International Journal of Information Technology and Knowledge Management
July-December 2010, Volume 2, No. 2, pp. 645-649

Totally Unconstrained Handwritten Character Recognition using Improved BP Neural Network

Amit Choudhary[1], Rahul Rishi[2], Savita Ahlawat[3] & Vijaypal Singh Dhaka[4]

In spite of fast evolution of electronic techniques in the past few decades, a number of large scale applications rely on the use of paper for storing important information or data. Here we examined the issues of optical character recognition (OCR) system for reading the data printed on the physical paper document using artificial neural network. We proposed an approach in which Back Propagation Algorithm is deeply investigated and as a result, the momentum term of the BP Algorithm was modified by introducing an additional momentum term. This term improved the process of adjusting connection strength fast and efficiently. The conjugate gradient descent of each presented training pattern was calculated to identify the minima on the error surface for each training pattern. Experiments were performed and the performance of the neural network was observed more convergent and accurate while training the network with newly introduced momentum term.
Keywords: OCR, Momentum Term, Back Propagation, Conjugate Gradient Descent, MLP

Introduction

The use of physical documents is still prevalent in most of the communications in daily life. As the paper is a very comfortable and feasible medium to store data, the demand for physical documents will continue for many years to come. Hence there is a great demand for the software techniques that can automatically extract, analyze and store information from the physical documents [1] for later retrieval. The OCR technology is successfully used by businesses like insurance companies and postal department etc., which process lots of handwritten documents. The automated processing of handwritten material [2] optimizes the processing speed as compared to human processing and shows high recognition accuracy and achieves high benefits in monetary terms.

The artificial neural network structure involved in our experiment of handwritten character recognition was a MLP (Multi-Layered Perceptron) which is a supervised learning network in nature. A MLP is a feed-forward neural network that uses error back-propagation algorithm for its training [3]. The experiments conducted in this paper have shown the effect of the modified momentum term on the learning and recognition accuracy of the BP neural network. Experiments were performed and the performance of the neural network was compared while using Classical Momentum Term (α) and Modified Momentum Term (β).

The remainder of the paper is organized as follows: Section 2 deals with the overall system design, character sample creation and the various steps involved in the OCR System. Neural Network Architecture and functioning of proposed experiments are presented in detail in section 3. Section 4 provides various experimental conditions for all the experiments conducted under this work. Discussion of Results and interpretations are described in section 5. Section 6 presents the conclusion and also gives the future path in this field.

2. Character Sample Creation and System Design

A typical handwriting recognition system is characterized by a number of steps, which include (1) Digitization / Image Acquisition, (2) Preprocessing, (3) Image Binarization (4) Network Training (5) Classification / Recognition and (6) Testing. Preprocessing aims at eliminating the variability that is inherent in cursive and hand-printed characters. Scaling, Noise Elimination [4], Slant Estimation and Correction and Contour Smoothing [5] are some preprocessing techniques that have been employed in an attempt to increase the performance of the recognition process.

All hand printed characters were scanned into gray scale images. Each character image was traced vertically after converting the gray scale image into binary form. The binary matrix is shown in Fig.1 (b).

[1]Dept. of Comp.Sc., MSI, Janakpuri, New Delhi, India.

[2]Dept.of CSE, TITS, Bhiwani, Haryana, India.

[3]Dept .of CSE, MSIT, Janakpuri, New Delhi, India.

[4]Dept .of IT., IMS, Noida, UP, India.

E-mail: [1]amit.choudhary69@gmail.com, [2]rahulrishi@rediffmail.com, [3]savita.ahlawat@gmail.com, [4]vijaypal.dhaka@gmail.com

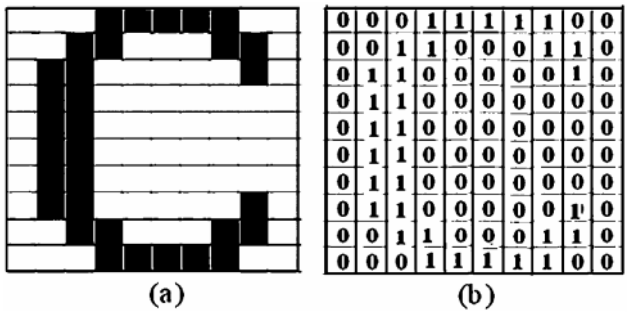

Fig. 1: (a) Binary Representation of Character 'C'and (b) Matrix Representation

This binary matrix of size 10 x 10 in Fig. 1 (b) was reshaped into a binary matrix of size 100 x 1 and the process is repeated for all the 26 characters. The reshaped characters were then clubbed together in a matrix of size 100 x 26 to form a SAMPLE which is made as an input to the neural network for learning and testing [6, 7].

3. NEURAL NETWORK'S ARCHITECTURE AND FUNCTIONAL DETAILS

The system was simulated using a feed forward neural network system that consists of 100 input neurons with linear activation function that are equivalent to the input character's size as we have resized every character into a binary matrix of size 10 x 10. The number of hidden neurons is directly proportional to the system resources. The bigger the number more the resources are required. The number of neurons in a hidden layer was kept 35 for optimal results. The activation function in the hidden layer and output layer was 'tansig'. The 26 output neurons in the output correspond to 26 letters of English alphabet. For sample creation, 250 characters were gathered form 50 people. After the preprocessing is over, 5 samples were considered for training such that each sample was consisting of 26 upper case letters (A-Z). Each sample was presented to the network 100 times thus 500 experiments have been conducted.

The processing units of input layer used the liner activation function and the units of hidden and output layers used the non-liner differentiable output function (tan-sigmoid) as shown in Fig 2.

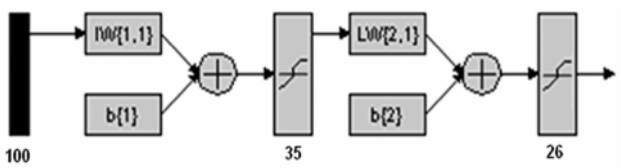

Fig. 2: Architecture of the Neural Network used in the Recognition System

The weight updates [6] for output unit can be represented as:

$$W_{ik}(t + 1) = W_{ik}(t) + \eta\Delta W_{ik}(t) + \alpha\Delta W_{ik}(t - 1) \quad (3.1)$$

Where:

$W_{ik}(t)$ is the state of weight matrix at iteration t

$W_{ik}(t + 1)$ is the state of weight matrix at next iteration

$W_{ik}(t - 1)$ is the state of weight matrix at previous iteration.

$\Delta W_{ik}(t)$ is current change/ modification in weight matrix and α is standard classical momentum variable to accelerate the learning process [7]. This variable depends on the learning rate of the network. As the network yields the set learning rate, the momentum variable tends to accelerate the process.

The network was made to learn the behavior with this Gradient Descent. For next trial, the gradient momentum term was modified by adding one more term i.e. the second momentum term (β).

$$W_{ik}(t + 1) = W_{ik}(t) + \eta\Delta W_{ik}(t) + \alpha\Delta W_{ik}(t - 1) + \beta\Delta W_{ik}(t - 2) \quad (3.2)$$

When the weight update with this equation is computed sequentially, it leads to the following benefits:

(i) A sum of the current (descent) gradient direction and a scaled version of the previous correction were computed.

(ii) Faster learning process was achieved which is evident by comparing learning graphs shown in Fig. 3 and Fig. 4.

(iii) Weight modification was based on the behavior learned from latest three iterations instead of two.

Second derivative (minima) of this gradient descent was computed using MATLAB's built in function Diff( ) to know about the status of convergence for the 500 trials. For a vector X, the value of Diff(X) was calculated by computing [X(2)–X(1) X(3) – X(2) ... X(n) – X(n – 1)]. For a matrix, Diff(X, N) is the N-th order difference along the first non-singleton Dimension (denoted by DIM). If N >= size(X, DIM), Diff takes successive differences along the next non-singleton dimension. This second derivative represents the points of local minima on the unknown error surface of the training input patterns. The convergence of learning process was judged based on the variation among these points of local minima.

The neural network was exposed to 5 different samples as achieved in section 2, each sample being presented 100 times. The same network was trained 50 times with classical momentum term and 50 times with modified momentum term. Actual output of the network was obtained by 'compet'

function. This function puts 1 at the output neuron in which the maximum trust is shown and rest neuron's result into '0' status. The output matrix was a binary matrix of size (26, 26). The output was of the size (26 × 26) because each character has 26 × 1 output vector. First 26 × 1 column stored the first character's recognition output, the following column was for next character and so on for 26 characters. For each character, the 261 vector contained value '1' at only one place. For example character 'A' if correctly recognized, will result in [1, 0, 0, 0 …all …0] and character 'B' if correctly recognized, will result in [0, 1, 0, 0 …all …0]

The difference between the desired and actual output was calculated for each cycle and the weights were adjusted with Eq.3.1 and Eq.3.2. This process was continued till the network converged to the allowable or acceptable error or the number of epochs reached to maximum allowed count.

4. EXPERIMENTAL CONDITIONS

The various parameters and their respective values used in the learning process are shown in Table I.

Table 1
Experimental Conditions

Parameters	Value
Learning/ Training Goal for entire network	0.01
Acceptable Error	0.001
Classical Momentum Term (α)	0.90
Modified Momentum Term (β)	0.05
Maximum Epochs	50000
Termination Conditions	Based on minimum Mean Square Error
Initial Weights and biased term values	Randomly generated values between 0 and 1

5. DISCUSSION OF RESULTS AND INTERPRETATIONS

In this section, the experiments and their outcomes at each stage are described.

5.1. Gradient Computation

The gradient descent is the characteristic of error surface. If the surface is not smooth, the gradient calculated will be a large number; this will give a poor indication of the "true error correction path". On the other hand, if the surface is relatively smooth, the gradient value will be a smaller one. Hence the smaller gradient is always the desirable one. This rationale is based on the knowledge of the shape of the error surface. For each trial of learning, the computed values of gradient descent are shown in Table 2. An average of 50 trials for each sample with same type of momentum term was taken into consideration.

Table 2
Comparison of Gradient Values with Classical and Modified Momentum Terms

Sample	Gradient 1 (Classical Method)	Gradient 2 (Modified Method)
Sample 1	1899300	2029280
Sample 2	5558000	2021500
Sample 3	6942400	1723310
Sample 4	1488300	1430094
Sample 5	7125400	2189700

Gradient 1 and Gradient 2 represents the present error in the network trained with classical weight update method and modified weight update method respectively. It is evident that, for each sample, the gradient values were reduced when modified momentum term was introduced in the weight update process.

5.2. Number of Epochs

The results of the learning process of the network in terms of the number of training epochs are represented in Table 3.The samples were presented to the neural network for training in the following fashion:

Sample1-> Sample2-> Sample3-> Sample4-> Sample5-> Sample1-> Sample2-> Sample3-> Sample4-> Sample5-> Sample1-> Sample2-> Sample3-> Sample4->…so on.

Table 3
Comparison of Network Training Iterations (epochs) between the Two Learning Trails

Sample	Epoch1 (Classical Momentum Term)	Epoch2 (Modified Momentum Term)
Sample1	367	179
Sample2	654	429
Sample3	7852	2143
Sample4	48024	14548
Sample5	50000	50000

Epoch1 represents the number of network iterations for a particular sample when presented to the neural network 50 times for learning with classical momentum term. Epoch2 represents the number of iterations with modified momentum term for each sample when presented to the neural network 50 times. Each epoch count is an average of 50 iterations. It is observed that the number of epochs required to train the network were reduced to a great extent when modified momentum term was introduced during the

weight update mechanism of the back propagation algorithm.

5.3. Error Estimation

The network performance is shown in Table 4. Error 1 represents the error present in the network trained with classical weight update method using classical momentum term and Error 2 represents the error present in the network trained with modified weight update method using additional momentum term. It is evident that the error is reduced when modified momentum term was used in the network during weight update mechanism.

In other words, we can conclude that by training the network with the modified momentum term, there is an increase in the probability of converging the network before the number of training epochs reaches it maximum allowed count.

Table 4
Error Level Attained by the Neural Network Trained with Both Methods

Sample	Error 1(Classical Weight Update Method)	Error 2 (With Modified Weight Update Method)
Sample1	0	0
Sample2	0	0
Sample3	0	0
Sample4	0.00343787	0
Sample5	0.00787574	0.00491716

5.4. Testing

The character recognition accuracy of the network with Classical Momentum Term and Modified Momentum Term are shown in Table 5. The networks were tested with 2 samples. These samples were new to both of the networks because they were never trained with these samples. The recognition rates for these samples are shown in Table -5.

Table 5
Character Recognition Accuracy

No. of characters in test sample	Network with Classical Momentum Term		Network with Modified Momentum Term		Recognition Percentage	
	Correct	Incorrect	Correct	Incorrect	Classical	Modified
26	17	9	24	2	65.4%	92.3%
26	20	6	25	1	80.0%	96.2%

It is clear that the recognition accuracy is improved when modified momentum term is introduced in the weight update process.

6. CONCLUSION AND FUTURE SCOPE

The proposed method for the handwritten character recognition using the descent gradient approach and modified momentum term showed the remarkable enhancement in the performance. As shown in Table-2, the smaller gradient values are achieved in case of weight update mechanism using modified momentum term. Smaller the gradient values, smoother will be the error surface and the probability that the neural network will get struck to the local minima will be the least. Smaller gradient values indicate that the error correction is downy and accurate.

This paper has introduced an additional momentum term in the weight-age modification. This additional momentum term accelerates the process of convergence and the network shows better performance. The number of epochs required to train a network are reduced when the second momentum term is introduced during weight update mechanism as shown in Table-3. The variation of MSE with the training epochs are shown in Fig. 3 and Fig. 4 when Sample-1 is presented to the network with classical and modified momentum terms respectively.

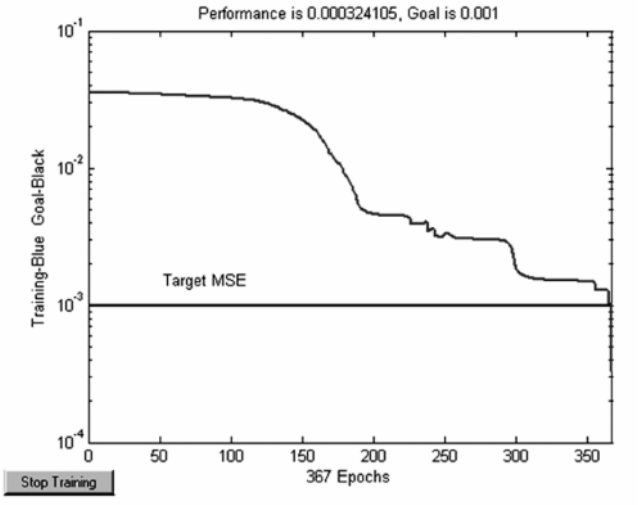

Fig. 3: MSE Plot with Classical Momentum Term

The testing process of the network is improved when a second momentum term is introduced. It is clear from Table-5 that the recognition accuracy of the network is excellent where modified momentum term is introduced for updating the weights during error back propagation. Due to the back-propagation of error element in Multilayer Perceptron

(MLP), it frequently suffers from the problem of Local-Minima, hence the samples may not converge so as the case with Sample 5 as shown in Table-4.

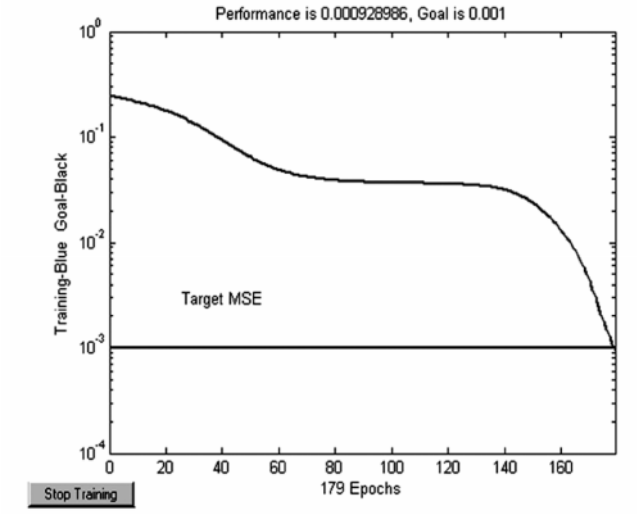

Fig. 4: MSE Plot with Modified Momentum Term

Nevertheless, more work needs to be done especially on the test for more complex handwritten characters. The proposed work can be carried out to recognize English words of different character lengths after proper segmentation of the words into isolated character images.

REFERENCES

[1] N. Arica and F. Yarman-Vural, "An Overview of Character Recognition Focused on Offline Handwriting", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 31, No.2, pp. 216—233, 2001.

[2] Neves, R.F.P., et al. 2008. "A New Technique to Threshold the Courtesy Amount of Brazilian Bank Checks", In Proceedings of the 15th IWSSIP, (Bratislava, Slovak Republic, June 2008), IEEE Press, in Press.

[3] B. K. Verma, "Handwritten Hindi Character Recognition Using Multilayer Perceptron and Radial Basis Function Neural Network", IEEE International Conference on Neural Network, 4, Pp. 2111—2115, 1995.

[4] S Srihari, V Govindraju, R Srihari. "Handwritten Text Recognition. Proc 4th International Workshop on Frontiers in Handwriting Recognition", Taiwan, 265-275, 1994.

[5] An OCR System for Telugu, "Proceedings of the Sixth International Conference on Document Analysis and Recognition", p.1110, September 10-13, 2001

[6] A. Choudhary, R. Rishi, S. Ahlawat, V. S. Dhaka, "Optimal Feed Forward MLP Architecture for Off-line Cursive Numeral Recognition," International Journal on Computer Science and Engineering, 2, No.1s, pp. 1-7, 2010.

[7] A.Choudhary, R. Rishi, S. Ahlawat, V. S. Dhaka, "Influence of Introducing an Additional Hidden Layer on the Character Recognition Capability of a BP Neural Network having One Hidden Layer" International Journal of Engineering and Technology, 2, No.1, pp. 24-28, 2010.