DEEP LEARNING ASSIGNMENT-2

Members:

1. Shika Rao 2019A3PS1237H
2. Akshat Agrawal 2019AAPS0264H
3. Vinita Bhat 2019A7PS1206H

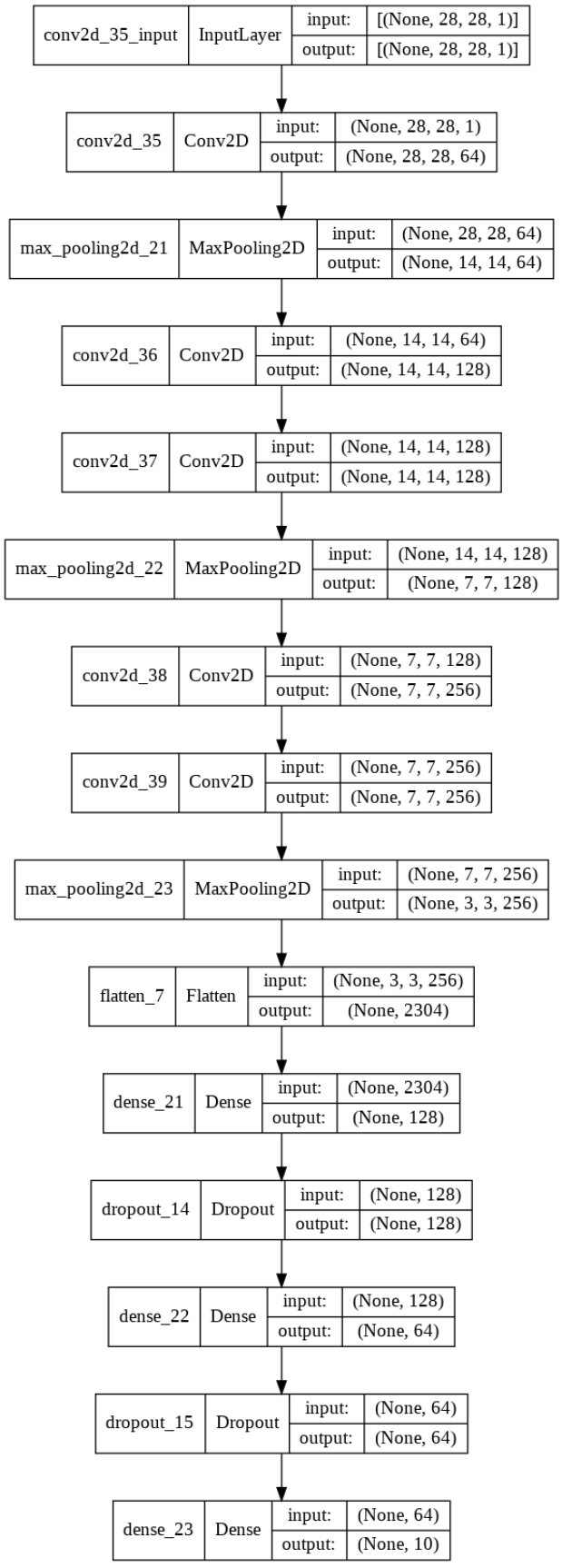
Aim: To build a convolutional neural network model to test on the Fashion MNIST dataset. The model is evaluated for understanding the effect of suitable hyperparameters on the performance of a model.

Date of Submission: 6/12/21

CONTENTS

Model - Details

1. 5 convolutional layers + 3 max pooling layers + 3 fully connected layers
2. Kernel size: 3x3
3. Number of filters: 64+128+128+256+256
4. Number of nodes: 128+64+10 nodes in total
5. Dropout used
6. Activation- ReLU
7. Loss- sparse categorical cross entropy
8. Time taken for 30 epochs- 1080s
9. Loss: `0.3684`
10. Accuracy: `90.52`


conv2d_35_input | InputLayer | input: [(None, 28, 28, 1)] | output: [(None, 28, 28, 1)]
conv2d_35 | Conv2D | input: (None, 28, 28, 1) | output: (None, 28, 28, 64)
max_pooling2d_21 | MaxPooling2D | input: (None, 28, 28, 64) | output: (None, 14, 14, 64)
conv2d_36 | Conv2D | input: (None, 14, 14, 64) | output: (None, 14, 14, 128)
conv2d_37 | Conv2D | input: (None, 14, 14, 128) | output: (None, 14, 14, 128)
max_pooling2d_22 | MaxPooling2D | input: (None, 14, 14, 128) | output: (None, 7, 7, 128)
conv2d_38 | Conv2D | input: (None, 7, 7, 128) | output: (None, 7, 7, 256)
conv2d_39 | Conv2D | input: (None, 7, 7, 256) | output: (None, 7, 7, 256)
max_pooling2d_23 | MaxPooling2D | input: (None, 7, 7, 256) | output: (None, 3, 3, 256)
flatten_7 | Flatten | input: (None, 3, 3, 256) | output: (None, 2304)
dense_21 | Dense | input: (None, 2304) | output: (None, 128)
dropout_14 | Dropout | input: (None, 128) | output: (None, 128)
dense_22 | Dense | input: (None, 128) | output: (None, 64)
dropout_15 | Dropout | input: (None, 64) | output: (None, 64)
dense_23 | Dense | input: (None, 64) | output: (None, 10)

Explanation of the choice of architecture

Convolution layers were used followed by max pooling layers. The max pooling layers are for extracting specific information in the image like edges. The pool size is taken as a 2x2 matrix. The flatten layer is needed for it to be in required output format.

The number of filters in the convolutional layer increases as the network gets deeper as the higher the number of filters, the more data the network is able to extract from the image. At the input layer, raw pixel data for the image is noisy thus the number of filters is ascending.

The number of nodes in the fully connected layer decreases as the network gets deeper.

3x3 kernel size is used for more parameters and info from the image. The first layer has a bigger kernel size 7x7 for more information extraction from the image.

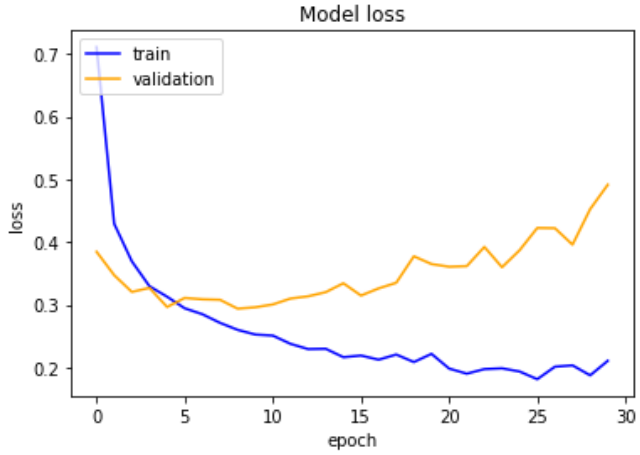
Dropout is used as it prevents overfitting.

1 max pooling layer is used after 2 convolutional layers as it is used to reduce the dimensions of the feature maps. Thus, the Pooling layer reduces the number of parameters to learn and reduces computation in the neural network.

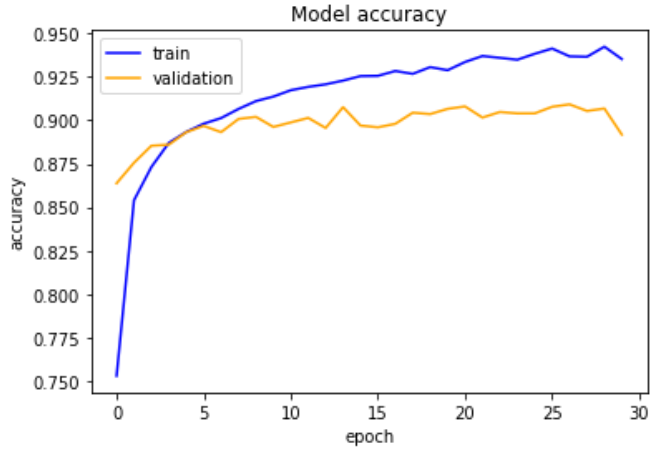
Softmax is used in the last layer to bring the output of the neural network to a probability distribution over predicting classes.

PLOTS

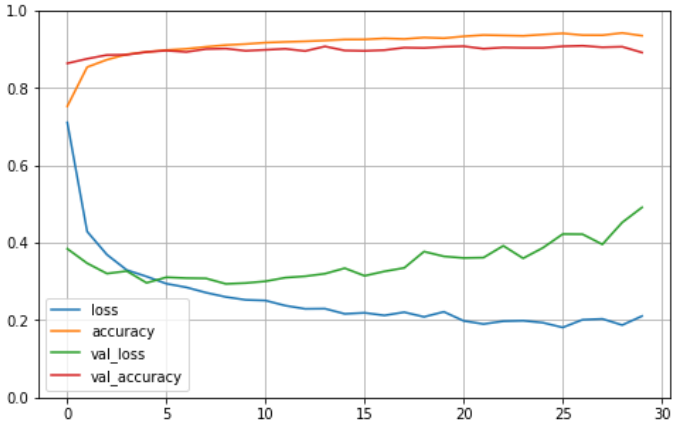
Loss Plot:



Accuracy plot:



Overall plot:



Confusion Matrix:

```
array([[[8813,  187],
        [ 133,  867]],

       [[8997,    3],
        [   7,  993]],

       [[8884,  116],
        [ 132,  868]],

       [[8856,  144],
        [  60,  940]],

       [[8860,  140],
        [ 145,  855]],

       [[8993,    7],
        [  69,  931]],

       [[8803,  197],
        [ 305,  695]],

       [[8910,   90],
        [  29,  971]],

       [[8977,   23],
        [  30,  970]],

       [[8959,   41],
        [  38,  962]]])
```

Classification Report:

	precision	recall	f1-score	support
T-shirt/top	0.82	0.87	0.84	1000
Trouser	1.00	0.99	0.99	1000
Pullover	0.88	0.87	0.87	1000
Dress	0.87	0.94	0.90	1000
Coat	0.86	0.85	0.86	1000
Sandal	0.99	0.93	0.96	1000
Shirt	0.78	0.69	0.73	1000
Sneaker	0.92	0.97	0.94	1000
Bag	0.98	0.97	0.97	1000
Ankle boot	0.96	0.96	0.96	1000
accuracy			0.91	10000
macro avg	0.91	0.91	0.90	10000
weighted avg	0.91	0.91	0.90	10000