Pierre Casco

Lab 09 – Fashion MNIST

IST718

Introduction

With so many approaches for classification, it is important to understand which ones perform best in terms of time and computing resources. For this lab I used three different classification algorithms on the Fashion MNIST dataset. The three algorithms used were Keras, neural network, and a random forest.

Research Question

Can we use algorithms and compute to identify clothing items?

Data

As mentioned before, Fashion MNIST is being used to evaluate the three algorithms for classification. Fashion MNIST is a dataset of Zalando's articles images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image associated with a label from 10 classes (figure 1).

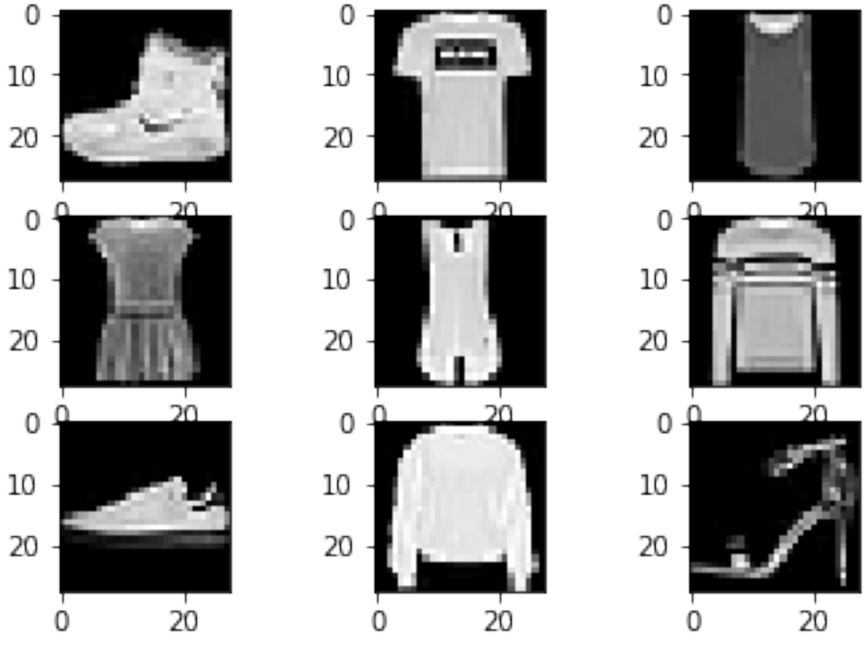


Figure 1: Fashion MNIST sample

Keras

The first algorithm used was Keras. Keras is a high-level neural network API written in Python and capable of running with TensorFlow. The code for this model was adapted from the Keras tutorial. The benefit to using this model was the ease of use. Setting up was not difficult, however, there was some compute time involved. The first pass with made using adam as the optimizer. When looking at the training and validation accuracy, the performance of the model did not seem consistent (figure 2). There were drop offs in the training accuracy of the model around 150 epochs. When I changed the optimizer to a stochastic gradient descent optimizer, the performance improved and the dips in accuracy went away (figure 3). The baseline error for the model was 13.5%.

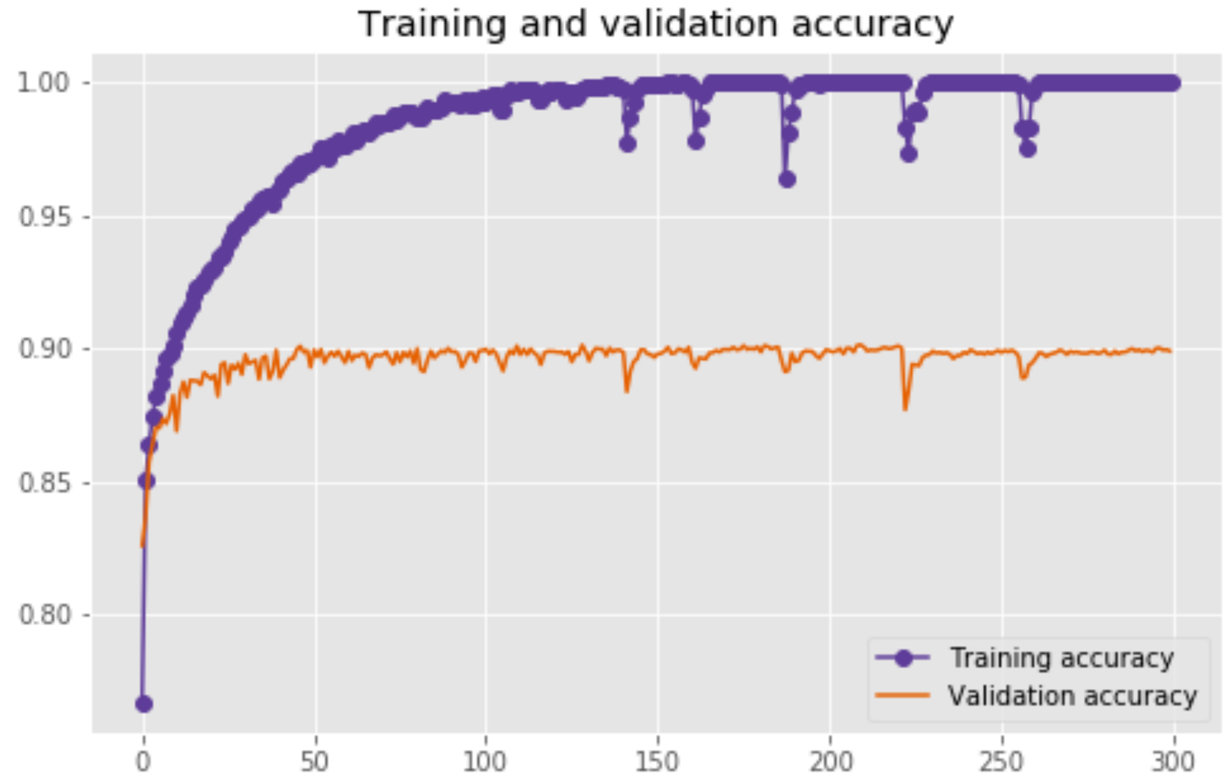


Figure 2: Keras model using adam as the optimizer.

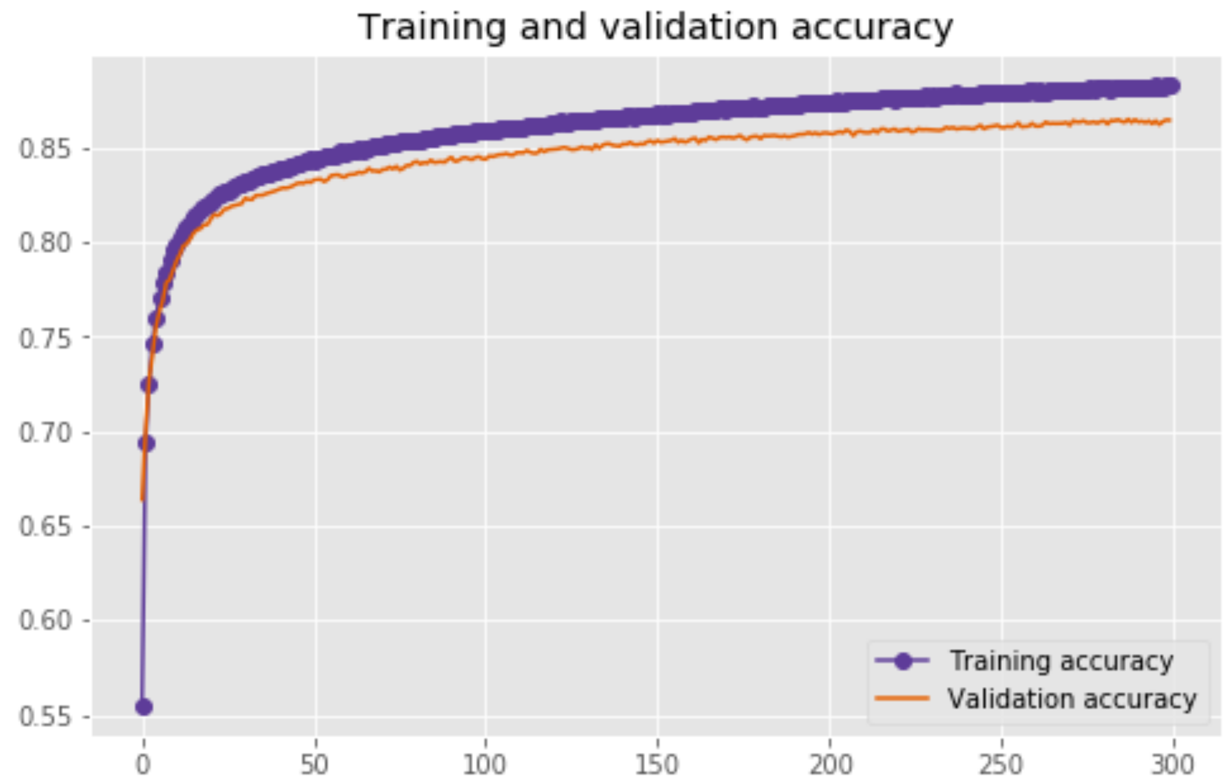


Figure 3: Keras model using SGD as the optimizer.

Neural Net

The next algorithm used was a neural net. The code for this model was adapted from the TensorFlow tutorials on basic classification. The dataset was passed through 256 nodes in the hidden layer. Prior to creating the model, the dataset was normalized. The model was fit using 10 epochs. Accuracy of the training model was 91.3%. The accuracy of the test was 87.8%. The model appeared to have trouble classifying sandals and sneakers (figure 4).

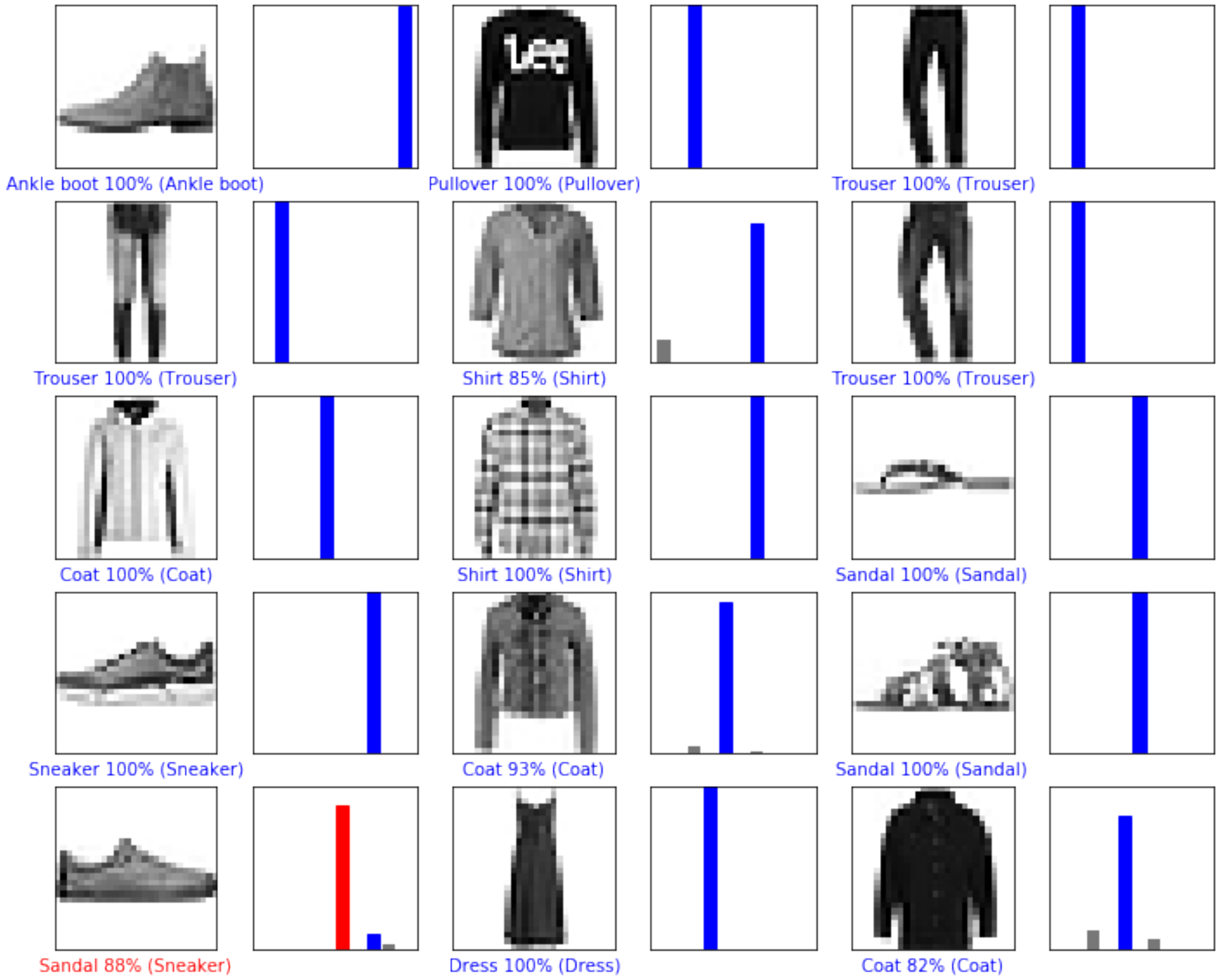
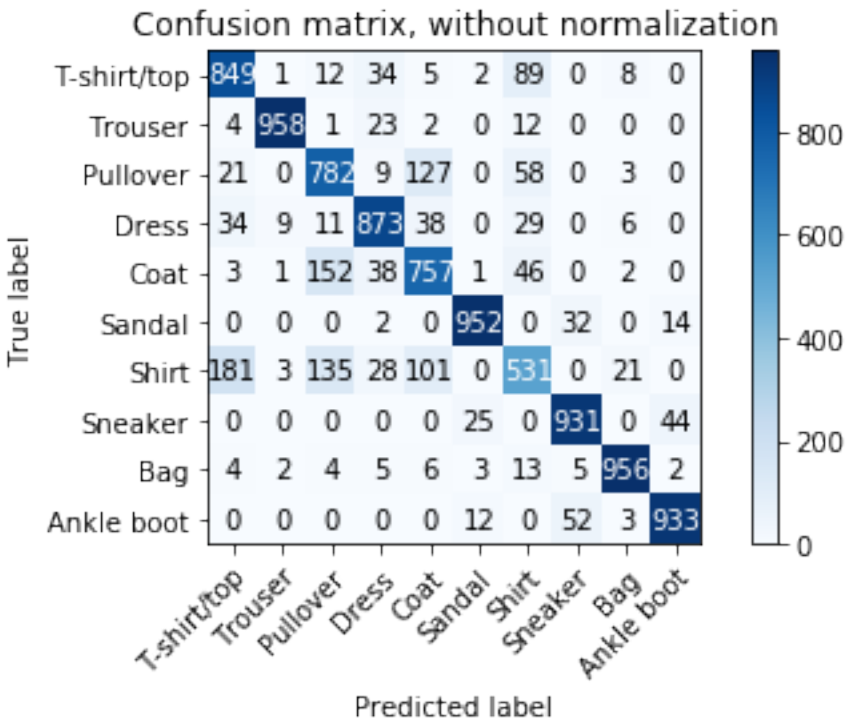


Figure 4: Performance of the neural net.

Random Forest

The last algorithm used was a random forest. The random forest was the easiest to model of all three algorithms used. Once the dataset was reshaped, the model was built, and the confusion matrix was generated. Performance of the training model was 99.5%. The testing model performance was 85.2%. Looking at the confusion matrix for the test model, the model had the most issues classifying t-shirt/top with shirts and it did the best classifying trousers.



	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	849	1	12	34	5	2	89	0	8	0
Trouser	4	958	1	23	2	0	12	0	0	0
Pullover	21	0	782	9	127	0	58	0	3	0
Dress	34	9	11	873	38	0	29	0	6	0
Coat	3	1	152	38	757	1	46	0	2	0
Sandal	0	0	0	2	0	952	0	32	0	14
Shirt	181	3	135	28	101	0	531	0	21	0
Sneaker	0	0	0	0	0	25	0	931	0	44
Bag	4	2	4	5	6	3	13	5	956	2
Ankle boot	0	0	0	0	0	12	0	52	3	933

Comparing 3 algorithms

	Keras	Neural Net	Random Forest
Training Accuracy	88.4%	91.3%	99.5%
Test Accuracy	86.5%	87.8%	85.2%
Compute Performance	99.427 seconds	50.712 seconds	10.589 seconds
Trade off	Slowest compute performance of all the algorithms.	Most accurate with the test set but tougher to set up.	Quick but lest accurate with test dataset.

Summary

Of the three models used, Keras, Neural Net, and Random Forest, the quickest model with the lowest accuracy was the random forest. The best model was the neural net but it was a bit tougher to set up.