Enhancing Bengali character recognition process applying heuristics on Neural Network

Golam Sarowar, M.A. Naser , S.M. Nizamuddin, Nafiz I.B. Hamid, Adnan Mahmud

Islamic University of Technology (IUT), Gazipur 1704, Bangladesh

Summary
Recognition of Bengali characters by neural network has received much attention already. Better preprocessing and feature extraction may fail to give better accuracy if the recognition process (includes training and testing) is incompetent. In this paper, some heuristics are proposed to make the back-propagation training algorithm perform better. Finally, character recognition accuracy results obtained from different algorithms are compared.

Key words:
Character recognition, neural network, backpropagation.

1. Introduction

A neural network performs pattern recognition by responding when an input vector close to a learned vector is presented. That is, it undergoes a training session first. If the training process is not proper, the network cannot classify new patterns as desired even with good preprocessing and feature extraction techniques. Backpropagation is a learning rule applied to train the network. There are methods that can significantly improve the performance of backpropagation algorithm as well as that of the network. The heuristics applied [1] are training algorithms with sequential update and faster convergence, transfer function, maximizing information content, normalization of the inputs and generalization of the trained network.

The preprocessing and feature extraction techniques prior to the recognition process are detailed in many publications [2-5]. This paper is based on transitions [4] taken as features. Artificial neural networks are used to classify characters as well as for segmentation [6], [7], [8]. A. A. Chowdhury et. al. [6.] employs neural network as classifier for their extracted features. Mahmud et. al. [7] uses normalized slope distribution of four regions as features for their neural network classifier. Conversely, Bhttacharya et. Al. [8] proposes segmentation based on neural network with the benefit of not selecting a feature set. In this paper, only the classification part done by neural network is emphasized. Section 2 describes the heuristics applied and section 3 contains the results. Finally, section 4 concludes the paper.

2. Heuristics:

2.1 Training algorithms

Backpropagation algorithm uses the gradient of the performance function to determine how to adjust the weights to minimize performance. In backpropagation, the gradient is determined by performing computations backwards through the network [9]. There are many variations of backpropagation. Some of them provide faster convergence while others give smaller memory requirement.

2.1.1 Backpropagation with adaptive Learning Rate:

Picking the learning rate is a challenge. The performance of the standard steepest descent algorithm is very sensitive to the proper setting of the learning rate. The algorithm may oscillate and become unstable if learning rate is too high. Conversely, algorithm will take longer time to converge or may never converge if the learning rate is too small. One thing can be done in this regard that an optimal rate can be assigned. But it is not practical to determine the optimal value for the learning rate before training and also the optimal learning rate changes during the training process. Instead of assigning an optimal rate, learning rate can be made adaptive which can keep the learning step size as large as possible while keeping learning stable [9]. The learning rate is made responsive to the complexity of the local error surface. The procedure is like this: first, the initial network output and error are calculated. At each epoch new weights, biases, outputs and errors are calculated using the current learning rate. If the new error exceeds the old error by more than a predefined ratio, the new weights and biases are discarded and the learning rate is decreased. If the new error is less than the old error, the learning rate is increased. The learning rate is increased but only to the extent that the network can learn without

Manuscript received June 5, 2009
Manuscript revised June 20, 2009

large error increases. When the error gets increased, learning rate gets decreased until stable learning resumes.

2.1.2 Backpropagation with momentum:

Momentum accelerates the descent to the minimum of the error surface [9]. Normally the backpropagation uses the weight change proportional to the negative gradient of current error. It uses only the first derivative of that error with respect to the weight. Weight changes can be estimated in a better way if information of second derivative is used. Momentum method is such a method where both the weight change at the previous step and the gradient at the current step are used to determine the weight change for the current step. Momentum allows the network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum. With momentum a network can slide through such a minimum. Thus it gives faster convergence.

2.1.3 Conjugate Gradient Algorithms:

As discussed earlier, the basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient). Though performance function decreases most rapidly in this direction, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions [9]. Some conjugate gradient algorithms are: Fletcher-Reeves Update, Polak-Ribiere update, Powell-Beale restarts and scaled conjugate gradient [10]. The last one avoids the time-consuming line search.

2.2 Antisymmetric activation function

Learning is faster if the activation or transfer function is antisymmetric [1]. Figure 1 shows an antisymmetric function in the form of hyperbolic tangent. For the learning time to be minimized, the use of nonzero mean inputs should be avoided. If the activation function is nonsymmetric, the output of each neuron is restricted to the value 0 and 1 (limiting value), shown in figure 2. This introduces systematic biases for the neurons located in the layers other than input layer. For the antisymmetric case the values can vary from -1 to 1 in which case it is likely to have the mean zero.

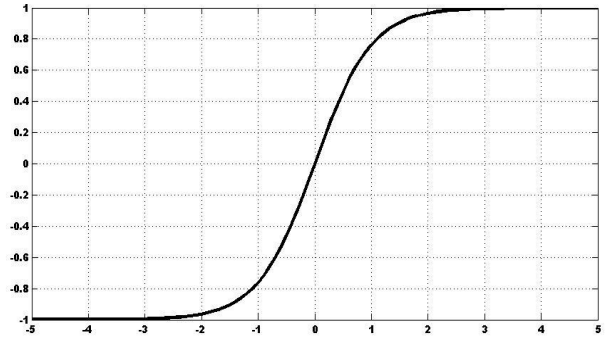

Fig. 1 Antisymmetric activation function

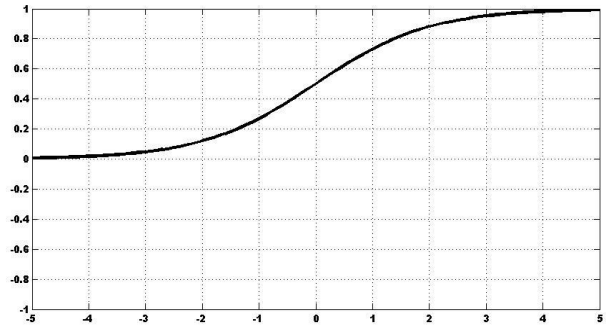

Fig. 2 Non-symmetric activation function

2.3 Maximizing information content

As a general rule, every training example presented to the back-propagation algorithm should be chosen such that they contain maximum possible information [11]. This is done by introducing a training set that results largest training error (in this case bad handwriting) and by the use of a set that is radically different from all those previously used (different fonts for the same character).

2.4 Target values

Desired response d_j of neuron j at the output layer should be offset by some amount $\in$ away from the limiting value of the transfer function (i.e. $d_j = a - \in$ for limiting value of +a and $d_j = -a + \in$ for limiting value of $-a$ to prevent the free parameters of the network being driven to infinity [1]. If this happens, the hidden neurons will be saturated slowing down the learning process. This is done in the opposite way, i.e. by making the limiting values greater than target values.

2.5 Normalization of the inputs

Each input variable should have zero-mean [11]. If input variables are consistently positive, then synaptic weights of a neuron in the first hidden layer can only increase or decrease together. While changing direction, the weight vector of that neuron will go zigzagging through the error surface which is slow. Hence inputs undergo the following steps: mean removal, decorrelation.

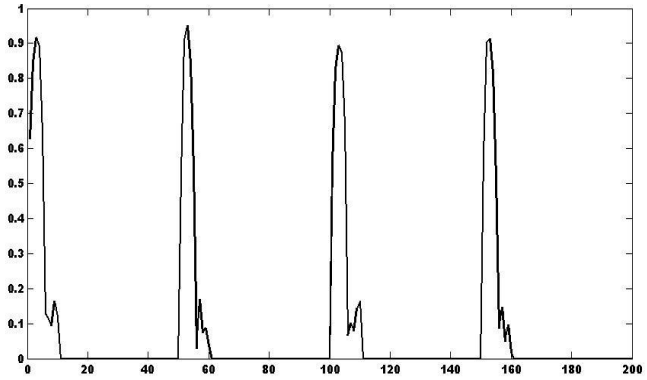

Fig. 3 Input data with non-zero mean

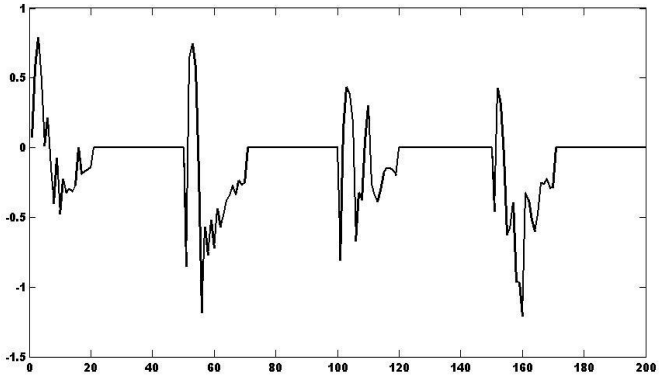

Fig. 4 Mean removed input data

Figure 3 shows the inputs having a non-zero mean and figure 4 shows the data after removing the mean. Decorrelation is done by principal component analysis. In pattern recognition, the dimension of the input vector is large, but the components of the vectors are highly correlated (redundant). It is useful in this situation to reduce the dimension of the input vectors. Principal component analysis (PCA) is an effective procedure for performing this operation [12]. This technique has three effects: it decorrelates the components of the input vectors; it orders the resulting principal components, so that those with the largest variation come first; and it eliminates those components that contribute the least to the variation in the data set.

2.6 Generalization

The error on the training set is driven to a very small value after the network is being trained, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. This happens when the network learns too many input-output examples. It may do so by finding a feature present in the training set but not in the test set. This situation is overfitting or overtraining. When the network is overtrained, it loses the ability to generalize.

It is very difficult to know when to stop training to prevent overfitting. Well, the onset of the overfitting may be identified through the use of cross-validation [13]. The training set is sub-divided into three sub-sets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases as usual. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training, as does the training set error. When the network begins to overfit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped. The test set error is used to compare different models. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this may indicate a poor division of the data set. The whole process is referred to early stopping [12], [14].

3. Results

The network is trained with and without generalization. Figure 5 and 6 shows those two types of training. In figure 5, the curved line is the performance of the training algorithm and straight one is the goal-line. It stopped training at 182 epochs with a performance 9.971e-006, while our desired goal was 1e-005. Another training procedure that can generalize the network to new data is shown in figure 6. To check the progress of training; the training, validation and test errors are plotted.

In Figure 6, the training stopped after 100 iterations because the validation error increased. The result is reasonable, since the test set error and the validation set error have similar characteristics as discussed in section 6, and it doesn't appear that any significant overfitting has occurred. One interesting but expected thing is that the

training stopped at 100 epochs which is earlier than the previous one. Hence the name early stopping.

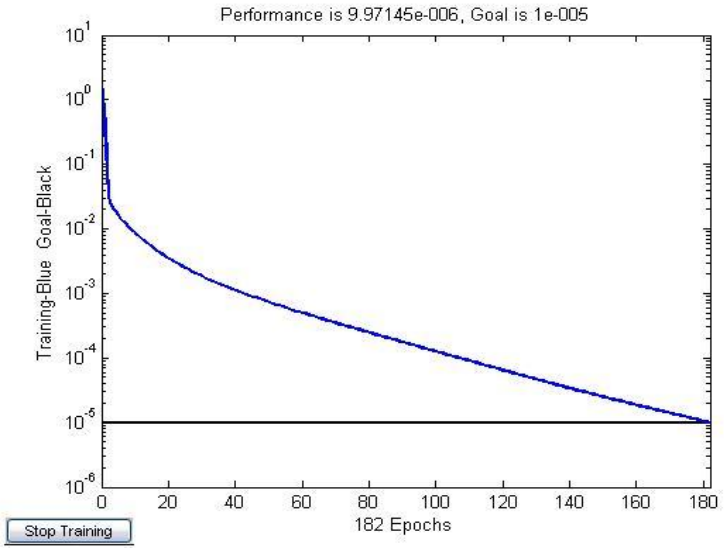

Fig. 5 Training without generalization

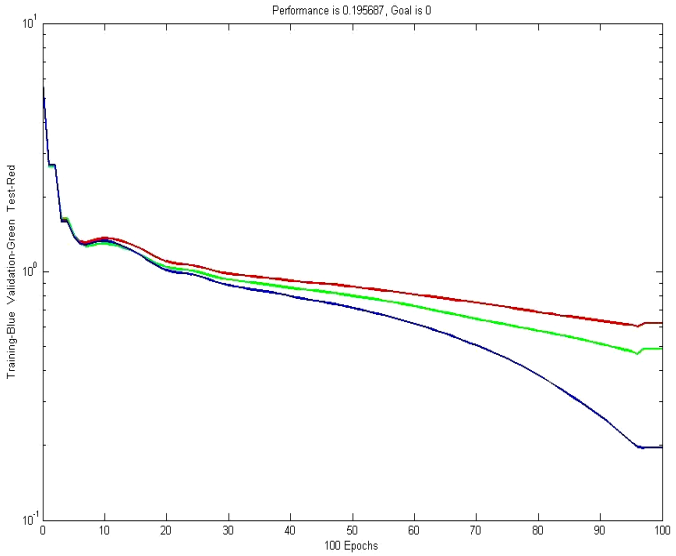

Fig. 6 Training with generalization

Bengali numerical characters are given to the previously trained network. Table 1 has the results for training without heuristics applied and table 2 has results for training with heuristics.

We can see that the accuracy rate has increased by a small amount after applying heuristics for Powell-Beale restarts and Polak-Ribiere update algorithms. We assume that our classifier has certain lacking during the building process. If the classifier was configured properly, it would give much better result.

Table 1: Accuracy without heuristics

Algorithm	***Performance Error***	***Epoch***	***Mis-classification***	***Accuracy %***
Backpropagation with momentum	0.021	1251	115	83.57
Backpropagation with adaptive learning rate	0.07	102	180	74.29
Powell-Beale restarts	0.0121	80	86	87.26
Polak-Ribiere update	0.02	74	95	86.43
Scaled Conjugate Gradient	0.021	150	110	84.29

Table 2: Accuracy with heuristics applied

Algorithm	***Performance Error***	***Epoch***	***Mis-classification***	***Accuracy %***
Backpropagation with momentum	0.40114	9163	102	85.43
Backpropagation with adaptive learning rate	0.69554	62	224	68
Powell-Beale restarts	0.405388	65	78	88.86
Polak-Ribiere update	0.4180	43	84	88.00
Scaled Conjugate Gradient	0.399	81	97	86.14

For training the network, different training algorithms are applied and their responses are compared. It is found that the fastest algorithm is Polak-Ribiere update which is a conjugate gradient algorithm. It converges within 43 epochs and accuracy is also good. Powell-Beale restarts algorithm is stopped at 65 epoch but it has higher accuracy.

4. Conclusions and future work

Design of a neural network is more of an art than a science. This is because there are many factors involved in the design process. Only an optimized design can give better result for a particular problem. In this paper, heuristics are applied to improve the performance of the training algorithm. Training by backpropagation can be improved with the help of heuristics. From the tables the improvement is evident. Accuracy rates are higher in the second table. There are some other algorithms that can be applied here and results can be compared. This work is totally based on the feature extraction technique called transitions [4]. Accuracy can be improved if better preprocessing, feature extraction [2],[3],[5] and segmentation techniques are applied. This is left for the future work.

References

[1] Haykin Simon, "Neural networks: A comprehensive foundation", Chapter 4, Pearson Education, 2003.

[2] Angshul Majumdar, "Bangla Basic Character Recognition Using Digital Curvelet Transform", Journal of Pattern Recognition Research, 2007.

[3] U. Pal, T. Wakabayashi and F. Kimura, "Handwritten Bangla Compound Character Recognition using Gradient Feature", 10th International Conference on Information Technology, 2007.

[4] Gader, P. D., Mohamed, M., and Chiang, J. -H., 1997, "Handwritten Word Recognition with Character and Inter-Character Neural Networks", IEEE transactions on systems, man, and cybernetics 27, pp. 158–164.

[5] Md. Abdur Rahman, Abdulmotaleb El Saddik, "Modified Syntactic Method to Recognize Bengali Handwritten Character", IEEE transaction on instrumentation and measurement, vol. 56, no. 6, December 2007.

[6] A. A. Chowdhury, E. Ahmed, S. Ahmed, S. Hossain and C. M. Rahman, "Optical Character Recognition of Bangla Characters using neural network: A better approach". 2nd ICEE, 2002.

[7] J. U. Mahmud, M. F. Raihan and C. M. Rahman, "A Complete OCR System for Continuous Bangla Characters", Proc. of Conf. on Convergent Tech. for Asia Pacific, 2003.

[8] U. Bhattacharya, B.B. Chaudhuri and S.K. Parui, "An MLP based segmentation method without selecting a feature set", Image and Vision Computing 15 (1997) page 937-948.

[9] Rajsekaran, G.A Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic, and Genetic Algorithms, Synthesis and Applications", Prentice-Hall India, page 34-86.

[10] http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/index.html?/access/helpdesk/help/toolbox/nnet/&http://www.mathworks.com/products/neuralnet/technicalliterature.html

[11] LeCun, Y., "Efficient Learning and Second-order Methods", A Tutorial at NIPS 93, Denver, 1993.

[12] http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/index.html?/access/helpdesk/help/toolbox/nnet/function.html&http://www.mathworks.com/products/neuralnet/

[13] B. Yegananarayana, "Artificial Neural Network". Prentice-Hall India

[14] Amari, S., N. Murata, K.R.Muller, M.Finke, and H. Yang, 1996a. "Statistical theory of overtraining-Is cross-validation asymptotically effective", Advances in Neural Information Processing Systems, vol. 8, pp-176-182, Cambridge, MA: MIT Press.