Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 167 (2020) 2606–2616

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

Fuzzy Neural Network for Pattern Classification

Arun Kulkarni[a], Nikita kulkarni[b]

[a]*Research Scholar, SGGS Institute of Engineering and Technology, Nanded-413601, India*
[b]*Associate Applications developer, Oracle Financial Services software Ltd., Goregaon(E) Mumbai, 400063, India*

Abstract

This research work proposes a fuzzy neural network (FNN) for pattern classification. The proposed network is the modified version of the Radial basis function neural network (RBFNN). FNN uses supervised fuzzy clustering and pruning algorithm to determine the precise number of clusters with proper centroid and width to form the processing nodes in the hidden layer. These clusters represent fuzzy set hyperspheres (FSHs), which are defined by the fuzzy membership function. The training between the hidden layer to output layer which is done by using the LMS algorithm in RBFNN is avoided, and the output is determined by using the fuzzy union operation. The fuzzy membership function shields the clustered patterns resulting in 100% accuracy for the data set used during training. Unlike other clustering algorithms used to construct the hidden layer of RBFNN, the proposed clustering algorithm is independent of tuning parameters and is fast in training and retrieval. Thus FNN reduces the computation time, guarantees 100% accuracy for any training set, and provides superior and comparable recognition accuracy for the datasets with the precise number of FSHs in the hidden layer. Hence the proposed FNN can be used for pattern classification.

Keywords:
Fuzzy neural network; Radial basis function neural network; Pruning algorithm; Fuzzy Membership function;

1. Introduction

In recent years, pattern recognition has become an active research area in various domains. For linearly separable patterns a single layer neural network is used for classification, whereas for non-separable patterns a multi-layer network like Error back propagation neural network (EBPNN) is utilized. EBPNN encompasses three layers, i.e. input layer, hidden layer and output layer [1][2]. Determining the number of hidden layer's neurons is the major drawback of EBPNN. To overcome this drawback radial basis function neural network was introduced by low [3].

* Corresponding author.Tel.:+91-983-342-8466; fax:+91-022-260-53590.
E-mail address: kkkarun@yahoo.com

1877-0509
10.1016/j.procs.2020.03.321

This is similar to EBPNN except supervised or unsupervised clustering algorithms are used to decide the appropriate count of neurons in the hidden layer. Various clustering algorithms have been developed by different researchers to optimize the neurons in the hidden layer. Apart from neural networks, fuzzy logic has emerged as a new phenomenon, as it supports the flexible sense of membership for defining the real world context. So a fuzzy neural network has been proposed by integrating fuzzy clustering algorithms to state the number of hidden neurons of radial basis function neural network. Fuzzy logic provides an alternative way to represent the linguistic and subjective attributes of the real world in computing [4]. Fuzzy sets provide membership value to the elements belonging to the universe of discourse while crisp sets give the binary output. The membership value depends on the fuzzy membership function. Different membership functions are used to generate the fuzzy set. Let the universe of discourse be P and b an element belonging to P, then a fuzzy set $\tilde{A}$ is

$$\tilde{A} = \{(b, \mu_{\tilde{A}}(b)), b \epsilon P\} \tag{1}$$

where each pair $(b, \mu_{\tilde{A}}(b))$ is called a singleton defining value of membership for b represented by $\mu_{\tilde{A}}(b)$.
Let $X = \{a_1, a_2, a_3, a_4\}$ be the reference set of students and let $\tilde{A}$ be the fuzzy set of the student getting distinction then depending on the grade, the fuzzy set can be represented as,

$$\tilde{A} = \{(a_1, 0.2), (a_2, 1), (a_3, 0.9), (a_4, 0.5)\} \tag{2}$$

From the fuzzy set stated above, we conclude that a_2 has the distinction as he has full fuzzy membership value equal to 1 whereas a_3 is close to distinction while a_1 is far from distinction.
The fuzzy sets can also be represented mathematically as,

$$\mu_{\tilde{A}}(y) = \frac{1}{(1+y)^2} \tag{3}$$

For various values of y depending upon its presence in the universe of discourse the membership function can be plotted and the same can be defined in the form of fuzzy set. Different shapes of membership functions exist in the literature along with the fuzzy set operations like union, intersection, complement, etc. to demonstrate the usefulness of fuzzy logic [5].

1.1. Fuzzy neural network

The Fuzzy neural network is the fusion of ANN and fuzzy logic. These networks have become popular as the output generated is not crisp but fuzzy. Due to this, the partial belonging to a particular class helps in better classification. The fuzzy membership value and the operations on the fuzzy set play an important role in the working of fuzzy neural networks. In view of the benefits of fuzzy logic and ANN, a fuzzy minmax neural network (FMNN) was proposed for classification by [6]. FMNN is based on the accumulation of fuzzy hyperboxes. FMNN training is done by creating hyperboxes belonging to different classes. The training algorithm has two parts: expansion and contraction. During the expansion process, the size of the hyperboxes is decided by the expansion coefficient. The contraction process removes the overlap between other class hyperboxes [6].

Later this concept was extended by [7] to construct the hyperboxes using labeled and unlabeled data. The fuzzy neural network with a compensatory neuron (FMCN) is proposed by Nandedkar and Biswas [8]. To increase the accuracy of FMNN many learning algorithms and new architectures [9], [10], [11], [12] have been proposed. The drawback of these networks is the effect of the parameters adjusting the size, expansion and contraction process along with the overlap test. Apart from this many other clustering algorithms were proposed by various researchers which include fuzzy clustering [13], [14], [15], [16], [17], [18], [19], [20], [21] for constructing the RBFNN hidden layer. The proposed FNN is the extension of [22] which uses the pruning algorithm [23] after fuzzy clustering to optimize the neurons in the hidden layer, and fuzzy union operation to find the output of the network. This classifier provides an improvement in earlier classifiers. The rest of the paper is organized as follows. The following section 2 gives the introduction of RBFNN in brief along with the clustering algorithms suggested by different researchers. Section 3 describes the architecture and learning of the FNN. In section 4 experimental results with different case studies have been discussed. Lastly, section 5 provides the conclusion and also the future scope.

2. Radial basis function neural network

The RBFNN model adopts the radial basis function as its activation function for hidden neurons and was first proposed by [3]. It is a feed forward network. As stated earlier RBFNN is a three layer network as shown in Fig. 1. The input layer is a non-processing layer and accepts the input patterns. The neurons in the input layer are equal to the total count of features present in the pattern. The hidden layer is created by clustering and the number of neurons is equal to the number of clusters. The output layer is a class layer having the number of neurons equal to the number of classes.

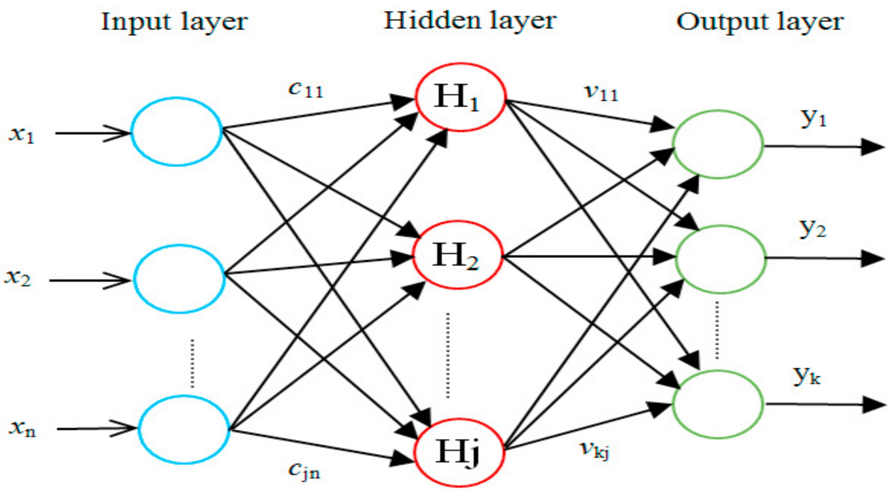

Fig. 1. Architecture of Radial Basis Function Neural Network

As shown in Fig. 1 the links between input layers to hidden layer are weights pertaining to the centroid of clusters which are determined during clustering. The weights between hidden layers to the output layer are determined by least mean square (LMS) algorithm.

The key issue with the RBFNN lies in the determination of centers and radii of the clusters, along with total processing nodes in the hidden layer[24], [25]. The cluster centers and radii parameters are estimated by using supervised, or unsupervised clustering. The simple technique is K-means clustering algorithm. Modjtaba Rouhani, Dawood S. Javan have proposed two heuristic clustering algorithms [26], and further improvement has been suggested in [27]. Class-specific clustering for RBFNN has been explained in [28].

3. Fuzzy neural network architecture and learning Algorithm

The architecture of the fuzzy neural network is shown in Fig. 2. As shown, the connection between the input layer to the hidden layer is fully connected. The output of each hidden neuron or FSH is determined by a fuzzy membership function. As seen from Fig. 2, there is the partial connection between output and hidden layer since the FSHs created during clustering for that class are only connected to the class node. The output of the class node is determined by the fuzzy union operation.

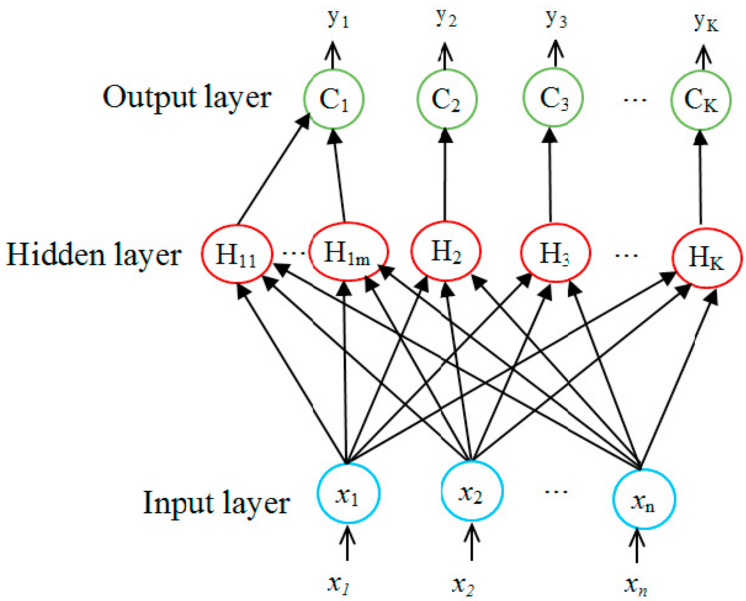

Fig. 2. Fuzzy Neural Network

From Fig. 2, it is observed that class 1 has m number of clusters while other classes are having only one cluster, which may vary in accordance with the application. The training of proposed architecture consist of two steps:
a) Creating FSH in the hidden layer of FNN by performing fuzzy clustering with Maximum count. In this process choose the pattern as centroid which clusters the maximum number of patterns of its own class using fuzzy membership function [22]. After completing the clustering process, perform Pruning algorithm for reducing the single pattern clusters.
The fuzzy membership function is given by

$$f(l, r_j) = \begin{cases} 1 & l \leq r_j \\ r_j/l & \text{otherwise} \end{cases} \tag{4}$$

where l is a euclidean distance between the input pattern and centroid of j^{th} FSH, while r_j is the radius of FSH.
b) The output layer is constructed by creating the class nodes connected with the associated FSHs from the hidden layer, that are created during clustering for that class.

3.1. Fuzzy clustering with maximum count and pruning algorithm(FCMCPA)

Let $\mathbf{V}$ be the training set consisting of N number of training pairs. The input pattern and its desired output is represented by $\{\boldsymbol{P}_n, d_k\}$, where $n = 1, 2, ..., N$, and $k = 1, 2, 3, 4..., K$, for K classes. Let t_k be the total number of patterns belonging to class c_k. The process of clustering is done in two steps: Initially, the possible number of clusters are formed and then the pruning algorithm is used to optimize the number of clusters by reducing the single pattern cluster.
1) Clustering: In this process, the clusters or FSHs are formed for each class by considering the individual class patterns along with other class patterns.
step 1: Let $\mathbf{X} = X_1, X_2, \ldots\ldots\ldots\ldots\ldots\ldots, X_{tk}$ be the total number of pattern of class k and $\mathbf{Y} = Y_1, Y_2, \ldots\ldots\ldots\ldots\ldots\ldots, Y_{N-tk}$ be

the pattern of other class from the training set **V**.
step 2: For each pattern, the maximum number of patterns it can cluster is determined by algorithm 1.
step 3: The pattern which clusters the maximum number of patterns will be chosen as centroid and the distance between the centroid and farthest pattern within the clustered patterns will be the radius. If it clusters only one pattern then the radius is optimized to half.
step 4: The above steps will be repeated until all the patterns of this class are clustered.
step 5: Step 1 to step 4 will be repeated for all the classes.

Algorithm 1 Count the number of clustered patterns

```
Input: X^k ,Y
Output: Count
  for i := 1 → t_k do
    for j := 1 → N − t_k do
      d(j) ← [||X_i − Y_j||]
    end for
    r ← min(d)
    n ← 0
    for q := 1 → t_k do
      l ← [||X_i − X_q||]
      if l < r then
        num ← num + 1
      end if
    end for
    Count(i) ← num
  end for
```

3.2. Pruning algorithm:

The proposed algorithm removes the single pattern clusters if these clusters are camouflaged by their own class clusters. Let $\mathbf{Q}^k = Q_1^k, Q_2^k, ..., Q_n^k$, be the set representing n cluster centroids clustering more than one pattern for class k with radius stored in $R^k = r_1^k, r_2^k, ..., r_n^k$, respectively.
where $k = 1, 2, 3, 4, ..., K$, for K classes. Let $\mathbf{S}^k = S_1^k, S_2^k, ..., S_m^k$, be the set representing m cluster centroids clustering only one pattern for class k with radius stored in $W^k = w_1^k, w_2^k, ..., w_m^k$, respectively. Following steps are carried to prune the clusters with a single pattern.

- Step 1. Compute the membership value of S_j^p where $j = 1, 2, ..., m$ with respect to the existing clusters in $\mathbf{Q}^k$ with radii in R^k for $k = 1, 2, ..., K$ along with clusters in $\mathbf{S}^k$ having respective radius in W^k, for $k = 1, 2, ..., K$ and $k \neq p$.
- Step 2. If the membership value for S_j^p where $j = 1, 2, ..., m$ is maximum for any of the clusters in $\mathbf{Q}^k$ for $k = p$, then prune the cluster by removing it from $\mathbf{S}^p$ and respective radius from W^k.
- Step 3. Repeat the steps 1 and 2 for all classes i.e. $p \neq K$.

3.3. Testing of FNN

Once the final clusters i.e. FSHs are created using the FCMCPA algorithm, accordingly the connection between the output layer and hidden layer is done as explained in the earlier section. The performance in terms of recognition rate is tested using the following procedure.

1. Apply the pattern from the data set to the input layer.

2. Using the membership function, calculate the membership value of the input pattern for each FSH in the hidden layer by considering its centroid and radius.

3. Finally, determine the output for each class by performing the union operation by considering the membership values of created FSHs for that class.

4. The input pattern is said to belong to the class which will give the maximum membership value.

4. Experimental results

The performance of FNN is evaluated by different case studies. The sub-sections also discuss the results obtained after the implementation of algorithms. The algorithms are implemented in Matlab R2010a. Different case studies have been discussed to understand the importance of the FNN classifier. The case study 1 defines the accuracy of the FNN with existing classifier. In case study 2 the number of FSHs versus recognition error is stated. The working of the FNN classifier is discussed in case study 3. Finally, the evaluation parameters of the FNN classifier are determined in case study 4. As seen from the results, the FNN classifier has a better and comparable recognition rate with respect to other classifiers.

4.1. Case study 1:

The seven UCI data sets [29] as shown in Table 1 were used for evaluating the performance of proposed FNN. The average percentage validation test accuracies using 5-fold are tabulated in Table 2. The results obtained are compared with the results given in [22], [26] and [30]. As seen from Table 2, for three data sets the results of the FNN classifier are superior and comparable with remaining data sets. The bar chart for test accuracy is as shown in Fig. 3

Table 1. Description of UCI dataset

Dataset	Ionosphere	Monks-3	Breast	Pima	Hepatitis	Heart	Liver
Features	34	6	9	8	19	13	6
No. of samples	351	432	599	768	80	297	345
No. of classes	2	2	2	2	2	2	2

Table 2. Test accuracy

Dataset	**FNN**	Rule 1	Rule 2	RBF-R	RBF-N	RBF-WTA
Ionosphere	92.0	92.0	92.6	95.5	95.2	94.3
Monks-3	85.74	87.1	85.7	99.0	95.8	68.6
Breast	95.85	98.1	98.1	96.3	96.4	97.0
Pima	**76.0**	75.5	75.5	75.3	72.1	73.8
Hepatitis	**92.94**	85.9	84.7	81.9	81.1	82.1
Heart	77.04	78.5	78.5	81.9	80.5	80.6
Liver	**68.4**	68.1	68.1	62.2	62.8	61.0

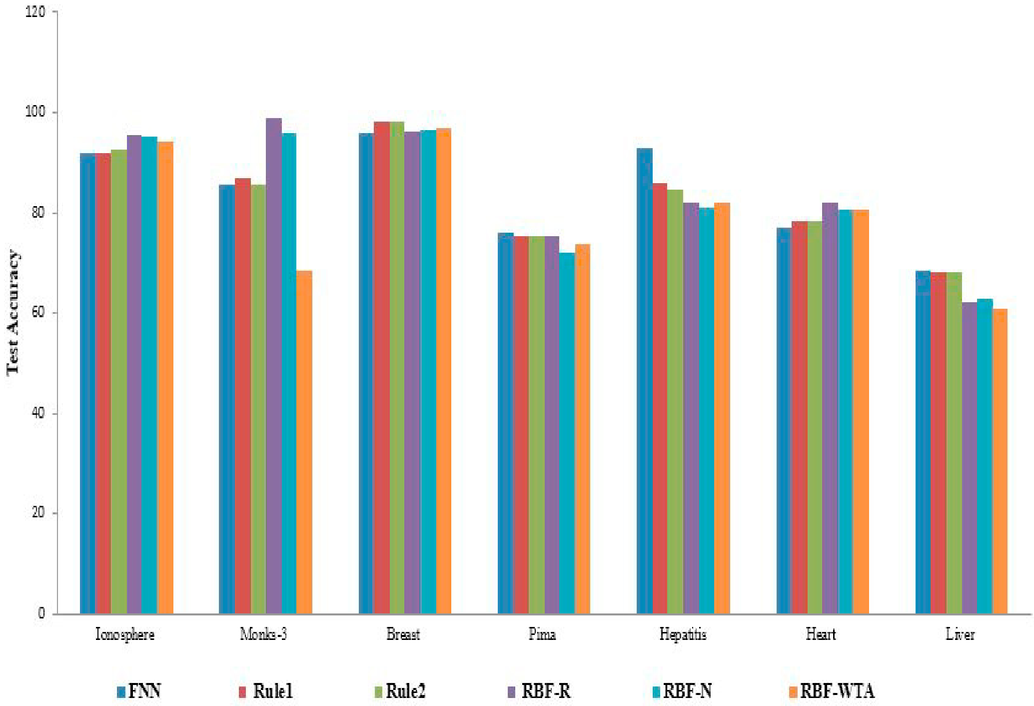

Fig. 3. Bar chart for Test accuracy

4.2. Case study 2:

In this case study, we compared the performance of the FNN classifier in regards to the total number of FSHs/Neurons and relative recognition error with other classifiers as shown in Table 3 and Table 4. The compared results are associated with the results for rule 1 in [22] and classifiers specified in [27]. From the results, it is seen that for fewer number of FSHs, the recognition error of FNN is superior or comparable. The bar chart for recognition error is as shown in Fig. 4 and the bar chart for the average number of FSHs/neurons in the hidden layer is as shown in Fig. 5

Table 3. Recognition error

Dataset	**FNN**	Complex-valued	Real-valued	Rule 1	RBF-R	RBF-N	RBF-WTA	RBF
Heart	22.96	17.08	19.68	21.5	18.1	19.5	19.4	21.1
Ionosphere	8.0	7.14	6.13	8.0	4.5	4.8	5.7	12.9
Breast	4.15	2.9	1.1	1.9	3.7	3.6	3.0	6.7
Pima	24	24.63	24.31	24.5	24.7	27.9	26.2	29.8

Table 4. Average FSHs/Neurons in hidden layer

Dataset	**FNN**	Complex-valued	Real-valued	Rule 1	RBF-R	RBF-N	RBF-WTA	RBF
Heart	48.6	44.12	46.15	66	24	27	46	54
Ionosphere	31.6	117.12	116.45	67	65	48	66.6	70
Breast	35	28.12	29.48	46	40	35	40	140
Pima	**153**	225.7	237.4	211	120	264	160	154

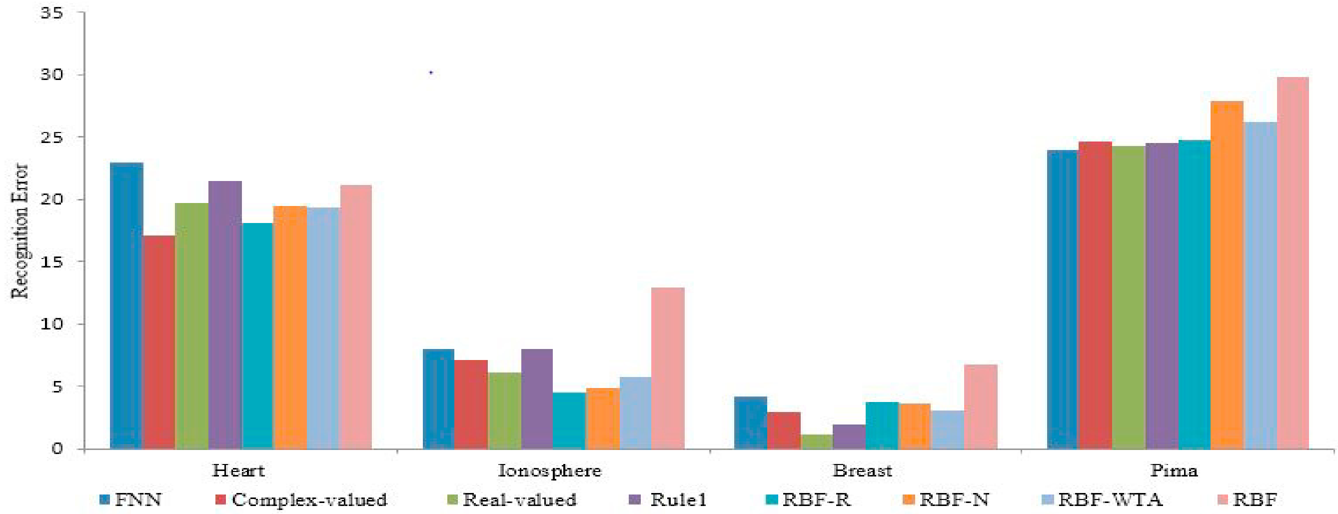

Fig. 4. Bar chart for recognition error

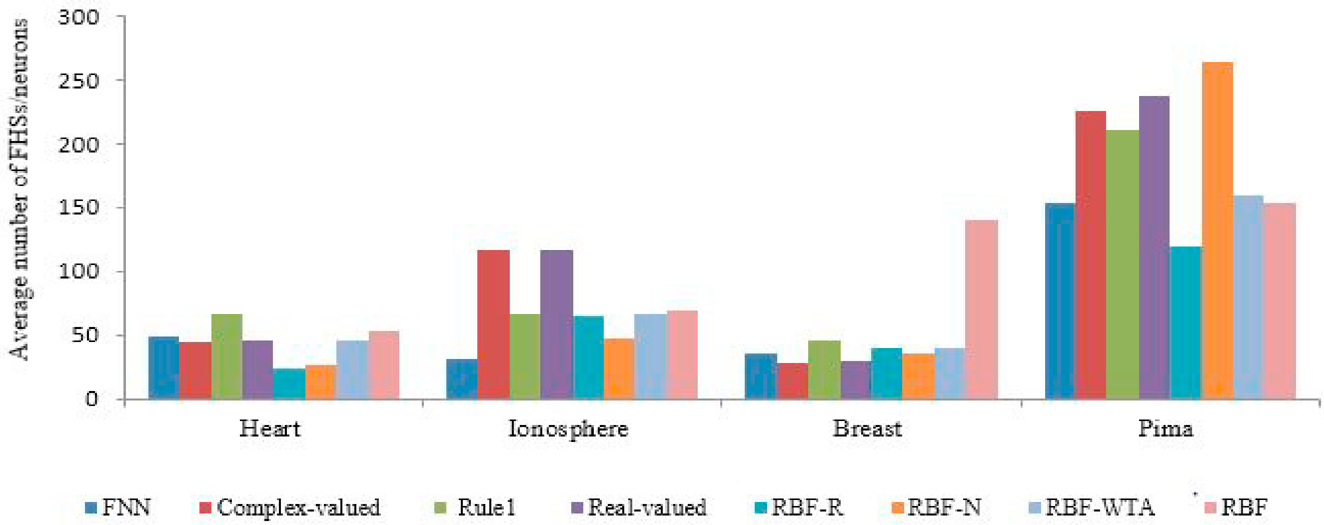

Fig. 5. Bar chart for average number of FSHs/neurons in hidden layer

4.3. Case study 3:

The FCMCPA algorithm for the FNN classifier is explained for two classes with 2-dimensional feature vectors. Consider that there are sixteen two dimensional patterns; (3, 3), (2, 4), (4, 4), (1, 3), (3, 5.5), (5, 4), (6, 3), (6, 7) belonging to the class 1 and (5.5, 2), (8, 2), (7, 3), (7, 4), (9, 2), (6.5, 5), (8, 4), (9, 3) belonging to the class 2 in the training set. Fig. 6 (a) and (b) shows the scatter plot of 2-d patterns and the constructed FSHs for class 1. The FSHs created for class 2 is shown in Fig. 7 (a) while Fig. 7 (b) shows the combined FSHs for class 1 and class 2. Finally after applying the pruning algorithm one of the clusters of class 2 is removed as shown Fig. 8 (a), and the constructed classifier with four hidden FSHs, two for each class is shown in Fig. 8 (b) respectively

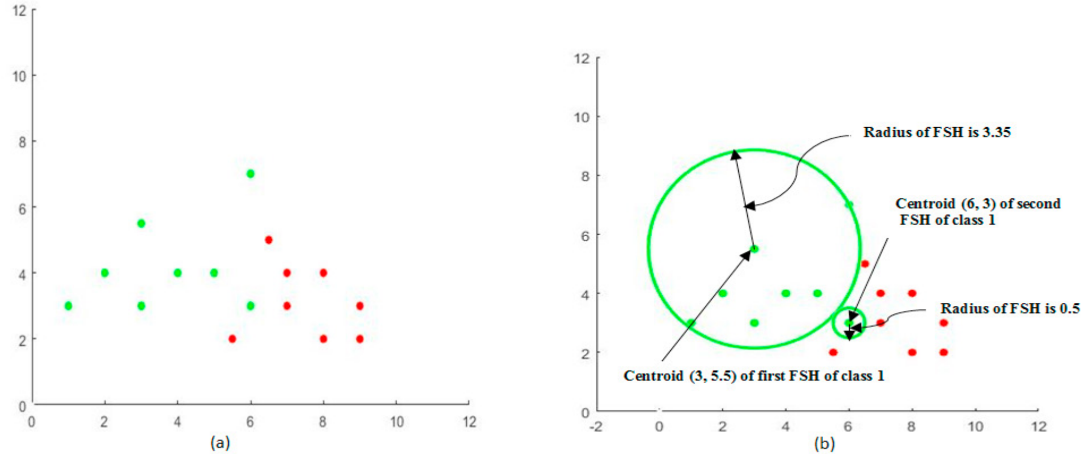

Fig. 6. (a) Scatter plot (b) FSHs for class 1.

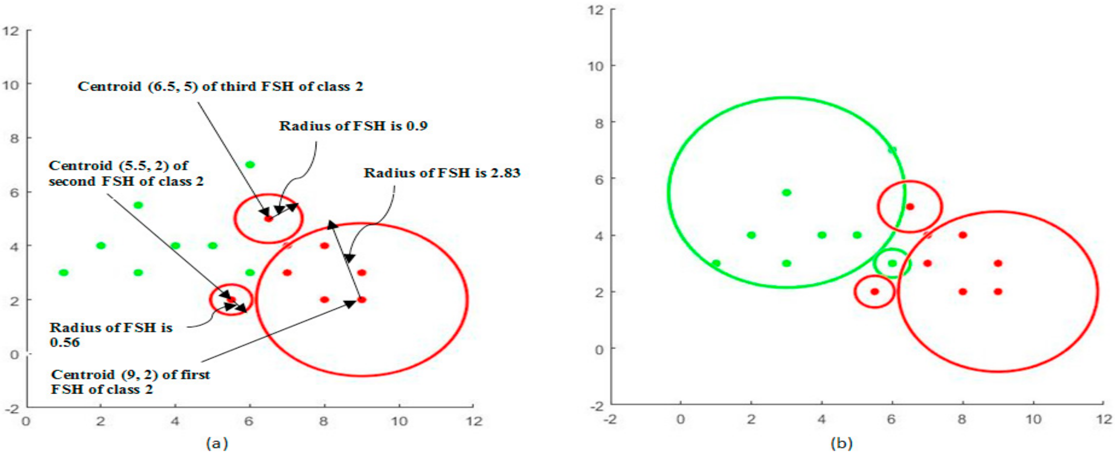

Fig. 7. (a) FSHs for class 2 (b) FSHs for class 1 and class 2.

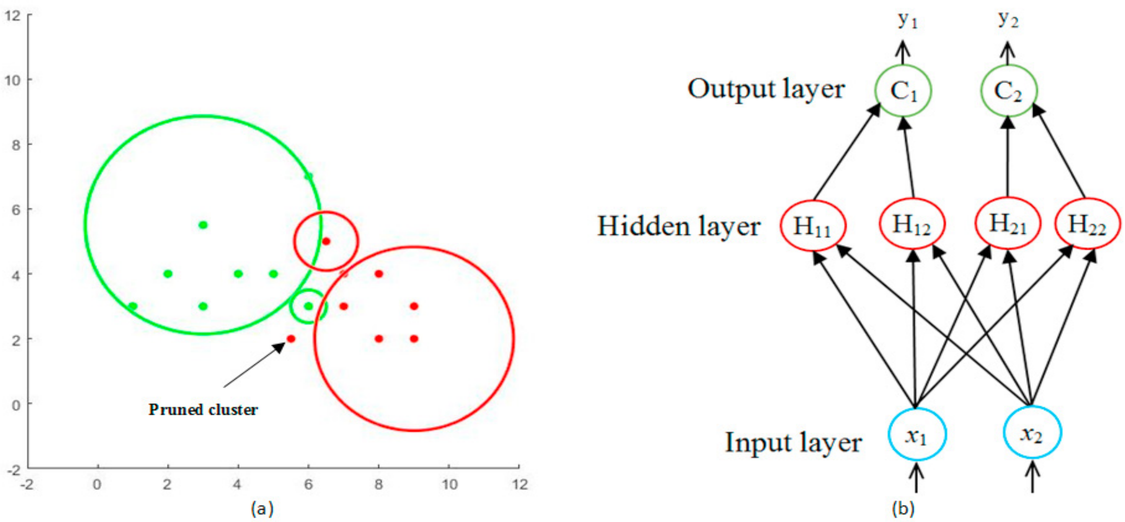

Fig. 8. (a) Final FSHs for class 1 and class 2 (b) FNN Classifier for 2-D example

4.4. Case study 4:

We know that the precision, accuracy, sensitivity, and specificity are the evaluation parameters that define the performance of a classifier [31]. These parameters for the FNN classifier using UCI datasets are estimated. The evaluation parameters for the FNN classifier are determined and the results are tabulated in Table 5. The comparison graph of performance parameters is shown in Fig. 9.

Table 5. Evaluation parameters of FNN classifier

Dataset	Accuracy	sensitivity	specificity	precision
Hepatitis	.94	.93	1.0	1.0
Heart	0.85	0.9	0.8	0.84
Liver	0.77	0.79	0.75	0.83
Ionosphere	0.94	.91	1.0	1.0
Monks-3	0.98	0.8	1.0	1.0
Breast	0.99	0.98	1.0	1.0
Pima	0.77	0.84	0.65	0.85

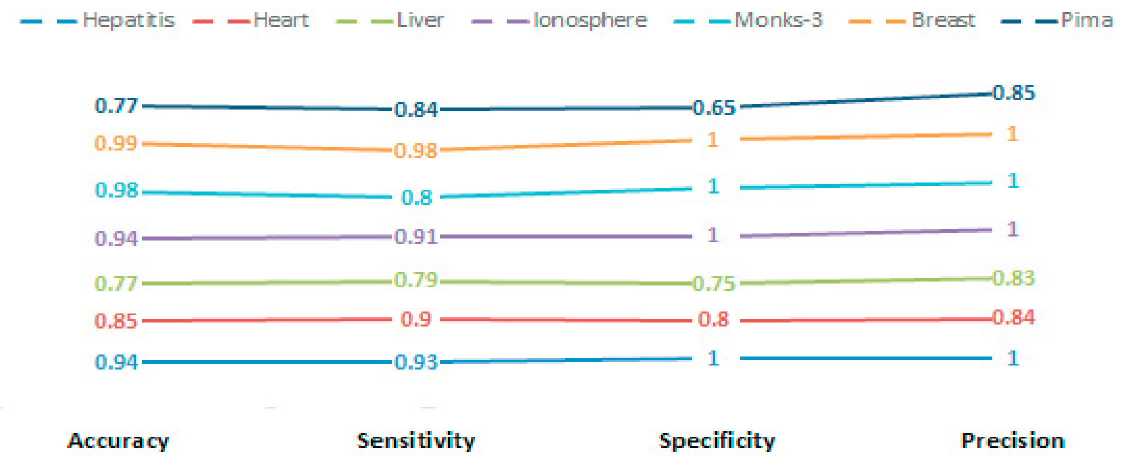

Fig. 9. Evaluation parameters for FNN classifier

5. Conclusions and future scope

The FSHs in the hidden layer of the FNN classifier are created on the basis of fuzzy clustering. The process of FSHs creation is done in two steps. First, the FSHs are created by selecting the FSHs which clusters more patterns of its class using fuzzy membership function and in the second step, the FSHs having a single pattern are pruned as these FSHs are camouflaged by their own class FSHs. Thus the FSHs of FNN are optimized, due to which the computation time for retrieval is reduced. The results of the proposed FNN classifier when compared with Rule 1 and other classifiers are better and comparable in regards to the test accuracy and the creation of FSHs in the hidden layer. Though there is an overlap of clusters of different classes, as seen from the graphical representation in case study 3, it is clearly seen that the efficiency for the training data set will be 100%. Thus the proposed FNN classifier is fast with respect to training and retrieval and should be used for pattern classification.

The algorithm can be modified in terms of selection of the centroid, designing a new fuzzy membership function, and finally pruning algorithm. The parallel clustering process can be adopted along with preprocessing to remove the outliers. The trade-off between the number of clusters and generalization can be minimized.

Acknowledgment: We would like to thank Dr. S. V. Bonde and Dr. U. V. Kulkarni for their valuable guidance and support.

References

[1] Jacek M.Zurada (1994) "Introduction to Artificial Neural Systems", *Jaico publicaton*
[2] Simon S. Haykin. (1994) "Neural Networks: A Comprehensive Foundation", *Prentice-Hall, NJ*
[3] D. Lowe. (1988) "Multi-variable functional interpolation and adaptive networks." *Complex System* **2**: 321-355.
[4] S. N. Sivanandam, S. N. Deepa (2016) "Principles of soft computing", *Wiley publication*
[5] S.Rajasekaran, G. A. Vijayalakshmi pai (2012) "Neural networks, Fuzzy Logic and Genetic Algorithms synthesis and Applications", *PHI Publication*
[6] Simpson, Patrick K. (1992) "Fuzzy min-max neural networks-part 1: Classification." *IEEE Transactions on Neural Networks* **5**: 776-786.
[7] Gabrys, Bogdan and Bargiela, Andrzej. (2000) "General fuzzy min-max neural network for clustering and classification." *IEEE Transactions on Neural Networks* **1**13: 769-783.
[8] Nandedkar, Abhijeet V and Biswas, Prabir K. (2004) "A fuzzy min-max neural network classifier with compensatory neuron architecture." *Proceedings of the 17th International Conference on Pattern Recognition* **4**: 553-556.
[9] Zhang, Huaguang and Liu, Jinhai and Ma, Dazhong and Wang, Zhanshan. (2011) "Data-core-based fuzzy min–max neural network for pattern classification." *IEEE transactions on neural networks* **2**212: 2339-2352.
[10] Davtalab, Reza and Dezfoulian, Mir Hossein and Mansoorizadeh, Muharram. (2014) "Multi-level fuzzy min-max neural network classifier." *IEEE transactions on neural networks and learning systems* **2**53: 470-482.
[11] Mohammed, Mohammed Falah and Lim, Chee Peng. (2015) "An enhanced fuzzy min–max neural network for pattern classification." *IEEE transactions on neural networks and learning systems* **2**63: 417-42.
[12] Mohammed, Mohammed Falah and Lim, Chee Peng.(2017) "Improving the Fuzzy Min-Max neural network with a K-nearest hyperbox expansion rule for pattern classification." *Applied Soft Computing* **52**: 135-145.
[13] Antonios D. Niros, George E. Tsekouras. (2012) "A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach." *Fuzzy Sets and Systems* **193**: 62-84.
[14] Wei Huang, Sung-Kwun Oh, Witold Pedrycz. (2014) "Design of hybrid radial basis function neural networks (HRBFNNs) realized with the aid of hybridization of fuzzy clustering method (FCM) and polynomial neural networks (PNNs)." *Neural Networks* **60**: 166-181.
[15] Antonino Staiano, Roberto Tagliaferri, Witold Pedrycz. (2006) "Improving RBF networks performance in regression tasks by means of a supervised fuzzy clustering." *Neurocomputing* **69**: 1570-1581.
[16] W. Pedrycz. (1998) "Conditional fuzzy clustering in the design of radial basis function neural networks." *Neural Networks, IEEE Trans.* **9**: 601-612.
[17] G. E. Tsekouras, A. Manousakis, C. Vasilakos, K. Kalabokidis. (2015) "Improving the effect of fuzzy clustering on rbf networks performance in terms of particle swarm optimization." *Advances in Engineering Software* **82**: 25-37.
[18] Mahardhika Pratama, Jie Lu, Sreenatha Anavatti, Edwin Lughofer, Chee-Peng Lim. (2016) "An incremental meta-cognitive-based scaffolding fuzzy neural network." *Neurocomputing* **171**: 89-105.
[19] Carl G. Looney. (2009) "Fuzzy connectivity clustering with radial basis kernel functions." *Fuzzy Sets and Systems* **160**: 1868-1885.
[20] Seok-Beom Roh,Tae-Chon Ahn, Witold Pedrycz. (2010) "The design methodology of radial basis function neural networks based on fuzzy K-nearest neighbors approach." *Fuzzy Sets and Systems* **161**: 1803-1822.
[21] Sung-Kwun oh, Witold Pedrycz, Ho-Sung Park. (2003) "Hybrid identification in fuzzy-neural networks." *Fuzzy Sets and Systems* **138**: 399-426.
[22] A.B.Kulkarni, S.V.Bonde, U.V.Kulkarni. (2018) "Class-Specific Fuzzy Hypersphere Neural Network."*procedia computer science* **143**:285-294.
[23] A.B.Kulkarni, S.V.Bonde, U.V.Kulkarni. (2018)"Inter-class and Intra-class fuzzy clustering with pruning Algorithm."*International journal of computer sciences and Engineering* **6**:94-99
[24] B. Carse, A.G. Pipe, T. C. Fogarty, T. Hill. (1995) "Evolving radial basis function neural networks using a genetic algorithm." *IEEE International Conference on Evolutionary Computation, Perth*: 300-305.
[25] M. J. Er, W. Chen, S. Wu. (2005) "High-speed face recognition based on discrete cosine transform and RBF neural networks." *IEEE Transactions on neural networks* **16**: 679-691.
[26] M. Rouhani, Javan D. S. (2016) "Two fast and accurate heuristic RBF learning Rules for data classification." *Neural Networks* **75**: 150-161.
[27] Yuanshan Liu, He Huang, Ting wen Huang b, Xusheng Qian. (2016) "An improved maximum spread algorithm with application to complex-valued RBF neural networks." *Neurocomputing* **216**: 261-267
[28] J. Raitoharju, S. Kiranyaz, M. Gabbouj. (2016) "Training radial basis function neural networks for classification via class-specific clustering." *IEEE Transactions on Neural Networks and Learning Systems* **27**: 2458-2471.
[29] Frank A., and Asuncion A. (2010) "UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science." URL: http://archive.ics.uci.edu/ml.
[30] Rehab Ali Ibrahim, Ahmed A. Ewees, Diego Oliva, Mohamed Abd Elaziz, Songfeng Lu. (2018) "Improved salp swarm algorithm based on particle swarm optimization for feature selection." *Journal of Ambient intelligence and Humantized Computing*
[31] Alaa Tharwat (2018) "Classification assessment methods ", *journal of Applied Computing and informatics*