Jeremy Pretty Week 3 Crit

Introduction:

The given Python script demonstrates a basic 2-layer Artificial Neural Network (ANN) that learns to approximate the XOR function using static backpropagation. The ANN consists of an input layer, a hidden layer, and an output layer. It is trained using the gradient descent optimization method and the sigmoid activation function.

Structure of the ANN:

The ANN is designed with the following structure:

1. Input layer: Takes input data as a matrix and passes it on to the next layer.
2. Hidden layer: Contains two neurons and utilizes the sigmoid activation function.
3. Output layer: Contains a single neuron that also uses the sigmoid activation function.

Training the ANN:

The training process involves the following steps:

1. Initialize the weights and biases for the hidden and output layers with random values.
2. Perform a forward pass, calculating the output of the hidden layer and the final output (predicted output) of the network.
3. Compute the error between the desired output (ground truth) and the predicted output.
4. Apply backpropagation to adjust the weights and biases based on the calculated error and gradients.
5. Repeat the process for a specified number of epochs (iterations).

In this example, the ANN is trained for 10,000 epochs with a learning rate of 0.1.

Functions Used:

1. Sigmoid activation function: A smooth, differentiable function that maps input values to a range between 0 and 1. It is used in both the hidden and output layers.
2. Sigmoid derivative function: The derivative of the sigmoid function, used in backpropagation to compute gradients.

Implementation:

The script is divided into three main sections:

1. Function definitions: sigmoid, sigmoid\_derivative, train, and predict functions are defined to perform various operations in the ANN.
2. Training phase: The input and output data for the XOR function are defined as NumPy arrays. The train function is called with the specified parameters (input data, output data, epochs, and learning rate), and it returns the trained weights and biases.
3. Testing phase: The predict function is called with the test data and the trained weights and biases to evaluate the performance of the ANN. The predicted output is then printed to the console.

Conclusion:

The provided script demonstrates a simple 2-layer ANN that learns to approximate the XOR function using static backpropagation, gradient descent, and the sigmoid activation function. The network can be customized to work with different functions and learning settings by adjusting the input and output data, the number of epochs, and the learning rate. This ANN serves as a basic example for understanding the fundamental concepts of feedforward neural networks and the backpropagation algorithm.