import numpy as np

import matplotlib.pyplot as plt

class McCullochPittsNeuron:

    def \_\_init\_\_(self, weights, threshold):

        self.weights = weights

        self.threshold = threshold

    def activate(self, inputs):

        # Calculate weighted sum

        weighted\_sum = np.dot(inputs, self.weights)

        # Step activation function

        return 1 if weighted\_sum >= self.threshold else 0

# Demonstrate logical AND function

and\_neuron = McCullochPittsNeuron(weights=[1, 1], threshold=2)

# Demonstrate logical OR function

or\_neuron = McCullochPittsNeuron(weights=[1, 1], threshold=1)

# Demonstrate logical NOT function (for one input)

not\_neuron = McCullochPittsNeuron(weights=[-1], threshold=0)

# Test inputs

inputs = [(0, 0), (0, 1), (1, 0), (1, 1)]

not\_inputs = [0, 1]

# Evaluate the neurons

print("McCulloch-Pitts Neuron Demonstrations:")

print("\nAND Gate:")

for input\_pair in inputs:

    output = and\_neuron.activate(input\_pair)

    print(f"Input: {input\_pair}, Output: {output}")

print("\nOR Gate:")

for input\_pair in inputs:

    output = or\_neuron.activate(input\_pair)

    print(f"Input: {input\_pair}, Output: {output}")

print("\nNOT Gate:")

for input\_val in not\_inputs:

    output = not\_neuron.activate([input\_val])

    print(f"Input: {input\_val}, Output: {output}")

# Visualize the decision boundaries for AND and OR

plt.figure(figsize=(12, 5))

# AND gate

plt.subplot(1, 2, 1)

x = np.linspace(-0.5, 1.5, 100)

y = (and\_neuron.threshold - and\_neuron.weights[0] \* x) / and\_neuron.weights[1]

plt.plot(x, y, 'r-', label='Decision Boundary')

plt.scatter([0, 0, 1, 1], [0, 1, 0, 1], c=['blue', 'blue', 'blue', 'red'])

plt.annotate('(0,0)=0', (0, 0), xytext=(-0.1, -0.1))

plt.annotate('(0,1)=0', (0, 1), xytext=(-0.1, 1.1))

plt.annotate('(1,0)=0', (1, 0), xytext=(1.1, -0.1))

plt.annotate('(1,1)=1', (1, 1), xytext=(1.1, 1.1))

plt.xlim(-0.5, 1.5)

plt.ylim(-0.5, 1.5)

plt.xlabel('Input 1')

plt.ylabel('Input 2')

plt.title('AND Gate')

plt.grid(True)

# OR gate

plt.subplot(1, 2, 2)

y = (or\_neuron.threshold - or\_neuron.weights[0] \* x) / or\_neuron.weights[1]

plt.plot(x, y, 'r-', label='Decision Boundary')

plt.scatter([0, 0, 1, 1], [0, 1, 0, 1], c=['blue', 'red', 'red', 'red'])

plt.annotate('(0,0)=0', (0, 0), xytext=(-0.1, -0.1))

plt.annotate('(0,1)=1', (0, 1), xytext=(-0.1, 1.1))

plt.annotate('(1,0)=1', (1, 0), xytext=(1.1, -0.1))

plt.annotate('(1,1)=1', (1, 1), xytext=(1.1, 1.1))

plt.xlim(-0.5, 1.5)

plt.ylim(-0.5, 1.5)

plt.xlabel('Input 1')

plt.ylabel('Input 2')

plt.title('OR Gate')

plt.grid(True)

plt.tight\_layout()

plt.show()