Critique of Various Algorithms for Handwritten Digit Recognition using Azure ML Studio

By Goutham Cheedella

Abstract- Handwritten Digit Recognition is probably one of the most exciting works in the field of science and technology as it is a hard task for the machines to recognize the digits which are written by different people. The handwritten digits may not be perfect and also consist of different flavors. And there is a necessity for handwritten digit recognition in many real-time purposes. The widely used MNIST dataset consists of almost 60000 handwritten digits. And to classify these kinds of images, many machine learning algorithms are used. This paper presents an in-depth analysis of accuracies and performances of Support Vector Machines (SVM), Neural Networks (NN), Decision Tree (DT) algorithms using Microsoft Azure ML Studio.

Keywords: handwritten recognition, digit recognition, azure machine learning studio, neural networks, support vector machines, decision tree, classification.

GJCST-D Classification: I.2.2

CRITIQUEOFVARIOUSALGORITHMSFORHANDWRITTENDIGITRECOGNITIONUSINGAZUREMLSTUDIO

Strictly as per the compliance and regulations of:

Critique of Various Algorithms for Handwritten Digit Recognition using Azure ML Studio

Goutham Cheedella

Abstract- Handwritten Digit Recognition is probably one of the most exciting works in the field of science and technology as it is a hard task for the machines to recognize the digits which are written by different people. The handwritten digits may not be perfect and also consist of different flavors. And there is a necessity for handwritten digit recognition in many real-time purposes. The widely used MNIST dataset consists of almost 60000 handwritten digits. And to classify these kinds of images, many machine learning algorithms are used. This paper presents an in-depth analysis of accuracies and performances of Support Vector Machines (SVM), Neural Networks (NN), Decision Tree (DT) algorithms using Microsoft Azure ML Studio.

Keywords: handwritten recognition, digit recognition, azure machine learning studio, neural networks, support vector machines, decision tree, classification.

I. Introduction

Handwritten Digit Recognition is probably one of the most exciting works in the field of science and technology as it is a hard task for the machines to recognize the digits which are written by different people. The handwritten digits may not be perfect and also consist of different flavors. And there is a necessity for handwritten digit recognition in many real-time purposes. The widely used MNIST dataset consists of almost 60000 handwritten digits. And to classify these kinds of images, many machine learning algorithms are used. This paper presents an in-depth analysis of accuracies and performances of Support Vector Machines (SVM), Neural Networks (NN), Decision Tree (DT) algorithms using Microsoft Azure ML Studio.

For many years, we are using handwritten digit recognition in several ways, and even though there is much advancement made in the system, there are minute errors in recognizing the digits. Still, we haven't achieved 100% accuracy, and research is going on. Seldom 1% or 2% errors may also lead to inapt results in several real-time applications.

And here, we have used the MNIST dataset for the training and testing of our model. In this dataset, it consists of almost 60000 images which are already normalized and centered. And so each image is of size 28x28, which forms an array size of 784 pixels with values ranging from 0 to 255. Whenever a pixel value is '1', it represents that the background is white, and for black it is '0'.

Author: Department of Computer Science Engineering, Narasaraopeta Engineering College. e-mail: goutham.ch567@gmail.com

II. Implementation

a) A glimpse of the MNIST Dataset

The handwritten digit recognition is a broad research topic which gives an extensive survey of the field, including significant feature sets, learning datasets, and algorithms. The MNIST stands for the Modified National Institute of Standards and Technology dataset. It is a dataset of 60000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.

It helps to train various image processing systems and for training and testing in the field of machine learning. It was created by remixing the samples from NIST's original datasets which stands for the National Institute of Standards and Technology, a unit of the U.S. Commerce Department.

There are several scientific papers on attempts to achieve the lowest error rate as possible. An extended dataset which is similar to MNIST called EMNIST published in 2017, which contains 240000 training images, and 40000 testing images of handwritten digits and characters.

Figure 1: An example featuring MNIST dataset

The dataset used in our experiment in which, training set consists of 60000 images and 10000 images for the test set. Depending on the training and test datasets, the accuracies and performance of the algorithms may vary.

b) Support Vector Machines

Support Vector Machines helps us to find a hyperplane in N-dimensional space (N- the number of features), which classifies the data points distinctly. It is

a supervised machine learning model used for classification and regression. In this, the data points get classified applying the concept of the hyper plane. Its

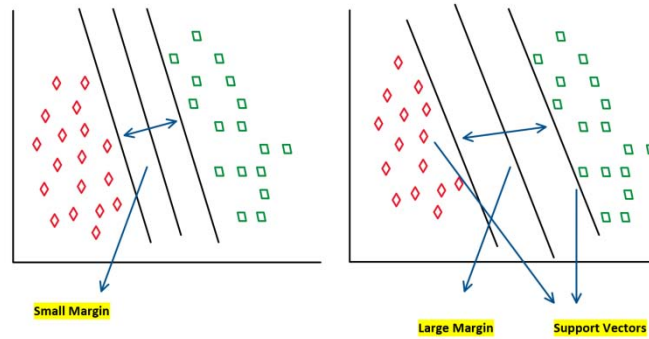dimensions may vary depending upon the number of features.

Figure 2: Support Vector Machine Classifier

The separation margin should be of equal distance from the classes. SVM offers notable immeasurable features for the classification problem. Initially, the MNIST dataset gets loaded into the Azure ML studio, and once the data preprocessing gets completed, the model training takes place using the SVM algorithm. Then the labels are predicted for the given inputs. Finally, we attain the accuracy, precision, recall, and F1 scores.

Using the two-class boosted SVM, we got an accuracy of 91% on the MNIST dataset.

In SVM, it's always wise to scale the input data so that the training time becomes less and also the model rendering takes more time if the data is more. Image of the SVM results

c) Decision Tree Algorithm

These are used successfully in many diverse areas. A Decision Tree mainly consists of three components, such as nodes: test for the value of a particular attribute, Edge/ Branch: outcome of a test and connect to the next node or leaf and leaf nodes/terminal nodes. The Decision Tree forms from the given input data. At long last, the classification procedure through a tree-like structure is consistently natural and interpretable. The two main types of Decision Trees are classification and regression. In this, we are using the Decision Tree for classification as yes/no types where a tree gets built using a process known as binary recursive partitioning. It is an iterative process of splitting the data into partitions and then splitting it up further on each of the branches.
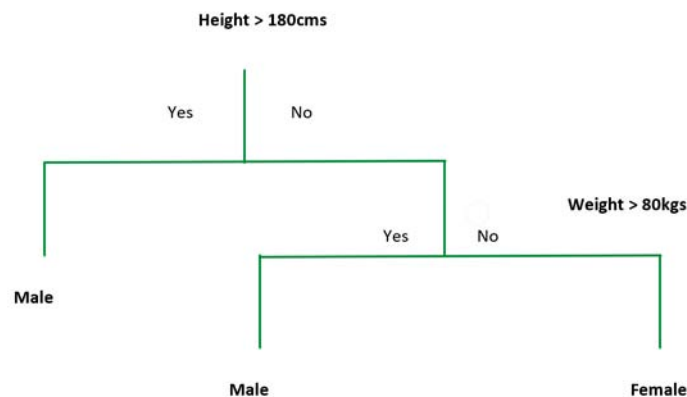

Figure 3: Decision Tree Classifier

The critical point is to use a decision tree to partition the data into cluster regions and empty regions. Initially, load the dataset into the Azure ML Studio, and after the data preprocessing, the model is trained using the two-class boosted decision tree, and then the scores are obtained.

With the use of this classifier, we have achieved an accuracy of 99.5% on the MNIST dataset.

d) Neural Networks

The usage of Neural Networks is found more in pattern recognition and image processing systems. They are multi-layer networks of neurons with non-linear mapping structures. For a set of inputs and a set of the target values, predictions made should match those target values as close as possible. A simple model consists of a connection, which transforms input to

output and a neuron that includes a bias term and activation function. A positive weight means an excitatory connection, while negative values mean inhibitory connections. All inputs are transformed and then summed. This activity refers to as a linear combination. Subsequently, an activation function controls the amplitude of the output.
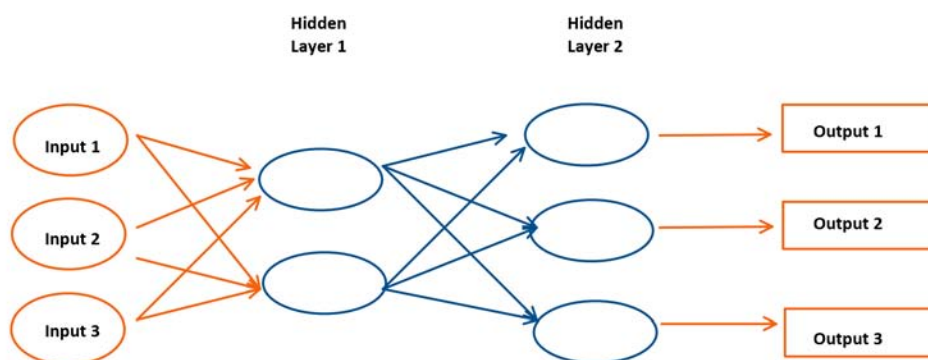
Figure 4: Neural Networks

The complexity of a model increases as the number of hidden layers increases. And so the prediction capability of the model for better performance also increases.

Here we initially load the MNIST dataset into the Azure ML Studio, and then the model is trained using the two-class Neural Networks. Using this, we achieved an accuracy of 99.8% on the MNIST dataset.

III. EXPERIMENTAL TOOLS

Azure Machine Learning Studio is a collaborative, drag, and drop tool developed by Microsoft Corporation to build, test, and then to deploy predictive analytics solutions of our data. Azure Machine Learning Studio publishes models also as web services that can easily be consumed by custom applications or tools such as Excel.

It is where everything like data science, predictive analytics, our data, and all our cloud resources meet. It is a robust and easy to use platform for deploying several machine learning models. Developing a model using this studio is like an iterative process. As we modify the various functions and their parameters, our results converge until we are satisfied that we have a trained and capable model.

It provides an interactive environment where everything is drag and drops supported to build and iterate on a predictive analysis model quickly. By connecting the required components such as datasets, algorithms, and specific analysis models (score model, train model), we can form an experiment. When the training completes, we can publish it as a web service so that others can access it. The Azure ML Studio allows us to experiment on any kind of dataset to construct a predictive analysis model for any system.

IV. RESULTS AND DISCUSSION

The scores obtained can be illustrated using the below table

Type of Algorithm	Support Vector M	Decision Tree	Neural networks
Accuracy	91.0%	99.5%	99.8%
F1 Score	95.2%	99.7%	99.9%
Precision	92.4%	99.6%	99.9%
Recall	98.1%	99.9%	99.9%

Figure 5: Accuracy, F1 Score, Precision and Recall for SVM, Decision Tree and Neural Networks

As can be seen from the above table, Neural Networks has an accuracy of 99.8%, followed by Decision Trees with 99.5% and the last Support vector Machines with 91%.

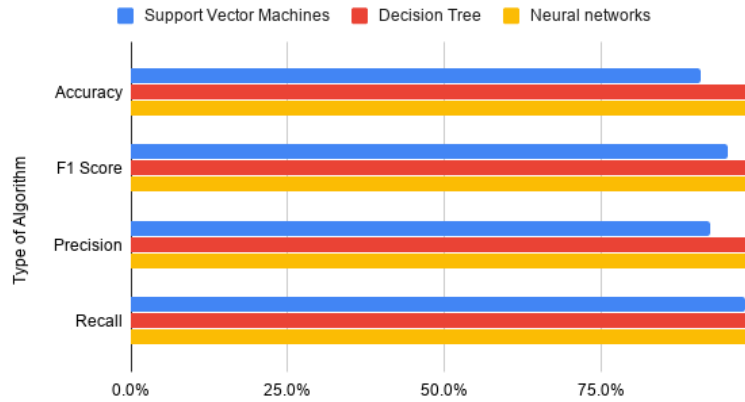## Support Vector Machines, Decision Tree and Neural networks

Figure 6: Graph for comparison of algorithms

When we see the other scores rather than accuracy, the highest F1 score is 99.9% for Neural Networks, and the lowest is 95.2% for Support Vector Machines, and the other Decision Tree has 99.7%, which is almost near to the value of Neural Networks.

Similarly, the highest Precision value is for Neural Networks with 99.9%, followed by Decision Trees with 99.6% and the other Support vector machines with 92.4%. Recall values for Neural Networks, Decision Trees, and Support Vector Machines are 99.9%, 99.9%, and 98.1%, respectively. Here the recall values for both Neural Networks and Decision Trees are the same. But when it comes to the overall score, such as Accuracy, F1 Score, Precision, and Recall values, the Neural Networks has scored best when compared to Decision Trees and Support Vector Machines.

V. CONCLUSION

In this paper, the performances of various algorithms like Support Vector Machines, Decision Trees and Neural Networks have been weighed and analyzed to reveal the best classifier for adequate recognition of handwritten digits using Microsoft Azure Machine Learning Studio. In any recognition process, the important thing is to preprocess the data and then adequately train the model using all necessary measures. Using the Azure Machine Learning Studio, we can do all the processing within less time.
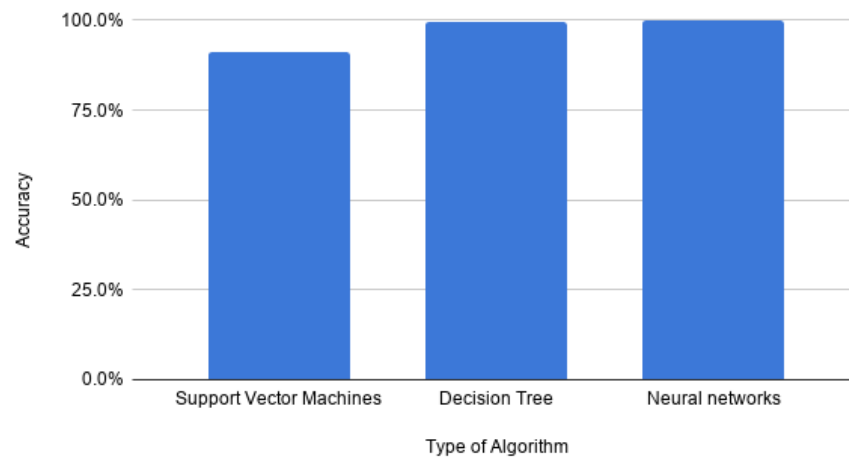
Accuracy vs. Type of Algorithm

Figure 7: Accuracy vs. type of algorithm

The overall highest accuracy, 99.8%, is achieved by Neural Networks in the recognition process. This paper is an attempt to analyze different models for the recognition process to unveil the best classifier. Therefore, we can conclude that Neural Networks give better performance for handwritten digit recognition.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Seewald, (2011). "On the brittleness of handwritten digit recognition models". ISRN Machine Vision, 2012.

2. Serbia & Montenegro, Belgrade, (2005). Handwritten Digit Recognition by Combining SVM Classifiers. EUROCON, 2005.

3. Perwej, Y., & Chaturvedi, A Machine recognition of handwritten characters using neural networks. arXiv preprint arXiv:1205.3964, 2012.

4. Support Vector Machine, Online: https://en. wikipedia.org/wiki/Support_vector_machine, Accessed, February 2020.

5. Neural Networks, Online: https://towardsdata science.com/understanding-neural-networks-19020 b758230, Accessed, March 2020.

6. D. Rajeswara Rao, Meer Zohra, A Comprehensive Data Analysis on Handwritten Digit Recognition using Machine Learning Approach, (IJITEE), Volume-8 Issue-6, April 2019.

7. Burel, G., Pottier, I., & Catros, J. Y. (1992, June). Recognition of handwritten digits by image processing and neural network. In Neural Networks, 1992. IJCNN, International Joint Conference on (Vol. 3, pp. 666-671). IEEE.

8. D. Gorgevik and D. Cakmakov, "Handwritten digit recognition by combining SVM classifiers", November 2005.

9. Tuan Trung Nguyen, "Adaptive Classifier Construction: An Approach to Handwritten Digit Recognition ", J.J. Alpigini et al. (Eds.): RSCTC, LNAI 2475, Springer-Verlag Berlin Heidelberg, pp. 578–585, 2002.

10. Arora, S. Bhattacharjee, D. Nasipuri, M. Malik, L. Kundu, M and Basu, D. K.2010. "Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, pp-18-26.

11. Nasien, D. Haron, H and Sophiayati Yuhaniz, S.2010. "Support Vector Machine (SVM) for English Handwritten Character Recognition," Computer Engineering and Applications (ICCEA), 2010 Second International Conference on, vol.1, no., pp.249-252.

12. MNIST Database, "The MNIST database of handwritten digits," Online: http://yann.lecun.com/ exdb/mnist, Accessed, October 2019.

13. Salzberg, S. L. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. Machine Learning, 16(3), 235-240.

14. Support Vector Machines, Online: https://towards datascience.com/support-vector-machine-introduc tion-to-machine-learning-algorithms-934a444fca47, Accessed, March 2020.

15. Decision Trees in Machine Learning, Online: https:// towardsdatascience.com/decision-trees-in-machine -learning-641b9c4e8052, Accessed, February 2020.