ML 2 Programming Assignment 3 Report

- Submitted by : Anurag Saraswat (M20CS066)

Q1. Bidirectional LSTM Network without Attention

Data Set : IMDB dataset id used having 50K movie reviews for natural language processing or Text analytics.This is a dataset for binary sentiment classification.

Model : Bidirectional LSTM is used for classification. Detailed information of model is as follow:

```
RNN(
  (embedding): Embedding(25002, 100, padding_idx=1)
  (rnn): LSTM(100, 256, num_layers=2, dropout=0.5, bidirectional=True)
  (fc): Linear(in_features=512, out_features=1, bias=True)
  (dropout): Dropout(p=0.5, inplace=False)
)
```

Hyper Parameters:

N_LAYERS = 2

DROPOUT = 0.5

EMBEDDING_DIM = 100

HIDDEN_DIM = 256

BATCH_SIZE = 64

MAX_VOCAB_SIZE = 25_000

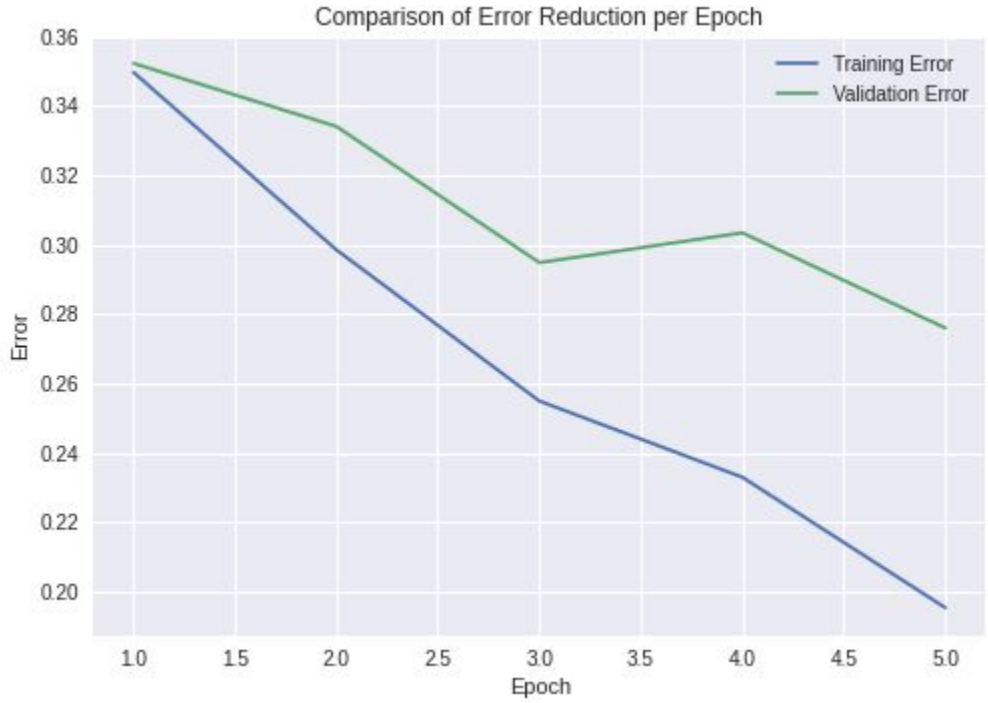
glove.6B.100d is used for calculating vectors

Observation:

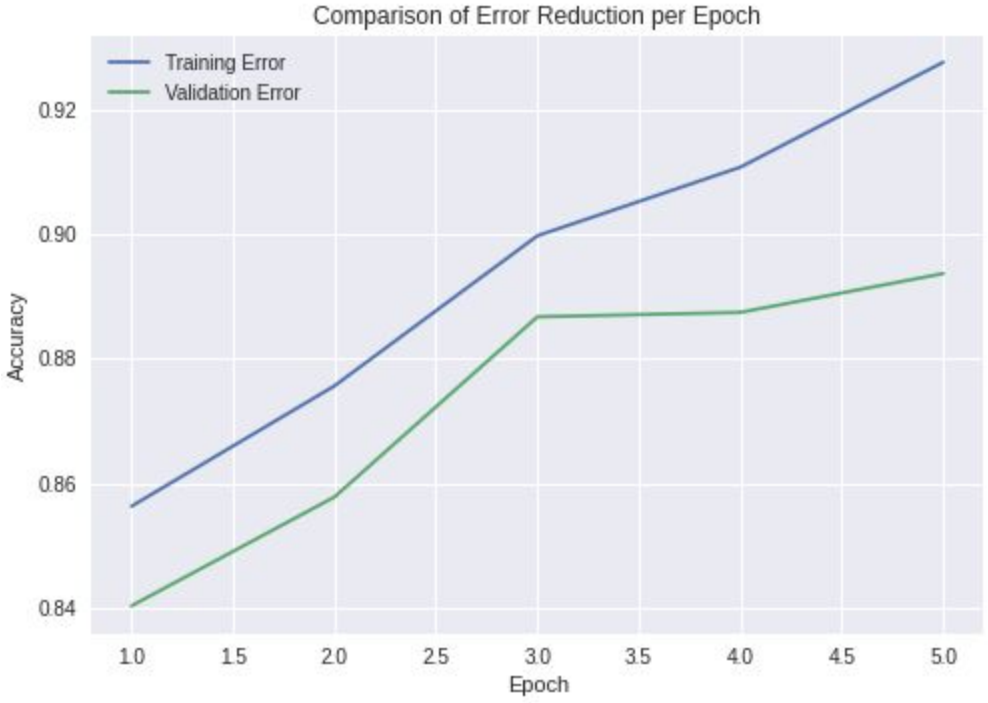
Model is trained for 5 Epoch and following result are observed:

- `Test Loss: 0.284 | Test Accuracy: 88.83%`

Following curve is obtained for reduction in error per epoch



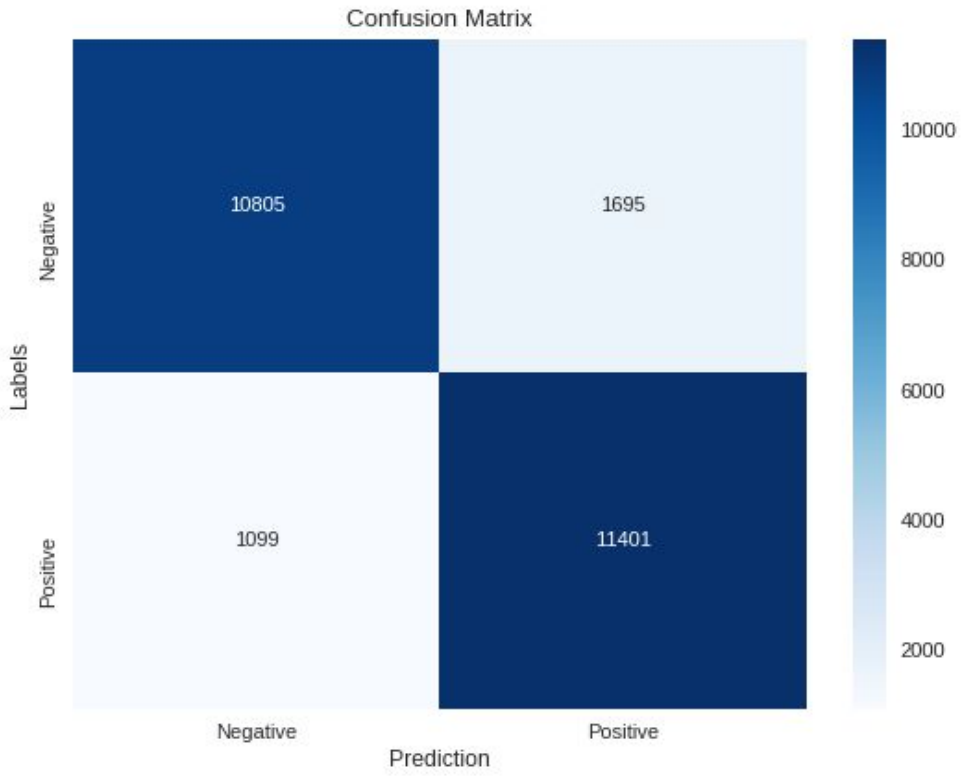
Following curve is obtained for accuracy achieved per epoch



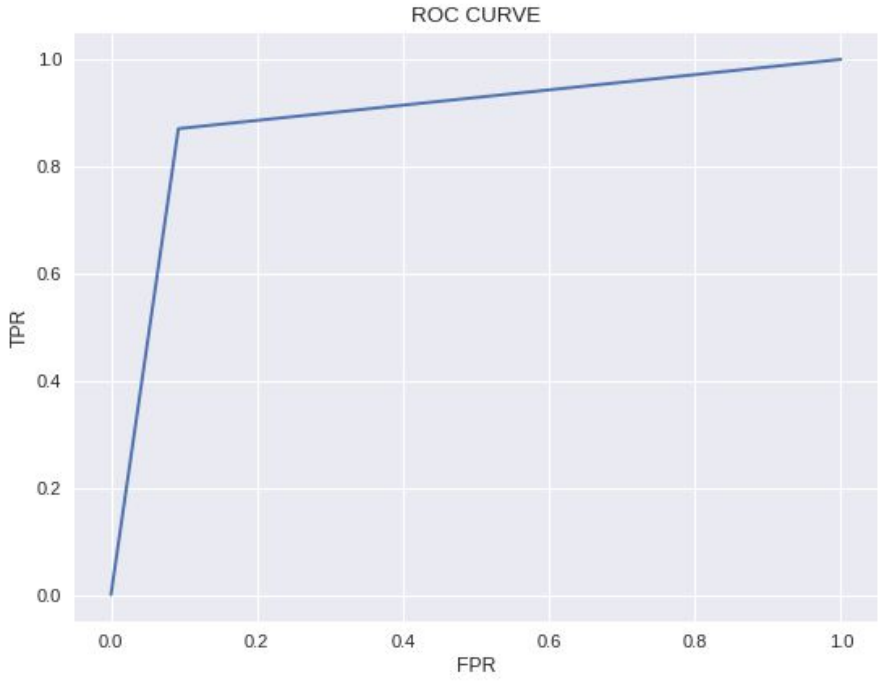
Confusion matrix obtained is

```
-------Confusion Matrix-------
         10805   1695
         1099    11401
```

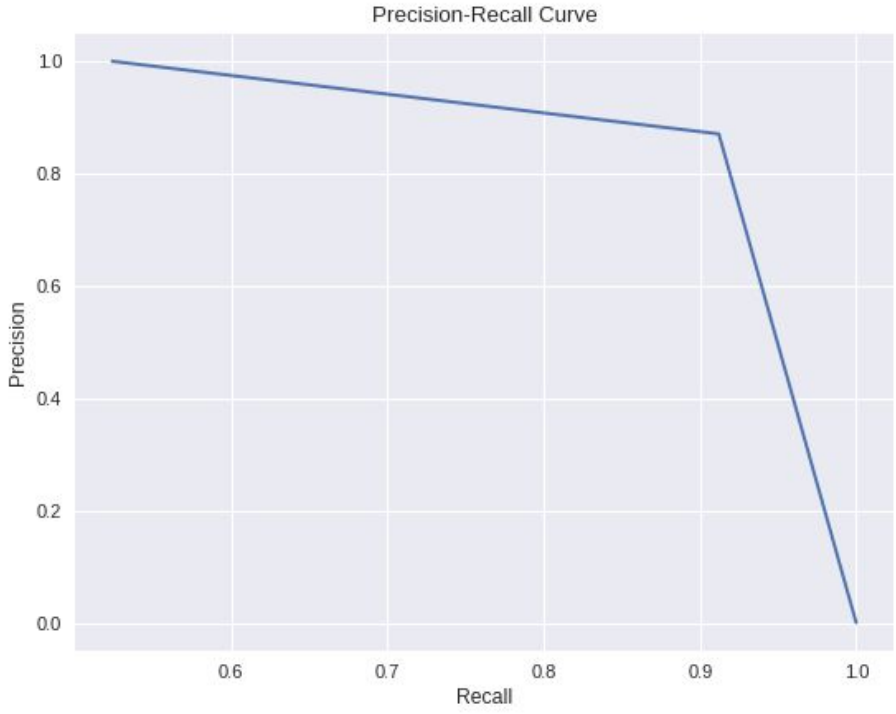
The Heat Graph of confusion matrix is as follow:



The **ROC Curve** is as follow:



Precision-Recall Curve is as follow



Following observation is made from ROC Curve and Precision-Recall Curve

- ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.
- Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.
- ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are appropriate for imbalanced datasets.

Q2. Finetune Densenet 121.

Model Information:

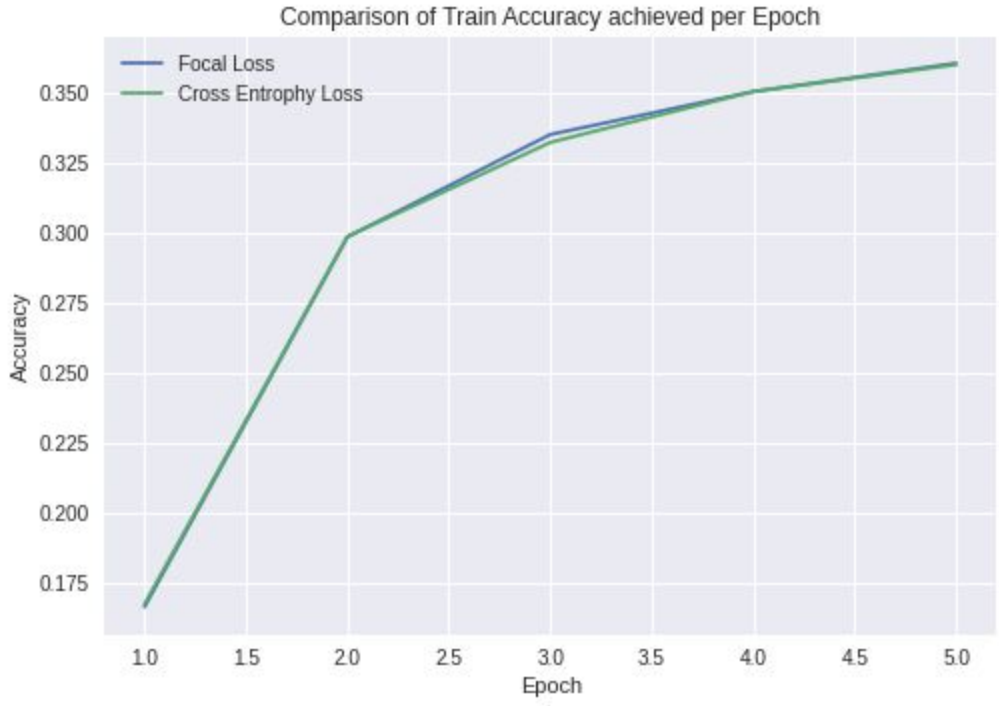
A Densenet 121 model trained on the ImageNet classification dataset is used. Dense Convolutional Network (DenseNet), connects each layer to every other layer in a feed-forward fashion. Model is first trained using Cross Entropy loss function and then Focal loss function .

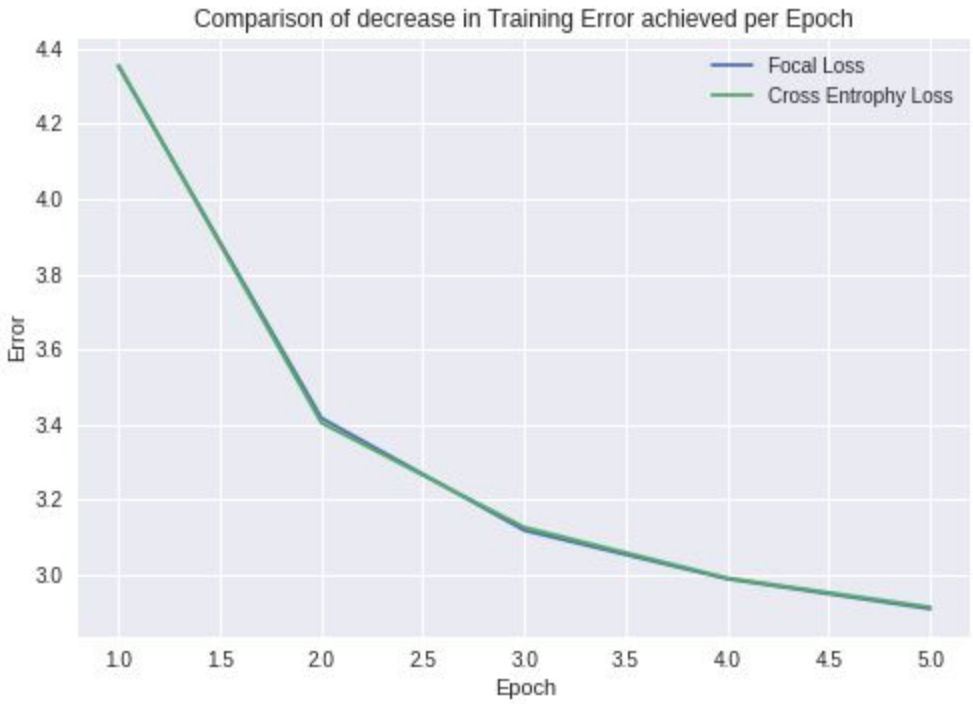
Observation:

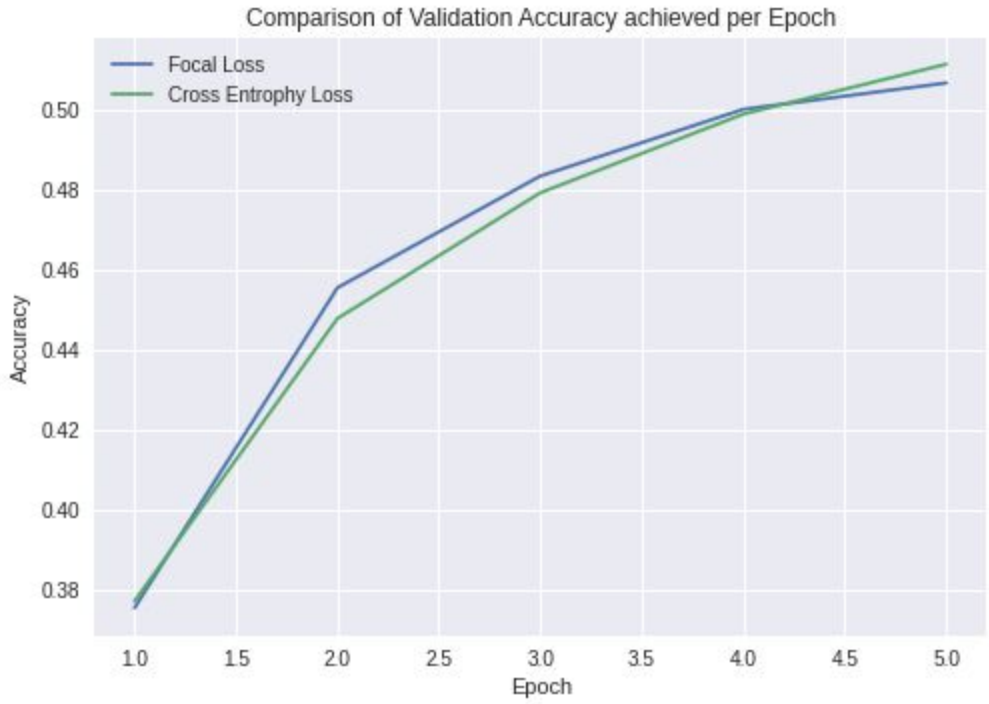
Model is tested for Tiny ImageNet dataset for 5 epochs(because of limited comptation)

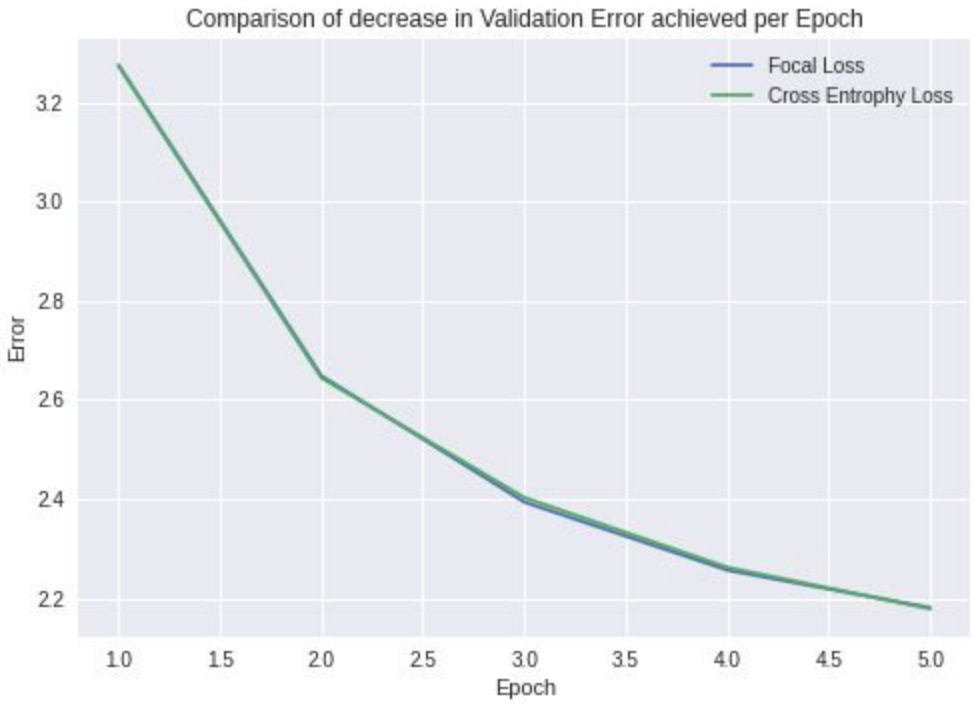
- Accuracy on test data is : **36.6%**
- Focal loss appears to converge faster than cross entropy loss function since it lays emphasis on a few candidate locations

Following curve are obtained between different metrics:




Comparison of decrease in Training Error achieved per Epoch
Focal Loss
Cross Entrophy Loss
Error
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Epoch


Comparison of Validation Accuracy achieved per Epoch
Focal Loss
Cross Entrophy Loss
Accuracy
0.50
0.48
0.46
0.44
0.42
0.40
0.38
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Epoch



References:

1. https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
2. https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html
3. https://cs231n.github.io/linear-classify/#interpret
4. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826
5. https://www.analyticsvidhya.com/blog/2019/10/how-to-master-transfer-learning-using-pytorch/
6. https://towardsdatascience.com/transfer-learning-with-convolutional-neural-networks-in-pytorch-dd09190245ce
7. https://www.kaggle.com/pmigdal/transfer-learning-with-resnet-50-in-pytorch
8. https://www.kaggle.com/columbine/pytorch-sentiment-analysis/notebook
9. https://github.com/bentrevett/pytorch-sentiment-analysis
10. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/