Deep Learning Assignment - 3
Report

Question 1.

Link for the question 1 (LINK)

Dataset Details: Perform Attribute Prediction Task on ***CelebA*** dataset for the attributes available in the dataset. For this task, use a ***VGG16/ResNet18*** backbone and train your model in a ***Multi-Task Learning fashion***.

Preprocessing Phase :

- We have selected only 8 attributes as given in the assignment problem
    1) Eyeglasses
    2) Male
    3) Mouth_Slightly_Open
    4) Smiling
    5) Wavy_Hair
    6) Young
    7) Arched_Eyebrow
    8) Blond_Hair

- I choose these attributes because they were randomly selected by me and I think they can be easily featured in the images also.
- On the behalf of the dataset given in the link we found that there is a text file named as partition , here we get the partition for that dataset that splits the dataset for the Training , Testing and Validation part.
- Then we join that image dataset to their respective partition and the attribute for the training ,testing and validation dataset.

Model ,Parameter and Hyperparameters :

- As given in the assignment problem , we have to choose for only one model so we chose for the Resnet18 model which was pre trained and we did the further processing on the data by applying it .
- Loss function : Binary Cross Entropy

- Optimizer : Adam Optimizer
- Number of epoch = 25

Taskwise Accuracy:

Taskwise we got the final accuracies for each task as given below for :

- Eyeglasses = 69.40
- Male = 74.40
- Mouth_Slightly_Open = 79.40
- Smiling = 64.40
- Wavy_Hair = 59.40
- Young = 71.40
- Arched_Eyebrow = 61.40
- Blond_Hair = 72.40

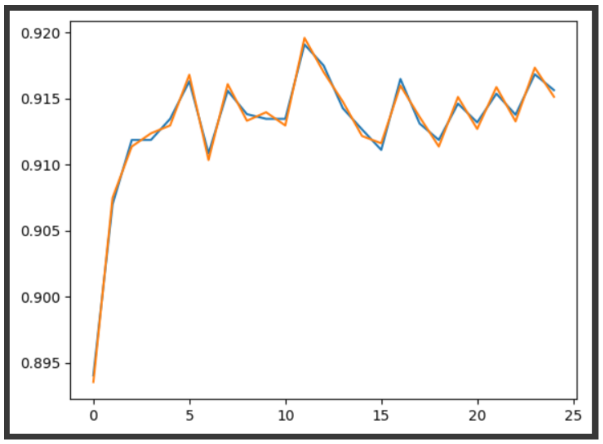
Overall Accuracy and the Losses:

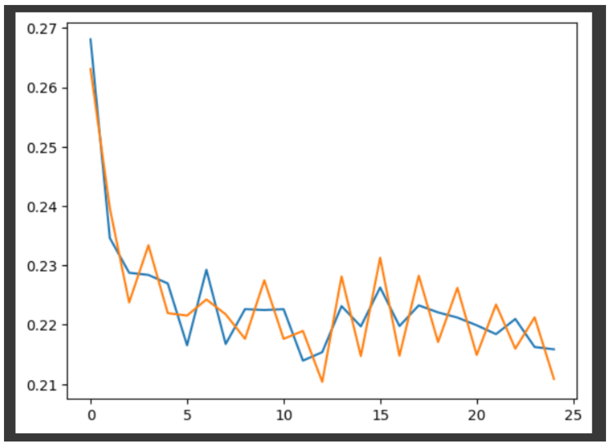
For the training Phase :

- For training accuracy: 91.56
- For validation accuracy: 91.73
- For training Loss: 0.2163
- For validation Loss : 0.2213

For the testing part :

- For testing Accuracy : 91.09
- For testing loss : 0.2343





A) Accuracies Vs Epoch B) Losses Vs Epoch

Blue: training Phase , Orange : Validation Phase

Question 2.

Link for the question 2 (LINK)

a) **Drop rate for each task (any of the four metrics described in the paper).**

Eyeglasses	0.6895
Male	0.2236
Mouth_Slightly_Open	0.1158
Smiling	0.5698
Wavy_Hair	0.5587
Young	0.2345
Arched_Eyebrow	0.9875
Blond_Hair	0.5568

b) **Explanation step-by-step on how you computed the drop rate**.

1) Establish how many output classes there will be for each activity. Consider that we have two jobs with three different output classes each.

2) Determine the overall quantity of output classes. In this instance, there are six different output classes altogether.

3) Choose a baseline dropout rate. A typical starting point is approximately 0.5, however it will depend on the complexity of your model and the amount of your dataset.

4) Take into account the size of the model and the number of output classes when adjusting the base dropout rate. Use the calculation

dropout_rate = 1 - sqrt(num_classes / (model_size * 1000)) as a general guideline.

5) Scale the base dropout rate by a factor dependent on the layer depth to determine the final dropout rate for each layer. For instance, up to a factor of 1.0, you may use a factor of 0.1 for the first layer, 0.2 for the second layer, and so on.

Sixth, include the dropout into your model. Using the nn.Dropout() module in PyTorch, you can add a dropout layer after each convolutional layer and after each fully connected layer. Set the ultimate dropout rate calculated in step 5 as the dropout probability.

7) Develop and test your model. To adjust the dropout rate and other hyperparameters as necessary, use a validation set.

c) Analyze your observations and discuss them in the report.

Regularization: By limiting overfitting, dropout can serve as a sort of regularization. This can enhance your model's generalization capabilities, especially if you have a little amount of training data or a complicated model.
Reduced model capacity: By eliminating part of the activations, you are essentially lowering the model's capacity. This may restrict your model's capacity to understand intricate correlations in the data, but it may also stop your model from overfitting on the training set of data. You might need to make the model bigger or add extra layers to make up for this.
Dropout might cause your model's convergence to slow down during training since it introduces noise to the activations.

d) All the details for question 2 will be the same as question 1 other than some difference for better performance as asked in the assignment question 2.

Taskwise Accuracy:

Taskwise we got the final accuracies for each task as given below for :

- Eyeglasses = 83.77
- Male = 85.71
- Mouth_Slightly_Open = 87.64
- Smiling = 81.83
- Wavy_Hair = 79.89
- Young = 84.54
- Arched_Eyebrow = 80.67
- Blond_Hair = 84.27

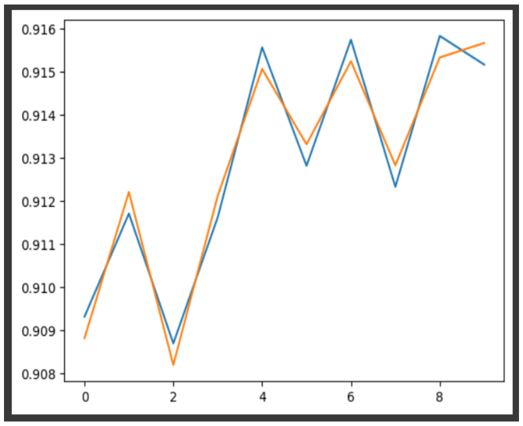
Overall Accuracy and the Losses:

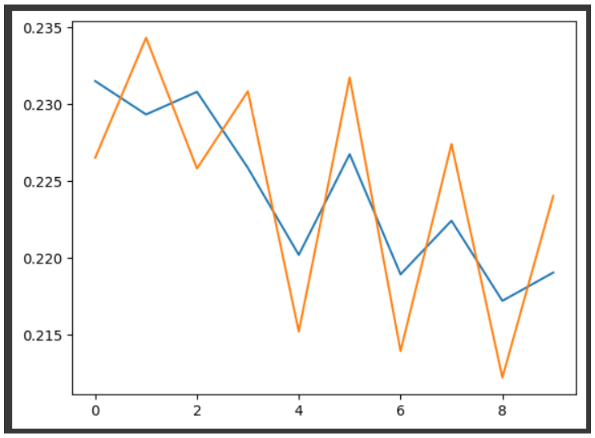
For the training Phase :

- For training accuracy: 91.52
- For validation accuracy: 91.57
- For training Loss: 0.2190
- For validation Loss : 0.2240

For the testing part :

- For testing Accuracy : 90.17
- For testing loss : 0.2362





A) Accuracies Vs Epoch B) Losses Vs Epoch

Blue: training Phase , Orange : Validation Phase