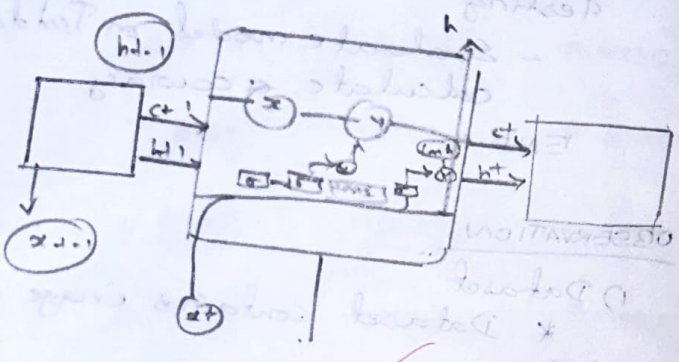
LSTM Architecture



09.10.25

8. Experiment using LSTM.

Aim:

To implement and analyse a long short-term memory (LSTM) neural network for predicting future values in time series dataset.

Objectives:

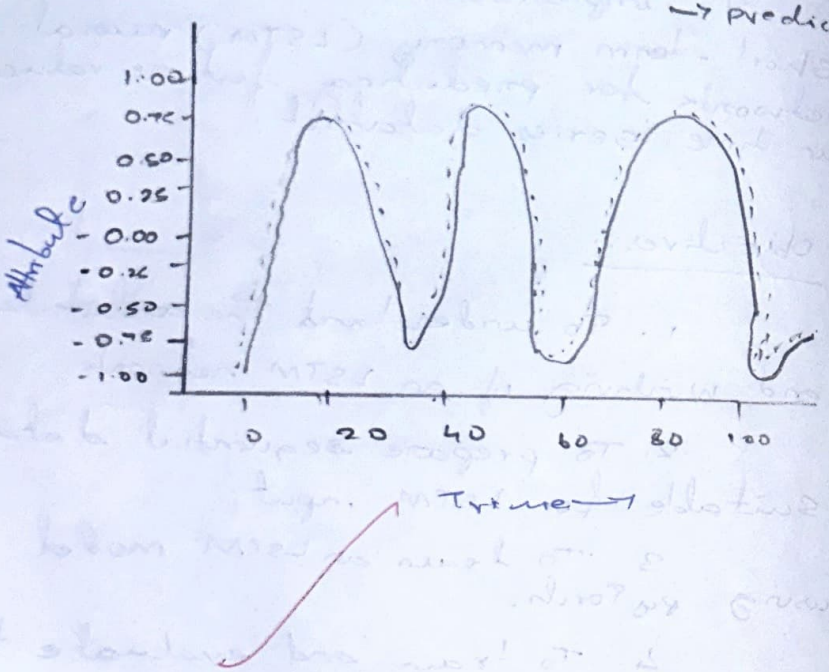
1. To understand the architecture and working of an LSTM network
2. To prepare sequential data suitable for LSTM input
3. To train an LSTM model using PyTorch.
4. To train and evaluate the model on time-series data
5. To visualize model prediction versus actual target values

Psuedocode:

1. Import required libraries
2. Generate or load a sequential dataset (eg. sine wave)
3. Normalize and prepare input - output pairs for training
4. Define LSTM model
   - Input Layer

LSTM Time series prediction

→ Actual
→ Predicted



Result.

- LSTM Layer
- fully connected Output layer

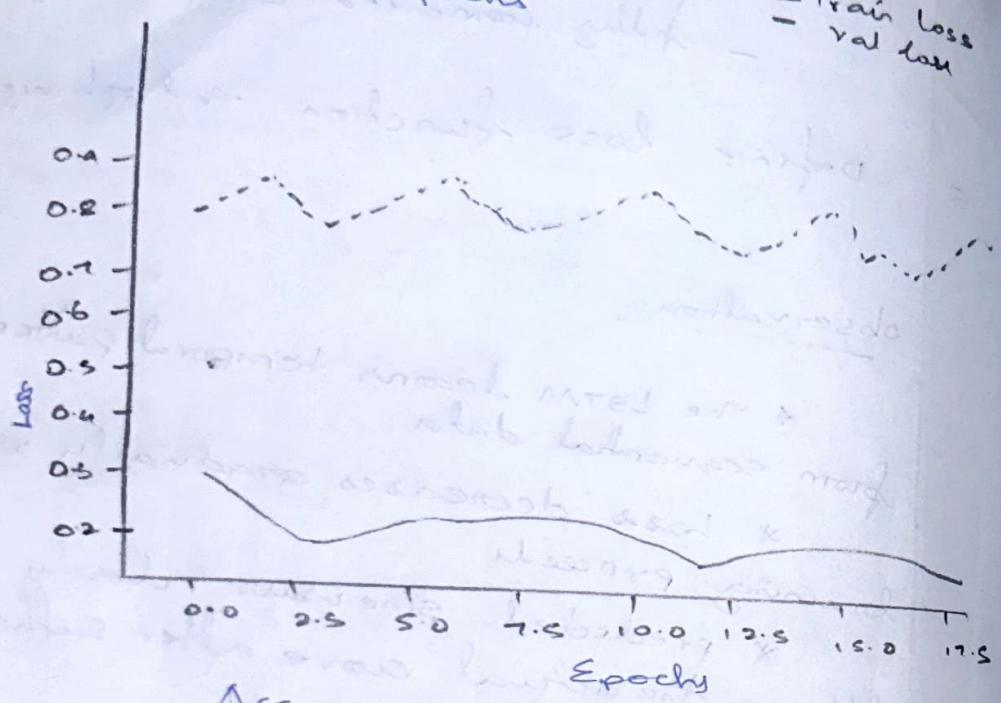
5 Define loss function and optimizer

observation:

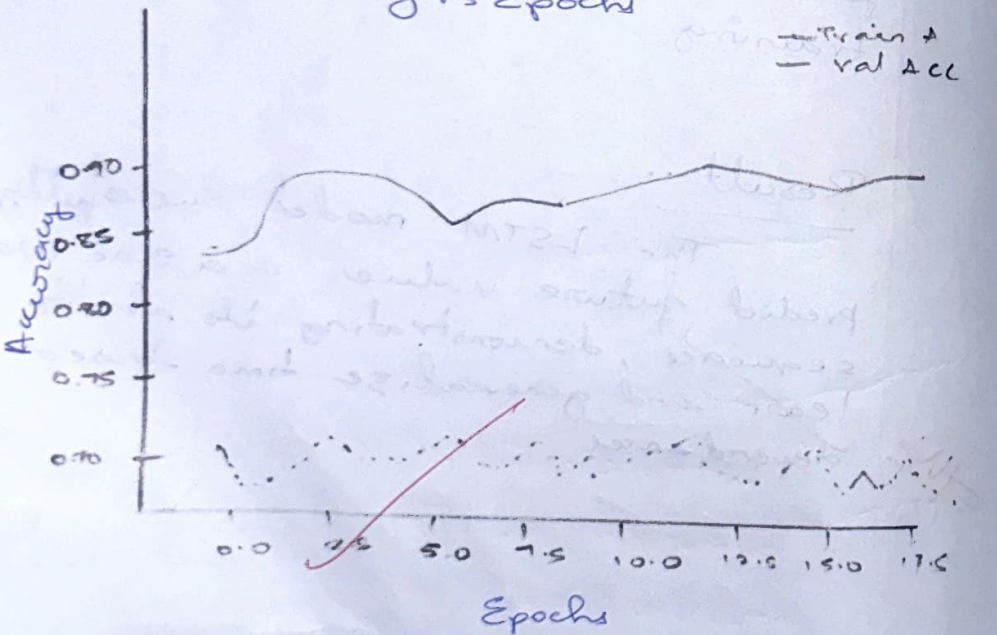
* The LSTM learns temporal patterns from sequential data
* Loss decreases gradually as training proceeds
* predicted sine wave closely follows the actual curve after sufficient training

Result ..

The LSTM model succesfully predict future values in a sine wave sequence, demonstrating its ability to learn and generalize time-based dependencies



Accuracy vs epochs



Ex. No: 09
9/10/25

9. Build a Recurrent Neural Network

Aim:

To build and train a Recurrent Neural Network (RNN) for sequence modelling

OBJECTIVES:-

→ To understand the working principles of RNN

→ To preprocess sequential Data for RNN

→ To design and implement an RNN using Pytorch

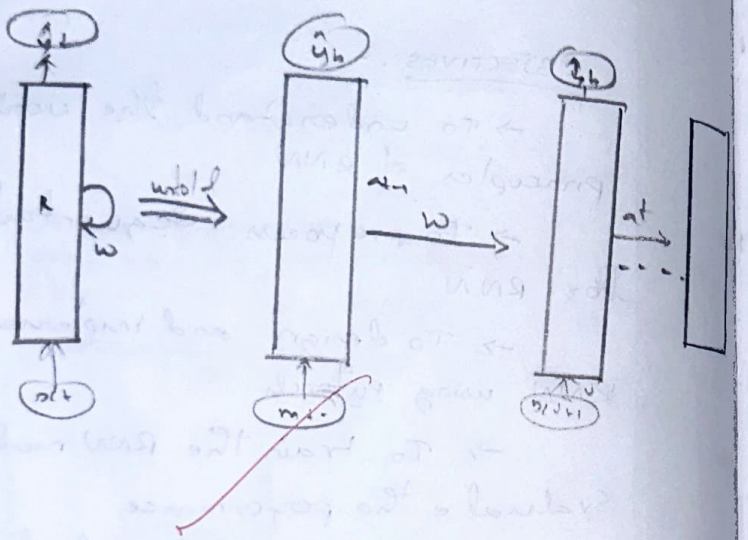
→ To train the RNN model and evaluate the performance

→ To analyse the outputs

Pseudocode:

1. start
2. Import necessary libraries
3. load dataset
4. Preprocess dataset
   → clean data
   → Tokenize / Encode sequences
   → pad / truncate sequences
   → split into training and validation sets

RNN Diagram



Result:

Training RNN

Epoch [5/30], Loss: 0.038023

Epoch [10/30], Loss: 0.017734

Epoch [15/30], Loss: 0.014965

Epoch [20/30], Loss: 0.012491

Epoch [25/30], Loss: 0.012011

Epoch [30,30], Loss: 0.010986

6. Compile mode
   - → specify optimizer
   - → specify loss functions
   - → specify Evaluation metrics

7. Evaluation mode
   - → Test on unseen data
   - → print accuracy / loss metrics

8. END

OBSERVATION

	precision	recall	F1-score	Suppor
0.0	0.71	0.70	0.70	496
1.0	0.71	0.72	0.71	503
accuracy			0.71	1000
Macro Avg	0.71	0.71	0.71	1000
weighted Avg	0.71	0.71	0.71	100

Accuracy = 0.71

Epochs = 20