**PROJECT REPORT**

INTRODUCTION

This code implements an LSTM model for sentiment analysis on movie reviews, using pre-trained GloVe embeddings. It trains on a subset of the IMDb dataset and evaluates performance on the test set. It also analyzes the impact of dataset size on model performance.

OBJECTIVE: Investigating the impact of dataset size on model performance by training the model on different subsets of the training dataset with increasing sample sizes and recording the final performance metrics for each subset.

LSTM model has been used for sentiment analysis on the IMDb movie review dataset. The dataset contains movie reviews labelled as positive or negative. The model uses pre-trained GloVe embeddings to encode words in the reviews and a single LSTM layer to learn the sequence representation of the reviews. The output from the LSTM layer is fed into a dense layer with a sigmoid activation function to predict the review's sentiment label (positive or negative).

The pre-trained GloVe embeddings has been extracted. The maximum number of words has been set to 10,000, and the maximum length of the reviews is 150. The IMDb dataset has then been loaded and keeps only the first 100 samples for training, pads and truncates the reviews to a fixed length, and creates an embedding matrix by mapping the words in the dataset to their corresponding pre-trained embeddings.

The LSTM model is then defined using the Sequential API from Keras. The first layer is an embedding layer that takes the pre-trained embedding matrix as input and maps each word in the reviews to its corresponding embedding. The second layer is an LSTM layer with 32 units and 20% dropout rate to avoid overfitting. The final layer is a dense layer with a sigmoid activation function that outputs the positive or negative review probability.

The model is compiled with the RMSprop optimizer and binary cross-entropy loss function. It is trained on the training dataset for 30 epochs with a batch size of 32, and validated on 10,000 samples from the test dataset. The training and validation accuracy and loss are plotted using Matplotlib.

Test accuracy: 60.56%

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated**

Test loss: 0.7662714719772339 Test accuracy: 0.5177199840545654

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated**

Test accuracy with 20000 training samples: 51.68%

**Chart, line chart

Description automatically generated**

CONCLUSION

The model is evaluated on the test dataset, and the test loss and accuracy are printed. The model has been on different subsets of the training dataset with increasing sample sizes and records the final training accuracy, validation accuracy, training loss, validation loss, and test accuracy for each subset. Early stopping is used to prevent overfitting. The test accuracy which has been acquired with 20000 training samples is 51.68%.

**Table

Description automatically generated**