Ensemble Neural Network Classifier Design using Differential Evolution

Shobha T[*], R J Anandhi[**]

[*] Department of Computer Science and Engineering, The Oxford College of Engineering
[**] Department of Information Science and Engineering, New Horizon College of Engineering

Abstract- In this paper, ensemble neural network has been designed for the classification purpose using the Differential Evolution (DEEANN). The ensemble structure has been designed into two different stages of classifier, in the first stage four multilayer perceptron classifier have been applied which has same size of training data but some new data also has been included to embed some new information in the process of learning. All the four classifiers have been trained through the gradient descent algorithm. In the second stage, there is decision neural network, which considers the output of all the four first stage classifiers as inputs and develop the final decision with the help of differential evolution. Two other possibilities of integrating the classifier outputs is based on majority voting method and mean decision value method have also been considered to compare the performances. The proposed method has shown not only the high efficiency but also resistivity with trial variation.

Index Terms- Classification, Differential evolution, Ensemble architecture, Feed-forward neural network, Neural network

I. INTRODUCTION

Machine learning refers to a system that has the capability to automatically learn knowledge from experience and other ways. In machine learning classification is one of the prime important mechanism available, where Classifier predicts categorical labels that exist with applied input data to take the necessary action or place the data in the different, already existed groups. In many situations, where single classifier is used, face challenge of not having generalized knowledge of classification, ensemble approach in which many neural networks are integrated in proper manner can increase the reliability of classification decision. The main causes of problem with single classifier may be limited size of data, inappropriate learning, and noisy data etc. The major challenges with ensemble neural network are structure formation and determination of role of each classifier involved. These two properties have significant effect over ensemble network generalization capability. The most powerful way to design the classifier is to use artificial neural network, which has better ability of learning and handling the nonlinearity in optimum manner.

Computing system which is inspired by biological nervous systems are defined as Artificial Neural Network (ANN). ANN consists of large number of processing elements called neurons. Neurons of network work parallel to learn from the input data, to organize internal processing, and to obtain optimized output. It is the most widely used algorithm for solving real-world computational applications. The challenges faced by ANN are choosing appropriate initial value of connection weights, selecting number of hidden nodes, convergence of learning algorithm and training error. To overcome these problems, evolutionary algorithms (EA) have been used in recent days to evolve ANN for obtaining better and efficient network performance. EA can help in choosing best connection weights, reduced number of hidden nodes and better convergence.

In this paper feed forward architecture has been considered to design the classifiers and its ensemble has been created with differential evolution. Each classifier has obtained the learning through the gradient descent method. Diversity within individual classifier has been created by providing some variation over training data set. A next stage classifier which is again a feed forward architecture, weights has been evolved through the stochastic search method. The proposed learning through differential evolution has been applied over benchmark XOR classification problem to analyze the training capability and then later ensemble network has been developed to define the classification over other data sets.

II. RELATED WORK

Based on Genetic algorithm a ensemble approach has been presented in [1]. An ensemble neural network based on the Akaike information criterion (AIC), has been proposed in [2]. Proposed algorithm first searches for best configuration weights of each component and then it is used as an automating tool to find the best configuration weights of ensemble neural network. An ENN in [3], uses entropy theory to combine the components of network. Using ENN, the best structure of each component network is searched first, and then employed to determine the best combining weights of ensemble NN. To forecast complex time series, authors of [4] have designed ensemble neural network using hybrid method based on particle swarm optimization with fuzzy aggregation of responses. Based on feature selection, a multi-sided multi-granular neural network ensemble optimization method has been presented in [5], using different attribute granularity and the corresponding subsets, proposed algorithm divides attribute granularity of dataset from multi-side, and structures multi-granular individual neural networks. Zhiye et al., have proposed ensemble neural network framework for dependency parsing of natural sentences. Ensemble models using a convolutional neural network has been discussed in [7]. An ensemble of models, each of which is optimized for a limited variety of poses, is capable of modeling a large variety of human body configurations. An ensemble of over complete patch-based neural networks has been presented in [8] which segments accurate quantification of white matter hyper intensities (WMH) from Magnetic Resonance Imaging. Due to over complete nature of proposed algorithm, accurate and regular segmentations have been obtained, and by using a boosted ensemble of neural networks the segmentation error has been minimized. An ensemble DNN (EDNN) algorithm using Deep neural network and its applicability to metabolomics studies has been presented in [9]. To predict PPIs based on different representations of amino acid sequences, an EnsDNN (Ensemble Deep Neural Networks) algorithm has been proposed in [10]. In article [11], to identify and classify four types of blur images, an ensemble convolution neural network (CNN) has been designed. To enhance model discriminability without incurring additional computing burden, a two-stage pipeline, comprised of deep compression and ensemble technique has been used. In [12], Computational Modification Sites with Ensemble Neural Network (CMSENN), has been proposed to detect protein modification.

III. PROPOSED WORK

The proposed work has two stages: (i) In the first stage, number of individual feed-forward architecture has been considered for classifier. Four same size architecture have been considered which share some overlapping in the training data of one classifier with other. Gradient descent has been applied to train the individual classifier as shown in the Fig.1. (ii) In the second stage a feed forward architecture (called integrated classifier) shown in Fig.2 takes the output of each 1st stage classifier as the input and get the training over complete data set. In this stage, integrated classifier has learnt how to assign the weightage to output delivered by each classifier to decide on final decision. The learning is given through the differential evolution algorithm which has high level of exploration capability. Fig.3. depicts architecture of evolving learning with Differential Evolution. The weight of neural network is first transformed into one dimensional representation and the pre-processing has been applied as linear normalization as defined by Eq. (1), where x_{mn} and x_{mx} are the minimum and maximum value of the particular attribute. This pre-processed input evolve by differential evolution as given in next section.

$$x_n = \frac{x - x_{mn}}{x_{mx} - x_{mn}} \quad \text{Eq. (1)}$$

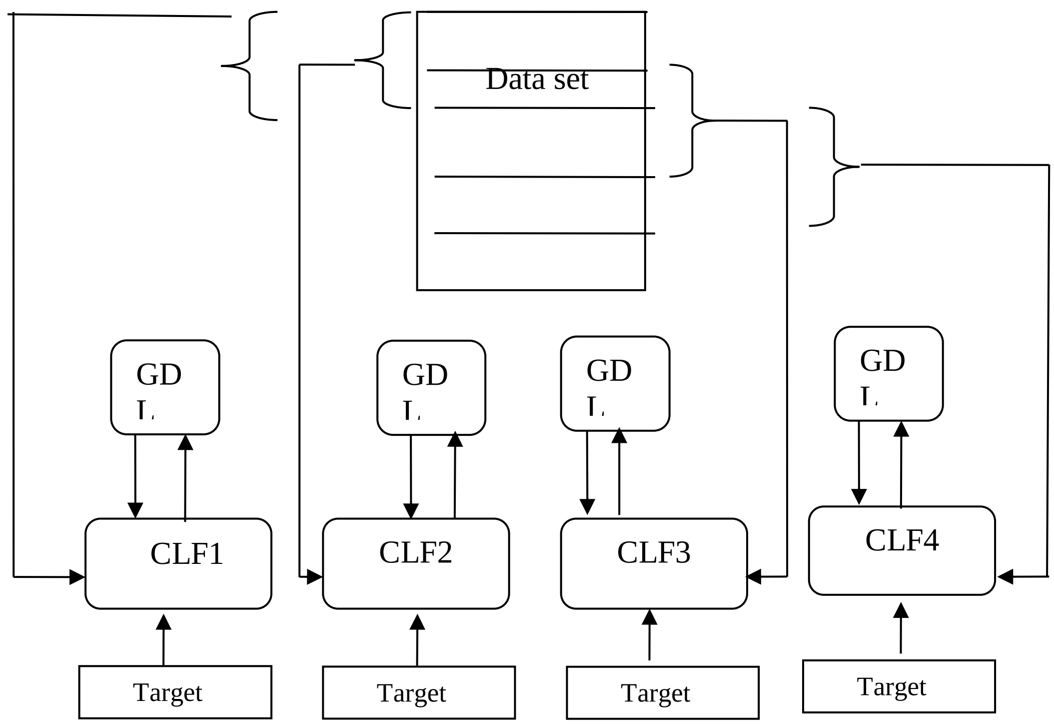


Fig.1: First stage training phase

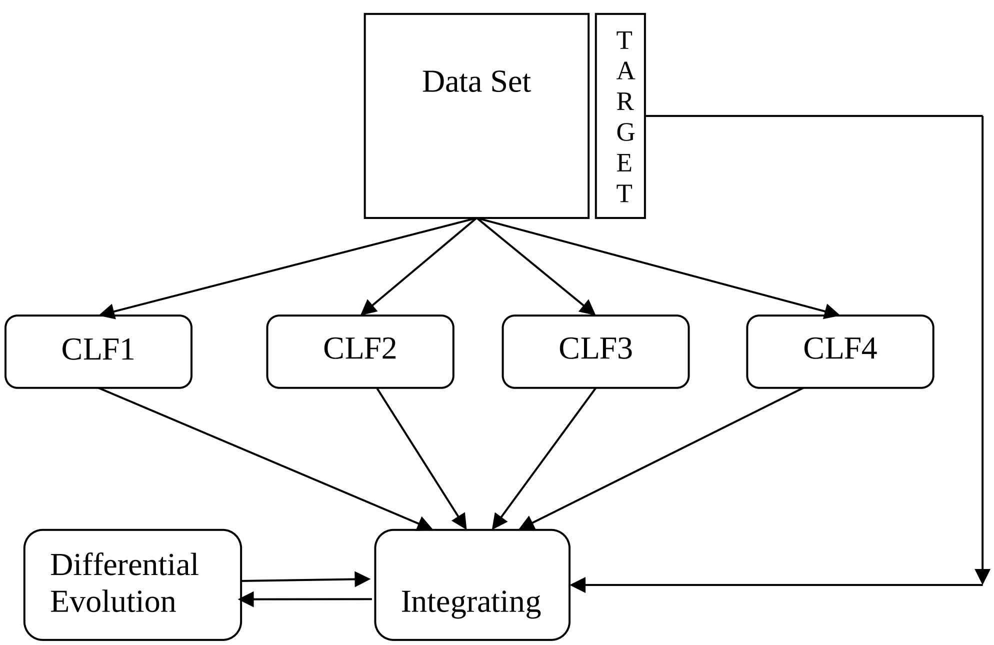


Fig.2: Functional module for Second stage development

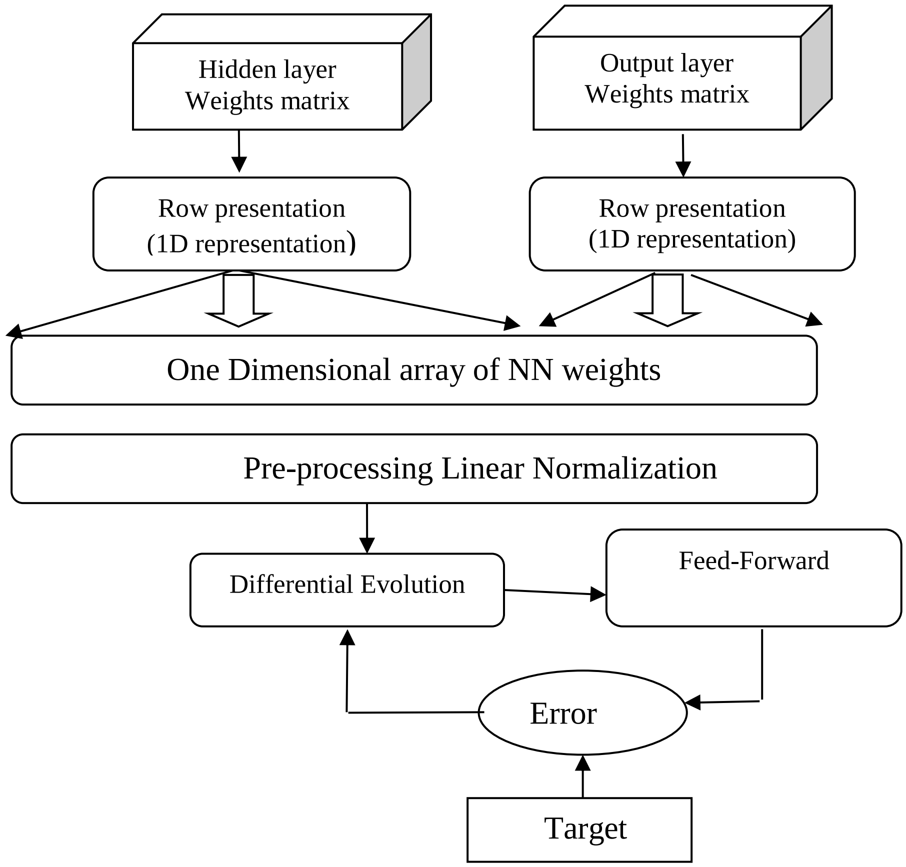


Fig.3. Evolving Learning with Differential Evolution

A. *Differential Evolution*

Differential evolution (DE) is one of the most powerful stochastic real-parameter optimization algorithms used to solve global optimization problems. It has 3 basic operations: mutation, crossover, and selection. During each generation, mutated vector is generated using linear combination of a base vector and one or more differential vector as defined in Eq. (2). Trial vectors have been developed by applying crossover operator under probabilistic environment as shown in Eq. (3). CR is a controlling parameter within the range [0,1]. jrand is a randomly selected index to make sure that at least 1-D of the mutated vector will enter into the newly generated individual. Then the selection process selects between target and trail vectors to choose vectors for the next generation according to Eq. (4).

$$V_i^{(G)} = X_{r1}^{(G)} + F * \left(X_{r2}^{(G)} - X_{r3}^{(G)}\right) \quad Eq.(2)$$

$$u_{\mathrm{ij}}^{(G)} = \begin{cases} v_{ij}^{(G)} & if\ rand(0,1) \le CR\ or\ j = j_{rand} \\ x_{ij}^{(G)} & otherwise \end{cases} \quad Eq.(3)$$

$$x_{\mathrm{ij}}^{(G)} = \begin{cases} u_i^{(G)} & if\ f\left(u_i^{(G)}\right) \le f\left(x_i^{(G)}\right) \\ x_i^{(G)} & otherwise \end{cases} \quad Eq.(4)$$

B. Learning algorithm with Gradient Decent

Weight update equations for Back Propagation, the rate of the updates are proportional to the derivative of the nonlinear activation functions. A typical activation function for neurons in multilayer perceptron neural network is of sigmoid type with bell-shaped derivatives. The change in hidden and output layer weights have been shown below. To make the learning faster, momentum has also been included as shown in Eq.7.

1. Initialize the weights in the network according to Gaussian distribution random number initialization process.
2. From data set, the set of training data, derive the network response.
3. Compare the preferred network responses with the definite output of the network and the local error is calculated according to

   For output layer: $\delta_i^s = \left(d_q - x_{out,i}^s\right) g(u_i^s)$ Eq. (5)

   For hidden layer: $\delta_i^s = \sum_{h=1}^{n2} \delta_h^{s+1} w_{hi}^{s+1}\, g(u_i^s)$ Eq. (6)

4. The weights of the network can be updated as

   $$w_{ij}^s(t+1) = w_{ij}^s(t) + \mu \delta_i^s x_{out,j}^s + \alpha\left[w_{ij}^s(t) - w_{ij}^s(t-1)\right] \qquad \text{Eq. (7)}$$

5. Stop the iteration if network converged, else go back to step 2.

IV. EXPERIMENTAL RESULTS & ANALYSIS

Proposed algorithms have been implemented using MATLAB. To understand the proposed form of learning of neural network through differential evolution, a benchmark problem of classification XOR has been considered. A proper classification of XOR problem will ensure the classification capability of developed algorithm. A feed forward architecture having size [2 3 1] has been selected and weights have been evolved for number of iterations with population size 100 until mean square error is greater than 0.007. The obtained result for hidden layer weights and output layer weights have been shown in Table1 and in Table2 respectively. The learning error with iteration for best solution and whole population mean error have been shown in Fig.4. It can be observed that there is nearly close chase by population mean error which indicate nearly complete population is having similar type of convergence. The final obtained output by neural network has been shown in Table3 and it is observed that output is very close to the target.

Table1: Hidden layer weights obtained after learning completion

Input node	1st Hidden node	2nd Hidden node	3rd Hidden node
1	-4.3460	6.1492	12.0641
2	15.4912	9.5074	-5.7446

Table2: Output layer weights obtained after learning completion

Hidden node	Out put node
1	-17.5399
2	22.6341
3	-16.2271

Table3: Final obtained output by DE-ANN

Input Data		Target	DE-ANN O/P
0	0	0	0.0038
0	1	1	0.9936
1	0	1	0.9978
1	1	0	0.0000

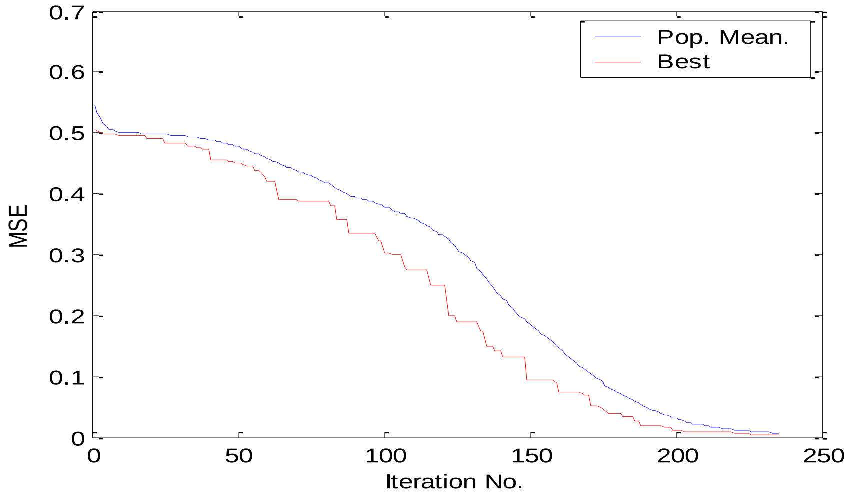


Fig.4: Learning error convergence characteristics

To get the practical benefit of proposed ensemble classifier," Heart-Disease" data set of UCI repository has been considered. It has total 270 data set and each data having 13 attributes and have two classes. In development of training data set for each classifier, 100 data set have been considered while remaining data set has been considered for the testing. (1st CLF: 1-100; 2nd CLF: 51-150; 3rd CLF: 101-200; 4th CLF: 151-250 data samples have been considered for training). The learning rate and momentum constant 0.2 and 0.1 respectively has been considered. The learning of each and every classifier has been allowed to update up to 2000 iterations. The obtained learning error convergence for all the 4 classifiers have been shown in Fig.5. It can be observed that for all the cases, proper convergence has been obtained. The classification efficiency for training and test data along with mean square error has been shown in Table4. It can be observed that, very fine learning happens for each classifier and performance is around [98.6%, 98.4%, 97.2% and 99.2%] on an average for training data. But performance is very poor over test data which is [76%, 77%, 83% and 79%]. Such kind of performance over test data may not be acceptable when considering the critical applications.

Table4. Mean square error/Training /test data performance for each classifier

Trail No.	CLF1 MSE / Tr. / Test	CLF2 MSE / Tr. / Test	CLF3 MSE / Tr. / Test	CLF4 MSE /Tr. / Test
1	0.108 / 99 / 79	0.197 / 98 / 76	0.324 / 97 / 81	0.0105 /99 /80
2	0.0109 / 99 / 78	0.0103 / 99 / 76	0.0156 / 98 / 76	0.0104 / 99 / 79
3	0.0123 /99 / 79	0.0203 / 98 / 77	0.0311 / 97 / 84	0.0014 / 100/ 78
4	0.0308 / 97 / 73	0.0203 / 98 / 77	0.0313 / 97 / 82	0.0102 / 99 / 79
5	0.0119 / 99 / 76	0.0103 / 99 / 77	0.0311 / 97 / 83	0.0104 / 99 / 79
Mean (Tr.	98.6000 /77.0000	98.4000 /76.6000	97.2000/81.2000	99.2000/79.0000

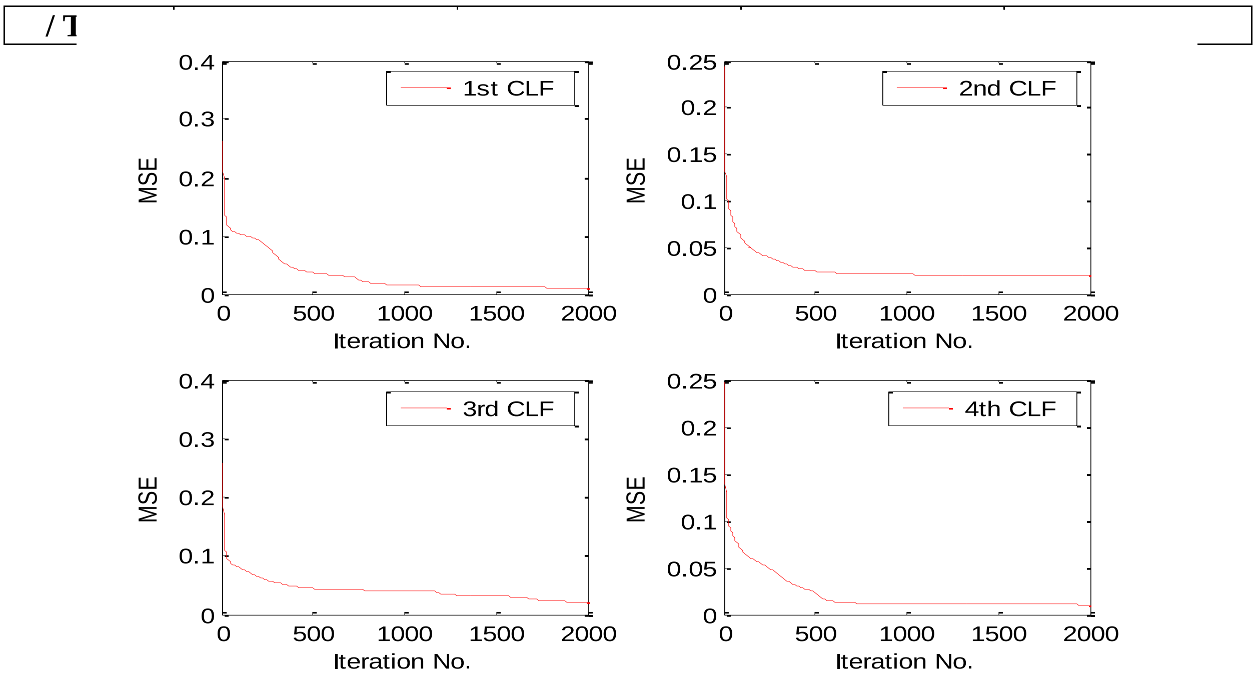
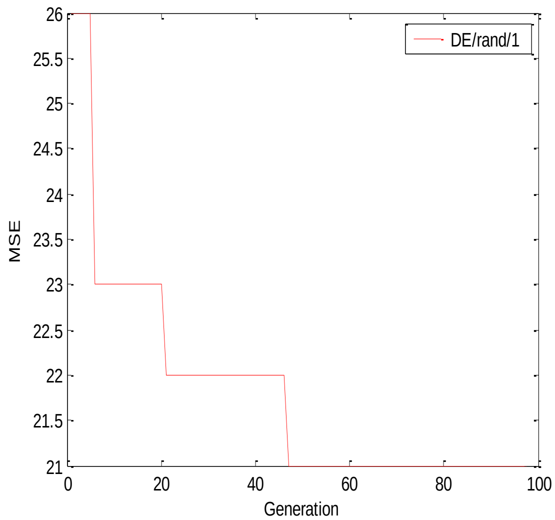


Fig.5.Individual classifier error convergence

Table5: Performance delivered by different types of ensemble network

Trial No.	Majority Decision Voting Efficiency %	Mean Decision Value Efficiency %	DEEANN				
			CLF1 WT	CLF2 WT	CLF3 WT	CLF4 WT	Efficiency %
1	88.5185	88.8889	0.3533	0.1890	0.1265	0.2744	92
2	87.4074	90	0.2522	0.3336	0.1976	0.3378	92
3	90	90.7407	0.3946	0.1857	0.2795	0.1000	93
4	88.8889	89.6296	0.1405	0.4412	0.2579	0.2868	92
5	88.6845	89.7253	0.2437	0.3349	0.1729	0.3448	92

Apart from differential evolution-based ensemble, two other different type of ensemble has been developed. The obtained result after 5 independent trials have been shown in Table5. In first case, final decision of each classifier has been considered depending upon the majority voting strategy. In second approach, rather than counting the final decision, numeric decision given for most favorable class has been considered and mean is estimated for overall decision values. In this case, there is advantage that, none of the classifier outcome is ignored completely. With this, process opinion of every expert is being utilized. These two processes are easy to implement, and it can be observed that there is betterment achieved by these two-ensemble compared to individual. But, it is also observed that ensemble network that used the mean value had better performance over majority voting method. When the differential evolution-based ensemble has been applied, the obtained weights for each classifier under five trials have been shown in Table5 along with achieved efficiency.



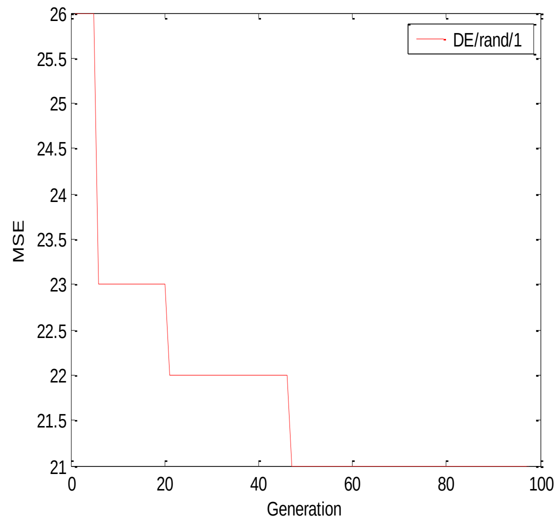


Fig.6. Weight evolution for ensemble by DE **Fig.7. Learning error minimization by DE for ensemble**

Table6: Comparative performance between all classifiers

CLF1	CLF2	CLF3	CLF4	MJVT	MNDS	DEEANN
77.0000	76.6000	81.2000	79.0000	88.6999	89.7969	92.2000

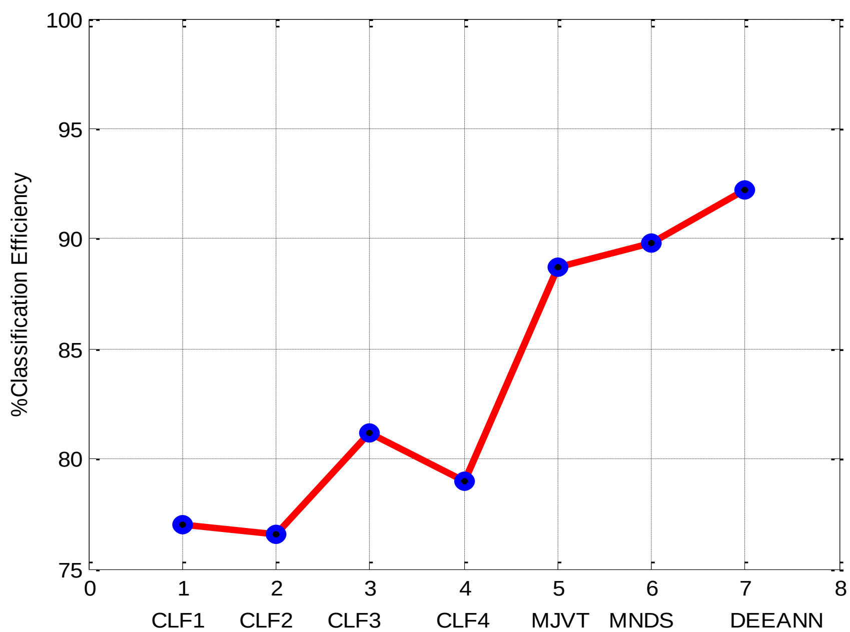


Fig.8.Comparative performance for different classifiers

It is very interesting to note that a very high efficiency around 92% has been achieved along with consistent performance. The obtained variation in weights for all the individual classifier of first stage has been shown in Fig.6. While the learning error minimization convergence by DE for ensemble has been shown in Fig.7.The final overall comparative performance of each classifiers of first stage of ensemble neural network (CLF1-CLF4), overall performance of ensemble neural network with majority voting (MJVT), ensemble neural network with mean decision (MNDS) and differential evolution based ensemble neural network(DEEANN) has been shown in Table 6. The comparative graphical plot for classification efficiency has been shown in Fig.8, which shows the benefit of ensemble as well as benefit of different structures of ensemble neural networks.

Table7: Sensitivity and Specificity performances by different classifier structures

Classifiers	Sensitivity (%)	Specitivity (%)
CLF1	85.83	83.33
CLF2	84.17	86.00
CLF3	82.50	92.67
CLF4	82.50	90.00
MJVT	78.33	97.33
MNDS	84.17	94.00
DEEANN	90.00	93.33

Ideally, classifier having 100% sensitivity and 100% specificity is absolutely best. To understand the predictability of individual classifier, in terms of sensitivity and specificity, performances obtained by a different form of classifiers have been evaluated and shown in Table7. It is clear that all the individual neural network classifier (CLF1-CLF4) have more or less nearly the same performances. A betterment in specificity and a remarkable drop in sensitivity has been observed in ensemble neural network with majority voting (MJVT). Performance has been improved further by ensemble neural network with mean decision (MNDS) when compared to individual classifiers and MJVT. But DEEANN has outperformed, when compared to the individual neural network classifiers (CF1-CF4), MJVT, and MNDS.

V. CONCLUSION

In this proposed work, the importance of ensemble neural network in the area of classification has been shown. The observations are very clear and appealing that instead of any single classifier, ensemble classifier is always beneficial. The structural form of ensemble has achieved with evolutionary process, which has delivered the outstanding efficiency as well as high level of consistency, which is very important in the critical applications. The proposed method is computation efficient and has shown quality improvement, compared to either individual classifier as well as conventional form of ensemble methods. Computational efficiency of DEEANN has been obtained at the cost of little extra computational cost. Here, proposed work has been evaluated only on Heart-Disease dataset. Further, the application-specific dataset can be used for evaluation of performance.

ACKNOWLEDGMENT

I would like to express my deep gratitude to Dr. R J Anandhi, Professor and Head, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, my research supervisor, for her patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to thank Dr. R V Praveena Gowda, Principal, The Oxford College of Engineering, Bengaluru, for his advice and assistance in keeping my progress on schedule. Finally, I wish to thank my husband, parents and family members for their support and encouragement throughout my study.

REFERENCES

[1] Zhi-HuaZhou, JianxinWu, ”Ensembling neural networks: Many could be better than all”, Artificial Intelligence, Volume 137, Issues 1–2, May 2002, Pages 239-263

[2] ZhiyeZhao Yun Zhang HongjianLiao,” Design of ensemble neural network using the Akaike information criterion”, Engineering Applications of Artificial Intelligence, Volume 21, Issue 8, December 2008, Pages 1182-1188

[3] ZhiyeZhao YunZhang,” Design of ensemble neural network using entropy theory”, Advances in Engineering Software, Volume 42, Issue 10, October 2011, Pages 838-845.

[4] Martha Pulido, Patricia Melin, Oscar Castillo, ”Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange”, Information Sciences, Volume 280, 1 October 2014, Pages 188-204

[5] HuiLi Xuesong Wang ShifeiDing,”Research of multi-sided multi-granular neural network ensemble optimization method”, Neurocomputing, Volume 197, 12 July 2016, Pages 78-85

[6] KangilKim, YunJin,”Center-shared sliding ensemble of neural networks for syntax analysis of natural language”, Expert Systems with Applications, Volume 83, 15 October 2017, Pages 215-225

[7] YukiKawana, NorimichiUkita, Jia-BinHuang,”Ensemble convolutional neural networks for pose estimation”, Computer Vision and Image Understanding, Volume 169, April’18, Pages 62-74

[8] Jose V.Manjón,”MRI white matter lesion segmentation using an ensemble of neural networks and overcomplete patch-based voting”, Computerized Medical Imaging and Graphics, Volume 69, November 2018, Pages 43-51

[9] TaigaAsakura, YasuhiroDate,” Application of ensemble deep neural network to metabolomics studies”, Analytica Chimica Acta, Volume 1037, 11 December 2018, Pages 230-236

[10] LongZhang GuoxianYu,” Protein–protein interactions prediction based on ensemble deep neural networks”, Neurocomputing, Volume 324, 9 January 2019, Pages 10-19

[11] WangRui , “Blur image identification with ensemble convolution neural networks”, Signal Processing, Volume 155, February 2019, Pages 73-82

[12] WenzhengBao, BinYang,” CMSENN: Computational Modification Sites with Ensemble Neural Network”, Chemometrics and Intelligent Laboratory Systems, Volume 185, 15 February 2019, Pages 65-72