MNIST digit recognition analysis
Lab report

Auer Thomas

September 12, 2022

Contents

1. Introduction

The following report describes various setups for experimentation on the MNIST database of handwritten digits[1] using support vector machines, principal component analysis and neural networks.

The goal is to analyze the results and runtime of the various algorithms, given multiple configurations. For this purpose the libraries "SKLearn"[2] and "Keras"[3] of the programming language "Python" are used.

2. Methods and Resources

The methods chosen for the analysis include several well-known libraries. The imported dataset for computation is the mnist-dataset provided by the keras.datasets package. All python scripts include the datetime and time packages for timing purposes, while matplotlib offers some visualization functionalities (only for the .IPYNB files) for datasets. With the implementation of 4 python programs, each script uses it's own respective set of packages.

1. SKLearn support vector machine
2. SKLearn support vector machine with principal component analysis
3. SKLearn multi-layer perceptron classification, with and without PCA
4. Keras sequential model

The scripts implementing the SKLearn-Kit also allow for hyperparameter search using GridSearchCV, which allows automated customization and inspection of each algorithm over various pre-defined parameters in search of an optimal score for a specific criteria, here "accuracy". The program implementing the Keras library is further sophisticated by displaying which images were difficult to classify for the classification algorithm and a confusion matrix over the general results.

Approach

The programs data initialization is the same for all algorithms. After fetching the data from the keras.datasets-database, it is necessary to reshape and normalize the data for machine

[1] http://yann.lecun.com/exdb/mnist/
[2] https://scikit-learn.org/
[3] https://keras.io/

learning purposes. The reshaping of the data is the transformation of the original three-dimensional dataset to an two-dimensional training set. We reshape the data such that we have access to every pixel of the image. The reason to access every pixel is that only then we can apply deep learning ideas and can assign color code to every pixel. Furthermore, we know the RGB color code where different values produce various colors. We know that each pixel has its unique color code and also we know that this code has a maximum value of 255. To perform Machine Learning, it is important to convert all the values from 0 to 255 for every pixel to a range of values from 0 to 1. Lastly, we optionally reduce the size of the training and testing data as to speed up the experimentation. Afterwards, the algorithms are supplied with the training data and the model fitting begins. In the context of this research, the accuracy and the execution time are used to rate each algorithm.

3. Results

The following results have been computed over one-fourth of the original data set size and with default parameters, except for the kernel-type for SVMs and Layer-sizes for Neural Nets, of each package.

Algorithm	PCA-Enabled	Accuracy	Time
Linear SVM	No	0.8935574229691877	2.54819655418396s
Linear SVM	Yes	0.101640656262505	2.3763396739959717s
Poly SVM	No	0.938375350140056	3.057605504989624s
Poly SVM	Yes	0.09683873549419768	4.935593128204346s
RBF SVM	No	0.9487795118047219	6.784686803817749s
RBF SVM	Yes	0.10004001600640255	8.273269176483154s
Sigmoid SVM	No	0.7911164465786314	4.3955934047698975s
Sigmoid SVM	Yes	0.10284113645458183	3.973496198654175s
SKLearn MLP	No	0.9451780712284914	0.005684375762939453s
SKLearn MLP	Yes	0.101640656262505	0.008093595504760742s
Keras Sequential	N/A	0.9766651391983032	0.6482634544372559

Table 3.1: Algorithms and their results.

These values serve as a baseline for further experimentation, in form of hyperparameter estimations. Hyperparameter search using GridSearch has suggested the following optimal choice of parameters.

4. Discussion

Surprisingly, the default setup using principal component analysis has significantly worsened the accuracy of every algorithm while not significantly improving the execution time.

5. Conclusion