Sentiment Analysis Using different LSTM/Bi-LSTM Models

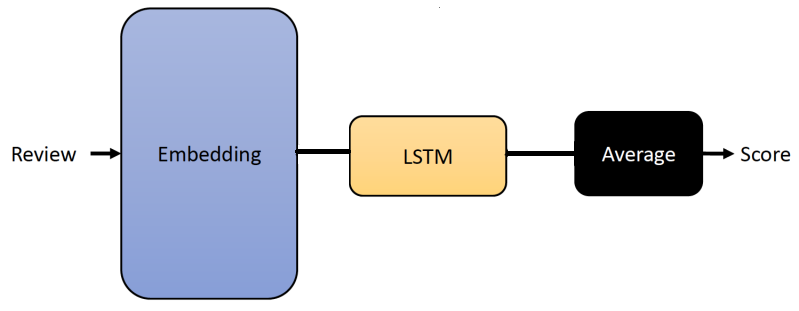
**Abstract**—With the popularity of social networks, and ecommerce websites, sentiment analysis has become a more active area of research in the past few years. On a high level, sentiment analysis tries to understand the public opinion about a specific product or topic, or trends from reviews or tweets. Sentiment analysis plays an important role in better understanding customer/user opinion, and also extracting social/political trends. There has been a lot of previous works for sentiment analysis, some based on hand-engineering relevant textual features, and others based on different neural network architectures. In this assignment, I present some models based on long short-term-memory (LSTM models). I am also able to achieve one model that gives comparatively high accuracy considering other models train for this assignment.

**Introduction**—Emotions exist in all forms of human communication. In many cases, they shape one’s opinion of an experience, topic, event, etc. We can receive opinions and feedback for many products, online or otherwise, through various means, such as comments, reviews, and message forums, each of which can be in the form of text, video, polls and so on. One can find some type of sentiment in every type of feedback, e.g. if the overall experience is positive, negative, or neutral. The main challenge for a computer is to understand the underlying sentiment in all these opinions. Sentiment analysis involves the extraction of emotions from and classification of data, such as text, photo, etc., based on what they are conveying to the users.

In the age of information, the Internet, and social media, the need to collect and analyse sentiments has never been greater. With the massive amount of data and topics online, a model can gather and track information about the public opinion regarding thousands of topics at any moment. This data can then be used for commercial, economical and even political purposes, which makes sentiment analysis an extremely important feedback mechanism.

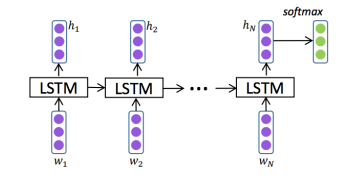
With the rising popularity of deep learning in the past few years, combined with the vast amount of labelled data, deep learning models have replaced many of the classical techniques used to solve various natural language processing and computer vision tasks. In these approaches, instead of extracting hand-crafted features from text and images, and feeding them to some classification model, end-to-end models are used to jointly learn the feature representation and perform classification. Deep learning-based models have been able to achieve state-of-the-art performance on several tasks.

In this assignment, I seek to improve the accuracy of sentiment analysis using different LSTM/bidirectional LSTM (Bi-LSTM) networks. and test them on the most popular sentiment analysis databases named SST datasets. The block-diagram of the proposed algorithm is shown in below figure.



**The Proposed Framework-**As mentioned previously, I did some trail on different LSTM models to SST dataset to perform sentiment analysis. Different LSTM models have been used to train models to evaluate accuracy and prediction done by test dataset.

**The LSTM model architecture-**LSTM is a popular recurrent neural network architecture for modelling sequential data, which is designed to have a better ability to capture long term dependencies than the vanilla RNN model. As other kind of recurrent neural networks, at each time-step, LSTM network gets the input from the current time-step and the output from the previous timestep, and produces an output which is fed to the next time step. The hidden layer from the last time-step (and sometimes all hidden layers), are then used for classification. The high-level architecture of a LSTM network is shown in below Figure



As mentioned above, the vanilla RNN often suffers from the gradient vanishing or exploding problems, and LSTM network tries to overcome this issue by introducing some internal gates. In the LSTM architecture, there are three gates (input gate, output gate, forget gate) and a memory cell. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

many applications, we are interested in the temporal information flow in both directions, and there is variant of LSTM, called Bidirectional-LSTM (Bi-LSTM), which can address this. Bidirectional LSTMs train two hidden layers on the input sequence. The first one on the input sequence as-is, and the second one on the reversed copy of the input sequence. This can provide additional context to the network, by looking at both past and future information, and results in faster and better learning.

In my final model I used one LSTM layer and one bi-LSTM to gives comparatively better accuracy and better prediction.

**Experimental Results-**Before presenting the experimental results, let us first discuss the hyper-parameters used in our training, and give an overview of the datasets used in our experiments. I train the proposed final bi-LSTM model for 250 epochs using CPU. The batch size is set to 16 for SST2 dataset. ADAM optimizer is used to optimize the loss function, with a by default learning rate and weight decay.

**Databases-** Stanford Sentiment Treebank (SST) is a binary sentiment analysis dataset, with train/dev/test splits provided. It is worth to mention that this data is actually provided at the phrase-level and hence one can train the model on both phrases and sentences. The vocabulary size for this dataset is around 16k.

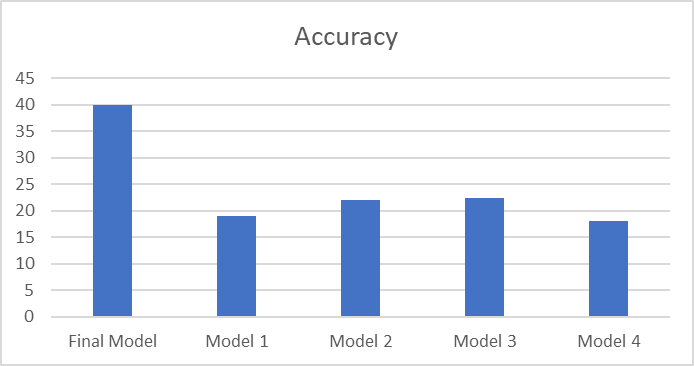
In this dataset there are separate text file named datasetSentences, datasetSplit, dictionary and sentiment\_labels.I prepared three separate csv file for train, dev and test by using above text files according to provided split data and levels. By processing all data finally, we get sentences with their levelled sentiment (negative, positive, very\_positive, neutral and very\_negative.

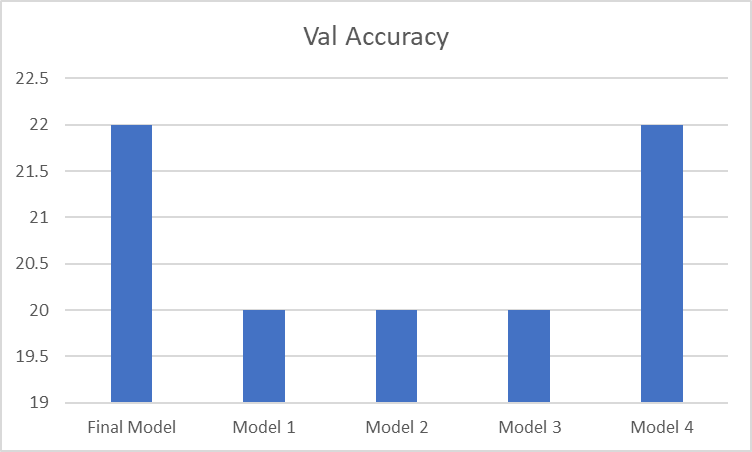
**Model Performance and Comparison-**

I will now present the experimental results of all the models I tried to train with SST-2 dataset.

I will first compare the classification accuracy of 5 different LSTM models and finally I will share in details of the model I got maximum accuracy.

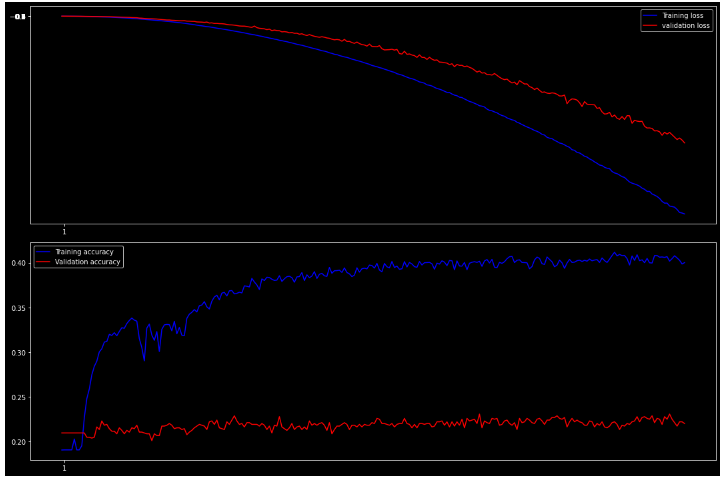
|  |  |  |
| --- | --- | --- |
| Model | accuracy | Val\_accuracy |
| Model 1 | 19 | 20 |
| Model 2 | 22 | 20 |
| Model 3 | 22.45 | 20 |
| Model 4 | 18 | 22 |
| Final Model | 40 | 22 |

****

****

In the final model I used total 9 layers in which 1 Embedding layer with unit 64 then 1 Bidirectional\_LSTM layer with unit 64 then 1 LSTM layer with unit 64 and finally 2 dense layers with activation relu and sigmoid.

With this configuration I got a maximum train accuracy of 40% and validation accuracy of 22%. With model I got a prediction accuracy of 19% using test dataset.



**Conclusion-** In this work I proposed a model with training accuracy 40%. This not a good fitted model as when I did prediction, I got only 19% prediction accuracy and majority time model predict with wrong sentiment. I believe with time I can do more trail and test and come up with good accuracy and better fitted model.