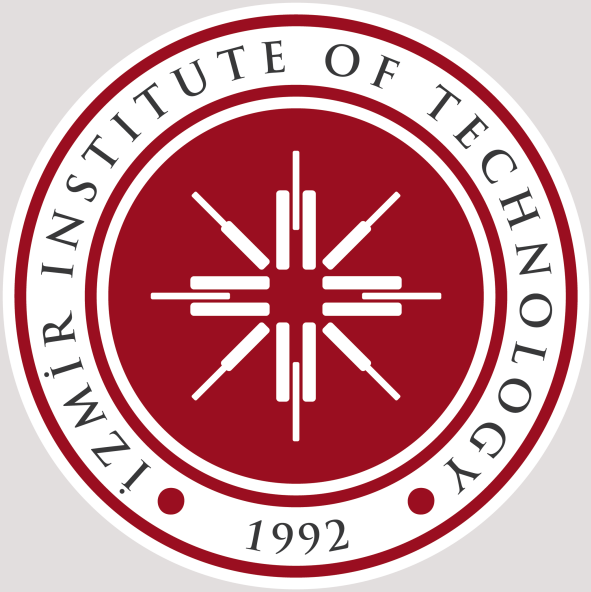
Izmir Institute of Technology
Computer Engineering Department
CENG464 Term Project Report

Student Name: Gökay Gülsoy Student No: 270201072
Student Name: Merve Nur Ozan Student No: 270201071

June 3, 2024



1 Introduction

Field of NLP has advanced over the last few years considerably. Fundamental NLP tasks such as email filtering, predictive text auto correction, automatic summarization, machine translation, etc. can be performed with much higher accuracy and speed than 10 years ago. Thanks to developments in the field of Deep learning, more sophisticated algorithms for processing and training data has devised, but this success is achieved via joint collaboration of two sub fields of computer science namely NLP and Computer Architecture. Large language models which are existing today are utilizing very powerful GPUs which have massively parallel computing capability, so even very large datasets can be processed in a reasonable amount of time in big clusters or in distributed computing systems.

1.1 Problem Definition

Our focus is on quite fundamental but very important problem known as sentiment analysis. It is widely used as an business intelligence tool that helps corporations enhance their products and services. We are applying sentiment analysis problem to two different cases, first one is labeling reviews of people on movies as positive or negative and second is to understand whether person's comments on twitter in some context are positive or negative. Task is actually an binary classification problem (being positive or being negative) in which deep learning model has to classify whether a review is positive or negative. Understanding the meaning of a sentence is not solely possible by looking at the syntactic structure of the sentence, because words can change the meaning of the whole sentence whenever it is used in another position or with another word. Sentiment analysis models should look for the context in which a word is used and its association with other words inside a specific word window.

2 Methodology

2.1 The Models and Methods Used

We have created two different models on two different datasets, our first model is a relatively older neural network model which is LSTM and second model is newer BERT model which was proven to produce state of the art results for many NLP tasks as well as sentiment analysis. In our LSTM model we have used encoder as the first layer for encoding the input words into indices, then embedding layer takes the output of encoding layer to create word embeddings. Embeddings are given to LSTM unit and output of the LSTM unit is given to Dense layer of size 64 with relu activation function. Finally classification head which is Dense layer with size 1 outputs the classification result. In our BERT model we have used Input layer as the first layer which takes the text in appropriate shape and then text is given to preprocessing layer in order to preprocess the text. Outputs of preprocessing layer are then given to BERT encoder, then dropout layer with dropout ratio 0.1 is added in order to provide regularization. Finally classification head which is of size 1 is used to output prediction value. We have followed an consistent deep neural network training workflow, we firstly installed all the required dependencies for working with datasets and models. Secondly we have preprocessed the data in order to make it ready for neural network training. Then we have created our LSTM and BERT models, Fourthly we have trained our models on corresponding training datasets and then evaluated our models with validation dataset. Finally for reusability of our model we have saved our trained models so that they can be used later on inference without requiring to be trained again.

3 Experiments and Results

3.1 The Datasets Used

First dataset we have used is related to movie reviews and predicting whether a movie review is positive or negative. Movie Review dataset consists of 25,000 movie reviews for training, and 25,000 for testing [3]. In the entire collection, no more than 30 reviews are allowed for any given movie because reviews for the same movie tend to have correlated ratings. Further, the train and test sets contain a disjoint set of movies, so no significant performance is obtained by memorizing, movie-unique terms and their associated observed labels. Second dataset we have used is related to predicting whether twitter comments have positive meaning or negative meaning under certain context. It contains 1,600,000 tweets extracted using the twitter api. The tweets have been annotated (0 negative, 4 = positive) and they can be used to detect sentiment. This dataset contains 6 different fields first one is target that indicates the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive), second field is id which denotes the id of the tweet, third field is the date of the field, fourth field is the query if there is no query this value is NO_QUERY, fifth field is the user that tweeted, sixth field is the text of the tweet. We have only used text column for the sentiment analysis in twitter dataset as other columns are irrelevant to classification task.

3.2 Experiments Performed

We have carried out 8 different experiments for LSTM model and 7 different experiments for BERT model. Experiments which were carried out for the LSTM is given as follows:

- Increasing LSTM layers (stacking)
- Increasing Dense layer number
- Changing activation function
- Changing dropout ratio or adding dropout layer
- Changing the batch size
- Changing the learning rate
- Changing the number of epochs

baseline model result for LSTM is as follows:

loss: 0.4491, mean_square_error: 0.2526, auc: 0.9044, binary_accuracy: 0.8211, f1_score: 0.8169, val_loss: 0.5084, val_mean_squared_error: 0.2668, val_auc: 0.9085, val_binary_accuracy: 0.8286, val_f1_score:

0.8380

Experiment result: stacking LSTM with 32 layers.) loss: 0.4458, mean_square_error: 0.1616, auc: 0.0.8733, binary_accuracy: 0.8045, f1_score: 0.8078, val_loss: 0.4558, val_mean_squared_error: 0.1634, val_auc: 0.9119, val_binary_accuracy: 0.8318, val_f1_score: 0.8319

Experiment result: Adding Dense layer with 128 units .) loss: 0.4546, mean_square_error: 0.1475, auc: 0.8884, binary_accuracy: 0.8045, f1_score: 0.8078, val_loss: 0.4558, val_mean_squared_error: 0.1634, val_auc: 0.8930, val_binary_accuracy: 0.8047, val_f1_score: 0.8084

Experiment result: Changing activation function from relu to tanh.) loss: 0.4778, mean_square_error: 0.1911, auc: 0.8733, binary_accuracy: 0.7849, f1_score: 0.7759, val_loss: 0.4812, val_mean_squared_error: 0.4113, val_auc: 0.9011, val_binary_accuracy: 0.8120, val_f1_score: 0.8194

Experiment result: Adding Dropout layer with dropout ratio of 0.5 .) loss: 0.6387, mean_square_error: 0.4273, auc: 0.7894, binary_accuracy: 0.6988, f1_score: 0.6562, val_loss: 0.5454, val_mean_squared_error: 0.1854, val_auc: 0.8677, val_binary_accuracy: 0.7026, val_f1_score: 0.6170

Experiments result: hidden layer size 128 .) loss: 0.5141, mean_square_error: 0.1708, auc: 0.8365, binary_accuracy: 0.7499, f1_score: 0.7292, val_loss: 0.4774, val_mean_squared_error: 0.1930, val_auc: 0.8873, val_binary_accuracy: 0.7974, val_f1_score: 0.8094

Experiment result: batch size 32 .) loss: 0.4374, mean_square_error: 0.1528, auc: 0.8964, binary_accuracy: 0.8112, f1_score: 0.8082, val_loss: 0.4823, val_mean_squared_error: 0.1579, val_auc: 0.8677, val_binary_accuracy: 0.7969, val_f1_score: 0.8083

Experiment result: batch size 128 .) loss: 0.6417, mean_square_error: 0.2253, auc: 0.6872, binary_accuracy: 0.8112, f1_score: 0.6012, val_loss: 0.6326, val_mean_squared_error: 0.2211, val_auc: 0.7105, val_binary_accuracy: 0.6370, val_f1_score: 0.5952

Experiment result: learning rate 2e-4 .) loss: 0.4970, mean_square_error: 0.5915, auc: 0.8952, binary_accuracy: 0.8114, f1_score: 0.8055, val_loss: 0.4609, val_mean_squared_error: 0.1542, val_auc: 0.8889, val_binary_accuracy: 0.7984, val_f1_score: 0.7858

Experiment result: learning rate 1e-5 .) loss: 1.2521, mean_square_error: 0.4174, auc: 0.5322, binary_accuracy: 0.5000, f1_score: 0.000e+00, val_loss: 1.1930, val_mean_squared_error: 0.4108, val_auc: 0.5302, val_binary_accuracy: 0.4917, val_f1_score: 0.0000e+00

Experiment result: number of epochs 10 .) loss: 0.4252, mean_square_error: 0.1596, auc: 0.9074, binary_accuracy: 0.8208, f1_score: 0.8259, val_loss: 0.4540, val_mean_squared_error: 0.3052, val_auc:

0.9087, val_binary_accuracy: 0.8250, val_f1_score: 0.8306

Experiments which were carried out for the BERT is given as follows:

- Increasing Dense layer number
- Changing dropout ratio
- Changing batch size
- Changing learning rate

baseline model result for BERT is as follows:

small_bert/bert-en-uncased-L-4_H_512_A-8 loss: 0.1513, binary_accuracy: 0.9418, auc: 0.9543, f1_score: 0.9407, mean_squared_error: 22.2461, val_loss: 0.4765, val_binary_accuracy: 0.8524, val_auc: 0.8667, val_f1score: 0.8514, val_mean_squared_error: 28.0825

Experiment result: Increasing Dense layer number by adding Dense layer with 64 units before classification head .) loss: 0.1519, binary_accuracy: 0.9437, auc: 0.9549, f1_score: 0.9428, mean_squared_error: 25.7636, val_loss: 0.4844, val_binary_accuracy: 0.8552, val_auc: 0.8675, val_f1score: 0.8548,
val_mean_squared_error: 31.4863

Experiment result: Increasing Dropout ratio from 0.1 to 0.5 .) loss: 0.1752, binary_accuracy: 0.9336, auc: 0.9549, f1_score: 0.9323, mean_squared_error: 23.0459, val_loss: 0.4848, val_binary_accuracy: 0.8510, val_auc: 0.8607, val_f1score: 0.8513,
val_mean_squared_error: 28.4279

Experiment result: Increasing Dropout ratio from 0.1 to 0.5 .) loss: 0.1752, binary_accuracy: 0.9336, auc: 0.9549, f1_score: 0.9323, mean_squared_error: 23.0459, val_loss: 0.4848, val_binary_accuracy: 0.8510, val_auc: 0.8607, val_f1score: 0.8513,
val_mean_squared_error: 28.4279

Experiment result: Changing learning rate from 3e-5 to 2e-5 .) loss: 0.2116, binary_accuracy: 0.9128, auc: 0.9332, f1_score: 0.9104, mean_squared_error:14.9712, val_loss: 0.4123, val_binary_accuracy: 0.8476, val_auc: 0.8467, val_f1score: 0.8467,
val_mean_squared_error: 19.0952

Experiment result: Changing batch size form 32 to 64 with learning rate 2e-5 .) loss: 0.2491, binary_accuracy: 0.8921, auc: 0.9165, f1_score: 0.8880, mean_squared_error: 10.1090, val_loss: 0.3773, val_binary_accuracy: 0.8454, val_auc: 0.8667, val_f1score: 0.8416, val_mean_squared_error: 13.2432

4 Discussion

4.1 The Evaluation of Experiments

when we compare our baseline LSTM model with the modified version in experiment 1 which stacks another LSTM with 32 layers, loss decreased from 0.4491 to 0.4458 and binary accuracy decreased from 0.8211 to 0.8045. Validation binary accuracy increased from 0.8286 to 0.8318 and validation loss decreased from 0.5084 to 0.4558 which indicates that stacking another LSTM layer is not necessarily increasing binary accuracy even though it may cause reduction in loss and validation loss. In experiment 2 we added one more Dense layer with 128 units just before the classification head, loss has increased from 0.4491 to 0.4546 and binary accuracy decreased from 0.8211 to 0.8045 and validation loss decreased from 0.5084 to 0.4558 which indicated that adding one more dense layer with 128 units just before the classification head decreased the binary accuracy and increased the loss, whereas it reduced validation loss. In experiment 3 we have changed activation function of classification layer from relu to tanh, so loss increased from 0.4491 to 0.4778 and binary accuracy decreased from 0.8211 to 0.7849 and validation loss decreased from 0.5084 to 0.4812 which indicated that changing activation function from relu to tanh badly affect the binary accuracy and increased loss, even though it has caused decrease in validation loss. In experiment 4 we have added dropout layer with drop out ratio 0.5 to our model, as a result loss increased from 0.4491 to 0.6387 and binary accuracy decreased from 0.8211 to 0.6387. Validation loss increased from 0.5084 to 0.5454, so adding dropout layer with 0.5 dropout ratio reduced the model's overall performance. In experiment 5 we have increased hidden layer size from 64 to 128, loss increased from 0.4491 to 0.5141 and binary accuracy decreased from 0.8211 to 0.8112. Validation loss decreased from 0.5084 to 0.4774 which indicates that increasing hidden layer size from 64 to 128 reduced the accuracy of model even though validation loss decreased. In experiment 6 we have changed batch size from 64 to 32, loss decreased from 0.4491 to 0.4774 and binary accuracy decreased from 0.8211 to 0.8112. Validation loss decreased from 0.5084 to 0.4823 which indicated that decreasing batch size from 64 to 32 reduced accuracy even though it caused reduction in validation loss. In experiment 7 increasing learning rate from 1e-4 to 2e-4 decreased loss from 0.4970 to 0.4609 and decreased binary accuracy from 0.8211 to 0.8114. Validation loss decreased from 0.5084 to 0.4609 which has shown that even though increase in learning rate decreased accuracy it has decreased validation loss. In experiment 8 reducing learning rate from 1e-4 to 1e-5 increased loss from 1.2521 and binary accuracy decreased from 0.8286 to 0.5000. Validation loss increased from 0.5084 to 1.1930 which indicates that reducing learning rate from 1e-4 to 1e-5 reduced the overall performance of the model. In experiment 9 number of epochs increased from 5 to 10, loss decreased from 0.4491 to 0.4252 and binary accuracy decreased from 0.8286 to 0.8208. Validation loss has decreased from 0.5084 to 0.4123 which indicates that increasing number of epochs from 5 to 10 caused a slight reduction in binary accuracy, but reduced loss and validation loss. In experiment 10 when we compare the baseline BERT model with the modified version in which we have added dense layer of size 64 just before the classification head, loss increased from 0.1513 to 0.1519 and binary accuracy increased from 0.9418 to 0.9437. Validation loss increased from 0.4765 to 0.4844. In

experiment 11 dropout ratio is increased from 0.1 to 0.5 for our BERT model, loss increased from 0.1513 to 0.1752 and binary accuracy increased from 0.8524 to 0.9336. Validation loss increased from 0.4765 to 0.4848. In experiment 12 learning rate decreased from 3e-5 to 2e-5, loss decreased from 0.4765 to 0.4123, binary accuracy decreased from 0.8524 to 0.8510. Validation loss decreased from 0.4765 to 0.4123. In final experiment batch size is changed from 32 to 64 while learning rate is 2e-5. Loss increased from 0.1513 to 0.2491 and binary accuracy decreased from 0.9418 to 0.8921. Validation loss decreased from 0.4844 to 0.3773.

5 Conclusion

In conclusion, BERT based models have better accuracy and validation score compared to LSTM based models in most cases. Experiments shown that changing different parameters that affect the performance of the model may not have uniform effect in most cases which means that there may not be straightforward relationship between metrics like accuracy, loss, or validation loss such as whenver accruacy increases, loss decreases and then validation loss decreases. Each model require different set of optimal parameters in order to fine-tune the model.