Assignment 3: Write a program for producing back propagation feed forward network

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 Data
X = np.array([[0,0], [0,1], [1,0], [1,1]])
y = np.array([[0], [1], [1], [0]])

# Initialize Neural Network Parameters
input_neurons = 2
hidden_neurons = 4
output_neurons = 1

np.random.seed(42)
hidden_weights = np.random.uniform(size=(input_neurons, hidden_neurons))
hidden_bias = np.random.uniform(size=(1, hidden_neurons))
output_weights = np.random.uniform(size=(hidden_neurons, output_neurons))
output_bias = np.random.uniform(size=(1, output_neurons))

# Hyperparameters
learning_rate = 0.5
epochs = 10000

# Training the Network
for epoch in range(epochs):
    # Forward Pass
    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)

    # Backpropagation
    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)
```

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

# Output Results
print("Final Predicted Output:")
print(predicted_output)
print("\nFinal Hidden Weights:")
print(hidden_weights)
print("\nFinal Output Weights:")
print(output_weights)
```

Output =>

```
[Running] python -u "c:\Users\Shreyash Musmade\Desktop\Practical\ANN\ANN_Prac-
3\Practical.py"
Final Predicted Output:
[[0.01422264]
 [0.9905565 ]
 [0.97896397]
 [0.01855849]]

Final Hidden Weights:
[[ 2.88454752  5.66841784  2.38355745  6.409118  ]
 [ 2.06022227 -3.98268118  5.09330327  5.39041736]]

Final Output Weights:
[[ 2.36470392]
 [-6.14074008]
 [-9.61054347]
 [ 8.99632791]]

[Done] exited with code=0 in 0.826 seconds
```