Deep learning Techniques

INDEX

RA23110470100 38
B.Tech – AI – 'A'section
2nd year

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		
2.	31/07/25	Implement a classifier using Open Source		
3	31/07/25	Study of the classifiers with respect to statistical Parameters		
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.		
6	09/09/25	Implement Gradient Descent & Backpropagation in DNN.		

BUILD A SIMPLE FEED FORWARD NEURAL NETWORK

To Recognize Handwritten Character.

Aim:
To design, implement, and train a feed-forward neural network using the MNIST dataset to recognize handwritten digits (0-9).

OBJECTIVES:
1. To understand the architecture of a simple feed-forward neural network.
2. To preprocessing image data for neural network training.
3. To implement the network using dense layers in (Tensorflow) Pytorch.
4. To analyze model performance using test data.

Pseudocode:
1. Import required libraries
2. Load dataset
3. Define feed forward model
4. Initialize model, loss function (NLLLoss) and Optimizer (Adam, lr=0.001)
5. For each epoch in range (1,5)
6. Evaluate Model
7. Sample Prediction

MNIST

Dataset

60,000 — Train Dataset

10,000 — Test Dataset

Labels

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

OBSERVATION :

- The feed forward neural network trained on the MNIST dataset showed consistent improved in accuracy across epochs, reaching around 97%.

- The training loss decreased steadily, indicating effective learning.

- Most misclassifications occured between digits with similar shapes such as (3, 5), (7, 9).

- The model generalized well due to MNIST's simplicity and clean data.

- Increasing hidden units can improve performance.

Output:

Epoch 1, Loss: 0.3782
Epoch 2, Loss: 0.0669
Epoch 3, Loss: 0.0659
Epoch 4, Loss: 0.0517
Epoch 5, Loss: 0.0059
Test Accuracy : 97.05%.

Justification

→ FFNNs are effective for basic classification but lack spatial awareness.

→ MNIST digits are 28×28 images; flattening them loses spatial structure.

RESULT :

A simple FFNN that flatten 28×28 images, passes them through two hidden layers, and predict the digit classes.