Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Handwritten Marathi Characters Recognition Using Hybrid Evolutionary Gradient Descent of Distributed Error in Multilayer Feed Forward Neural Networks

Girish S. Katkar
Head, Dept. of Computer Science, Arts,Comm & Sci. College, Koradi, Nagpur(M.S.), India

Dr. M. P. Singh
Inst. of Computer and Information Sc., Dr. B. R. Ambedkar University, AGRA, India

Abstract- In this paper the performance of feedforward neural network with descent gradient of distributed error and genetic algorithm is evaluated for the recognition of handwritten characters of 'Marathi' script. The performance index for the feedforward multilaye.r neural networks is considered here with distributed instantaneous unknown error i.e. different error for different layers. The genetic algorithm is applied here to make the search process more efficient to determine the optimal weight vector from the population of weights. The genetic algorithm here is applied with distributed error and the fitness function for the genetic algorithm is also considered as the mean of square distributed error that is different for each layer. Hence the convergence is obtained only when the minimum of different errors is determined.
In this performance evaluation it has been analyzed that the proposed method of descent gradient of distributed error with genetic algorithm commonly known as hybrid distributed evolutionary technique for the multilayer feed forward neural performs better in terms of accuracy, epochs and number of optimal solutions for given training set and test pattern sets for the pattern recognition problem.

Keywords- Gradient Descent, Hybrid Evolutionary Distributed Technique, Multilayer Feedforward Neural Network, Pattern Recognition.

I. INTRODUCTION

Pattern recognition is an emerging area of the machine learning and intelligence. The problem of pattern recognition has been considered in many ways. The one of the most popular way is in the form of the pattern classification. Pattern classification is a problem in which the machine can distinguish the different input stimuli in meaningful categorization according to the present features in these inputs. The recognition of handwritten curve script as in the form of character classification, character association has been considered as the dominate area in field of pattern recognition with machine learning techniques [1, 2]. Soft computing techniques have been identified as a powerful tool to perform the task of pattern recognition for hand written curve script in the domain of machine learning [3 - 5]. The neural network techniques and evolutionary search methods have been used in various form of hybrid evolutionary algorithms for accomplish the task of pattern classification of handwritten curve scripts of many languages [6 - 8]. The feedforward multilayer neural network with gradient descent of backpropagated error is used widely for generalize pattern classification [9]. The analysis of this neural network architecture with generalized delta learning rule (backpropagation learning) has highlighted the performance and limitation of this architecture due to unavailability of more information for the units of output layer for handwritten character recognition [10].

II. GENERALIZED DESCENT GRADIENT LEARNING FOR DISTRIBUTED SQUARE ERROR

A multilayer feed forward neural network with at least two intermediate layers commonly known as hidden layer, in addition to the input and output layer can perform any complex generalized pattern classification task. The generalized delta learning rule [11] is a very common and widely used technique to train the multilayer feedforward neural networks for the pattern classification & pattern mapping. In this learning the optimum weight vector may be obtained for the given training set, if the weights are adjusted in such a way that the gradient descent is made along the total error surface in the weight space. The error for the minimization is actually not the least mean square error for the entire training set instead of this it is an instantaneous square error for each presented pattern on each time. Thus, for every pattern on each time there will be an unknown local error and there is the incrementally updating of the weight for each local error. Hence each time the weights are updated to minimize this known local error by propagating this error back to all hidden layers from the output layer. In this current work we are considering the distributed error instead of the backpropagated error. The instantaneous square error is not same for the each layer because each layer has its own actual output pattern vector. This distributed instantaneous square error imposes a constraint on the architecture of multilayer feed forward neural network. The generalized method for obtaining the weight update for hidden layers and output layer is formulated as:

Let (a_l, d_l) for $l = 1,2,\cdots,L$ be the current input pattern vector set of the training set of L pattern samples is presented to the multilayer feed forward neural network for formulating the generalized descent gradient of instantaneous square distributed error. As we have discussed already about the constraint of this multilayer feed forward neural network for keeping same the number of units in hidden and output layer as shown in figure

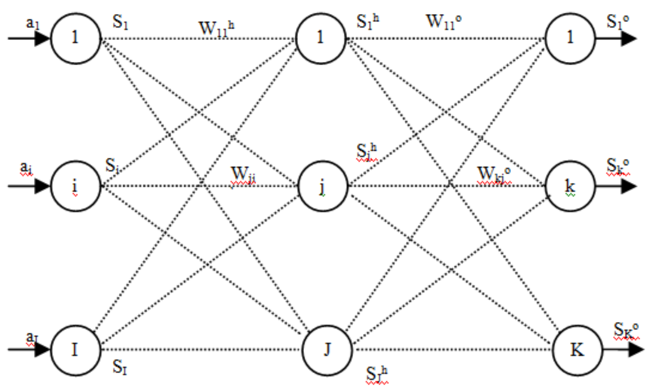


Fig. 1: Multilayer Feed Forward Neural Network Architecture

II.I Genetic algorithm with descent gradient of distributed Error

The majority of implementation of the GA is a derivative of Holland's innovative specification. In our approach the genetic algorithm is incorporated with descent gradient for distributed instantaneous mean square error learning in the multilayer feed forward neural network architecture for the generalized pattern classification. The input pattern vector with its corresponding output pattern vector form the training set is presented to the neural network. The neural network with its current setting of weights obtained the actual output for each unit of hidden layers and output layer. The distributed instantaneous mean square error is obtained and the proposed descent gradient learning rule for distributed error is applied up to some fixed arbitrary n iterations. Thus, the weights between the layers and bias values of units are updated up to n iterations for the given input pattern and improved from their initial stage. After this the iteration for weight update stops and the genetic algorithm is employed to evolve the population of modified weights and bias values. The genetic algorithm is applying for obtaining the optimal weight vector from the large size of weight space for the given training set with following three elements.

i. The genetic code for the weight vector representation in the form of chromosome;
ii. The technique for evolving the population of weight vectors;
iii. The fitness function for evaluating the performance of evolved weight vector;

Selection

The selection process of genetic algorithm selects good or fit population from the newly generated population. Here the selection process simultaneously considers newly generated sub chromosomes of hidden layer and output layer i.e. $C_H^{N\_new}$ and $C_O^{N\_new}$ respectively for selecting the good population for further cycle. Let a sub chromosome C_H^{Sel} from $C_H^{N\_new}$ is selected for which the distributed instantaneous mean square error for the hidden layer i.e. E_l^H for the pattern l reached to its accepted minimum level. Likewise a sub chromosome C_N^{Sel} from $C_O^{N\_new}$ is selected for which the distributed instantaneous mean square error for the output layer i.e. E_l^O for the same pattern l reached to its accepted minimum level.

III. Simulation Design and Implementation

In this simulation design and implementation, two proposed multilayer feed forward neural networks are considered. Both neural networks are trained with proposed descent gradient of distributed instantaneous mean square algorithm. Since every input pattern consist with 16 distinct features so that each neural network architecture contains 16 processing units in the input layer. First neural network architecture consists with input layer, two hidden layers with five units in each and one output layer with 5 units. Second neural network architecture consists with input layer, one hidden layer of 5 units and output layer also with 5 units.

Feature Extraction

There are five different samples of handwritten characters of *'Marathi'* script from five different people are collected in this simulation as input stimuli for the training pattern set. These scanned images of distinct handwritten characters of '*Marathi'* scripts are shown in figure 5 as:

Fig. 5: Scanned images of handwritten distinct '*Marathi'* scripts

The scanned images of hand written characters of '*Marathi'* scripts as shown in figure 5 are partition into sixteen equal parts, and the density values of the pixels for each part were calculated and obtained the center of density gravity. Therefore for each scanned image of handwritten characters of '*Marathi'* scripts we obtained the sixteen values as the input pattern vector of training set. Thus, we have the training set, which consist with sampled patterns of handwritten characters of '*Marathi'* scripts and each sample pattern is considered as pattern vector of dimension 16×1 with real number values. The output pattern vector corresponds to input pattern vector is of dimension 5×1 of the binary values. The test input patterns set is also considered with same method for the sample patterns those were not used in training set. The sample test patterns were used to verify the performance of trained neural networks.

Genetic algorithm with distributed error: The parameters used in the simulation of both the experiments for genetic algorithm with descent gradient learning for distributed error are as follows:

Table 1: Parameters used for decent gradient learning with distributed error

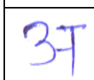
Parameter	**Value**
Learning rate for output layer (η_{O})	0.01
Learning rate for hidden layers $(\eta_{\mathrm{H}_1} \,\&\, \eta_{\mathrm{H}_2})$	0.1
Momentum term for output layer (α)	0.9
Momentum term for output layer (β)	0.7
Adaption rate (K)	3.0
Minimum error for the output layer $(\mathrm{MAXE_O})$	0.0001
Minimum error for the hidden layers $(\mathrm{MAXE_H})$	0.001
Mutation probability	Smaller than 0.01
Mutation population size for sub-chromosome of output layer	3

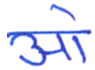
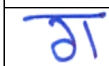
Mutation population size for sub-chromosome of hidden layers	3 each
Crossover population size for output layer	1000
Crossover population size for first hidden layer(for 16-5-5-5 architecture)	1000
Crossover population size for second hidden layer(for 16-5-5-5 architecture)	500
Crossover population size for hidden layer(for 16-5-5 architecture)	1000
Number of iteration prior to applying GA	5000
Initial population	Values of weights & bias in each sub chromosomes up to 5000 iterations of descent gradient for distributed error.
Fitness evaluation functions (two fitness function for 16-5-5 architecture and three fitness function for 16-5-5-5 architecture)	Distributed instantaneous sum of squared errors $E_l^O = \frac{1}{2}\sum_{k=1}^{K}(d_k^l - S_k(y_k^O))^2$ $E_l^{H_1} = \frac{1}{2}\sum_{g=1}^{G}(d_k^l - S_g(y_g^{H_1}))^2$ $E_l^{H_2} = \frac{1}{2}\sum_{j=1}^{J}(d_k^l - S_j(y_j^{H_2}))^2$

IV. RESULTS AND DISCUSSION

The results from Simulation design and implementation for the neural network architectures i.e. for 16-5-5 are considered for 65 training sample examples of Handwritten '*Marathi'* scripts with two hybrid techniques. The techniques commonly used are genetic algorithm with descent gradient for backpropagated instantaneous mean square error and genetic algorithm with descent gradient for distributed instantaneous mean square error. The performance of both the neural network architectures have been evaluated with these two hybrid techniques of learning for the given training set and the performance analysis is also performed. The results of performance evaluation are shown with tables 5. The entries of tables are presenting mean values of iterations and number of convergence weight matrices of five trials with each hybrid technique for given training set.

Table 2: Performance evaluation for GA with descent gradient of distributed Error and back Propagated Error for 16-5-5 architecture

Distinct Marathi handwritten curve script	Samples	GA with Descent Gradient of Distributed Error		GA with Descent Gradient of Backpropagated Error	
		Iteration	Count	Iteration	Count
	Sample 1	5	56	2	46
	Sample 2	1	200	1	173
	Sample 3	1	396	4	136
	Sample 4	1	394	1	198
	Sample 5	1	395	1	157
	Sample 1	240	201	156	48
	Sample 2	1	196	3	160
	Sample 3	1	200	4	197
	Sample 4	1	195	2	200
	Sample 5	1	196	1	198
	Sample 1	130	96	21	12
	Sample 2	2	382	1	195
	Sample 3	2	51	3	198
	Sample 4	2	396	2	192

	Sample 5	2	392	1	197
	Sample 1	90	397	29	20
	Sample 2	1	178	2	107
	Sample 3	1	201	1	198
	Sample 4	1	396	3	196
	Sample 5	1	391	3	199
	Sample 1	198	30	123	21
	Sample 2	3	396	2	197
	Sample 3	1	389	3	199
	Sample 4	1	202	2	196
	Sample 5	1	296	4	198
	Sample 1	50	18	34	6
	Sample 2	2	385	3	200
	Sample 3	2	384	1	199
	Sample 4	2	388	2	195
	Sample 5	1	386	3	198
	Sample 1	80	83	67	46
	Sample 2	2	395	2	197
	Sample 3	2	196	1	180
	Sample 4	2	197	3	154
	Sample 5	2	389	2	198
	Sample 1	18	198	25	15
	Sample 2	2	393	1	198
	Sample 3	2	392	3	195
	Sample 4	2	392	2	196
	Sample 5	2	396	1	198
	Sample 1	190	38	102	20
	Sample 2	2	391	2	199
	Sample 3	1	395	1	199
	Sample 4	1	193	3	170
	Sample 5	1	195	1	119

In the results tables are containing the information about counts. The counts are here representing the number of optimum solutions i.e. the number of weight matrices on which the network is convergence for the given training set. The integer value for the epoch in tables is representing the number of iterations performed by each learning method to classify the given input pattern. It has been observed from the results that no case of non convergence is found. Thus the network is able to successfully converge for more than one optimum weight vectors or solution for the given input pattern. Table 5 of simulated result is showing the performance evaluation between GA with descent gradient of instantaneous mean square distributed error and GA with descent gradient of backpropagated error for the network architecture 16-5-5.

V. CONCLUSION

In this work we have considered the simulation of two neural network architectures for their performance evaluation with descent gradient of instantaneous mean square distributed error with GA and descent gradient of instantaneous mean square backpropagated error with GA for the classification of handwritten '*Marathi'* curve scripts. Therefore on the basis of simulation results & analysis the following observations can be drawn:

1. This is obvious that number of iteration for GA with descent gradient of distributed error are more because the in this method there are three objective functions and all of them should minimize for the optimal solution.
2. It can also see from the results that the behavior of GA with descent gradient of distributed error is more consistent & exhibiting less randomness in compare to GA with descent gradient of backpropagated error.
3. Generally the GA starts form the random solutions and converge towards the optimal solution. Hence in multi objective optimization the randomness of GA more increases and possibility to obtain optimal solution decreases.
4. The multi objective optimization is a dominate thrust area in soft computing research. There are various real world problems where multi objective optimization is required.

References

[1] Impedovo, S., “Fundamentals in Handwriting Recognition.” NATO-Advanced Study Institute, vol. 124, Springer-Verlag (1994)

[2] Mori, S., Suen, C.Y. and Yamamoto, K., “Historical review of OCR research and development,” Proceeding of the IEEE, vol.80 (7), pp. 1029-1058 (1992)

[3] Fukushima, K. and Wake, N., “Handwritten alphanumeric character recognition by the neocognitron,” IEEE transaction on Neural Networks, vol. 2(3), pp. 355-365 (1991)

[4] Blackwell, K.T., Vogl, T.P., Hyman S.D., Barbour, G.S. and Alkon, D.L., “A New Approach to Handwritten Character Recognition,” Pattern Recognition vol. 25, pp. 655-666 (1992)

[5] Ie Cun, Y., Boser, B., Denkar, J.S., Henderson, D., Howard, R.E., Hubbard, W., and Jackel, L.D., “Handwritten Digit Recognition with a Back-Propagation Network,” Advances in Neural Information Processing Systems, vol. 2, pp. 396-404 (1990)

[6] Kharma, N.N., and Ward, R.K., “A novel invariant mapping applied to hand-written Arabic character recognition,” Pattern Recognition vol. 34(11), pp. 2115-2120 (2001)

[7] Badi, K. and Shimura, M., “Machine recognition of Arabic cursive script,” Trans. Inst. Electron. Commun. Eng., vol.65(E), pp. 107-114 (1982)

[8] Suen, C.Y., Nadal, C., Lagault, R., Mai, T.A., and Lam, L., “Computer recognition of unconstrained handwritten numerals,” Proc. IEEE, vol. 80(7), pp. 1162-1180 (1992)

[9] Knerr, S., Personnaz, L., and Dreyfus, G., “Handwritten digit recognition by neural networks with single-layer training,” IEEE Trans. on Neural Networks, vol. 3, pp. 962-968 (1992)

[10] Lee, S.W., and Song, H.H., “A New Recurrent Neural Network Architecture for Visual Pattern Recognition,” IEEE Trans. on Neural Networks, vol. 8(2), pp. 331-340 (1997)

[11] Rumelhart, D.E., Hinton G.E., and Williams R.J., “Learning internal representations by error propagation.”, MIT Press, Cambridge, vol. 1,pp. 318–362 (1986).

[12] Sprinkhuizen-Kuyer, I.G., and Boers, E.J.W., “The Local Minima of the error surface of the 2-2-1 XOR network,” Annals of Mathematics and Artificial Intelligence, vol. 25(1-2), pp. 107-136 (1999)

[13] Zweiri, Y.H., Seneviratne, L.D., and Althoefer, K., “Stability Analysis of a Three-Term Backpropagation algorithm,” Neural Networks Journal, vol. 18(10), pp. 1341-1347 (2005)

[14] Abarbanel, H., Talathi, S., Gibb, L., and Rabinovich, M., “Synaptic plasticity with discrete state synapses,” Phys. Rev., vol. E, 72:031914 (2005)

[15] Shrivastava, S. and Singh, M.P., “Performance evaluation of feed-forward neural network with soft computing techniques for hand written English alphabets”, Journal of Applied Soft Computing, vol. 11, pp. 1156-1182 (2011)

***

