A Report

On

**Deep Learning based classification of the Fashion MNIST dataset**

BY

**Abhinava M 2018A8PS0844H**

**Praneet Sai Madhu Surabhi 2019A7PS0060H**

**Rohit Kushwah 2018B5A71062H**

**D Sri Krishna Karthik 2019A7PS0189H**

Under the supervision of

**Prof. N.L. Bhanu Murthy**

**SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS OF**

**CS F425: Deep Learning**

**Assignment-2**



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)**

**HYDERABAD CAMPUS**

**(December 2021**)

**The Dataset:**

Fashion-MNIST is a Zalando article image dataset that includes a training set of 60,000 samples and a test set of 10,000 examples. Each sample is a 28x28 grayscale image with a label from one of ten categories. Fashion-MNIST is intended to be a drop-in replacement for the original MNIST dataset for evaluating machine learning methods, according to Zalando. The training and testing splits have the same picture size and structure.

**Preprocessing:**

Each sample in the dataset has a 28x28 pixel image with an integer associated with the pixel ranging from 0 to 255. The higher the number, the more the intensity. Each sample is assigned with one of the 10 labels:

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

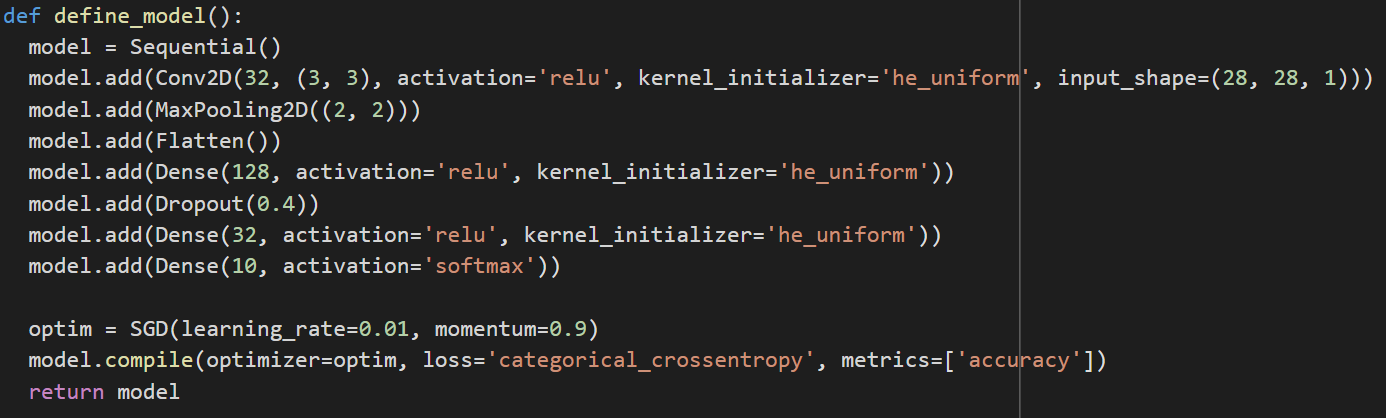
The data is normalized by dividing the entire set with 255.

**Comparative study of Models**

We have evaluated an (CNN) model with different designs and architectures by varying the Number of Hidden layers, activation functions, neurons in each layer and loss functions. The training and testing errors are as follows.

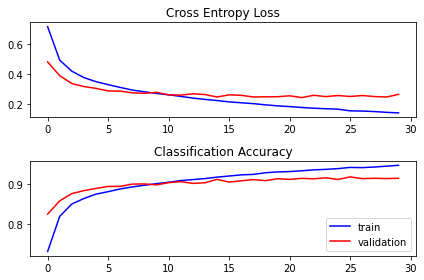
| Model | Loss | #hidden layers | Activation Function | #Hidden Neurons | Train Accuracy | Validation Accuracy | Test Accuracy |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CCE | 6  1 2d Convolution layer,  1 Max pool layer,  1 flatten layer,  3 fully connected layers. | relu, relu, softmax | 32(2DConvolution layer 1 channel),  Pooling layer (Max)  128,32,10(Fully connected layers) | 94.68% | 91.45% | 91.18% |
| 2 | CCE | 19  6 Convolution 2d layers,  6 Batch normalization layers,  3 Max pool layers,  1 Flatten layer,  3 Fully connected layers. | relu, relu, softmax | 64(2DConvolution layer 1 channel),  64(2DConvolution layer),  Batch Normalization layer,  128(2DConvolution layer),  128(2DConvolution layer),  256(2DConvolution layer),  256(2DConvolution layer),  Pooling layer (Max)  1024,512,10(Fully connected layers) | 100% | 93.48% | 93.28% |
| 3 | CCE |  |  |  |  |  |  |

Model 1:

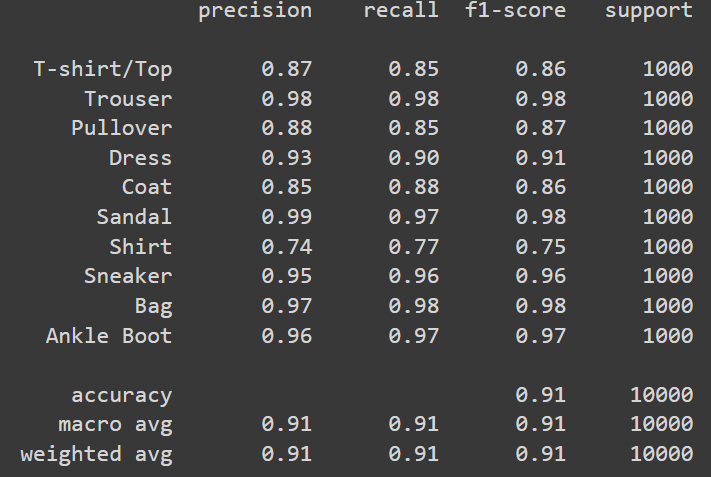


Here the first layer is a 2D convolution layer with 32 filters and 3x3 kernel that takes the 28x28 images in a single channel since images are not rgb. Next is the max pooling layer with a 2x2 pool window for downsampling the images and extracting features followed by a flattening layer that flattens the feature maps into a 1d array to input to the next 3 fully connected layers, 2 with relu and the last output layer with softmax activation. We have included a dropout of 0.4 after the first fully connected layer to improve validation and test performance.

Loss and accuracy vs Epochs



Classification report:

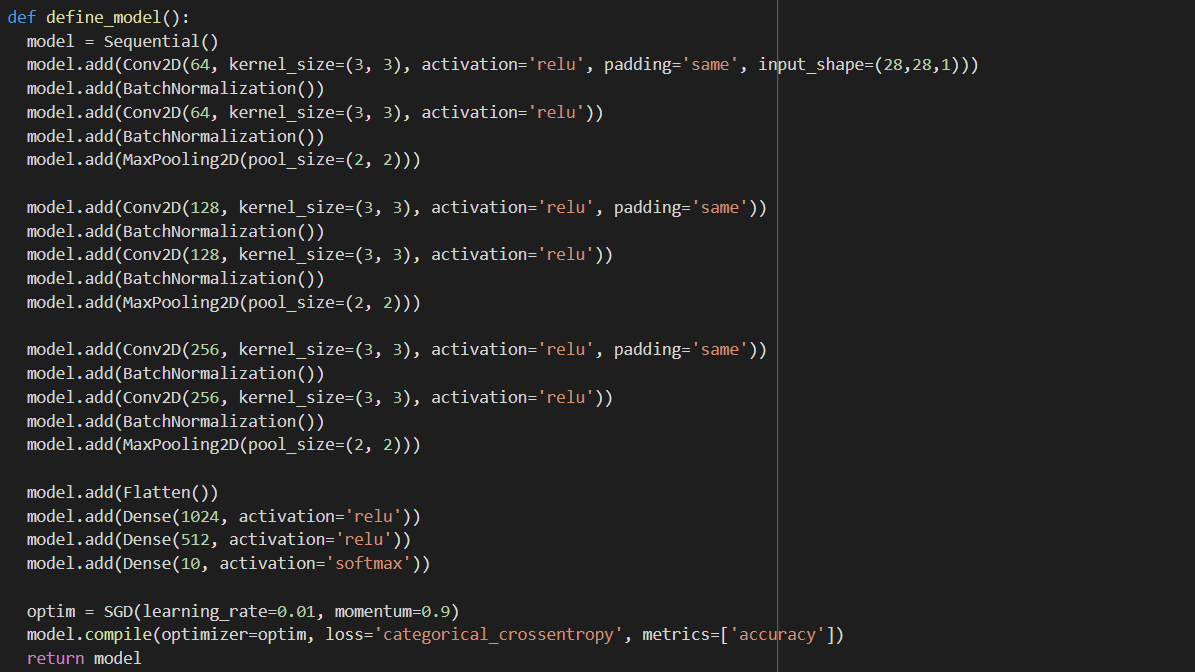


Confusion matrix:



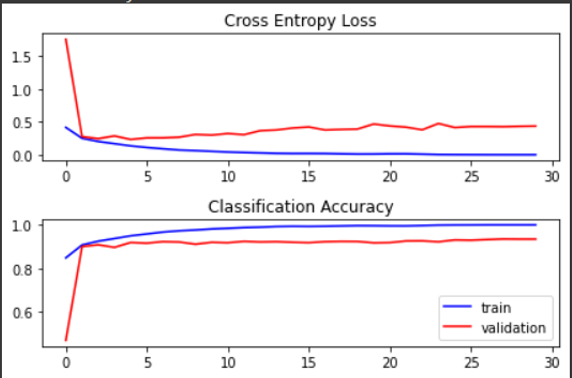
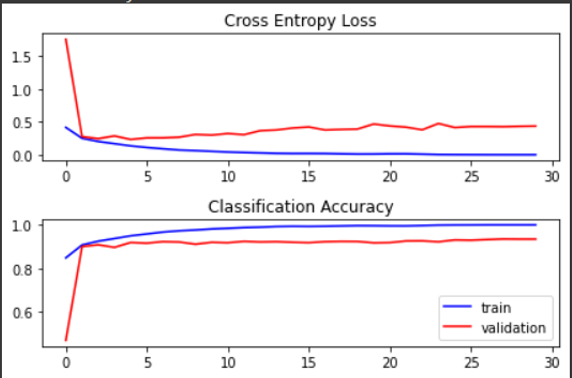
From the confusion matrix, we can see that a good number of t-shirt/top images, 119, got wrongly classified as shirts. Similarly 89 shirt images got classified as t-shirts/tops, 67 of them as coats and 44 as pullovers. The model did not perform the best when it came to t-shirt/top, shirt, coat and pullover labels.

Model 2:

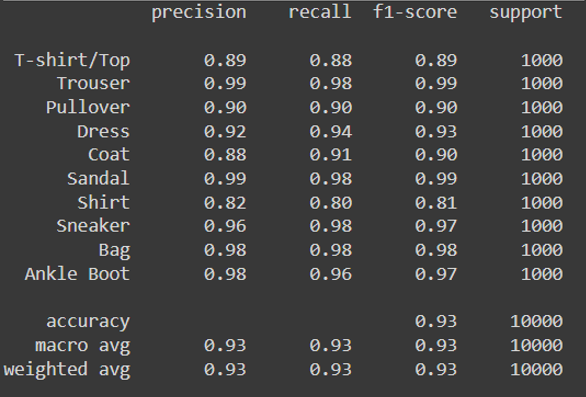


Here the first layer is a 2D convolution layer with 64 filters and 3x3 kernel that takes the 28x28 images in a single channel. Batch normalisation is a technique for training very deep neural networks that standardizes inputs to a layer. This stabilises the learning process and significantly reduces the number of training epochs needed to create deep networks. During training(fit()), it normalizes its output using the mean and standard deviation of the current batch of inputs, during inference(evaluate() or predict()), it normalizes its output using a moving average of the mean and standard deviation of the batches it has seen during training. Hence it is performed after each convolution layer. After passing through all the convolution layers next comes MaxPooling2D layer with a pool window of 2x2 for downsampling the images and extracting features followed by a flattening layer that flattens the feature maps into a 1d array to input to the next 3 fully connected layers. Except the last layer, two other layers use the ReLu activation function, and the output one has softmax activation and 10 neurons for the 10 classes.

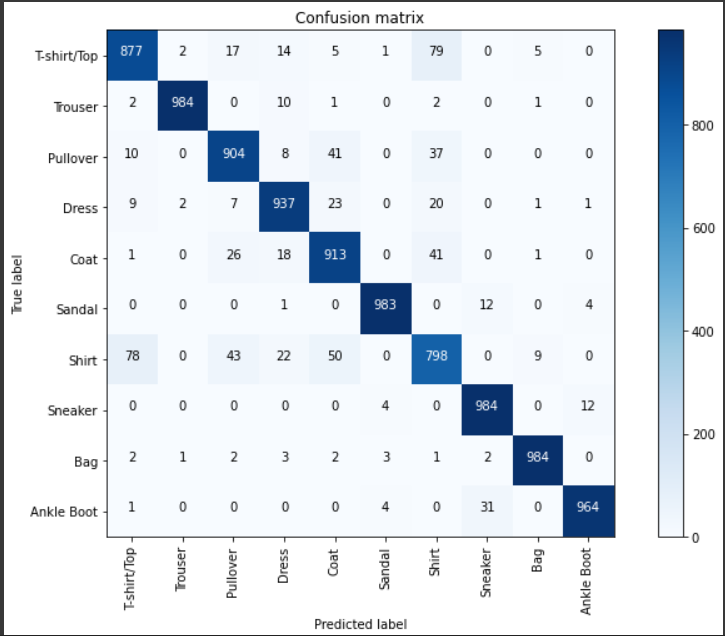
Loss and Accuracy vs Epochs



Classification Report:



Confusion matrix:



It can be observed that the labels that were mismatched(t-shirt/top,shirts; shirts,pullover,coats etc) have significantly decreased when compared to model 1, although the model still seems to get confused between them.

**Observation and Comparative Study Of Models**

By comparing both models, we can observe that adding convolution and pooling layers helps in extracting features better and batch normalization and dropout prevent overfitting the model to training data. Hence the second model performs better.