A Modified Optical Backpropagation Neural Network Model in Online Handwritten Character Recognition System

*Fenwa O.D., Omidiora E. O., Fakolujo O. A., Ajala F. A., Oke A.O
Department of Computer Science and Engineering,
LadokeAkintola University of Technology, P.M.B 4000, Ogbomoso, Nigeria.
Tel: +2348163706286
(* Corresponding Author)

Abstract

Neural networks are powerful data modelling tool that are able to capture and represent complex input/output relationships.. Though they yield high accuracy, most of the neural networks are computationally heavy due to their iterative nature. Hence, there is a significant requirement for a neural classifier which is computationally efficient and highly accurate. Backpropagation algorithm has been reported to be an effective and most widely used supervised training algorithm for multi-layered feedforward neural networks but has the shortcomings of longer training time and entrapment into a local minimal. Several research works have been proposed to improve this algorithm but some of these research works were based on 'learning parameter' which in some cases slowed down the training process. Hence, this paper has focused on alleviating the problem of standard backpropagation algorithm based on 'error adjustment'. To this effect, a 'Modified Optical Backpropagation (MOBP)' neural network was employed in developing an online handwritten character system which proves to be faster than the existing Optical Backpropagation in literature.

Keywords: Artificial Neural Network, Optical Backpropagation, Modified Optical Backpropagation, Character Recognition, Feature Extraction, Mean Square Error and Cubic Error

1. Introduction

As technology advances, computer systems become an invaluable asset of humans. It enhances communication and increases efficiency. Computer systems are greatly influencing the lives of human beings and their usage is increasing at a tremendous rate. As computer systems become increasingly integrated into our everyday life, it is therefore necessary to make them more easily accessible and user friendly. The ease with which we can exchange information between user and computer is of immense importance today because input devices such as keyboard and mouse have limitations. Owing to these limitations, researchers for over decades have been attracted to device a quick and natural way of communication between computer systems and human beings (Anita and Dayashankar, 2010).The field of personal computing has begun to make a transition from the desktop to handheld devices, thereby requiring input paradigms that are more suited for single hand entry than a keyboard. Online handwriting recognition allows for such input modalities. Handwriting processing is a domain in great expansion. The interest devoted to this field is not explained only by the exciting challenges involved, but also the huge benefits that a system, designed in the context of a commercial application, could bring (Morita et. al., 2003).

The online methods have been shown to be superior to their offline counterpart in recognizing handwritten characters due the temporal information available with the former (Pradeep et al., 2011). Handwriting recognition system can further be broken down into two categories: writer-independent recognition system which recognizes wide range of possible writing styles and a writer-dependent recognition system which recognizes writing styles only from specific users (Santosh and Nattee, 2009).Neural network classifiers exhibit powerful discriminative properties and features such as ability to be trained automatically from examples, good performance with noisy data, parallel implementation and efficient tools for learning large databases. Hence, it has been popularly and extensively used in handwriting recognition (Jude, Vijila and Anitha, 2010).

2. Research Methodology

In this paper, four stages of development of the proposed character recognition system which include; data acquisition, character processing which consists feature extraction and character digitization, training and classification using modified Optical Backpropagation neural network model and testing werepresented as shown in Figure 2.1.

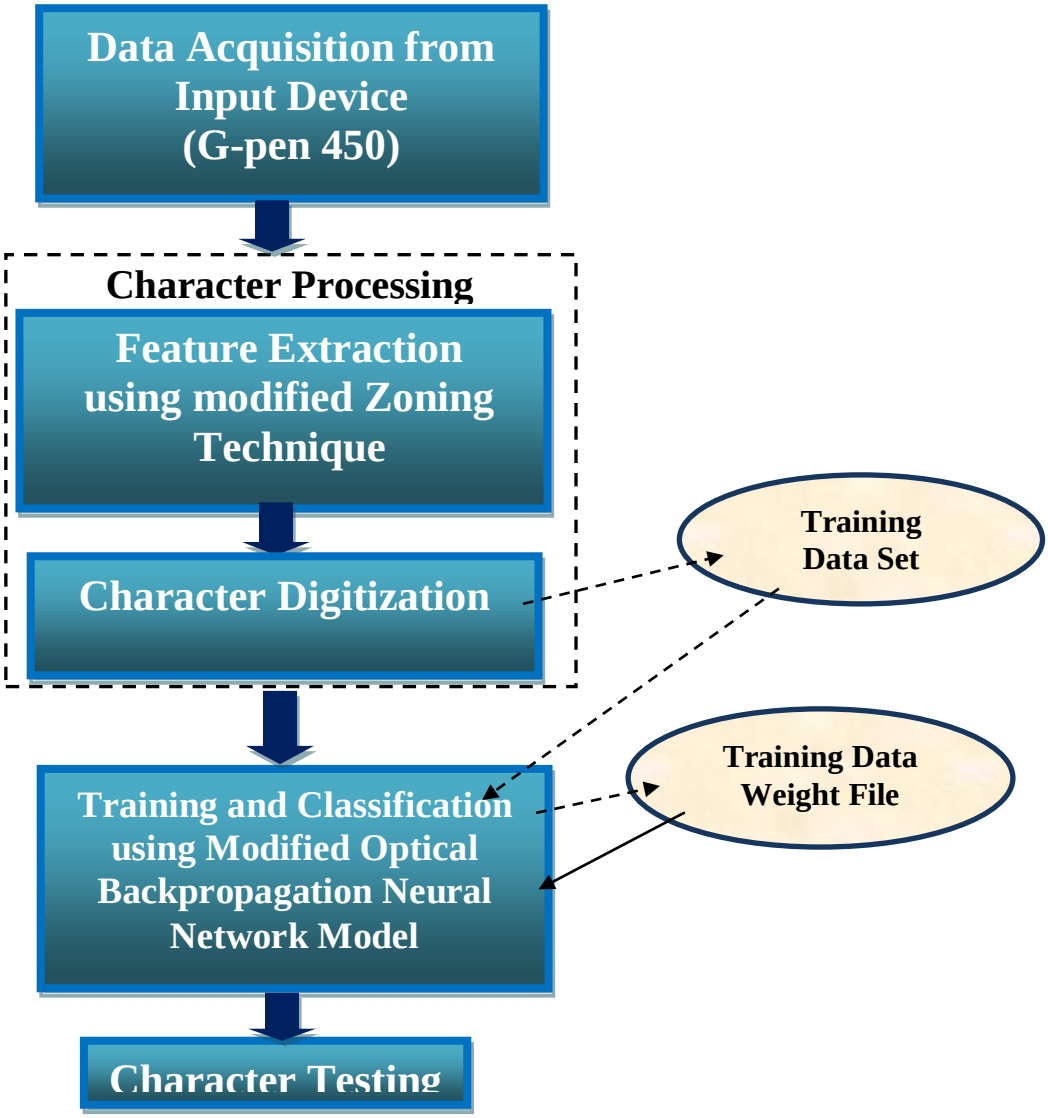


Figure 2.1: Block Diagram of the proposed Character Recognition System

Experiments were performed with 2,000 handwriting character samples (uppercase (A-J)English alphabet) collected from 100 subjects using G-Pen 450 digitizer and the system was tested with 100 character samples written by people who did not participate in the initial data acquisition process. The performance of the system was evaluated based on the training time and the number of epochs required totrain the network. The results of the proposed system showed better training and recognition performances compared with similar work in literature.

Character Acquisition and Feature Extraction

Character Acquisition

The data used in this work were collected using Digitizer tablet (G-Pen 450). The G-Pen has an electric pen with sensing writing board. An interface was developed using C# to acquire data (character information) such as stroke number and pressure of the stroke from different subjects using the Digitizer tablet. Characters considered were the first 10 upper case (A-Z)English alphabets. 2000 characters (10 x 2 x 100) were collected from 100 subjects as each individual was requested to write each of the characters twice (this was done to allow the network learn various possible variations of a single character and become adaptive in nature). This serves as the training data set which was the input data that was fed into the neural network. Samples of characters collected were as shown in Figure 2.2

Figure 2.2: Sample handwritten characters

Proposed Feature Extraction Method

For extracting the feature, the modified hybrid zone based feature extraction is used. The major advantage of this approach stems from its robustness to small variation, ease of implementation and provides good recognition rate. Zone based feature extraction method provides good result even when certain pre-processing steps like filtering, smoothing and slant removing are not considered.

The Zoning Algorithm

In this paper, a hybrid of modified Image Centroid and Zone-based (ICZ) and Zone Centroid and Zone-based (ZCZ) distance metric feature extraction algorithm proposed by Fenwa, Omidiora and Fakolujo (2012). Modifications of the two algorithms were in terms of:

(i) Number of zones being used
(ii) Measurement of the distances from both the Image Centroid and Zone Centroid
(iii)The area of application.

Few zones were adopted in their work but emphasis was being laid on how to effectively measure the pixel densities in each zone. However, pixels pass through each of the zones at varied distances that was why an average of five distances was measured for each of the zones at an angle of 20^0.

Hybrid of the Modified Zoning Feature Extraction Algorithms

The following were the algorithms to show the working procedure of the modified hybrid zoning feature extraction methods:

Modified Algorithm1: Image Centroid and Zone-based (ICZ) distance metric feature extraction algorithm (Fenwa, Omidiora and Fakolujo, 2012)

Input: Pre-processed character images

Output: Features for classification and recognition

Method Begins

Step 1: Divide the input image in to 25equal zones as shown in Figure 2.3

Step 2: Compute the input image centroid as shown in Figure 2.4 using the formula:

$$\text{Centre of gravity in the horizontal direction (x–axis)} = \sum_{n=0}^{n-1} x_i \text{, where n = width} \quad (2.1)$$

$$\text{Centre of gravity in the vertical direction (y–axis)} = \sum_{m=0}^{m-1} y_i \text{, where m = height} \quad 2.2)$$

Step 3: Compute the distance between the image centroid to each pixel present in the zone as shown in Figure 2.4

Step 4: Repeat step 3 for the entire pixel present in the zone (five points in this case):

$$d = d_1 + d_2 + d_3 + d_4 + d_5 \quad (2.3)$$

Step 5: Compute average distance between these points as:

$$\text{Average Image Centroid Distance } D_I = d/5 \quad (2.4)$$

where d = total distance between the image centroid to the pixel measured at an of angle 20^0

Step 6: Repeat this procedure sequentially for the entire zone (25 zones).

$$\text{Total Distance (P)} = D_{I1} + D_{I2} + D_{I3} + \ldots + D_{Im} \quad (2.5)$$

$$\text{Total Average Distance } j = \sum_{z=0}^{m-1} \frac{D_Z}{m} \quad (2.6)$$

where m = 25 (total number of zones)

Step 7: Finally, 25such features was obtained for classification and recognition.

Method Ends.

Modified Algorithm2: Zone Centroid and Zone-based (ZCZ) distance metric feature extraction algorithm (Fenwa, Omidiora and Fakolujo, 2012)

Input: Pre-processed character image

Output: Features for classification and recognition

Method Begins

Step 1: Divide the input image in to 25equal zones as shown in Figure 2.3

Step 2: Compute the zone centroid for the entire pixel present in the zone as shown in Figure 2.5 using the formula:

Centre of gravity in the horizontal direction (x–axis) = $\sum_{n=0}^{n-1} x_i$, where n = width

(2.7)

Centre of gravity in the vertical direction (y–axis) = $\sum_{m=0}^{m-1} y_i$, where m = height

(2.8)

Step 3: Compute the distance between the zone centroid to each pixel present in the zone.

Step 4: Repeat step 3 for the pixel present in a zone (5 points in this case).

Total Distance $D = D_1 + D_2 + D_3 + D_4 + D_5$

(2.9)

Step 5: Compute average distance between these points as:

Average distance $D_Z = D/5$

(2.10)

where D = distance between the zone centroid measured at angle 20^0 to the pixel in the zone.

Step 6: Repeat this procedure sequentially for the entire 25 zones.

Total Distance (Q) = $D_{Z1} + D_{Z2} + D_{Z3} + \ldots + D_Z m$

(2.11)

Total Average Distance $k = \sum_{z=0}^{m-1} \frac{D_Z}{m}$

(2.12)

where m = 25 (total number of zones)

Step 7: Finally, 25such features was obtained for classification and recognition.

Method Ends

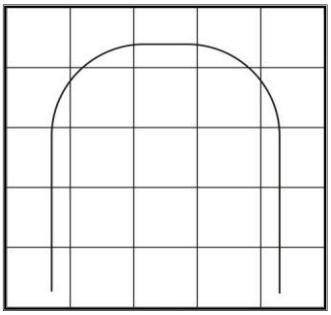


Figure 2.3: Character ‘n’ in 5 by 5 (25 equal zones)

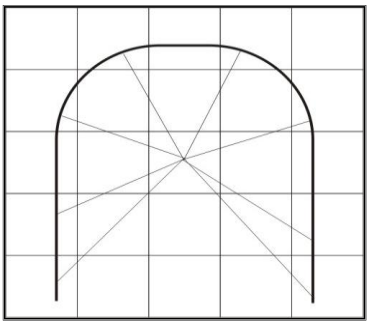


Figure 2.4:Image Centroid of character ‘‘n’’ in zoning

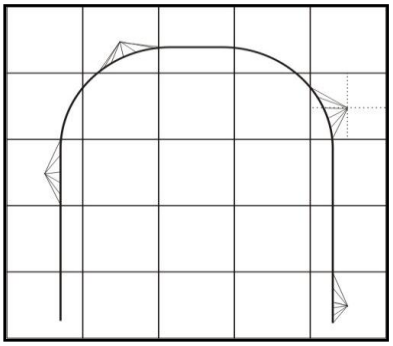


Figure 2.5: Zone Centroid of character ‘n’ in zoning

The Hybrid Zoning Algorithm: Hybrid of Modified ICZ and Modified ZCZ
(Fenwa, Omidiora and Fakolujo, 2012)

Input: Pre-processed character image
Output: Features for Classification and Recognition
Method Begins
Step 1: Divide the input image into 25equal zones.
Step 2: Compute the input image centroid
Step 3: Compute the distance between the image centroid to each pixel present in the zone.
Step 4: Repeat step 3 for the entire pixel present in the zone.
Step 5: Compute average distance between these points.
Step 6: Compute the zone centroid
Step 7: Compute the distance between the zone centroid to each pixel present in the zone.
Step 8: Repeat step 7 for the entire pixel present in the zone
Step 9: Compute average distance between these points.
Step 10: Repeat the steps 3-9 sequentially for the entire zones.
Step 11: Finally, **2xn (50)** such features were obtained for classification and recognition.
Method Ends

Digitization

Digitization of an image into a binary matrix of specified dimensions makes the input image invariant of its actual dimensions. Hence an image of whatever size gets transformed into a binary matrix of fixed pre-determined dimensions. This establishes uniformity in the dimensions of the input and stored patterns as they move through the recognition system. It is the conversion of the features selected in the canvas into binary digit which is then fed into the input layer of the neural network**.** The fact that all Neural Networks take numeric input and produces alphanumeric output inspired the idea that the data gathered for the Neural Network be represented numerically. The data were carefully scaled to enable the network to learn. If a white area is found in input character matrix then it will mark the corresponding pattern vector element as 0 (zero) and for black area it will mark as 1 (one). A method known as one-of-N encoding, which employs the use of numeric variables to represent a single nominal variable (e.g. character 'A' which is represented with the bits (000000011110000000..111000000000000111) was employed in the data representation for the network.

3. Related Work

Backpropagation Training Algorithm

Backpropagation is a systematic method for training multiple-layered (two or more) artificial neural networks.The standard Backpropagation algorithm is as given below:

1. Build a network with the chosen number of input, hidden and output units.
2. Initialize all the weights to low random values.
3. Choose a single training pair at random.
4. Copy the input pattern to the input layer
5. Cycle the network so that the activations from the inputs generate the activations in the hidden and output layers
6. Calculate the error derivative between the output activation and the target output.
7. Back propagate the summed products of the weights and errors in the output layer in order to calculate the error in the hidden units.

8. Update the weights attached to each unit according to the error in that unit, the output from the unit below it, and the learning parameters, until the error is sufficiently low or the network settles.

The purpose of training is to adjust the weights so that the application of a set of inputs produces the desired outputs. To accomplish this, the network is usually trained with a large number of input-output pairs. The training procedure of Backpropagation algorithm was described by Freeman and Skapura's book (1992) and given as follows:

Backpropagation Training Algorithm:

Assume there are *m* input units, *n* hidden units, and *p* output units.

1. Apply the input vector, $X_p = (X_{pl}, X_{p2}, X_{p3}, ..., X_{pm})$to the input units.
2. Calculate the net- input values to the hidden layer units:
   $net^h_{pj} = (\sum_{i=1}^{n} W^h_{ji} . X_{pi})$
   (2.13)
3. Calculate the output from the hidden layer:
   $_{ipj} = f^h_j(net^h_{pj})$
   (2.14)
4. Move to the output layer. Calculate the net-input values to each unit:
   $net^o_{pk} = (\sum_{j=1}^{L} W^o_{kj}.i_{pj})$
(2.15)
5. Calculate the outputs:
   $O_{pk} = f^o_j(net^o_{pk})$
   (2.16)
6. Calculate the error terms for the output units:
   $\delta^o_{pk} = (Y_{pk} - O_{pk}).f^{o'}_k(net^o_{pk})$
(2.17)
   where, $f^{o'}_k(net^o_{pk}) = f^o_k(net^o_{pk}).(1- f^o_k(net^o_{pk}))$
   (2.18)

7. Calculate the error terms for the hidden units
   $\delta^h_{pj} = f^{h'}_j(net^h_{pj}) . (\sum_{k=1}^{M} \delta^o_{pk} .W^o_{kj})$
   (2.19)
8. Update weights on the output layer
   $W^o_{kj}(t+1) = W^o_{kj}(t) + (\eta .\delta^h_{pk} . i_{pj})$
   (2.20)
9. Update weights on the hidden layer
   $W^h_{ji}(t+1) = W^h_{ji}(t) + (\eta .\delta^h_{pj} . X_i)$
   (2.21)

The description of notation used in training procedure in 3.7.2 is as given below:

X_{pi} net input to the *i*th input unit

net^h_{pj} net input to the *j*th hidden unit

W^h_{ji} weight on the connection from the *i*th input unit to *j*th hidden unit

i_{pj} net input to the *j*th hidden unit

net^o_{pk} net input to the *k*th output unit

W^o_{kj} weight on the connection from the *j*th hidden unit to *k*th output unit

O_{pk} actual output for the *k*th output unit

Y_{pk} desired output for the *k*th output unit

f Sigmoid activation function

f ' derivative of activation function

δ^{o}_{pk} signal error term for the *k*th output unit

δ^{h}_{pj} signal error term for the *j*th output unit

t_1 desired output

y_1 network output

η learning rate

Optical Backpropagation Algorithm

The difficulty encountered in the standard Backpropagation algorithm is when the actual value $f^{o}_{k}(net^{o}_{pk})$ approaches either extreme value, the factor $f^{o}_{k}((net^{o}_{pk}) . (1 - f^{o}_{k}(net^{o}_{pk})))$ makes the error signal very small. The Optical Backpropagation algorithm (an enhanced Backpropagation) proposed by Otair and Salameh (2005) focused on this delay of the convergence that is caused by the derivative of the activation function. However, a slight modification of the error signal function of the standard Backpropagation algorithm has resolved this shortcoming and indeed greatly accelerates the convergence to a solution. The adjustment of the new algorithm (OBP) is described to improve the performance of the *BP* algorithm. The convergence speed of the training process was improved significantly by OBP through maximizing the error signal, which was transmitted backward from the output layer to each unit in the intermediate layer.

In *BP*, the error at a single output unit is defined according to equation (2.17) as:

$\delta^{o}_{pk} = (Y_{pk} - O_{pk}) . f^{o'}_{k}(net^{o}_{pk})$ where the subscript "p" refers to the p_{th} training vector, and "k" refers to the k_{th} output unit. In this case, Y_{pk} is the desired output value, and O_{pk} is the actual output from k_{th} unit, then δ^{o}_{pk} will propagate backward to update the output-layer weights and the hidden-layer weights while the error at a single output unit in adjusted OBP is given as (Otair, &Salameh, 2005):

$$\text{New } \delta^{o}_{pk} = (1+e^{(Ypk - Opk)2} . f^{o'}_{k}(net^{o}_{pk})), \quad \text{if } (Y - O) >= \text{zero} \quad (2.22a)$$

$$\text{New } \delta^{o}_{pk} = - (1+e^{(Ypk - Opk)2} . f^{o'}_{k}(net^{o}_{pk})), \quad \text{if } (Y - O) < \text{zero} \quad (2.22b)$$

An OBPuses two forms of Newδ^{o}_{pk}, because the *exponential* function always return *zero* or *positive* values, while adapts operation for many output units need to decrease the actual outputs rather than increasing it. The New$\delta^{o}_{pk}$was propagated backward to update the output-layer weights and the hidden-layer weights. This Newδ^{o}_{pk} minimized the errors of each output unit more quickly than the old δ^{o}_{pk}, and the weights on certain units change very *large* from their starting values.

The steps of an OBP (Otair and Salameh, 2005)

1. Apply the input example to the input units.
2. Calculate the net-input values to the hidden layer units.
3. Calculate the outputs from the hidden layer.

4. Calculate the net-input values to the output layer units
5. Calculate the outputs from the output units
6. Calculate the error term for the output units, using the Newδ^{o}_{pk} (using equations 2.22a and 2.22b) instead of δ^{o}_{pk} in equation **(**2.17)
7. Calculate the error term for the hidden units, through applying Newδ^{o}_{pk}, also
   Newδ^{h}_{pj}= $f^{h'}_{j}(net^{h}_{pj})$. $(\sum_{k=1}^{M} New\delta^{o}_{pk} . W^{o}_{kj})$
   (2.23)
8. Update weights on the output layer.
   $W^{o}_{kj}(t+1) = W^{o}_{kj}(t) + (\eta . New\delta^{o}_{pk} . i_{pj})$
   (2.24)
9. Update weights on the hidden layer.
   $W^{h}_{ji}(t+1) = W^{h}_{ji}(t) + (\eta . New\delta^{o}_{pj} . X_i)$
   (2.25)
10. Repeat steps from step 1 to step 9 until the error $(Y_{pk} - O_{pk})$ is acceptably small for each training vector
    pairs.

The proposed algorithm as classical BP is stopped when the squares of the differences between the actual and target values summed over units and all patterns are acceptably small.

Modified Optical Backpropagation Neural Network

The error function defined in Optical Backpropagation earlier is proportional to the square of the Euclidean distance between the desired output and the actual output of the network for a particular input pattern. As an alternative, other error functions whose derivatives exist and can be calculated at the output layer can replace the traditional square error criterion (Haykin, 2003). In this paper, error of the third order (Cubic error) has been adopted to replace the traditional square error criterion used in Optical Backpropagation. The equation of the cubic error is given as:

$\delta^{o}_{pk} = -3(Y_{pk} - O_{pk})^2 . f^{o'}_{k}(net^{o}_{pk})$
(2.26)

The cubic error in equation (2.26) was manipulated mathematically in order to further maximize the error signal of each output units which were transmitted backward from the output layer to each unit in the intermediate layers. The derived equations are as shown in equations (2.27a) and (2.27b) below:

Modified $\delta^{o}_{pk} = 3((1+ e^{t})^2 . f^{o'}_{k}(net^{o}_{pk}))$ If $(Y_{pk} - O_{pk})^2 >= 0$
(2.27a)

Modified $\delta^{o}_{pk} = -3((1+ e^{t})^2 . f^{o'}_{k}(net^{o}_{pk}))$ If $(Y_{pk} - O_{pk})^2 <= 0$
(2.27b) Where Y_{pk} = Target or Desired output
O_{pk} = Network output
$t = (Y_{pk} - O_{pk})^2$

However, one of the ways to reduce the training time is through the use of momentum, as it enhances the stability of the training process. The momentum is used to keep the training process going in the same general direction (Haykin, 2003). In the modified Optical Backpropagation network, momentum was introduced. Hence, equation (3.30) becomes

$W^{o}_{kj}(t+1) = W^{o}_{kj}(t) + \mu W^{o}_{kj}(t) + (\eta .$ Modified $\delta^{o}_{pk} . i_{pj})$
(2.28)

where μ is the momentum coefficient typically about 0.9 and η is the learning rate.

The Modified Optical Backpropagation Algorithm:

Modification of the algorithm is in terms of Error signal function.

With the introduction of Cubic error function and Momentum, the modified Optical Backpropagation is given as:

1. Apply the input example to the input units.
2. Calculate the net-input values to the hidden layer units.
3. Calculate the outputs from the hidden layer.
4. Calculate the net-input values to the output layer units
5. Calculate the outputs from the output units
6. Calculate the error term for the output units, using equation (2.27a) and (2.27b)instead of equations (2.22a) and (2.22b)
7. Calculate the error term for the hidden units, through applying improve δ^{o}_{pk} also Modified $\delta^{h}_{pj} = f^{h'}_{j}(net^{h}_{pj}) .(\sum_{k=1}^{M} \text{Modified}\delta^{o}_{pk} . W^{o}_{kj})$ (2.29)
8. Update weights on the output layer. $W^{o}_{kj}(t+1) = W^{o}_{kj}(t) + \mu W^{o}_{kj}(t) + (\eta . \text{Modified } \delta^{o}_{pk} .i_{pj})$ (2.30)
9. Update weights on the hidden layer. $W^{h}_{ji}(t+1) = W^{h}_{ji}(t) + (\eta . \text{Modified } \delta^{h}_{pj} . X_i)$ (2.31)

Repeat steps from step 1 to step 9 until the error ($Y_{pk} - O_{pk}$) is acceptably small for each of the training vector pair. The proposed algorithm as OBP is stopped when the cubes of the differences between the actual and target values summed over units and all patterns are acceptably small.

.

4. Results and Discussion

The system was evaluated on samples taken from individuals who did not participate in the initial process of data acquisition for the training data set. This was done keeping in view the eventual aim of using the model in practical online recognition system. The experimental results, as shown in table 2.1 revealed that modified Optical Backpropagation was able to train the network with fewer numbers of epochs compared with the existing Optical Backpropagation. However, the results showed that learning rate parameter variation had an effect on the network performance, the smaller the value of learning rate, the higher the number of epochs and the higher the recognition accuracy because weight updates were done in a more refined manner. In table 2.2, the number of epochs required in MOBP to achieve a minimum error of 0.00706 is smaller than that of OBP. Bar Chart representation and Graphical representation of the results are as shown in Figures 2.6 and 2.7. The quality of the developed online handwriting recognizer is also related to its ability to translate drawn characters irrespective of writing styles.

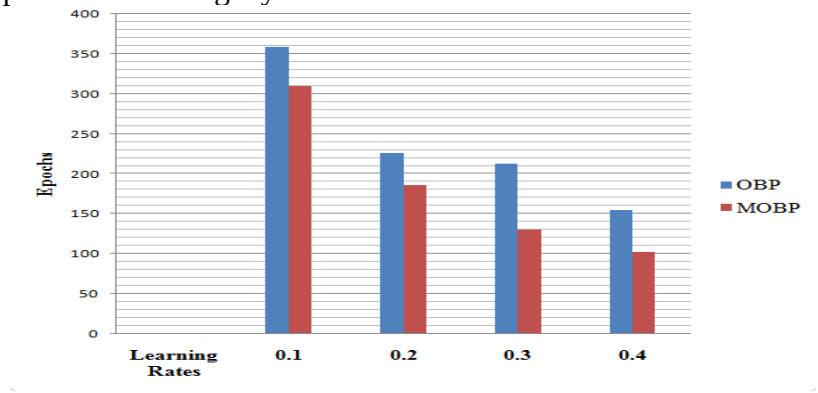
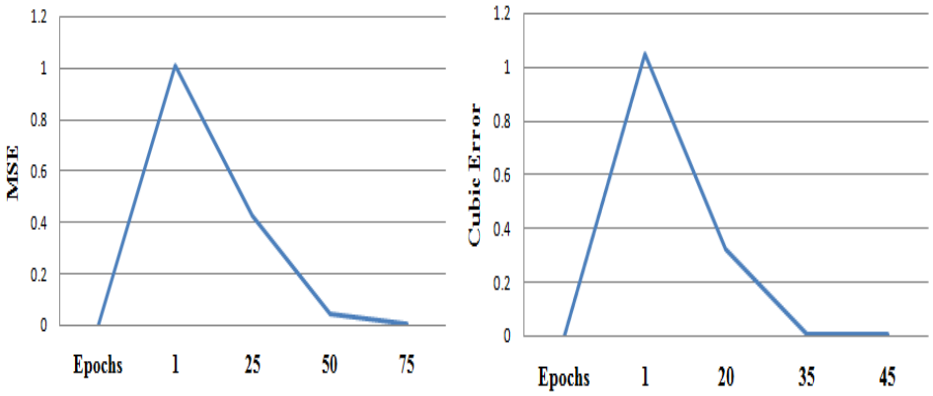


Figure 2.6: Bar Chart representation of performance comparison of OBP and MOBP of Table 2.1



(a) (b)

Figure 2.7: (a) Decrease in MSE using OBP to achieve MSE= 0.00706 with 75 epochs. (b) Decease in Cubic Error using MOBP to achieve Cubic Error= 0.00706 with 45 epochs.

Table 2.1: Experimental results of number of Epochs required for training OBP and MOBP using variants of learning rate.

	OBP	**MOBP**
Learning Rates	**Epochs**	**Epochs**
0.1	359	310
0.2	226	186
0.3	213	130
0.4	155	102

Table 2.2: Convergence of error in training the network with a learning rate of 0.1.Using OBP the MSE was 0.00706 after 75 epochs while MOBP required 45 epochs to achieve 0.00706 cubic error.

OBP		MOBP	
Epochs	MSE	Epochs	Cubic Error
1	1.00705	1	1.04713
25	0.42370	20	0.32510
50	0.04571	35	0.00967
75	0.00706	45	0.00706

5. Conclusion

In conclusion, the result of this work showed better performances in terms of training epochs and training timed speed compared with existing work in literature. Future work can explore the necessity to integrate into the learning algorithm, an optimization algorithm to further enhance the performance of the system.

6. References

Ahmed, S.M. (1995): ''Experiments in character recognition to develop tools for an optical character recognition system'', IEEE Inc. First National Multi Topic Conference Proceedings, NUST, Rawalpindi, Pakistan, 61-67.

Anita, P. And Dayashankar S. (2010): ''Handwritten English Character Recognition usingNeural Network'',
International Journal of Computer Science and Communication, 1(2): 141-144.

Fenwa O.D., Omidiora E.O., Fakolujo O.A. (2012): ''Development of a Feature ExtractionTechnique for Online
Character Recognition System'', Journal of Innovation System Design and Engineering, 3(3):10-23.

Haykin, S. (2003): Neural Networks: A Comprehensive Foundation, PHI, New –Delhi, India

Hecht-Nielsen, R. (1990): Neurocomputing: Addison-Wesley Publishing Company.

Jude, D. H, Vijila, C. K. andAnitha, J (2010): ''Performance Improved PSO Based ModifiedCounterpropagation
Neural Network for Abnormal Brain Image Classification'',International Journal
Advanced SoftComputing Application, 2(1): 65-84.

Morita M., Sabourin R., Bortolozzi F., Suen C. Y. (2003): '' ARecognition and Verification
Strategy for Handwritten WordRecognition'', ICDAR'03), Edinburgh-Scotland: 482-486.

Otair, M.A. and Salameh, W.A. (2005): '' Speeding up Backpropagation Neural Network'' Proceedings 0f the
2005 Informing Science and IT Education Joint Conference 2: 167-173.

Pradeep, J., Srinivasan, E. and Himavathi, S. (2011): ''Diagonal Based Feature Extraction forHandwritten
Alphabets Recognition using Neural Network'', International Journal of Computer Science and Information
Technology (IJCS11), 3(1): 27-37.

Santosh, K.C. and Nattee, C. (2009): ''A Comprehensive Survey on Online Handwriting Recognition Technology
and Its Real Application to the Nepalese Natural Handwriting'', Kathmandu University Journal of Science,
Engineering Technology, 5(1): 31-55.