Deep learning Techniques

RA23110[illegible]

B.Tech - AI - 'A' section

3rd year

UD 812

INDEX

NAME : TAMILSELVAN M STD : SEC : ROLL NO.: SUB

S.No.	Date	Title	Page No.	Teacher's Sign / Remarks
1	24/07/25	Exploring the deep learning Platform		[signature]
2.	31/07/25	Implement a classifier Using Open Source		
3	31/07/25	Study of the classifiers with respect to statistical Parameters		[signature] 7/8/25
4	07/08/25	Build a simple feed forward neural network to recognize handwritten character (MNIST Dataset)		
5	22/08/25	Study of Activation function & Their Role		[signature] 1/9/25
6	09/09/25	Implement Gradient Descent & Backpropagation in DNN.		
7	16/9/25	Build a CNN model to classify cats and dog image		[signature] 25/9/25
8	23/9/25	Build a RNN		30/9
9	30/9/25	Experiment using LSTM		[signature] 1/10/25

IMDB - dataset (50,000)

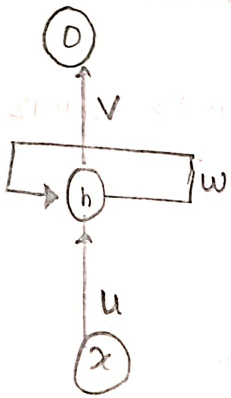
→ 35,000 (Train)

→ 15,000 (Test)

Label

Positive, Negative

ARCHITECTURE



$$h^{(t)} = \sigma\left(W_h h^{(t-1)} + W_x e^{(t)} + b\right)$$

30/9/25
Exp:8

BUILD A RECURRENT NEURAL NETWORK

AIM:

To implement a RNN model with IMDB dataset.

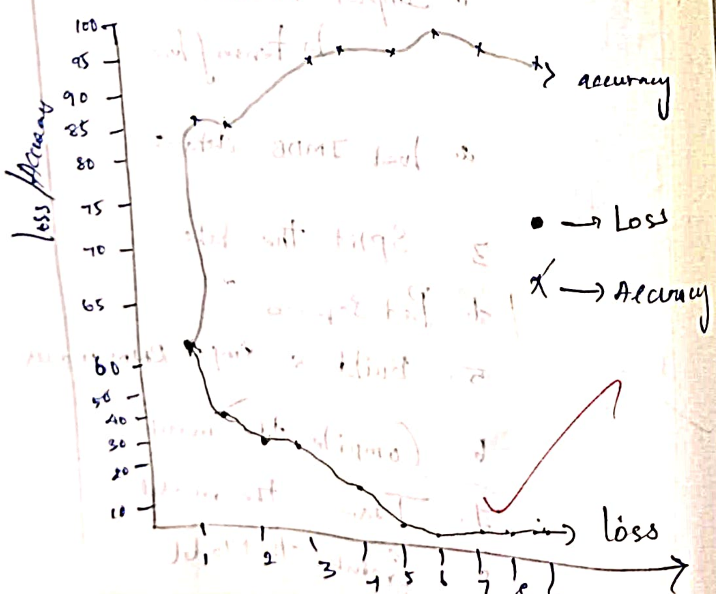
OBJECTIVE:

- To understand the working of RNN in NLP.
- To Perform binary sentiment classification using the IMDB dataset.
- To utilize Pre-trained word embedding to improve model Performance.

PSEUDOCODE:

1. Import libraries
   ↳ tensorflow
2. Load IMDB dataset
3. Split the data
4. Pad Sequences
5. Build a Simple RNN model
6. Compile the model
7. Train the model
8. Evaluate the Model

Epoch	Accuracy	Loss
1/9	62%	62%
2/9	87%	31%
3/9	86%	31%
4/9	93%	17%
5/9	98%	6%
6/9	98%	4%
7/9	99%	1%
8/9	97%	7%
9/9	96%	9%



OBSERVATION

→ The RNN is able to learn temporal relationship in text, even though it is simpler than GRU.

→ Accuracy is decent for a basic model, especially without advanced tuning or regularization.

JUSTIFICATION

→ The use of Pre-trained Embeddings or a learned embedding layer allows the model to capture semantic meaning of word Efficiently.

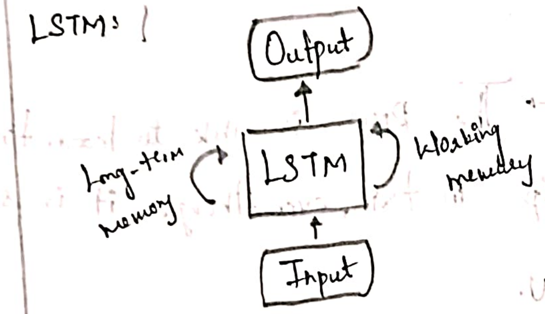
TRAIN & Loss	Accuracy
96%	0.09

CONCLUSION:

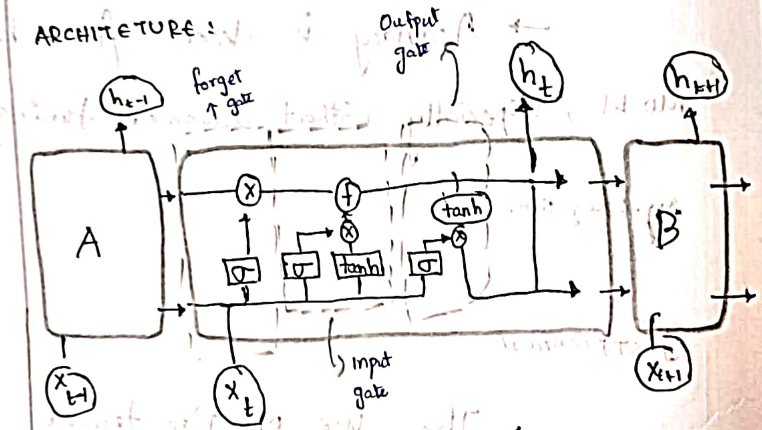
Successfully implemented a Simple RNN

30/9

LSTM:



ARCHITETURE:



APPLICATION : Image Captioning, Machine Translation, Music Generation, Auto Handwriting Generation

Ex: 9
30/9/25

Experiment Using LSTM

AIM:

Long Short Term Memory (LSTM) is an enhanced Version of the RNN. Designed by Hochreiter and Schmidhuber. Build a LSTM to predict the next value.

Objective

To implement a simple LSTM network using Keras and Tensorflow in python.

Pseudocode:

1. Installing required libraries
2. Data preparind
3. Build the LSTM model
4. Train model
5. Make a Prediction

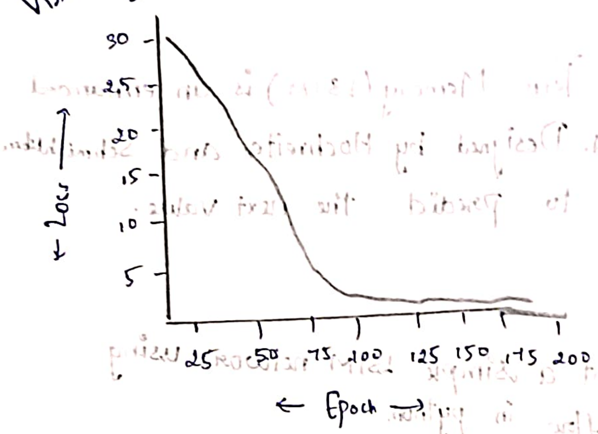
OBSERVATION:

Data prepartion:

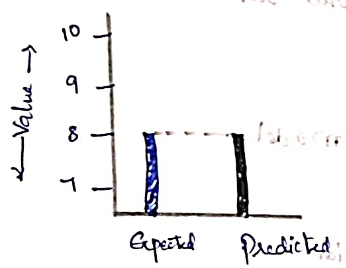
$X = [[1,2,3], [2,3,4], [3,4,5], [4,5,6]]$

$Y = [4,5,6,7]$ In sequence and features is the number of features at each timestamp

Visual Representation



Expected Vs Predicted



Model Building

- 50 LSTM units → 'ReLU' activat function
- A dense (fully connected) layer with

one unit is added (1)

- Optimizer → Adam
- Loss functio →(MSE)" Mean Square Error"
- Training

  Epoch → 200 with Suppress Verbose
- Make prediction

Output

Predicted next value for [5,6,7] : 8.01

Result:

Implemented a simple LSTM model. Predicted next value from trained sequence.