EE-782 : Advanced Machine Learning

Assignment-2 Sentiment Analysis in NLP using RNNs

-Submitted by
Bhavesh Garg **17D070031**
Jayesh Choudhary **170070038**

Part (1) : Word Embedding

Observations:

Word embedding : glove-twitter-50

- L2 norm of princess is : 5.012761
- L2 norm of prince - boy + girl is : 5.573292
- L2 norm distance between 2 vectors is: 2.9694865
- 5 closest vectors to 'prince'-'boy'-'girl' are prince, queen, princess,beautiful and girl in this order. ‘princess’ is 3rd closest vector.

Word embedding : glove-twitter-200

- L2 norm of princess is : 6.5795627
- L2 norm of prince - boy + girl is : 7.1876793
- L2 norm distance between 2 vectors is: 5.001938
- 5 closest vectors to 'prince'-'boy'-'girl' are prince, princess, queen, girl and beautiful in this order. ‘princess’ is 2nd closest vector.

Word embedding : glove-wiki-gigaword-50

- L2 norm of princess is : 5.7119613
- L2 norm of prince - boy + girl is : 5.589257
- L2 norm distance between 2 vectors is: 3.2500508
- 5 closest vectors to 'prince'-'boy'-'girl' are prince, princess, queen, duchess and eldest in this order. ‘princess’ is 2nd closest vector.

Word embedding : glove-wiki-gigaword-300

- L2 norm of princess is : 7.2799983
- L2 norm of prince - boy + girl is : 7.830418
- L2 norm distance between 2 vectors is: 5.800549
- 5 closest vectors to 'prince'-'boy'-'girl' are prince, princess, queen, duchess and crown in this order. ‘princess’ is 2nd closest vector.

Word embedding : word2vec-google-news-300

- L2 norm of princess is : 3.2880943
- L2 norm of prince - boy + girl is : 3.4940193
- L2 norm distance between 2 vectors is: 2.3989635
- 5 closest vectors to 'prince'-'boy'-'girl' are prince, princess, duchess, princes and monarch in this order. ‘princess’ is 2nd closest vector.

Conclusion:

1. As the vector dimension increases, L2 norm is greater but since the number of dimension increases, the average effective vector value for 1 dimension is still smaller for case when we use larger dimension. Thus using larger dimensions is better for analogy tasks.
2. Closest words are more or less same for one particular corpora with different dimensions.
3. For different corpora, words are different as the distance from the vector increases.

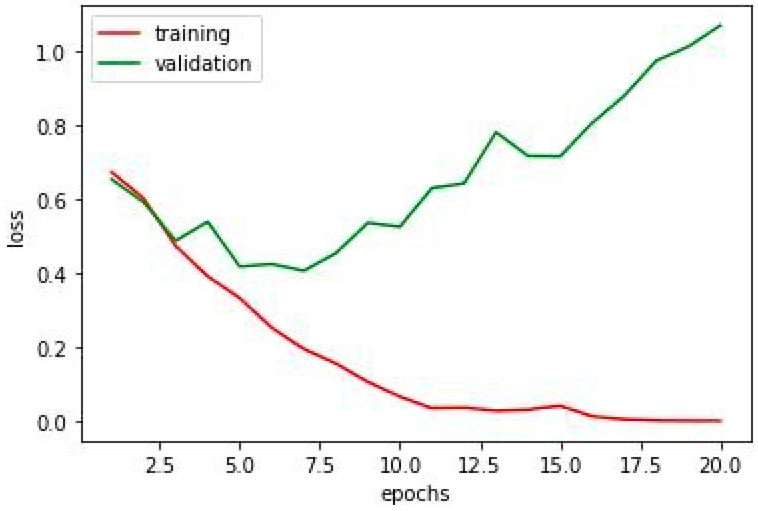
Part (2) : Sentiment Analysis

Sentiment classifier for IMDB movie database was made using pre-trained word embedding and LSTM unit. Output is one dimensional real number between 0 and 1. Values closer to 0 represent bad review and those closer to 1 represent good review.

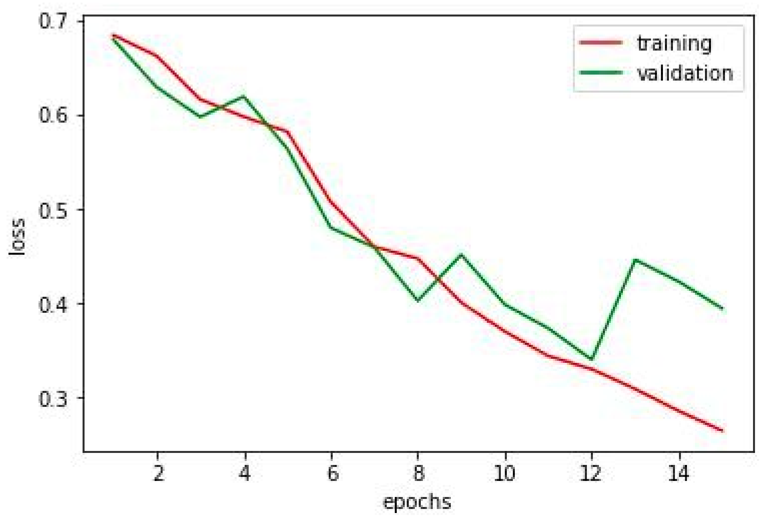
Observations and Inference (Sentiment Analysis)

Baseline model setting: Dropout = 0, Bidirectional = False, Pre-trained-embedding = False, N_layers = 1, Hidden_dim = 256

As evident from the below loss curve, no dropout leads to heavy overfitting in the model, **Test accuracy = 85.98%**.



Adding dropout(p=0.5) and using pre-trained embedding: the model stops overfitting and the test accuracy increases as expected from dropout regularization. The loss curve is shown below.**Test accuracy = 87.31%**



Part (3) : Hyperparameter tuning

Baseline model : Dropout = 0, N_layers = 2, Bidirectional = False, Hidden_dim = 256, Pre-trained-embedding = True, Num_weights = 3.39M, Test_acc = 84.52%. All models have been trained for 15 epochs

Effect of dropout:

- dropout (p=0.3) applied to baseline: test accuracy = 85.40%
- dropout (p=0.5) applied to baseline: test accuracy = 87.71%
- dropout (p=0.8) applied to baseline: test accuracy = 87.82%

Inference : Adding dropout to our model helps control overfitting in the model and this helps performing better on the test set with both loss and accuracy. This is better evident from the previous loss curve.

Effect of weight decay (L2 regularization, dropout=0.5):

- weight decay = 10^-3 applied to baseline: test accuracy = 54.57%
- weight decay = 10^-4 applied to baseline: test accuracy = 87.03%
- weight decay = 10^-5 applied to baseline: test accuracy = 86.03%

Inference : Applying regularization by weight decay after dropout does not improve the test accuracy as compared to with only dropout. Applying very high (10^-3) weight decay has a very adverse effect and causes underfitting.

- weight decay = 10^-5 applied to baseline **without dropout**: test accuracy = 86.96%
- weight decay = 5*10^-4 applied to baseline **without dropout**: test accuracy = 86.41%

Inference : Applying weight decay alone increases the test accuracy but does not prevent overfitting. The effect of weight decay and dropout independently on test accuracy is similar but dropout prevents overfitting and thus it should be chosen for regularization.

Effect of hidden dimensions (dropout=0.5):

- hidden dim = 128: no. of weights = 2.75 M, test accuracy = 88.02%
- hidden dim = 512: no. of weights = 5.86 M, test accuracy = 87.58%

Inference : Decreasing number of hidden dims slightly increases the test accuracy here (87.71 to 88.02) whereas increasing hidden dims slightly reduces it (87.71 to 87.58). This can mean that the model does not require a lot of parameters to fit on the training dataset and access parameters can start overfitting.

Effect of number of LSTM layers (dropout=0.5):

- num_layers = 3: no. of weights = 3.92 M, test accuracy = 87.14%
- num_layers = 4: no. of weights = 4.44 M, test accuracy = 88.00%

Inference : Increasing the number of LSTM layers does not affect the test accuracy by a significant margin and it also increases the training time per epoch. Therefore, number of LSTM layers = 2 in the model is sufficient for the dataset at hand and increasing the layers can also start overfitting.

Part(4) : Testing on custom input:

Model setting : dropout = 0.5, bidirectional = False, pre-trained-embedding = false, n_layers = 2, hidden_dim = 256, Testing Accuracy : 88.73%

Custom dataset:

● "it is a disgusting and stupid film"	- 0.075 - negative sentiment
● "That was a very good film"	- 0.933 - positive sentiment
● "This is a very good film"	- 0.998 - positive sentiment
● "Star wars is a very good film"	- 0.998 - positive sentiment
● "Tees Maar Khan was a stupid film"	- 0.013 - negative sentiment
● "I liked the movie very much"	- 0.991 - positive sentiment
● "I would recommend everyone to watch this movie"	- 0.997 - positive sentiment
● "Good movie"	- 0.996 - positive sentiment
● "I will watch it again"	- 0.994 - positive sentiment
● "Waste of money"	- 0.006 - negative sentiment

Inference : The model performs well on our custom inputs. All the inputs were classified correctly. Even when particular movie names like "Star Wars" are included, the results are good.

Part(5) : Bonus part

Using bidirectional LSTM: the number of weights and the training time per epoch of the model increases considerably, however the test accuracy does not show much improvement, test accuracy for baseline model = 87.84% (with dropout enabled).

Inference : Using bidirectional LSTMs can help model apply context from the end of a sentence or review. On retraining many of the above models using bidirectional LSTMs, we find that the performance gains are not significant, however, the no. of weights, the training time per epoch and the GPU usage increases significantly.