ANN from Scratch in Python

Python Code

```
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)

# Define the Neural Network class
class NeuralNetwork:
def __init__(self, input_size, hidden_size, output_size):
# Initialize weights and biases
self.weights_input_hidden = np.random.randn(input_size,
hidden_size)
self.bias_hidden = np.zeros((1, hidden_size))
self.weights_hidden_output = np.random.randn(hidden_size,
output_size)
self.bias_output = np.zeros((1, output_size))

def forward(self, X):
# Forward pass
self.input = X
self.hidden_input = np.dot(self.input, self.
weights_input_hidden) + self.bias_hidden
self.hidden_output = sigmoid(self.hidden_input)
self.output_input = np.dot(self.hidden_output, self.
weights_hidden_output) + self.bias_output
self.output = sigmoid(self.output_input)
return self.output

def backward(self, X, y, learning_rate):
# Backward pass
output_error = y - self.output
output_delta = output_error * sigmoid_derivative(self.output)

hidden_error = np.dot(output_delta, self.weights_hidden_output.
T)
hidden_delta = hidden_error * sigmoid_derivative(self.
hidden_output)

# Update weights and biases
```

```
    self.weights_hidden_output += np.dot(self.hidden_output.T,
    output_delta) * learning_rate
    self.bias_output += np.sum(output_delta, axis=0, keepdims=True)
     * learning_rate
    self.weights_input_hidden += np.dot(self.input.T, hidden_delta)
     * learning_rate
    self.bias_hidden += np.sum(hidden_delta, axis=0, keepdims=True)
     * learning_rate

    def train(self, X, y, epochs, learning_rate):
    for _ in range(epochs):
    self.forward(X)
    self.backward(X, y, learning_rate)

    def predict(self, X):
    return self.forward(X)

    # Example usage
    if __name__ == "__main__":
    # Define dataset
    X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
    y = np.array([[0], [1], [1], [0]])  # XOR problem

    # Initialize and train the neural network
    nn = NeuralNetwork(input_size=2, hidden_size=2, output_size=1)
    nn.train(X, y, epochs=10000, learning_rate=0.1)

    # Test the neural network
    predictions = nn.predict(X)
    print("Predictions:\n", predictions)

```

Listing 1: Artificial Neural Network from Scratch