

Practice Lab Neural Network 1

Name: Ayush Bhalerao

PRN Number: 202302090015

Batch: DL [MDM] (T4)

Date of Submission: 15-01-2025

Neural Network Implementation from Scratch

Objective

The goal of this project is to implement a simple feedforward neural network in Python from scratch without using any deep learning libraries like TensorFlow or PyTorch. The implementation includes

- Forward pass,
- Backpropagation,
- Gradient descent.

Problem Definition

The dataset used is a binary classification problem based on the AND operation.

Dataset:

Inputs:

[0, 0], [0, 1], [1, 0], [1, 1]

Expected Outputs:

[0], [0], [0], [1]

Neural Network Architecture

1. **Input Layer:**
 - 2 neurons (representing two binary inputs).
2. **Hidden Layer:**
 - 3 neurons.
 - Activation Function: Sigmoid.

3. Output Layer:

- 1 neuron (binary classification output).
- Activation Function: Sigmoid.

Task:

The neural network is trained to predict the AND operation results for the given binary input combinations.

Methodology

Forward Pass:

1. Multiply inputs with weights, add bias, and pass through the Sigmoid activation function (hidden layer).
2. Multiply hidden layer output with weights, add bias, and pass through Sigmoid (output layer).

Backpropagation:

- Calculate error between predicted and actual outputs.
- Update weights and biases using gradient descent.

Loss Function:

- **Mean Squared Error (MSE):** Measures the squared difference between actual and predicted outputs.

Optimization :

- **Vectorization:** Ensure that the operations are performed using matrix operations rather than loops, which can speed up computations.
- **Batch Training:** Instead of updating the weights for every single data point (stochastic gradient descent), we can use mini-batch gradient descent, which often leads to faster convergence.
- **Learning Rate Scheduling:** Adjusting the learning rate during training can improve convergence by decreasing it gradually.
- **Momentum:** Add momentum to the gradient updates, which can help accelerate gradient descent in the relevant direction and dampen oscillations.
- **Early Stopping:** Stop training when the model performance no longer improves significantly.

Declaration

I, **Ayush Bhalerao**, confirm that the work submitted in this assignment is my own and has been completed following academic integrity guidelines. The code is uploaded to my GitHub repository, and the repository link is provided below: **GitHub Repository Link:**

https://github.com/ayushbhalerao7/Neural_network_from_scratch

Signature:

Ayush Ganesh Bhalerao