**Name:** Pushkar Charkha  
**Roll No:** 381002  
**PRN:** 22310023

**Assignment No: -** 6  
**Title: -** Sentiment Analysis using LSTM Network

**Problem Statement:**

Implement **Sentiment Analysis** on movie reviews using **LSTM (Long Short-Term Memory) network or GRU (Gated Recurrent Unit)** for text classification. The model should classify sentiments as **Positive** or **Negative** based on review text.

**Objective:**

* To preprocess text data for NLP tasks.
* To build and train an **LSTM-based deep learning model** for binary classification.
* To analyze performance using accuracy, precision, and loss curves.
* To predict sentiment on example reviews.

**S/W Packages and H/W apparatus used:**

* **Operating System:** Windows/Linux/MacOS
* **Kernel:** Python 3.x
* **Tools:** Google Colab / Jupyter Notebook
* **Libraries:** TensorFlow, Keras, NumPy, Matplotlib
* **Hardware:** CPU/GPU

**Theory:**

**Sentiment Analysis:**

* A Natural Language Processing (NLP) technique to determine whether a review is **positive or negative**.
* Widely used in product reviews, feedback analysis, and social media monitoring.

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

* A type of Recurrent Neural Network (RNN) capable of learning **long-term dependencies**.
* Useful for sequential data like text, speech, and time series.
* LSTM uses gates (input, forget, output) to control memory flow.

**GRU (Gated Recurrent Unit):**

* A simplified version of LSTM.
* Uses fewer parameters but provides similar performance.

**Methodology:**

1. **Dataset Loading:** IMDb dataset (50,000 movie reviews).
2. **Preprocessing:**
   * Limit vocabulary to top 10,000 words.
   * Pad sequences to fixed length (200 tokens).
3. **Model Design:**
   * Embedding Layer: Converts words into dense vectors.
   * LSTM Layer: Captures sequential dependencies.
   * Dense Layer: Sigmoid activation for binary classification.
4. **Compilation:** Optimizer = Adam, Loss = Binary Crossentropy, Metric = Accuracy.
5. **Training:** Train for 3 epochs with batch size 64.
6. **Evaluation:** Test dataset accuracy measured.
7. **Visualization:** Accuracy and loss curves plotted.
8. **Prediction:** Example reviews classified as Positive/Negative.

**Results:**

* **Training & Validation Accuracy:** Improved with each epoch.
* **Test Accuracy:** 84.35%
* **Correct Predictions:** 21088
* **Wrong Predictions:** 3912
* **Sample Predictions:**
  + *“This movie was fantastic! I loved every minute of it.” → Positive*
  + *“The plot was confusing and the acting was terrible.” → Negative*
  + *“It was an okay movie, nothing special but not bad either.” → Negative*

**Advantages:**

* Captures **long-term word dependencies**.
* Works well for large text datasets.
* Can generalize sentiment polarity in unseen reviews.

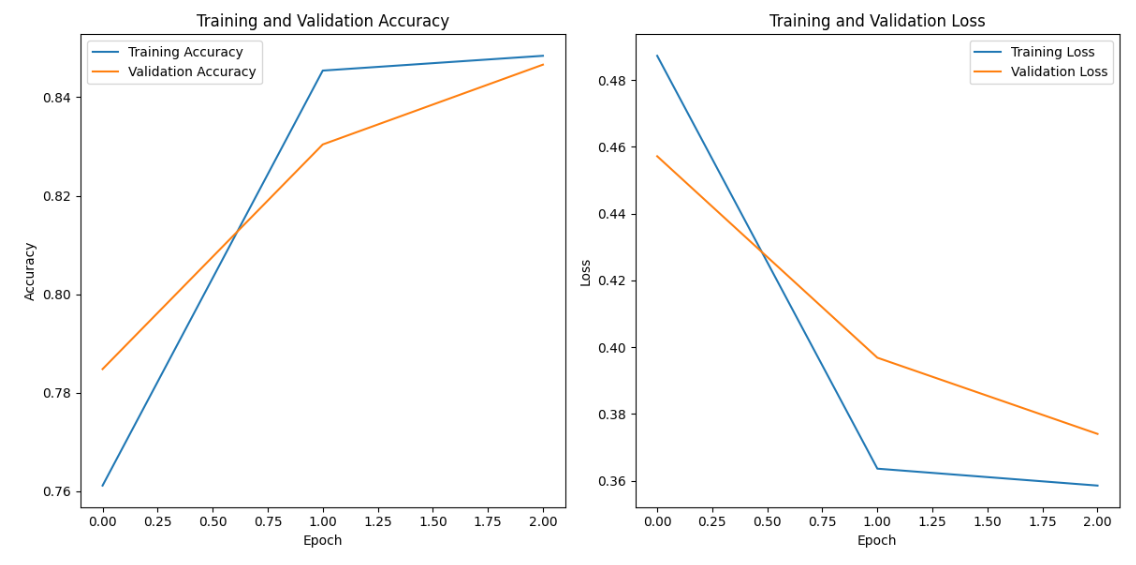
**Limitations:**

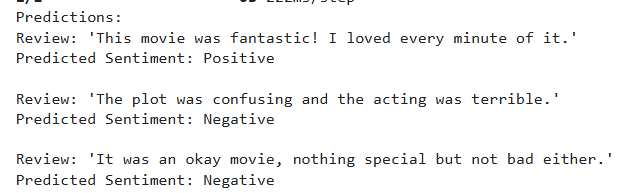
* Training is **time-consuming**.
* Requires large memory & computational power.
* Sensitive to **sequence length and preprocessing**.

**Applications:**

* Customer feedback analysis.
* Social media sentiment monitoring.
* Product rating prediction.
* Market research and brand monitoring.

**Results:**

****

****

**Conclusion:**

Sentiment analysis using **LSTM network** was successfully implemented. The model achieved **84% accuracy** on IMDb dataset and correctly predicted sentiments of sample reviews. The system demonstrates the power of deep learning in NLP tasks and can be extended with **GRU, BiLSTM, or Transformer-based models** for better performance.