Analysis of Back Propagation of Neural Network Method in the String Recognition

Amit Kumar Gupta and Yash Pal Singh

Abstract—**This paper aims that analysing neural network method in pattern recognition. A neural network is a processing device, whose design was inspired by the design and functioning of human brain and their components. The proposed solutions focus on applying Back Propagation Algorithm model for pattern recognition. The primary function of which is to retrieve in a pattern stored in memory, when an incomplete or noisy version of that pattern is presented. An associative memory is a storehouse of associated patterns that are encoded in some form. In auto-association, an input pattern is associated with itself and the states of input and output units coincide. When the storehouse is incited with a given distorted or partial pattern, the associated pattern pair stored in its perfect form is recalled. Pattern recognition techniques are associated a symbolic identity with the image of the pattern. This problem of replication of patterns by machines (computers) involves the machine printed patterns. There is no idle memory containing data and programmed, but each neuron is programmed and continuously active.**

Index Terms—**Neural network, machine printed string, pattern recognition, a perceptron-type network, learning, recognition, connection.**

I. Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the biological nervous systems, such as the brain. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. A Neural Network is configured for pattern recognition or data classification, through a learning process. In biological systems, Learning involves adjustments to the synaptic connections that exist between the neurons. Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements working in parallel to solve a specific problem. Neural networks learn by example. A neuron has many inputs and one output. The neuron has two modes of operation (i) the training mode and (ii) the using mode. In the training mode, the neuron can be trained for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the training rule is used. Neural network has many applications. The most likely applications for the neural networks are (1) Classification (2) Association and (3) Reasoning. An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. Four significant approaches to PR have evolved. These are [5].

Statistical pattern recognition: Here, the problem is posed as one of composite hypothesis testing, each hypothesis pertaining to the premise, of the datum having originated from a particular class or as one of regression from the space of measurements to the space of classes. The statistical methods for solving the same involve the computation other class conditional probability densities, which remains the main hurdle in this approach. The statistical approach is one of the oldest, and still widely used [8].

Syntactic pattern recognition: In syntactic pattern recognition, each pattern is assumed to be composed of sub-pattern or primitives string together in accordance with the generation rules of a grammar string of the associated class. Class identifications accomplished by way of parsing operations using automata corresponding to the various grammars [15, 16]. Parser design and grammatical inference are two difficult issues associated with this approach to PR and are responsible for its somewhat limited applicability.

Knowledge-based pattern recognition: This approach to PR [17] is evolved from advances in rule-based system in artificial intelligence (AI). Each rule is in form of a clause that reflects evidence about the presence of a particular class. The sub-problems spawned by the methodology are:

1. How the rule-based may be constructed, and
2. What mechanism might be used to integrate the evidence yielded by the invoked rules?

Neural Pattern Recognition: Artificial Neural Network (ANN) provides an emerging paradigm in pattern recognition. The field of ANN encompasses a large variety of models [18], all of which have two important string.

1. They are composed of a large number of structurally and functionally similar units called neurons usually connected various configurations by weighted links.
2. The Ann's model parameters are derived from supplied I/O paired data sets by an estimation process called training.

II. Methodology

Different neural network algorithms are used for

Manuscript received May 28, 2011; revised June, 22, 2011.

Amit Kumar Gupta is MCA Department KIET, Ghaziabad(UP),India (E-mail: amitid29@gmail.com)

Yash Pal Singh is Reader and Head,CSE Deptt BIET, Jhansi (UP), India (E-mail:yash_biet@yahoo.co.in)

recognizing the pattern. Various algorithms differ in their learning mechanism. Information is stored in the weight matrix of a neural network. Learning is the determination of the weights. All learning methods used for adaptive neural networks can be classified into two major categories: supervised learning and unsupervised learning. Supervised learning incorporates an external teacher. After training the network, we should get the response equal to target response. During the learning process, global information may be required. The aim is to determine a set of weights, which minimizes the error. Unsupervised learning uses no external teacher and is based on clustering of input data. There is no prior information about input's membership in a particular class. The string of the patterns and a history of training are used to assist the network in defining classes. It self-organizes data presented to the network and detects their emergent collective properties. The characteristics of the neurons and initial weights are specified based upon the training method of the network. The pattern sets is applied to the network during the training. The pattern to be recognized are in the form of vector whose elements is obtained from a pattern grid. The elements are either 0 and 1 or -1 and 1. In some of the algorithms, weights are calculated from the pattern presented to the network and in some algorithms, weights are initialized. The network acquires the knowledge from the environment. The network stores the patterns presented during the training in another way it extracts the features of pattern. As a result of this, the information can be retrieved later.

III. Problem Statement

The aim of the paper is that neural network has demonstrated its capability for solving complex pattern recognition problems. Commonly solved problems of pattern have limited scope. Single neural network architecture can recognize only few patterns. In this paper discusses on neural network algorithm with their implementation details for solving pattern recognition problems. The relative performance evaluation of this algorithms has been carried out. The comparisons of algorithm have been performed based on following criteria:

(1) Noise in weights
(2) Noise in inputs
(3) Loss of connections
(4) Missing information and adding information.

IV. Back Propagation Algorithm

Multilayer feed forward network can be applied to a variety of classification and recognition problems. In back propagation algorithm, it is not necessary to know the mathematical model of the classification of recognition problem to train and then recall information from the network. In this algorithm, if suitable network architecture is selected and sufficient training set is presented, the network may give acceptable solution.

The back propagation algorithm allows input/output mapping within multilayer feed forward neural networks. Input patterns are submitted during training sequentially. If classification of a submitted pattern is found to be erroneous, then the weights and threshold are adjusted so that current least mean square classification error is reduced. The input/output mapping, comparison of the target and output values and adjustments are continued until all the training patterns are learned within acceptable error. During the classification phase, the trained neural network operates in a feed forward manner. The weight adjustments by the learning rule propagate backward from the output layer through hidden layer to the input layer[23].

Multilayer feed forward network shown in figure no-4.1

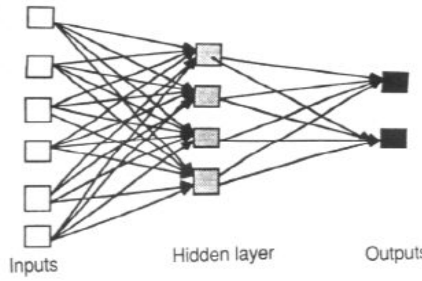

Fig. 4.1.An example of a simple Multilayer feed forward network

V. Algorithm

Consider that there are P training pairs {a_1, b_1; a_2, b2 …ap, b_p} a_i is the input vector and b_i is target output vector.

Step 1: Weights W and V are initialized at some random value. V is the weight matrix between input and hidden layer; W is the weight matrix between hidden and output layer.

Step 2: Pattern Z is submitted to the network and responses of neurons of hidden and output layer are calculated. y is the output hidden layer of neurons and o is the output of output layer neurons.

$$y = \Gamma[V_Z]$$
$$o = \Gamma[W_y]$$

where Γ is a nonlinear operator.

$$\Gamma[.] = \begin{bmatrix} f(.) & 0 & \cdots & \cdots & 0 \\ 0 & f(.) & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & f(.) \end{bmatrix}$$

Step 3: The error is calculated as

$$E \leftarrow E + \frac{1}{2}\|d - o\|^2$$

Step 4: Errors δ_o and δ_y are calculated as

$$\delta_o = \frac{1}{2}\left[(d_k - o_k)(1 - o_k{}^2)\right]$$
$$\delta_y = w_j^t \delta_o f_y^'$$
$$f_y^' = \frac{1}{2}\left[1 - y_j{}^2\right]$$

Step 5: weight s of output layer are adjusted by

$$W \leftarrow W + \eta \delta_o y^k$$

Step 6: weight of hidden layer are adjusted by

$$V \leftarrow V + \eta\delta_y z^t$$

Step 7: If there are more patterns in the training set then whole process from step 2 is repeated. If there are no more patterns left then it is checked whether $E<E_{max}$. If yes then the process is complete. If no then a training cycle is started from step2[23].

VI. Result

Back Propagation Algorithm **,**The network architecture was used 35-16-26. Characters are represented by a pattern grid of 5×7. All the 26 characters were stored as 35-dimensional vector with entries 0 and 1. The target output vector is also stored which is identity matrix of size 26×26. An error goal of 0.01 was taken. The network was trained for maximum 5000 epochs. After training the network was capable of recognizing all the characters correctly.

For Back Propagation Algorithm**,** the range of % increase of weights, for which does work properly is, 0.0485-53.85 applicable and the range of % increase of weight, for which there is 0-10.77.

Effect of Noise in Inputs on Algorithm

Back Propagation Algorithm , When random numbers are added to the weight matrices of all layers, 21 characters are recognized correctly. If at one time, noise is introduced in weight only one layer then all the characters is recognized correctly.

Loss of Connection

In the network, There are three sets of weights. First connecting input to neurons of input layer. Second connecting input layer neuron to the hidden layer neurons. Third connecting hidden layer neurons to output layer neurons. If the weights connecting 15 input to 35 neurons of input layer, 15 neurons of input layer to 15 neurons of hidden layer and 8 neurons of hidden layer to all the 26 neurons of out put layer are removed, than also the network recognizes all the characters correctly. So total number of redundant connection is 958. If connection of input neuron's to all the output neuron is removed, and the pixel corresponding to that neuron number is off than it makes no difference. But if that pixel is on, in the output that becomes off.

Missing Information

TABLE 6.1: MISSING INFORMATION: NO OF PIXELS THAT CAN BE MADE OFF IN THIS ALGORITHM

Character	Adaptive Resonance Theory
A	6
I	5
L	4
M	4
P	2
X	5

Missing information means some of the on pixels in pattern grid are made off. For the algorithm, how many information we can miss so that the strings can be recognized correctly varies from string to string. We cannot switch off pixel from any place. Which pixel is being switched also matters. For few strings Table-6.1 shows the number of pixels that can be switched off for all the stored strings in algorithm.

Adding Information

There are three weight matrices. Weight connecting inputs to the input layer, input layer to hidden layer, and hidden layer to output layer.

If we add 0.08 to all the weights, then it does have any effect on the network. If we increase weights of only one layer at one time then a large number can be added. It has been observed that if 0.4 is added to any of three weights at one time, then the network works as before.Table-6.2 shows detailed description about the number of pixels that can be made on for all the strings that can be stored in networks.

TABLE 6.2: ADDING INFORMATION: NO OF PIXELS THAT CAN BE MADE IN THIS ALGORITHM

Character	Back Propagation Algorithm
A	6
I	11
L	5
M	11
P	10
X	9

Merits and Demerits

This method provides accurate input –output mapping. The advantage of this method is that each adjustment step is computed quickly. All the patterns need not be presented simultaneously. Each step is computed without finding an overall direction of the descent for the training cycle.

Multilayer feed forward neural networks trained by back propagation algorithm are slow to learn.

TABLE 7.1: PERFORMANCE OF ADAPTIVE RESONANCE THEORY ALGORITHMS UNDER DIFFERENT CRITERION

Criteria	Back Propagation Algorithm
Number of Neurons	76
Number of Unknowns	2216
Capacity	26
Effect of Noise in Weight (Random No. Added)	21
Effect of Increase of Weight	0.08
Noise in Input	N.S.
Range of No. of Pixels that can made off	N.A.
Range of No. of Pixels that can made on	N.A.
No of Connection we can lose (wt=0)	958

VII. CONCLUSION

The performance of ART1 algorithm has been studied under nine criteria. It has been observed that a certain algorithm performs best under a particular criterion. The algorithm have also been compared based on the number of neurons and the number of unknowns to be computed. The detailed description in Table-7.1

REFERENCES

[1] Basit Hussain and M.R.Kabuka, "A Novel Feature Recognition Neural Network and its Application to string Recognition",IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 16, No. 1,PP. 98-106, Jan. 1994.

[2] Z.B. Xu, Y.Leung and X.W.He," Asymmetrical Bidirectional Associative Memories", IEEE Transactions on systems, Man and cybernetics, Vol.24, PP.1558-1564, Oct.1994.

[3] Schalkoff R.J.," Pattern Recognition: Statistical Structured and Neural; Approach", John Wiely and Sons, 1992

[4] Fukuanga K., "Stastitical Pattern recognition-2nd ed."., Academic Press, 1990

[5] V.K.Govindan and A.P. Sivaprasad, "String Recognition- A review", Pattern recognition, Vol.23,No. 7, PP. 671-683, 1990

[6] Fu K.S., "Syntactic Pattern Recognition," Prentice Hall, 1996.

[7] Y.K. Chen, J.F. Wang, Segmentation of single or multiple touching handwritten numeral strings using background and foreground analysis, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1304–1317.

[8] Jacek M. Zurada, "Introduction to artificial Neural Systems", Jaico Publication House. New Delhi, INDIA

[9] C.L. Liu, K. Nakashima, H. Sako,H.Fujisawa,Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognition 36 (2003) 2271–2285.

[10] R. Plamondon, S.N. Srihari, On-line and o8-line handwritten recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 62–84.

[11] C.L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37 (2004) 265–279.

[12] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic segmentation of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1438–1454.

[13] U. Bhattacharya, T.K. Das, A. Datta, S.K. Parui, B.B.Chaudhuri, A hybrid scheme for handprinted numeral recognition basedon a self-organizing network andMLP classiHers, Int. J. Pattern Recognition Artif. Intell. 16 (2002) 845–864.

[14] C.L. Liu, H. Sako, H. Fujisawa, Effects of classifier structure and training regimes on integrated segmentation and recognition of handwritten numeral strings, IEEE Trans. Pattern Recognition Mach.Intell. 26 (11) (2004) 1395–1407.

[15] K.K. Kim, J.H. Kim, C.Y. Suen, Segmentation-based recognition of handwritten touching pairs of digits using structural features, Pattern Recognition Lett. 23 (2002) 13–24.

[16] Jain, A., Duin, P., Mao, J., 2000. Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Machine Intell. 22 (1), 4–37.

[17] Tax, D.M.J., van Breukelen, M., Duin, R.P.W., Kittler, J., 2000. Combining multiple classifiers by averaging or by multiplying?Pattern Recognition 33, 1475–1485.

[18] L.S. Oliveira, R. Sabourin , F. Bortolozzi , C.Y. Suen" Impacts of verification on a numeral string recognition system" Elsevier Science, Pattern Recognition Letters 24 (2003) 1023–1031.

[19] U. Pal, B.B. Chaudhuri "Indian script character recognition: a survey" Elsevier Science, Pattern Recognition, 37 (2004) 1887 – 1899

[20] Xiangyun Ye, Mohamed Cheriet ,Ching Y. Suen "StrCombo: combination of string recognizers" Elsevier Science, Pattern Recognition Letters 23 (2002) 381–394.

[21] Symon Haikin, "Neural Networks: A comprehensive Foundation", Macmillan College Publishing Company, New York, 1994.

[22] Javad Sadri, ChingY. Suen, Tien D. Bui "Agenetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings" Elsevier Science, Pattern Recognition 40 (2007) 898 – 919

[23] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 2003.