# **Assignment No. 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.

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 and Hardware Packages**

**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.

### **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 Short-Term Memory)** LSTM is a special type of RNN capable of learning long-term 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, non-alphabet 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: positive = 1, negative = 0.
4. **Splitting Data**
   * Split the cleaned dataset into training and testing sets using an 80:20 ratio.
5. **Tokenization and Padding**
   * Use the Keras Tokenizer to convert reviews into sequences of integers.
   * Pad or truncate sequences to a fixed length.
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.
8. **Model Evaluation**
   * Evaluate the trained model on the test set.
   * Use a confusion matrix to further assess model performance.

### **Advantages**

* Handles sequential data effectively.
* Robust to noisy data after preprocessing.
* Adaptable to different languages and datasets.
* High accuracy when trained with sufficient data.

### **Limitations**

* Computationally expensive, especially with large datasets.
* Training can be time-consuming without GPU acceleration.
* Sensitive to preprocessing and hyperparameter choices.
* Risk of overfitting if not properly regularized.

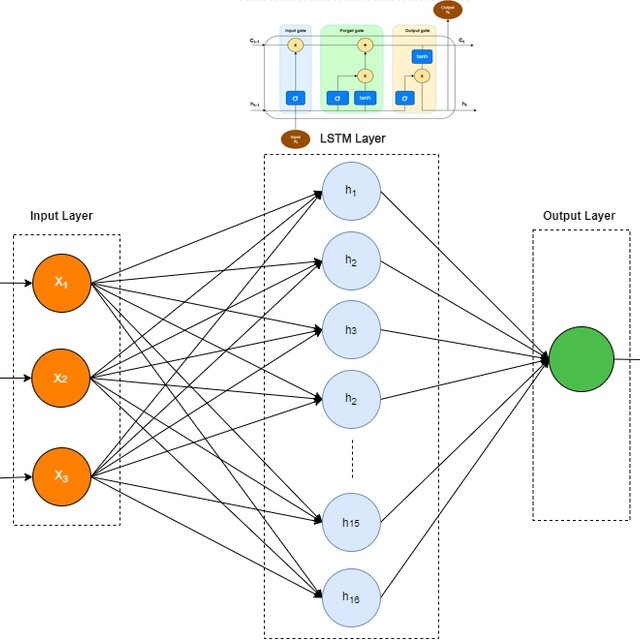
### **Applications**

* **Customer Review Analysis:** Classifying product or movie reviews.
* **Social Media Monitoring:** Analyzing sentiment on platforms like Twitter.
* **Healthcare:** Analyzing patient feedback and reviews.
* **Financial Markets:** Understanding sentiment in news articles to predict market trends.

### **Working / Algorithm**

1. Import required libraries.
2. Load the IMDB dataset.
3. Preprocess the text data:
   * Remove unwanted characters.
   * Convert text to lowercase.
   * Remove stopwords.
4. Encode labels (positive = 1, negative = 0).
5. Split data into train (80%) and test (20%).
6. Tokenize and pad sequences.
7. Build the model:
   * Embedding layer.
   * LSTM (or GRU) layer with dropout.
   * Dense output layer with sigmoid activation.
8. Compile and train the model.
9. Evaluate the model on the test set.
10. Save the model.

### **Diagram**

**

### **Conclusion**

In this practical, an LSTM-based model was developed for sentiment analysis of IMDB movie reviews. The model leveraged sequential learning to capture context in text data, enabling accurate sentiment classification. Although computationally demanding, with proper preprocessing and hyperparameter tuning the model achieved satisfactory accuracy, demonstrating its usefulness in real-world applications like review analysis, social media monitoring, and opinion mining.