Homework 1 – Sentiment Analysis using LSTM

Fasi Ullah Khan Mohammed – 906576890

1 Introduction:

Sentiment analysis is the process of determining the emotional tone or attitude expressed in a piece of text. It is a type of natural language processing (NLP) that helps in identifying the polarity of opinions or feelings conveyed in written language. Here, we will be using the IMDB dataset which consists of reviews about movies classified into positive or negative.

2 Dataset:

The IMDB dataset has a total of 50,000 reviews, containing 25,00 positive reviews and 25,000 negative reviews.

3 Pre-processing:

The sentiment column was converted into a binary format where "positive" means 1 and "negative" means 0 Text cleaning was performed by converting the text to lowercase, removing HTML tags and eliminating any special characters.

3.1 Tokenization:

Keras Tokenizer was used to convert the words in the reviews into a sequence of integers. The vocabulary size was limited to 10,000 most frequent words.

3.2 Padding:

Since movie reviews have different lengths, all sequences were padded to a fixed length of 200 words.

4 Splitting the dataset:

The dataset was split into training set (80%), Development set (10%) and Test set (10%)

5 Building the LSTM model:

Three layers were used

- Embedding layer, to help the model learn word relationships by transforming the integer-encoded words into dense vectors of fixed size (128 dimensions)
- LSTM layer, to remember long-term dependencies between words in a sentence. 64 units or neurons were used
- Dense output layer, Sigmoid activation function was used to predict the probabilities

6 Training the model:

- binary_crossentropy was used as the loss function since it was a binary classification task
- Adam optimizer was used in the first two models and Adamax was used in the third model
- Model’s performance was tracked using accuracy

Model -1:
LSTM layer: 64 units
Return sequence: False
Loss: binary_crossentropy
Optimizer: Adam
Epochs: 5

Model -2:
LSTM layer-1: 64 units
Return sequence: True
LSTM layer-2: 64 units
Return sequence: False
Loss: binary_crossentropy
Optimizer: Adam
Epochs: 3

Model -3:
LSTM layer: 64 units
Return sequence: False
Loss: binary_crossentropy
Optimizer: Adamax
Epochs: 5

7 Evaluation:

The model was evaluated on the Test set. F-score was calculated to measure the balance between precision and recall.

Model-1:
Training accuracy: 94.24%
Validation accuracy: 88.60%
F-score: 88.57%

Model-2:

Training accuracy: 87.85%
Validation accuracy: 87.66%
F-score: 88.35%

Model-3:

Training accuracy: 92.41%
Validation accuracy: 87.90%
F-score: 88.93%

Conclusion:

While there was not any significant difference between the accuracies of the three models. The model with optimizer as **‘Adamax’ (Model-3)** gave slightly better result with a F-score of **88.93%.**