

Project: Fashion Mnist project report

Introduction:

Fashion-MNIST is a dataset of Zalando's article images. This dataset consists of 70000 data (60000 for training and 10000 for test the model). Each data is 28×28 (total 784 pixels) grayscale images. This images will be classified in 10 groups:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress
- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boot

In this project, we want to make a model to specify from the picture what kind of cloths a person wore. We can use this technology for suggesting products to customers based on their profile picture or etc.

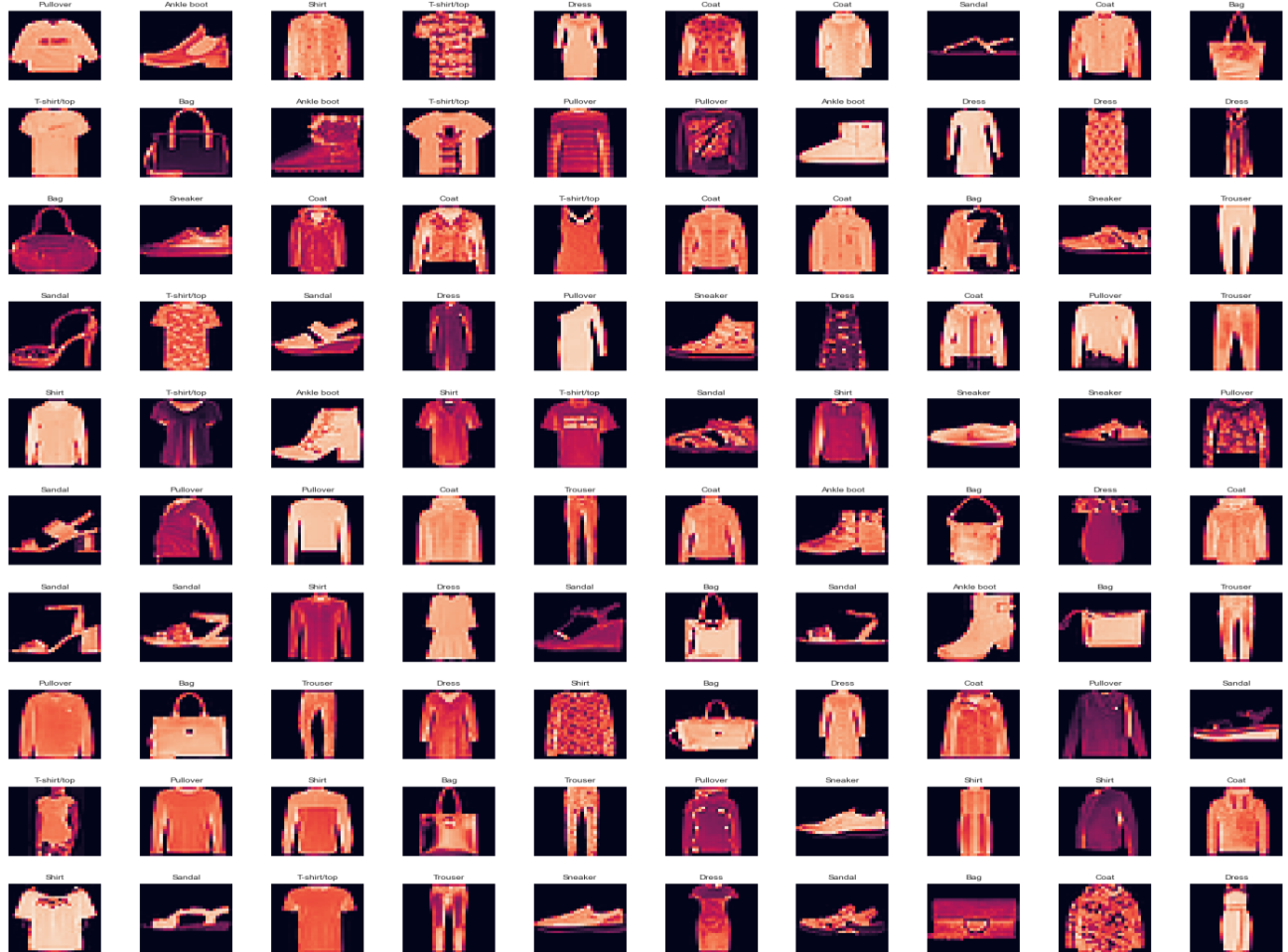
Dataset:

In this dataset, column 0 specify the group of the picture that belongs to.

From column 1 to 784 will should the value for images from pixel 1 to pixel 784.

Visualizing data:

We can visualize data. For example, here we define to show 10×10 images (10 image in each row and 10 image in each column, totally 100 images from training dataset) and each picture label (each image belongs to which class).



Make model:

To make a model, we will use CNN (convolutional neural network).

We have 3 types of dataset here:

1. Training dataset (80 % percent of training dataset (60000 data))
2. Validation dataset (20 % percent of training dataset (60000 data))
3. Test dataset (consist of 10000 data)

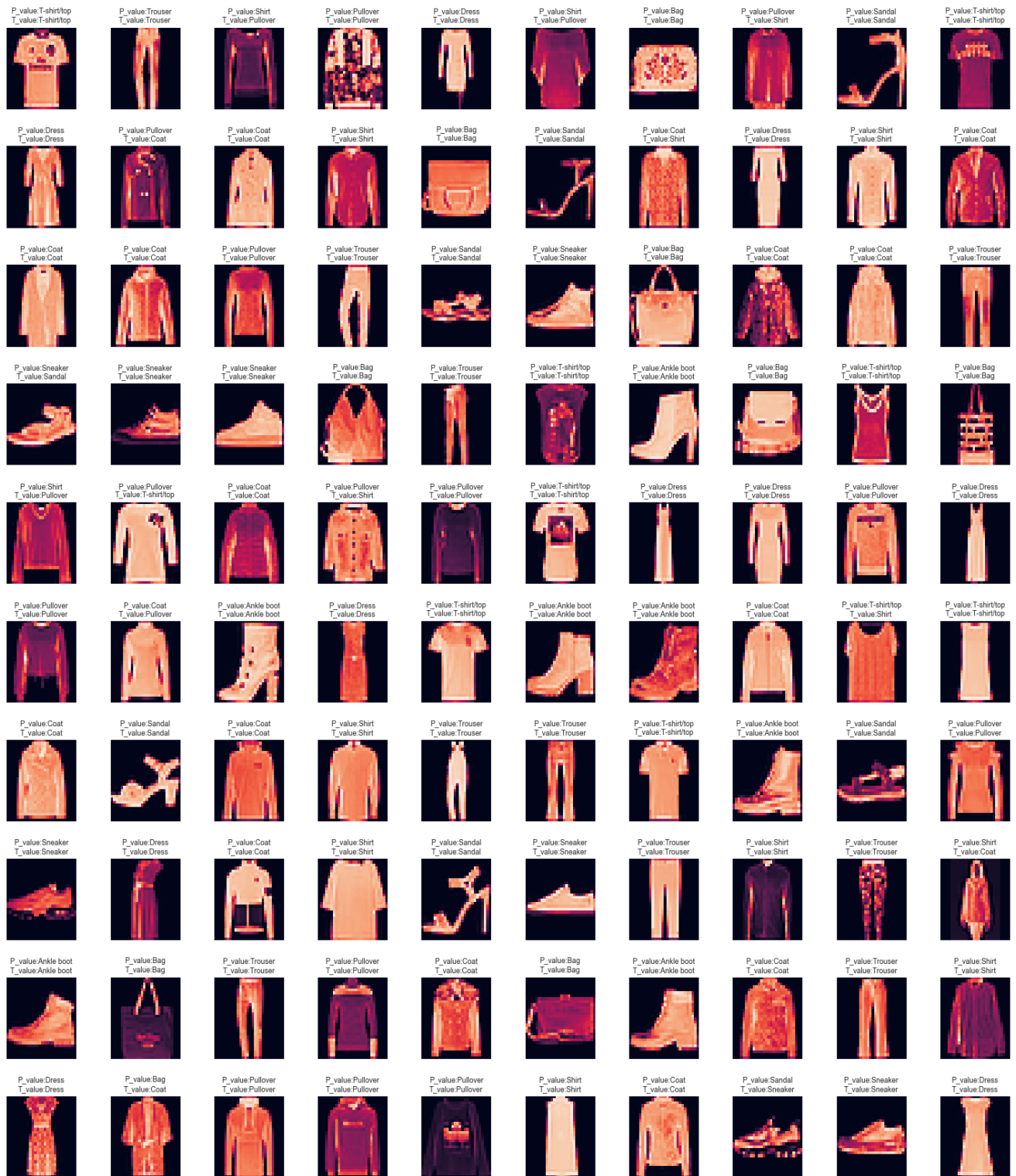
After defining these 3 types of dataset, we convert X values for these dataset to 28*28 matrices.

Then make a CNN model and train that model by using training set and validation set.

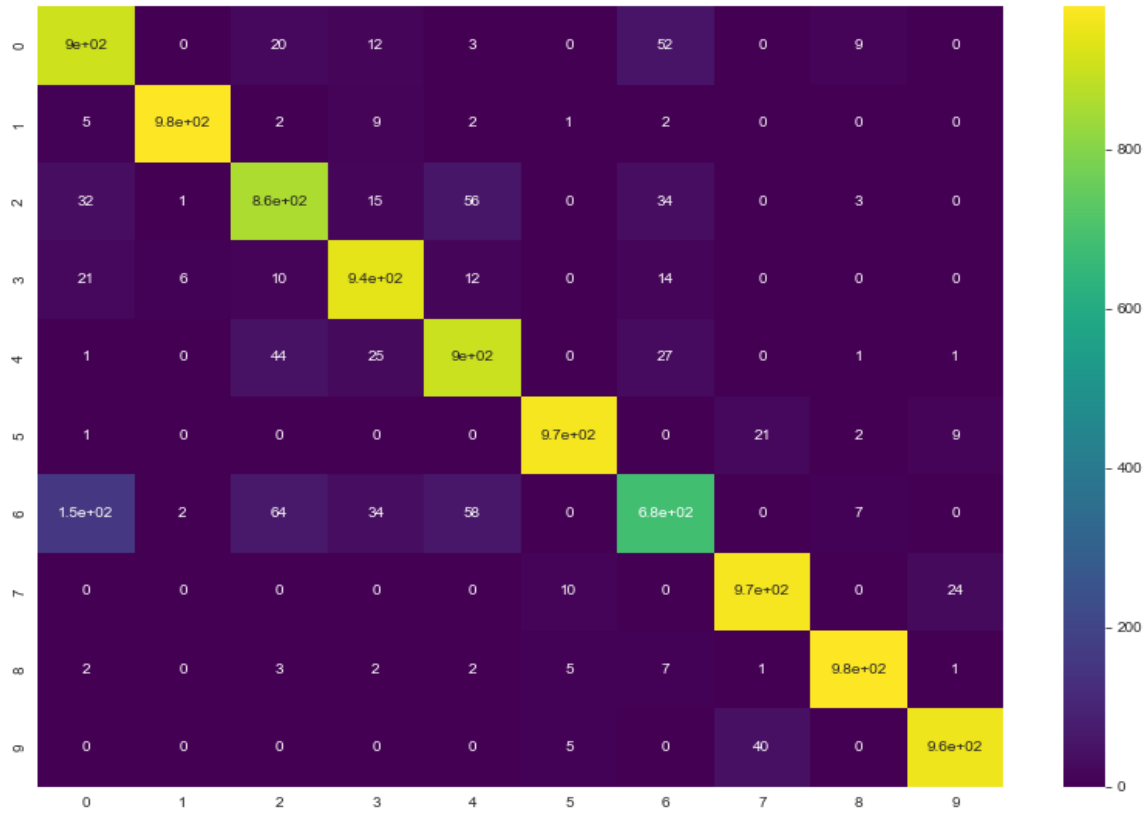
Evaluate model:

After training model, we predict classes for test set and then compare predicted values with the real values. Accuracy with 32 kernel and without dropout is 0.9653 and test accuracy is 0.9171.

Actually, we can visualize part of data and demonstrate their predicted value and real value.



We can draw heat map plot to see confusion matrix for created model.



Y axis shows predicted values and X axis shows real values.

Based on this plot, we can see the model can make a mistake when a picture belongs to class (0 = T-shirt/top) but the model predicted as class (6 = shirt).

	precision	recall	F1-score	Support
Class0	0.81	0.90	0.85	1000
Class1	0.99	0.98	0.98	1000
Class2	0.86	0.86	0.86	1000
Class3	0.91	0.94	0.92	1000
Class4	0.87	0.90	0.89	1000
Class5	0.98	0.97	0.97	1000
Class6	0.83	0.68	0.75	1000
Class7	0.94	0.97	0.95	1000
Class8	0.98	0.98	0.98	1000
Class9	0.96	0.95	0.96	1000
Accuracy			0.91	10000
Macro avg	0.91	0.91	0.91	10000
Weighted avg	0.91	0.91	0.91	10000

Based on precision, recall and F1-score, we can see this model cannot predict class 6 properly. So we need to improve our model.

Improve model:

To improve our model, we can increase the filter when we add convolutional layer to our CNN model (instead of 32 we can use 64 or 128).

Or we can use dropout technique.

Accuracy with 64 kernel and without dropout is 0.9795 and test accuracy is 0.9193.

Accuracy with 64 kernel and with dropout is 0.9427 and test accuracy is 0.921.

Conclusion:

In this project, we want to make a model to labeling the grayscale pictures. first of all, we used CNN model with 32 kernel without dropout layer, here we can see accuracy for training dataset is 0.9653 and for test dataset is 0.9171, then we used 64 kernel without dropout layer, accuracy for training set is 0.9795 and for test set is 0.9193, then we used 64 kernel with dropout layer, accuracy for dataset is 0.9427 and for test set is 0.921.

We can see when we use dropout layer that decrease the accuracy for training set but will increase the accuracy for test set. For our project accuracy for test set is more important than accuracy for training set because test set data are data that model have been never seen them before.