

Expt-6: Implement Gradient Descent & Backpropagation in Deep Neural Network.

AIM:

To implement gradient descent optimization and backpropagation

Objective:

1. Build a simple deep neural network with one hidden layer
2. Implement forward propagation to compute predictions.
3. Use backpropagation and gradient descent to update weights
4. Train the model on a small dataset and observe the loss reduction.

Description:

1. Gradient Descent:

- * An optimization algorithm that of weight update

$$w \leftarrow w - \eta \cdot \frac{\partial L}{\partial w}$$

η = learning rate
 L = loss function

2. Backpropagation

An algo to compute gradient of loss func. w.r.t weight steps

Forward pass - computes output of network
Loss calculation

Backward pass

→ weight update - apply gradient descent

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Pseudocode

1. Import libraries
2. Define dataset (inputs and labels)
3. Define neural network structure:
 - Input layer
 - Hidden layer
 - Output layer
4. Define loss function (e.g. mean squared error)
5. Define optimizer (gradient descent)
6. Training loop:
 - a. Forward pass (compute prediction)
 - b. Compute loss
 - c. Backward pass (compute gradients using backpropagation)
 - d. Update weights using gradient descent
 - e. Repeat for several epochs
7. Print loss to show learning.

Observations

The loss decreased steadily with each epoch showing that backpropagation and gradient descent updated the weights correctly. The final prediction were close to target values, proving the network learned the input-output mapping.

Result:

Implementation of gradient descent and backpropagation is successfully completed.

✓
8/19

Output

Epoch 200, loss : 0.6927

Epoch 400, loss : 0.6927

Train Accuracy : 51.50%

Test Accuracy : 44.00%

Epoch 0 loss = 1.076

Epoch 1000, loss = 0.6932

Epoch 2000, loss = 0.6931

Epoch 3000, loss = 0.6931

Epoch 4000, loss = 0.6931

Epoch 5000, loss = 0.6931

Epoch 6000, loss = 0.6931

Epoch 7000, loss = 0.6931

Epoch 8000, loss = 0.6931

Epoch 9000 loss = 0.6931

Final Prediction

[0]

[1]

[0]

[1]

Final Prediction

0.7

0.6

0.55

0.5

0.45

loss

2000

4000

6000

8000

10000

epoch vs loss