Neural Network Approach To Mathematical Expression Recognition System

Surendra P. Ramteke
Dhanashri V. Patil
Nilima P. Patil

Department of E&TC Engineering, SSBT’s College of Engineering & Technology, Jalgaon. (M.S.), INDIA

Abstract— Now a days there is a lot of development in smart devices which are combination of human intelligence and machines. Character recognition is one of the example of smart device and Mathematical expression recognition is belongs to such device which is developed to recognize handwritten as well as printed mathematical symbols and expressions. This system converts scientific or engineering documents in to its electronic form. In this work a purely offline approach is utilized. The main phases carried out to achieve the goal are pre-processing, segmentation, feature extraction, classification and recognition of symbols as well as mathematical expression. Out of these phases mainly important are feature extraction and classification because it affects overall accuracy of the system. In this system centroid and bounding box are the main features that are extracted from each character and this system is accomplished using neural network approach for the recognition of the expressions and symbols which provides 90% recognition rate.

Key Words— mathematical expression recognition, segmentation, feature extraction, classification.

I. INTRODUCTION

There is a wealth of mathematical knowledge that could be potentially very useful in many computational applications. Mathematical expression recognition is one of the research topics from several decades but it is still an area of research topic because there are lots of challenges in this system. It is very important regarding scientific document image analysis and this system have applications like scientific document digitization, information retrieval or accessibility for blind people [14]. The input for this system is mathematical expressions and symbols. The input of mathematical expression into computers is very difficult than that of plain text because mathematical expression contains special symbols, Latin or Greek letters and different operators with digits [6].

For on-line approach system utilizes the temporal information about strokes input. Where off-line recognition deals with the image representation of mathematical expressions, which may be printed or handwritten. This system faces lots of difficulties because if we consider handwritten mathematical expression then there are variations occur in the size, font of symbol, writing style varies person to person and quality of image also matters.

Figure.1 shows the system Overview for mathematical expression recognition [15]. The first step carried out is data acquisition. This data is then acquired from optical scanner. Next step is pre-processing image cleaning takes place. Along with that image is converted to form of suitable subsequent processing like size normalization, skeletonization and noise removal takes place. After pre-processing image is applied to segmentation and feature extraction. In next step the document is segmented in sub components and separating of each character is takes place. After segmentation the feature set which is useful for training of the system and recognition is extracted.

II. STEPS FOR RECOGNITION

During classification, a character is placed in the appropriate class to which it belongs. In recognition step training and testing are apply.

A. Data Collection and Pre-processing

In this section the important processes that are carried out in the mathematical expression recognition are discuss briefly [4]. At first expression is written on a white plain paper with use of black color marker pen. The program designed for this system is able to read the expression may available in the file format of .jpg, .bmp or .gif.

Figure 1: Overview of Mathematical Expression Recognition System.

During classification, a character is placed in the appropriate class to which it belongs. In recognition step training and testing are apply.

A. Data Collection and Pre-processing

At first data is collected from different persons and a database is created [18]. In this system we have considered 0 to 9 numbers, some alphabetical characters and symbols as well as special characters. All the numbers, symbols and characters are putted in sequential form. They are isolated with certain distance from each other and a fixed sequence has been maintained throughout in rows and column. The database is as shown in fig. 2 as follows. Here we have considered handwritten as well as printed set of characters.
On the input image to the system first pre-processing is carried out. We have to normalize the expression, resize it to proper dimensions, and remove the background noise. The raw data is subjected to different processes under pre-processing so that it can be applied to further. It essentially enhances the image makes it suitable for segmentation. This produces data which is easy for character recognition of system so it can operate accurately. The main functions carried out under this step are as follows.

1. Read input image.
2. Convert RGB image into gray scale image.
3. Convert gray scale into black and white image.
4. Remove component whose size less than 50 pixels.
5. Invert and reshape image.

First the input image is read. This is generally available in color format so it is converted into gray scale image first. RGB to gray conversion takes place with certain values 0.2989, 0.5870 and 0.1140 respectively for RGB. And this gray scale image is then converted in black and white image with the threshold value of 0.5. To perform certain morphological operations it is necessary to carried out this conversion in to binary image. Image cleaning is takes place to remove unwanted spots from the image. The component whose size is less than 50 pixels is removed from the image. After that image inversion and reshaping takes place the result for this process is as shown.

\[ y = m x + c \]

Figure 3: input to the preprocessing

\[ y = m x + c \]

Figure 4: output of preprocessing

### B. Segmentation

In the segmentation, the input image is segmented into individual characters and then, each character is resized into m x n pixels towards the training network. In the proposed system, the pre-processed input image is segmented into isolated characters by assigning a number to each character using a labeling process. In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character [10].

This labeling provides information about number of characters in the image. In the labeling operation two values are find out first is total number of objects connected in the image and second is each and every component present in the image an integer value greater than 0 is assigned.

### III. Feature Extraction

After pre-processing and segmentation step feature set is extracted this is further used for training and recognition step. Feature sets play one of the most important roles in a recognition system. A good feature set should represent characteristic of a class that helps distinguish it from other classes, while remaining invariant to characteristic differences within the class.

1. After labeling region properties are find out from the labeled component.
2. Rectangle is drawn with respect to all number of components, bounding box, position, and edge color, and radius and line width.
3. Properties compute the features that are area, bounding box and centroid.
4. Calculate centroid with respect to two directions horizontally as well as vertically and store it in two variables column wise.

Area finds out number of pixels in the region. Centroid means to find out weights of component and it locates the centre of component. For these numbers of dots are first find out and then average value is taken. Center is calculated of each and every component. The first element of Centroid is the horizontal coordinate or x-coordinate of the center of mass, and the second element is the vertical coordinate or y-coordinate. Fig shows the bounding box and centroid of the image.

As shown in figure the centroid is indicated with blue color * indicate in each component and a bounding box is drawn with red color box. And values of centroid in interfer form for the same expression is as follow.

\[ \begin{align*}
    x &= 66.3574 \quad 118.5674 \quad 120.3139 \quad 212.2624 \quad 296.2003 \\
    y &= 65.7469 \quad 72.6738 \quad 56.9854 \quad 53.5622 \quad 57.5554 \\
    &\quad 51.5280 \quad 50.4820
\end{align*} \]

Here in the expression there are 7 components are present so seven x-value of centroid and seven y-value of centroid
are evaluated. In this expression equal to (=) component is treated as two separate components. Means upper horizontal line and lower horizontal line both are treated as different components.

The smallest rectangle containing the region are bounding box. Those objects which are not present in bounding boxes are the unwanted components and these are discarded. Width to height ratio (WH) of bounding box can be expressed as:

\[ WH = \frac{w}{h} \]

The relative height ratio (RH) and relative width ratio (RW) also required for the bounding box [15].

IV. CLASSIFICATION

A Neural Network is defined as a computing architecture that consists of massively parallel interconnection of simple neural processors. It can perform computations at a higher rate compared to the classical techniques because of its parallel nature [4]. ANN is inspired by the way biological nervous system such as the brain process information. The key element of this paradigm is the novel structure of the information processing system. And it is composed of large number of highly interconnected processing elements (neurons). ANN is configured for specific application, such as data classification.

When ANN processed data then there are two main stages that are training stage and classification stage. In classification stage samples are passed as input to the ANN, resulting an output representing what ANN believes to be the most correct output. To be a successful classification it must be preceded by a training stage [12].

![Diagram of a neural network](image)

**Training Algorithm**

1. Initialization of weights.
2. Feed forward of input training patterns.
3. Calculation of back propagation error.
4. Updating weights and biases.

The neurons present in the hidden and output layer have biases, which are connections from unit whose activation is always one. The inputs apply to BPN and output from it is always either binary or bipolar.

**Initialization of weights**

Step 1: Initialize weights and biases.

Step 2: Until stopping condition is false repeat step 3 to 10.

Step 3: For each training pair do steps 3 to 9.

**Feed forward of input training patterns**

Step 4: Each input unit receives input signal \( x_i \) and transfer it to the next layer which is nothing but hidden layer.

Step 5: Each hidden unit \( z_j \) sums it weighted input signals to calculate net input which is given by

\[ z_{inj} = v_{oj} + \sum x_i v_{ij} \]  \( \text{(1)} \)

Calculate the output of hidden unit by applying its activation function over \( z_{inj} \):

\[ z_j = f(z_{inj}) \]

The output signal from hidden layer is transferred to output layer.

Step 6: For each output unit \( y_k \), calculate net input.

\[ y_{ink} = w_{ok} + \sum z_j w_{jk} \]  \( \text{(2)} \)

Apply activation function to compute the output signal which is given by \( y_k = f(y_{ink}) \).

**Calculation of back propagation error**

Step 7: Each output unit \( y_k \) receives a target pattern corresponding to input training pattern and compute error correction term.
\[ \delta_k = (t_k - y_k)f'(y_{\text{inh}}) \quad (3) \]

On the basis of basis of calculated error correction term, update changes in weight and biases: \( \Delta w_{jk} = a\delta_k z_j \) and \( \Delta w_{ok} = a\delta_k \). It sends \( \delta_k \) to hidden layer backward.

**Calculation of back propagation error**

Step 8: Each hidden unit sums its delta input from output unit:

\[ \delta_{\text{inj}} = \sum_{k=1}^{m} \delta_k w_{jk} \quad (4) \]

The term \( \delta_{\text{inj}} \) gets multiplied with derivative of \( f(z_{\text{inj}}) \) to calculate the error term.

\[ \delta_j = \delta_{\text{inj}} f'(z_{\text{inj}}) \quad (5) \]

On the basis of calculated \( \delta_j \), update the change in weights and biases: \( \Delta v_{ij} = \alpha \delta_j x_i \) and \( \Delta v_{oj} = \alpha \delta_j \)

**Updating weights and biases**

Step 9: Each output unit \( y_k \) updates the bias and weights:

\[ w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk} \quad (6) \]
\[ w_{ok}(\text{new}) = w_{ok}(\text{old}) + \Delta w_{ok} \quad (7) \]

Each hidden unit updates its bias and weights:

\[ v_{ij}(\text{new}) = v_{ij}(\text{old}) + \Delta v_{ij} \quad (8) \]
\[ v_{oj}(\text{new}) = v_{oj}(\text{old}) + \Delta v_{oj} \quad (9) \]

Step 10: Check for the stopping condition. The stopping condition may be a certain number of epochs reached [12, 20].

**Network parameters:**

Table 1 shows the parameters that are set for network training and its respective values also listed [17].

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer Function</td>
<td>logsig</td>
</tr>
<tr>
<td>Network training function</td>
<td>Gradient descent</td>
</tr>
<tr>
<td>Learning Rule</td>
<td>Momentum</td>
</tr>
<tr>
<td>network performance function</td>
<td>Mean squared error (MSE)</td>
</tr>
<tr>
<td>Learning constant</td>
<td>0.01</td>
</tr>
<tr>
<td>Goal</td>
<td>0.001</td>
</tr>
<tr>
<td>Number of Epochs</td>
<td>5000</td>
</tr>
<tr>
<td>Number of iteration after which output have shown</td>
<td>20</td>
</tr>
<tr>
<td>Momentum Term</td>
<td>0.95</td>
</tr>
</tbody>
</table>

**Results:**

Here in this approach we have used GUI, so in output window we have created two axes, two buttons that are read expression and recognize. It consists of two text boxes. Her we have read handwritten expression first and on second axis each and every component is recognized and on output text box recognized expression is generated in printed form.
Table 2: Comparison of different Classifiers

<table>
<thead>
<tr>
<th>Author</th>
<th>Method</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taik Heon Rhee, Jin Hyung Kim[8]</td>
<td>Structural Analysis using hypothesis generation</td>
<td>89.0%</td>
</tr>
<tr>
<td>N. Venkateshrao, A. Shrikrishna, B. Ravindrababu, G. Rama Mohan Babu[12]</td>
<td>Artificial Neural Network</td>
<td>95.6%</td>
</tr>
<tr>
<td>Francisco Álvaro, Joan Andreu Sánchez[3]</td>
<td>Weighted nearest neighbor, Hidden Markov models (HMM), K-Nearest Neighbor</td>
<td>98.5%</td>
</tr>
<tr>
<td>Ahmad Montaser Awal, Harold Mouchère, Christian Viard-Gaudin[2]</td>
<td>multi-layer perceptions neural network (MLP)</td>
<td>87.5%</td>
</tr>
<tr>
<td>Our Approach</td>
<td>Neural Network</td>
<td>90%</td>
</tr>
</tbody>
</table>

Table 2 shows various methods used by different authors for symbol and mathematical expression recognition system. This comparison gives idea about recognition rate of the different systems.

The above table shows the different methods for mathematical expression recognitions and has recognition rate details respectively. So among these methods it is concluded that neural network has good recognition rate 90%. And it is more advantageous due to its Adaptive learning, Self-Organization, Real Time Operation, Fault Tolerance via Redundant Information Coding.

V. CONCLUSION

Recognition of mathematical expression consists main steps pre-processing, segmentation, feature extraction and Classification. In this technique to obtain maximum recognition rate is very difficult due to the ambiguities of mathematical expressions. This paper provides a neural network approach which gives 90% recognition rate.

VI. REFERENCES


[10] Kam Fai Chan, Dit Yan Yeung, Department of Computer Science, The Hong Kong University of Science


