Finding Optimized Machine Learning Model For Recognizing English Handwritten Digit

Nowfel Mashnoor
Roll: 1503069

Amir Faruk
Roll: 1503075

Abstract

This paper is about the comparison between different Machine Learning models(classifiers) trained and tested on MNIST dataset. For declaring a model as best, we only considered low error score. A standard machine learning library written in Python Programming Language is used during this research.

1 Introduction

Handwritten Digit Recognition has been very successful in recent years. A lot of research and studies has been done in recent years on it like Devnagari Handwritten Character Recognition[1]. Handwritten digit recognition technique is used in various fields like PDA, bank cheque, handwritten fields in form etc.[2] Using machine learning technique, which which can be briefly defined as enabling computers make successful predictions using past experiences, [3] handwritten digit recognition system is greatly improved. Handwritten Digit recognition is a supervised learning algorithm problem. There are many classifier in supervised learning like Neural Network, Decision Tree, Bayesian Network, Support Vector Machine(SVM), Random Forest etc[4]. A comparison study has been already done where *Base Linear Classifier, Baseline Nearest Neighbor Classifier, Large Fully Connected Multi-Layer Neural Network, Tangent Distance Clasifier(TDC), LeNet 4 With KNN, Optimal Margin Classifier* are compared among.[5] In our research we are going to compare among algo1, algo2, algo3. We will chose the best classifier among them based on their accuracy on testing set.

2 MNIST Dataset

For training and testing our classifiers, we used MNIST (Modified National Institute of Standards and Technology) Dataset. This dataset contains *70000* images. Among 70000 images, 60000 images are for training and 10000 images are for testing[6]. It is basically a subset of NIST dataset. the black and white images from NIST were normalized to fit into a 28x28 pixel and converter to grayscale levels.[7]

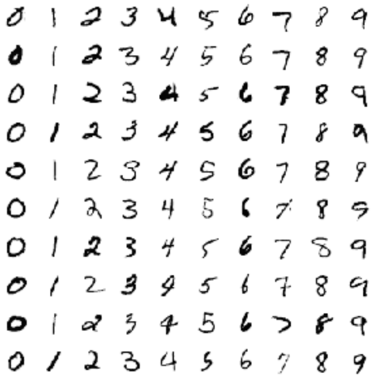

Figure 1: MNIST Dataset Sample Images

For each and every classifier we are using in our study, we will use use all the 60000 images for training and the rest 10000 images for evaluating our models.

3 Scikit Learn Machine Learning Library

Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems[8]. Most of the algorithms are written in Python programming language where some of the algorithms are written in Cython for achieving optimal performance. Also for efficient operation on large arrays and multidimensional matrices, we used NumPy library.[9] For graphing and other visualization purpose, we used Matplotlib, which is other Python Library. [10] For training and testing our classifiers, we will use the Scikit Learn Library. This library also provides tools for model evaluation.

4 The Classifiers

We have trained and tested three classifiers on MNIST Dataset. We are going to give a quick and brief description of all of them.

4.1 Random Forest

Random Forest is a powerful Machine Learning model. It can do both regression and classification.[11] It is basically a collection of random decision trees. Each decision tree predict a class and after all the prediction, random forest outputs that class which got the most vote. The main benefit of Random Forest over decision tree is that it removes the problem of over fitting of decision tree.[12] Random forest uses Bootstrap aggregating or Tree Bagging algorithm. It reduces variance and helps to prevent over fitting. In decision tree learning, for splitting the variable at each step, different metrics can be used like Gini Impurity, Information Gain. We used Gini Index or Gini Impurity in our work. For calculating Gini Index, we have to use the following formula:

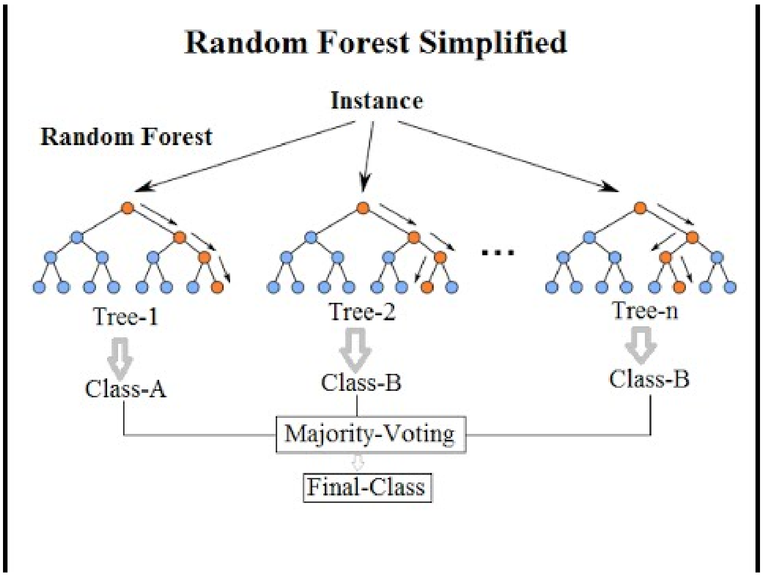

Figure 2: Random Forest

$$I_G(P) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k$$

Where,
J = No. of Classes
i ϵ 1, 2, 3, ..., J

p_i = Probability of an item with label i being chose
p_k = Probability of a mistake categorizing that item [13]

References

[1] U. Pal, T. Wakabayashi, and F. Kimura, "Comparative study of devnagari handwritten character recognition using different feature and classifiers," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pp. 1111–1115, IEEE, 2009.

[2] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[3] Y. Baştanlar and M. Özuysal, "Introduction to machine learning," in *miRNomics: MicroRNA Biology and Computational Analysis*, pp. 105–128, Springer, 2014.

[4] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[5] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, p. 276, 1995.

[6] E. Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on mnist database," *Image and Vision Computing*, vol. 22, no. 12, pp. 971–981, 2004.

[7] Y. LeCun, "Courant institute, nyu corinna cortes, google labs, new york the mnist database of handwritten digits."

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[9] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[10] G. Hackeling, *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2014.

[11] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[12] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.

[13] Wikipedia, "Decision tree learning," 2018. [Online; accessed 05-June-2018].