith cluster: $u_{ik} = 0.95$ represents a strong association of X_k to cluster i, while $u_{ik} = 0.01$ represents a very weak one. Other useful information about cluster substructure can be conveyed by identifying prototypes (or cluster centres) $v = (v_1, \ldots, v_c)^T \in R^{cs}$, where v_i is the prototype for class i, $1 \leq i \leq c$, $v_i \in R^s$. 'Good' partitions U of X and representatives (v_i for class i) may be defined by considering minimisation of the c-means objective function J_m: $(M_{fcN} \times R^{cs}) \rightarrow R$, defined by

$$J_m(U, v) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m |X_k - v_i|^2 \tag{8}$$

where $1 \leq m < \infty$ is the fuzzifier and $|\cdot|$ is any inner product induced norm on R^s. For $m > 1$, Bezdek [8] gave the following necessary conditions for a minimiser (U^*, v^*) of $J_m(U,v)$ over $M_{fcN} \times R^{cs}$:

$$v_i^* = \frac{\sum_{k=1}^{N} (u_{ik}^*)^m X_k}{\sum_{k=1}^{N} (u_{ik}^*)^m} \tag{9}$$

and

$$u_{ik}^* = \left(\sum_{j=1}^{c} \left(\frac{d_{ik}^*}{d_{jk}^*} \right)^{2/(m-1)} \right)^{-1} \tag{10}$$

for all i, where $d_{ik}^* = |X_k - v_i^*|^2$.

3. Fuzzy Radial Basis Function (FRBF) Network

Here we describe how fuzzy concepts are incorporated at the input, output and hidden layers of the RBF network. The input space is partitioned using overlapping linguistic sets, thereby utilising more local information that aids in better classification. Linguistic as well as numeric inputs can be handled by using S and π functions. The output is modelled in terms of class membership values. The input-hidden layer weights are initialised by cluster centres using fuzzy c-means, instead of the more conventional hard c-means. The intermediate (hidden) layer is suitably modified to incorporate fuzzy c-means clustering during learning, such that each output node receives the weighted membership value (as opposed to a Gaussian function-based measure of proximity) of the enhanced input vector within each cluster. The resultant FRBF architecture is depicted in Fig. 1.

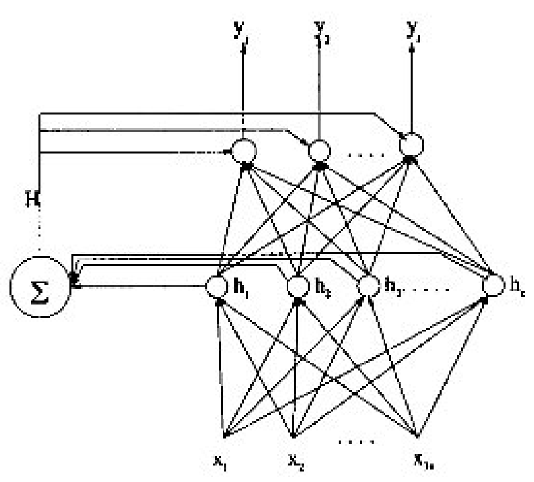

Fig. 1. Fuzzy RBF network.

3.1. Input Vector

Each input feature X_j can be expressed in terms of membership values to each of the three linguistic property sets *low* (L), *medium* (M) and *high* (H). Therefore, a n-dimensional pattern $\vec{X}_i = [X_{i1}, X_{i2}, \ldots, X_{in}]$ may be represented as a $3n$-dimensional vector [10].

$$[x_1, \ldots, x_{3n}] = [\mu_{low(X_{i1})}(\vec{X}_i), \mu_{medium(X_{i1})}(\vec{X}_i), \mu_{high(X_{i1})}(\vec{X}_i), \ldots, \mu_{high(X_{in})}(\vec{X}_i)] \tag{11}$$

The linguistic properties *low*, *medium* and *high* are modelled using $1 - S$, π and S functions [4], respectively. Note that we could have used more linguistic variables at the expense of increased computational complexity.

The S and π functions of Figs 2(a) and (b) are defined as

$$\begin{aligned} S(r; \alpha, \beta, c) &= 0 && \text{for } r \le \alpha \\ &= 2\left(\frac{r-\alpha}{c-\alpha^{a}}\right)^2 && \text{for } \alpha \le r \le \beta \\ &= 1 - 2\left(\frac{r-c}{c-\alpha}\right)^2 && \text{for } \beta \le r \le c \\ &= 1 && \text{for } r \ge c \end{aligned} \tag{12}$$

$$\begin{aligned} \pi(r; c, \lambda) &= S\left(r; c-\lambda, c-\frac{\lambda}{2}, c\right) && \text{for } r \le c \\ &= 1 - S\left(r; c, c+\frac{\lambda}{2}, c+\lambda\right) && \text{for } r \ge c \end{aligned} \tag{13}$$

In $S(r; \alpha, \beta, c)$, the parameter β, $\beta = (\alpha + c)/2$, is the *crossover point*, i.e. the value of r at which S takes the value 0.5. In $\pi(r; \lambda, c)$, λ is the *bandwidth* (diameter), i.e. the distance between the crossover points of π, while c is the point at which π is unity. For ease of representation, let us define the S function of Eq. (12) in terms of c and λ as $S(r; c, \lambda)$, where $\alpha = c - \lambda$. When X_j is numerical, we use the S and π-fuzzy sets of Eqs (12)–(13) with appropriate $\boldsymbol{c}$ and λ to calculate the μ values in Eq. (11).

When the input feature X_j is linguistic, its membership values for *low*, *medium* and *high* are quantified as

$$\begin{aligned} low &= \left\{\frac{0.95}{L}, \frac{\pi\left(X_j\left(\frac{0.95}{L}\right); c_m, \lambda_m\right)}{M}, \frac{S\left(X_j\left(\frac{0.95}{L}\right); c_h, \lambda_h\right)}{H}\right\} \\ medium &= \left\{\frac{1 - S\left(X_j\left(\frac{0.95}{M}\right); c_l, \lambda_l\right)}{L}, \frac{0.95}{M}, \frac{S\left(X_j\left(\frac{0.95}{M}\right); c_h, \lambda_h\right)}{H}\right\} \\ high &= \left\{\frac{1 - S\left(X_j\left(\frac{0.95}{H}\right); c_l, \lambda_l\right)}{L}, \frac{\pi\left(X_j\left(\frac{0.95}{H}\right); c_m, \lambda_m\right)}{M}, \frac{0.95}{H}\right\} \end{aligned} \tag{14}$$

where c_l, λ_l, c_m, λ_m, c_h, λ_h refer to the centres and radii (bandwidths) of the three linguistic properties, and $X_j\left(\frac{0.95}{L}\right)$, $X_j\left(\frac{0.95}{M}\right)$, $X_j\left(\frac{0.95}{H}\right)$ refer to the corresponding feature values X_j at which the three linguistic properties attain membership values of 0.95.

Let M_j be the mean of the pattern points along the jth axis. Then M_{jl} and M_{jh} are defined as the mean (along the jth axis) of the pattern points having co-ordinate values in the range $[X_{j_{min}}, M_j)$ and $(M_j, X_{j_{max}}]$, respectively, where $X_{j_{max}}$ and $X_{j_{min}}$ denote the upper and lower bounds of the dynamic range of feature X_j (for the training set) considering numerical values only. For the three linguistic property sets, we define the centres as [17]

$$\begin{aligned} c_{medium(X_j)} &= M_j \\ c_{low(X_j)} &= M_{j_l} \\ c_{high(X_j)} &= M_{j_h} \end{aligned} \tag{15}$$

and the corresponding radii as

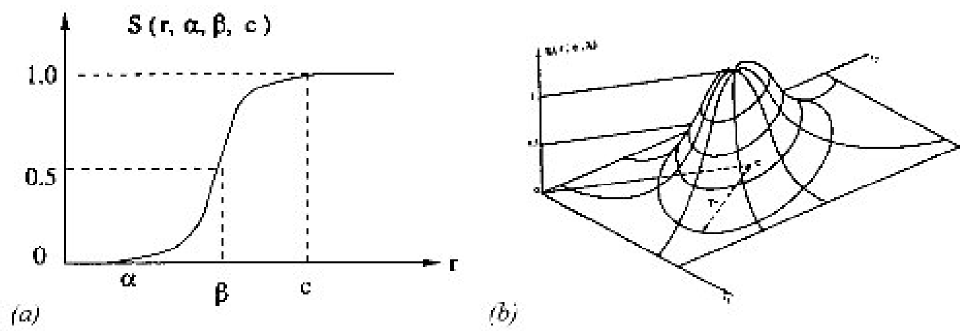

Fig. 2. (a) S function, and (b) π function.

$$\lambda_{low(X_j)} = 2\,(c_{medium(X_j)} - c_{low(X_j)})$$

$$\lambda_{high(X_j)} = 2\,(c_{high(X_j)} - c_{medium(X_j)}) \qquad (16)$$

$$\lambda_{medium(X_j)} = \frac{\lambda_{low}(X_j)\,(X_{j_{max}} - c_{medium(X_j)}) + \lambda_{high(X_j)}\,(c_{medium(X_j)} - X_{j_{min}})}{X_{j_{max}} - X_{j_{min}}}$$

Here we take into account the distribution of the pattern points along each feature axis, while choosing the corresponding centres and radii of the linguistic properties. Besides, the amount of overlap between the three linguistic properties can be different along the different axes, depending on the pattern set.

This combination of choices for the λ's and c's automatically ensures that each quantitative input feature value r_j along the jth axis for pattern $\vec{X}_i$ is assigned membership value combinations in the corresponding three-dimensional linguistic space of Eq. (11), in such a way that at least one of $\mu_{low(X_{ij})}(\vec{X}_i)$, $\mu_{medium(X_{ij})}(\vec{X}_i)$ or $\mu_{high(X_{ij})}(\vec{X}_i)$ is greater than 0.5 in the interval $[c_{low}, c_{high}]$. This allows a pattern $\vec{X}_i$, to have strong membership of at least one of the properties *low*, *medium*, or *high*.

3.2. Output Vector

Consider an l-class problem domain such that we have l nodes in the output layer. Let the n-dimensional vectors $\boldsymbol{O}_k$ and $\boldsymbol{V}_k$ denote the mean and standard deviation (respectively) of the numerical training data for the kth class. The weighted distance of the training pattern $\vec{X}_i$ from the kth class is defined as

$$z_{ik} = \sqrt{\sum_{j=1}^{n}\left[\frac{X_{ij} - O_{kj}}{V_{kj}}\right]^2} \quad \text{for } k = 1, \ldots, l \qquad (17)$$

where X_{ij} is the value of the jth component of the ith pattern point, and C_k is the kth class.

The membership [10] of the ith pattern to class C_k is defined as follows:

$$\mu_k(\vec{X}_i) = \frac{1}{1 + \left(\dfrac{z_{ik}}{f_d}\right)^{f_e}} \qquad (18)$$

where z_{ik} is the weighted distance from Eq. (17), and the positive constants f_d and f_e are the denominational and exponential fuzzy generators controlling the amount of fuzziness in this class-membership set (i.e. in the distance set). Obviously, $\mu_k(\vec{X}_i)$ lies in the interval [0,1].

3.3. Architecture

In the fuzzy c-means algorithm, the membership value of any pattern vector $\boldsymbol{X}_j$ to a class k can be represented from Eq. (10) as

$$u_{kj} = \frac{1}{\sum_{i=1}^{c}\left(\dfrac{d'_{kj}}{d'_{ij}}\right)^{\frac{2}{m-1}}} \qquad (19)$$

where d'_{kj} is the distance of the pattern vector from the centre v_k of the kth class.

In the architecture of a conventional RBF, the transfer function of a hidden node is modelled by Gaussian distribution function, which allows a hidden node to produce non-zero response, even when the input pattern vector does not match with the corresponding cluster centre. The nonzero response depends upon the variance of the Gaussian distribution function. Since the Gaussian distribution transfer function of each hidden node is its local property, the output of each hidden node can be computed locally.

In the fuzzy radial basis function network, the objective is to perform a fuzzy partitioning of the data in the hidden layer. In other words, the objective is to compute the membership value of any pattern to a class corresponding to any hidden node (Eq. (19)) without using a Gaussian distribution type transfer function, and combining the responses of the hidden nodes in the output layer. However, the membership value of any pattern to any cluster depends upon the distances of the pattern from all existing clusters (Eq. (19)). Therefore, if the architecture of the Fuzzy RBF is exactly the same as that of a conventional RBF, then it is not possible to compute the fuzzy membership values of a pattern locally. To perform the local computation of Eq. (19), a modified architecture for Fuzzy RBF is used.

Equation (19) can be written as

$$u_{kj} = \frac{h_k^{(j)}}{\sum_{i=1}^{c} h_i^{(j)}} \qquad (20)$$

where

$$h_i^{(j)} = \left(\frac{1}{d_{ij}}\right)^{\frac{2}{m-1}} \qquad (21)$$

The activation of each output node in the output layer is given as

$$y_i^{(p)} = \sum_{j=1}^{c} W_{ij} h_j^{(p)} \qquad (22)$$

where $y_i^{(p)}$ is the response of the ith output node when X_p is present at the input of the network. From Eqs (20) and (21), $y_i^{(p)}$ can be written as

$$y_i^{(p)} = \frac{1}{H_a^{(p)}} \sum_{j=1}^{c} W_{ij} h_j^{(p)} \tag{23}$$

where

$$H_a^{(p)} = \sum_{i=1}^{c} h_i^{(p)} \tag{24}$$

Equations (21) and (23) reveal the fact that h_i's can be computed locally in the hidden nodes, and the activation of the output nodes can be computed from the hidden node activations with an additional normalisation by the total output in the hidden layer (H_a). For this purpose, we introduce an auxiliary hidden node in the Fuzzy RBF (as shown in Fig. 1) to compute the total activation in the hidden layer, and feed it to the output layer. The weights of the links from all hidden nodes to the auxiliary hidden node are set to unity. Note that the membership value u_{kj} of Eq. (19) is implicitly included in the network architecture in terms of the hidden node activations.

During training, the rule for updating the weights (Eq. (4)) is accordingly modified by the response of the auxiliary node, and is given as

$$\Delta W_{ij}^{(p)} = \frac{\eta}{H_a^{(p)}} ({}^*y_i^{(p)} - y_i^{(p)})\, h_j^{(p)} \tag{25}$$

where η is the learning rate and ${}^*y_i^{(p)}$ is the target output computed from Eq. (18) as ${}^*y_i^{(p)} = \mu_i(\bar{X}_p)$. $\Delta W_{ij}^{(p)}$ is the change in W_{ij} during training when X_p is presented at the input in the $3n$-dimensional form of Eq. (11).

4. Results

The Fuzzy RBF has been tested on a set of 871 Indian Telugu vowel sounds, available from *http://www.isical.ac.in/~sushmita/patterns*. These were uttered by three male speakers aged between 30 and 35 years, in a Consonant-Vowel-Consonant context. The data set has three features F_1, F_2 and F_3, corresponding to the first, second and third vowel formant frequencies obtained through spectrum analysis of the speech data. Thus, the dimension of the input vector in Eq. (11) is 9. Note that the boundaries of the classes in the given data set are seen to be ill-defined (fuzzy). Figure 3 shows a 2D projection of the 3D feature space of the six vowel classes (∂, a, i, u, e, o) in the $F_1 - F_2$ plane (for ease of depiction). The parameters $f_d = 5$ and $f_e = 1$ in Eq. (18) were chosen after several experiments. The training set consisted of 50% data, selected randomly, while the remaining patterns constituted the test set.

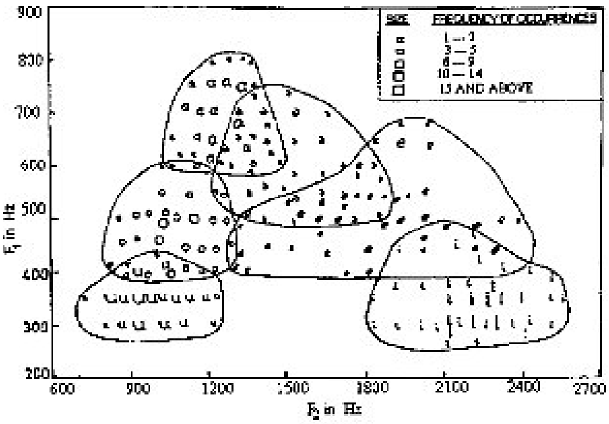

Fig. 3. Vowel diagram.

Table 1 shows the recognition scores obtained by the Fuzzy RBF for different numbers of hidden nodes (clusters) c and fuzzifier m. It can be observed that the overall performance is better with a larger number of hidden nodes. This is natural, since more hidden nodes imply a larger number of clusters. Generally, the performance starts degrading from around $m = 3$. There exists a band of m-values, around the middle of the range along each c, where the performance of the model is poor. These are also evident from Fig. 4, which demonstrates the change in classification performance of the FRBF for different combinations of fuzzifier (m) and number of hidden nodes (c).

It can be seen that a value of $m = 1$ indicates a crisp partition of the data. Since no separate Gaussian distribution to model the cluster structures is considered here, the output of a hidden node is zero if there is a small mismatch between the input data and the corresponding cluster centre; and it is unity only when the input data perfectly matches with the corresponding cluster centre. Therefore, the output of the FRBF is zero for all input data, except the points representing the cluster centres. As m increases, the membership value of an input takes nonzero values, and it decreases with the distance of the point from the cluster centre. The rate of decrease falls with the increase of m. In the limit, when $m \to \infty$ membership values of all points to all clusters become equal, and the FRBF loses the classification capability. Depending on the cluster structure, the FRBF network exhibits an optimum performance for a certain range of m.

Sample confusion matrices generated by the Fuzzy RBF over the training and test sets are provided in

Table 1. Recognition score (%).

Nodes c	Class	Fuzzifier m = 1.25	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5
8	∂	10.0	12.5	0.0	27.5	25.0	17.5	0.0	17.5	20.0	10.0	0.0	5.0	5.0	5.0
	a	73.3	77.8	77.8	70.0	66.7	57.8	57.8	55.6	55.6	77.8	68.9	66.7	66.7	66.7
	i	83.1	80.7	89.2	95.2	88.0	73.5	75.9	89.2	91.6	89.2	91.6	78.3	79.5	80.7
	u	96.2	96.2	92.3	93.6	96.2	92.3	75.6	64.1	64.1	92.3	64.1	96.2	96.2	96.2
	e	78.3	83.0	76.4	73.6	72.6	77.4	73.6	67.9	60.4	74.5	58.5	82.1	80.2	80.2
	o	78.8	80.0	94.1	80.0	71.8	61.2	67.1	56.5	57.6	16.5	63.5	87.1	90.6	90.6
	Net	75.7	77.4	78.3	77.1	74.6	68.7	64.8	63.2	62.2	63.6	62.5	76.2	76.7	76.9
10	∂	22.5	25.0	62.5	20.0	5.0	0.0	5.0	12.5	0.0	40.0	30.0	5.0	5.0	5.0
	a	77.8	95.6	48.9	75.6	55.6	55.6	57.8	55.6	60.0	51.1	57.8	66.7	66.7	66.7
	i	97.6	94.0	86.7	86.7	90.4	78.3	88.0	89.2	92.8	88.0	86.7	84.3	88.0	84.3
	u	97.4	92.3	67.9	96.2	93.6	46.2	60.3	55.1	64.1	85.9	93.6	93.6	93.6	93.6
	e	66.0	77.4	85.8	90.6	74.5	68.9	64.2	41.5	60.4	84.0	84.0	84.0	84.0	84.0
	o	72.9	80.0	98.8	77.6	76.5	94.1	70.6	68.2	60.0	90.6	75.3	90.6	92.9	94.1
	Net	76.2	80.8	79.4	80.3	73.0	63.8	63.2	57.0	61.6	79.0	76.9	78.0	79.2	78.7
12	∂	25.0	10.0	17.5	70.0	17.5	10.0	20.0	0.0	5.0	5.0	5.0	5.0	10.0	12.5
	a	97.8	88.9	97.8	55.6	55.6	55.6	55.6	55.6	71.1	71.1	68.9	68.9	68.9	68.9
	i	77.1	89.2	91.6	94.0	89.2	83.1	84.3	86.7	68.7	71.1	90.4	84.3	84.3	84.3
	u	96.2	96.2	87.2	96.2	97.4	85.9	96.2	92.3	96.2	96.2	93.6	93.6	93.6	93.6
	e	87.7	87.7	84.0	74.5	73.6	78.3	78.3	62.3	91.5	89.6	84.0	84.0	84.0	84.0
	o	77.6	80.0	96.5	70.6	69.4	94.1	69.4	96.5	85.9	88.2	84.7	84.7	84.7	84.7
	Net	80.6	81.0	83.8	79.0	73.0	75.1	73.2	72.5	76.9	77.4	78.3	77.1	77.6	77.8
14	∂	22.5	15.0	62.5	65.0	47.5	22.5	7.5	10.0	5.0	5.0	5.0	5.0	5.0	5.0
	a	95.6	95.6	66.7	55.6	55.6	71.1	57.8	93.3	73.3	73.3	75.6	77.8	77.8	77.8
	i	80.7	90.4	83.1	95.2	81.9	88.0	94.0	89.2	84.3	84.3	84.3	85.5	85.5	85.5
	u	82.1	91.0	92.3	88.5	92.3	97.4	93.6	96.2	93.6	93.6	93.6	93.6	93.6	93.6
	e	85.8	82.1	84.0	79.2	83.0	84.0	70.8	85.8	86.8	86.8	86.8	86.8	86.8	86.8
	o	92.9	91.8	95.3	92.9	96.5	38.8	55.3	80.0	94.1	94.1	94.1	94.1	94.1	94.1
	Net	80.8	82.4	83.8	82.8	81.0	71.4	69.1	81.0	80.1	80.1	80.3	80.8	80.8	80.8
16	∂	20.0	20.0	35.0	5.0	10.0	10.0	12.5	5.0	7.5	10.0	10.0	10.0	10.0	10.0
	a	97.8	97.8	93.3	88.9	55.6	57.8	55.6	68.9	91.1	68.9	93.3	82.2	88.9	88.9
	i	95.2	89.2	89.2	81.9	83.1	92.8	86.7	95.2	90.4	94.0	91.6	94.0	94.0	94.0
	u	91.0	93.6	92.3	87.2	92.3	92.3	96.2	91.0	93.6	91.0	96.2	96.2	96.2	96.2
	e	77.4	84.9	82.1	96.2	83.0	76.4	81.1	77.4	82.1	80.2	82.1	82.1	82.1	82.1
	o	92.9	80.0	91.8	92.9	78.8	85.9	71.8	97.6	88.2	97.6	83.5	85.9	85.9	85.9
	Net	83.1	81.7	84.0	82.2	74.4	76.2	74.1	79.6	81.0	80.6	81.2	81.0	81.7	81.7
20	∂	25.0	22.5	32.5	52.5	17.5	40.0	20.0	47.5	12.5	42.5	40.0	47.5	35.0	42.5
	a	97.8	97.8	95.6	51.1	55.6	53.3	73.3	77.8	97.8	75.6	97.8	91.1	97.8	80.0
	i	88.0	88.0	89.2	88.0	86.7	81.9	86.7	90.4	89.2	94.0	92.8	94.0	92.8	94.0
	u	85.9	94.9	92.3	94.9	94.9	96.2	94.9	94.9	94.9	92.3	94.9	96.2	94.9	92.3
	e	83.0	87.7	86.8	86.8	84.9	84.9	84.0	82.1	83.0	80.2	79.2	82.1	80.2	80.2
	o	94.1	91.8	94.1	91.8	91.8	92.9	88.2	92.9	91.8	96.5	89.4	92.9	92.9	96.5
	Net	82.8	84.9	85.6	82.6	79.2	80.6	80.3	84.4	83.1	84.2	84.9	86.7	85.4	84.7

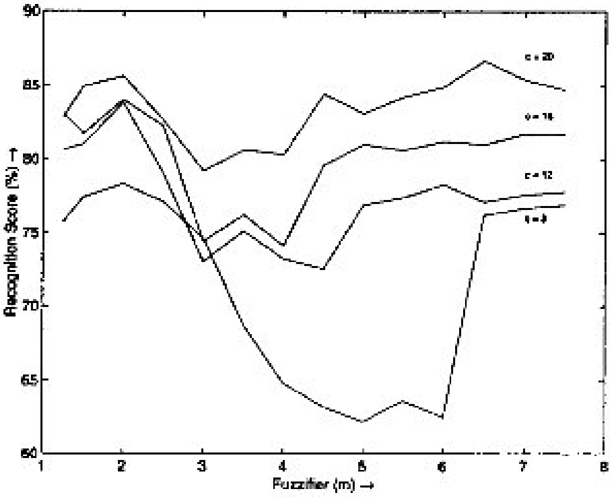

Fig. 4. Recognition scores (%) of FRBF for different combinations of fuzzifier (m) and number of hidden nodes (c) with *Vowel* data.

Table 2. Sample confusion matrices of FRBF.

	Training						Testing					
	∂	a	i	u	e	o	∂	a	i	u	e	o
∂	20	5	0	0	12	3	17	7	0	0	12	4
a	8	25	0	0	0	12	10	24	0	0	0	11
i	0	0	72	0	11	0	0	0	70	0	13	0
u	0	0	0	74	0	4	0	0	0	72	0	5
e	1	0	10	0	90	5	1	0	12	0	86	5
o	0	0	0	6	1	78	0	0	0	8	2	75

Table 3. Comparison between various models.

# Nodes	RBF			FRBF (m = 2.0)			Bayes' classifier
	6	9	12	8	10	12	
Score (%)	59.7	71.2	71.4	78.3	79.4	83.8	79.2

Table 2. The overall recognition scores are 82.2% and 79.3%, respectively. It can be observed that most misclassifications are made between adjacent vowel classes in Fig. 3, *viz.* ∂ and e, a and o, and i and e. This behaviour is similar in the case of the RBF, the only differences being the poorer classification performance and longer time for convergence.

Table 3 demonstrates the performance of the RBF, Fuzzy RBF and Bayes' classifier on the Vowel data. The conventional RBF used n input nodes, a crisp l-dimensional output and hard c-means clustering at the hidden layer. The Bayes' classifier has been implemented for multivariate normal patterns with the *a priori* probabilities $p_i = \frac{|C_i|}{N}$, where $|C_i|$ indicates the number of patterns in the ith class and N is the total number of pattern points. The covariance matrices were considered different for each pattern class. The choice of normal densities for the vowel data has been found to be justified [18]. It should be noted that the results of the RBF was obtained after 20,000 iterations. The fuzzy RBF, on the other hand, provided a superior recognition score with around only 5000 iterations.

5. Conclusions

A new fuzzy radial basis function network is developed by integrating the merits of the fuzzy c-means algorithm and the RBF network. The architecture of the RBF network is suitably modified to incorporate fuzzy c-means computation within the model. The input is in terms of linguistic values *low, medium* and *high*, modelled using S and π-functions. The output is provided as class membership values.

It has been experimentally demonstrated that the FRBF model exhibits better classification performance than the conventional RBF model, on real-life data, for a suitable range of fuzzifier values. The fuzzifier value depends upon the cluster structure. Theoretical and/or experimental investigation towards adaptively selecting a suitable range of fuzzifier values, for a given input data, holds promise for future research.

References

1. Huntsberger TL, Ajjimarangsee P. Parallel self-organizing feature maps for unsupervised pattern recognition. Int J General Systems 1990; 16: 357–372
2. Carpenter GA, Grossberg S, Rosen DB. Fuzzy ART: fast stable learning and categorisation of analog patterns by an adaptive resonance system. Neural Networks 1991; 4: 759–771
3. Simpson PK. Fuzzy min-max neural networks: 1. Classification. IEEE Trans Neural Networks 1992; 3: 776–786
4. Pal SK, Mitra S. Neuro-fuzzy Pattern Recognition: Methods in Soft Computing. New York: John Wiley, 1999

5. Zadeh LA. Fuzzy sets. Infor and Control 1965; 8: 338–353
6. Haykin S. Neural Networks: A Comprehensive Foundation. New York: Macmillan College Publishing, 1994
7. Bezdek, JC, Tsao EC, Pal NR. Fuzzy Kohonen clustering networks. Proc 1st IEEE Int Con on Fuzzy Systems, San Diego, CA, 1992; 1035–1043
8. Bezdek JC. Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1981
9. Carpenter GA, Grossberg S, Markuzon N, Reynolds JH, Rosen DB. Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans Neural Networks 1992; 3: 698–713
10. Pal SK, Mitra S. Multi-layer perceptron, fuzzy sets and classification. IEEE Trans Neural Networks 1992; 3: 683–697
11. Mitra S, Pal SK. Fuzzy multi-layer perceptron, inferencing and rule generation. IEEE Trans Neural Networks 1995; 6: 51–63
12. Jang JSR, Sun CT. Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Trans Neural Networks 1993; 4: 156–159
13. Pedrycz W. Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Trans Neural Networks 1998; 9: 601–612
14. Moody J, Darken CJ. Fast learning in networks of locally-tuned processing units. Neural Computation 1989; 1: 281–294
15. Hush DR, Horne B, Salas JM. Error surfaces for multilayer perceptrons. IEEE Trans Systems, Man and Cybernetics 1993; 22: 1152–1161
16. Tou JT, Gonzalez RC. Pattern Recognition Principles. London: Addison-Wisley, 1974
17. Mitra S. Fuzzy MLP based expert system for medical diagnosis. Fuzzy Sets and Systems 1994; 65: 285–296
18. Pal SK. Studies on the Application of Fuzzy Set-Theoretic Approach in Some Problems of Pattern Recognition and Man-Machine Communication by Voice. PhD thesis, University of Calcutta, India, 1978