**Assignment No. 6**

**Sentiment Analysis**

**Problem Statement:**

Sentiment analysis using LSTM network or GRU.

**Objectives:**

1. To implement a sentiment analysis system using Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks.
2. To classify the sentiment (positive, negative, or neutral) of text data such as reviews, social media posts, or comments.
3. To understand how LSTM and GRU networks handle sequential text data and their effectiveness in analyzing sentiment by capturing contextual dependencies in the input text.

**Theory:**

Sentiment analysis, also known as opinion mining, is the process of determining the sentiment expressed in a piece of text—whether it is positive, negative, or neutral. It is an important task in natural language processing (NLP) and has applications in areas like customer feedback analysis, social media monitoring, and product reviews.

**LSTM (Long Short-Term Memory)** and **GRU (Gated Recurrent Unit)** are advanced types of Recurrent Neural Networks (RNNs) specifically designed to handle long-term dependencies in sequences. They are used in NLP tasks because they can learn from sequences of text, remembering past data in order to understand the context of words in sentences. LSTMs and GRUs address the vanishing gradient problem in traditional RNNs, making them more effective for tasks like sentiment analysis, where context and the order of words are critical.

**Methodology:**

1. **Data Collection**:
   * Obtain a labeled dataset of text samples (e.g., movie reviews, tweets) with corresponding sentiment labels (positive, negative, or neutral).
2. **Data Preprocessing**:
   * Tokenize the text data: Convert sentences into sequences of words or word embeddings (e.g., using Word2Vec or GloVe).
   * Padding: Ensure all sequences have the same length by padding shorter sequences or truncating longer ones.
   * Split the dataset into training, validation, and test sets.
3. **Model Design**:
   * Choose between an LSTM or GRU model for the recurrent layers.
   * Build the model with an embedding layer (to convert words to vectors), one or more LSTM or GRU layers, and a dense layer for output.
4. **Training**:
   * Train the model using a suitable loss function (e.g., categorical cross-entropy for multi-class sentiment) and optimizer (e.g., Adam).
   * Use early stopping or model checkpoints to prevent overfitting.
5. **Evaluation**:
   * Evaluate the model on the test set using metrics such as accuracy, precision, recall, and F1-score to measure sentiment classification performance.
6. **Deployment**:
   * Deploy the trained model to classify sentiment in new text data (e.g., real-time social media sentiment tracking).

**Working Principle / Algorithm:**

1. **Input**:
   * Input sequences of text data are fed into the model. Each sequence consists of words or word embeddings.
2. **Embedding Layer**:
   * Converts words into dense vector representations that capture semantic meaning.
3. **LSTM or GRU Layers**:
   * Processes the sequences and retains the context of previous words through memory units (LSTM cells or GRU gates).
   * These layers capture the order of words and longer-term dependencies in the text, allowing the model to understand the overall sentiment.
4. **Dense Output Layer**:
   * The output from the LSTM/GRU layers is passed through a dense layer with a softmax activation function to classify the text into sentiment categories (e.g., positive, negative, or neutral).
5. **Backpropagation**:
   * The model adjusts its weights during training using backpropagation to minimize the classification error on the training data.
6. **Output**:
   * The final output is a sentiment label (positive, negative, or neutral) with a corresponding confidence score.

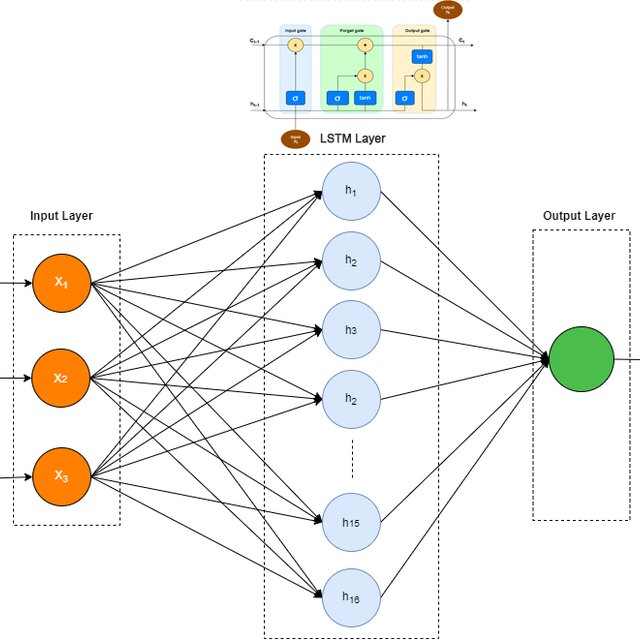
**Advantages:**

1. **Effective Context Capture**: LSTM and GRU models excel at capturing the context and meaning of words based on their surrounding text, making them ideal for tasks like sentiment analysis, where word order is important.
2. **Handling Long-Term Dependencies**: Both LSTM and GRU networks are designed to remember long-term dependencies in text, which is crucial for understanding complex sentences with shifting sentiments.
3. **Prevents Vanishing Gradient Problem**: These networks address the vanishing gradient problem, which is common in traditional RNNs when learning from long sequences of text.
4. **Versatility**: LSTM and GRU models can be trained on various types of text data and are effective across multiple languages and domains.

**Disadvantages / Limitations:**

1. **High Computational Cost**: LSTM and GRU networks require more computational resources and time for training compared to simpler models like logistic regression or support vector machines (SVMs).
2. **Requires Large Datasets**: To achieve good performance, these models often need large amounts of labeled data for training, which may not always be available.
3. **Tuning Complexity**: LSTM and GRU networks have many hyperparameters (e.g., number of layers, hidden units, learning rate) that need careful tuning for optimal performance.
4. **Overfitting Risk**: If the model is too complex or the training data is limited, there is a risk of overfitting, where the model performs well on training data but poorly on unseen test data.

**Diagram:**



**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.