**Experiment 7**

**Aim: Design and implement LSTM model with tensorflow/keras and check accuracy**

**Description:**

### ****Part 1: Sequence Classification using LSTM****

#### ****Objective****

To implement a basic LSTM-based model for binary sequence classification using synthetic data. The goal is to demonstrate how an LSTM processes sequential inputs and classifies them into one of two categories.

#### ****Description****

This program uses an LSTM neural network to classify sequences of random numerical data into binary categories. The architecture includes:

* **Input Layer:** Accepts sequences of shape (10, 1).
* **LSTM Layer:** A recurrent layer with 32 units, designed to extract sequential patterns.
* **Dense Output Layer:** A single neuron with a sigmoid activation function to output probabilities for binary classification.

#### ****Dataset****

* **Input Data:** Randomly generated numerical sequences of size (100, 10, 1).
* **Labels:** Random binary labels (0 or 1).

#### ****Training Configuration****

* **Loss Function:** Binary crossentropy, measuring the error in binary classification tasks.
* **Optimizer:** Adam, ensuring efficient and adaptive learning.
* **Batch Size:** 32.
* **Epochs:** 10.
* **Metric:** Accuracy to monitor training progress.

#### ****Limitations****

* The use of synthetic random data does not allow the model to learn meaningful patterns.
* Constant accuracy (~54%) indicates the need for a real-world dataset to evaluate the model's capability.

### ****Program 2: IMDB Sentiment Analysis****

#### ****Objective****

To develop a sentiment analysis model that classifies movie reviews as either positive or negative using the IMDB dataset.

#### ****Description****

The program utilizes an LSTM-based deep learning model for binary sentiment classification. It involves:

* **Preprocessing:** Text data is converted into sequences of integers representing words and padded to a uniform length of 100 words.
* **Embedding Layer:** Maps words to 128-dimensional dense vectors.
* **LSTM Layer:** Processes the sequential data with 64 units, capturing temporal dependencies and contextual information.
* **Dense Output Layer:** A single neuron with a sigmoid activation function for binary classification.

#### ****Dataset****

* **IMDB Dataset:** Comprises 25,000 training samples and 25,000 test samples.
* **Features:** Reviews are tokenized and padded to ensure uniformity.
* **Labels:** Binary labels indicating positive (1) or negative (0) sentiments.

#### ****Training Configuration****

* **Loss Function:** Binary crossentropy, suitable for binary classification.
* **Optimizer:** Adam, providing adaptive learning rates.
* **Batch Size:** 32.
* **Epochs:** 5.
* **Metrics:** Accuracy for evaluation.

#### ****Performance****

* Training accuracy improved from 81.32% to 97.35%.
* Validation accuracy stabilized around 83%-85%, indicating good generalization.
* Test accuracy: 83.18%.

#### ****Limitations****

* Slight overfitting observed as validation accuracy plateaus while training accuracy continues to improve.
* Model performance on unseen data could be improved with more regularization or additional data.

**RESULT ANALYSIS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hyperparamter** | **Value** | **Training Loss** | **Validation Loss** | **Training Accuracy** | **Validation Accuracy** | **Analysis** |
| **LSTM Units** | 32 | 0.0752 | 0.6073 | 97.35% | 83.18% | Moderate performance. Larger number of units may capture more patterns but increase complexity. |
|  | 64 |  |  |  |  |  |
|  | 128 |  |  |  |  |  |
| **Learning Rate** | 0.001 (default) | 0.0752 | 0.6073 | 97.35% | 83.18% | Default value; balanced training. |
|  | 0.0005 |  |  |  |  |  |
|  | 0.005 |  |  |  |  |  |
| **Batch Size** | 32 | 0.0752 | 0.6073 | 97.35% | 83.18% | Smaller batches stabilize learning but slow training time. |
|  | 64 |  |  |  |  |  |
| **Dropout Rate** | 0.2 (default) | 0.0752 | 0.6073 | 97.35% | 83.18% | Default setting; balanced regularization. |
|  | 0.3 |  |  |  |  |  |
| **Optimizer** | Adam | 0.0752 | 0.6073 | 97.35% | 83.18% | Default optimizer; good convergence speed and balanced performance. |
|  | SGD |  |  |  |  |  |
|  | RMSprop |  |  |  |  |  |
|  | Adagrad |  |  |  |  |  |
|  | Adamax |  |  |  |  |  |

**Conclusion:**