Lab1: Training a Neural Network from Scratch

Objective

Train a simple neural network from scratch using Python and NumPy, without deep learning frameworks like TensorFlow or PyTorch.

Tasks

1. **Understanding the Basics**
    - Briefly explain what a neural network is.
    - Describe the architecture of a simple feedforward neural network (input, hidden, and output layers).
2. **Data Preparation**
    - Generate a simple dataset (e.g., classify points in 2D space).
    - Normalize the data.
    - Split data into training and testing sets.
3. **Initialize Network Parameters**
    - Define the number of input neurons, hidden neurons, and output neurons.
    - Randomly initialize weights and biases.
4. **Forward Propagation**
    - Compute activations using weighted sums and activation functions (ReLU, Sigmoid).
    - Implement the forward pass for a single training example.
5. **Loss Computation**
    - Define a loss function (e.g., Mean Squared Error or Cross-Entropy).
    - Compute the loss for a batch of data.
6. **Backpropagation**
    - Derive gradients of the loss with respect to weights and biases.
    - Update parameters using Gradient Descent.
7. **Training Loop**
    - Implement the full training loop (forward pass, loss computation, backpropagation, and weight updates).
    - Train the network for multiple epochs.
8. **Evaluation**
    - Test the trained model on new data.

Discussion and Improvements

- Discuss how training could be improved (e.g., adding more layers, changing learning rates).
- Briefly introduce more advanced concepts like batch normalization or dropout.