Q7. XOR with ReLU Network

We use the XOR ReLU network from the slides

$$h_1 = \text{ReLU}(x_1 + x_2), \quad h_2 = \text{ReLU}(x_1 + x_2 - 1), \quad y = \text{ReLU}(h_1 - 2h_2)$$

Extend it with an additional hidden unit:

$$h_3 = \text{ReLU}(2x_1 - x_2)$$

and new output:

$$y = \text{ReLU}(h_1 - 2h_2 + h_3)$$

Tasks

a) Compute outputs for all four XOR inputs (0,0),(0,1),(1,0),(1,1).

1. $\mathbf{(x_1, x_2) = (0, 0)}$

- $h_1 = \text{ReLU}(0) = 0$
- $h_2 = \text{ReLU}(-1) = 0$
- $h_3 = \text{ReLU}(0) = 0$
- $y = \text{ReLU}(0 - 0 + 0) = \mathbf{0}$

2. $\mathbf{(0, 1)}$

- $h_1 = \text{ReLU}(1) = 1$
- $h_2 = \text{ReLU}(0) = 0$
- $h_3 = \text{ReLU}(-1) = 0$
- $y = \text{ReLU}(1 - 0 + 0) = \mathbf{1}$

3. $\mathbf{(1, 0)}$

- $h_1 = \text{ReLU}(1) = 1$
- $h_2 = \text{ReLU}(0) = 0$
- $h_3 = \text{ReLU}(2) = 2$
- $y = \text{ReLU}(1 - 0 + 2) = \mathbf{3}$

4. $\mathbf{(1, 1)}$

- h_1 = ReLU(2) = 2
- h_2 = ReLU(1) = 1
- h_3 = ReLU(1) = 1
- y = ReLU(2 − 2·1 + 1) = ReLU(1) = **1**

Summary table (y):
(0,0) → 0
(0,1) → 1
(1,0) → 3
(1,1) → 1

(With a 0/positive threshold for class 1, this corresponds to labels: 0,1,1,1.)

b) Compare the decision boundary with the original 2-hidden-unit XOR network.

Original XOR network (with $y = \mathrm{ReLU}(h_1 - 2h_2)$) has kinks/lines at $\mathbf{x_1 + x_2 = 0}$ and $\mathbf{x_1 + x_2 = 1}$, producing class 1 only in the two "off-diagonal" unit squares.

The **extended network adds $\mathbf{h_3 = \mathrm{ReLU}(2x_1 - x_2)}$**, introducing an **additional boundary at $\mathbf{2x_1 - x_2 = 0}$**.
This extra positive contribution **expands the positive region**, notably turning (1,1) positive.

c) Does this extension still compute XOR exactly? If not, which inputs differ?

No. XOR requires outputs (0,1,1,0) on the four corners, but the extended network yields (0,1,1,1).
The **mismatch is at (1,1)** (it should be 0, but this model outputs 1).