Assignment 2: Write a program to show back propagation network for XOR function with binary input and output

Code =>

```
import numpy as np


def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def sigmoid_derivative(x):
    return x * (1 - x)

# Input and Output for XOR Gate
X = np.array([[0,0], [0,1], [1,0], [1,1]])
y = np.array([[0], [1], [1], [0]])

# Initialize Weights and Biases
np.random.seed(42)
input_layer_neurons = 2
hidden_layer_neurons = 2
output_neurons = 1

hidden_weights = np.random.uniform(size=(input_layer_neurons,
hidden_layer_neurons))
hidden_bias = np.random.uniform(size=(1, hidden_layer_neurons))
output_weights = np.random.uniform(size=(hidden_layer_neurons,
output_neurons))
output_bias = np.random.uniform(size=(1, output_neurons))

# Learning Rate and Epochs
lr = 0.5
epochs = 10000

# Training Process
for epoch in range(epochs):
    # Forward Propagation
    hidden_layer_activation = np.dot(X, hidden_weights) + hidden_bias
    hidden_layer_output = sigmoid(hidden_layer_activation)
    output_layer_activation = np.dot(hidden_layer_output, output_weights) +
output_bias
    predicted_output = sigmoid(output_layer_activation)

    # Backward Propagation
    error = y - predicted_output
    d_predicted_output = error * sigmoid_derivative(predicted_output)
    error_hidden_layer = d_predicted_output.dot(output_weights.T)
```

```
    d_hidden_layer = error_hidden_layer *
sigmoid_derivative(hidden_layer_output)

    # Update Weights and Biases
    output_weights += hidden_layer_output.T.dot(d_predicted_output) * lr
    output_bias += np.sum(d_predicted_output, axis=0, keepdims=True) * lr
    hidden_weights += X.T.dot(d_hidden_layer) * lr
    hidden_bias += np.sum(d_hidden_layer, axis=0, keepdims=True) * lr

# Print Results
print("Final Hidden Weights:")
print(hidden_weights)
print("Final Hidden Bias:")
print(hidden_bias)
print("Final Output Weights:")
print(output_weights)
print("Final Output Bias:")
print(output_bias)

# Test the Trained Model
print("\nPredicted Output:")
print(predicted_output)
```

Output =>

```
[Running] python -u "c:\Users\Shreyash Musmade\Desktop\Practical\ANN\ANN_Prac-
2\tempCodeRunnerFile.py"
Final Hidden Weights:
[[4.59244504 6.47246975]
 [4.5971031  6.49153682]]
Final Hidden Bias:
[[-7.05239171 -2.8842842 ]]
Final Output Weights:
[[-10.32676834]
 [  9.62121009]]
Final Output Bias:
[[-4.44969307]]

Predicted Output:
[[0.01890475]
 [0.98371361]
 [0.98369334]
 [0.01686123]]

[Done] exited with code=0 in 0.885 seconds
```