Objective:

The objective of this script is to implement a Multi-Layer Perceptron (MLP) using NumPy to learn the XOR Boolean function.

The MLP is trained using backpropagation and gradient descent to adjust weights and biases for accurate predictions.

Description:

This script defines an MLP with an input layer (2 neurons), a hidden layer (4 neurons), and an output layer (1 neuron).

It uses the sigmoid activation function and updates weights through backpropagation for 10,000 epochs with a learning rate of 0.5.

The goal is to correctly classify XOR inputs.

Description of Code:

1. The input and output for the XOR function are defined.

2. The network is initialized with random weights and biases.

3. The forward propagation computes hidden and output activations.

4. Backpropagation calculates the error and updates weights.

5. Training runs for 10,000 epochs, and loss is printed at intervals.

6. The final predictions are printed after training.

Output and Performance:

1. The script outputs the loss at every 1000 epochs to track performance.

2. The final predictions are rounded to match XOR truth table outputs.

3. The network successfully learns XOR, with near-zero loss after training.

My Comments:

1. Increasing the hidden layer size may improve convergence.

2. Using additional activation functions like ReLU could enhance learning.

3. The script successfully demonstrates how a simple neural network learns non-linearly separable data.