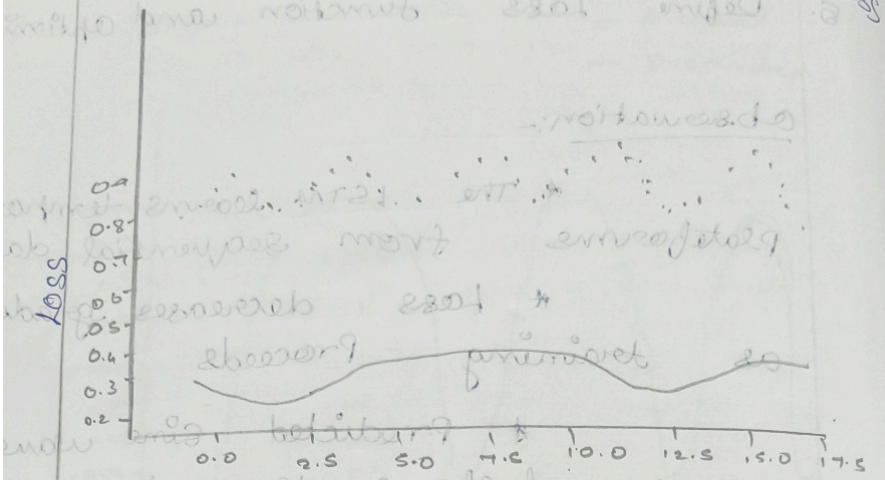
		Signature
8.	Long Short Term Memory	
9.	Building a RNN	
10.	Perform compression using MNIST dataset using autoencoder	
11.	Experiment using variations	
12.	Implement A Deep convolutional GAN	
13.	Understanding the architecture of Pre-Trained Model	
14.	Implement a Pre-trained CNN model as a Feature.	
15.	Implement a YOLO model for object Detection.	

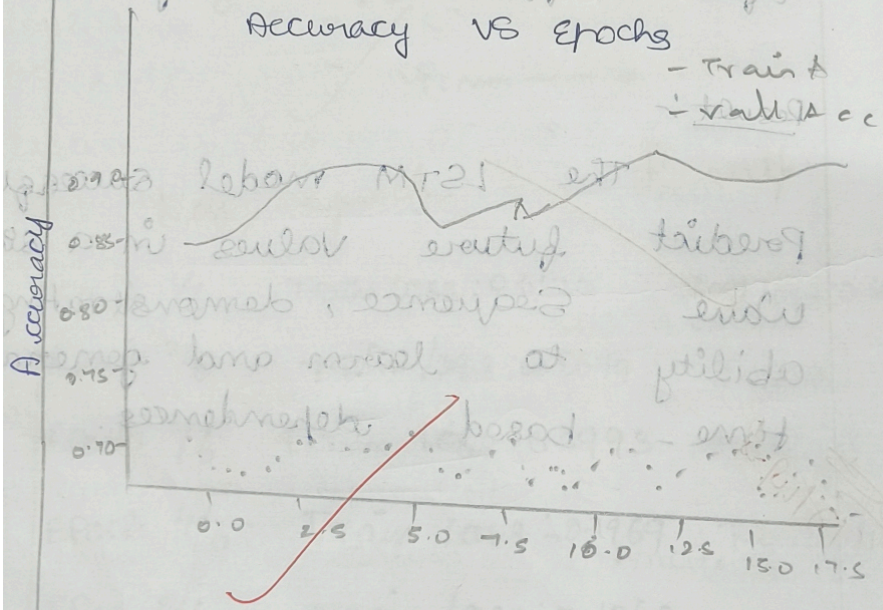
Loss vs epochs

- Train Loss
- Val Loss



Accuracy vs Epochs

- Train A
- Val A



Ex No: 9
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

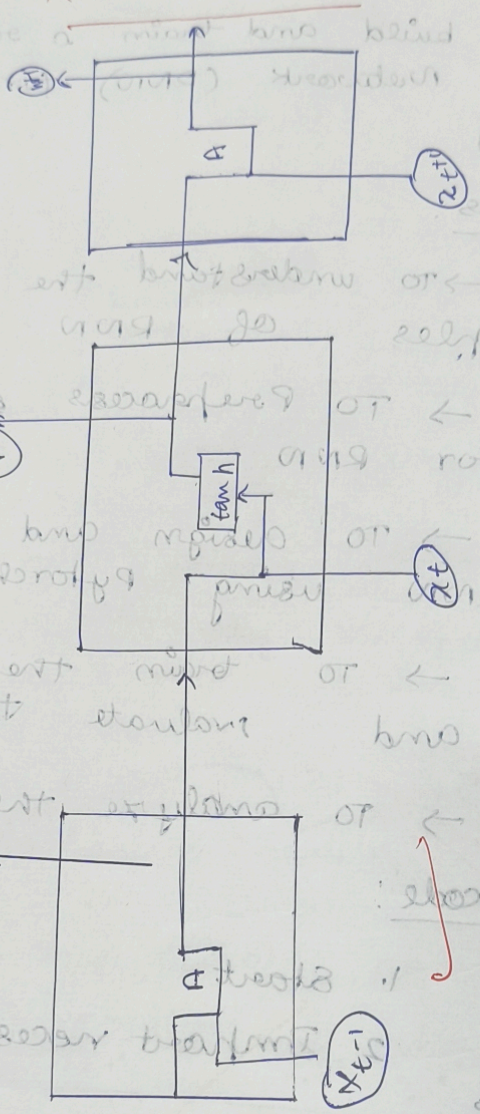
→ To train the RNN model and evaluate the Performan

→ To analyze the outputs.

Pseudocode:

1. Start
2. Import necessary Libraries
3. Load dataset
4. Preprocess dataset
   → clean data
   → tokenize / Encode Seq
   → Pad / truncate Sequ
   → Split into training & Validation sets.

RNN Architecture



6. Compile mode

→ Specify optimizer
→ Specify loss functions
→ Specify Evaluation metrics

7. Evaluation mode

→ test on unseen data
→ Print accuracy / loss metrics

8. END

Observations:-

	Precision	recall	F1-Score	support
0.0	0.71	0.70	0.70	4961
1.0	0.71	0.72	0.71	5039
accuracy			0.71	10000
Macro Avg	0.71	0.71	0.71	10000
weighted Avg	0.71	0.71	0.71	10000

Accuracy = 0.71

Epoches = 20.