

ML Assignment 2

Submitted By:

Name: Chetan Pant

SR no : 18071

Question 1:

In this question, We build a convolutional neural network to classify for Fashion-MNIST dataset into one of the ten classes. For this we use Pytorch .

Procedure:

1. Processing the dataset
2. Defining CNN model
3. Tuning the Hyperparameter
4. Training the model
5. Testing the Model.

1. Processing the dataset:

We load the Fashion-MNIST data set using `torchvision.datasets`. We divide the training set into training and validation set.

2. Defining the Model:

Our model class contains two convolutional layers and a maxpool layer with every convolutional layer. We use batch normalisation and ReLU activation in convolutional layers.

After the convolutional layer, the model contains a deep neural network with one hidden layer and one output layer. For Final output layer we use Softmax function. Forward function of the class takes the input and gives the output of the model which is the probability of the classes given the image.

Architecture:

```
CNN(  
  (layer1): Sequential(  
    (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ReLU()  
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
  )  
  (layer2): Sequential(  
    (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ReLU()  
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
  )  
  (layer3): Sequential(  
    (0): Linear(in_features=1568, out_features=256, bias=True)  
    (1): ReLU()  
    (2): Linear(in_features=256, out_features=10, bias=True)  
    (3): Softmax(dim=None)  
  )  
)
```

3. Hyperparameter tuning:

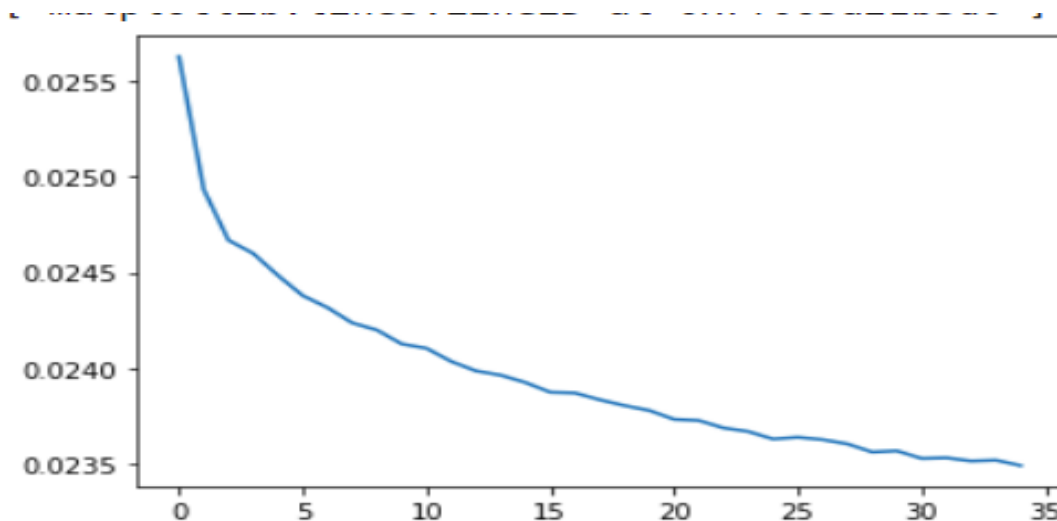
We tune the hyperparameter(learning rate and Batch size) by evaluating the performance of our model on the validation set.After this we find the hyperparameter which gives the best performance on validation set and used these parameter for final training

Optimal Learning rate=0.001 and BatchSize=64

4.Training the Model:

We train our model for 35 epochs on the training set and find the loss during training per epoch and also the loss and accuracy on the validation set.After every 5 epoch we save the model.We also save the final model.

We plot the graph between training loss vs epoch.



5. Testing the model and Results:

For testing ,we first load the final model using torch.load(). After this we evaluate the model in the training set.

Accuracy on the training set: **96.344%**

Accuracy on the testing set: **91.29%**

Confusion Matrix:

```
[[896  1  27  17  1  0 106  0  2  1]
 [  0 979  1  4  1  0  1  0  0  0]
 [  9  0 839  7 40  0  56  0  3  0]
 [ 12 15  9 901 18  0  30  0 11  0]
 [  3  1 64  34 907  0  66  0  3  0]
 [  1  0  0  0  0 975  0  8  1  6]
 [ 72  2 60  37 32  0 736  0  4  0]
 [  1  0  0  0  0 13  0 946  2 18]
 [  5  2  0  0  1  1  5  2 974  2]
 [  1  0  0  0  0 11  0 44  0 973]]
```