

3rd year

INDEX

S.No.	Date	Title	Page No.	Teacher's Sign / Remarks
1	24/07/25	Exploring the deep learning Platform		24/07/25
2	31/07/25	Implement a classifier Using Open Source		31/07/25
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)		07/08/25
5	22/08/25	Study of Activation function & Their Role.		
6	09/09/25	Implement Gradient Descent & Backpropagation in DNN.		09/09/25

Ex No 6 Implement Gradient Descent Backpropagation 9/9/25 in Deep Neural Network.

AIM:

To implement Gradient Descent and backpropagation algorithm in a deep neural network and study their role in training

OBJECTIVE:

- Standard and Build a small fully-connected neural network (1 hidden layer, ReLU, Sigmoid Output)

• Training Using:

1. fixed learning rate GD
 2. GD with Backpropagation (Autograd)
- Compare loss curves and final accuracies in a toy binary problem.

Pseudocode:

Create dataset x, y

Initialize network parameters for each epoch:

forward pass \rightarrow Compute predict AL

Compute loss $L(\theta_1, Y)$

backward pass \rightarrow Compute Gradient grad

Update parameters

\rightarrow Fixed LR

\rightarrow Backtracking

$$\frac{x - 0.5}{\sqrt{0.5 \times 0.5}} = \frac{(x - 0.5)}{0.5}$$

Data Set

Two-Moons Dataset

2D

x_1 x_2 label

$\phi = 76$ $\sigma = 0.005$ 1

(find weight & bias) using mini-batch

$x_1 \rightarrow$ horizontal axis

$x_2 \rightarrow$ vertical axis

$$\nabla J = \frac{\partial J}{\partial w, b}$$

$$\theta = \theta - \alpha \nabla J(\theta)$$

Batch GD

with computing gradient at every step

• Loss, bias, weight, gradient

Observation:

- Backpropagation & Autograd.

PyTorch Computed all gradient with loss.backward()

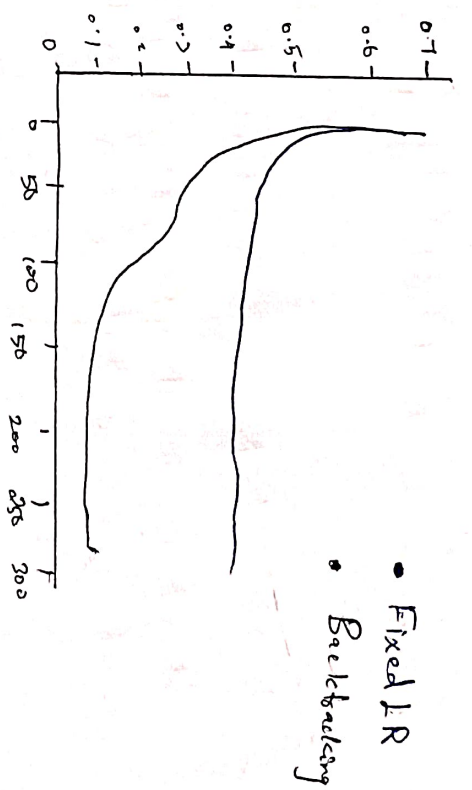
It understood how gradient are stored in .grad and that Optimization is just $\text{Params} \leftarrow \text{Params} - \text{grad}$.

Output

Fixed LR is Rigid, block early on, but can still hit the Learning rate is too high / low for later training.

Backtracking adapts: Step size is tuned per update, so it squeezes out better convergence without you guessing a learning rate.

Lower loss is better Optimization, If network is learning a more accurate decision boundary.



- Fixed LR
- Backtracking

Final Summary

Fixed LR

Train Accuracy : 0.86 , Test Accuracy : 0.86
 Back Track : 0.98 , Test Accuracy : 0.98

~~9/19/25~~
 Result.

Finally we implemented Backtracking
 Success fully.