Nepali Handwriting Recognition using Convolution Neural Network

1Ashutosh Ghimire, 2Adhista Chapagain, 3Utsav Bhattarai, 4Anku Jaiswal

1Software Engineer, Bent Ray Technologies (Pvt) Ltd, Chakupat, Lalitpur, Nepal
2Associate Technical Writer, Logpoint Nepal, Jwalakhel, Lalitpur, Nepal
3Software Engineer, Tech101 Pvt Ltd, Kamaladi, Kathmandu, Nepal
4Assistant Professor, Dept. of Computer and Electronics Engineering, Pulchowk Campus, IOE, Kupondol, Lalitpur, Nepal

Abstract - The recognition of texts from the scanned image can have various applications, which are based on optical character recognition. This paper was proposed and carefully experimented to analysis and recognize handwritten Nepali character using Convolution Neural Network. The preliminary experiment has been done with 92 thousand images of 46 different classes of 32 * 32 characters of Nepali Handwriting which went through different preprocessing stages like clipping and cropping, grayscale conversion and through different processes like feature extraction etc. The recognition has been experimented with the help of template matching technique. This proposal will employ Back Propagation algorithm along with Gradient descent algorithm will be used to update the weights, an artificial neural network training and testing. Thus, this experiment concluded that the convolution neural network model has more accuracy than the Feed Forward neural network in character recognition.

Keywords: Convolution, Neural Network, Artificial Intelligence, Back Propagation, Gradient Descent, Modified National Institute of Standards and Technology (MNIST), Dataset, Epoch.

I. INTRODUCTION

Devanagari, an alphabetic script, was developed to write Sanskrit but was later adapted to write many other languages such as Nepali, Marathi, Hindi, Konkani etc. Devanagari script is a logical composition its constituent symbols in two dimensions. It has eleven vowels and thirty-three simple consonants. A horizontal line is drawn on top of all characters which is then scanned into digital format using scanner. The most common neural network used in this system is multilayer perception with feed forward networks. It is still not clear that how combination strategy can fully utilize power of sub-classifiers and to deal with the tradeoff between the

Artificial Neural Network (ANNs) are computing systems vaguely inspired by the biological neural networks that follows a different way from traditional computing methods to solve problems. ANNs consists of three layers. The input layer receives an input data from external resources. The middle layer receives input from adjacent neurons. The output layer is the last layer that receives its input from the hidden layer. The output is sent out as the post processing result.

II. LITERATURE REVIEW

2.1 Devanagari Script Character Recognition using Genetic Algorithm

Character recognition is the automatic or digital translation of handwritten, typewritten or printed text images into machine editable text. The input image is scanned and proceeds further for noise removal. Then, the images are normalized. The normalized in the form of grayscale is converted to binary image by taking some threshold value and the obtained binary formed image is then extracted to provide a shape to them by thinning mechanism. Further, the pre-processed image goes through a segmentation which provides a character image with some information. Then, the feature extraction technique is used to remove that information from the image. Different problems occurred in the system could have been solved even if genetic algorithm is not used here.

2.2 Offline Hindi Handwriting Character Recognition

This system is mainly focused to recognize the characters produced by the person by writing with pen/pencil in a paper medium which is then scanned into digital format using scanner. The most common neural network used in this system is multilayer perception with feed forward networks. It is still not clear that how combination strategy can fully utilize power of sub-classifiers and to deal with the tradeoff between the
combinations and effectiveness. The errors in recognizing the printed characters are mainly due to incorrect character segmentation of touching or broken characters. Due to the lower and upper modifiers of Nepali text, any part of two consecutive lines may overlap and proper segmentation of the overlapped portions are needed in order to get higher accuracy.

2.3 Stroke Number and Order Free Handwriting Recognition for Nepali

This system was developed in the year 2006. This system uses structural properties of those alphanumeric characters, which have variable writing units. It uses a string of pen tip's positions and tangent angles of every consecutive point as feature vector sequence of a stroke. “Dynamic Time Warping” algorithm is used to align handwritten strokes with stored templates of strokes and find their similarity.

2.4 Google Translate

Google Translate, developed by Google, is a free multilingual machine translation service. It translates text from one language into another which is available in website interface, apps interface for Android and iOS platforms, and an API which helps developers to build software applications and browser extensions. Over 100 languages are supported at various levels and as of May 2017, over 500 million people served daily rather than translating languages directly. On several occasions, Google translate looks for patterns to help decide on the best translated to the target language. In millions of documents, Google translate looks for patterns to help decide on the best translation, during a translation. It was a lot easier and faster to feed the CSV content into the neural network as an input.

The dataset is converted to an easier-to-work CSV format. It consists of 92 thousand images of 46 different classes of characters of Devanagari script. The dataset separated into two categories: training dataset and testing dataset. Each image is of 32x32 dimensions, so there are in total 1025 columns in the CSV file: 1024 columns for the image pixels, and 1 column for the actual digit value. The first column contains the actual digit that is represented by the next 1024 pixels/columns.

The dataset was made available with the help of the students of grade 6 and 7 of Mount Everest Higher Secondary School (http://www.mes.edu.np/), Suryabinayak, Bhaktapur, Nepal of 2071 B.S. batch.

IV. METHODOLOGY

Image Acquisition is done either entering image file directly or through gesture input of user handwriting. After image acquisition, it is taken through different pre-processing steps which are:

- Image Clipping and Cropping
- Grayscale Conversion
- Segmentation
- Header line elimination

Clipping is performed by searching the pixel in horizontal and vertical manner and find the end points of the image. The imaginary lines from top, left, bottom and right edges of the image dragged over it crops out the image.

In a grayscale digital image, the value of each pixel is a single sample meaning which it carries only intensity information. In segmentation the header line is removed and the characters are separated.
The pixel values of the character image itself are used as the feature which is mapped into a matrix of size 32 X 32 using matrix mapping algorithm. Then this matrix values are linearized into an array.

Steps for Pre-processing:

1. Take input images.
2. Make grayscale and then make it Gaussian blur.
3. Canny edges help to reduce noise.
4. After that, the contours are found.
5. Bounding rectangle is created around the contours.
6. Remove noises outside the boundary.
7. Blur the image for removing tiny noises and contours.
8. Convert into 32 x 32 pixels.
9. Flatten the pixel values into 1 x 1024 array.
10. Outputs the result for classification.

IV. BUILDING CONVOLUTION NEURAL NETWORK

Resultant two-dimensional arrays are converted into a single long continuous linear vector. Each local feature map channel in the output of a CNN layer is a “flattened” 2D array created by adding the results of multiple 2D kernels (one for each channel in the input layer).

SoftMax function calculates the probabilities of each target class over all possible target classes for determining the target class for the given inputs. The output probabilities range i.e. 0 to 1 whose sum will be equal to one.
Figure 8: SoftMax function

The formula computes the exponential \( e \)-power of the given input value and the sum of exponential values of all the values in the inputs.

\[
\sigma(z)_j = \frac{e^{z_j}}{\sum_{i=1}^{K} e^{z_i}}
\]

Then the ratio of the exponential of the input value and the sum of exponential values is the output of the SoftMax function.

A rectified linear unit has output 0 if the input is less than 0, and if the input is greater than 0, the output is equal to the input. ReLUs' machinery is more like real neuron in body.

\[
R(z) = \max(0, z)
\]

Figure 9: Rectified Linear Units

Tanh function is nonlinear in nature, so layers can be stacked which is bound to range (-1, 1).

V. RESULT AND ANALYSIS

5.1 Calculations

Intuitively, it makes sense that if it is planned to explore the gradient decent for much longer (more epochs), shorter steps (learning rate) can be afforded to take. It does seem that 5 epochs is probably the sweet spot for this neural network against MNIST learning rate.

<table>
<thead>
<tr>
<th>Layer(type)</th>
<th>Output shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv2d_1(Conv2D)</td>
<td>(None, 100, 28, 28)</td>
<td>2600</td>
</tr>
<tr>
<td>max_pooling2d_1(MaxPooling2)</td>
<td>(None, 100, 14, 14)</td>
<td>0</td>
</tr>
<tr>
<td>activation_1(Activation)</td>
<td>(None, 100, 14, 14)</td>
<td>0</td>
</tr>
<tr>
<td>conv2d_2(Conv2D)</td>
<td>(None, 250, 10, 10)</td>
<td>625250</td>
</tr>
<tr>
<td>max_pooling2d_2(MaxPooling2)</td>
<td>(None, 250, 5, 5)</td>
<td>0</td>
</tr>
<tr>
<td>activation_2(Activation)</td>
<td>(None, 250, 5, 5)</td>
<td>0</td>
</tr>
<tr>
<td>flatten_1(Flatten)</td>
<td>(None, 6250)</td>
<td>0</td>
</tr>
<tr>
<td>dense_1(Dense)</td>
<td>(None, 1000)</td>
<td>6251000</td>
</tr>
<tr>
<td>dropout_1(Dropout)</td>
<td>(None, 1000)</td>
<td>0</td>
</tr>
<tr>
<td>activation_3(Activation)</td>
<td>(None, 1000)</td>
<td>0</td>
</tr>
<tr>
<td>dense_2(Dense)</td>
<td>(None, 46)</td>
<td>46046</td>
</tr>
<tr>
<td>activation_4(Activation)</td>
<td>(None, 46)</td>
<td>0</td>
</tr>
<tr>
<td>Total Params:</td>
<td></td>
<td>6,924,896</td>
</tr>
<tr>
<td>Trainable Params:</td>
<td></td>
<td>6,924,896</td>
</tr>
<tr>
<td>Non-trainable Params:</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
5.2 Result

The resulting performance with 2 epochs is 0.9579, a small improvement over just 1 epoch. Similarly, it was done with tweaking learning rate and experimented with few different epochs and plotted a graph to visualize the effect this has. Intuition suggests the more training gives the better the performance. But too much training is not good because the network overfits to the training data and then forms badly against new data that it hasn’t seen before.

The following graph shows performance against epochs with learning rate 0.1 and learning rate 0.2.

![Performance of Convolution Model](image)

Figure 10: Performance and Epoch. MINIST dataset with 3-layer Neural Network

It was seen that calming down the learning rate did indeed produce better performance with more epochs. That peaks of 0.9689 represents an appropriate error of 3%, which is comparable to the benchmarks in Yann LeCun’s website.

VI. CONCLUSIONS

Hence, this convolution neural network based handwriting recognition helps to recognize handwritten or printed hardcopy characters or gesture inputs as per the input given by the user. This utilize of Back Propagation algorithm along with Gradient descent algorithm concluded that the convolution neural network model has more accuracy than the Feed Forward neural network in character recognition.

REFERENCES


Citation of this Article:


********