Artificial Neural Network Training Using Differential Evolutionary Algorithm for Classification

Tapas Si*, Simanta Hazra, and N.D. Jana

Department of Information Technology
National Institute of Technology, Durgapur
West Bengal, India
{c2.tapas,simanta.hazra,nanda.jana}@gmail.com

Abstract. In this work, we proposed a method of artificial neural network learning using differential evolutionary(DE) algorithm. DE with global and local neighborhood based mutation(**DEGL**) algorithm is used to search the synaptic weight coefficients of neural network and to minimize the learning error in the error surface.**DEGL** is a version of DE algorithm in which both global and local neighborhood-based mutation operator is combined to create donor vector.The proposed method is applied for classification of real-world data and experimental results show the efficiency and effectiveness of the proposed method and also a comparative study has been made with classical DE algorithm.

1 Introduction

Artificial neural network(ANN)[13] is a useful tool in machine learning. ANN acts a important roll as classifier in classification of non-separable data.To apply ANN to any problem, it is necessary to train the ANN.A well-known algorithm named Back-propagation (BP) algorithm is used to train the ANN in supervise learning. BP Algorithm is a gradient descent optimize technique to search the synaptic weight coefficients of ANN and to minimize the learning error in the error surface. But BP algorithm has several drawbacks. The error function of ANN is a multi-modal function which has several local minima.The BP algorithm gets stuck into local minima easily. secondly, it has slow convergence speed.Therefore evolutionary algorithms like Genetic Algorithm(GA) [12],Particle Swarm Optimization(PSO) [6,7],Differential Evolutionary algorithm [1,3,4,5] are used to train the ANN as an alternative of BP algorithm. In the ANN training using GA method(GANN),weight coefficients of neural network are encoded in chromosome and selection,cross-over and mutation operators are used to minimize the error. But GANN suffers fom early convergence. GA has diversity in its population but lacks of convergence speed towards global optimia.On the other hand, PSO is applied successfully to train the ANN training[6].PSO has faster convergence speed than that of GA but it lacks of diversity in population.Recently DE

* Corresponding author.

S.C. Satapathy et al. (Eds.): Proceedings of the InConINDIA 2012, AISC 132, pp. 769–778.
springerlink.com

algorithm is successfully applied for training of ANN. Advantages of DE algorithm are as follows: a possibility of finding the global minimum of a multi-modal function regardless of initial values of its parameters, quick convergence and a small number of parameters to set up at the start of the algorithm operation.In the year 2003, Fan and Lampinen [10] introduced Trigonometric DE(TDE) algorithm and applied to train the ANN as a test case for their proposed algorithm. Recently,in paper [4],DE algorithm was used to train ANN and applied to classification of parity-p problem.In Ref.[1],Adam Slowik applied adaptive DE algorithm with multiple trial vectors to ANN learning to classify parity-p problem.Liu Mingguang and Li Gaoyang combined the BP algorithm and the differential evolutionary algorithm to train the neural network in order to achieve better local search and optimizing speed in paper [3].Yuelin Gao and Junmin Liu introduced a modified DE algorithm and trained the neural network for exclusive-OR (XOR) classification and function approximation problem in paper [5]. In this work, we trained a feed forward neural network using a DE with Local and global mutation proposed by Das et al.[2] and applied for classification of real-world data.

2 Artificial Neural Network

The n attributes in data set are used as input to NN. In this experiment, we used feed forward multi-layer perceptron (MLP, see Figure 1) that has three layers known as input, hidden and output layers respectively. Each processing node, except the input layer nodes,calculates a weighted sum of the nodes in the preceding layer to which it is connected. This weighted sum passes through the transfer function to derive its output which is fed to the nodes in the next layer. Thus, the input to node j is obtained as

$$net_j = \sum_{j=1}^{M} W_{ij} O_i + bias_j \tag{1}$$

and output as

$$O_j = F_a(net_j) \tag{2}$$

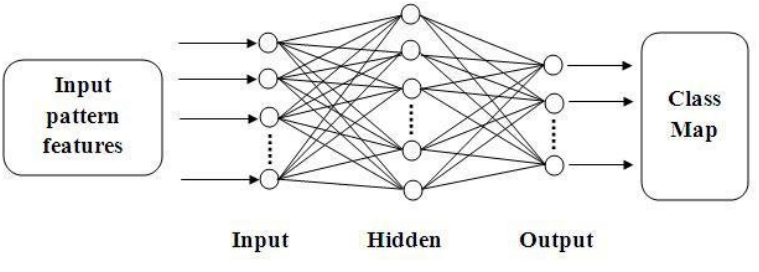

Fig. 1. Feed-forward neural network

where W_{ij} is the synaptic weight for the connection linking node i to j, value for node j, is the output of node j, and is the activation function (AF). Here the AF is considered as a sigmoid function[13] and is defined as

$$F_a(net_j) = \frac{1}{1 + e^{net_j}} \tag{3}$$

MLP uses back-propagation (BP) learning algorithm [13] for weight updating. The BP algorithm basically reduce the sum of square error called as cost function (CF), between the actual and desired output of output-layer neurons in a gradient descent manner. The CF is given as

$$CF = \sum_{i=1}^{N} \sum_{j=1}^{M} (O_{ij}^{des} - O_{ij}^{pred}) \tag{4}$$

where i is a training pattern and j is the output node. O_{ij}^{pred} denotes the predicted output of node j when the training pattern i is applied to the network, and O_{ij}^{des} is the corresponding desired output. The details of BP algorithm including derivation of equation can be obtained from [13].The number of input nodes in the input-layer is equal to the number of attributes and the number of nodes in the output-layer is equal to the number of classes present in the data set.

3 Differential Evolutionary Algorithm

DE [8] algorithm is a floating-point population based derivative free global optimization technique. A differential operator is used to create new offspring from parent chromosomes instead of classical crossover or mutation operator in genetic algorithm(GA).In Table 1,a DE scheme, namely $DE/rand/1/bin$ is described.

3.1 DEGL Algorithm

DE with local and global mutation(DEGL) is proposed by Das et al.[2].In DEGL algorithm, local mutant vector L is created using the Eq.(5)

$$L_i(t+1) = X_i(t) + \alpha_1.(X_{nbest}(t) - X_i(t)) + \beta_1.(X_p(t) - X_q(t)) \tag{5}$$

where $X_{nbest}(t)$ is the neighborhood best of i^{th} vector in iteration t and p and q are the neighborhoods of the same vector and $p, q \in [i-k, i+k]$ where $i \neq p \neq q$ .k is the radius of the neighborhood of i^{th} vector in the ring topology.In this work,two neighbors are selected in the radius(k)=1 of i^{th} vector based on positional index(not geometric position). global mutant vector G is created using the Eq.(6)

$$G_i(t+1) = X_i(t) + \alpha_2.(X_{best}(t) - X_i(t)) + \beta_2.(X_r(t) - X_s(t)) \tag{6}$$

Table 1. The Main Steps of DE Algorithm

Begin
N = population size;
D=dimensional size;
X=current population;
t=the generation index;
V=donor Vector
U=trial Vector
i=population index
j=dimension index
Initialize the population of size N
while ($generation\, t \leq MaxGeneration$)
for $i = 1$ **to** N
Calculate the fitness value $f(X_i(t))$
//Mutation:
for $j = 1$ **to** D
$V_{ij}(t+1) = X_{r1j}(t) + F.(X_{r2j}(t) - X_{r3j}(t))$
// $r1, r2\, and\, r3 \in [1, N]$,are integers and mutually exclusive, and $F \in (0, 2)$
//is a scale factor.
for end
//Crossover:
for $j = 1$ **to** D
if $R_j(0,1) \leq CR$ then
$U_{ij}(t+1) = V_{ij}(t+1)$
//$R_j(0,1)$ is uniformly distributed random number in (0,1) and $CR \in (0,1)$
//is crossover rate
else
$U_{ij}(t+1) = X_{ij}(t)$
End if
for end
// Selection:
if $f(U_i(t+1)) \leq f(X_i(t))$ then
$X_i(t+1) = U_i(t+1)$
else $X_i(t+1) = X_i(t)$
End if
for end
while end
End

where $X_{best}(t)$ is the best solution in the population in generation t.r and s are selected from $[1, NP]$ and $r \neq s \neq i$.Local mutant vector L and global mutant vector G is combined in order to create actual donor vector V using the Eq. (7)

$$V_i(t+1) = w.G_i(t+1) + (1-w).L_i(t+1) \tag{7}$$

where $w \in (0,1)$ is a weighted factor to adjust exploration and exploitation of the search capability.

4 ANN Training Using DE Algorithm – A Review

The error function of ANN is a highly multi-modal function which has lot of local minimas.The ANN is training using well-known BP algorithm which has several drawbacks:one is that it gets stuck in local minima and another is slow convergence speed . But it has strong local search capability.The training process is to minimize the error function by adjusting weights to obtain a desired accuracy as well as to achieve faster convergence speed.

In recent few years, a lot of contributions have been given in ANN training using DE algorithm. In order to keep a reasonable balance between convergence speed and the capability of global search,Liu Mingguange et al. [3] combined the BP and DE algorithm to optimize the weights and threshold value adjustments of ANN.

Yueline Gao et al. [5] introduced a novel mutation operator in DE algorithm to obtain a good balance between global and local search and applied in BP neural network to solve exclusive-OR and function approximation problem.And as result ,reduced training time and improved testing accuracy are achieved.

Adam Slowik and Michal Bialko [4] presented artificial neural network training using DE algorithm with adaptive selection of control parameters in DE and applied to classification of parity-p problem.

Adam Slowik [1] applied adaptive DE algorithm with multiple trial vectors to ANN learning to classify parity-p problem.But it takes additional training time $(m-1) \times n_t \times G$ compare to classical DE algorithm where m is the number of trial vectors and n_t is the time taken to calculate the error function values for n records in training data set and G is the maximum generation.

Hui-Yuan Fan and Jouni Lempinen [10] introduced Trigonometric Differential Evolutionary (TDE) algorithm and they applied it to train the ANN with considering XOR problem and aerodynamic five-hole probe calibration problem.

As DEGL algorithm provides a good balance between local and global search, DEGL is used in ANN training in this work with the hope that it will provide a good performance for classification problems.

5 ANN Training Using DEGL Algorithm

In this work, ANN is trained using DEGL Algorithm(**DEGL-ANN**) to search the synaptic weight coefficients of a feed forward neural network as well as to minimize the mean-square-error in the error surface. We used a feed forward multi-layer perceptron(MLP) having n input nodes in input-layer, m output nodes in output-layer and (2n+1) hidden nodes in the hidden-layer. Mean Square Error (MSE) is calculated by following Equation (8) and it is used as a fitness function for DE algorithm.

$$MSE = \frac{1}{N.M} \sum_{i=1}^{N} \sum_{j=1}^{M} (O_{ij}^{des} - O_{ij}^{pred}) \tag{8}$$

where i is a training pattern and j is the output node. O_{ij}^{pred} denotes the predicted output of node j when the training pattern i is applied to the network,O_{ij}^{des} is the corresponding desired output,N=number of training samples and M=number of outputs.For the outputs, a binary 1-of- m encoding is used in which each bit represents one of the m-possible output classes of the problem definition. Only the correct output class carries a $(1 - \epsilon)$, whereas all others carry $\epsilon(= 0.1)$ and winner-takes-all policy is adopted.

Total number of weight coefficients in the ANN is $D = (n * (2n + 1) + (2n + 1) * m) = (n + m)(2n + 1)$.These weight coefficients are initialized in the interval [-1,1] with uniform distribution and treated as vector elements in DE.Each and every vector in DE represents a neural network and is trained with the complete training set.After completion of maximum number of iteration or after meeting to the minimum error criteria, best neural network is used to check with the unknown test data.Finally,classification accuracy is measured by confusion matrix.

6 Data Set Description

In this work, we used five different data sets collected from UCI machine learning repository[15]. The data are normalized between [0, 1] and missing values are coded as zeros. The details of the data sets are described in below:

1. Fishers iris data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. It contains 4 attributes and one class attribute.
2. Cleveland heart disease data set has 303 records. Among the entire records, 160 are healthy, 137 are sick, and six are missing records. The data set has 13 attributes and 1 class attribute. The class attribute refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1, 2, 3, 4) from absence (value 0).
3. Breast cancer data set has total 286 instances and 9 attributes with one class attribute.201 instances has the no-recurrence-events class whereas 85 instances has recurrence-events.This data set has missing values.
4. BUPA liver disorders data set has 345 Number of Instances and 7 attributes including one class attribute. Attribute characteristics are categorical, integer and real.This data set has no missing values.
5. Hepatitis data set has 155 number of Instances and 20 attributes including one class attribute. Attribute characteristics are categorical, integer and real.This data set has missing values.

7 Experimental Setup

K-fold cross-validation is used to obtain a reliable estimate of classifier accuracy where K=10 and best individual is selected in a run for testing the unknown data.

Classification accuracy was measured by Confusion Matrix [16].The confusion matrix is a useful tool for analyzing how well a classifier can recognize tuples of different classes.

7.1 Parameters

1. Population Size(N)=30.
2. Number of Generations=100 for iris data and 200 for rest of the data sets
3. Minimum Mean Square Error (MSE)=0.005
4. $\alpha_1 = \beta_1 = 0.8$
5. $\alpha_2 = \beta_2 = 0.8$
6. CR=0.8
7. $w = 0.729$

7.2 PC Configuration

1. System:Fedora 13(i386)
2. CPU: P IV 2GHz (Core 2 Duo)
3. RAM: 3 GB
4. Software: Matlab 2010b

8 Result and Discussion

In this work, ANN is trained with both DEGL and classical DE algorithm and applied to classify the data that are described in Sect. 6. The proposed method is run with 10-fold cross-validation.The sensitivity,specificity, testing accuracy and training accuracy from best run(providing best testing accuracy) have been described in Table 2 and 3 for DEGL-ANN and DE-ANN respectively. Mean,standard deviation of training accuracy and average time of training for each data set have been described in Table 4 for DEGL-ANN and DE-ANN.Mean,standard deviation of testing accuracy for each data set have been described in Table 5 for both DEGL-ANN and DE-ANN.Convergence of mean square errors for DEGL-ANN is given in Fig. 2 .The results in boldface in tables are better in the comparative analysis of two aforementioned DE algorithm.From Table 2, it is found that DEGL is able to produce a better generalization performance for neural network.From Table 4 & 5,it can be said that both classical DE and DEGL have a good efficiency in training and testing performances of neural network. The highest testing accuracy for cancer data set is 78% as per reported in UCI machine learning repository[15].In this work, highest testing accuracy 77.19% is achieved for the same data.The highest testing accuracy is 83% as per reported in Ref.[15] for hepatitis data whereas 90.32% is achieved from this experiment.For liver data, though DE-ANN produced better testing accuracy(e.g 74.29) than that of DEGL-ANN whereas DEGL-ANN provides better mean testing accuracy.

Table 2. Best results for each data set in DEGL-ANN

Data Set	Sensitivity(%)	Specificity(%)	Testing Accuracy(%)	Training Accuracy(%)
Iris	100.00	100.00	100	97.78
Heart	89.29	93.75	**91.67**	85.96
Cancer	32.35	96.25	**77.19**	76.89
Liver	84.00	57.24	72.75	74.85
Hepatitis	100.00	40.00	**90.32**	95.16

Table 3. Best results for each data set in DE-ANN

Data Set	Sensitivity(%)	Specificity(%)	Testing Accuracy(%)	Training Accuracy(%)
Iris	100.00	100.00	100	100
Heart	78.94	87.50	83.57	83.92
Cancer	32.00	91.67	74.12	74.90
Liver	85.00	60.00	**74.29**	66.45
Hepatitis	89.66	100.00	90.21	83.06

Table 4. Mean,standard deviation of training accuracy and average time of training for each data set

Data Set	**DEGL-ANN**			**DE-ANN**		
	Mean	Std. Dev.	Avg. Time(hrs.)	Mean	Std. Dev.	Avg. Time(hrs.)
Iris	98.85	0.5040	0.005	100.00	0.0	0.007
Heart	86.46	0.3698	0.27	83.53	0.4572	0.268
Cancer	77.3123	0.2251	0.26	74.83	0.6511	0.26
Liver	75.59	0.72	0.32	70.26	1.4323	0.31
Hepatitis	92.66	2.6462	0.13	83.79	3.0756	0.128

Table 5. Mean,standard deviation of testing accuracy for each data set in both DEGL-ANN and DE-ANN

Data Set	**DEGL-ANN**		**DE-ANN**	
	Mean	Std. Dev.	Mean	Std. Dev.
Iris	98.71	0.9367	**100.00**	0.0
Heart	**86.44**	3.1712	82.79	0.5945
Cancer	**73.97**	2.2862	72.54	1.3943
Liver	**70.67**	2.1930	68.50	2.4320
Hepatitis	**82.90**	6.8091	81.94	5.3114

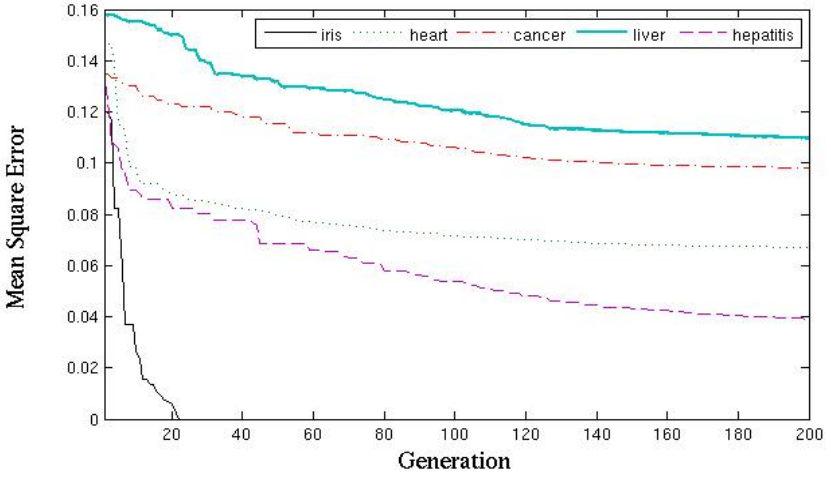

Fig. 2. Convergence graph for DEGL-ANN

9 Conclusion and Future Works

In this work, artificial neural network is trained using DE with local and global mutation and classical DE algorithm and application has been done for classification of real-world data set. From the experimental results, it has been shown that DEGL algorithm is efficient and effective in neural network learning with producing a good generalization performance than classical DE algorithm.Performances of a classifier are varied due to complexity of data set,preprocessing of data(i.e removing or replacing the missing values in data),transformation of categorical values to numerical values. Proper preprocessing scheme can be adopted to enhance the performances of the proposed method.In this work, fixed parameter values are used in all iteration of the DEGL algorithm. Different control mechanism for parameters setting in DEGL algorithm can be adopted while training the neural network.Form this study,it may be concluded that DEGL algorthm can be used in ANN learning for classification problems.

References

1. Slowik, A.: Application of an Adaptive Differential Evolution With Multiple Trial Vectors to Artificial Neural Network Training. IEEE Transactions On Industrial Electronics 58(8), 3160–3167 (2011)
2. Das, S., Abraham, A., Chakrabarti, U.K., Konar, A.: Differential Evolution Using a Neighborhood-Based Mutation Operator. IEEE Transactions On Evolutionary Computation 13(3), 526–553 (2009)
3. Mingguang, L., Gaoyang, L.: Artificial Neural Network Co-optimization Algorithm based on Differential Evolution. In: Second International Symposium on Computational Intelligence and Design, pp. 256–559 (2009)

4. Slowik, A., Bialko, M.: Training of Artificial Neural Networks Using Differential Evolution Algorithm. In: 2008 Conference on Human System Interactions, pp. 60–65 (2008)
5. Gao, Y., Liu, J.: A Modified Differential Evolution Algorithm and Its Application in the Training of BP Neural Network. In: Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Xi'an, China, pp. 1373–1377 (2008)
6. Lee, Y.S., Shamsuddin, S.M., Hamed, H.N.: Bounded PSO Vmax Function in Neural Network Learning. In: Eighth International Conference on Intelligent Systems Design and Applications, pp. 474–479. IEEE (2008)
7. Junyou, B.: Stock Price forecasting using PSO-trained neural networks. IEEE Congress on Evolutionary Computation, 2879–2885 (2007)
8. Price, K., Storn, R., Lampinen, J.: Dierential Evolution – A Practical Approach to Global Optimization. Springer, Heidelberg (2005)
9. Yan, H., Zheng, J., Jiang, Y., Peng, C., Li, Q.: Development of a Decision Support System for heart Disease Diagnosis using Multilayer Perceptron. In: Proceedings of the 2003 International Symposium on Circuits and Systems, pp. 709–712 (2003)
10. Fan, H.Y., Lampinen, J.: A Trigonometric Mutation Operation to Differential Evolution. International Journal of Global Optimization 27, 105–129 (2003)
11. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proc. IEEE World Congr. Comput. Intell., pp. 69–73 (1998)
12. Tsi, D.-Y.: Classification of Heart Diseases in Ultrasonic Images using Neural Networks Trained by Genetic Algorithm. In: Proceedings on International Conference on Signal Processing, pp. 1213–1216 (1998)
13. Haykin, S.: Neural Networks - A Comprehensive Foundation, 2nd edn. PHI (1994)
14. Rajasekaran, S., Pai, G.A.V.: Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications. PHI (2008)
15. `http://cml.ics.uci.edu`
16. Han, J., Kamber, M.: Data Mining - Concepts and Technique. Elsevier (2006)