International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2, Issue-2, May2012

Hybrid Evolutionary Techniques to Restricted Feed Forward Neural Network with Distributed Error for Recognition of Handwritten Hindi 'MATRAS'

Meenu Gupta, Ajay Rana

ABSTRACT:- This paper evaluates the performance of restricted feed forward neural network trained by hybrid evolutionary algorithm with generalized delta learning rule for distributed error to obtain the pattern classification for the given training set of Handwritten Hindi 'MATRAS'. Generally, the feed forward neural network considers the performance index as back-propagated instantaneous unknown error for output of hidden layers. Within this proposed endeavor, we are considering the performance index of distributed instantaneous unknown errors i.e. different errors for different layers. In this case, the convergence is obtained only when the minimum of every error on different layer is determined. The simulation for the performance evaluation is conducted for hand-written 'MATRAS' of Hindi language scripted by five different people. These samples are stored as scanned images. The MATLAB is used to determine the densities of these scanned images after partitioning each image into 16 portions. These 16 densities for each character are used as an input pattern of training set. We consider five trials for each learning method and results are presented with their mean value.

Keywords:- Genetic Algorithm, Handwritten Hindi MATRAS, Multilayer Feed Forward Neural Network, Pattern Recognition

I. INTRODUCTION

Recognition is a basic property of all human beings; when a person sees an object, he or she first gathers all information about the object and compares its properties and behaviors with the existing knowledge stored in mind to make a proper match with the known. The concept of recognition is simple in human, but in the world of computer science, recognizing any object is an amazing feat. Pattern recognition is the study in the field of machine intelligence that how machine can observe the environment, learn to distinguish patterns of interest from their background and make sound and reasonable decisions about the categories of the patterns. Handwriting Recognition has been one of the most fascinating and challenging research areas in field of image processing and pattern recognition in the recent years. It contributes immensely in the direction of improving the interface between man and machine in numerous applications. Several research works have been focusing toward evolving newer techniques and methods that would reduce the processing time while providing higher recognition accuracy [7] The multilayer neural network, trained by the Back-Propagation (BP) algorithm, is currently the most widely used neural network since it can solve many complex pattern recognition problems[2,8]. The back-propagation algorithm explores the mechanism for training to the neural network by incremental adjustment of the set of weights for the given training set of patterns. However the back-propagation algorithm has several limitations and there are various solutions and modifications to minimize the limitation of BP learning algorithm. But still it has no guarantee of convergence and the nature of error is still random and back-propagated.

The evolutionary search algorithm i.e. Genetic Algorithm (GA) is also considered as a better alternative to search the global minima and for convergence if the search space is large. As the complexity of the search space increases, GA presents an increasingly attractive alternative to gradient based techniques such as back-propagation algorithm [4,5]. It can sample the search space irrespective of where the existing solution is to be found while remaining is based towards good solution.

In this paper, we are dealing with performance index i.e. instantaneous unknown error in different ways instead of considering it in back-propagated nature. We are considering it as distributed error for the multilayer feed forward neural network, in which the number of units in hidden layer and output layers are equal i.e. the restricted multilayer feed forward neural network architecture. Thus, the same desired output pattern for a presented input pattern will distribute to every unit of hidden layers and output layer. Each unit of hidden layers and output layer `has its own actual output, so the performance measure or the error will different for each layer. Thus, the instantaneous error is now distributed instead of back propagated. The hybrid evolutionary algorithm is used as learning method in multilayer neural network architecture for the classification of handwritten '*MATRAS'* of Hindi. In the implementation of hybrid evolutionary algorithm, the performance index which works as the fitness evaluation function, is defined with distributed instantaneous error instead of instantaneous back propagated error only for proposed restricted multi layer feed forward neural network architectures.

II. RESTRICTED MULTILAYER FEED-FORWARD NEURAL NETWORK WITH DISTRIBUTED ERROR

It is known that a multilayer feed forward neural network with atleast two intermediate layers commonly known as hidden layers, in addition to the input and output layer can perform the pattern classification task. If the units in hidden layers and in the output layer are non-linear, then the number of unknown weights depend on the number of units in the hidden layer, besides the number of units in the input and output layers. To accomplish the complex pattern classification for the handwritten Hindi MATRAS, a systematic way is required to determine the method for updating these

Manuscript received on April 20, 2012
Meenu Gupta, Director, G L Bajaj Group of Institutions, Mathura (UP),
Ajay Rana, Director, ASET, Amity University, Noida (UP).

weights for each input-output pattern pair provided to the network from the given training set. In order to do the updation of weights in a supervisory mode, it is necessary to know the desired output for each unit in the hidden and output layers. Once the desired output is known, the error i.e. the difference between the desired and actual outputs from each unit may be used for updating the weights leading to the units from the units in the previous layer. Actually, we only know the information about desired output for the units in the output layer, and not for the units in the hidden layers. Hence, in order to develop a learning rule for this multilayer feed forward neural network which is constructed in restricted manner as the number of units in output layer and hidden layer are equal. The concept of error back-propagation is used with a differentiable non-linear output function for each unit in the hidden and output layers. This rule is commonly known as generalized delta rule or back-propagation learning rule [8].

A. GENERALIZED DELTA LEARNING RULE WITH DISTRIBUTED ERROR

As we know in generalized delta learning rule, the output is calculated for the units of each hidden layer and for the units of output layer using the current setting of the weights in all the layers. The optimum weights may be obtained if the weights are adjusted in such a way that the gradient descent is made along the total error surface. Here, we are considering the instantaneous error for each provided input pattern as the squared difference between the desire pattern vector and the actual output from each layer only for the restricted multilayer feed forward neural network, where the number of units in the hidden layer and output layer are same. Thus, for every hidden layer and output layer, we have different squared error. The optimum weights may be obtained for each layer if the weights are adjusted in such a way that the gradient descent is made along the squared error of that layer. It exhibits that we have multiple objective function or error's minima, one each for each layer except the input layer. Hence this problem should be considered as the multi-objective optimization problem. Thus the objective is to obtain the minimum of each instantaneous squared error simultaneously to determine optimum weight vector for the given input pattern. The squared error of the layer will be used to update the weights of the layer and the gradient descent of each error for each layer will obtain at the same time. Thus, at one time there will be more than one gradient descent of individual errors depending on the number of hidden layers. In this approach, the adjustment of weights leading to the units in a hidden layer and also for the units of output layer will proportional to their corresponding gradient descent i.e. the weight change for a layer will guide with the error for the same layer.

The generalized delta learning rule uses the instantaneous distributed squared error for the output and hidden layer, defined as

$$E_p^o = \frac{1}{2}\sum_{k=1}^{K}[d_k^p - S_k(y_{kp}^o)]^2 \tag{1}$$

$$where\ \ y_{kp}^0 = \sum_{j=1}^{j} S_j(y_{jp}^h) w_{kj} \ \ and\ \ S_k(y_{kp}^o) = f(y_{kp}^o) = \frac{1}{1+e^{-y_{kp}^o}}$$

$$E_p^h = \frac{1}{2}\sum_{j=1}^{J}[d_j^p - S_j(y_{jp}^h)]^2 \tag{2}$$

$$where\, y_{jp}^h = \sum_{i=1}^{I} a_i w_{ji} \ \ and\ \ S_j(y_{jp}^h) = f(y_{jp}^h) = \frac{1}{1+e^{-y_{jp}^h}}$$

The change in weights for the units of output layer will determine with gradient descent along the error surface of the error as specified in equation (1). The weight updates for the k^{th} unit of output layer can be represented for the k^{th} pattern at iteration t as:

$$\Delta w_{kj}^p(t) = -\eta_{kj}\frac{\partial E_p^o}{\partial w_{kj}} \tag{3}$$

where η_{kj} is the learning rate for the output layer.

$$= -\eta_{kj}\frac{\partial E_p^o}{\partial y_{kp}^o}\frac{\partial y_{kp}^o}{\partial w_{kj}} = -\eta_{kj}\frac{\partial E_p^o}{\partial y_{kp}^o} S_j(y_{jp}^h)$$

$$= -\eta_{kj}\frac{\partial E_p^o}{\partial S_k(y_{kp}^o)}\frac{\partial S_k(y_{kp}^o)}{\partial y_{kp}^o} S_j(y_{jp}^h)$$

$$= -\eta_{kj}\frac{\partial E_p^o}{\partial S_k(y_{kp}^o)} S_k(y_{kp}^o)\,(1 - S_k(y_{kp}^o))\, S_j(y_{jp}^h)$$

$$= \sum_{k=1}^{K}\eta_{kj}\,[d_k^p - S_k(y_{kp}^o)]\ S_k(y_{kp}^o)\,(1 - S_k(y_{kp}^o))\, S_j(y_{jp}^h) \tag{4}$$

[using equation (1)]

Hence, the new weights for the units of output layer at the iteration (t+1) with momentum term.

$$w_{kj}^p(t+1)$$

$$= \sum_{k=1}^{K}\eta_{kj}\,\delta_k^o\ S_k(y_{kp}^o)\,(1 - S_k(y_{kp}^o))\ S_j(y_{jp}^h) + \alpha\ \ \Delta w_{kj}^p(t-1) \tag{5}$$

where 0 < α < 1 is the momentum constant for the output layer.

The change in weights for the units of hidden layer will determine with gradient descent along the error surface of the error as specified in equation (2). The weight update for the j^{th} unit of hidden layer can be represented for the p^{th} pattern at iteration *t* as:

$$\Delta w_{ji}^p(t) = -\eta_{ji}\frac{\partial E_p^h}{\partial w_{ji}} \tag{6}$$

where η_{ji} is the learning rate for the hidden layers.

$$= -\eta_{ji}\frac{\partial E_p^h}{\partial y_{jp}^h}\frac{\partial y_{jp}^h}{\partial w_{ji}} = -\eta_{ji}\frac{\partial E_p^h}{\partial y_{jp}^h} a_i$$

$$= -\eta_{ji}\frac{\partial E_p^h}{\partial S_j(y_{jp}^h)}\frac{\partial S_j(y_{jp}^h)}{\partial y_{jp}^h} a_i$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

ISSN: 2231-2307, Volume-2, Issue-2, May2012

$$= -\eta_{ji} \frac{\partial E_p^h}{\partial S_j(y_{jp}^h)} S_j(y_{jp}^h)(1-S_j(y_{jp}^h))\, a_i$$

$$= \sum_{j=1}^{J} \eta_{ji} \, [d_k^p - S_j(y_{jp}^h)] \, S_j(y_{jp}^h)(1-S_j(y_{jp}^h))\, a_i$$

[using equation (1)] (7)

Hence the new weight for the units of hidden layer at the iteration (t+1) with momentum term.

$$w_{ji}^p(t+1)$$

$$= \sum_{j=1}^{J} \eta_{ji} \, \delta_j^{hp} \, S_j(y_{jp}^h)(1-S_j(y_{jp}^h))\, a_i + \beta_{\;\Delta w_{ji}^p(t-1)}$$

(8)

where $0 < \beta < 1$ is the momentum constant for the hidden layer.

It can be seen from equation (8) that the less number of terms are appearing for the weights updating in hidden layer with respect to the back-propagation learning rule. Thus, the less computational complexity is involved in this proposed generalized delta learning rule with distributed error for the restricted multilayer feed forward neural network architecture. It provides the fast convergence rate.

B. HYBRID EVOLUTIONARY LEARNING WITH DISTRIBUTED ERROR

Hybrid evolutionary learning is considered as an incorporation of genetic algorithm in restricted multilayer feed forward neural network architecture where the numbers of units in the output layer are equal to the number of units in hidden layer trained with generalized delta learning rule as discussed in subsection 2.1 with limited number of iteration. Thus, the weights between the layers and bias values of the units have been adjusted or updated up to n number of iterations for the given training patterns. It can be seen that for the given set of problem, the gradient descent algorithm does not perform upto non-convergence extends. The genetic algorithm is now employed to evolve the population of sub-optimal weights and bias values. After n iterations, the network does not converge for the given training set but it exhibits the trend in the direction of convergence. Thus, we can consider these weights as the suboptimal values or solution for the given pattern. Normally, the GA considers the initial solution with randomly generated initial population. In this proposed approach, the initial population for the GA will not be generated randomly but it starts with the suboptimal solution as the initial population. Thus, the initial population for the GA will be the values of weights and bias which have been obtained from the execution of neural network upto n iterations for the given training set. Thus, the GA explores from suboptimal solution to multi objective optimal solution for the given problem. The multi objective optimal solution reflects that every layer except input layer has its own different error surface or objective function.

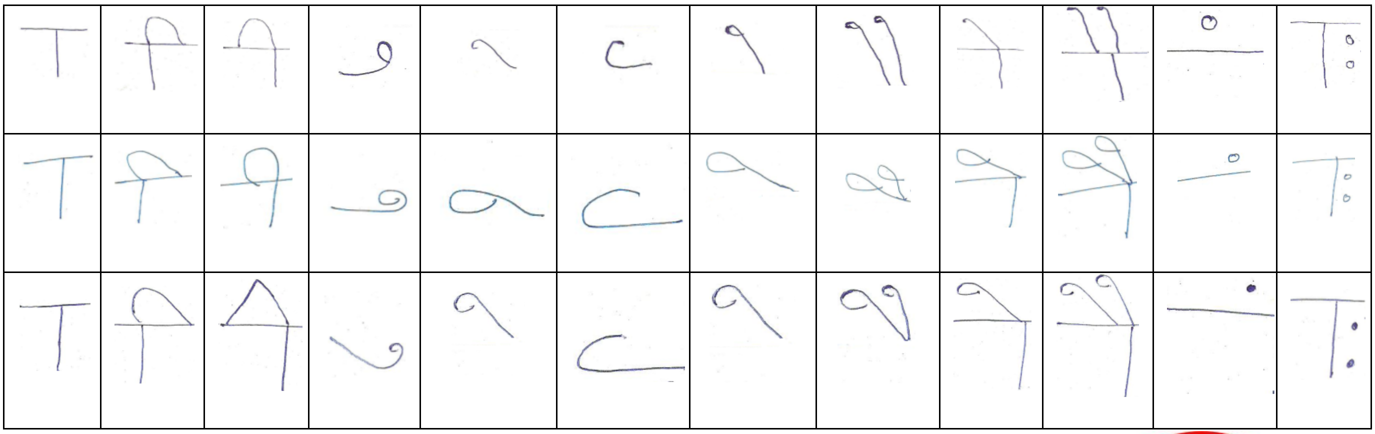
III SIMULATION DESIGN AND IMPLEMENTATION

In this section, we are demonstrating the step by step process for recognition of handwritten Hindi MATRAS.

A. FEATURE EXTRACTION

Feature extraction and selection can be defined as extracting the most representative information from the raw data, which minimizes within class. The pattern variability while enhancing between class pattern variability so that, a set of features are extracted from each class that helps to distinguish it from other classes, while remaining invariant to characteristic differences within the class.

In this paper, we are considering the feature extraction from the input stimuli by using density function of MATLAB, considering the input data in the form of five different sets of each handwritten Hindi '*MATRA'* by five different persons. In this way, we have 60 samples of the '*MATRAS'*. Each sample is scanned and stored as a bit map image. The bitmap image of every sample is now partitioned into 16 equal parts. The row and column wise mean values are calculated and thus we get 16 real number values for each scanned image. Hence, every scanned image is now considered in the form of input pattern vector of 16 dimensions. The scanned images of handwritten Hindi MATRAS of five different persons are shown in figure 1.



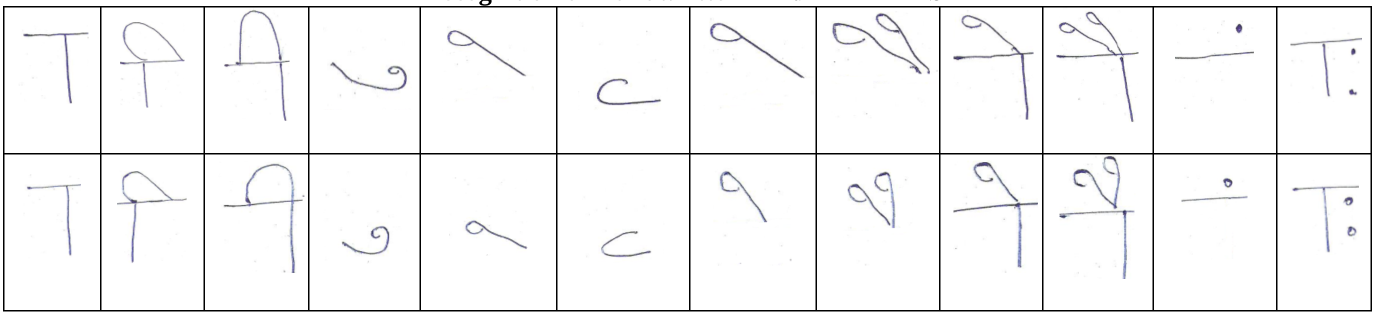


Figure 1: Scanned images of handwritten Hindi *''MATRAS'* by five persons

As each scanned image is partitioned into 16 equal parts, the density values of the pixels for each part are calculated using MATLAB density function. Further, the density values of the centers of gravity for these partitioned images are calculated. Consequently, 16 values obtained from each image. These values then used the input pattern vector for the feed forward neural network. Thus, we have the training set, which consists of sampled patterns of Hindi '*MATRAS'*. Each sample pattern is considered as pattern vector of dimension 16x1 with real number values as shown in figure 2.

	[2.514091; 2.135156; 2.292104; 2.433057; 2.016629; 2.177502; 2.102489; 2.065017; 1.924126; 2.103291; 2.158773; 2.246736; 1.893306; 2.180586; 2.324606; 2.457668]

Figure 2: Input Pattern Vector of order 16x1 for one sample

The output pattern vector corresponds to input pattern vector is considered of dimension 4x1 of the binary values. Since, to distinguish each character set from other character set for classification during the training, the target output is needed. Therefore to classify these '*MATRAS'* of Hindi language, there must be 12 different classes. As we know, the single neuron can differentiate between two classes, we require minimum 4 output neurons to differentiate 12 different classes. Hence, the target output pattern for each input pattern will be of dimension 4, shown in figure 3.

ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ	ं	ः
0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

Figure 3: Target Output pattern for Handwritten Hindi '*MATRAS'*

B. RESTRICTED FEED FORWARD NEURAL NETWORK ARCHITECTURE

We are considering the network as it is restricted in the sense that the number of units in the hidden layer are same as the number of units in the output layer. We get optimal solutions by taking two hidden layers in the restricted feed forward neural network architecture comparing to the results, we get by taking single hidden layer in the network. In figure 4, we have taken the architecture consisting of 16 neurons p_1, p_2, …,p_{16} in the input layer due to the 16 features in the input pattern, 4 neurons in each of the two hidden layers and 4 neurons in the output layer. The first hidden layer has the weight matrix W^{h_1} of order 4x16, the bias vector b^{h_1} of dimension 4 and the activation function f^{h_1}. Now the error E^{h_1} of the first hidden layer will be calculated and adjusted here itself before inputting to the second hidden layer. Next, the second hidden layer will have the weight matrix W^{h_2} of order 4x4, the bias vector b^{h_2} of dimension 4 and the activation function f^{h_2}. Again the error of the second hidden layer E^{h_2} will be calculated and adjusted in the same layer before inputting to the output layer. Finally, the weights given to the output layer will be the matrix W^o of order 4x4, the bias vector b^o of size 4 and the activation function f^o. At last we will be able to get the error of the output layer E^o. After adjustment in the output layer, we get the final output of dimension 4.

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

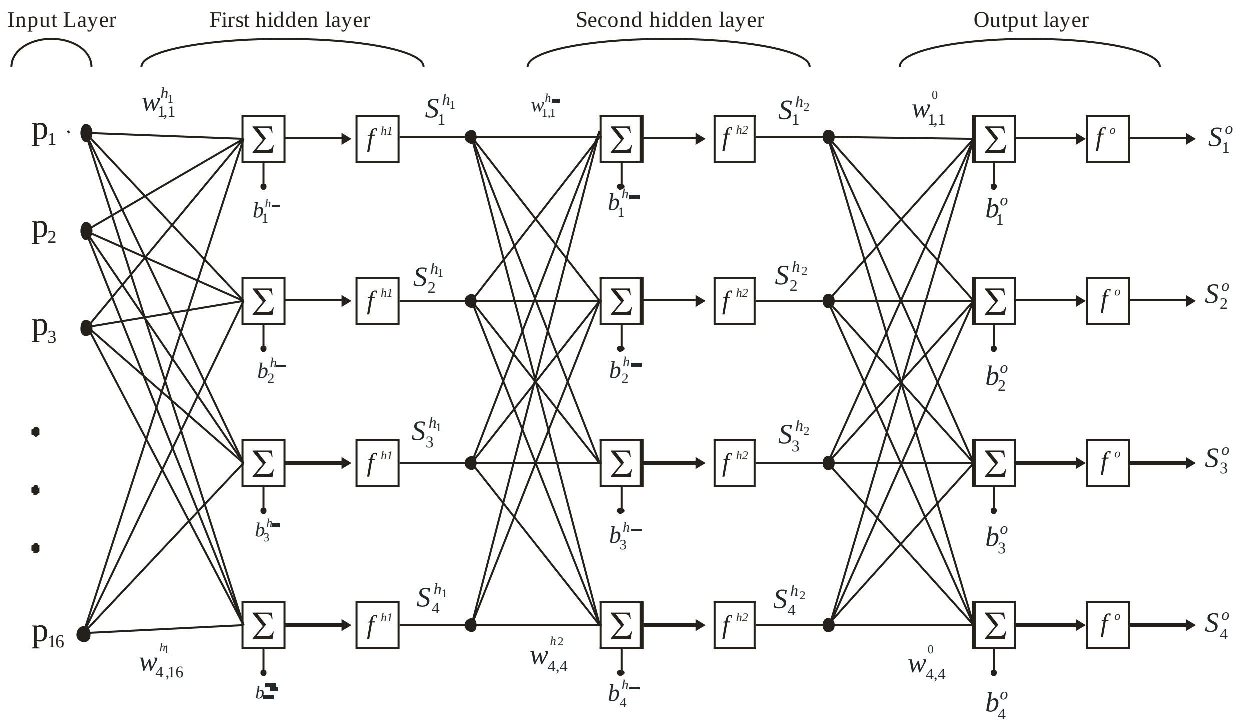


Figure 4: Restricted Feed Forward Neural Network Architecture

The errors at different layers can be calculated as

$$E_p^o = \frac{1}{2}\sum_{k=1}^{K}(d_k^p - S_k(y_{kp}^o))^2 \quad (9)$$

$$E_p^{h_1} = \frac{1}{2}\sum_{g=1}^{G}(d_k^p - S_g(y_{gp}^{h_1}))^2 \quad (10)$$

$$E_p^{h_2} = \frac{1}{2}\sum_{j=1}^{J}(d_k^p - S_j(y_{jp}^{h_2}))^2 \quad (11)$$

where *k*, *g*, *j* are equal in number

The generalized delta learning rule with these distributed errors has performed for t iterations for the given patterns of the training set. After t iterations, the weights and biases for the different layers and units have modified in the direction of convergence. Now, we are employing the genetic algorithm to evaluate the population of weights and biases. In this case, the genetic algorithm starts from the sub-optimal initial solution instead of random initial solution; these values have been updated with generalized delta learning rule upto t iterations.

C. PARAMETERS USED

In our simulation frame work, the various parameters are used as per the nature of learning rules as represented below.

Table 1: Parameters used for Back-Propagation Algorithm

Parameter	Learning rate for output layer (η_o)	Learning Rate for first hidden layer (η_{h_1})	Learning Rate for second hidden layer (η_{h_2})	Momentum Term (α)	Adaptation Rate (k)	Minimum error exist in the network (MAXE)	Initial weights and biased term values
Value	0.01	0.2	0.3	0.9	3.0	0.0001	Between 0 and 1

Table 2: Parameters used for Evolutionary Algorithm

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Parameter	Mutation Probability	Crossover Probability	Mutation Population size	Crossover Population size	Initial Population	Fitness evaluation function	Minimum Error (MAXE)
Value	Less than 0.01	Less than 0.01	3	2000	Between 0 and 1	Back propagated instantaneous squared error E_p	0.00001

Table 3: Parameters used for Generalized Delta Learning with Distributed Error

Parameter	Learning Rate for output layer (η_o)	Learning Rate for hidden layers (η_{h_1}, η_{h_2})	Momentum term for output layer (α)	Momentum term for output layer (β)	Adaptation rate *(K)*	Minimum error for the output layer ($MAXE_O$)	Minimum error for the hidden layer ($MAXE_H$)	Initial weights and biases term values
Value	0.01	0.2	0.9	0.7	3.0	0.0001	0.001	Between 0 and 1

Table 4: Parameters used for Hybrid Evolutionary Learning with Distributed Error

Parameter	**Value**
Mutation Probability	Less than 0.01
Crossover Probability	Less 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 (in case of 16-4-4-4 architecture)	500
Crossover Population size for second hidden layer (in case of 16-4-4-4 architecture)	500
Number of Iteration prior to GA	5000
Initial Population	Values of weights & bias in each sub chromosomes up to 5000 iterations of generalized delta learning rule with distributed error
Fitness Evaluation Function	Distributed instantaneous sum of squared errors E_p^o , $E_p^{h_1}$ *and* $E_p^{h_2}$ as shown in equation (9), (10) & (11)

IV RESULTS & DISCUSSION

The purpose of this simulation is to evaluate the performance of restricted neural network architecture with the four learning methods to perform classification task for the given training set. The methods are i) Evolutionary algorithm i.e. GA with back propagated error, ii) Generalized delta learning with distributed error, iii) Simple GA, iv) Hybrid evolutionary algorithm i.e. GA with generalized delta learning with distributed error. The training set of 60 samples (5 samples of each MATRA) is used to train this neural network. The results are illustrated in table 5. The entries in the table are based on mean values of iterations and counts of convergence weight matrices of five trials with each learning methods for the entire training set. The table of counts is representing the number of optimal solutions i.e. the number of weight matrices on which the convergence is explored for the given training sample. The integer values for the epochs are representing the number of iterations performed by each learning method to classify the samples of handwritten '*MATRAS'* of Hindi language. In this simulation, no case of non-convergence has been found for evolutionary algorithm, and Hybrid evolutionary algorithm with distributed error. Thus, for these algorithms, the weights are successfully converged for all the training samples with more than one count.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 5 : Performance Comparison between various Learning Methods for Hindi MATRAS

‘*MATARS*’	Samples	Hybrid GA with Distributed Error		Hybrid GA with Back Propagated Error		Hybrid GA with Distributed Error on Gradient Descent		Random GA		Hybrid GA with Distributed Error on Gradient Descent & Mutation operator	
		Iteration	**Count**	**Iteration**	**Count**	**Iteration**	**Count**	**Iteration**	**Count**	**Iteration**	**Count**
ा	Sample 1	5	2	2	46	31	6	289	1	22	20
	Sample 2	1	200	1	173	1	250	1	195	1	86
	Sample 3	32	396	702	13	1	284	1	196	1	40
	Sample 4	937	394	1	198	1	278	1	194	1	295
	Sample 5	4	395	1	198	1	290	1	195	1	296
ि	Sample 1	5	201	756	4	3	1	42	1	699	6
	Sample 2	2	196	1	200	1	279	1	190	1	25
	Sample 3	3	200	1	197	1	283	1	195	1	292
	Sample 4	3	195	1	200	1	288	1	193	1	292
	Sample 5	2	196	1	198	1	287	1	195	1	290
ी	Sample 1	13	396	21	1	1	3	44	1	1007	1
	Sample 2	6	392	1	195	1	290	1	195	1	292
	Sample 3	6	5	1	198	1	292	1	195	1	290
	Sample 4	8	393	1	198	1	284	1	191	1	289
	Sample 5	4	392	1	197	1	288	1	198	1	294
ु	Sample 1	9	397	29	2	11	1	23	1	331	3
	Sample 2	1	399	1	147	1	15	1	193	1	293
	Sample 3	2	201	1	198	1	280	1	195	1	291
	Sample 4	5	396	1	196	1	283	1	194	1	287
	Sample 5	6	391	1	199	1	277	1	195	1	297
ू	Sample 1	1	3	31	196	32	8	118	1	1	2
	Sample 2	3	396	1	197	1	287	1	198	1	287
	Sample 3	2	389	1	199	1	290	1	194	1	287
	Sample 4	2	202	1	196	1	291	1	193	1	288
	Sample 5	2	196	1	198	1	297	1	194	1	285
ृ	Sample 1	5	1	434	193	9	1	175	189	14	2
	Sample 2	1	385	1	200	1	296	1	196	1	287
	Sample 3	8	384	1	199	1	291	1	195	1	287
	Sample 4	1	388	1	195	1	285	1	196	1	292
	Sample 5	3	386	1	198	1	290	1	193	1	291

	Sample 1	8	383	1	1	1	4	341	1	491	295
	Sample 2	4	395	1	197	1	295	1	195	1	295
	Sample 3	4	196	1	198	1	292	1	197	1	294
	Sample 4	133	193	1	199	1	297	1	196	1	292
	Sample 5	2	389	1	198	1	293	1	196	1	295
	Sample 1	18	198	25	1	713	2	88	1	9	1
	Sample 2	2	393	1	198	1	282	1	197	1	285
	Sample 3	6	392	1	195	1	281	1	194	1	298
	Sample 4	2	392	1	196	1	282	1	194	1	291
	Sample 5	5	396	1	198	1	287	1	196	1	292
	Sample 1	19	398	2	2	1	7	199	1	466	289
	Sample 2	4	391	1	199	1	288	1	194	1	289
	Sample 3	2	395	1	199	1	287	1	189	1	283
	Sample 4	3	193	1	198	1	292	1	194	1	290
	Sample 5	2	195	1	198	1	291	1	194	1	288
	Sample 1	35	2	86	1	3	1	96	1	808	5
	Sample 2	3	198	1	197	1	291	1	190	1	29
	Sample 3	2	2	1	199	1	288	1	200	1	287
	Sample 4	2	196	1	198	1	294	1	197	1	296
	Sample 5	2	392	1	199	1	287	1	195	1	289
	Sample 1	10	4	1	1	1	5	116	1	1	3
	Sample 2	2	390	1	196	1	293	194	1	1	294
	Sample 3	4	394	1	198	1	298	1	194	1	295
	Sample 4	4	396	1	200	1	290	1	196	1	290
	Sample 5	1	390	1	197	1	294	1	196	1	289
	Sample 1	31	198	254	1	32	5	47	1	702	1
	Sample 2	22	201	1	198	1	289	1	195	1	290
	Sample 3	2	581	1	198	1	287	1	195	1	283
	Sample 4	1	393	1	199	1	290	1	194	1	235
	Sample 5	1	579	1	199	1	288	1	197	1	282

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

V. CONCLUSION

It has been shown in the results that this neural network is trained with two different approaches for the error. The error of output layer is not back-propagated but instead of this, error is calculated for each unit of hidden layers individually. Hence, the same desired response is considered for each layer but the actual response of units of output layer is used only for output layer and actual output of units of hidden layers is used only for respective hidden layers. Thus, the error is considered as distributed error. The proposed study is exhibiting the performance evaluation for both the approaches i.e. multi objective criteria of convergence and single objective criteria of convergence for the classification of training set consisting of handwritten '*MATRAS'* of Hindi language.

REFERENCES

[1] Du K.L. and Swamy, M.N.S. (2006)," Neural Networks in a Soft computing Framework", Springer-Verlag London Limited

[2] Hagan, M.T., Demuth, H.B. and Beale, M.H. (1996), "Neural Network Design," PWS Publishing Co., Boston, MA

[3] Kim, H.B., Jung, S.H., Kim, T.G., and Park, K.H. (1996), "Fast Learning method for back propagation neural network by evolutionary adaption of learning rates", Neurocomputing, vol. 11(1), pp. 101-106

[4] Mangal M. and Singh,M.P. (2007), 'Analysis of multidimensional XOR classification problem with evolutionary feed-forward neural networks.", International Journal on Artificial Intelligence Tools , volume 16(1),pp. 111-120.

[5] Mangal,M. and Singh,M.P. (2007), "Analysis of pattern classification for the multidimensional parity-bit-checking problem with hybrid evolutionary feed-forward neural network.", Neurocomputing, volume 70,pp. 1511-1524.

[6] Mangal,M. and Singh,M.P. (2006), "Handwritten English vowels Recognition using Hybrid Evolutionary Feed-Forward Neural Networks.", Malaysian Journal of computer science, vol. 19(2), pp. 169-187.

[7] Pradeep J, Srinviasan E & Himavathi S (2010) "Diagonal Feature Extraction Based Handwritten Character System Using Neural Network", International Journal of Computer Applications Volume 8– No.9

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

[9] Shrivastava, S. and Singh, M.P. (2011), "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

[10] Shrivastava, S., and Singh, M. P. (2007), " Analysis of soft computing techniques for minimizing the problem of local minima in back-propagation for handwritten English alphabets", Proc Int. Con. of soft computing and intelligent systems, vol. 2, pp. 307-313

Dr Meenu Gupta, Director at G L Bajaj Group of Institutions, Mathura has earned Ph.D in the field of Information Theory, M Tech in Computer Science & Engineering and MBA in Operations Management. Prior to this, she has dispensed her administrative accumen as a Director for 8 years at Rajiv Academy for Technology & Management, Mathura. Her scholastic pool is filled with a vast of Hand-on Experiences of 20 years. She is credited with the publications of two books. Along with this, she has also published a number of research papers, articles in the journals of repute. Dr Gupta has also put forth her innovative and pioneering research works through participating in various national and international conferences. She has visited many countreis regarding research and education. Remarkably, Dr Gupta is enjoying the membership of a number of national & international societies. Markedly, Dr Gupta is also providing her astute directions as a Chief Editor to two International Journals, ***Technohub****-an International Journal(ISSN 2277-7008)* and ***Knowledge Hub****-an International Journal of Management (ISSN 0973-6425).* As well, she is a member of Editorial Advisory Board of various journals & conferences at national & international level. As a True depiction of her vast expertise, Dr Gupta has played an instrumental role in organizing international and national conferences in engineering & management, MDPs and workshops. Impressed with her noble contributions in the field of higher education, Dr Meenu Gupta has also been honoured with number of awards.

Prof (Dr) Ajay Rana, Director at Amity University has had an outstanding academic record (Two Time Gold Medalist – Throughout First Class) and is a product of prestigious system throughout. He obtained Ph.D in Computer Science and Engineering and Master of Technology (M.Tech) in Computer Science and Engineering from Uttar Pradesh Technical University and Kurukshetra University respectively. He has visited many Top Worlds' Universities w.r.t Research and Education. He has published more than 72 Research Papers in reputed Journals and Proceedings of International and National Conferences. He has co-authored four books and co-edited 18 conference proceedings. He has delivered Invited lectures in more than 36 Technical and Management Workshop / Conferences programs in India and abroad. He is a member of Board of Governance (BOG), Advisory Council (AC), Academic Executive Member, Board of Studies (BOS) and Special Member of many Indian and Foreign Universities as well as Industry. He is Editor in Chief, Technical Committee Member, Advisory Board Member for 15 Plus Technical Journals and Conferences at National and International Levels. Dr. Ajay Rana has received a number of Awards and Honors and was recipient of National Merit Scholarship.