1. Implement a perceptron model to identify numbers with a circular or round shape from the MNIST dataset. The numbers with a round shape are: {0, 6, 8, 9}. Evaluate the model's accuracy on the test set and display a few test images with their predicted labels and actual labels.
2. The following code represents the architecture of a multilayer perceptron as show in the diagram below.

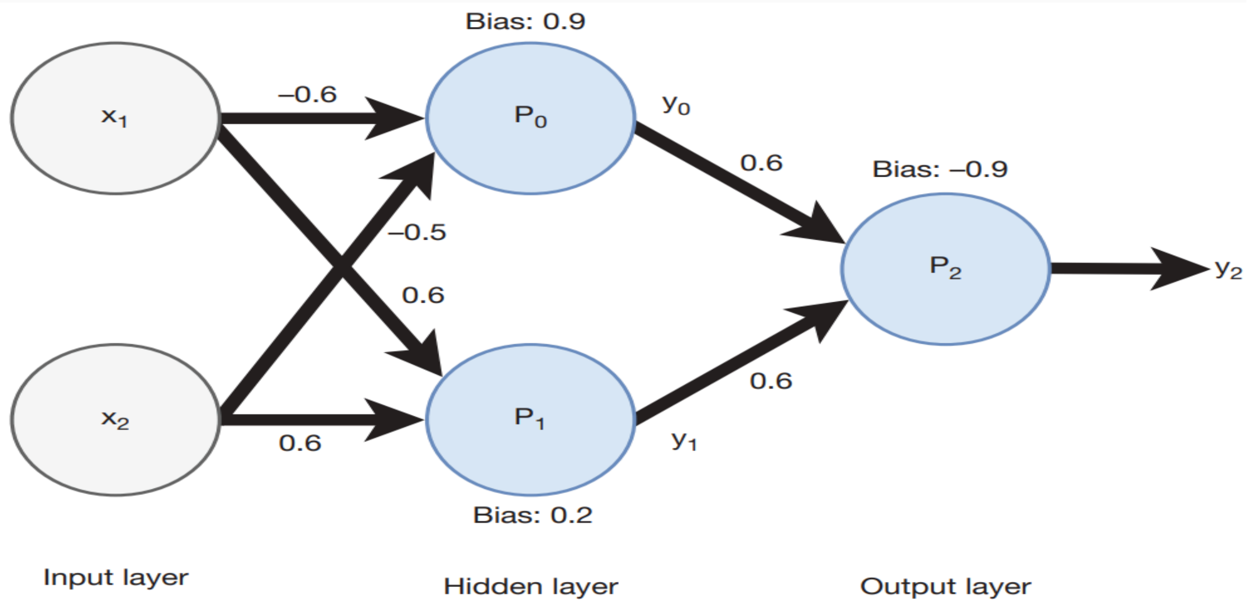

Figure 1-9 Two-level feedforward network implementing XOR function

$$\overline{(A\cdot B)}\cdot(A+B)$$

```
def compute_output(w, x):
    z = 0.0
    for i in range(len(w)):
        z += x[i] * w[i]  # Weighted sum of inputs
    return 1 if z >= 0 else -1

def perceptron_XOR(x1, x2):
    # Define the weights for the first perceptron (NAND)
    w1 = [1, 1]  # Weights for AND (perceptron 1)
    b1 = -1  # Bias 1

    # Define the weights for the second perceptron (OR)
    w2 = [1, 1]  # Weights for OR (perceptron 2)
    b2 = 0  # Bias 2
```

```
# First perceptron output (NAND)
p1_output = compute_output(w1, [x1, x2])

# Second perceptron output (OR)
p2_output = compute_output(w2, [x1, x2])

# Final output perceptron (XOR)
w3 = [-2, 2]  # Weights for the final perceptron
b3 = 0  # Bias 3
xor_output = compute_output(w3, [p1_output, p2_output])

return xor_output
```

Hint: train each perceptron individually—one for the NAND gate, another for the OR gate, and the third for the AND gate.