Comparison of the Multi Layer Perceptron and the Nearest Neighbor Classifier for Handwritten Numeral Recognition.

Article *in* Journal of Information Science and Engineering · November 2005

Source: DBLP

5 authors, including:

Kaushik Roy
West Bengal State University

Chitrita Chaudhuri
Jadavpur University

Mahantapas Kundu
Jadavpur University

Mita Nasipuri
Jadavpur University

JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 21, 1247-1259 (2005)

Short Paper

Comparison of the Multi Layer Perceptron and the Nearest Neighbor Classifier for Handwritten Numeral Recognition

K. ROY[+], C. CHAUDHURI, M. KUNDU, M. NASIPURI AND D. K. BASU
Computer Science and Engineering Department
Jadavpur University
Kolkata 700032, India
[+]*C.V.P.R. Unit*
Indian Statistical Institute
Kolkata 700108, India

The work presents the results of an investigation conducted to compare the performances of the Multi Layer Perceptron (MLP) and the Nearest Neighbor (NN) classifier for handwritten numeral recognition problem. The comparison is drawn in terms of the recognition performance and the computational requirements of the individual classifiers. The results show that a two-layer perceptron performs comparably to a NN like standard pattern classifier in recognizing unconstrained handwritten numerals, while being computationally more cost effective. The work signifies the usefulness of the MLP as a standard pattern classifier for recognition of handwritten numerals with a large feature set of 96 features.

Keywords: artificial neural network, multi layer perceptron, pattern recognition, nearest neighbor classifier, learning, generalization, training

1. INTRODUCTION

Multi Layer Perceptrons (MLPs) constitute an important class of feed-forward Artificial Neural Networks (ANNs), developed to replicate *learning* and *generalization* abilities of humans with an attempt to model the functions of biological neural networks. They have many potential applications in the areas of Artificial Intelligence (AI) and Pattern Recognition (PR). Handwritten numeral recognition is a benchmark problem of PR. It has a clearly defined commercial importance and a level of difficulty that makes it challenging, yet it is not so large as to be completely intractable. Optical Character Recognition (OCR) of handwritten numerals is central to many commercial applications related to reading amounts from bank cheques, extracting numeric data from filled in forms, interpreting handwritten pin codes from mail pieces and so on. The work presented here mainly aims for establishing the usefulness of the MLP as a pattern classifier compared to the Nearest Neighbor (NN) classifier used as a suboptimal traditional classifier.

Received March 13, 2003; revised August 11, 2003 & March 15, 2004; accepted April 29, 2004.
Communicated by Kuo-Chin Fan.

1.1 Previous Work

Previously, Burr [1] used a small feature set of seven *shadow codes* for the recognition of handwritten numerals with the MLP. After Burr's work, Weideman *et al.* [2] conducted an extensive study to compare recognition performances and computational requirements of an MLP like feed-forward neural network to those of a NN classifier for handwritten numeral recognition problem. The work considered a set of 98 features for representing handwritten numerals. The feature set included 36 features for Geometric moments, 18 topological features, 28 specially selected components of 2-D Fast Fourier Transform (FFT) features and 16 shadow features. All these will be explained later. The work of Weideman et al. finally concluded that for complex problems neural networks perform comparably to NN classifiers, while being significantly more cost effective.

1.2 Motivation

The MLP like feed-forward neural network designed for the work of Weideman et al. was a 3 layer one. In addition to an input layer, a 3 layer perceptron consists of two hidden and one output layers. Synaptic connections in the network designed by Weideman et al. were more complex than what is found in usual MLPs. In addition to layer to next layer synaptic connections, it was designed with layer to next-next layer connections from each of its two hidden layers. But, if only layer to next layer synaptic connections can be maintained in the network with one fewer hidden layers (i.e., if the complexity of the network can be reduced) keeping the recognition rate at least same, then the design of a more cost effective network becomes a possibility. Motivation behind the present work comes from this point.

1.3 The Present Work

This work presents a cost effective design of a 2 layer MLP for recognition of handwritten numerals. It also presents a comparative assessment of the performances of the MLP and a NN classifier for the same. The feature set selected for representing a handwritten numeral includes all the features selected by Weideman *et al.* excepting two, one for the ratio of the energies at the top to the bottom of the loop, if the numeral image contains any, and the other for the number of loops in the numeral image. Exclusion of these features has not affected the recognition performance of the network in the present work.

2. DESIGN OF THE FEATURE VECTOR

As mentioned before, a 96 component feature vector is considered here to represent each of the numeral images in the feature space. For extracting features from the numeral images, the images are first bounded by the smallest rectangles and then each scaled to a size of 32×32 pixels. The scaled images so obtained are finally binarized through thresholding. This is to make the numeral images *size invariant* and avoid *translation* at the same time. All the features, extracted from the numeral images require *normalization* before actual use.

2.1 Shadow Features

Shadow features are computed by considering the lengths of projections of the numeral images, as shown in Fig. 1, on the four sides and eight octant dividing sides of the minimum size rectangles. The feature vector considered for the present work includes 16 shadow features.

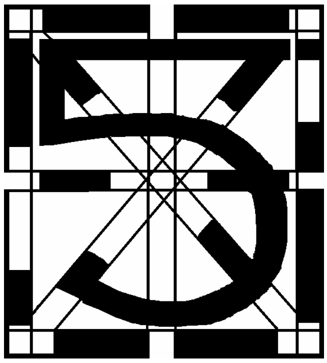


Fig. 1. The shadow features of a numeral image.

2.2 Topological Features

Topological features provide information about the outlines of the numeral patterns. For example, diagonal-wise distances of the first black pixels from the four corner points of the smallest bounding rectangle of the image pattern '0' will differ from the distances computed in the same way for the image pattern '1'. The present work considers 16 such topological features [2] to form the feature vector.

2.3 2-D FFT Features

For extraction of FFT features from a 2D digital image, the image needs to be represented as a discrete function of *x*-*y* co-ordinates on the image plane to give the values of gray levels at those points. The image so represented can be decomposed into a number of weighted periodic functions (sinusoids) by application of the FFT to it. The weights associated with these functions constitute a set of Fourier coefficients (FCs), which are complex numbers. The number of FCs thus produced from an image is equal to the number of pixels. The square of the magnitude of a coefficient is called its energy.

The feature vector used for the present work includes 24 low frequency FFT coefficients extracted from a numeral image. These coefficients are less sensitive to rotation and scaling compared to high frequency ones. They have joint Gaussian probability density due to the central limit theorem [2].

The feature vector also includes the frequency domain energies computed with the FFT coefficients from four adjacent regions, on the upper half of the major diagonal of the image plane. Each of these four regions subtends an angle of 45° at the midpoint of the diagonal. The 4 features are obtained from these regions for detecting the presence of horizontal, vertical and diagonal lines in the numeral images.

2.4 Geometric Moments

2D Geometric moments are commonly used features for describing the shape of an object from its image. Mathematically, a set of moments can be viewed as the projections of the two variable binary image function onto a set of two variable polynomials. Moments can be made invariant to translation, rotation, scaling and reflection. The present work uses 36 normalized moments (up to order four), computed in four image quadrants with nine moments per quadrant.

3. DESIGN OF THE MLP

The MLP is in general a layered feed-forward network, pictorially represented with a directed acyclic graph as shown in Fig 2. Each node in the graph represents an artificial neuron of the MLP, and each directed arc represents a synaptic connection between two neurons and the direction of the signal flow in the MLP. The labels used with the arcs in the graph denote the strengths of synaptic connections of the MLP, also called weights. Each layer of the MLP consists of a specific number of neurons, each of which is connected with all the neurons of the immediately following layer to replicate synaptic connections of the biological neural networks. The functions of a biological neuron are modeled by computing a differentiable nonlinear function (such as a sigmoid) for each artificial neuron of the MLP. Use of such a sigmoid function is also biologically motivated, since it attempts to account for the refractory phase of biological neurons.

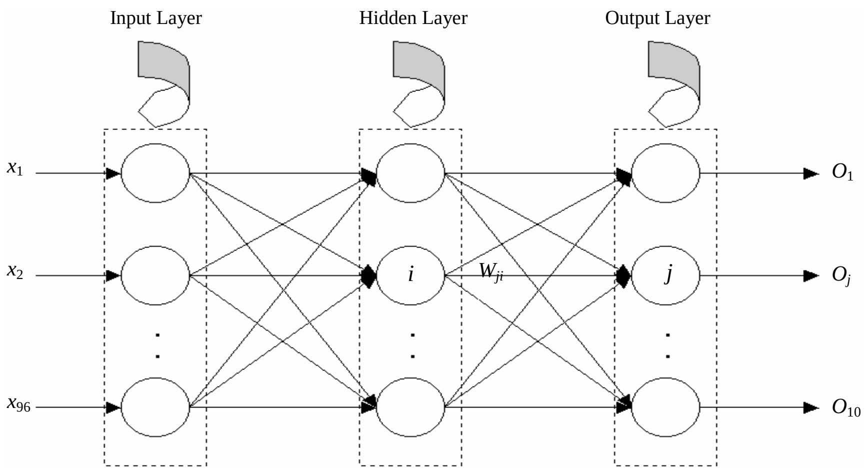


Fig. 2. The graphical representation of a 2-layer MLP.

For pattern classification, the number of neurons in the input layer of an MLP is determined by the number of features selected for representing the relevant patterns in the feature space. The neurons in the input layer, acting as sensory units, compute the identity function, $y = x$. Each of the neurons in the hidden and output layers computes the *sigmoidal function* of the sum of products of the input values and weight values of the corresponding connections. The output of j^{th} neuron, O_j, in a hidden or the output layer is thus mathematically represented as $\boldsymbol{O_j} = 1/(1 + e^{-net_j})$ where, $net_j = \sum_i W_{ji}O_i$ and W_{ji} = the weight of the connection between i^{th} and j^{th} neuron.

The number of neurons in the output layer of the MLP is determined by the number of possible pattern classes to be dealt with for some problem of interest. The class label assigned to the output neuron producing the highest output value determines the class of the input pattern supplied to an MLP.

The training process of an MLP mainly involves tuning of the strengths of its synaptic connections, adjustment of the number of hidden layers and the number of neurons therein so that it can respond appropriately to every input taken from the training set. It incorporates learning ability in an MLP. Generalization ability of the same is tested after training by checking its responses to input patterns which were not used for training.

3.1 Back Propagation Algorithm

The Back Propagation (BP) algorithm, a supervised learning method, is used for training of the MLPs with patterns of known classes. It aims to find a set of weight values for the perceptron to minimize the sum of the squared errors produced with all the training patterns. In doing so, the BP algorithm conducts a gradient descent search on the error surface of the perceptron in the weight space. It can be proved that the amount of weight change, ΔW_{ji}, needed to minimize the sum of squared errors is

$$\Delta W_{ji} = \eta \delta_j O_i \qquad \text{where } \eta = \text{learning rate parameter } (0 < \eta < 1),$$
$$\delta_j = \text{the error gradient of } j^{th} \text{ neuron}$$

$$\delta_j = \begin{cases} O_j(1-O_j)(d_j - O_j) & \text{if } j^{th} \text{ neuron belongs to output layer} \\ O_j(1-O_j)\sum_k \delta_k W_{kj} & \text{if } j^{th} \text{ neuron belongs to a hidden layer} \end{cases}$$

By introducing the *momentum* term, α, $(0 < \alpha < 1)$, the formula to update the weight used by the BP algorithm is

$$W_{ji}(t+1) = W_{ji}(t) + \eta\delta_j O_i(t) + \alpha(W_{ji}(t) - W_{ji}(t-1)), \text{ where } t \text{ stands for time.}$$

Weight updating is continued iteratively with the training patterns until certain *stopping criteria* are reached The stopping criteria for the present work are selected as the condition under which the sum of the squared errors for all 900 training patterns falls below a positive number as small as 0.6.

3.2 Design of Input, Output and Hidden Layers

Compared to the 3 layer neural network designed by Weideman *et al.* for handwritten numeral recognition, the present work selects a two layer perceptron. According to the Universal Approximation theorem [3], a single hidden layer is sufficient to compute a uniform epsilon approximation to a given training set represented by the set of inputs $x_1, \ldots, x_n$ and a target output function $f(x_1, \ldots, x_n)$.

The number of neurons in the input and output layers of the perceptron selected here are set to 96 and 10 respectively as the number of features selected for the feature set is 96 and the number of possible classes in handwritten numerals is ideally 10. For choosing an optimal number of neurons in the hidden layer, a curve showing the variation of the sum of global errors for the individual training patterns versus the number of hidden units is plotted in Fig 3. The sum of global errors is recorded each time after training the perceptron for 1000 epochs. In recording the sum of global errors, the perceptron is tested with the training set. The curve plotting experiment is conducted after choosing the values of η and α as to 0.65 and 0.7, respectively, and using a small training set of 100 numerals.

The training set used at this stage consists of 100 sample patterns randomly chosen in equal numbers from each of the 10 classes. High values of η and α are chosen to have faster convergence of the training algorithm at the initial stage of experimentation. But, such a choice runs the risk of overshoot. To test the generalization ability of the designed MLPs, a test set of 100 handwritten numerals is formed, arbitrarily choosing 10 samples of each class from a large database. The number of hidden units is ultimately set to a value of 17 as the curve shown in Fig. 3 yields the smallest error.

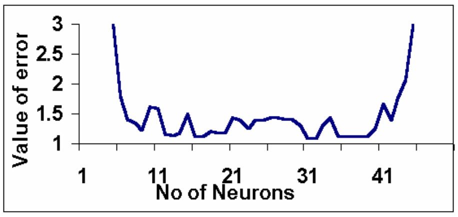


Fig 3. Variation of global sum of errors with increasing number of hidden neurons in a 2 layer perceptron. (η = 0.65, α = 0.7, training set size = 100, number of epochs = 1000)

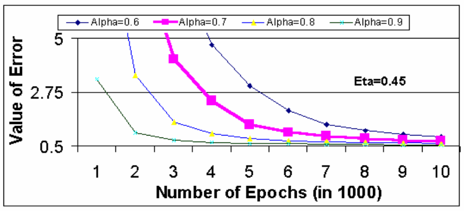
3.3 Selection of Values for η and α

In BP learning, the learning rate η is chosen to control the degree of exhaustiveness of search for the global minima on the error surface generated for the network in the weight space. The less is the values of η learning becomes slower and search becomes more exhaustive. High values of η, though, lead to faster convergence and may result divergent behavior, especially at high curvature points over the error surface.

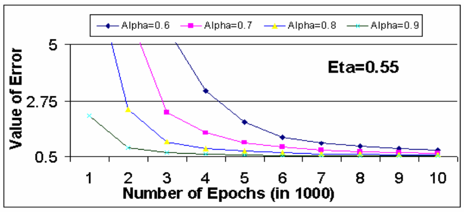
The momentum parameter α, which adds a fraction of the previous weight change to the refined weight value, controls the degree of deviation in each iteration. The value of α can be used to compensate the effect of η in the overall search process. Suitable selection of values of η and α requires a trade off between the speed of learning and the quality of solution.

To observe the effect of different combinations of values of η and α on the speed of learning and the quality of solution, curves showing the error value versus the number of epochs are drawn in Fig. 4 (a-e) for η = 0.45, 0.55, 0.65, 0.75, 0.85 and α = 0.6, 0.7, 0.8, 0.9. The error values shown in the curves are computed with the training set.

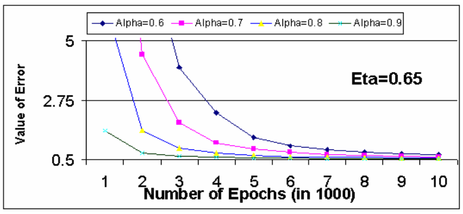
From the curves, it can be observed that for high values of η the error falls sharply as the number of epoch's increases. This may run the risk of overshoot on the error surface during search for global minima. So the choice of η is to be made from {0.45, 0.55}. The curves showing the error values versus the number of epochs are shown in Fig. 4 (a-b) for these values of η. Out of these curves, the one drawn for η = 0.45 and α = 0.7 shows a steady fall in the error value with an increase in the number of epochs. Hence the values of η and α are finally chosen as 0.45 and 0.7, respectively.



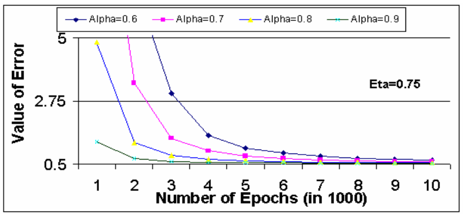
(a)



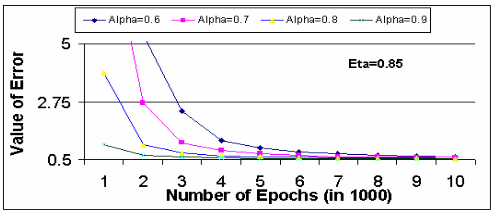
(b)



(c)



(d)



(e)

Fig. 4. Variation of the global error sum with increasing number of epochs.

3.4 Final Selection of the Number of Hidden Neurons

For finer tuning of the number of hidden neurons, percent recognition rates of the MLP with different numbers of hidden neurons are recorded on the test set by increasing the number of epochs from 1000 to 10,000 in steps of 1000 during training. The values of η and α are kept fixed to 0.45 and 0.7 respectively during the experiment. The results are shown in Table 1.

Table 1. Variation of the percent recognition rate with increase in the number of epochs for different numbers of hidden layer neurons.

Epochs / No of H. U.	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
7	86	92	94	95	95	95	95	95	95	95
10	91	93	94	95	95	95	95	96	96	96
13	94	96	97	97	97	97	97	97	97	97
15	**95**	**96**	**97**	**97**	**97**	**97**	**97**	**97**	**97**	**97**
17	94	95	96	96	97	97	97	97	97	97
23	94	95	96	96	96	96	96	96	96	96
30	94	95	95	96	96	96	96	96	96	96
37	94	95	95	95	96	96	96	96	96	96
45	94	95	96	96	96	96	96	97	97	97
96	92	95	96	96	96	96	96	96	96	96

It can be observed from the table that the highest percent recognition rate that can be achieved is 97%. Also for 15 hidden units, the highest recognition rate has been reached with fewer epochs. This is why the number of hidden neurons for the MLP is finally chosen as 15 with $\eta = 0.45$ and $\alpha = 0.7$.

3.5 Reject Criteria for the MLP

After choosing the values of η, α and a convergence criteria for the present work, as mentioned above, a 2 layer MLP (96-15-10) is designed by running the BP algorithm on the training set of 900 samples. In line with the work of Weideman *et al.*, some *reject capability* is also introduced in the MLP. The reject criteria are as follows:

i) An input pattern is rejected if the highest activation level of output neurons lie below a certain threshold (0.7 in this case). Or,
ii) An input pattern is rejected if the first and the second highest activation levels of the output neurons are very close (i.e., if the difference between the two are 0.2 or less in this case).

4. DESIGN OF THE NN CLASSIFIER

The NN classifier requires a set of sample patterns of known classes, called *reference* set. To classify an unknown pattern X, it first finds the reference pattern closest to X in the feature space and then assigns the class label of the same to X.

4.1 Why the NN Classifier?

Performance-wise, the Bayes' classifier is *optimal* as it ensures the lowest probability of committing classification errors. But its design requires knowledge of a *priori* *probabilities* and *densities* of all classes as well as the cost of decision, which are not available with the samples used for the present work. So the NN classifier, which follows nonparametric approach, is chosen. Its *error rate* in no way exceeds twice the Bayesian error rate for low probabilities of error [4]. It can be considered to be a good approximation for the Bayes' classifier especially when the Bayesian error rate approaches to zero or the error rate of random guessing.

4.2 The *k*-Nearest Neighbor Classifier

To classify an unknown pattern X, the k-nearest neighbor (k-NN) classifier first finds k of the reference patterns closest to X and then classifies X to a class which is found to have the maximum number of samples among the k reference patterns. The error rate for the k-NN classifier approaches that of the Bayes' classifier as both the values of k and the ratio of the total number of reference samples to the value of k approach to infinity.

The present work uses a weighted vote [4] for each of the k nearest neighbors of the unknown pattern. The expression for computing the weight is given as $1/(1 + d_e)$, where d_e is the Euclidean distance between an unknown pattern and one of its k nearest neighbors in the feature space.

4.3 Design of the Reference Set

The present work employs *Maximin* distance clustering [5] to find the cluster centers in the same training set of 900 samples previously used for training the MLP. The cluster centres thus obtained form a primary reference set. This reference set is used for a 1-NN classifier to classify the non-centre samples of the training set. Non-centre samples are thus grouped with the cluster centres. The cluster centres, which are associated with more than 50% misclassified samples, are eliminated from the primary reference set to refine it for the use of the k-NN classifier. The reference set finally consists of 179 sample patterns.

4.4 Choice of the Value of *k*

In the reference set used for the work here, the smallest number of prototypes that a class has is 3. And the maximum number of prototypes that a class has is 38. On the basis of this information, the value of k is chosen to be 3 for the present work.

4.5 Reject Criteria for the NN Classifier

The reject decision criterion as incorporated into the NN classifier is as follows. An input pattern X is rejected if its distance from its nearest reference sample is less than 3% of the distance between X and its second nearest reference sample of a different class in the feature space.

5. EXPERIMENTAL RESULTS

The training and the test sets for this work consist of 900 and 100 randomly selected samples of handwritten numeral images, respectively (Fig. 5). As mentioned before, all these samples are binary images of 32×32 pixel size each.

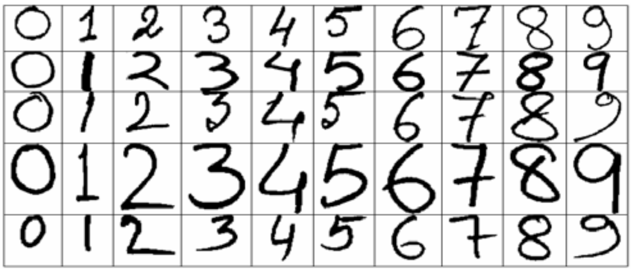


Fig. 5. BMP images of the first five samples of each numeral drawn from the training set.

5.1 Recognition Performances

The recognition performances of the MLP (96-15-10) classifier, as recorded on the training and the test sets, are given in Table 2 (a). The recognition performances of the 3-NN classifier with 179 cluster centers are given in Table 2 (b). The

Table 2. (a) Recognition performances of the two layer perceptron.

Training Set			Test Set		
% of Classification	% of Miss-Classification	% of Rejection	% of Classification	% of Miss-Classification	% of Rejection
99.78	0.00	0.22	97.00	0.00	3.00

Table 2. (b) Recognition performances of the 3-NN classifier.

Training Set			Test Set		
% of Classification	% of Miss-Classification	% of Rejection	% of Classification	% of Miss-Classification	% of Rejection
95.00	4.13	0.86	90.00	8.00	2.00

5.2 Computational Complexity

To study the computational requirement, a reasonable suggestion might be to compare the number of multiplications, additions and the number of function evaluations required by the algorithm. A layered network requires $N_h(N_i + N_o)$ weights, where subscripts refer to the number of hidden units, input units and output units, respectively. So, the 96-15-10 network has 1,615 weights including those associated with the outputs of two bias neurons, one for input to hidden layer and one for hidden to output layer. This implies that for each character decision, the MLP network chosen here requires 1,615 multiplications, 1,615 additions, and 25 evaluations of the logistic function. This gives a total of 3,255 operations per decision. On the other hand, the 3-NN classifier designed here with a reference set of 179 patterns stores 17,184 values. Each character decision made by the 3-NN classifier requires 17,187 multiplications, 34,371 additions/subtractions, and 179 evaluations of the square root function. This amounts to a total of 51,737 operations per decision. So the 3-NN classifier takes 15.9 times more computations than does the MLP (96-15-10) to classify one handwritten numeral image. Besides this the 3-NN classifier has a storage requirement of 10.64 times than that of the MLP for the feature set.

6. CONCLUSIONS

The above results suggest that for handwritten numeral recognition, even with 96 features, a two layer perceptron can be designed to show a recognition performance which is comparable to that of the NN classifier.

The possible reason behind the superior performance of the MLP to that of the *k*-NN classifier, as observed here, can be explained as follows. Both the NN classifier and the MLP can approximate arbitrary decision regions for classification of unknown patterns in the feature space. The performance of the *k*-NN classifier used here depends on the reference set and the value of *k* chosen for the work. The reference set is formed by selectively collecting some of the cluster centres identified by the maxim in cluster seeking algorithm, using the Euclidean distance as a measure of similarity. The performance of a cluster seeking algorithm again depends on the similarity measure chosen for it. Thus the scope for fine tuning of the *k*-NN classifier's performance is much limited here. But an adjustable number of hidden layer neurons and tunable connection weights of the MLP provide a wider opportunity for fine tuning of the decision regions. For this, the recognition performances of the MLP can be made better compared to those of the NN classifier used for the classification of handwritten numeral patterns.

The best recognition performance observed on the handwritten numeral recognition problem from among the top 10 classifiers, selected from the First Census Optical Character Recognition System Conference [6] in 1992, was 98 percent. All these classifiers were based on either some type of multilayer feed-forward network or a NN classifier. Compared to the best performance observed for these classifiers, the recognition performance of the MLP designed for the present work is observed to be 97 percent. The value is close to the best performance of the top ten performers of the First Census Optical Character Recognition System Conference.

Compared to the work of Weideman *et al.*, the computational requirements of the MLP designed for the present work have been further reduced by the selection of a 2 layer perceptron with only layer-to-next layer synaptic connections. This has no adverse effect on the recognition performance in respect to what was observed for Weideman *et al.*'s 3 layer MLP neural network with additional layer to next-next layer synaptic connections.

Finally, the work presented here provides supportive evidence for establishing the utility of BP algorithm for a complex real world problem such as recognition of handwritten numerals using a large feature set and also the usefulness of the MLP as a standard pattern classifier.

ACKNOWLEDGMENTS

The authors are thankful to the Centre for Microprocessor Application and the Video for Multimedia Project, Computer Science and Engineering Department, Jadavpur University, for providing necessary infrastructural facilities during the progress of this work.

REFERENCES

1. D. J. Burr, "Experiments on neural net recognition of spoken and written text," *IEEE Transactions on Neural Networks*, ASSP, Vol. 36, 1988, pp. 1162-1168.
2. W. E. Weideman, M. T. Manry, H. C. Yau, and W. Gong, "Comparisons of a neural network and a nearest-neighbor classifier via the numeric handprint recognition problem," *IEEE Transactions on Neural Networks*, Vol. 6, 1995, pp. 1524-1530.
3. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Pearson Education Asia, 2001, pp. 208-209.
4. E. Gose, J. Richard, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice-Hall of India Pvt. Ltd., 1999, pp. 167-169, pp. 174-175.
5. J. T. Tou and C. G. Rafael, *Pattern Recognition Principles*, Addison-Wesley Publishing Company, 1974, pp. 92-94.
6. R. A. Wilkinson, *et al*., "First census optical character recognition system conference," Technical Report, NISTIR 4912, US Dept. Commerce, National Institute of Standards and Technology, Gaithersburg, MD, 1992.

Kaushik Roy received his B.E and M.E degrees in Computer Science and Engineering from Assam University, Silchar, India and Jadavpur University (J.U) in 1998 and 2002, respectively. He has been working as a Project linked personnel in I.S.I, Kolkata, since 2003. His areas of current research interest are pattern recognition, image processing, and artificial intelligence.

Chitrita Chaudhuri received her B.E.Tel.E. and M.E.Tel.E. degrees from Jadavpur University, in 1980 and 1982, respectively. She has been working as a Lecturer in J.U. since 2001. Her areas of current research interest are pattern recognition, image processing, multimedia techniques, and data mining.

Mahantapas Kundu received his B.E.E, M.E.Tel.E and Ph.D. (Engg.) degrees from Jadavpur University, in 1983, 1985 and 1995, respectively. He joined J.U. as a lecturer in 1988 and is currently working there as a Reader. His areas of current research interest include pattern recognition, image processing, multimedia database, and artificial intelligence.

Mita Nasipuri received her B.E.Tel.E., M.E.Tel.E., and Ph.D. (Engg.) degrees from Jadavpur University, in 1979, 1981 and 1990, respectively. Prof. Nasipuri has been a faculty member of J.U since 1987. Her current research interest includes image processing, pattern recognition, and multimedia systems. She is a senior member of the IEEE, U.S.A., Fellow of I.E (India) and W.B.A.S.T, Kolkata, India.

Dipak Kumar Basu received his B.E.Tel.E., M.E.Tel., and Ph.D. (Engg.) degrees from Jadavpur University, in 1964, 1966 and 1969 respectively. Prof. Basu has been a faculty member of J.U since 1968. His current fields of research interest include pattern recognition, image processing, and multimedia systems. He is a senior member of the IEEE, U.S.A., Fellow of I.E. (India) and W.B.A.S.T., Kolkata, India and a former Fellow, Alexander von Humboldt Foundation, Germany.