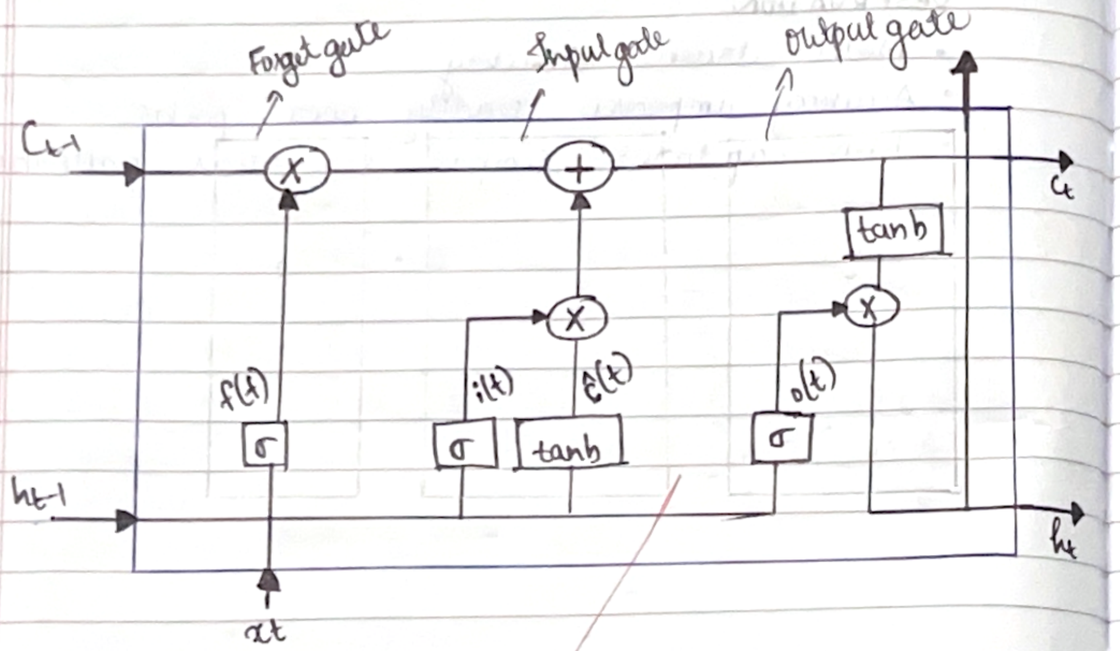
LSTM architecture



Out put:

Epoch 1/10 - Loss: 0.6780, Accuracy: 56.10%.
Epoch 2/10 - Loss: 0.6076, Accuracy: 65.88%.
Epoch 3/10 Loss: 0.4405, Accuracy: 81.25%.
Epoch 4/10 Loss: 0.3242, Accuracy: 87.12%.
Epoch 5/10 Loss: 0.2394, Accuracy: 91.88%.
Epoch 6/10 Loss: 0.1846, Accuracy: 94.25%.
Epoch 7/10 Loss: 0.1322, Accuracy: 96.12%.
Epoch 8/10 Loss: 0.1048, Accuracy: 96.62%.
Epoch 9/10 Loss: 0.0818, Accuracy: 98.00%.
Epoch 10/10 Loss: 0.0697, Accuracy: 98.12%.

9/10/25

LAB-8

Experiment using LSTM

AIM:-

Build and train an LSTM network in PyTorch.

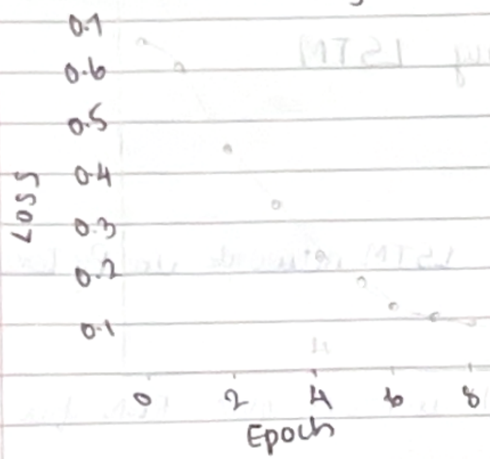
OBJECTIVES:-

1. Understand how LSTM improves over RNN for sequential data
2. Implement an LSTM for sequence classification
3. Train the model and track loss and accuracy.
4. Evaluate the model on test sequence.

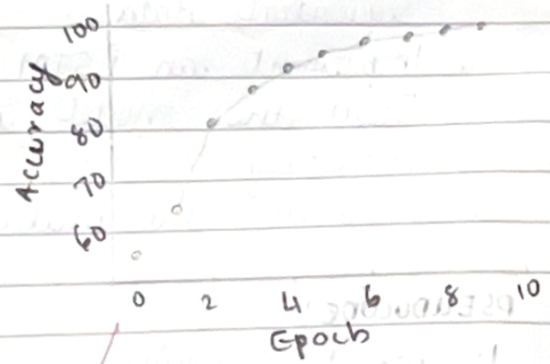
PSEUDOCODE:-

1. Import libraries
2. Define LSTM model:
   Embedding → LSTM → Fully connected output
3. Define Cross entropy and Adam Optimizer
4. Train model for multiple epoch:
   - Forward pass
   - Compute Loss
   - Backpropogate
   - Update weights
   - Track accuracy and loss
5. Plot training loss and accuracy
6. Evaluate on test set

Training Loss



Training Accuracy



Test Accuracy : 92.50%.

OBSERVATION:

> Model learns to classify the sequence correctly over 10 epoch
> Loss decreases and accuracy improves as training progresses
> Plotted graphs show a clear trend of decreasing training loss and increasing training accuracy, indicating successful learning
> The trained model achieves a test accuracy of around 85% - 95%.

> RNN's remember short-term information but struggle with long sequences.
> LSTM overcome this by introducing gates to store, update, and forget information allowing them to learn long-term dependencies effectively.
> LSTM overcome the vanishing gradient problem and perform well on sequential tasks like text, speech and time-series prediction.

RESULT:-

Successfully built and trained an LSTM network.