Machine Learning Project Report

First Author

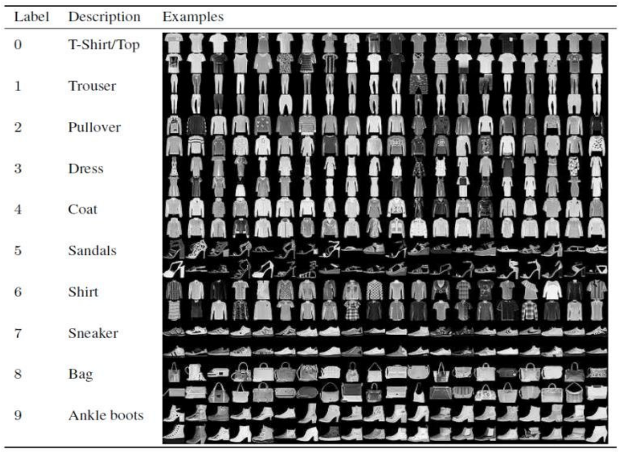
Berke Furkan Küsmenoğlu

1. Introduction

This document presents an application of a neural network for image classification problems. This work is part of my experiments with the Fashion-MNIST dataset using various Machine Learning algorithms and models. Our aim is to predict different fashion products from the given images using various Machine Learning Models and compare their results to find out the best ML model. I used the Keras model and for the training algorithms, I did my experiments on ”Decision Tree, Random Forest, Logistic Regression, KNN, and SVC. Also analyzed the accuracy with changing parameters (e.x: branch size, epoch) or changing the training algorithms parameters plus I did an experiment for seeing the effects of adding more layers to the model. For analyzing the accuracy scores I used graphs and I did different values for different models and algorithms. I achieve my best score by using Neural Network and it was %89.90. I will be showing my graphs and tables following section on the document

2. Dataset

MNIST(Modified National Institute of Standards and Technology Database) problem is one of the most common problem that has been used in Machine Learning. The purpose is to be able to identify the digit based on an image. But we can not say MNIST is the best option to use because we can achieve a very good value of accuracy even if we are looking at a few pixels in the image. The original MNIST dataset contains handwritten digits. Machine Learning and Data Science community prefer this dataset to validate their algorithms. There are common ideas that MNIST became the first option to try because everybody is sure that it will work on the MNIST data set but there is a big chance that can fail in other datasets. MNIST is usually preferred as the first dataset they would work on. So, because of that better option to work for this problem and the testing algorithms is Fashion-MNIST. Fashion MNIST consists of 60,000 images and each image has 784 features (28×28 pixels). 60.000 images are had in the training set and the test set has 10, 000 images.[1]



Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

As part of my project, I applied transactions and reshaping to my data. It is often the case that neural networks are trained with 32-bit precision, so at some point, the training data will need to be converted to 32-bit floats. Because the dataset fits easily in RAM, we might as well convert to float immediately. Regarding the division by 255, this is the maximum value of a byte, so this ensures that the input features are scaled between 0.0 and 1.0 in order to ensure the default learning rate works properly and so that the cost can take on reasonable values.

3. Method

For my data, I used Neural Network because with NN we are able to learn non-linear and complex relationships and model them. Lots of people in Machine Learning and Data Science confirmed that NN plays a big role in image and character recognition. There are no restrictions on the input variables of NN, unlike many other prediction methods. Additionally, many studies have shown that NN can better model for data with high volatility and non-constant variance, given their ability to learn hidden relationships in the data without imposing any fixed relationships in the data[2]. With this information, I decided to apply NN to my data but I also applied other models to my data to analyze the values and we make sure about the accuracy of my research.

3.1. Methods

I applied five different methods other than NN. We present some classification results in the tables below to create a comparison of this dataset. A separate trial was made for each algorithm and optimal values were determined by changing the parameters of the algorithm.

3.1.1 Decision Tree

After analyzing the table for the Decision Tree we can say if we decrease the max_depth, the value of accuracy increases for this dataset.

Decision Tree		Test
max_depth	10	0.8052
max_depth	50	0.7848
max_depth	100	0.7857

Table 1. Decision Tree

3.1.2 Random Forest

In Random Forest I analyzed the accuracy by changing the parameters which are max_dept and n_estimators. We got the highest accuracy when n_estimators is equal to 100 and max_dept is equal to 50. We can assume that these are values for optimal results for Random Forest.

Random Forest		Test
max_depth n_estimators	100 100	0.8719
max_depth n_estimators	100 50	0.8699
max_depth n_estimators	100 10	0.0.8476
max_depth n_estimators	50 100	0.8727
max_depth n_estimators	50 50	0.8708
max_depth n_estimators	50 10	0.8539
max_depth n_estimators	10 100	0.844
max_depth n_estimators	10 50	0.8443
max_depth n_estimators	10 10	0.8344

Table 2. Random Forest

3.1.3 Logistic Regression

For the Logistic Regression, I analyzed my data by giving values to ”C” and multi_class parameters. We got the highest accuracy when C is equal to 100 and multi_class is equal to ”auto”. We can not say accuracy increase when the value increases but we can say if multi_class is equal to ”auto” when the value of C increases our accuracy increase too.

Logistic Regression		Test
C = multi_class	1 ”ovr”	0.8409
C = multi_class	1 ”auto”	0.8343
C = multi_class	10 ”ovr”	0.8267
C = multi_class	10 ”auto”	0.8437
C = multi_class	100 ”ovr”	0.8411
C = multi_class	100 ”auto”	0.8449

Table 3. Logistic Regression

3.1.4 KNeighbour

KNeighbour has another parameter which is ”p”. I gave ”1” and ”2” as the value of p but when p got 2, execution time increased a lot. Because of that I gave ”1” as the value of p and made changes on n_neigbour and weight. We got the highest value when n_neigbour is equal to 5 and weights’ value was ”uniform”.

KNeighbour		Test
n_neigbour = weight	1 ”uniform”	0.8428
n_neigbour = weight	1 ”distance”	0.8428
n_neigbour = weight	5 ”uniform”	0.8527
n_neigbour = weight	5 ”distance”	0.8491
n_neigbour = weight	8 ”uniform”	0.8477
n_neigbour = weight	8 ”distance”	0.8486

Table 4. KNeighbour

3.1.5 SVC

When analysing the SVC model, I changed the ”C” parameter. For large values of C, the optimization will choose a smaller-margin hyperplane if it does a better job of categorizing all the training points correctly. In my project, I experienced an issue that for SVC when we increase the value of C execution time increase extremely high. When C is equal to 10, the execution time was 12 minutes.

SVC		Test
C =	1	0.8425
C =	10	0.8328
C =	100	0.8258

Table 5. SVC

4. Experiments

As an experiment, I did work on the analyze the model accuracy and model loss based on the changes in the epoch size. Also for NN, we have different optimization algorithms like SGD and ADAM. With SGD, the optimization algorithm can only handle a small sample of data examples.Adam, on the other hand, replaces stochastic gradient descent for deep learning model training.. I did an experiment for these optimizers and analyzed the value changes with different epoch values. In addition to that, I analyze the accuracy and loss graph by making changes on epoch and batch size values to find the optimal values for our dataset.[3]

4.1. Optimizer

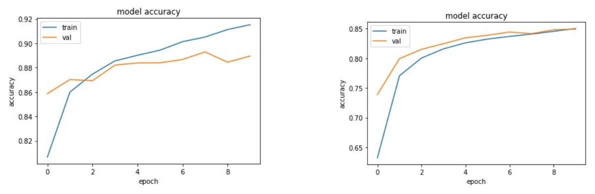


Figure 1. Model accuracy

We tried SGD and ADAM optimizer with different values of epoch and we obtain the table above. We can come to this conclusion that when epoch values increase for both of the optimizers, their accuracy rate increases too but here I notice if we look at the accuracy graph above (Figure 1.) SGD’s graph is better than ADAM’s graph.

4.2. Batch size

For this experiment epoch value was ”10”. If we analyze the table, we got the highest value when the value of the accuracy was 200. Besides that, all of the accuracy rate

Optimizer	epoch	Accuracy rate
SGD	10	0.8428
SGD	50	0.8432
SGD	100	0.8629
ADAM	10	0.8764
ADAM	50	0.8909
ADAM	100	0.8990

Table 6. SGD-ADAM

values are very close to each other. Here we need to mention how did our values come up with these batch values. For analyzing that we can look at model accuracy and model loss graph.

Batch size	Accuracy rate
16	0.8781
32	0.8792
64	0.8766
128	0.8763
200	0.8835
300	0.8796
400	0.8753
500	0.8735

Table 7. Batch-size

In the below graphs we see model accuracy graph but values of batch size for the first graph 16 and for the second graph 500. We can clearly say that the differences in the batch sizes values affect the values and makes them more balanced. For this graph as an optimizer, I used ”ADAM”.

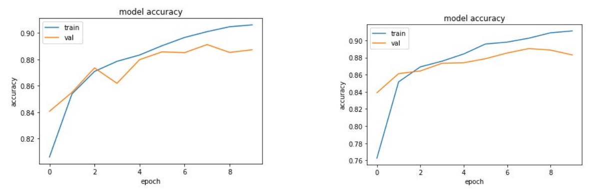


Figure 2. Model accuracy

5. Reference

[1] Deepak Singh on March 19, 2019 Fashion-MNIST using Machine Learning

[2] Jahnavi Mahanta on July 10, 2017 Introduction to Neural Networks, Advantages and Applications

[3] Jason Brownlee on July 3, 2017 in Deep Learning Performance