**ML EXPERIMENT 9**

class MPNeuron:

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

        self.threshold = threshold

        self.weights = weights

    def activate(self, inputs):

        weighted\_sum = sum([x \* w for x, w in zip(inputs, self.weights)])

        if weighted\_sum >= self.threshold:

            return 1

        else:

            return 0

# Implementing AND gate using MP neuron

def main():

    # Define the threshold and weights for the AND gate

    threshold = 2  # Sum of weights

    weights = [1, 1]  # Equal weights for both inputs

    print("Threshold value:", threshold)

    print("Weights:", weights)

    # Create an instance of the MPNeuron

    and\_gate = MPNeuron(threshold, weights)

    # Define input combinations for AND gate

    inputs = [

        [0, 0],

        [0, 1],

        [1, 0],

        [1, 1]

    ]

    # Test the AND gate

    print("\nAND Gate:")

    for input\_pair in inputs:

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

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

if \_\_name\_\_ == "\_\_main\_\_":

    main()

