A Novel Fuzzy Clustering Algorithm for Radial Basis Function Neural Network

Arun Kulkarni
Research scholar, CSE Department, SGGSIE&T
Nanded, India
e-mail: kkkarun@yahoo.com

Sanjeev Bonde
Professor, Department of EXTC, SGGSIE&T
Nanded, India
e-mail:svbonde@sggs.ac.in

Uday Kulkarni
Professor, CSE Department, SGGSIE&T
Nanded, India
e-mail: uvkulkarni@sggs.ac.in

***Abstract* —** A Fuzzy Radial basis function neural network (FRBFNN) classifier is proposed in the framework of Radial basis function neural network (RBFNN). This classifier is constructed using class-specific fuzzy clustering to form the clusters which represent the neurons i.e. fuzzy set hyperspheres (FSHs) in the hidden layer of FRBFNN. The creation of these FSHs is based on the maximum spread from inter-class information and intra-class fuzzy membership mechanism. The proposed approach is fast, independent of parameters, and shows good data visualization. The Least mean square training between the hidden layer to output layer in RBFNN is avoided, thus reduces the time complexity. The FRBFNN is trained quickly due to the fast converge of input data to form the FHSs in the hidden layer. The output is determined by the union operation of the FHSs outputs which are connected to the class nodes in the output layer. The performance of the proposed FRBFNN is compared with the other RBFNNs using ten benchmark datasets. The empirical findings demonstrate that the proposed FRBFNN is highly efficient classifier for pattern recognition.

Keywords- *Radial basis function neural network; fuzzy membership function; fuzzy neuron; fuzzy set hypersphere; fuzzy clustering.*

*****

I. Introduction

Pattern recognition has become an active research area in recent years, and the RBFNNs have become popular pattern classifiers which have been applied in several engineering applications. In RBFNN, apart from the determination of the centroids and the width of the clusters along with their locations in the hidden layer, the Gaussian function (GF) also plays an important role. The RBFNN model is a feed-forward neural network which adopts the radial basis function as its activation function for hidden neurons and was first proposed by [1]. The key issue with the RBFNN is the determination of centers and radii of the radial basis functions along with the number of hidden nodes in the hidden layer [2]. The RBFNN centers and radii parameters are determined using supervised, or unsupervised clustering. The popular approach to determine the hidden nodes in RBFNN is K-means clustering algorithm [3]. Many clustering algorithms were proposed by various researchers to create the hidden layer of RBFNN which includes enhanced K-means [4], subtractive clustering [5], fuzzy clustering [6], ART [7], scatter based clustering [8], input-output clustering [9], output-constricted clustering [10], conditional Fuzzy C-Means (CFCM) [11], particle swarm optimization (PSO) [12], genetic algorithm (GA) [13], ant colony optimization (ACO) [14], artificial fish swarm optimization (AFSO) [15], etc. Recently Modjtaba Rouhani, Dawood S. Javan have proposed two heuristic clustering algorithms [16] and an improvement in one of the algorithm has been suggested in [17] for RBFNN. An approach to train RBFNN using class-specific clustering has been described in [19].

The proposed approach aims to develop a classifier based on the combination of RBFNN and fuzzy clustering having following advantages over earlier RBFNNs and Fuzzy neural network classifiers.

1. The Gaussian neurons in the hidden layer of the RBFNN are replaced by fuzzy neurons. The fuzzy neurons are characterized by the fuzzy membership function which camouflages the clustered patterns leading to 100% training accuracy for any dataset.
2. The proposed approach does not make use of any tuning parameter.
3. The learning between the hidden layer to output layer does not use traditional least mean square algorithm, thus it eliminates the local minima and reduces the time complexity.
4. The connection between the hidden layer to output layer i.e the creation of class nodes is done concurrently with the creation of FSHs in the hidden layer. The output of the class nodes is determined by the union operation of the respective output of FSHs.
5. The retrieval time is also less.

So, the proposed FRBFNN classifier overcomes most of the drawbacks of earlier RBFNNs and results into a precise classifier for pattern recognition. The rest of the paper is organized as follows. The following section describes fundamentals of RBFNN in brief. Section III describes the architecture of FRBFNN classifier. In section IV learning rule i.e. maximum spread fuzzy clustering algorithm to construct FRBFNN classifier is discussed. Simulation results with case studies are explained in section V. Finally the section VI concludes the paper with future work.

II Radial Basis Function Neural Network

The RBFNN is a type of feed-forward artificial neural network (ANN) [20]. It has three layers, namely input layer, hidden layer and output layer and has recently attracted extensive research interest because of its simple architecture as shown in Fig.1. The RBFNN high regularization capability, good local specialization, global generalization ability and is the main rival to the multi-layered perceptron.

The unique feature of the RBFNN is the process of forming the clusters/nodes/neuron in the hidden layer. If the centers $C_1, C_2, ..., C_j$ of these clusters are known, then the distance of the input pattern from the cluster center can be measured. If C_j is a center of cluster then C_j is represented as $C_j = [c_{j1}, c_{j2}, ..., c_{jn}]$.

For each neuron in the hidden layer, the weights between the input layer and hidden layer represent the co-ordinates of the center of the cluster. when input layer receives a pattern, $X = [x_1, x_2, ..., x_n]$ the output of j^{th} hidden neuron i.e. ψ_j is determined using the following equation

$$\psi_j = \exp\left(-\frac{\sum_{i=1}^{n} (X_i - c_{ji})^2}{2\sigma^2} \right) \quad (1)$$

Where σ is the width of the cluster.

The weights between hidden layer to output layer are determined by gradient descent approach and the output of i^{th} node in the output layer of the RBFNN is given as

$$y_i = f(\sum_{j=1}^{J} w_{ij} * \psi_j), where\ i = 1, 2, ..., m \quad (2)$$

where f is the non-linear activation function

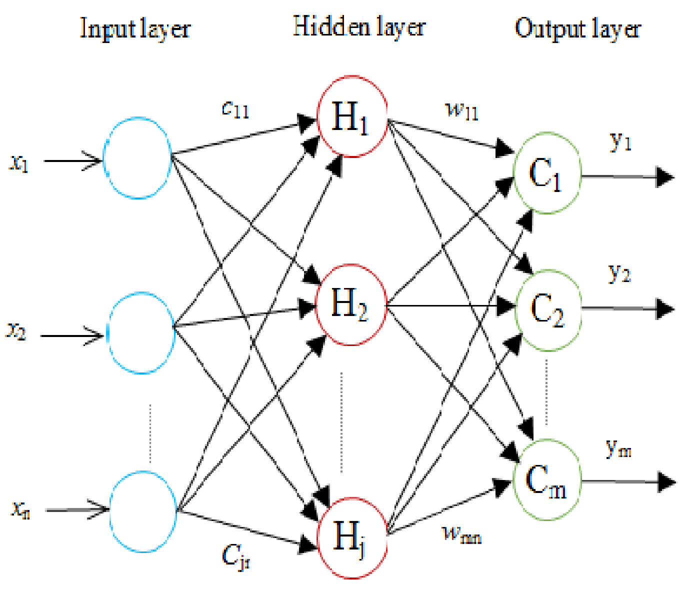

Figure 1. Architecture of RBFNN

III FRBFNN architecture

The architecture of FRBFNN is shown in Fig. 2. FRBFNN has three layers namely the input layer F_I, hidden layer F_H and output layer F_C. As depicted, the n nodes in F_I layer accept n dimensional pattern and the nodes in F_I layer do not perform any processing and simply forward input to the nodes in the hidden layer. The nodes in F_H layer and F_C layer are created using MSFC algorithim which is described in the next section and each node in the hidden layer is a FHS whose processing is characterized by a fuzzy membership function defined as:

$$m_j\left(X_h, C_j, r_j\right) = f(l, r_j) \quad (3)$$

where, $X_h = (x_{h1}, x_{h2}, \ldots, x_{hn})$ is an input pattern, r_j is a radius of j^{th} FHS H_j with a centroid $C_j = [c_{j1}, c_{j2}, ..., c_{jn}]$. $f()$ is defined as:

$$f(l, r_j) = \begin{cases} 1 & l \le r_j \\ r_j / l & otherwise \end{cases} \quad (4)$$

where l is an euclidean distance between X_h and C_j. The weights of the links between F_I and F_H are stored in the matrix C which represent the centroids of the FSHs. The k nodes in F_C layer which are also constructed during training represent the classes.

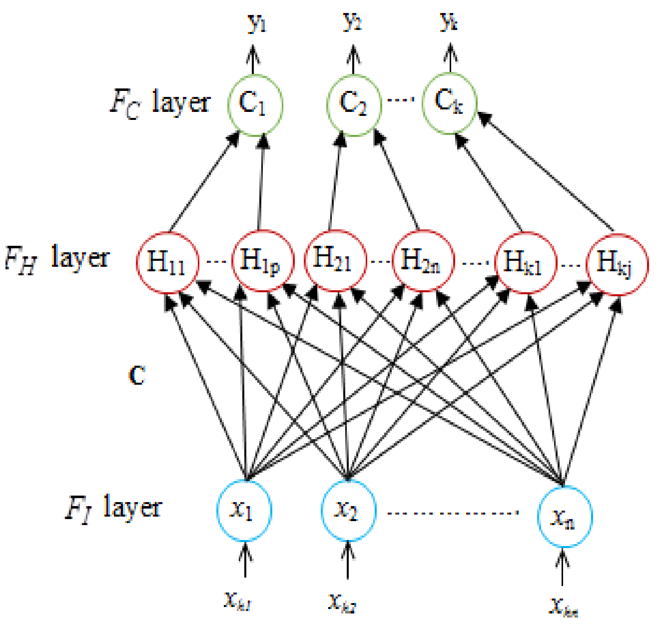

Figure 2. FRBFNN architecture

As shown in Fig. 2 the class 1 has p number of FHSs, class 2 has n number of FHSs and class 3 has j number of FHSs. All these FHSs are connected to their respective class nodes.

ISSN: 2454-4248
751 – 756

IV. FRBFNN learning

The Maximum spread fuzzy clustering (MSFC) algorithm to construct the FRBFNN classifier is described in the following section.

Let Z be the training set containing P training pairs, let h^{th} be the input pattern represented as (X_h, d_h), where d_h represents the desired output of X_h.

Consider α_k patterns of class C_k to be the subset of set Z, then following steps are executed for k classes varying from $k = 1,........,K$.

Step 1: The distance between the patterns of k^{th} class with other class patterns is determined and stored in A^k.

$$A^k = \left[\left\|X_i - X_j\right\|\right]_{\alpha_k \times t_k}, i = 1,2,.....,\alpha_k, and\ j = 1,2,.....,t_k, \quad (5)$$

$$where\ X_i \in C_k,\ X_j \notin C_k, and\ t_k = P - \alpha_k.$$

Step 2: The minimum distance of each pattern from the patterns of other class is determined as

$$B^k = \min\left(A^k\right) \quad (6)$$

Step 3: From B^k the pattern X_j^k having maximum distance is considered to be the centroid of FHS with initial radius equal the maximum distance.

$$g^k = \max\left(B^k\right) \quad (7)$$

Step 4: Using X_j^k, g^k and the membership function mentioned in (3), the patterns of the same class are clustered.

Step 5: Determine the final radius by following equation

$$r_j^k = \begin{cases} \max\limits_{i=1}^{n_j} d_i, & for\ n_j > 1 \\ \dfrac{g^k}{2}, & for\ n_j = 1 \end{cases} \quad (8)$$

Where d_i is the distance between i^{th} clustered pattern and the centroid X_j^k, where n_j is the total number of patterns clustered by the FHS and calculate $\alpha_k \leftarrow \alpha_k - n_j$.

Step 6: If $\alpha_k \neq 0$ repeat the above steps else go to step 7.

Step 7: Create a class node k in the output layer and connect the created FHSs.

Step 8: Repeat the above steps for all classes i.e. till $k \neq K$.

Once the FRBNN classifier is constructed, check the performance in terms of recognition rate. The procedure for testing is done as given below.

1. Apply input pattern from the data set to the input layer.
2. Using membership function, calculate the membership of input pattern for each FHS in the hidden layer.
3. The output of each class node is determined byfinding maximum membership value i.e. union of FHSs connected to the respective class node.
4. The pattern belongs to the class node which gives maximum value.

V. Simulation results

To evaluate the performance of FHNN classifier two case studies along with obtained results have been discussed in the following sub-sections.The learning algorithm is implemented in Matlab 2016a

Case Study 1:

To have better visualization of MSFC algorithm to construct FRBFNN, a 2-dimensional 3 class example is illustrated. The training set consists of twenty four 2D patterns; (1, 5), (2, 4.5), (0.75, 6), (1, 7.5), (1.5, 7), (0.75, 8), (1.5, 8.5), (1, 9) belonging to class 1, (1, 1), (2, 1.5), (3, 1.5), (0.75, 3), (1.5, 2.5), (2, 3.5), (3.25, 4), (4, 1) belonging to class 2 and (3.7, 6), (4, 5), (4, 7), (3, 8), (3.75, 8), (3.5, 8.5), (3.75, 9), (3, 9) belonging to class 3. The scatter plot of these patterns is shown in Fig. 3

The MSFC algorithm constructed four FSHs; two FSHs for class 1 and one FSH for class 2 and class 3 each. The centroids of class 1 FSHs are (0.75, 6.0), and (1.0, 9.0) with radii 2.61, and 1 respectively. The class 2 FHS centroid is (4, 1) with radius 3.8161. Similarly, class 3 FSH centroid is (4.0, 7.0) with radius 2.2361. The Fig. 8 shows the constructed FSHs. The detail description of these FSHs is explained below.

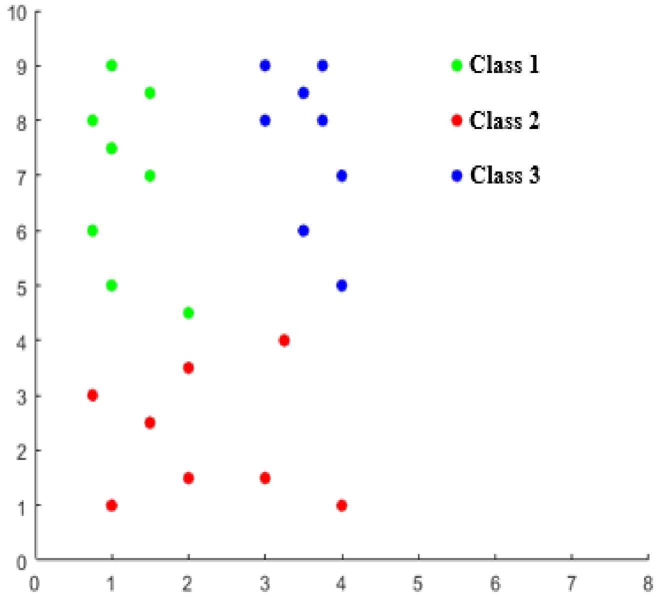

Figure 3. 2-D Scatter plot of patterns

Initially as per the algorithm, the clustering process for the patterns in class 1 is as follows: As per the step 1, the inter class distance of class1 patterns with class 2 and class 3 is calculated. Using step 2 and step 3 the pattern (.75, 6) of class 1 is considered as centroid as it has maximum spread with initial radius 2.7951 due to the pattern (4, 6) of class3. Later by step 4, (.75, 6) as the centroid of first FHS with the radius equal to 2.7951, cluster the patterns of class 1 using the defined membership function. The first FHS clusters all the patterns except (1, 9) as shown in Fig. 4. Then using step 5 we calculate the final radius i.e. 2.6101 equal to the maximum distance between the centroid and clustered pattern (1.5, 8.5) which is shown in Fig. 4.

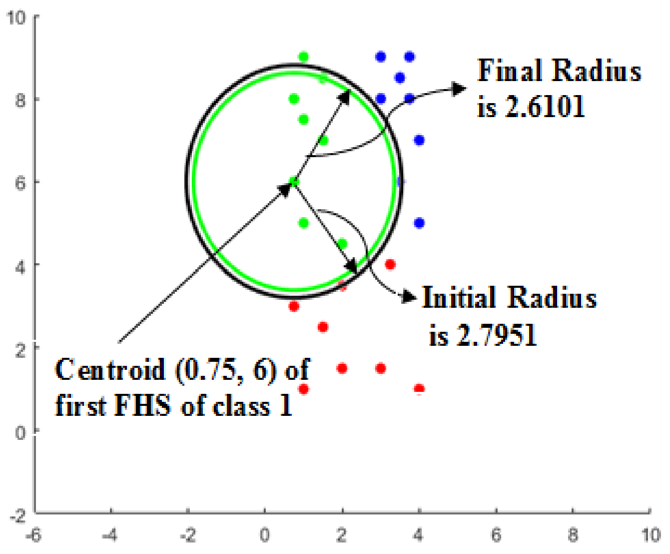

Figure 4. First FHS of class 1 with centroid and radius

Since one more pattern of class 1 is left the same process is repeated as per step 6, which is shown in Fig. 5. As this FHS has only one pattern as per step 5 the final radius assigned is equal to half of the initial radius. So the process of creation of FSHs for this class is over and as per the step 7 the class node for this class is created and the connection between the two FHSs and the class node is done as shown in Fig. 9.

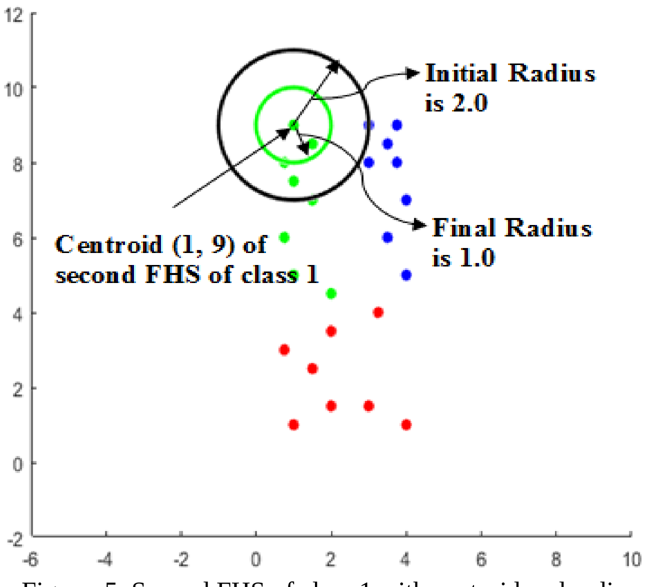

Figure 5. Second FHS of class 1 with centroid and radius

Once the possible FHSs for one class along with their connections with class nodes is done then as per step 8 the same procedure is repeated for other classes.So for class 2 as per the MSFC algorithm, the pattern (4, 1) is selected as centroid with initial radius to be 4.0, due to the pattern (2, 4.5) of class 1.Then the final radius 3.8161 is adjusted by using step 5 because of the maximum distance between the centroid and the pattern (.75, 3) of class 2 which is shown in Fig. 6.

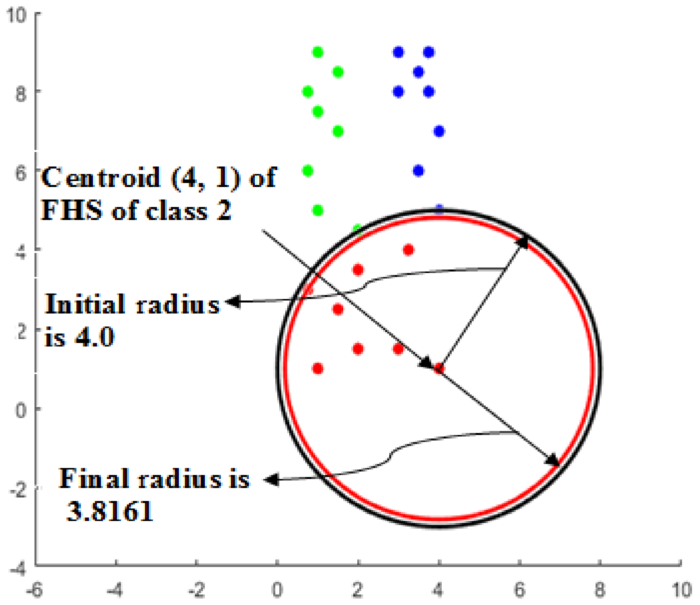

Figure 6. FHS of class 2 with centroid and radius

As all the patterns of class 2 are clustered in one FSH a class node is created and connection between this hypersphere and the class node is done which is shown in Fig. 9.
Till this point, we have created three FSHs and two class nodes for two classes and the respective connections are also done. Similarly, we do the same process and obtain the FHS with centroid (4, 7) with final radius to be 2.5 due to the pattern (3.5, 8.5) of class 3 as shown in Fig. 7.The connection between this FHS and the class node is shown in Fig. 9.

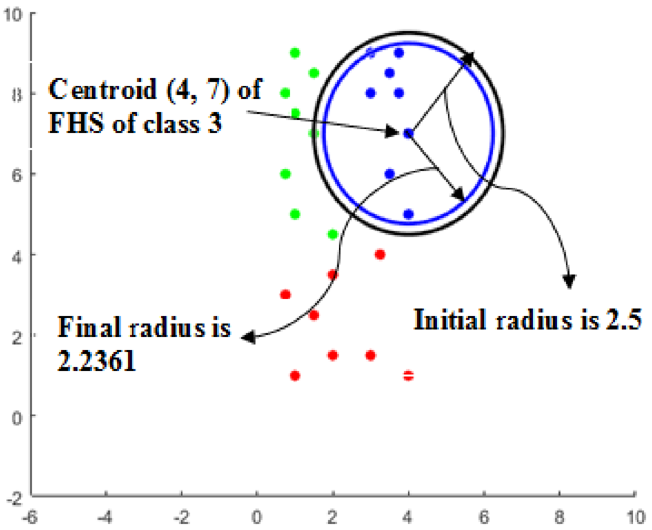

Figure 7. FHS of class 3 with centroid and radius

Finally, we created four FHSs for all three classes i.e. two FHSs for class 1, one for class 2 and class 3 each.The Fig. 8 Shows this FHSs with their centroid and final radius.

 ISSN: 2454-4248
751 – 756

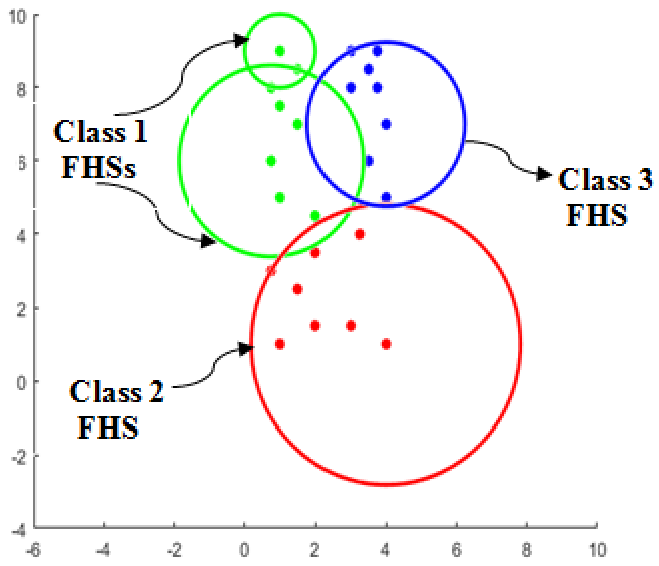

Figure 8. FHSs of all 3 classes with centroid and radius

The constructed FRBFNN for 2-D example of 3 classes is shown in Fig. 9. As seen in the figure the input layer has two nodes for two-dimensional input patterns, four FHSs in the hidden layer with three class nodes in the output layer constructed as per MSFC algorithm. The centers of the hyperspheres are shown in C.

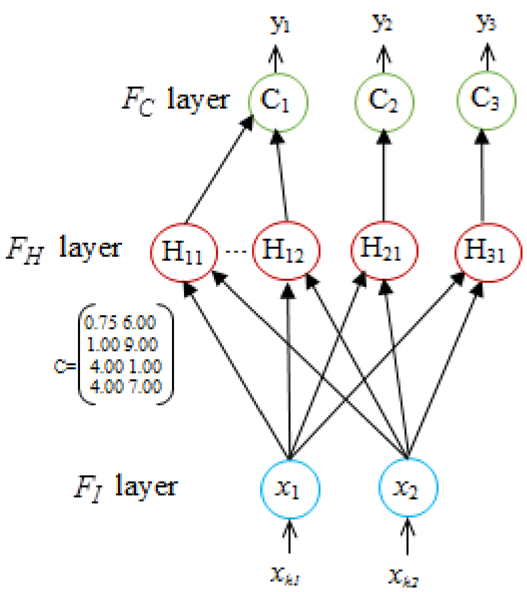

Figure 9. FRBFNN architecture for 2-D example

Case Study 2:

The performance of proposed algorithms is verified using ten UCI datasets [18]. The experimental procedure in [16] is followed, to have a fair comparison between the FRBFNN and other classifiers. The average percentage 5-fold validation test accuracies are tabulated in Table 1 along with the results given in [16]. The results show that FRBFNN classifier is superior than [16] for six datasets and comparable with other datasets.

Table 1: Average percentage 5-fold validation test accurac accuracies

Dataset	MSFC	RBF	RBF-R	RBF-N	RBF-WTA
Hepatitis	**88.2**	65.0	81.9	81.1	82.1
Zoo	92.4	83.8	95.2	94.3	96.2
Glass	**75.0**	38.7	66.1	66.3	69.1
Heart	77.0	73.5	81.9	80.5	80.6
Ecoli	**83.5**	69.5	78.5	79.3	81.0
Liver	**69.3**	53.8	62.2	62.8	61.0
Ionosphere	90.0	81.5	95.5	95.2	94.3
Monks-3	84.8	97.5	99.0	95.8	68.6
Breast	**96.3**	94.1	96.3	96.4	97.0
Pima	**76.9**	71.0	75.3	72.1	73.8

VI. Conclusion and Future work

The proposed FRBFNN classifier creates fuzzy set hyperspheres on the basis of inter-class and intra-class fuzzy membership metric with the maximum spread. The Gaussian neurons from the hidden layer of traditional RBFNNs are transformed to fuzzy neurons characterized by the membership function, assuring 100% training efficiency for any dataset. The MSFC algorithms is quick as it avoids LMS training and local minima. Hence, the proposed FRBFNN is precise and quick converging pattern classifier which provides 100% training and significant testing efficiency. The most important feature of algorithm is about the inclusion of pattern by the membership function and not by the shortest distance. So irrespective of the distance, the patterns are classified by the prominence of radius of the FSH.

The proposed FRBFNN classifier can be implemented in the complete and effective system for biometric authentication. Randomizing the order of clustering may improve the test efficiency as well as the number of FSHs formed by MSFC algorithm. If the outlier handling techniques can be implemented before clustering then in turn this may increase the test efficiency and reduce the number of FSHs for the classes. Different techniques of selection of centroids for FSHs can be investigated.

References

[1] D. Lowe, "Multi-variable functional interpolation and adaptive networks," Complex System, vol. 2, pp. 321-355, March 1988.

[2] B. Carse, A.G. Pipe, T. C. Fogarty, T. Hill, "Evolving radial basis function neural networks using a genetic algorithm," IEEE International Conference on Evolutionary Computation, Perth, pp. 300-305, December 1995.

[3] J. Moody, C.J. Darken, "Fast learning in networks of locally-tuned processing units," Neural Computing, vol.1, pp. 281-294, October 1989.

[4] S. Chen, "Nonlinear time series modeling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," Electronic Letters, vol 3, pp. 117-118, January 1995.

[5] H. Sarimveis, A. Alexandridis, G. Bafas, "A fast training algorithm for RBF networks based on subtractive clustering," Neurocomputing, pp. 501-505, April 2003.

[6] Gorge E. Tsekouras, J. Tsimikas, "On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization," Fuzzy sets and systems vol 221, pp. 65-89, June 2013.

[7] L. Shie-Jue, H. Chun-Liang, "An ART-based construction of RBF networks," IEEE Trans. On Neural Network, vol 13, pp. 1308-1321, November 2002.

[8] I. Sohn, N. Ansari, "Configuring RBF neural networks," Electronic Letters, vol 34, pp.684-685, April 1998.

[9] Z. Uykan, C. Guzelis, M. E. Ã‡elebi, H. N. Koivo, RBFNN," IEEE Trans. Neural Network, vol 11, pp. 851-858, July 2000.

[10] D. Wang, X. J. Zeng, J. A. Keane, "A clustering algorithm for radial basis function neural network initialization," Neurocomputing, vol 77, pp. 144-155, February 2012.

[11] Antonios D. Niros, George E. Tsekouras, "A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach," Fuzzy Sets and Systems, vol 193, pp. 62-84, April 2012

[12] H. M. Feng, "Self-generation RBFNs using evolutional PSO learning," Neurocomputing, vol 70, pp. 241-251, December 2006.

[13] S. A. Billings, G. L. Zheng, "Radial basis function network configuration using genetic algorithms," Neural Network, vol (8), pp. 877-890, January 1995.

[14] J. Li, X. Liu, "Melt index prediction by RBF neural network optimized with an adaptive new ant colony optimization algorithm," J.Appl.Polym.Sci., vol 119, pp. 3093-3100, September 2011.

[15] W. Shen, X. Guo, C. Wu, D. Wu, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm," Knowl. Based System, vol 24, pp. 378-385, April 2011.

[16] M. Rouhani, D. S. Javan, "Two fast and accurate heuristic RBF learning Rules for data classification," Neural Networks, vol 75, pp. 150-161, January 2016.

[17] Yuanshan Liu, He Huang, Ting wen Huang b, Xusheng Qian,"An improved maximum spread algorithm with application to complex-valued RBF neural networks," Neurocomputing , vol 216, pp. 261-267 , Jully 2016.

[18] A. Frank, and A. Asuncion, "UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science," URL: http://archive.ics.uci.edu/ml, 2010.

[19] J. Raitoharju, S. Kiranyaz, M. Gabbouj, "Training radial basis function neural networks for classification via class-specific clustering,"IEEE Transactions on Neural Networks and Learning Systems, vol 27, pp. 2458-2471, November 2015.

[20] Simon S. Haykin. "Neural Networks: A Comprehensive Foundation,"Prentice-Hall, NJ. 1994.