Basic Sentiment Analysis of IMDB Reviews

(COMP3125 Individual Project)

\*Note: Do not used sub-title

Rabih Elmassih

*Abstract*— This report focuses on implementing sentiment analysis for IMDB movie reviews using the TensorFlow package, a powerful tool for deep learning. The primary aim with the model is classifying reviews into positive and negative categories based off it’s contents. The dataset consisted of 50,000 movie reviews with binary sentiment labels. The layers of the model are an embedding layer, a dense layer consisting of a ReLU into sigmoid activation functions of different sizes. After training the model was evaluated utilizing accuracy and loss metrics. The results show a accuracy of 88.53% and a loss of 29.6% on testing data. This shows promise with future work with advanced methods aimed at improving this to at least 90% on unseen data.

Keywords—sentiment analysis, TensorFlow, IMDB reviews, test classification

# Introduction (*Heading 1*)

Sentiment analysis is a Natural Language Processing (NLP) task that involves classifying text data according to the sentiment it conveys; typically, as positive, negative, or neutral sentiment. Some applications are with project reviews, analysis of social media mentality, and customer feedback. This report focuses on classifying IMDB movie reviews into a binary positive and negative sentiment using TensorFlow. Sentiment analysis is essential in analyzing public opinion on trends which can easily be applied in marketing and public relations.

The performance of sentiment analysis tasks has improved with the use of neural networks. This report uses TensorFlow to build a simple sentiment classifier using a database of 50,000 IMDB movie reviews.

# Datasets

## Source of dataset (Heading 2)

The dataset was accessed via the TensorFlow package specifically tensorflow.keras.datasets import imdb. It is a standard dataset commonly used for binary sentiment classification in Natural Language Processing.

## Character of the datasets

The dataset consists of 2 lists. A list of each review encoded as integers and a corresponding list of 0s and 1s classifying the sentiment of the corresponding review. The dataset was loaded with only the top 10,000 words using the num\_words parameter to reduce the complexity of the model. The data was then split into training and testing portions. Then the number of each sentiment was determined along with shape of training and testing data ensuring they were split correctly leaving 15% for training. Then the average length of a review and a sample of the raw data of a review were found to better understand the data. Lastly the data was padded to a max length of 1000 based off the average size of the review with being added whitespace in the form of 0s to the end of a review and characters after 1000 being truncated off to ensure constant size of each of reviews.

# Methodology

The TensorFlow ecosystem was utilized for the building and training of the model given its powerful tools and flexibility making suitable for the task.

## Model Architecture

1. *Embedding*

An embedding layer was used to map words to dense vectors of a fixed size. Allows it to capture the sematic relationship between words, enabling the model to have a better understanding of the context of the text while reducing the dimensionality of the input data while retaining relevant information,

1. *Flatten*

The output of the embedding layer is a 2D matrix so a flatten layer was used to transform it into a 1D vector. This ensures capacity with following layer

1. *Dense*

A fully connect layer with 16 units was included to learn train complex, non-linear relationships between features. ReLU (Rectified Linear Unit) was chosen as a activation function to introduce non-linearity to enable a far better ability to capture the trends of the data.

1. *Output*

A single unit dense layer with a sigmoid activation function was used to predict the binary sentiment of each review

## Model Compileing

1. *Optimizer*

The optimizer of Adam was used given its adaptive learning capabilities and general robustness across many tasks alongside its suitability for training deep learning models.

1. *Loss*

A Loss of binary cross-entropy was used since it is ideal for binary classification tasks. The loss function measures the difference between predicted probabilities and the actual labels to enable a better model.

1. *Metrics*

The metric of accuracy was used as a way of evaluating model performance.

## Model Training

The model was trained with 5 epochs, a batch size of 512 and a 0.2 validation split. The number of epochs and validation splits were chosen in a attempt to fight over and under fitting the model.

# Results

## Model on training data

### Training Accurarcy: 0.9509

* The proportion of correctly classified reviews in the training dataset.

### Valudation Accurarcy: 0.8912

* The proportion of correctly classified reviews on unseen validation data.

### Training Loss: 0.1418

* Quantitative error in predictions made in training dataset.

### Valudation Loss: 0.2902

* Quantitative error in predictions made on unseen validation data.

### Overall trends

* Models seem to stop improving at the 3rd epoch with potential of overfitting but showing overall high accuracy across both types of data.

### Methods for metrics

* Metrics come from TensorFlow libraries alongside graph built using matplotlib.pyplot.

## Evaulation of Model on testing data

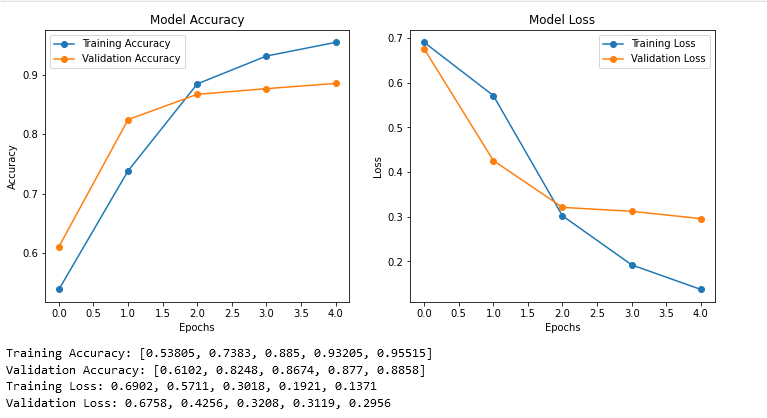
### Test Loss: 0.2857

* The error on unseen testing data with the model performing like its validation set.

### Test Accuracy: 0.8892

* The proportion of correctly classified reviews on the unseen test dataset.

### Overalls robust preformence of model on unseen data.



# Discussion

The model performed incredibly well for its simplicity with a test accuracy of 88.92%. However, there is major room for improvement via the use of more advanced models such as LSTM (Long Short-Term Memory) or a pre-trained BERT model to achieve better results due to a better ability to grasp the deep nuances of human language. Alongside this, the ability to work with neutral sentiment is also a wanted improvement given the binary of positive and negative doesn’t truly capture the ranges of human sentiment.

# Conclusion

This project performed sentiment analysis upon IMDB movies reviews using a simple deep learning model built with TensorFlow to classify sentiment into positive and negative sentiment with a test accuracy of 88.92% achieved by the final model demonstrating a effectiveness in its straightforward approach. Sentiment analysis has many practical applications, especially in analyzing trends.