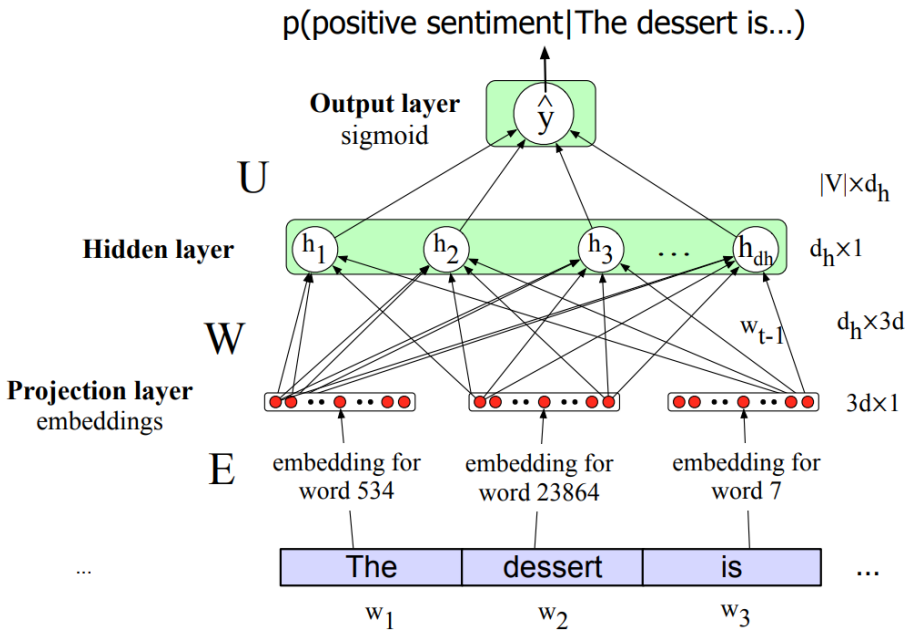
ASSIGNMENT 2
CS60075

Sentiment classification

Submitted by:
Name: Shruti Shreyasi
Roll No: 19EC10086

Sentiment Analysis using Bi-LSTM:



OVERVIEW OF FUNCTIONS AND STAGES IN THE CODE

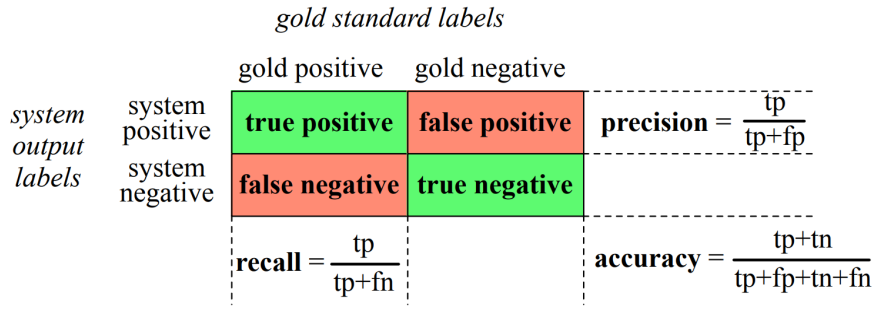
- DATA PRE PROCESSING (NLTK)
  - The dataset was read into a pandas dataframe for handling.
  - English stop words were removed from the dataset because they do not contribute to sentiment of a review.
  - Every word was converted to lowercase.
  - All the non-English words were removed. For this, NLTK was used.
  - It was noted that there were many html tags in the reviews. To remove them Beautiful Soup could have been used. This step was skipped because these tags are not english words and would have been removed in the previous step automatically.

- DATA PRE PROCESSING (FIXING LENGTH)
    - All the punctuations were removed. For this, all the words following the RegEx construct r'\w+' were kept intact and the rest were removed.
    - It was noted that there were many words remaining which would not have any GloVe embedding (the static embedding used). Hence a vocabulary of such words were created which were also present in the GloVe embedding list. The words not present in the above vocabulary were removed.
    - The length of the maximum sized review after all the preprocessing was 1175 (still very high), hence a threshold was applied on the review length. Reviews were cut short to LEN = 200 words. Smaller reviews were padded in the beginning.
    - The preprocessed dataset is thus obtained.

- PRETRAINED GloVE EMBEDDING
    - GloVe 6B, d = 50 is used for word embeddings
    - The words from corpus not present in GloVe are removed from the reviews
    - The padding tokens are also defined with a zero GloVe vector
    - The words in the preprocessed review are encoded with their respective GloVe vectors appended together. This forms the input to the Bi-LSTM layer.

- CLASS TO HANDLE DATASET
    - The dataset is divided into train, validation and test dataset.
    - All tokens present in the dataset are encoded with GloVe embeddings and the appended tensors are given as input to the Bi-LSTM model
    - For using dataloader, the <input, label> pair is defined for every index as a review stored at index i and its labelled sentiment.

- MODEL CLASS AND TRAINING
    - The output is passed to a Bidirectional LSTM layer which has 1 layer, embedding dimensions 50 and hidden size of 25.
    - The result obtained is passed to a linear layer of output size equal to the sentiment label size (2)
    - For training, batch size is set to 200 and 15 epochs are taken. Length of sequence is set to 100.
    - Adam optimizer with learning rate 0.005 is used.
    - Cross entropy loss is taken as the loss function.
    - F1 score is computed for the validation set as the model is trained. The trained model is saved.



- VALIDATION AND TESTING
    - While training, the validation set F1 score is considered for stopping.
    - For validation of the model, the batch size, learning rate and learning rate are experimented with. Hyperparameter tuning is done.
    - For testing the model, the fine tuned model is selected and F1 score is calculated on it.

OBSERVATIONS AND RESULTS OBTAINED

- TRAINING THE MODEL AND VALIDATION ON VALIDATION SET

```
{'epoch': 0, 'loss': 0.6946176290512085, 'tp': 14, 'tn': 89, 'fp': 9, 'fn': 88}
{'tp': 51, 'tn': 54, 'fp': 46, 'fn': 49, 'F1': 0.5177664974619289}
{'epoch': 1, 'loss': 0.6806864738464355, 'tp': 61, 'tn': 56, 'fp': 42, 'fn': 41}
{'tp': 85, 'tn': 16, 'fp': 84, 'fn': 15, 'F1': 0.6319702602230484}
{'epoch': 2, 'loss': 0.6715728044509888, 'tp': 96, 'tn': 24, 'fp': 74, 'fn': 6}
{'tp': 95, 'tn': 9, 'fp': 91, 'fn': 5, 'F1': 0.6643356643356644}
{'epoch': 3, 'loss': 0.6636114716529846, 'tp': 100, 'tn': 18, 'fp': 80, 'fn': 2}
{'tp': 95, 'tn': 13, 'fp': 87, 'fn': 5, 'F1': 0.6737588652482269}
{'epoch': 4, 'loss': 0.6540464758872986, 'tp': 100, 'tn': 21, 'fp': 77, 'fn': 2}
{'tp': 91, 'tn': 16, 'fp': 84, 'fn': 9, 'F1': 0.6618181818181819}
{'epoch': 5, 'loss': 0.6415708661079407, 'tp': 100, 'tn': 32, 'fp': 66, 'fn': 2}
{'tp': 86, 'tn': 27, 'fp': 73, 'fn': 14, 'F1': 0.6640926640926641}
{'epoch': 6, 'loss': 0.626455545425415, 'tp': 97, 'tn': 39, 'fp': 59, 'fn': 5}
{'tp': 74, 'tn': 36, 'fp': 64, 'fn': 26, 'F1': 0.6218487394957983}
{'epoch': 7, 'loss': 0.610329806804657, 'tp': 88, 'tn': 54, 'fp': 44, 'fn': 14}
{'tp': 65, 'tn': 46, 'fp': 54, 'fn': 35, 'F1': 0.593607305936073}
{'epoch': 8, 'loss': 0.594404935836792, 'tp': 81, 'tn': 70, 'fp': 28, 'fn': 21}
{'tp': 63, 'tn': 52, 'fp': 48, 'fn': 37, 'F1': 0.5971563981042654}
{'epoch': 9, 'loss': 0.5752900838851929, 'tp': 75, 'tn': 75, 'fp': 23, 'fn': 27}
{'tp': 70, 'tn': 51, 'fp': 49, 'fn': 30, 'F1': 0.639269406392694}
{'epoch': 10, 'loss': 0.5510165691375732, 'tp': 84, 'tn': 71, 'fp': 27, 'fn': 18}
{'tp': 78, 'tn': 38, 'fp': 62, 'fn': 22, 'F1': 0.65}
{'epoch': 11, 'loss': 0.5303834676742554, 'tp': 91, 'tn': 61, 'fp': 37, 'fn': 11}
{'tp': 76, 'tn': 46, 'fp': 54, 'fn': 24, 'F1': 0.6608695652173913}
{'epoch': 12, 'loss': 0.5088310837745667, 'tp': 86, 'tn': 68, 'fp': 30, 'fn': 16}
{'tp': 66, 'tn': 65, 'fp': 35, 'fn': 34, 'F1': 0.6567164179104478}
{'epoch': 13, 'loss': 0.4914488196372986, 'tp': 76, 'tn': 76, 'fp': 22, 'fn': 26}
{'tp': 71, 'tn': 66, 'fp': 34, 'fn': 29, 'F1': 0.6926829268292682}
{'epoch': 14, 'loss': 0.533403754234314, 'tp': 78, 'tn': 70, 'fp': 28, 'fn': 24}
{'tp': 71, 'tn': 73, 'fp': 27, 'fn': 29, 'F1': 0.7171717171717171}
```

- PERFORMANCE ON TEST SET

```
{'tp': 81, 'tn': 60, 'fp': 40, 'fn': 19, 'F1': 0.7330316742081447}
```

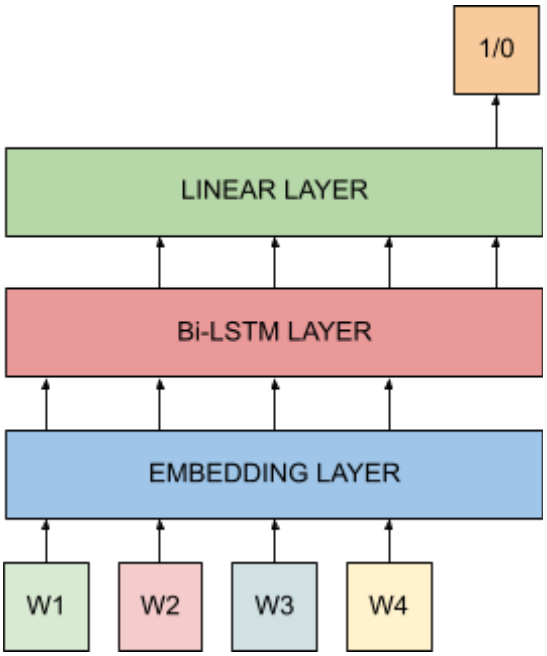


FIG: Schematic of the layers used in the model

LAYERS:

- GloVe embedding layer
- Bi-LSTM Layer
- Linear layer

LOSS FUNCTION:

- Cross Entropy Loss

EVALUATION METRIC:

- F1 Score

EVALUATION:

- F1 score of 71.72% is achieved on the validation set
- F1 score of 73.3% is achieved on the test set.

LINK TO TRAINED MODEL:

[model.txt]

DISCUSSION:

- The performance of neural models is better for natural language tasks such as sentiment detection as compared to statistical and linear models.
- We do not use accuracy as our evaluation metric because the distribution of data is not always uniform. Hence we need to consider both Precision and Recall for a model and hence we use the F1 score metric for evaluation.

- Train is done on the train set, tuning on the validation set and testing on the test set to avoid overfitting.
- Batching is done to speed up the training process on the GPU.
- Static embedding was used for the model. Using context based embedding (e.g. using Transformer) could yield better results.

PROBLEMS FACED:

- Many reviews were very long (the maximum length of review after preprocessing was 1175). For such cases, the reviews were only considered for a fixed length and the later parts were chopped off.
- As the number of layers of Bi-LSTM was limited to 1, the dropout layer was not added. This might lead to overfitting so the number of epochs was increased keeping that into consideration.

CONCLUSION:

Neural models provide nice and acceptable results for natural language tasks such as sentiment classification