International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Handwritten Character Recognition using Convolutional Neural Networks (CNN)

Verdha Chaudhary, Mahak Bansal and Gaurav Raj
Jaypee University Anoopshshr, India
verdha98@gmail.com, bansalmahak1234@gmail.com, gaurav.raj@mail.jaypeeu.ac.in

Abstract—Handwritten character recognition is a translational problem of human writings into machine-editable text format. In this paper, Convolutional Neural Networks (CNN) is presented for handwritten character recognition. Handwritten character was transformed into graphs based on its underlying skeleton structure. Edges of the extracted graph were categorized into shape types and vertices were extracted from each of the edges and their layer wise evaluation using deep learning. Matching procedure of the graph was performed in Convolutional Neural Networks (CNN) approach. Performance evaluation of the proposed method was conducted using validated kaggle dataset which exclude ambiguous and unidentified writing samples. The use of neural network can improve the quality of recognition while achieving good performance and encouraging.

Keywords—Convolutional neural network, Handwritten character recognition.

I. Introduction

The purpose of this project is to take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern or structure of characters, and matches the character or the output with the desired input. Pattern recognition is perhaps the most common use of neural networks. The neural network is presented with a input which contains the pattern information, this could be an image, and hand written data. The neural network then attempts to determine if the input data matches a pattern that the neural network has mentioned. A neural network trained for classification is designed to take input samples and classify them into groups. These input groups may be fuzzy, without clearly defined boundaries. This project concerns detecting free handwritten characters.

II. NEURAL NETWORKS

Neural network is a system inspired by human brain function, consists of neurons connected in parallel with the ability to learn. A basic design of neural network has input layer, hidden layer, and output layer. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract pattern and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been to analyze. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining.

III. NETWORK LAYERS

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of "**output**" units.(Figure 1)

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and the hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

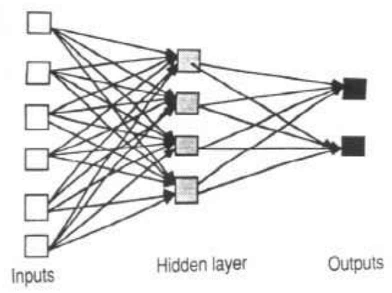


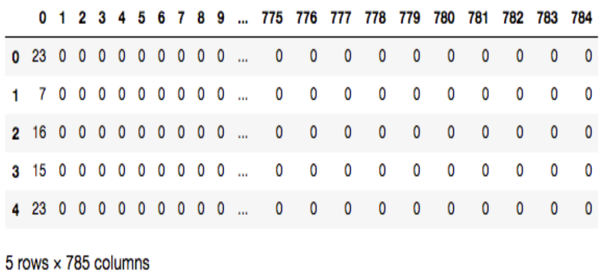
Figure 1

IV. CONVOLUTIONAL NEURAL NETWORKS

A special type Neural Networks that works in the same way of a regular neural network except that it has a convolution layer at the beginning. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and softmax layers.

V. DATASET

In handwritten character recognition we have used the EMNIST dataset. The EMNIST dataset is a set of handwritten character, which is derived from the NIST Special Dataset 19 and are converted into a 28x28 pixel image format and dataset structure, which directly matches the MNIST dataset. As we known the MNIST dataset has become a standard benchmark for learning, classification and computer vision systems. Whenever we get a new image, it is compared with the database that is already stored in the system.



	0	1	2	3	4	5	6	7	8	9	...	775	776	777	778	779	780	781	782	783	784
0	23	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	16	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	15	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	23	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

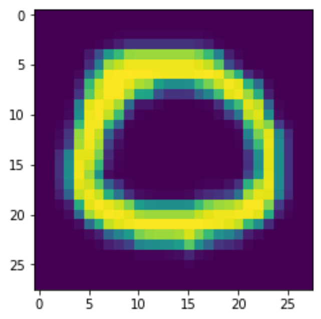
5 rows × 785 columns

A. Letters Dataset

The EMNIST dataset of letters consists a balanced set of the uppercase and lowercase letters into a single 26- classes.

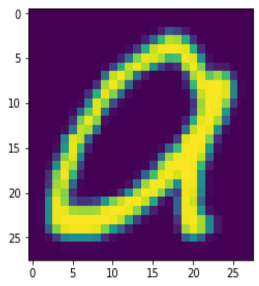
- Train:88,800
- Test:14,800
- Total:103,600

```
Out[34]: <matplotlib.image.AxesImage at 0x109ab6d68>
```



Example of Training Data Set

```
Out[36]: <matplotlib.image.AxesImage at 0xb39e13ba8>
```



Example of Test Data Set

VI. EXPERIMENTAL DATASET AND OBSERVATION

For recognizing the handwritten characters we have used Convolutional Neural Network. There are several techniques and processes that are used to get a successful outcome. The techniques or the processes that are used as they are as follows:

A. Pre-processing

The pre-processing is s series of operations which is performed on the scanned input image. It essentially enhances the image rendering it. The dilation of an image and filling the holes present in it are the operations executed in the last two stages to generating the pre-processed image.

First, we need to add a little bit of variance to the data since the images from the dataset are very organized and contain little to no noise. We're going to do a random combination of the following to the images:

- Crop parts of the image
- Flip image horizontally
- Adjust hue, contrast and saturation

B. Splitting Our Dataset

It takes a long time to calculate the gradient of the model using the entirety of a large dataset . We therefore will use a small batch of images during each iteration of the optimizer. The batch size is normally 32 or 64—we'll use 256 since we have fairly a large number of images . The dataset is then divided into training set containing 88,800 images, and test set containing 14,800 images.

C. Building A Convolutional Network

Now that we're done pre-processing and splitting our dataset we can start implementing our neural network. We're going to have 2 convolution layers with 2 x 2 max-pooling. After that, we add 1 fully connected layers. Since the input of fully connected layers should be two dimensional, and the output of convolution layer is four dimensional, we need a flattening layer between them. At the very end of the fully connected layers is a softmax layer.

1. Pooling Layer

Pooling(Subsampling) serves to change the input feature into a representation of statistical results of the features around it. Most of the subsampling on CNN uses Max pooling. Its function is to reduce the spatial size of the representation or reduce the amount of parameters and computation in the network and to also control over fitting. This layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.

Max-pooling: A technique used to reduce the dimensions of an image by taking the maximum pixel value of a grid. This also helps reduce overfitting and makes the model more generic. The example below show how 2 x 2 max pooling works.

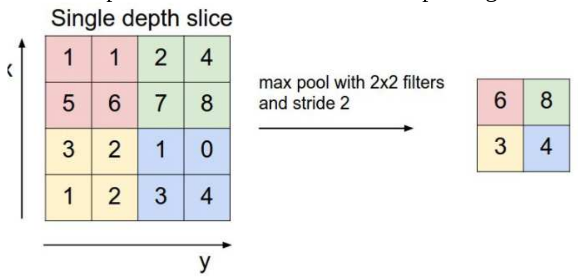


Figure 2

D. **Training The Network**

For training the model we have used different kinds of layers that are as follows:

1. **Dropout Layer**

   The original paper on Dropout provides experimental results on a suite of standard machine learning problems. As a result they provide a number of useful heuristics to consider when using dropout in practice.

- Generally, use a small dropout value of 20%-50% of neurons with 20% providing a good starting point. A probability too low has minimal effect and a value too high results in under-learning by the network.
- Use a larger network. You are likely to get better performance when dropout is used on a larger network, giving the model more of an opportunity to learn independent representations.
- Use dropout on incoming (visible) as well as hidden units. Application of dropout at each layer of the network has shown good results.
- Use a large learning rate with decay and a large momentum. Increase your learning rate by a factor of 10 to 100 and use a high momentum value of 0.9 or 0.99.
- Constrain the size of network weights. A large learning rate can result in very large network weights. Imposing a constraint on the size of network weights such as max-norm regularization with a size of 4 or 5 has been shown to improve results.

2. Dense Layer

A dense layer is just a regular layer of neurons in a neural network. Each neuron receives input from all the neurons in the previous layer, thus densely connected. The layer has a weight matrix **W,** a bias vector **b,** and the activations of previous layer **a.**
A dense layer represents a matrix vector multiplication. (assuming your batch size is 1) The values in the matrix are the trainable parameters which get updated during back propagation.
$uT.W,W\in Rn\times m$uT.W, W∈Rn×m
So you get a m dimensional vector as output. A dense layer thus is used to change the dimensions of your vector. Mathematically speaking, it applies a rotation, scaling, translation transform to your vector.

3. Flatten Layer

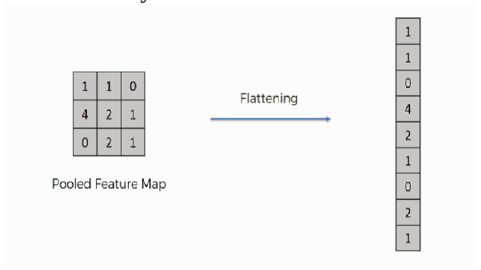


Figure 3

The reason we do this is that we're going to need to insert this data into an artificial neural network later on.
.

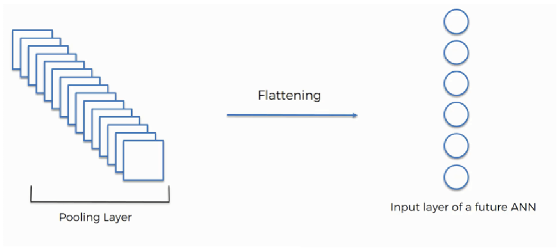


Figure 4

As you see in the image above, we have multiple pooled feature maps from the previous step. What happens after the flattening step is that you end up with a long vector of input data that you then pass through the artificial neural network to have it processed further. The flattening step is needed so that you can make use of fully connected layers after some convolutional layers. Fully connected layers don't have a local limitation like convolutional layers (which only observe some local part of an image by using convolutional filters). This means you can combine all the found local features of the previous convolutional layers. Each 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).

E. Activation function — ReLU

The sigmoid is not the only kind of smooth activation function used for neural networks. Recently, a very simple function called **rectifier linear unit (ReLU)** became very popular because it generates very good experimental results and the nonlinear function is represented in the following graph. As we can see in the following graph, the function is zero for negative values, and it grows linearly for positive values. A relu function is defined as:

$$f(x) = \max(0, x),$$

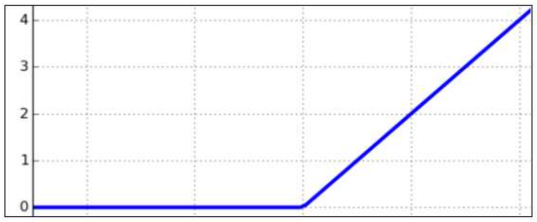
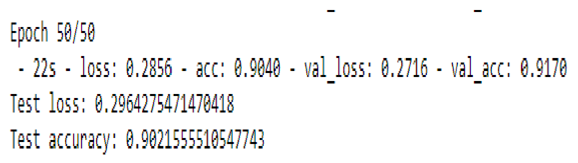


Figure 5

VII. RESULT

We then made the computer interpret 145,600 unknown images and got an accuracy of 90.2% (9022 / 10000) on the test data in 50 iterations. Also, an analysis of experimental result has been performed and shown in fig.



```
Epoch 50/50
 - 22s - loss: 0.2856 - acc: 0.9040 - val_loss: 0.2716 - val_acc: 0.9170
Test loss: 0.2964275471470418
Test accuracy: 0.9021555510547743
```

VIII. CONCLUSION

The convolutional neural network based handwritten character recognition has been introduced in this paper for classifying and recognizing the 26 English alphabets. The Experimental results shown that the machine has successfully recognized the Handwritten English characters with the highest recognition accuracy of 90.2%. The handwritten recognition system described in this paper will find applications in handwritten names recognition, document reading, conversion of any handwritten document into structural text form and postal address recognition.

Acknowledgment

Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. Dataset retrieved from https://www.nist.gov/itl/iad/image-group/emnist-dataset. *Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik*

The MARCS Institute for Brain, Behaviour and Development
Western Sydney University
Penrith, Australia 2751

References

[1] Yuauf Perwej, Ashish Chaturvedi, "Neural networks for Handwritten English Alphabet Recognition, International Journal of Computer Application(0975-8887) Volume-20-No.7,April 2011".

[2] Savitha Attigeri," Neural networks based Handwritten Character Recognition System, International Journal of Engineering and Computer Science, p.No.:23761-23768"

[3] Jason Brownlee," DropOut Regularization in Deep Learning Models With Keras".

[4] Chirag l Patel, Ripal Patel, Palak Patel, "Handwritten character recognition using neural network",International Journal of Scientific and Research Volume 2,Issue may-2011".

[5] Anita Pal & Dayashankar Singh, "Handwritten English Character Recognition using Neural Network, International Journal of Computer Science & Communication, Volume-1, No.2 , july-December 2010, pp.141-144."

[6] Abdellatif Abdelfattah, "Image Classification using Deep Neural Networks- A beginner friendly approach using Tenserflow".

[7] Sujit Pal, Antonio Gulli, "Deep learning with Keras-learning algorithms in Keras".

[8] Batuhan Balci, Dan Saadati, Dan Shiferaw, "Handwritten Text Recognition using Deep learning".

[9] J. Pradeep, E. Srivasan and S. Himavathi, "Diagonal Based feature extraction for Handwritten alphabets recognition system using neural network",International Journal of Computer Science & Information Technology(IJCSIT), Vol 3, No.1 Feb,2011".