```python
import numpy as np

# Define the sigmoid activation function and its derivative
def sigmoid(x):
  return 1 / (1 + np.exp(-x))

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

# Input dataset
inputs = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
expected_output = np.array([[0], [1], [1], [0]])

# Initialize weights with random values
epochs = 10000
lr = 0.1
inputLayerNeurons, hiddenLayerNeurons, outputLayerNeurons = 2, 2, 1

hidden_weights = np.random.uniform(size=(inputLayerNeurons,hiddenLayerNeurons))
hidden_bias =np.random.uniform(size=(1,hiddenLayerNeurons))
output_weights = np.random.uniform(size=(hiddenLayerNeurons,outputLayerNeurons))
output_bias = np.random.uniform(size=(1,outputLayerNeurons))

# Training algorithm
for _ in range(epochs):
  # Forward Propagation
  hidden_layer_activation = np.dot(inputs,hidden_weights)
  hidden_layer_activation += hidden_bias
  hidden_layer_output = sigmoid(hidden_layer_activation)

  output_layer_activation = np.dot(hidden_layer_output,output_weights)
  output_layer_activation += output_bias
  predicted_output = sigmoid(output_layer_activation)

  # Backpropagation
  error = expected_output - 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) * lr
  output_bias += np.sum(d_predicted_output,axis=0,keepdims=True) * lr
  hidden_weights += inputs.T.dot(d_hidden_layer) * lr
  hidden_bias += np.sum(d_hidden_layer,axis=0,keepdims=True) * lr

print("Final hidden weights: ",end='')
print(*hidden_weights)
print("Final hidden bias: ",end='')
print(*hidden_bias)
print("Final output weights: ",end='')
print(*output_weights)
print("Final output bias: ",end='')
print(*output_bias)

print("\nOutput from neural network after 10,000 epochs: ",end='')
print(*predicted_output)
```

```
Final hidden weights: [3.63964239 5.85933632] [3.63578021 5.8383332 ]
Final hidden bias: [-5.56159607 -2.41768812]
Final output weights: [-8.0619703] [7.4229874]
Final output bias: [-3.33216592]

Output from neural network after 10,000 epochs: [0.05977854] [0.9442818] [0.94434861] [0.06060218]
```