Bengali character recognition using Bidirectional Associative Memories (BAM) neural network

M. M. Mahbubul Syeed, Fazlul Hasan Siddiqui, Abu Saleh Abdullah Al-Mamun, Syed Khairuzzaman Tanbeer and M. Abdul Mottalib

Department of Computer Science and Information Technology (CIT)
Islamic University of Technology, Board Bazar, Gazipur-1704

Emails: rajit_cit@hotmail.com, fhsani@yahoo.com, mamuncitiut@hotmail.com, tanbeer2000@yahoo.com

Abstract: This paper presents the recognition features of Bengali text using BAM (Bidirectional Associative Memories) neural network with a proposal of feature extraction procedure of a Bengali character. To do this, the conventional methods are used for text scanning to segmentation of a text line to a single character. In this paper an efficient procedure is proposed for boundary extraction, scaling of a character and the BAM neural network which increases the performance of character recognition are used.

Keyword: Bengali, Character, Neural Network, BAM (Bidirectional Associative Memories), Feature, Scaling, Recognition.

1. INTRODUCTION

Bengali character recognition has become an active area of research for last few years with a wide variety of applications. Lots of works already have been done on printed Bengali characters [1] and handwritten Bengali characters [2].

In this paper an approach on recognizing scanned Bengali text is proposed. The Bengali text is scanned first with a scanner and converted into an image format. Then applying several techniques the characters in the text are separated. These separated characters are then applied to a classifier that recognizes the characters as several Bengali characters stored in memory. In this phase BAM (Bidirectional Associative Memories) neural network model is used.

Neural network is developed to perform some of the activities of human brain. As the recognition of images, voice etc. are performed most efficiently and accurately by human brain, so the artificial neural networks are tried to be developed in computer to perform these recognitions.

Like human brain a neural network has a parallel-distributed architecture with a large number of nodes and connections. Each connection points from one node to another and is associated with a weight. A typical model of neural network is shown in figure 1.

Construction of a neural network involves the following tasks:

a) Determination of network properties, as network topology or connectivity, type of connection between the nodes of the network, the order of connection is decided.

b) Determination of node properties, as the activation range (discrete [0 & 1] or continuous [0,1]) of a node, type of node activation function (Hard limiting function or Sigmoid function) is determined.

c) Determination of system dynamics, as weight initialization scheme, the activation calculation formula, the network learning rule (weight adjustment) is determined.

A neural network classifier technique say, BAM is used for recognition. BAM is an associative neural network. An associative neural network is one that retrieves an object or memory based on part of the object itself. The term memory in associative network can be defined as, If a binary n-dimensional vector X is a memory then for each component (neuron) i = 1, 2, …., n

\[ O_j = F \sum_{i=1}^{n} (W_{ij}X_i) \]

\[ F= Activation\ function \]

Figure 1: The neural network computational model
\[ X_i = F_h \left( \sum_{j=0}^{n} W_{ij} X_j \right) \]

Here, 
\( F_h \) is the hard limiting function. 
\( W_{ij} \) is weight from node \( j \) to node \( i \). 
\( X_j \) is the input on node \( j \).

This means that \( X \) is a memory if the network is stable at that point. Detailed discussion on BAM is given on the section of recognition.

2. PHASES OF BENGALI TEXT RECOGNITION

The full cycle of Bengali text recognition consists of the following parts:
- Data acquisition.
- Text digitization and noise removing.
- Oblique/skew detection and removing.
- Block detection.
- Segmentation.
- Feature extraction.
- Learning and character recognition by neural network.

A block diagram representation of this recognition system is shown in figure2.

The Bengali text is first scanned by a scanning device and then stored in digital image format. The histogram threshold technique is used for its better result. Other techniques are also there, as [3] and [4].

From this digital image noise is cleaned and the oblique is removed. For this, an algorithm proposed in [5] is used. Other skew detection algorithms such as algorithms based on [6], [7], [8], [9] and [10] are also found in the literature.

Then the text regions are separated from non-text regions by using any one of the methods mentioned in [9], [10], [13], [14]. Among these Page segmentation and classification method in [14] is exercised here.

Then the lines are segmented from the text, then each line is segmented into words and finally the words are segmented into constituent characters. In this case the algorithm in [5] for segmentation is used. These characters are then fed for Feature Extraction.

3. FEATURE EXTRACTION

Feature extraction [15] is an important part for character recognition. Feature Extraction helps to convert the segmented character pixels into the approximate binary valued character. There are two approaches for feature extraction, namely statistical and structural approach. Here feature extraction has been done in two phases: Boundary extraction and scaling.

3.1. Boundary extraction

It is necessary to find the boundary position of the character image. In this phase a single character placing in a single window will be extracted by horizontal and vertical scanning starts from the upper left and bottom right position of the window. This scanning is halted only when it faces a single pixel. The proposed algorithm for boundary extraction is given below:
1. Get the square boundary within which a single character exists.
2. Continue horizontal scanning from the top most line
towards the bottom of the window until meet a single pixel.
3. Continue vertical scanning from the left most line towards the right of the window until meet a single pixel.
4. Say the innermost two scanned lines meet at a pixel $P$.
5. Similarly continue horizontal scanning from the bottom most line towards the top of the window until a single pixel is met.
6. Continue vertical scanning from the right most line towards the left of the window until a single pixel is met.
7. Say the innermost two scanned lines meet at a pixel $Q$.
8. So the extracted boundary is found whose upper-left boundary is positioned at $P$ and bottom-right boundary is positioned at $Q$.

3.2. Scaling

After the boundary position is determined the character is needed to be scaled. Say the horizontal and vertical lengths of the extracted character are denoted by $L_H$ and $L_V$ respectively. It is to be converted into 16x16 matrix containing binary digits (as the inputs of the BAM neural network is taken into 16x16 matrix), so the length of the each unit region is $X$ (horizontal) and $Y$ (vertical). Where $X=\frac{L_H}{16}$, $Y=\frac{L_V}{16}$.

Each pixel in a unit region causes folding of some other pixels around the original pixel. From our experiment it is found that the amount of flooded pixels is 0.03 times of the maximum capacity of each unit region. Now to convert each unit region into binary matrix the following procedure is used:

For each unit region
If
    The region contains 38% of the maximum pixels of each unit region
Then
    Set it as a matrix block to 1
Else
    Set it as a matrix block to 0

In the experiments it is found that if a considerable 38% area of a unit region is flooded by pixels, the maximum accuracy for converting that region into binary matrix [1] can be obtained. The overall activities for feature extraction are shown in the figure 3.

4. RECOGNITION

The neural network BAM is used for recognition. A model of BAM neural network is shown in the figure 4. As shown in this figure BAM has two layer recurrent architecture in which the backward weight (the weights from the output to the input) matrix is the transpose of the forward weight (from input to the output) matrix. The size of the input and output layers are determined by the dimensions of the pairs of associated vectors. As in the figure, an input pattern $P$ is applied to the weight matrix $W$ and produces an output vector $Q$, which is then applied to $W^T$ to produce a new input vector. This process is repeated until the network reaches a stable state (a state in which the input vector do not change in further iteration). This vector is then compared with the previously stored patterns that were used to train the network. The stored pattern which is best matched with the pattern in the stable state is taken as the output.

4.1. Working mechanism of BAM

BAM neural network must be given some stored input patterns and there associated output patterns for its learning. During the learning process the network calculate its weights for each link (from each input node to each output node and vice versa) using these given stored patterns. For our purpose we trained our network with fifty Bengali characters and their associated output patterns. Example of such patterns are given below:
BAM network works using the following procedure:

1. Calculation of weight
The calculation of forward weight (as in figure 4) is done by the following algorithm:

\[ W_{ji} = \sum_{p=1,q=1}^{m} (P_{i,p}Q_{j,q}) \]

Here, \( m \) is the number of pairs stored patterns.

\( W_{ji} \) is the connection weight from unit \( i \) to unit \( j \).

\( P_{i,p} \) is the \( i \)th component in the pattern vector \( p \).

\( Q_{j,p} \) is the \( j \)th component in the pattern vector \( q \).

The backward weight is calculated by \( W_{ij} = W_{ji} \).

2. Activation calculation
   a. Initialization of the input units at time 0 with
      \( O_j(0) = P_i \).
      Here, \( P_i \) is the \( i \)th component of the input pattern.
      \( O_j(0) \) is the \( i \)th component of the output at time \( t=0 \).

   b. At time \( t(t>0) \)
      \( O_j(t+1) = F_h(\sum_i W_{ji} O_i(t)) \)
      Here, \( O_j(t) \) is the activation level of unit \( j \) at time \( t \) and
      \( F_h \) is a hard limiting function.
      \( F_h \) given by the following values:
      \( F_h(a) = 1, \) if \( a > 0 \).
      \( F_h(a) = -1, \) if \( a < 0 \).
      \( O_j(t), \) if \( a = 0 \).

3. Step 2 is repeated until equilibrium (i.e. the activation level remains unchanged with further iterations). Then this pattern is compared with the stored patterns and the best-matched stored pattern is taken as the output.

4.2 Performance of BAM

For testing the performance of BAM, fifty stored input Bengali characters and their associated output patterns are used. Each input pattern is in a 16x16 matrix, hence there are 256 input nodes in the network and each output pattern is in a 4x4 matrix, hence 16 output nodes are there in the network. The performance is measured in the following criteria:

a) In case of deformed/distorted input to the network:
If the input pattern is very close to the actual pattern (character) then BAM network gives the maximum accuracy in recognizing the character. The accuracy is shown in the accuracy diagram below.

![Accuracy diagram](image)

This diagram shows the recognition accuracy of BAM in different deviations. For example, 6% deviation means, the bits in the input character matrix (16x16) is changed by 6% from the actual or correct character.

b) Execution time for recognition (in different platforms):
The average execution time for recognizing a character in different machines, as Pentium IV (1.5 GHz), II (350 MHz), III (700 MHz) and Seleron-733 MHz (keeping all other configurations same) are measured. The time against different machines are represented in diagram 1.

<table>
<thead>
<tr>
<th>Deviation (%)</th>
<th>0</th>
<th>6</th>
<th>12</th>
<th>18</th>
<th>24</th>
<th>30</th>
<th>48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness (%)</td>
<td>99.9</td>
<td>97.6</td>
<td>95.4</td>
<td>90.5</td>
<td>83.3</td>
<td>66.6</td>
<td>50.2</td>
</tr>
</tbody>
</table>

Input pattern “BA”

```
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

Associated output pattern:

```
-1,-1,-1,-1,
1, 1, 1, 1,
1, 1, 1, 1,
1, 1, 1, 1,
```
5. CONCLUSION

The nature of BAM neural network is to reach a stable state (a state at which successive input patterns to the network do not change in further iteration). This state is reached by considering the given input pattern and the information about the stored patterns that was used in the learning phase of the network. For this reason in the case of largely deformed input patterns BAM gives higher accuracy in recognizing the character.

Again the learning time of BAM is negligible compared to other networks (as for Back Propagation Network the average learning time is about 54.945 millisecond for only two input nodes with one hidden layer of a three layer network, this time is further increased with the increment of input nodes and the hidden layers) in Pentium-IV (1.5 GHz).

Code for this system is implemented using Turbo C++ (version 3.0).

REFERENCES
