Hand-written Digit Recognition using MNIST dataset

Noshin Nawal, Nadia Farhin Chowdhury, Anuradha choudhury, Moshiur Rahman, Ashiqur Rahman

Department of Computer Science & Engineering
BRAC University
66 Mohakhali, Dhaka, Bangladesh
{noshin.nawal199599, nadiafarhinchowdhury,piyaachy,moshiur.bracu,,ashik.bracu,}@gmail.com

ABSTRACT

In this paper, we propose a method for recognition of handwritten digit utilizing MNIST dataset. Handwritten digit recognition is an important benchmark task in pattern recognition and computer vision. Here, we will be using python's scikit-learn library which within few steps helps to train machine learning classifier (logistic regression and decision tree etc.) based on images. After training the model, it will be able to predict an image label (0-9) given an image and give the accuracy (for logistic regression-91.3% and decision tree-83%).

Keywords

Handwritten digits, MNIST dataset, Logical regression, Decision Tree, Python.

1. INTRODUCTION

Handwritten digit recognition is one of the most practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry and many others. The purpose of this step is to highlight important information for the recognition model. [1] This paper has achieved results by using different algorithms such as Decision Tree, Logical regression. A complete handwriting recognition system also handles formatting, performs correct segmentation into characters and finds the most plausible digit. Thus to learn all the basic techniques and use it in the future, we have chosen MNIST dataset which consists of 70000 samples.[2]

The MNIST database was constructed from NIST's

Special Database 3 and Special Database 1 which contain binary images of handwritten digits. NIST originally designated SD-3 as their training set and SD-1 as their test set. [3] However, SD-3 is much cleaner and easier to recognize than SD-1. The reason for this can be found on the fact that SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. Drawing sensible conclusions from learning experiments requires that the result be independent of the choice of training set and test among the complete set of samples.[3] Therefore, it was necessary to build a new database by mixing NIST's datasets. The MNIST training set is composed of 30,000 patterns from SD-3 and 30,000 patterns from SD-1. The 60,000 pattern training set contained examples from approximately 250 writers.[4] Thus after the collection of dataset we have performed the algorithms. There are lots of algorithms with which we can train and test them for example SVM, KNN, CNN etc, but we have chosen Logistic Regression and Decision Tree because these two are majorly taught in our course.

2. METHODS

2.1 Dataset

For the very first step we needed to do some work to fix the dataset. All images of dataset will be of size 28x28 and we will use transfer learning to train on the smaller number of digits classes.

TABLE I: MNIST dataset files and their sizes

Files	Size in Bytes
Training set images	9912422
Training set labels	28881
Test set images	1648877
Test set labels	4542

We have split each dataset into a train and test portion. T he t est-size=1/7.0 makes the training set size 60,000 images and the test set size of 10,000.

Fig :1

2.2 Logistic Regression

First algorithm we used is logistic regression. At first we import the model and create instance of the model. One thing is very important is the parameter tuning. Here, we have used the solver= ”lbfgs” to speed up the fitting of our model.[5] It may not have mattered much for small datasets but make huge difference in case of bigger and complex datasets.[6] Next step is to train the model on data, storing the information learned from data. Here, our model learns relationship between x(digits) and y(labels). After that the labels of new data(images) are predicted one by one which leads to prediction of entire test data.

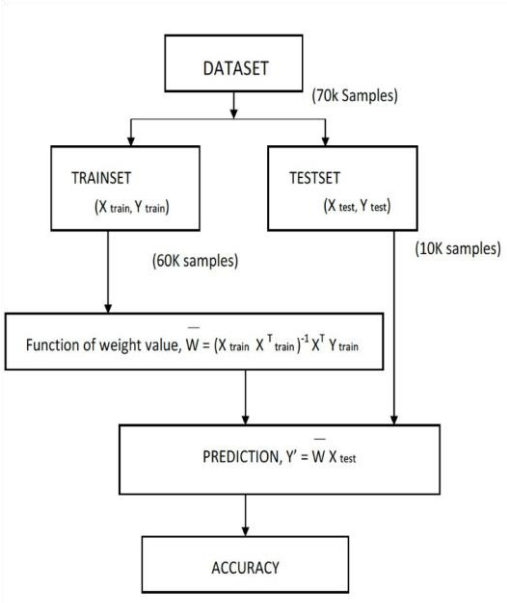

Fig 2: Diagram of logistic regression

2.3 Decision Tree

Decision Tree is non-parametric supervised learning model which is used for classification and predicts the value of a target variable by learning simple decision rules inferred from the data features.[5] Here in our project, decision tree classifier is used in order to have a smart tree model that is classifying the class of unknown instance by discovering list of rules from the dataset of handwritten digits. By selecting the attribute for the root node it creates the branch for each possible attribute.[7] After that, it selects the best spilt of the dataset by measuring the impurity for each child nodes and splitting instances into subsets for each branch extending from the node. Finally, it makes iteration for each node using only records that reach the node and when all instances have the class, then it will stop.

3 .RESULT & DISCUSSION

There are many other ways of measuring model performance, we are going to keep this simple and use accuracy as our metric. To do this, we are going to see how the model performs on the new data (test set)

Accuracy is defined as:
(Fraction of correct predictions): correct predictions / total number of data points.

From logistic regression we got the accuracy rate as 91.3%.

```
score = logisticRegr.score(test_img, test_lbl)
print(score)
```

```
0.9131
```

Fig 3: accuracy score of logistic regression

The result of accuracy can be visualized in the following confusion matrix where x exist represents the predicted label and x represents the actual label. Our model has predicted correct diagonally in the figure.

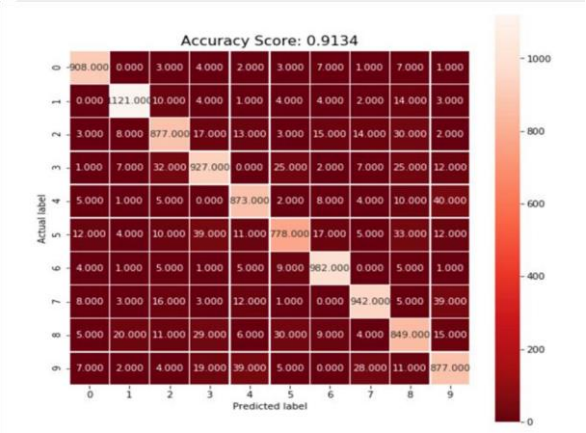

Fig: 4

We have also found out the accuracy results for 100 samples, 1000 samples and 6000 thousand samples which is shown below through confusion matrices where the accuracy rate is 71.3%, 82.8% and then 91.3% respectively.

Fig 5: result accuracy

We have also found out the misclassified images that couldn't be predicted by our model.

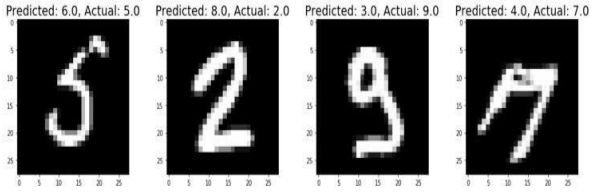

Fig: 6

For decision tree classifier we got the accuracy score of 83.48%

4.Conclusion

This paper presents an approach to predict that digits are differently written in hands. We have successfully applied the algorithms to find out whether this model will able to identify the written number or not. We have used Logistic Regression and Decision Tree Classifier for the prediction. We tested in multiple scenarios and added some new features to classify easily which is fairly accurate and can be implemented. Our purpose is to create a model that can be used in the future to predict the best possible handwritten character as fast and effectively as possible.

5. Reference

1. Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. Signal Processing Magazine, IEEE, 29(6):141–142, Nov 2012.

2 .h ttps://www.kaggle.com/c/digit-recognizer/data 3.h ttps://machinelearningmastery.com/machine-learnin g-in-python-step-by-step/

4. h ttps://scikit-learn.org/stable

5. h ttp://yann.lecun.com/exdb/mnist/

6.(IJACSA) International Journal of Advanced Computer Science and Applications.

7.https://www.researchgate.net/publication/22886161 0 _Handwritten_digit_classification