DL Assignment 4
Report

Neemesh Yadav (2020529)
Mohammad Aflah Khan (2020082)

TASK I (Unimodal: Image-Only)

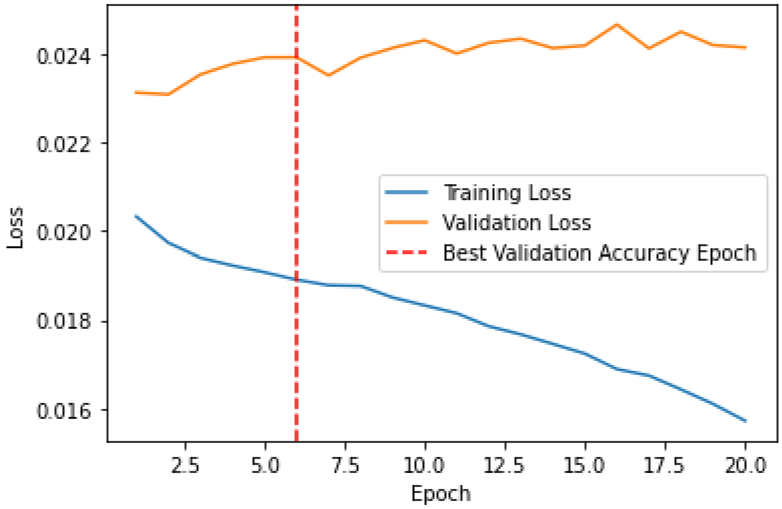
Preprocessing -

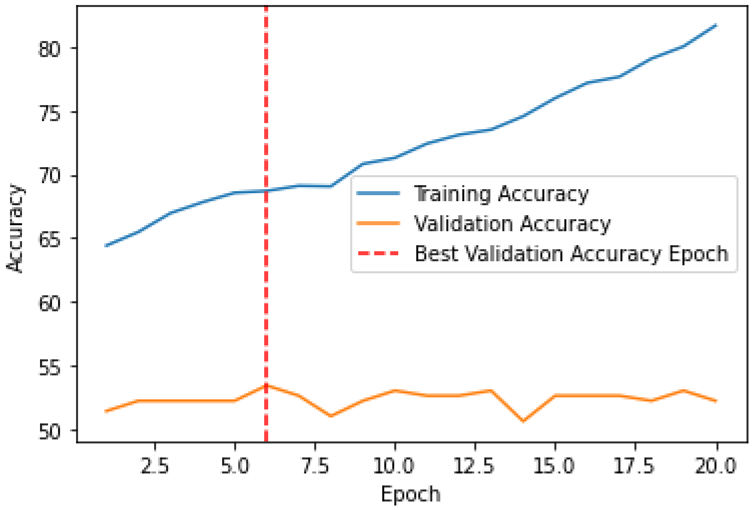
- Resize to 224 x 224
- Normalizes using the Mean and Std Dev from Imagenet (Since we're using Pretrained ResNet-50)

Model -

We first produce embeddings using the outputs of the pen-ultimate layer of the ResNet50 Model. These 2048 dimensional embeddings are then used to represent the images as input to a MLP Classifier. We use a 3 layer MLP Classifier with Relu activation in intermediate layers and Softmax activation in the output layer.

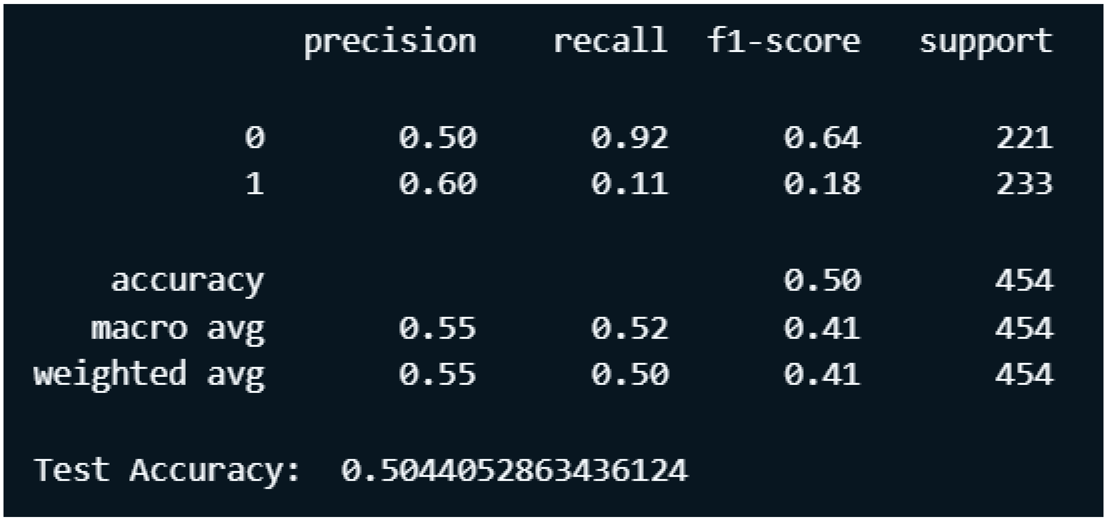
We also use checkpointing to later pick the checkpoint with best validation accuracy to choose the checkpoints before the model starts to overfit





We find a decently early peak and hence we use that checkpoint. We notice that the change in val loss is very less and in all tasks the val loss and val accuracy sort of stabilize post the first few epochs. This might be because 2-3 epochs are sufficient to learn a good classification boundary or an analogue to it in the vector space with the given representations which are far from being the best due to the complexity of the task.

Test Metrics -



```
              precision    recall  f1-score   support

           0       0.50      0.92      0.64       221
           1       0.60      0.11      0.18       233

    accuracy                           0.50       454
   macro avg       0.55      0.52      0.41       454
weighted avg       0.55      0.50      0.41       454

Test Accuracy:  0.5044052863436124
```

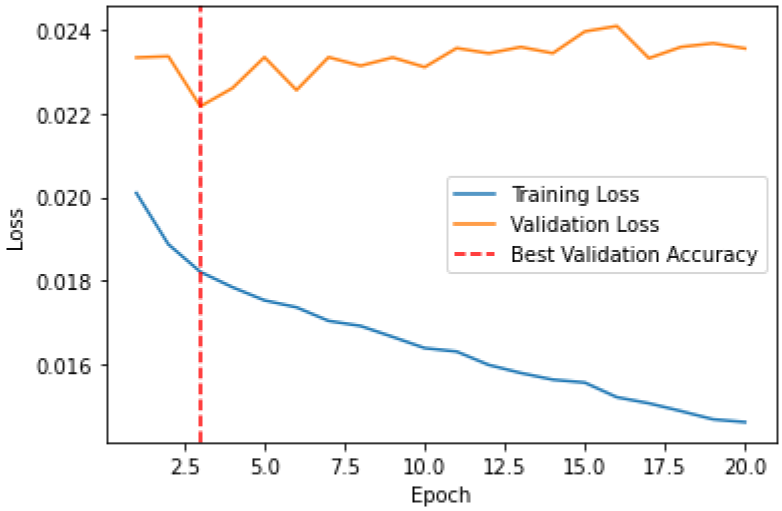
TASK II (Unimodal: Text-Only)

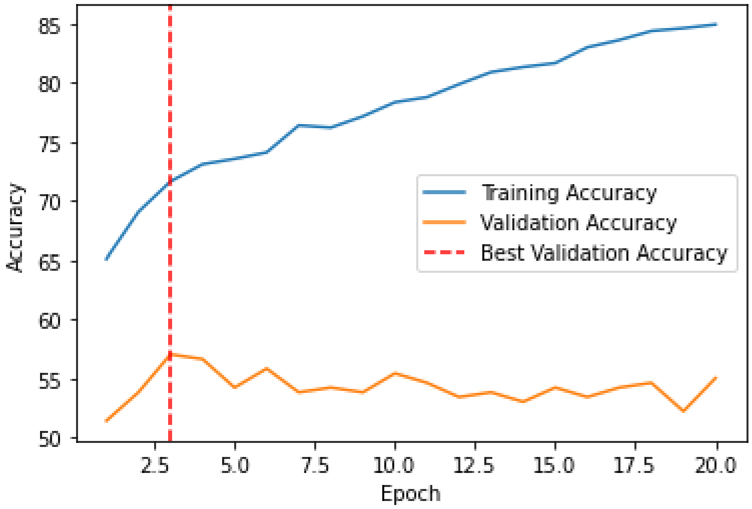
Preprocessing -

We find that the text is already in good shape without much noise so we only do some minor preprocessing. We simply remove extra white spaces and also lowercase the text. We decide not to remove punctuations as it would lead to loss of valuable information.

Model -

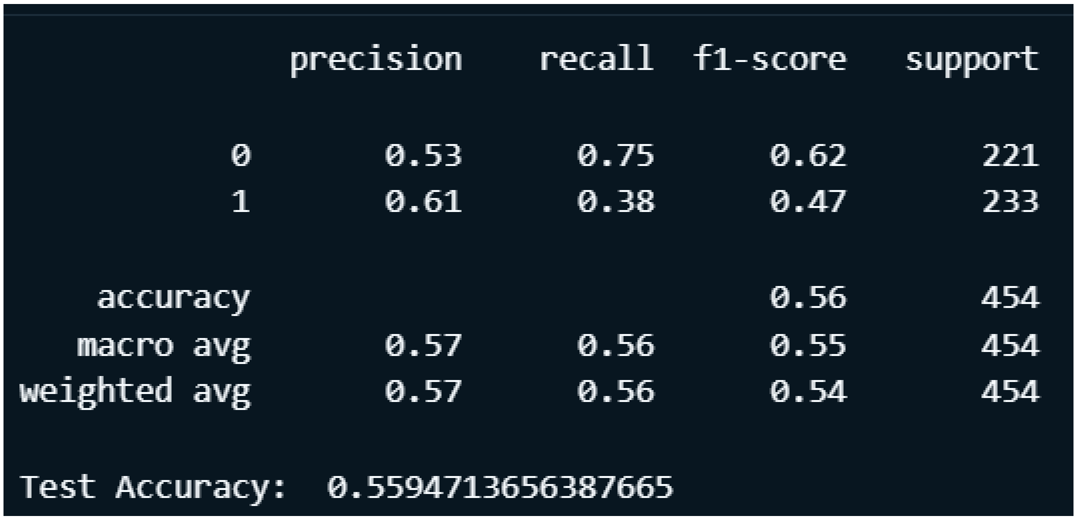
We first produce 768 dimensional embeddings of the sentences using “bert-base-uncased” from HuggingFace. We use these embeddings as input to a MLP Classifier which has the same architecture as the one used in Task I to make the comparisons fair. We also perform similar checkpointing.





We find an early peak and hence we use that checkpoint. We notice that the change in val loss is very less and in all tasks the val loss and val accuracy sort of stabilize post the first few epochs. This might be because 2-3 epochs are sufficient to learn a good classification boundary or an analogue to it in the vector space with the given representations which are far from being the best due to the complexity of the task.

Test metrics -



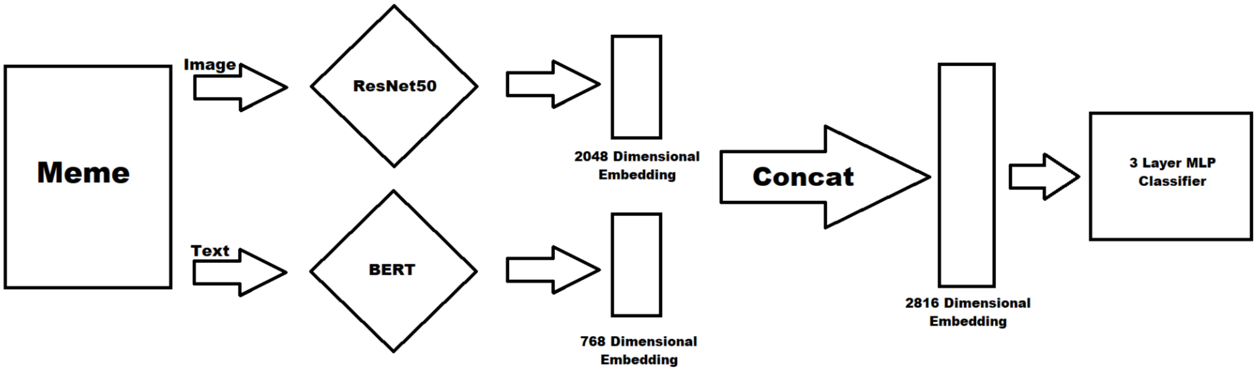
```
              precision    recall  f1-score   support

           0       0.53      0.75      0.62       221
           1       0.61      0.38      0.47       233

    accuracy                           0.56       454
   macro avg       0.57      0.56      0.55       454
weighted avg       0.57      0.56      0.54       454

Test Accuracy:  0.5594713656387665
```

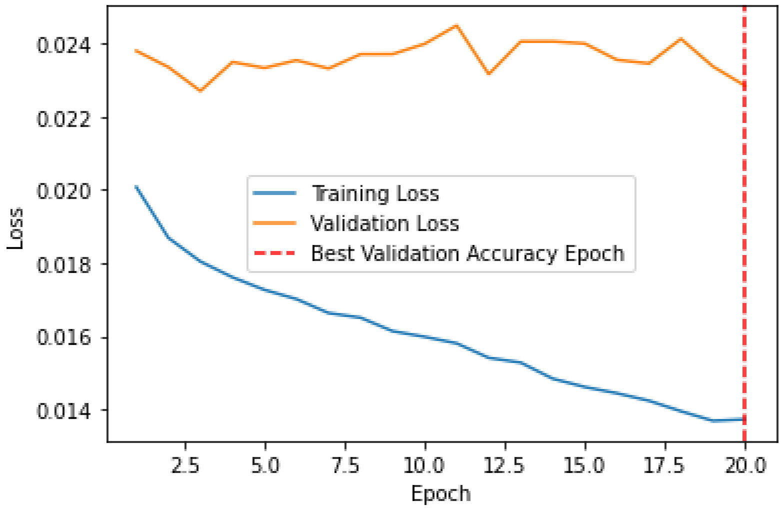
TASK III (Multimodal: Image+ Text-Based Classification)

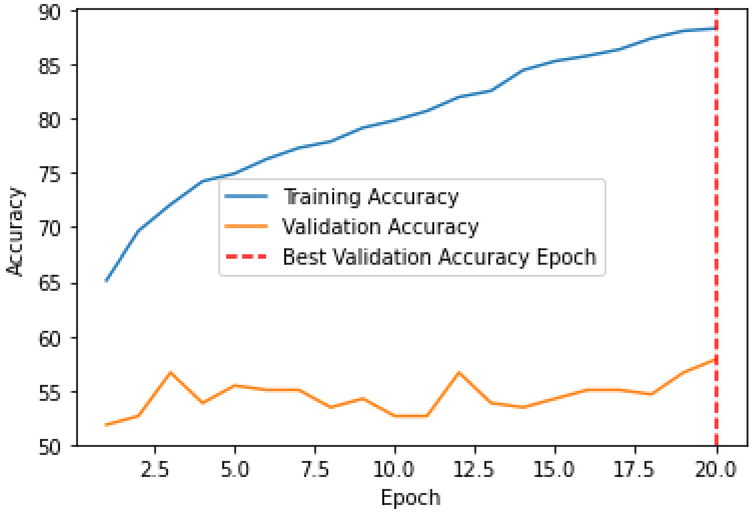
Preprocessing -
We use the same embeddings produced in the previous 2 parts hence we use the same preprocessing steps here

Our base Image Model is the ResNet50 model to produce image embeddings
Out base Text Model is bert-base-uncased model to produce text embeddings

We combine the 2 using concatenation to produce a new vector representation which captures both lexical as well as visual nuances of the meme.

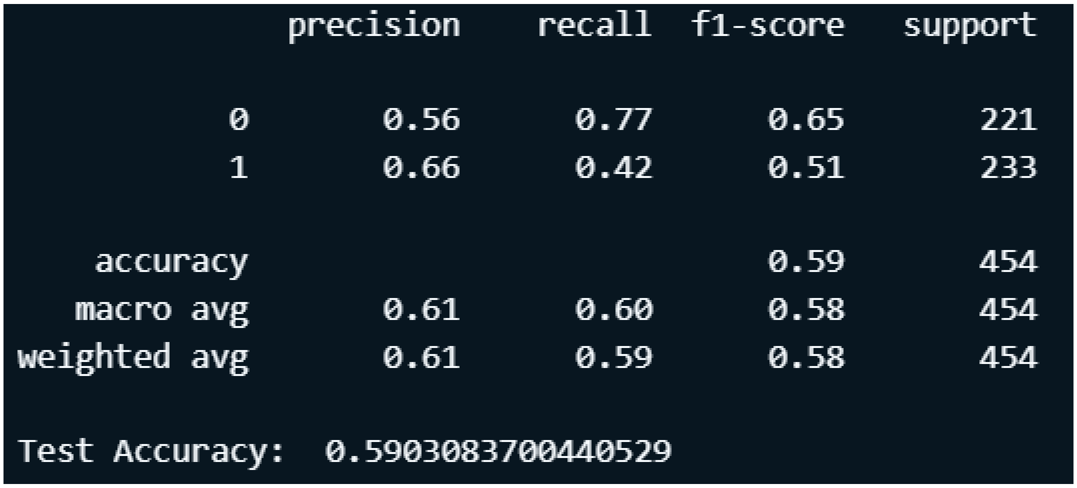

Meme
Image
Text
ResNet50
BERT
2048 Dimensional Embedding
768 Dimensional Embedding
Concat
2816 Dimensional Embedding
3 Layer MLP Classifier





We notice that the change in val loss is very less and in all tasks the val loss and val accuracy sort of stabilize post the first few epochs. This might be because 2-3 epochs are sufficient to learn a good classification boundary or an analogue to it in the vector space with the given representations which are far from being the best due to the complexity of the task. The best epoch being the last one doesn’t really matter as we see the difference in validation metrics is very less and hence it can be attributed to randomness.

Test Metrics -



```
              precision    recall  f1-score   support

           0       0.56      0.77      0.65       221
           1       0.66      0.42      0.51       233

    accuracy                           0.59       454
   macro avg       0.61      0.60      0.58       454
weighted avg       0.61      0.59      0.58       454

Test Accuracy:  0.5903083700440529
```

Our model outperforms the base models by a large margin in both Accuracy and F1 score.

TSNE Visualizations:

We sample 50 random samples from both classes for this visualization.

Combined Model's Penultimate Layer Outputs -

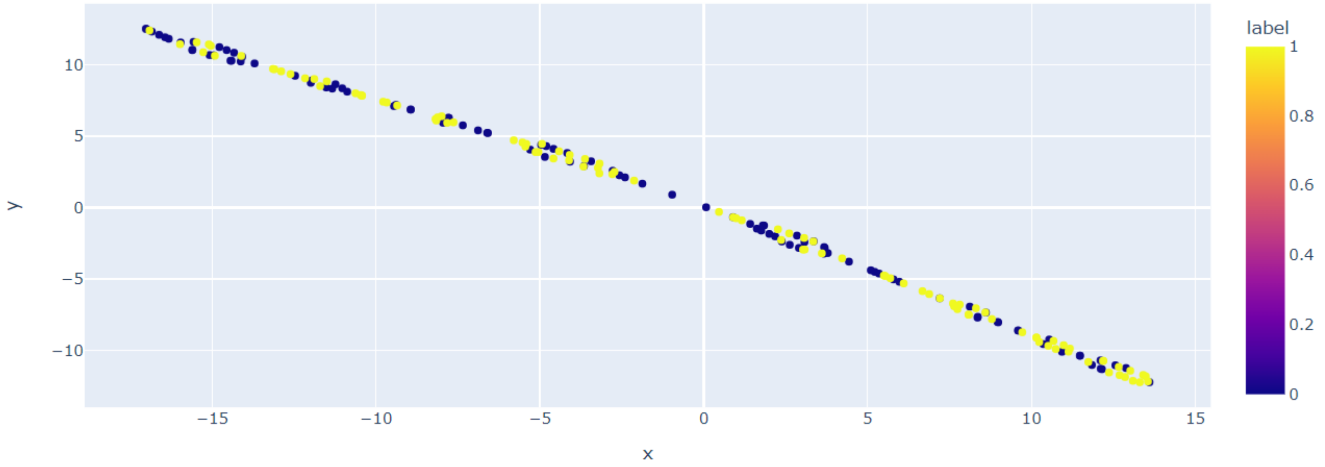
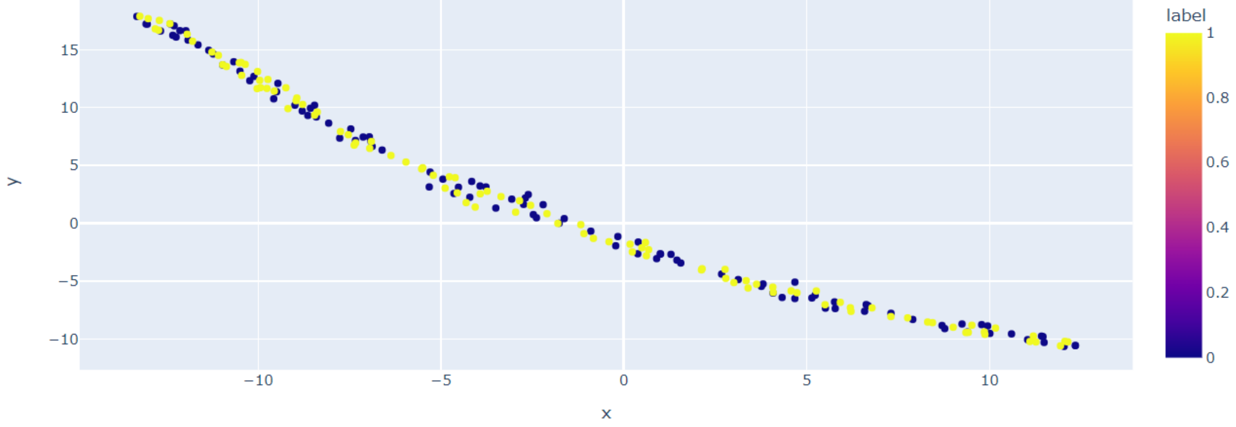
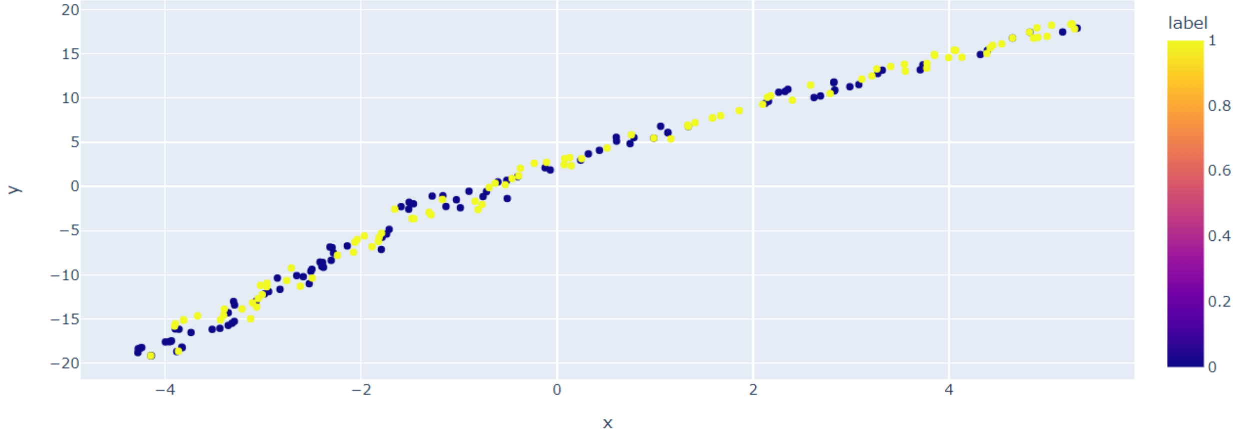


Image Only Model's Penultimate Layer Outputs -



Text Only Model’s Penultimate Layer Outputs -



The plots seem very similar except from the fact that the combined model’s plot and image only model’s plot start high and decrease while the text only model’s plot are the reverse. There also is clearly high discriminability in the text only model’s plot as compared to the image model plot as we see more blue patches without yellow. Similarly we find such class patches in the first plot which might be an indicator for why the model performs better. Ofcourse TSNE Visualization loses information and maybe in a more higher dimensional space the segregation is more clearly visible.