

RA2311047010016

S.No	Date	Title	Signature
9.	30.09.2025	Building a RNN	<del>30/9</del> write neatly
8	09.10.2025	Long Short Term Memory	<del>9/10/25</del>
10.	17.10.2025	perform compression using airist dataset using auto encoders	
11.	17.10.2025	Experiment using Variations (VAE)	
12	27.10.2025	Implement a Deep convolutional GAN to generate color Images	copy
13	27.10.2025	Understanding the Architecture of a pre-trained model	
14	27.10.2025	Implement a pre-trained CNN model as a feature extractor	
15	27.10.2025	Implement a Yolo model object detection	copy

Completed  
3/11/25

Lab 8 · Long Short Term Memory.

Aim: To implement and train long short term memory network using pytorch by classifying the number sequences as either increasing or decreasing

objective

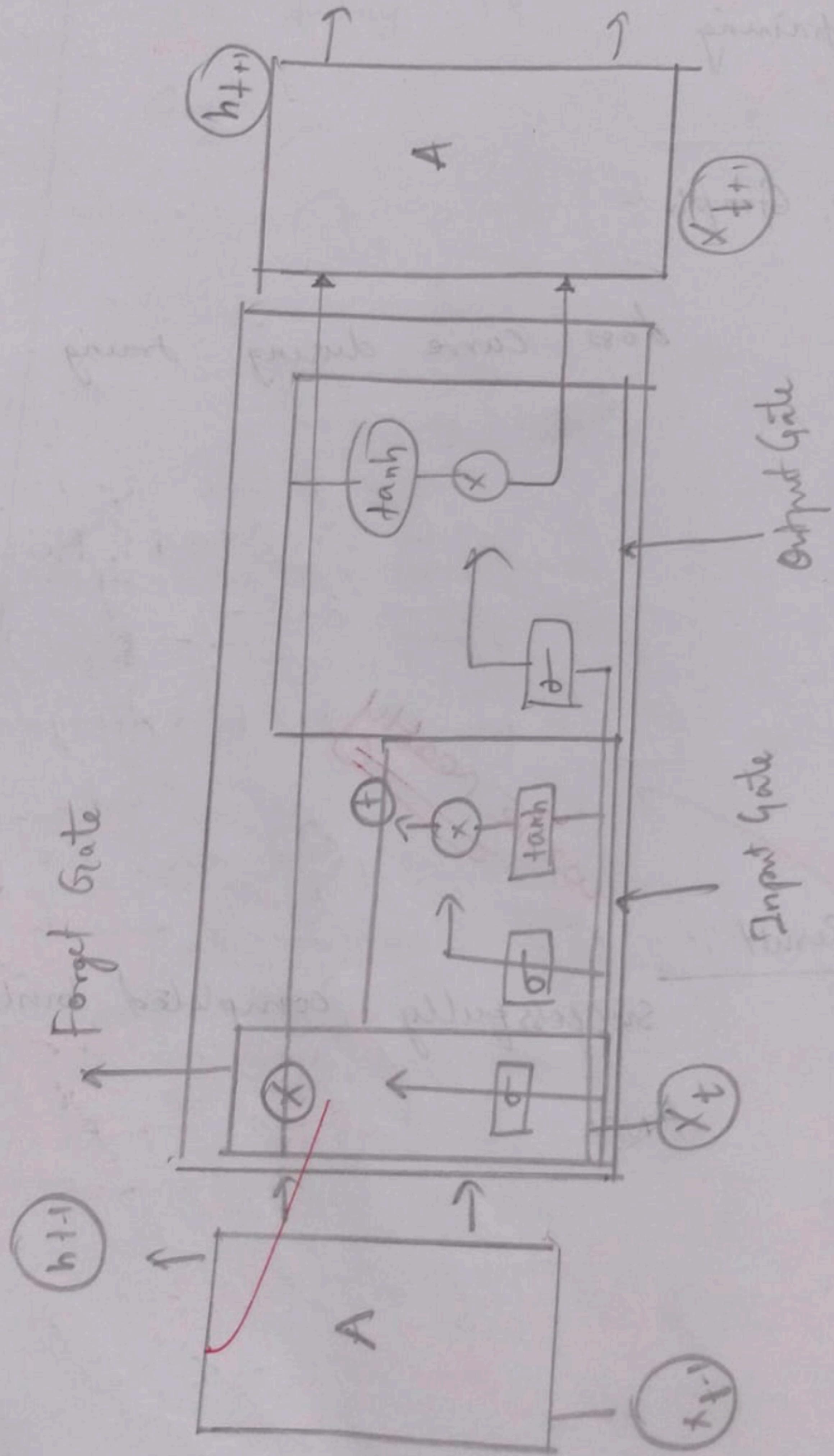
To understand how LSTMs process sequential data

To build and train an LSTM model for classification tasks

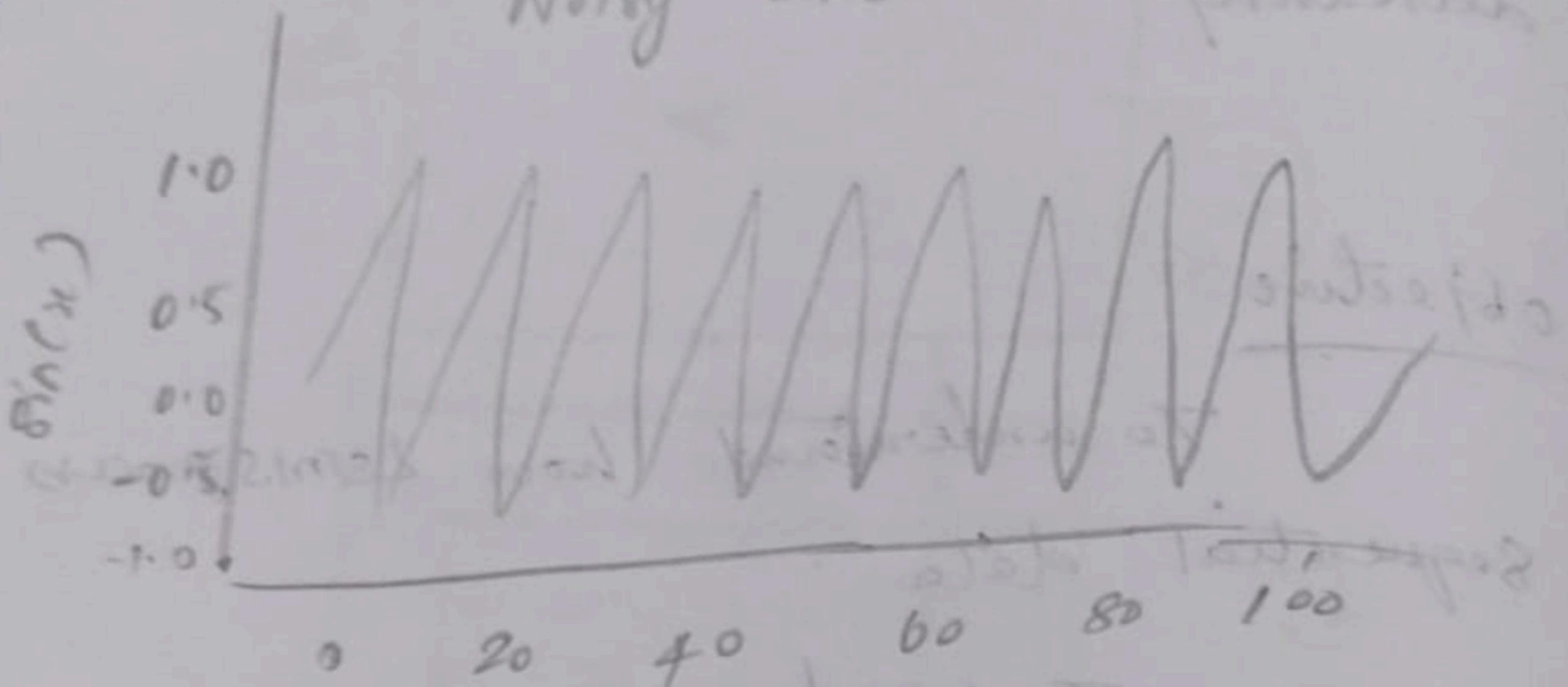
To evaluate model accuracy and visualize predictions

To analyze the performance of LSTM on pattern recognition tasks

## LSTM Architecture



Noisy Sine Wave Data



Input shape : torch.size [1960, 30, 1]

Target shape : torch.size [1960, 10, 1]

Training Loss

Epoch [1|30], Loss: 0.0798

Epoch [2|30], Loss: 0.0462

Epoch [3|30], Loss: 0.0355

Epoch [4|30], Loss: 0.0291

Epoch [5|30], Loss: 0.0247

Epoch [6|30], Loss: 0.0212

Test Accuracy 98.5%.

The loss curve decreases rapidly in the first few epochs.

After 50 epochs the loss stabilizes, showing that the model has converged.

Result :

The LSTM model has successfully classified with numerical sequences with high accuracy ( $\approx 98\%$ ).

## pseudocode

- Import necessary libraries
- Generate synthetic dataset
- preprocess data
- Define aNN model . (input-size, hiddden-size)
- Define loss function and optimizer -
  - Cross entropy loss
  - Adam optimizer
- Train the model .
  - i) for pass
  - ii) Compute loss
  - iii) Backward pass
  - iv) print loss and accuracy every few epochs
- Evaluate the model .
- Visualise a test example -
- Display results and accuracy graph .

~~Observation~~

Epoch	Training loss	Training Accuracy (%)
10	0.6201	72.3
20	0.4352	80.1
40	0.2103	90.7
60	0.0982	95.5
80	0.0524	97.8
100	0.0310	99.0