

Pruned Modified Fuzzy Hyperline Segment Neural Network and Its Application to Pattern Classification

S. B. Bagal

Associate Professor,
Dept. of Electronics & Telecomm. Engineering
KCT's LGNSCOE, Nashik, (M. S.), India

U. V. Kulkarni

Professor and Head,
Dept. of Computer Science and Engineering
SGGSIOET, Nanded, (M. S.), India

ABSTRACT

The Fuzzy hyperline segment neural network (FHLSNN) is supervised classifier that forms n-dimensional hyperline segments (HLS) defined by two end points with a corresponding membership function for learning and testing. In this paper, the Pruned fuzzy hyperline segment neural network (PFHLSNN) and Pruned modified fuzzy hyperline segment neural network (PMFHLSNN) are proposed. The pruning method is based on a confidence factor calculated for each hyperline segment in the prediction phase after learning. The new definition of confidence factor is proposed. In PFHLSNN, the hyperline segments with low confidence factor are pruned using user defined threshold to reduce the network complexity. In order to improve the classification performance of PFHLSNN, the modification is proposed in its testing phase and the network is referred as PMFHLSNN. In this modification, the Euclidean distance is computed between the applied input pattern and the centroid of the patterns falling on the hyperline segment to decide the class of pattern. Finally, the HLS with smallest distance is selected as winner and the pattern is so classified that it belongs to the class associated with that HLS. The performance of PFHLSNN and PMFHLSNN is evaluated using benchmark problems and real world handwritten character recognition data set. The results are analyzed, discussed and compared with the FHLSNN. Thus, the proposed approach improved the classification accuracy without affecting the incremental learning of FHLSNN and reduces the network complexity by pruning the hyperline segments of low confidence factor.

General Terms

Pruned modified fuzzy hyperline segment neural network, Pattern Classification.

Keywords

Fuzzy hyperline segment neural network, Pruning, Centroid, Euclidean Distance computation, classification.

1. INTRODUCTION

The fuzzy neural networks (FNN) have been widely used for pattern classification and recognition applications and generally proved to be more powerful than conventional statistical techniques [1]. Patrick K. Simpson proposed supervised fuzzy min-max neural network (FMN) that utilizes fuzzy sets as pattern classes, where each fuzzy set is a union of fuzzy set hyperboxes [2]. He has also proposed unsupervised fuzzy min-max clustering neural network (FMCN), in which clusters are implemented as fuzzy set using membership function with a hyperbox core that is constructed from a min point and a max point [3]. Gabrys and Bargiela have proposed general fuzzy min-max neural (GFMM) for classification and clustering, which is fusion of supervised

and unsupervised learning [4]. In the same sequel, Kulkarni U. V. et al. proposed fuzzy hyperline segment neural network classifier (FHLSNN), which utilizes fuzzy set as pattern classes in which each fuzzy set is a union of fuzzy set hyperline segments. This classifier is applied for rotation invariant handwritten character recognition and found superior than FMN [5]. U.V. Kulkarni et al. have also proposed unsupervised fuzzy hyperline segment clustering neural network and its performance is found superior as compare to FMCN when applied for clustering of Fisher Iris data [6]. P. M. Patil and T. R. Sontakke have proposed general fuzzy hyperline segment neural network (GFHLNN) and applied to rotation, scale and translation invariant handwritten Devanagari numeral character recognition. This proposed approach uses supervised and unsupervised learning and can be used for pure classification, pure clustering and hybrid classification and clustering [7].

However, the use of fuzzy neural networks requires decisions on the part of the user which may affect the accuracy of the resulting classification. One of these decisions concerns the determination of the optimum network structure/size for a particular problem. Unfortunately, usually it is not obvious what size is best; a system that is too small will not be able to learn the data, while one that is just big enough may learn very slowly and very sensitive to initial conditions and learning parameters. Thus, an optimal neural network topology not only reduces the computational complexity, but also improves its generalization capacity [8]. Russel Reed survived various pruning algorithms which train a network that is larger than necessary and then remove the parts that are not needed. In this approach of pruning, the large initial size allows the network to learn reasonably quickly with less sensitivity to initial conditions while the reduced complexity of the pruned system favors improved generalization [9].

The researchers proposed the various pruning methods to reduce the network complexity with acceptable classification accuracy. Rudy Setiono proposed the use of penalty function and the magnitude-based weight elimination criterion for the pruning of feedforward neural network [10]. To reduce the complexity of fuzzy ARTMAP, G. Carpenter and A. Tan proposed a rule-pruning procedure to select a small set of rules from a trained network, based on each rule's confidence factor [11]. S. M. Kamruzzaman and A. H. Hasan proposed pruning algorithm which identifies redundant weights, redundant input and hidden units and removed from the network to yield a simplified network [12]. Anas Quteishat and C. P. Lim proposed confidence factor based pruning procedure to prune the hyperboxes of low confidence factor from the FMN to reduce its complexity [13]. They also proposed two types of modifications in the testing phase of FMN. In the first approach, the Euclidean distance is

computed in the testing phase to decide the class of pattern. In the second modification, they propose to employ both the membership value of the hyperbox fuzzy sets and the Euclidean distance for classification. These modifications improve the performance of classifier, in situations when the large hyperboxes are formed by the network [14].

In this paper, the pruned fuzzy hyperline segment neural network (PFHLSNN) and pruned modified fuzzy hyperline segment neural network (PMFHLNSNN) are proposed. The pruning approach used in this paper is to train the network larger than necessary and then remove the parts (hyperline segments) that are not needed. The objective of this pruning approach is to reduce network complexity with reasonable generalized classification accuracy. This method is based on a confidence factor calculated for each hyperline segment in the prediction phase after learning. The hyperline segments with low confidence factor are pruned using user defined threshold to reduce the network complexity. The confidence factor identifies, hyperline segments that are frequently used and generally give high classification accuracy, and the hyperline segments that are rarely used and, yet highly accurate, and the hyperline segments which are having significant length factor. In order to improve the classification performance of PFHLSNN, the modification is proposed in its testing phase. In this modification, the Euclidean distance is computed between the applied input pattern and centroid of the patterns falling on the hyperline segment, to decide the class of patterns. Finally, the HLS with smallest distance is selected as winner and the pattern is so classified that it belongs to the class associated with that HLS. The performance of PFHLSNN and PMFHLNSNN is evaluated using benchmark problems and real world handwritten character recognition data set. The results are analyzed, discussed and compared with the FHLSNN.

The organization of this paper is as follows. In Section 2, the architecture and learning algorithm of FHLSNN is explained. The proposed confidence factor calculation to prune FHLSNN to get proposed PFHLSNN is explained in Section 3. The proposed modifications of PMFHLNSNN are explained in Section 4. The experimental procedure, simulation result, description of data sets and discussions on the results are presented in Section 5. Finally, the paper is concluded in Section 6.

2. THE FUZZY HYPERLINE SEGMENT NEURAL NETWORK (FHLSNN)

2.1 Topology of FHLSNN

The readers are advised to refer the [5] for the detail description of architecture and algorithm of FHLSNN. To make this article self contained, this section explain the architecture and algorithm of FHLSNN in short. The architecture of FHLSNN consists of four layers as shown in Figure 1. In this architecture first, second, third and fourth layer are denoted as F_R , F_E , F_D and F_C respectively. The F_R layer accepts an input pattern and consists of n processing elements, one for each dimension of the pattern. The F_E layer consists of m processing nodes that are constructed during training. There are two connections from each F_R to each F_E node. Each connection represents an end point for that particular hyperline segment. These end points are stored in

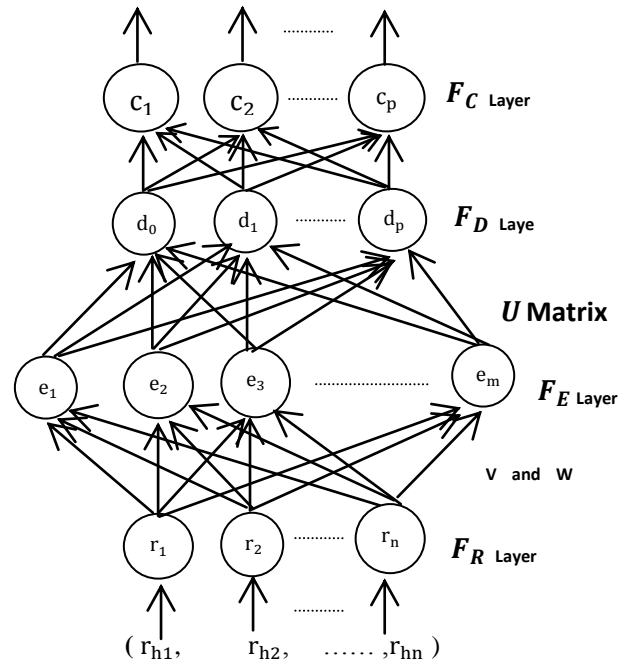


Fig. 1 Fuzzy Hyperline Segment Neural Network

the two matrices V and W . Each F_E node represents hyperline segment fuzzy set and is characterized by the membership function.

Let $R_h = (r_{h1}, r_{h2}, \dots, r_{hn})$ represents the h^{th} input pattern, $V_j = (v_{j1}, v_{j2}, \dots, v_{jn})$ is the one end point of hyperline segment e_j and $W_j = (w_{j1}, w_{j2}, \dots, w_{jn})$ is the other end point of e_j . Then the membership function of the j^{th} F_E node is defined as

$$e_j(R_h, V_j, W_j) = 1 - f(x, \gamma, l), \quad (1)$$

in which $x = l_1 + l_2$, and the distances l_1, l_2 and l are defined as

$$l_1 = \left(\sum_{i=1}^n (w_{ji} - r_{hi})^2 \right)^{1/2}, \quad (2)$$

$$l_2 = \left(\sum_{i=1}^n (v_{ji} - r_{hi})^2 \right)^{1/2}, \quad (3)$$

$$l = \left(\sum_{i=1}^n (w_{ji} - v_{ji})^2 \right)^{1/2}, \quad (4)$$

and $f(\cdot)$ is the three parameter ramp threshold function defined as

$$f(x, \gamma, l) = \begin{cases} 0 & \text{if } x = l, \\ x\gamma & \text{if } 0 \leq x\gamma \leq 1, \\ 1 & \text{if } x\gamma > 1 \end{cases} \quad (5)$$

The F_D layer gives soft decision and output of k^{th} F_D node represents the degree to which the input pattern belongs to the class d_k . The binary weights assigned to the connections between F_E and F_D layers are stored in the matrix U . The values assigned to these connections are defined as

$$u_{jk} = \begin{cases} 1 & \text{if } e_j \text{ is a hyperline segment of the class } d_k \\ 0 & \text{otherwise} \end{cases}$$

$$\text{for } k = 1, 2, \dots, p \text{ and } j = 1, 2, \dots, m. \quad (6)$$

where e_j is the j^{th} F_E node and d_k is the k^{th} F_D node.

The transfer function of each F_D node perform the union of the appropriate (of same class) hyperline segment fuzzy values which is described as

$$d_k = \max(e_j u_{jk}) \text{ for } j = 1 \text{ to } m \text{ and } k = 1 \text{ to } p \quad (7)$$

Each F_C node delivers nonfuzzy output described as

$$c_k = \begin{cases} 0 & \text{if } d_k < T \\ 1 & \text{if } d_k = T \end{cases} \text{ where } T = \max(d_k), \text{ for } k=1 \text{ to } p. \quad (8)$$

2.2 Learning Algorithm of FHLSNN

The supervised FHLSNN learning algorithm for creating HLSs in the hyperspace consists of following steps.

Step 1: Initialization. To initialize HLS start with first pattern in the database, as

$$W_j = V_j = R_h. \quad (9)$$

Step 2: Creation of hyperline segments. The maximum length of HLS is bounded by the parameter θ , where $0 \leq \theta \leq 1$, which is a user defined value and depends on the dimension of feature vector. The extension criterion that has to be met before HLS can extend to include R_h is

$$e_j(R_h, V_j, W_j) \geq 0. \quad (10)$$

Let the set of pattern is R , where $R \in \{R_h \mid h = 1, 2, \dots, P\}$. Given the h^{th} training pair (R_h, d_h) , find all the HLSs belonging to the class d_h . After this following cases are carried out for possible inclusion of the input pattern R_h .

Case 1: By using membership function, find out whether the pattern R_h falls on any one of the exiting HLSs. If R_h falls on any of the HLS then it is included. Therefore, in the training process all the remaining steps are skipped and training is continued with the next training pair.

Case 2: If the input pattern R_h falls on any one of the hyperlines passing through the two end points of HLS, then extend the HLS to include the pattern. Suppose e_j is that hyperline segment with end points V_j and W_j then l_1, l_2 and l are calculated using equation (2), (3), and (4). Subsequently algorithm executes sub-step (i) if $l_1 > l_2$, else the sub-step (ii). Otherwise the Case 3 is considered.

(i) Test whether the point V_j falls on the HLS formed by the points W_j and R_h using equation (1) and if verified then include the pattern by extending e_j as

$$V_j^{\text{new}} = R_h \text{ and } W_j^{\text{new}} = W_j \quad (11)$$

(ii) Test whether the point W_j falls on the hyperline segment formed by the points V_j and R_h and if verified, then include the pattern by extending e_j as

$$W_j^{\text{new}} = R_h \text{ and } V_j^{\text{new}} = V_j. \quad (12)$$

Case 3: If HLS is a point i.e. $W_j = V_j$, then extend it to include the pattern R_h , if extension criteria is satisfied as described by equation (11).

Case 4: If the pattern R_h is not included by any of the HLSs then create a new HLS as

$$V_j^{\text{new}} = W_j^{\text{new}} = R_h. \quad (13)$$

Step 3: Intersection test. The learning algorithm allows intersection of HLSs from the same class and eliminates the intersection between HLSs from separate classes. Intersection

test is carried out as soon as the HLS is either extended by Case 2, Case 3 or created in Case 4.

Let $W_{1st} = [x_1, x_2, \dots, x_n]$, and $V_{1st} = [y_1, y_2, \dots, y_n]$ represent two end points of the extended or created HLS and $W_n = [x'_1, x'_2, \dots, x'_n]$, $V_n = [y'_1, y'_2, \dots, y'_n]$ are the end points of the HLS of other class. The equation of hyperline passing through W_{1st} and V_{1st} is

$$\left[\frac{a_i - x_i}{y_i - x_i} \right] = r_1 \text{ for } i = 1, 2, \dots, n. \quad (14)$$

and the equation of the hyperline passing through W_n and V_n is

$$\left[\frac{b_i - x_i}{y_i - x_i} \right] = r_2 \text{ for } i = 1, 2, \dots, n. \quad (15)$$

where r_1, r_2 are the constants and a_i, b_i are the variables. The equations (14) and (15) leads to set of n simultaneous equations which are described as

$$r_1(y_i - x_i) + x_i = r_2(y_i - x_i) + x_i \quad (16)$$

For $i = 1, 2, \dots, n$.

The values of r_1 and r_2 can be calculated by solving any two simultaneous equations. If remaining $n-2$ equations are satisfied with the calculated values of r_1 and r_2 then two hyperlines are intersecting and the points of intersection P_t is

$$P_t = (r_1(y_1 - x_1) + x_1, \dots, r_1(y_n - x_n) + x_n) \quad (17)$$

The point of intersection P_t , if falls on both hyperlines segments then these HLSs are also intersect. This can be verified by the equation (1) and eliminated by contraction of appropriate HLS.

Step 4: Removing intersection. Depending on the cases, if extension of HLS produces an intersection then it is removed by restoring the end point V_j as $V_j^{\text{new}} = V_j^{\text{old}}$, and point W_j is restored as, $W_j^{\text{new}} = W_j^{\text{old}}$. Create a new HLS to include R_h as in equation (13).

If Case 4 creates intersection then it is removed by restoring the end points of previous HLS of other class as

$$W_{\text{new}+1} = V_{\text{new}+1} = V_n \text{ and } V_n = W_n. \quad (18)$$

3. PRUNED FUZZY HYPERLINE SEGMENT NEURAL NETWORK (PFHLSNN)

After the learning of FHLSNN, a pruning procedure, based on a confidence factor is incorporated to reduce the number of hyperline segments (HLS). The pruning approach used in this paper is to train the network larger than necessary and then remove the parts (hyperline segments) that are not needed. The objective of this pruning approach is to reduce network complexity with reasonable generalized classification accuracy. This method is based on a confidence factor calculated for each hyperline segment in the prediction phase after learning. The hyperline segments with low confidence factor are pruned using user defined threshold to reduce the number of hyperline segments and hence the network complexity.

A data set is divided into three subsets: training set (for learning), prediction set (for pruning), and testing set (for performance evaluation). The definition of confidence factor calculation by Carpenter and Tan [11] is modified by adding the effect of length of hyperline segment. Thus the confidence factor for each HLS created during learning is calculated

based on its usage frequency, predictive accuracy and length factor on the prediction data set, using equation (19), as below:

$$CF_j = [(1 - \alpha) - \gamma] U_j + \gamma A_j + \alpha L_j \quad (19)$$

Where, $\gamma = \gamma (1 - \alpha)$ and U_j is the usage of HLS j , A_j is the accuracy of HLS j , L_j is the length factor of HLS j and $\alpha, \gamma \in [0, 1]$ are the weighing factors.

The value of U_j is defined as the number of patterns in the prediction set classified by any HLS j , divided by the maximum number of patterns in the prediction set classified by any HLS with the same classification class. On the other hand, the value of A_j is defined as the number of correctly predicted set of patterns classified by any HLS j , divided by the maximum correctly classified patterns with the same classification class. The length factor L_j is defined as the ratio of length of any HLS j to the maximum length of any HLS with same classification class. Thus, the confidence factor identifies, hyperline segments that are frequently used and generally give high classification accuracy, and the hyperline segments that are rarely used and, yet highly accurate and the hyperline segment which are having the significant length.

After learning of FHLSNN, the confidence factor for each HLS created during learning is calculated by using equation (19). Then the hyperline segments with a confidence factor lower than a user defined threshold are pruned and remaining selected hyperline segments are used in the testing phase for the performance evaluation of the network. Thus the proposed pruning method reduces the HLSs and the network complexity without affecting the incremental learning of FHLSNN.

In the testing phase, the PFHLSNN classifies the applied pattern based on the membership function value calculated using equation (1). When the pattern is applied to the classifier for testing, it calculates the membership value for all the selected hyperline segments after pruning. The applied pattern is classified to the class associated with the HLS that gives the highest membership value for this pattern.

4. PRUNED MODIFIED FUZZY HYPERLINE SEGMENT NEURAL NETWORK (PMFHLSSN)

The PFHLSNN classifies the applied pattern based on the membership function value. When the pattern is applied to the classifier for testing, it calculates the membership value for all the selected hyperline segments after pruning. The applied pattern is classified to the class associated with the HLS that gives the highest membership value for this pattern.

In order to improve the classification performance of PFHLSNN, the modification is proposed in its testing phase. In this modification, the Euclidean distance is computed between the applied input pattern and centroid of the patterns falling on the hyperline segment, to decide the class of patterns. In this method, instead of calculating the membership value, the centroid of patterns falling on the each HLS is computed using equation (20), as below.

$$C_{ji} = C'_{ji} + \frac{|R_{hi} - C_{ji}|}{N_j} \quad (20)$$

where, with reference to h^{th} input pattern, C'_{ji} is the centroid of the j^{th} HLS in the i^{th} dimension, C_{ji} is the centroid of the

patterns falling on the j^{th} HLS in the i^{th} dimension, and N_j is the number of patterns falling on the j^{th} HLS.

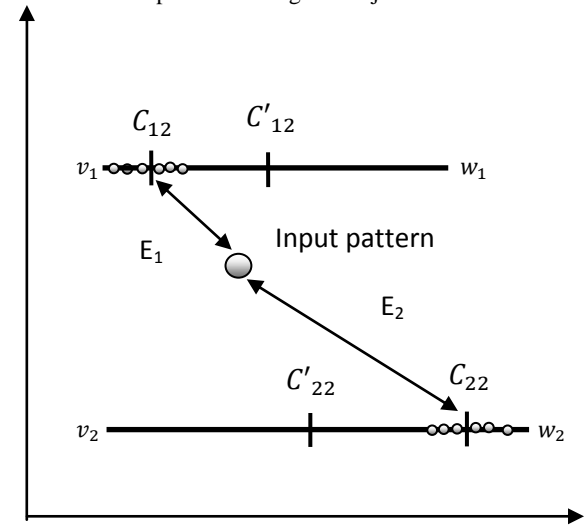


Fig. 2 The classification process of PMFHLSSN

Then the Euclidean distance [15], between the centroid of patterns falling on the j^{th} HLS in the i^{th} dimension and the applied input pattern is calculated using equation (21),

$$E_{jh} = \sqrt{\sum_{i=1}^n (C_{ji} - R_{hi})^2} \quad (21)$$

where, E_{jh} is the Euclidean distance between the centroid of patterns falling on the j^{th} HLS in the i^{th} dimension and the applied h^{th} input pattern.

Finally, the HLS with the smallest Euclidean distance is selected as winner and the pattern is so classified that it belongs to the class associated with that HLS.

This proposed classification process for a two dimensional input pattern is demonstrated in the Figure 2. Figure shows the two hyperline segments of class 1 and class 2 with centroids of hyperline segment as C'_{12} , C'_{22} and the centroids of the patterns falling on the HLS as C_{12} , C_{22} respectively. Let E_1 is the Euclidean distance between the input pattern and centroids C_{12} of first HLS. Similarly, E_2 is the Euclidean distance between the input pattern and centroid C_{22} of second HLS. As the distance E_1 is the smaller than distance E_2 , the HLS 1 is selected as winner and the pattern is so classified that it belongs to the class associated with that HLS 1. Therefore applied input pattern is classified as class 1.

Thus the proposed modification in the testing phase improves the classification accuracy of PFHLSNN, without affecting the incremental learning process of FHLSNN.

5. EXPERIMENTS AND RESULTS

5.1 Benchmark Problem

This proposed approaches are implemented using MATLAB R2013a and ran on Intel core i3 2328M, 2.2GHz PC. To explore the different capabilities of a pattern classifier, its performance was evaluated using five benchmark data sets selected from the UCI machine learning repository [16] and the real handwritten character database. The five benchmark data sets are Glass data set, Wine data set, Iris data set, PID dataset and Sonar data set. From the total available patterns, training data set consists of approximately 80 % patterns with equal proportion of all class and remaining 20 % patterns with

equal proportion of all class are used as testing data set. As PFHLSNN and PMFHLSNN required a prediction data set for pruning, 50 % of the training data samples were taken to form the prediction data set. A description of each data set is as follows.

1) The Glass data set: This data set contains 214 samples, each with nine continuous features, from six classes. 166 patterns are randomly selected with equal proportion of all six classes and given for training. Remaining 48 patterns with equal proportion of all six classes are given for testing.

2) The Wine data set: This data set is another example of multiple classes with continuous features. This data set contains 178 samples, each with 13 continuous features from three classes. 136 patterns are randomly selected with equal proportion of all classes and given for training. Remaining 42 patterns with equal proportion of all classes are given for testing.

3) The Iris data set: This data set contains 150 samples, each with four continuous features (sepal length, sepal width, petal length, and petal width), from three classes (Iris setosa, Iris versicolor, and Iris virginica). This data set is an example of a small data set with a small number of features. One class is linearly separable from the other two classes, but the other two classes are not linearly separable from each other. 120 patterns are randomly selected with equal proportion of all classes and given for training. Remaining 30 patterns with equal proportion of all classes are given for testing.

4) The PID data set: This data set consists of 768 cases with eight features from two classes (diabetic and healthy). A total of 268 cases (35%) are from patients diagnosed as diabetic and the remaining as healthy. The PID data samples overlap each other, making it a challenging classification problem. 576 patterns are randomly selected with equal proportion of

both classes and given for training. Remaining 192 patterns with equal proportion of both classes are given for testing.

5) The Sonar data set: This is a high-dimensional data set that contains 208 samples, each with 60 input features (s1,s2,s60). The data set contains 111 and 97 samples from two classes, i.e., sonar signals from mine (metal cylinders) (class 1) and rocks (class 2), respectively. This data set is a high-dimensional problem and is useful for evaluating the scalability capability of a pattern classifier. 156 patterns are randomly selected with equal proportion of two classes and given for training. Remaining 52 patterns with equal proportion of two classes are given for testing.

5.2 Handwritten Character Data Set:

This database consists of consists of 1000 Devanagari numeral character. Ten numerals from one hundred writers are scanned and stored in BMP format. After moment normalization [17], the rotation invariant ring-data features defined by Ueda and Nakamura [18] and extended by Chiu and Tseng [19], are extracted from the character by setting ring width to two. The extracted ring-data vector is a 16-dimensional feature vector. 800 patterns are randomly selected with equal proportion of ten classes and given for training. Remaining 200 patterns with equal proportion of ten classes are given for testing. As mentioned earlier, 50 % of training samples were taken to form prediction data set for PFHLSNN and PMFHLSNN.

The experimentation is done with different values of θ and the weighing parameters α and γ are selected and adjusted in the range of 0 to 1 to get higher recognition rate. The user defined threshold to prune the hyperline segments of low confidence factor is adjusted in such way that after pruning the remaining hyperline segments should of all classes.

Table 1. Recognition rate (%) for Glass Data Set

Theta θ	FHLSNN		PFHLSNN		PMFHLSNN	
	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)
0.03	89	72.91	13	47.91	13	58.33
0.032	90	72.91	14	50	14	62.5
0.034	88	72.91	12	54.16	12	56.25
0.036	87	72.91	10	54.16	9	45.83
0.072	84	72.91	9	52.08	9	22.91

Table 2. Recognition rate (%) for Wine Data Set

Theta θ	FHLSNN		PFHLSNN		PMFHLSNN	
	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)
0.034	67	64.28	7	66.66	7	85.71
0.044	65	64.28	3	66.66	3	85.71
0.072	62	61.90	4	71.42	4	85.71
0.082	60	61.90	4	71.42	4	85.71
0.09	59	61.90	5	64.28	5	80.95

Table 3. Recognition rate (%) for Iris Data Set

Theta θ	FHLSNN		PFHLSNN		PMFHLSNN	
	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)
0.03	98	96.66	4	70	4	66.66
0.044	80	96.66	8	100	8	100
0.048	79	96.66	9	100	9	100
0.056	73	96.66	10	100	10	100
0.062	68	96.66	11	100	11	100

Table 4. Recognition rate (%) for PID Data Set

Theta θ	FHLSNN		PFHLSNN		PMFHLSNN	
	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)
0.03	350	71.87	16	53.12	16	55.72
0.038	323	71.87	16	62.5	16	66.14
0.046	307	71.87	21	61.97	21	66.14
0.076	295	71.87	6	66.14	6	67.18
0.5	288	71.87	3	48.43	3	53.64

Table 5. Recognition rate (%) for Sonar Data Set

Theta θ	FHLSNN		PFHLSNN		PMFHLSNN	
	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)
0.5	138	30.76	3	13.46	3	65.38
0.55	136	30.76	6	15.38	6	59.61
0.6	133	30.76	8	13.46	8	55.76
0.65	131	30.76	6	13.46	6	65.38
0.7	128	30.76	6	13.46	6	48.07

The Table 1 to 5 shows the % recognition rate and number of hyperline segments with different values of theta for the five selected benchmark problems using FHLSNN, PFHLSNN and PMFHLSNN classifier. As shown in the Table 1, for the Glass data set the PFHLSNN can give 50 % recognition rate with only 14 hyperline segments as compare to 90 hyperline segments of original FHLSNN. This classification accuracy can be improved to 62.5 % using PMFHLSNN. As shown in the Table 2, for the Wine data set, the PFHLSNN can give 71.42 % recognition rate with only 4 hyperline segments as compare to 62 hyperline segments of original FHLSNN. This classification accuracy can be improved to 85.71 % using PMFHLSNN. Thus the pruned network can give more recognition rate than the original FHLSNN, because the proposed pruning approach prunes the hyperline segments which were responsible for misclassification. As shown in the Table 3, for the well known Iris data set the PFHLSNN and PMFHLSNN can give 100 % recognition rate with only 8 hyperline segments as compare to 80 hyperline segments of

original FHLSNN. As shown in the Table 4, for the PID data set the PFHLSNN can give 66.14 % recognition rate with only 6 hyperline segments as compare to 295 hyperline segments of original FHLSNN. This classification accuracy can be improved to 67.18 % using PMFHLSNN. As shown in the Table 5, for the Sonar data set the PFHLSNN gives comparatively less recognition rate of 15.38 % with only 6 hyperline segments as compare to 136 hyperline segments of original FHLSNN. This classification accuracy can be improved to remarkable value of 59.61 % using PMFHLSNN which is far more than the FHLSNN.

The Table 6 shows the % recognition rate and number of hyperline segments with different values of theta for real handwritten character recognition dataset using FHLSNN, PFHLSNN and PMFHLSNN classifier. As shown in the Table the PFHLSNN can give 22 % recognition rate with only 15 hyperline segments as compare to 400 hyperline segments of original FHLSNN. This classification accuracy can be improved to 23 % using PMFHLSNN.

Table 6. Recognition rate (%) for Handwritten Data Set

Theta θ	FHLSNN		PFHLSNN		PMFHLSNN	
	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)
0.5	402	41	19	21	19	21.5
0.65	400	41	15	22	15	23
0.7	400	41	16	22	16	23
0.75	400	41	15	22	15	23
0.8	400	41	15	22	15	23

Table 7. Recognition rate (%) for different Classification systems using different Data Set

Methods	Recognition Rate (%) for the various Data Set				
	Glass Dataset	Wine Dataset	Iris Dataset	PID Dataset	Sonar Dataset
C4.5	70.23	91.09	91.60	71.02	-
C4.5 Rules	67.96	91.90	91.58	71.55	-
ITI	67.49	91.09	91.25	73.16	-
LMDT	60.59	95.40	95.45	73.51	-
CN2	70.23	91.09	91.92	72.19	-
LVQ	60.69	68.90	92.55	71.28	-
OCI	57.72	87.31	93.89	50.00	-
Nevprop	44.08	95.41	90.34	68.52	-
FMN	69.07	96.85	95.60	68.42	81.20
FHLSNN	72.91	64.28	96.66	71.87	30.76
PFHLSNN	50.00	66.66	100	66.14	13.46
PMFHLSNN	62.5	85.71	100	67.18	65.38

The Table 7 compares the results of FHLSNN, PFHLSNN, and PMFHLSNN with other classification systems published in [20], and [21]. The proposed PFHLSNN and PMFHLSNN shows highest classification performance for Iris dataset and comparable classification performance for all other selected benchmark datasets with a very less number of HLS. Thus, the experimental results summarized in all the tables shows that the proposed pruning approach (PFHLSNN) gives comparable classification performance for all the selected benchmark problems and real world handwritten character recognition dataset with small number of hyperline segment. This implied that a lot of hyperline segments in FHLSNN could be removed without adversely affecting its performance. Even for some benchmark problems, the proposed pruning approach shows higher recognition rate than the original FHLSNN. This implied that the hyperline segments which were responsible for misclassification are removed in PFHLSNN and PMFHLSNN to achieve higher recognition rate. The proposed modification (PMFHLSNN) in the testing phase of PFHLSNN improves the classification performance for all the selected benchmark problems and real world handwritten character recognition dataset. Thus this proposed pruning method reduces the HLSs and the network complexity without affecting the incremental learning of FHLSNN.

6. CONCLUSIONS

In this paper, the pruned fuzzy hyperline segment neural network (PFHLSNN) and pruned modified fuzzy hyperline segment neural network (PMFHLSNN) are proposed. The pruning method is based on a confidence factor calculated for each hyperline segment in the prediction phase after learning. The hyperline segments with low confidence factor are pruned using user defined threshold to reduce the network complexity. In order to improve the classification performance of PFHLSNN, the modification is proposed in its testing phase. In this modification, the Euclidean distance is computed between the applied input pattern and centroid of the patterns falling on the hyperline segment, to decide the class of pattern. The performance of PFHLSNN and PMFHLSNN is evaluated using benchmark problems and real world handwritten character recognition dataset. The experimental results shows that the proposed pruning approach (PFHLSNN) give comparable classification performance for all the selected benchmark problems and real world handwritten character recognition dataset, with small number of hyperline segments. This implied that a lot of hyperline segments in FHLSNN could be removed to reduce the complexity without adversely affecting its performance. Even for some benchmark problems, the proposed pruning approach shows higher recognition rate than the original FHLSNN. This implied that the hyperline segments which were responsible for misclassification are removed in

PFHLSNN and PMFHLSNN to achieve higher recognition rate. Thus this proposed pruning method reduces the HLSs and the network complexity without affecting the incremental learning of FHLSNN.

7. REFERENCES

- [1] J. J. Buckley and Y. Hayashi, "Fuzzy neural network: a survey," *Journal of Fuzzy Sets and Systems*, vol.66, Issue 1, pp. 1-13, Aug. 1994.
- [2] P. K. Simpson, "Fuzzy min-max neural networks—Part 1: Classification," *Journal of IEEE Trans. Neural Networks.*, vol. 3, no. 5, pp. 776–786, Sep. 1992.
- [3] P. K. Simpson, "Fuzzy min-max neural networks—Part 2: Clustering," *IEEE Trans. Fuzzy systems*, vol. 1, no. 1, pp. 32–45, Feb. 1993.
- [4] B. Gabrys and A. Bargiela, "General Fuzzy Min-Max Neural Network for Clustering and Classification," *IEEE Trans. neural networks*, vol.11, pp. 769-783, May 2000.
- [5] U. V. Kulkarni, T. R. Sontakke, and G. D. Randale, 2001. Fuzzy hyperline segment neural for rotation invariant handwritten character recognition. In *Proceeding of the Int. joint conf. on neural networks: IJCNN'01* held in Washington DC, USA, pp. 2918–2923.
- [6] U. V. Kulkarni, T. R. Sontakke and A. B. Kulkarni, "Fuzzy Hyperline Segment Clustering Neural Network," In *IEEE Electronics letters* vol. 37, no. 05, pp. 301-303, March 2001.
- [7] P. M. Patil and T. R. Sontakke, 2006. "Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy Neural Network" *Journal of Pattern Recognition*, vol. 40, pp. 2110–2117, 2007.
- [8] Sietsma J. and Dow R. J. F., 1988. Neural Net Pruning-why and how. In *Proceeding of Int. Joint Conf. on Neural Networks* held in San Diego, C. A. USA., Vol. 1, pp. 325-333.
- [9] R. Reed, "Pruning algorithms-A survey," *Journal of IEEE Trans. Neural Networks*, vol. 4, pp. 740-747, 1993.
- [10] R. Setiono, "A penalty function approach for pruning Feedforward neural networks", *Journal of Neural Computation*, vol. 9, no. 1, pp. 185-204, 1997.
- [11] G. Carpenter and A. Tan, "Rule extraction: From neural architecture to symbolic representation," *Journal of Connection Sci.*, vol. 7, no. 1, pp. 3–27, 1995.
- [12] S. M. Kamruzzaman and Ahmed Ryadh Hasan, 2005. Pattern classification using simplified neural networks with pruning algorithm. In *Proceeding of Int. Joint Conf. on Information and Communication Technology in Management, Multimedia University, Malaysia*.
- [13] Anas Quteishat and Chee Peng Lim, 2008. Application of the fuzzy Min-Max neural networks to medical diagnosis. In *Proceeding of 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems , KES2008, Zagreb, Croatia, Part III*, pp 548-555.
- [14] Anas M. Quteishat and Chee Peng Lim, "A Modified fuzzy min-max neural network and its application to fault classification," *Journal of Soft computing in industrial applications, ASC*, vol. 39, pp.179-188, 2007.
- [15] Johnson, R.A., and D.W. Wichern, 1998. *Applied multivariate Statistical Analysis*. New Jersey: Prentice Hall. pp. 226-235.
- [16] P. M. Murphy and D. W. Aha, 1995. *UCI Repository of Machine Learning Databases, (Machine-Readable Data Repository)*. Irvine, CA: Dept. Inf. Computer Sci., Univ. California.
- [17] Perantonis S. J. and P.J.G. Lisboa, "Translation, rotation and scale invariant pattern recognition by high-order neural networks and moment classifiers," *Journal of IEEE Trans. Neural networks*, Vol. 3, No. 2, 1992, pp. 241-251.
- [18] Udea K. and Y. Nakamura, 1984. Automatic verification of sealimpression pattern. In *Proceeding of 9th Int. Conf. on pattern recognition*. Vol. 2, pp. 1019-1021.
- [19] Hung-Pin Chiu and Din-Chang Tseng, "Invariant handwritten Chinese character recognition using fuzzy min-max neural networks," *Pattern Recog. Letters*, Vol. 18, 1997, pp. 481-491.
- [20] A. Hoang, 1997, Supervised classifier performance on the UCI data set, in *Department of Computer Science, vol. M. Sc., University of Adelaide, Australia*.
- [21] Anas Q., C. P. Lim, and K. S. Tan, "A modified fuzzy min-max neural network with a genetic-algorithm-based rule extractor for pattern classification". *IEEE Trans. Systems, Man, and Cybernetics—part a: systems and humans*, vol.40, no.3, May 2010.