MNIST Handwritten Digit Classification

Nishant Gupta
Technische Universität München
nishant.gupta@tum.de, Mtk. Nr.: 03646526

***Abstract*— The central idea of this exercise is to provide a comparative analysis on SVM, Decision Tree and Random Forest classifiers to classify handwritten digits.**

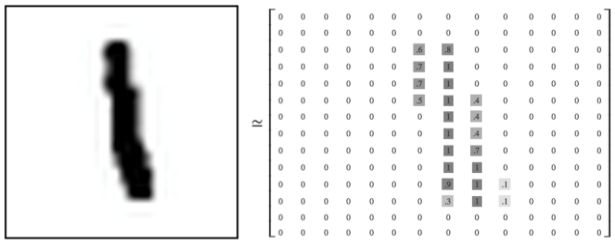
I. Introduction

Handwritten digits can be classified into 10 possible classes ('1', '2', '3', '9', .., '0') using supervised learning techniques. To enhance the performance of system, preprocessing can be done on the training set.

II. Dataset Description

The MNIST dataset [1] contains 60,000 digits ranging from 0 to 255, each value represents the equivalent shade of the pixel. The higher the value the more dark the pixel is, for example 255 represents the darkest pixel. Each image in the dataset has a height of 28 pixels and a width of 28 pixels. This makes the data to have 28 x 28 = 784 features to train on. The dataset also consists of a training set of 10,000 digits. However, the dataset seems to provide best training results when normalized.

A sample representation of a digit from the dataset (after normalization) can be:



III. Data Preprocessing

Data preprocessing is the step to carefully screen the data for a problem statement in order to improve the quality of data that is being fed into prediction algorithms [2]. In this exercise, mean pixel intensity subtraction has been done as a part of preprocessing.

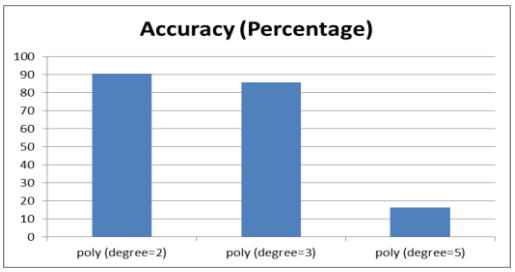
IV. Classifiers Specification

A. Support Vector Machines

Only 1000 training samples were taken into account to start with. Parameters were tuned by using different kernels and setting their corresponding parameters. Following types of SVM kernels settings were used in the experiments where 1000 samples from training set were trained and then tested on complete test set of 10000 entries:

a) Linear Kernel:Accuracy of 87.58% was achieved.

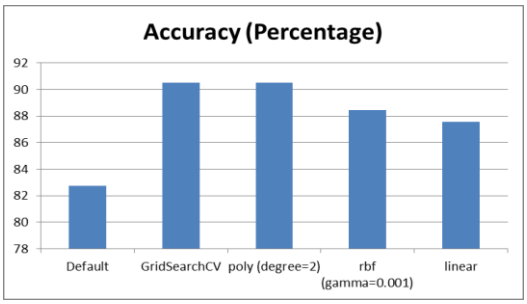
b) Polynomial Kernel: Accuracy of 90.53% was achieved with degree=2 and C=1000



c) RBF Kernel: Accuracy of 88.42% is achieved with gamma = 0.001 and C = 1000

d) SVM with default settings: Accuracy of 82.75% is observed.

e) Cross Validation using GridSearchCV(): After 5 fold cross validation on 1000 records training set, it was concluded that best parameter settings for this dataset is that of Polynomial Kernel with degree=2 and C=1000.



After training complete dataset of 60000 records using SVM with polynomial kernel, an accuracy of 98.29% was achieved on complete test set of 1000 records.

B. Decision Trees

Following scenarios were experimented:

a) Default Parameters: Accuracy of 88.32% was achieved when trained on complete train dataset.

b) Parameter Tuning: GridSearchCV() was again used for cross validation in order to get best parameters for achieving maximum accuracy. Three parameters – criterion, max_depth, and max_features were altered in the experiment to achieve best parameter set. Maximum score in cross validation was achieved for the parameter settings: criterion: entropy, max_depth: 50, max_features: 500. Accuracy of 89.49% was achieved when trained on dataset of 60000 records and tested on test set of 10000 records.

Following image shows the pixel importance in the 28x28 image for Decision Tree Classifier.

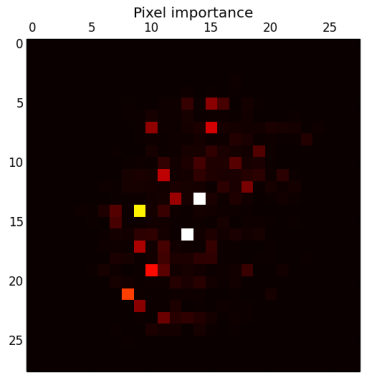


Figure 1. Decision Trees Classifier

C. *Random Forest*

Following scenarios were experimented:

a) Default Parameters: Accuracy of 94.96% was achieved when trained on complete train dataset

b) Parameter tuning with GridSearchCV(): Four parameters – n_estimators, criterion, max_depth, and max_features were tuned in the experiment. Parameters settings: n_estimators: 50, criterion: entropy, max_depth: 50, max_features: 50 gave the best score. With this parameter setting, when trained on complete train dataset and then tested on given test set, an accuracy of 97.82%.

Following image shows the pixel importance in the 28x28 image for Random Forest Classifier.

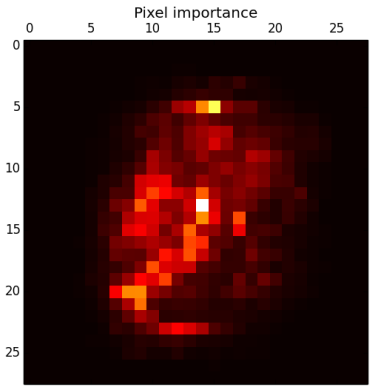


Figure 2. Random Forest Classifier

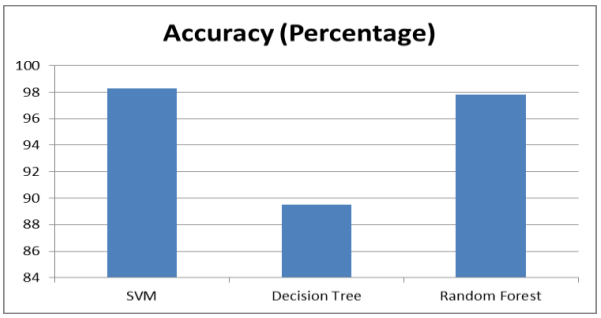
V. RESULTS

Following table shows the comparative analysis of three different classifiers for the MNIST dataset used for handwritten digit recognition. Preprocessing led to improvement in accuracy percentage for each classifier. Among these, SVM outperforms the rest by considerable margin, both for raw data and preprocessed data.

TABLE I.

Classifier	Parameter Setting		Accuracy
SVM	Default	Raw Data	82.75%
		Preprocessed	83.56%
	Fine Tuned	Raw Data	98.02%
		Preprocessed	98.29%
Decision Tree	Default	Raw Data	87.86%
		Preprocessed	88.32%
	Fine Tuned	Raw Data	88.88%
		Preprocessed	89.49%
Rndom Forest	Default	Raw Data	94.96%
		Preprocessed	96.46%
	Fine Tuned	Raw Data	96.84%
		Preprocessed	97.82%

Following graph compares the best cases of each of three classifiers used in the experiments. SVM as inferred clearly has best accuracy rate of 98.29%



STATE OF THE ART WORK

Back-propagation for plain multi-layer perceptrons yields a very low 0.35% error rate [3]. Convolutional Neural network again brought very minimal error rate when applied on MNIST dataset [4]. With 0.21% error rate achieved, DropConnect is very successful. DropConnect [6] is a generalization of Dropout [5], for regularizing large fully connected layers within neural networks.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998

[2] Cloah's blog on visualizing the MNIST dataset, http://colah.github.io/posts/2014-10-Visualizing-MNIST.

[3] Chellapilla, K., Puri, S. & Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing. 10th International Workshop on Frontiers in Handwriting Recognition, 2006.

[4] Dan Cires¸an, Ueli Meier and Jurgen Schmidhube "Multi-column Deep Neural Networks for Image Classification"

[5] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580, 2012

[6] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus "Regularization of Neural Networks using DropConnect".