International Journal of Information Sciences and Application (IJISA). ISSN 0974-2255, Vol.11, No.1, 2019, (Special Issue)

Handwritten Digits Recognition Using Machine learning

Shikhar Tandon
IMS ENGINEERING COLLEGE, GHAZIABAD, UTTAR PRADESH

Shadab Akhter
IMS ENGINEERING COLLEGE, GHAZIABAD, UTTAR PRADESH

Vaibhav Pratap Singh

IMS ENGINEERING COLLEGE, GHAZIABAD, UTTAR PRADESH

Mr. Nizam Uddin Khan

Associate Professor, IMS ENGIMEERING COLLEGE, GHAZIABAD, UP

ABSTRACT

The problem of handwritten digit recognition has long been an open problem in the field of pattern classification. A ton of studies have shown that Neural networks, machine learning have great and efficient performance. In data classification Deep learning and Neural Network algorithms are a branch of Machine learning that can automatically identify patterns in the data, and then use the uncovered patterns to predict future data, or to perform other alternative kinds of decision making under unreliability Deep Learning algorithms are used to model high level abstractions in data. Digit Recognition is a combination of Deep Learning and Neural Network algorithms, which uses TensorFlow tool as an interface to develop a model.

This paper describes the recognition of handwritten scanned digits where the input is given by the user and displays the output as digital numbers referring as the input provided accordingly by using Machine Learning methods with the help of TensorFlow, MNIST Database, python thus the image may be sensed by the system as the user provides bare handed input to it and then the system shows the respective recognition accordingly.

Keywords: Handwritten Character Recognition, TensorFlow, MNIST Database, python, Machine Learning, node, images.

1. INTRODUCTION

Machine learning and deep learning plays an important role in computer sciences its paraphernalia's and artificial intelligence. The use of machine learning, deep learning and related principles have lowered the human efforts on industry. Handwritten digit recognition has gained a good amount of popularity from the very beginning of machine learning and deep learning to an expert who has been practicing for years. Developing such a machine needs proper understanding of classification of digits and the difference between the minor and major points to properly differentiate between different digits which can be only possible with proper training and testing

Handwritten recognition (HWR) is the ability of a computer to receive and understand intelligible handwritten input from sources such as paper documents, user input

touch-screens and other devices. The image of the written text may be sensed from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition or by user input.

Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface, a generally easier task as there are more clues available This paper presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset using TensorFlow framework(library) and python as language and its libraries as user enters the respective digit the machine would recognize and show the results with accuracy percentage

2. METHODOLOGY

2.1 Tensorflow

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions Tensorflow is an open source library created by the Google Brain Trust for heavy computational work, geared towards machine learning and deep learning tasks. Tensor Flow is built on c, c++ making it very fast while it is available for use via Python, C++, Haskell, Java and Go API depending upon the type of work.

It created data graph flows for each model, where a graph consists of two units – a **tensor** and a **node**.

- **Tensor:** A tensor is any dimensional array which is not single dimensional.
- **Node:** A node is a mathematical computation that is being worked at the moment to give the desired result.

A data graph flow essentially maps the flow of information via the interchange between these two components. As the graph is completed, the model is executed for the output.

2.2 MNIST Dataset

MNIST (Modified National Institute of Standards and Technology) consists of samples of handwritten digits, they contain total 70,000 images us of which 60,000 are used in training set and 10,000 are used in testing set, both with appropriately labelled images 10 digits (0 to 9). Handwritten digits are images referring the form 28*28 gray scale intensities of images representing an image with the first column to be labelled as (0 to 9) for every image. Similarly, it has opted for the case of the testing set as 10,000 images with a label of 0 to 9 thus. MNIST is a computer science and vision database consisting of handwritten digits, with labels identifying the digits appropriately,

every MNIST data point has two parts: an image of a handwritten digit and its corresponding label. To start with Tensorflow, we will be using the MNIST database to create an image identifying model based on simple feedforward neural network with no hidden layers respectively.

The following figure represents the example sample of the MNIST dataset which is to be used using which the system will be trained and then tested for respected output.

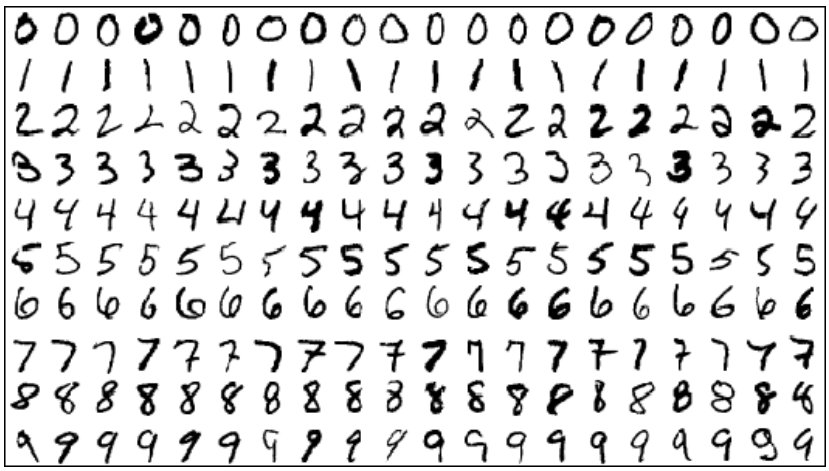

Figure 1: MNIST dataset

3. IMPLEMENTING THE HANDWRITTEN DIGIT'S RECOGNITION MODEL

We will be building simple feedforward neural network using softmax to predict the number in each image. We begin by calling in a Python environment.

3.1 Download the MNIS

- from tensorflow.examples.tutorials.mnist import input_data

  mnist = input_data.read_data_sets("model_data/", one_hot=True)

3.2 Import Tensorflow to your environment

- import tensorflow as tf

3.3 Initializing parameters for the model

- *batch =100*

  learning_rate=0.01

  training_epochs=10

In machine learning, an epoch is a full iteration over samples. Here, we are restricting the model to 10 complete epochs or cycles of the algorithm running through the dataset.

The batch variable determines the amount of data being fed to the algorithm at any given time, in this case, 100 images.

The learning rate controls the size of the parameters and rates, thereby affecting the rate at which the model "learns".

3.4 Creating Placeholders

- x = tf.placeholder(tf.float32, shape=[None, 784])

  y = tf.placeholder(tf.float32, shape=[None, 10])

The method tf.placeholder allows us to create variables that act as nodes holding the data. Here, x is a 2-dimensionall array holding the MNIST images, with none implying the batch size (which can be of any size) and 784 being a single 28×28 image. y_ is the target output class that consists of a 2-dimensional array of 10 classes (denoting the numbers 0-9) that identify what digit is stored in each image.

3.5 Creating Variables

- W = tf.Variable(tf.zeros([784, 10]))

  b = tf.Variable(tf.zeros([10]))

Here, W is the weight and b is the bias of the model. They are initialized with tf**.**Variable as they are components of the computational graph that need to change values with the input of each different neuron

3.6 Initializing the model

- y = tf.nn.softmax(tf.matmul(x,W) + b)

3.7 Defining Cost Function

- cross_entropy = tf.reduce_mean (-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))

This is the cost function of the model – a cost function is a difference between the predicted value and the actual value that we are trying to minimize to improve the accuracy of the model.

3.8 Determining the accuracy of parameters

- correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))

  accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

3.9 Implementing Gradient Descent Algorithm

- train_op = tf.train.GradientDescentOptimizer(learning_rate).minimize(cross_entropy)

Tensorflow comes pre-loaded with a lot of algorithms, one of them being Gradient Descent. The gradient descent algorithm starts with an initial value and keeps updating the value till the cost function reaches the global minimum i.e. the highest level of accuracy.

3.10 Initializing the session

- with tf.Session() as sess:

  sess.run(tf.initialize_all_variables())

3.11 Creating batches of data for epochs

- for epoch in range(training_epochs):

  batch_count = int(mnist.train.num_examples/batch)

  for i in range(batch_count):

  batch_x, batch_y = mnist.train.next_batch(batch)

3.12 Executing the model

- sess.run([train_op], feed_dict={x: batch_x, y_: batch_y})

3.13 Print accuracy of the model

- if epoch % 2 == 0:

  print "Epoch: ", epoch

  print "Accuracy: ", accuracy.eval(feed_dict={x: mnist.test.images, y_: mnist.test.labels})

  print "Model Execution Complete"

4. CONCLUSION

This paper has practiced machine learning techniques including use of Tensorflow to obtain the appropriate digit recognition This study built handwritten recognizers evaluated their performances on MNIST (Mixed National Institute of Standards and Technology) dataset and then improved the training speed and the recognition performance. The error rate thus obtained is of 1.25 and training accuracy is 98% and test accuracy 97% demonstrating significant and promising performance.

Thus by practicing this we have achieved success in properly identifying the digits drawn at different angles and properly displaying the correct digit at a single turn.

Hence the system would be able to recognize the introduced digit according to the formations made and according to the values in the dataset.

5. REFERENCES

[1]. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J. et al. (2016). TensorFlow: A System for Large Scale Machine Learning. In 12thUSENIX Symposium on Operating Systems Design and Implementation (OSDI'16) (Vol. 16, pp. 265-283).

[2]. Handwritten digit recognition using state-of-the-art techniques, Cheng-Lin Liu Central Res. Lab., Hitachi Ltd., Tokyo, Japan K. Nakashima Central Res. Lab., Hitachi Ltd., Tokyo, Japan H. Sako Central Res. Lab., Hitachi Ltd., Tokyo, Japan H. Fujisawa Res. Lab., Hitachi Ltd., Tokyo, Japan, IEEE published.

[3]. Offline Handwritten Digits Recognition Using Machine learning, Shengfeng Chen, Rabia Alm amlook, Yuwen Gu, Proceedings on the International Conference on Industrial Engineering and Operations management Washington DC, USA, Sep 27-29, 2018

[4]. Digit Recognition using TensorFlow Tool, K. V. N. Rajesh, K.V.N. Ramesh, M. Hymavathi , K. Syam Sundar Reddy, K. S. S. (2017), i-manager's Journal on Pattern Recognition, 4(3), 27-31. https://doi.org/10.26634/jpr.4.3.13887