Objective:

In this lab, you will develop a simple neural network to classify the Iris dataset, a classic dataset in machine learning. You will implement the network from scratch, including the forward pass, the computation of the loss, and the backward pass using gradient descent for parameter updates.

1. Data Preparation

The Iris dataset contains 150 samples of iris flowers. There are 4 features (sepal length, sepal width, petal length, petal width) and 3 species of iris (setosa, versicolor, virginica).

1.1. Load the Iris dataset. If you are using sklearn, you can load the dataset using `sklearn.datasets.load_iris()`.

1.2. Normalize the feature data.

1.3. Split the data into training and test sets.

2. Neural Network Setup

2.1. Initialize the weights and biases for a single-layer neural network. For simplicity, use a single neuron for binary classification (Iris setosa vs. non-setosa).

2.2. Define the sigmoid activation function and its derivative (Write your own functions for sigmoid and sigmoid_derivative)

2.3. Implement the mean squared error loss function (Write your own mse_loss function)

3. Forward Pass

3.1. Calculate the weighted sum of inputs. $z=wx+b$

3.2. Apply the sigmoid activation function. $F = \text{sigmoid}(z)$

4. Backward Pass

4.1. Compute the derivative of the loss function with respect to the prediction.

4.2. Compute the gradients with respect to weights and biases.

5. Parameter Update

5.1. Update the weights and biases using gradient descent.

6. Training the Model

6.1. Implement the training loop:

- For each epoch, perform the forward pass on the entire training set.
- Compute the loss.
- Perform the backward pass to update the weights.

6.2. Evaluate the model on the test set after training.

7. Analysis

7.1. Plot the training loss over epochs.

7.2. Report the final accuracy on the test set.