Name: Jiya Sharma
RollNo: 381056
Prn: 22311741

Assignment 6

Sentiment Analysis

Problem Statement:

Sentiment analysis using LSTM network or GRU.

Objective:

The primary objective of this practical is to develop a sentiment analysis model using LSTM or GRU networks that can accurately classify the sentiment of movie reviews as positive or negative. The specific objectives are:

- To preprocess and clean the textual data to make it suitable for modeling.
- To build and train an LSTM or GRU based deep learning model.
- To evaluate the performance of the model using accuracy and other relevant metrics.
- To compare the efficiency of LSTM and GRU for sentiment analysis.

Software Packages and Hardware Packages Used:

- Software Packages:
    - Python: A popular programming language for machine learning and NLP.
    - Jupyter Notebook: An interactive environment for coding, visualization, and analysis.
    - TensorFlow/Keras: A deep learning library used for building, training, and evaluating neural network models.
    - NLTK: For natural language processing tasks like removing stop words.
    - Scikit-Learn: For splitting the dataset and evaluating model performance.

- Hardware Packages:
    - A computer with at least 8 GB RAM for efficient training and testing.
    - GPU (Graphics Processing Unit) is recommended for faster training of the LSTM or GRU model.
    - A CPU can be used if a GPU is not available, but the training time will be longer.

Libraries Used:

- Pandas: To load and manipulate the IMDB dataset.
- NumPy: For numerical operations, such as data reshaping and mathematical calculations.
- NLTK: To clean the text data by removing stop words.

- Scikit-Learn: For splitting the dataset into training and testing sets, and for accuracy evaluation.
- TensorFlow/Keras:
  - `Sequential`: To define the deep learning model.
  - `Embedding`: For creating word embeddings from the input data.
  - `LSTM`/`GRU`: For adding the recurrent layers that process sequences.
  - `Dense`: To create fully connected layers for the output.
  - `ModelCheckpoint`: To save the best model during training.
  - re: For cleaning text using regular expressions, such as removing HTML tags and nonalphabetic characters.

Theory:

- Sentiment Analysis

Sentiment analysis is a branch of NLP that focuses on determining the emotional tone behind a body of text. It is widely used in customer reviews, social media analysis, and opinion mining. The analysis aims to classify input text into categories like positive, negative, or neutral.

- LSTM (Long ShortTerm Memory)

LSTM is a special type of RNN capable of learning longterm dependencies in sequential data. It addresses the vanishing gradient problem of traditional RNNs by using memory cells and gates:

Forget Gate: Decides which information from the previous state should be discarded.

Input Gate: Updates the cell state with new information.

Output Gate: Determines the output based on the cell state.

- GRU (Gated Recurrent Unit)

GRU is a simplified variant of LSTM that combines the forget and input gates into a single update gate. It has fewer parameters and can be faster to train while maintaining similar performance levels.

Methodology:

1. Data Loading:
   - Load the IMDB reviews dataset, which consists of reviews labeled as 'positive' or 'negative'.
2. Data Cleaning and Preprocessing:

   - Remove HTML tags, nonalphabet characters, and convert the reviews to lowercase.
   - Remove stopwords using `nltk` to reduce noise.
3. Sentiment Encoding:
   - Encode the target variable 'sentiment' as binary values, where 'positive' is 1 and 'negative' is 0.
4. Splitting Data:
   - Split the cleaned dataset into training and testing sets using an 8020 ratio.
5. Tokenization and Padding:
   - Use the Keras `Tokenizer` to convert reviews into sequences of integers.
   - Pad or truncate sequences to a fixed length to ensure uniform input size for the model.
6. Model Building:
   - Use the `Sequential` model in Keras to stack layers.
   - Add an `Embedding` layer to convert words into dense vectors.
   - Use an `LSTM` or `GRU` layer to capture sequential dependencies.
   - Add a `Dense` layer with a sigmoid activation function for binary classification.
7. Model Training:
   - Compile the model with the 'binary_crossentropy' loss function and 'adam' optimizer.
   - Train the model using the training data and validate on the test data.
   - Use `ModelCheckpoint` to save the best model during training.
8. Model Evaluation:
   - Evaluate the trained model on the test set to measure accuracy and loss.
   - Use a confusion matrix to further assess model performance.

Advantages:

- Handles Sequential Data: LSTM and GRU models are highly effective in capturing longterm dependencies in sequences.
- Robust to Noisy Data: Preprocessing steps like removing stopwords and nonalphabetic characters make the model more robust.
- Adaptability: The model can be adapted for different languages and text data with minimal adjustments.
- High Accuracy: With enough data and proper tuning, LSTM and GRU models can achieve high accuracy on sentiment analysis tasks.

Limitations:

- Computationally Expensive: Training LSTM models requires significant computational power, especially for large datasets.

- Long Training Times: Due to the complexity of the models, training time can be long without access to a GPU.
- Data Sensitivity: The performance of the model is highly sensitive to data preprocessing and hyperparameters like sequence length, learning rate, and number of LSTM units.
- Overfitting Risk: Without proper regularization, LSTM models can overfit, especially when the training data is limited.

Applications:

- Customer Review Analysis: Understanding customer feedback on products, services, or movies by classifying reviews as positive or negative.
- Social Media Monitoring: Analyzing public sentiment on platforms like Twitter to gauge reactions to events, brands, or social issues.
- Healthcare: Analyzing patient feedback and reviews of medical facilities for better service delivery.
- Financial Markets: Assessing the sentiment behind news articles or reports to predict market trends.

Working/Algorithm:

1. Import required libraries.
2. Load the IMDB dataset.
3. Preprocess the text data:

- Remove unwanted characters and convert text to lowercase.
- Remove stopwords.

4. Encode labels: Positive = 1, Negative = 0.
5. Split data: Train (80%) and Test (20%).
6. Tokenize and pad sequences.
7. Build the model:

- Embedding layer (input_dim, output_dim).
- LSTM layer with dropout for regularization.
- Dense output layer with 'sigmoid' activation.

8. Compile and train the model using 'adam' optimizer.
9. Evaluate the model on the test set.
10. Save the model for future inference.

Conclusion:

In this practical, an LSTM based model was developed for sentiment analysis of IMDB movie reviews. The model leverages the sequential nature of LSTM to capture the context in text data, enabling accurate sentiment classification. Despite requiring considerable computational resources, the model's ability to learn from sequential data makes it a powerful tool for NLP tasks. By employing proper data preprocessing and hyperparameter tuning, the model achieved satisfactory results, demonstrating its potential in realworld applications like review analysis and opinion mining.