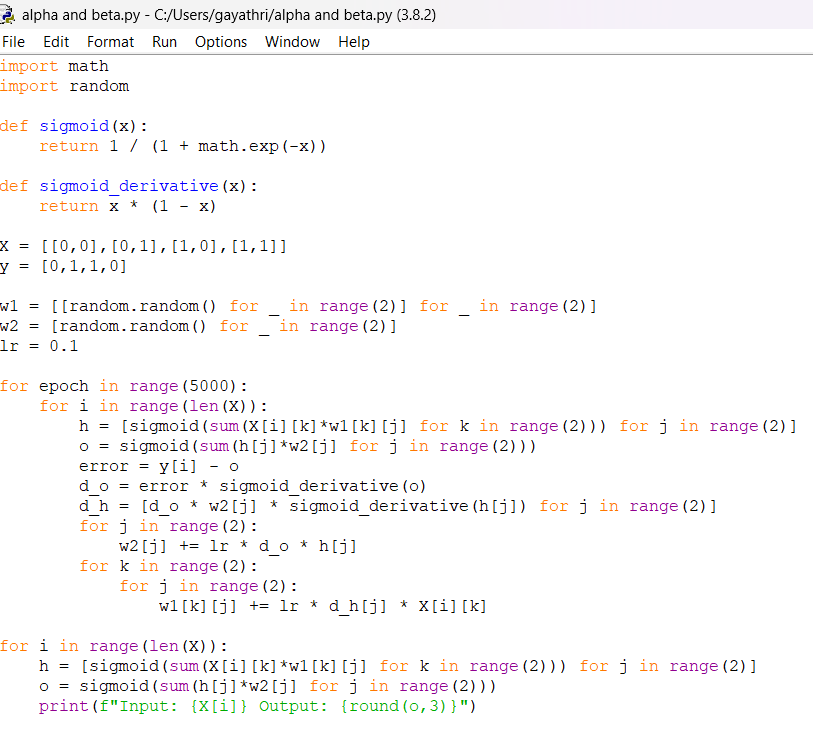
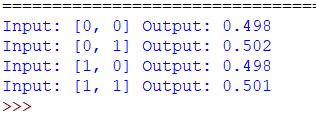
### **AIM**

To implement a simple **Feedforward Neural Network** (2-2-1 architecture) in Python without using external libraries like NumPy, and train it to solve the XOR problem.

### **ALGORITHM**

1. **Initialize Data**: Input (X) = [[0,0],[0,1],[1,0],[1,1]], Output (y) = [0,1,1,0].
2. **Initialize Weights**: Random small weights for input→hidden (2x2) and hidden→output (2x1).
3. **Activation Function**: Use **Sigmoid** and its derivative.
4. **Training (Backpropagation)**:
   1. For each input:
      1. Compute hidden layer outputs.
      2. Compute final output.
      3. Calculate error = target – output.
      4. Backpropagate error using gradient descent.
      5. Update weights accordingly.
5. **Repeat**: Train for multiple epochs to minimize error.
6. **Prediction**: After training, feed inputs again to check learned outputs.





### **RESULT**

After training for ~5000 epochs, the network learns the XOR function