**Assignment No. 6**

**Sentiment Analysis using LSTM and GRU**

Name: Shubham Gautam Dahane

PRN: 22310312

Roll: 382015

**Problem Statement:**

Perform sentiment analysis on airline tweets using deep learning techniques. Implement both LSTM and GRU based recurrent neural networks to classify tweets into **negative, neutral, or positive** categories and compare their performance.

**Objective:**

The primary objective of this practical is to develop a sentiment analysis model using LSTM and GRU networks that can accurately classify the sentiment of airline-related tweets. The specific objectives are:

* To preprocess and clean the textual tweet data.
* To build and train deep learning models using LSTM and GRU.
* To evaluate the performance of the models using accuracy, confusion matrix, and classification reports.
* To compare the efficiency of LSTM and GRU for sentiment analysis tasks.

**Software and Hardware Requirements:**

**Software Packages:**

* **Python**: Programming language used for model building and analysis.
* **Jupyter Notebook / IDE**: For development and testing.
* **TensorFlow / Keras**: For building and training LSTM/GRU models.
* **Scikit-learn**: For splitting dataset and evaluating performance.
* **Seaborn & Matplotlib**: For data visualization and plotting confusion matrices.
* **Regex (re)**: For text cleaning.

**Hardware Requirements:**

* System with at least 8 GB RAM.
* GPU recommended for faster training, though CPU can also be used with longer training times.

**Libraries Used:**

* **Pandas**: Data manipulation and cleaning.
* **NumPy**: Mathematical operations.
* **Scikit-Learn**: Data splitting, metrics.
* **Matplotlib/Seaborn**: Visualization of performance and confusion matrices.
* **TensorFlow/Keras**:
  + Embedding: Word embedding representation.
  + LSTM & GRU: Sequence modeling layers.
  + Bidirectional: To capture both forward and backward dependencies.
  + Dense: Fully connected classification layers.
  + Dropout: Regularization to reduce overfitting.
  + EarlyStopping: Callback to stop training at optimal point.

**Theory:**

**Sentiment Analysis**:  
A natural language processing (NLP) task that classifies text into categories such as positive, negative, or neutral. It is widely used for analyzing reviews, tweets, and public opinions.

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

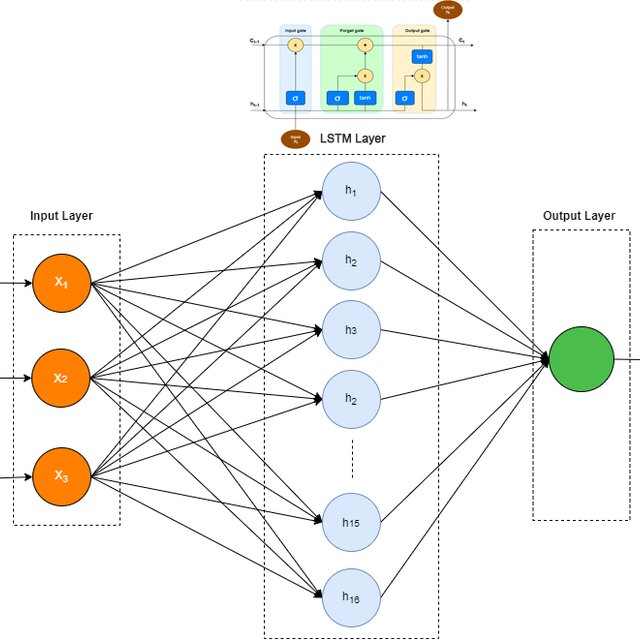
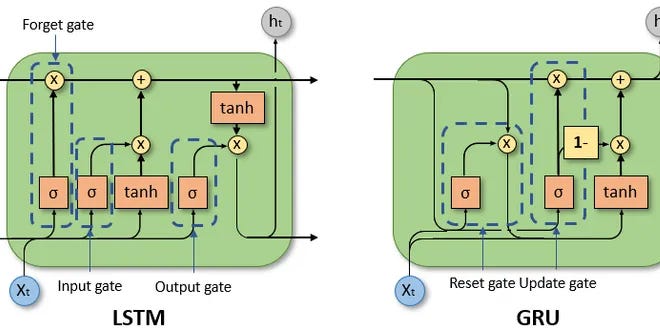
* A type of RNN designed to overcome the vanishing gradient problem.
* Contains **Forget Gate, Input Gate, Output Gate** to decide which information to keep or discard.
* Effective at learning long-term dependencies in text sequences.

**GRU (Gated Recurrent Unit)**:

* A simplified variant of LSTM.
* Combines forget and input gates into a single **update gate**.
* Requires fewer parameters, thus faster to train, while often achieving similar accuracy.

**Methodology:**

1. **Data Loading**:  
   Load Tweets dataset (Tweets.csv) containing tweets with sentiment labels (negative, neutral, positive).
2. **Data Cleaning & Preprocessing**:
   * Convert text to lowercase.
   * Remove links, special characters, and extra spaces using regex.
3. **Sentiment Encoding**:
   * Encode labels:
     + Negative → 0
     + Neutral → 1
     + Positive → 2
4. **Tokenization & Padding**:
   * Convert words into integer sequences using Keras Tokenizer.
   * Apply pad\_sequences to ensure all sequences are of fixed length (100 tokens).
5. **Train-Test Split**:
   * Split dataset into **80% training, 20% testing**.
6. **Model Building**:
   * **LSTM Model**:
     + Embedding Layer (128 dimensions).
     + Bidirectional LSTM (128 units).
     + Dense (64, ReLU) + Dropout (0.5).
     + Dense output layer (3 classes, Softmax).
   * **GRU Model**:
     + Same architecture as LSTM, but with Bidirectional GRU layer instead.
7. **Model Training**:
   * Loss function: sparse\_categorical\_crossentropy.
   * Optimizer: Adam.
   * Epochs: 10, Batch Size: 128.
   * EarlyStopping used to avoid overfitting.
8. **Evaluation**:
   * Metrics: Accuracy, Precision, Recall, F1-score.
   * Confusion Matrix for detailed class-level performance.
   * Training/Validation Accuracy & Loss comparison plotted.



**Results:**

**LSTM Model Performance:**

* Accuracy: **80.36%**
* Precision, Recall, F1 (average): ~0.74
* Performs well on Negative class, moderate on Neutral & Positive.

**GRU Model Performance:**

* Accuracy: **80.49%**
* Precision, Recall, F1 (average): ~0.74
* Comparable results to LSTM, slightly faster training.

**Comparison:**

* Both models achieved similar accuracy (~80%).

A graph of a line

AI-generated content may be incorrect.A graph of a line

AI-generated content may be incorrect.

* GRU trained slightly faster with fewer parameters.

A graph of loss comparison

AI-generated content may be incorrect.

* Confusion matrices show misclassifications mainly between Neutral and Positive classes.

A graph with numbers and a blue box

AI-generated content may be incorrect.A graph with numbers and a blue box

AI-generated content may be incorrect.

**Advantages:**

* LSTM and GRU effectively capture sequential patterns in tweets.
* Robust preprocessing reduces noise and improves model reliability.
* Achieved high classification accuracy with limited preprocessing.

**Limitations:**

* Training is computationally expensive on CPU.
* Neutral sentiment classification is more challenging due to ambiguity.
* Requires careful hyperparameter tuning (units, dropout, embedding size).

**Applications:**

* Airline industry: Monitoring customer feedback.
* Social media analysis: Tracking public sentiment.
* Product review classification.
* Opinion mining for decision-making.

**Working/Algorithm (Summary):**

1. Import libraries.
2. Load dataset (Tweets.csv).
3. Preprocess text (lowercase, remove noise).
4. Encode labels (0 = Negative, 1 = Neutral, 2 = Positive).
5. Tokenize and pad sequences.
6. Build models (LSTM & GRU).
7. Compile & train models.
8. Evaluate using classification report and confusion matrix.
9. Compare performance of LSTM vs GRU.

**Conclusion:**

In this practical, sentiment analysis of airline tweets was performed using both **LSTM** and **GRU** networks. Both models achieved comparable performance (~80% accuracy), with GRU showing slightly better efficiency in training. This demonstrates that GRUs can be a faster alternative to LSTMs for sentiment analysis without significant accuracy loss.

The experiment highlights the importance of proper preprocessing, sequential modeling, and evaluation for NLP tasks. Such models can be effectively applied to **customer feedback an**