**Assignment No. 6**

**Sentiment Analysis**

**Problem Statement:**

Sentiment analysis using LSTM network or GRU.

**Objective:**

1. **Preprocessing**: Clean and preprocess the textual data to make it suitable for model training.
2. **Model Building**: Develop and train a deep learning model based on LSTM or GRU for sentiment analysis.
3. **Evaluation**: Assess the model's performance using metrics such as accuracy, precision, recall, and F1-score.
4. **Comparison**: Compare the effectiveness of LSTM and GRU models for sentiment classification.

**Software and Hardware Packages:**

* **Software**:
  + Python (with libraries like TensorFlow/Keras, NLTK, and Scikit-learn).
  + Jupyter Notebook for interactive coding.
  + TensorFlow/Keras for building the deep learning models.
  + NLTK for text preprocessing (stopwords removal).
  + Scikit-learn for splitting datasets and evaluation.
* **Hardware**:
  + Minimum 8GB RAM.
  + GPU (for faster training), though CPU can also be used with longer training times.

**Libraries:**

* **Pandas**: Data manipulation and loading.
* **NumPy**: Numerical operations for data handling.
* **NLTK**: Natural Language Toolkit for text preprocessing.
* **Scikit-learn**: Train-test splitting and performance evaluation.
* **TensorFlow/Keras**: Deep learning framework.

**Theory:**

**Sentiment Analysis:**

Sentiment analysis is a Natural Language Processing (NLP) task that classifies the polarity (positive, negative, or neutral) of text. It is widely applied to review analysis, customer feedback, and opinion mining.

**LSTM (Long Short-Term Memory):**

LSTM networks are a type of Recurrent Neural Network (RNN) designed to remember long-term dependencies in sequential data. LSTM uses memory cells and gates (forget, input, output) to regulate information flow, addressing the vanishing gradient problem.

**GRU (Gated Recurrent Unit):**

GRU is a simplified version of LSTM that combines the forget and input gates into a single "update" gate, making it faster and easier to train. Despite its simplicity, GRU performs comparably to LSTM in many tasks.

**Methodology:**

1. **Data Loading**:
   * Load the **IMDB reviews** dataset, which consists of labeled reviews (positive/negative).
2. **Data Cleaning and Preprocessing**:
   * Convert text to lowercase, remove HTML tags, and filter out non-alphabetic characters using regular expressions.
   * Remove stopwords (like "the", "is", etc.) to reduce noise using NLTK.
3. **Label Encoding**:
   * Encode sentiments as binary values: Positive = 1, Negative = 0.
4. **Train-Test Split**:
   * Split the dataset into **training** (80%) and **testing** (20%) sets.
5. **Tokenization and Padding**:
   * Convert text into sequences of integers using Keras' Tokenizer.
   * Apply padding/truncation to ensure uniform sequence length across inputs.
6. **Model Building**:
   * Define a sequential model with an Embedding layer to convert words into dense vectors.
   * Add LSTM/GRU layers for processing sequential data.
   * Add a Dense layer with sigmoid activation for binary classification.
7. **Model Training**:
   * Compile the model using binary\_crossentropy as the loss function and adam optimizer.
   * Train the model on the training dataset, using early stopping or checkpointing to save the best model.
8. **Evaluation**:
   * Evaluate the model on the test set using accuracy and loss.
   * Use a confusion matrix to visualize performance metrics like precision, recall, and F1-score.

**Advantages:**

* **Effective on Sequential Data**: LSTM and GRU handle sequential patterns and long-term dependencies well.
* **Adaptable**: The model can be applied to various text data and languages.
* **High Accuracy**: Deep learning models like LSTM and GRU, when tuned properly, achieve high accuracy.

**Limitations:**

* **Training Time**: LSTM models can take a long time to train, especially on large datasets.
* **Sensitive to Preprocessing**: The performance of LSTM/GRU models is sensitive to the quality of data preprocessing.

**Applications:**

* **Customer Review Analysis**: Understanding customer feedback (positive/negative) through review classification.
* **Social Media Analysis**: Analyzing sentiment on platforms like Twitter and Facebook to gauge public opinion.
* **Market Analysis**: Sentiment analysis of financial reports or news articles to predict market behavior.

**Working Steps/Algorithm:**

1. **Import Required Libraries**:

import pandas as pd

import numpy as np

import nltk

import tensorflow as tf

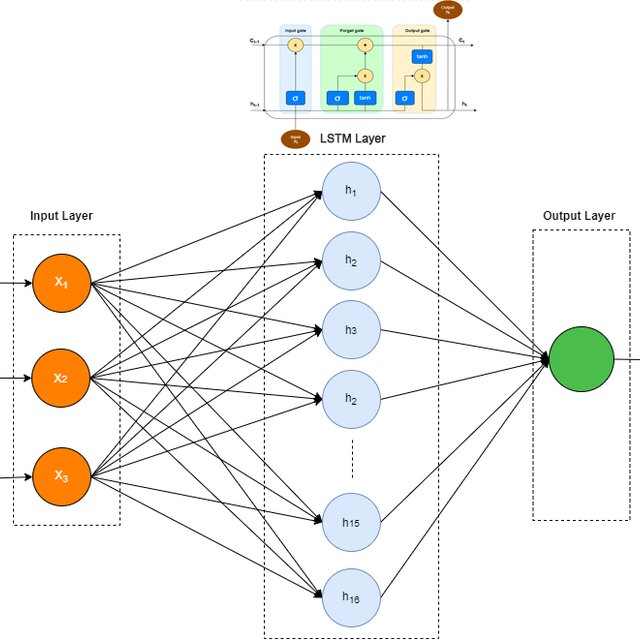
from sklearn.model\_selection import train\_test\_split

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

1. **Data Loading**:
   * Load the IMDB movie review dataset and display a few rows.
2. **Preprocessing**:
   * Clean the text (remove stop words, convert to lowercase, etc.).
3. **Train-Test Split**:
   * Use train\_test\_split() to create training and test datasets.
4. **Tokenization**:
   * Convert the text into numerical sequences and pad them to ensure uniform input size.
5. **Model Building**:
   * Build and compile the LSTM/GRU model.
6. **Training**:
   * Train the model on the training dataset with validation on the test dataset.
7. **Evaluation**:
   * Evaluate the model using test accuracy, loss, and a confusion matrix.

**Diagram:**



**Conclusion:**

This project demonstrated the effectiveness of LSTM and GRU networks in sentiment analysis tasks. By preprocessing text data and utilizing deep learning techniques, we can achieve high accuracy in classifying movie reviews. While training time can be long without a GPU, the sequential modeling capabilities of LSTM and GRU make them powerful tools in natural language processing tasks such as sentiment analysis.