Name(s) : Adel Alkhamisy
Kaggle Team Name:aalkhami

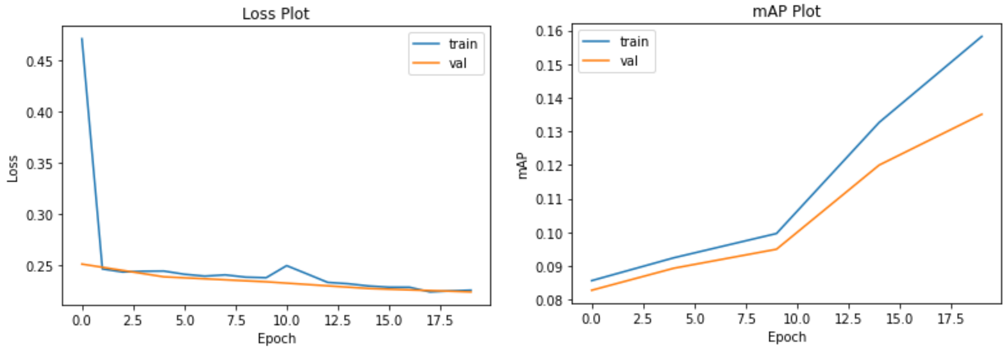
Part1.1: Pre-designed network for multi-label classification

In this part, you will practice to train a neural network both by training from scratch or fine-tuning.

CS747_Assignment2_part1.ipynb should provide you with enough instruction to start with.

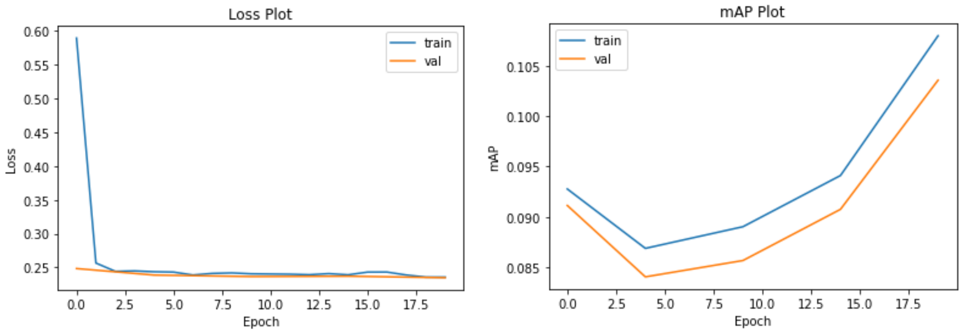
We are asking you to provide the following results.

1. Simple classifier
    a. Report test mAP for simple classifier: 0.1321
    b. Visualize loss and mAP plots:

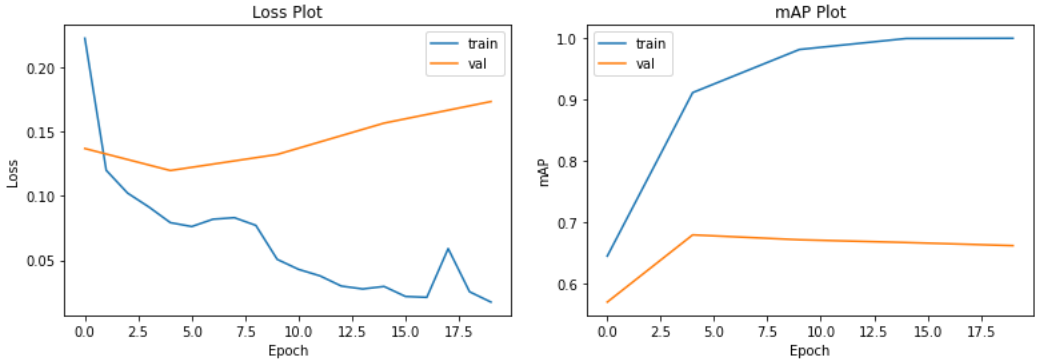


    c. Provide analysis of the plots (at least 3 sentences):
    The loss function on the training dataset and validation over the 20 training epochs are displayed in the first plot. Over the first few epochs, the loss is observed to be decreasing. After that, it reaches a plateau and varies a little bit during the remaining epochs. This suggests that the model is converging, however it could improve through additional training iterations. The mAP curves display the classifier's performance on the train and validation sets for each training session. Although the mAP is initially not very high, it rises quickly. effectively learning the features of the training data and generalizing well to the validation set This shows that the model is effectively learning the features of the training data and generalizing well to the validation set. Overall, the mAP is relatively low, which suggests that there is still room for improvement.

2. AlexNet from Scratch
    a. Report test mAP for alexnet: 0.0984
    b. Visualize loss and mAP plots:



3. Pretrained AlexNet
   a. Report test mAP for pretrained alexnet: 0.6817
   b. Visualize loss and mAP plots:



   c. Provide analysis on differences to training from scratch (at least 3 sentences):

The results show that the pre-trained AlexNet outperforms the AlexNet trained from scratch. The pre-trained model achieved a mean average precision (mAP) of 0.6795 on the validation set after 5 epochs, while the scratch model achieved an mAP of 0.1036 on the validation set after 20 epochs. Additionally, the average loss of the pre-trained model was lower than that of the scratch model, indicating better model performance.

This difference can be attributed to the fact that the AlexNet Pretrained model was initialized with weights that were learned from a large dataset (ImageNet), whereas the AlexNet Trained from Scratch model was initialized randomly. The pretraining allows the model to learn useful features that can be applied to the new dataset (PASCAL VOC), making it easier for the model to generalize to new examples.

In summary, the results suggest that using pretrained models can be a more effective approach than training from scratch, especially when working with limited amounts of data.

Part1.2: Self designed network for multi-label classification

CS747_Assignment2_part2.ipynb should provide you with enough instruction to start with. You upload your output of your self-designed network to kaggle.

Did you upload final CSV file on Kaggle: **Yes**

1. My best mAP on Kaggle: 0.81
2. Factors which helped improve my model
    a. Added two layers.

The architecture of the network is as follows:

1. Convolutional layer with 3 input channels and 64 output channels, kernel size 5x5
2. ReLU activation layer
3. Max pooling layer with kernel size 2x2
4. Convolutional layer with 64 input channels and 32 output channels, kernel size 3x3
5. ReLU activation layer
6. Max pooling layer with kernel size 2x2
7. Convolutional layer with 32 input channels and 16 output channels, kernel size 3x3
8. ReLU activation layer
9. Max pooling layer with kernel size 2x2
10. Flatten layer
11. Fully connected layer with 16 * 26 * 26 input neurons and 120 output neurons
12. ReLU activation layer
13. Fully connected layer with 120 input neurons and 84 output neurons
14. ReLU activation layer
15. Fully connected layer with 84 input neurons and 64 output neurons
16. ReLU activation layer (added layer)
17. Fully connected layer with 64 input neurons and 32 output neurons (added layer)
18. Output layer with 32 input neurons and NUM_CLASSES output neurons (added layer)

The initial network provided to you can be considered as the BaseNet. A very important part of deep learning is understanding the ablation studies of various networks. So we would like you to do a few experiments. Note, this **doesn't need to be very exhaustive** and can be in a cumulative manner in an order you might prefer. Fill in the following table :

Serial #	**Model architecture**	**Best mAP on test set**
1	BaseNet	0.17
2	BaseNet + fc4 = nn.Linear(64, 32) + self.fc5 = nn.Linear(32, NUM_CLASSES)	0.1853

3	BaseNet + fc4 = nn.Linear(64, 32) + self.fc5 = nn.Linear(32, NUM_CLASSES) + Adam + lr=0.001	0.16
4	BaseNet+ fc4 +fc5 + fc6 + fc7	0.12