Single Layer Perceptron

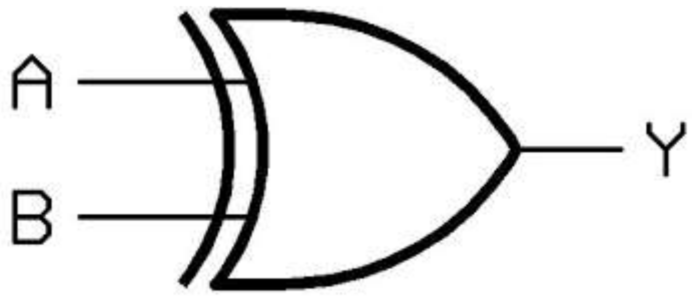
CT-466 | Week 2 - Lecture 4

Instructor: Mehar Fatima Shaikh

XOR problem

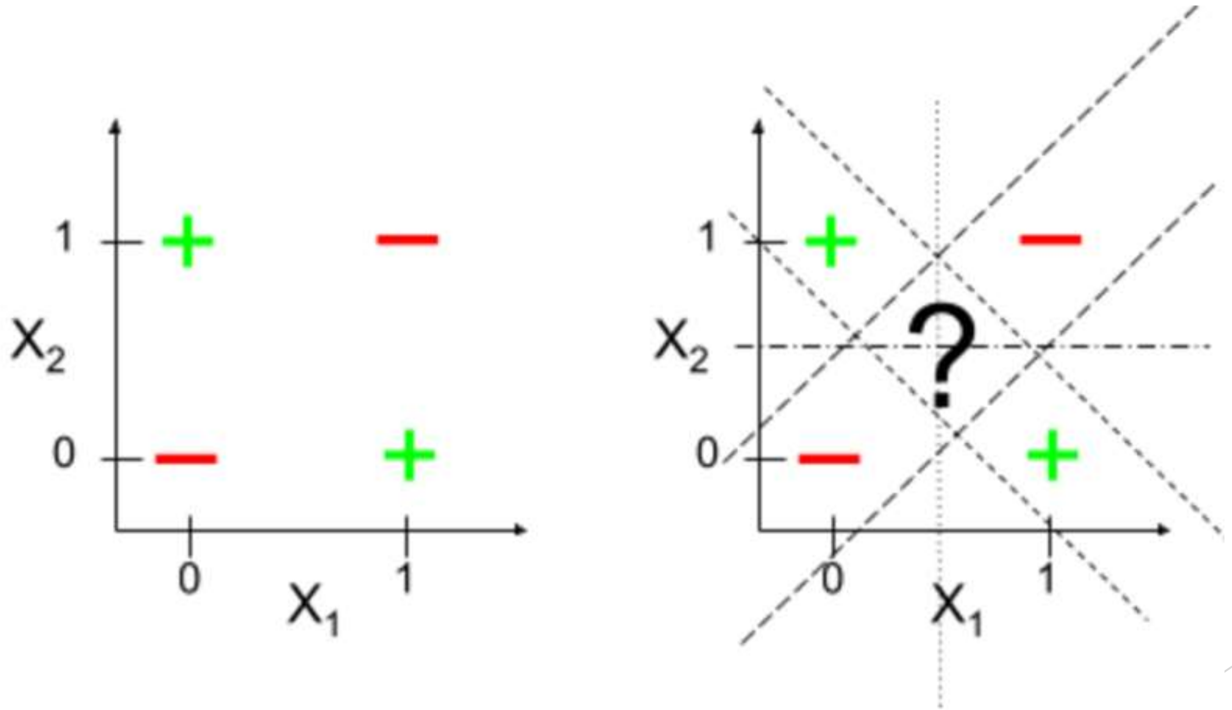
- XOR (exclusive OR) outputs 1 only when exactly one input is 1; otherwise, it outputs 0. It’s a classic example that a single perceptron cannot learn (non-linearly separable), motivating multi-layer networks.

Input 1 (x_1)	**Input 2 (x_2)**	**XOR**
0	0	0
0	1	1
1	0	1
1	1	0



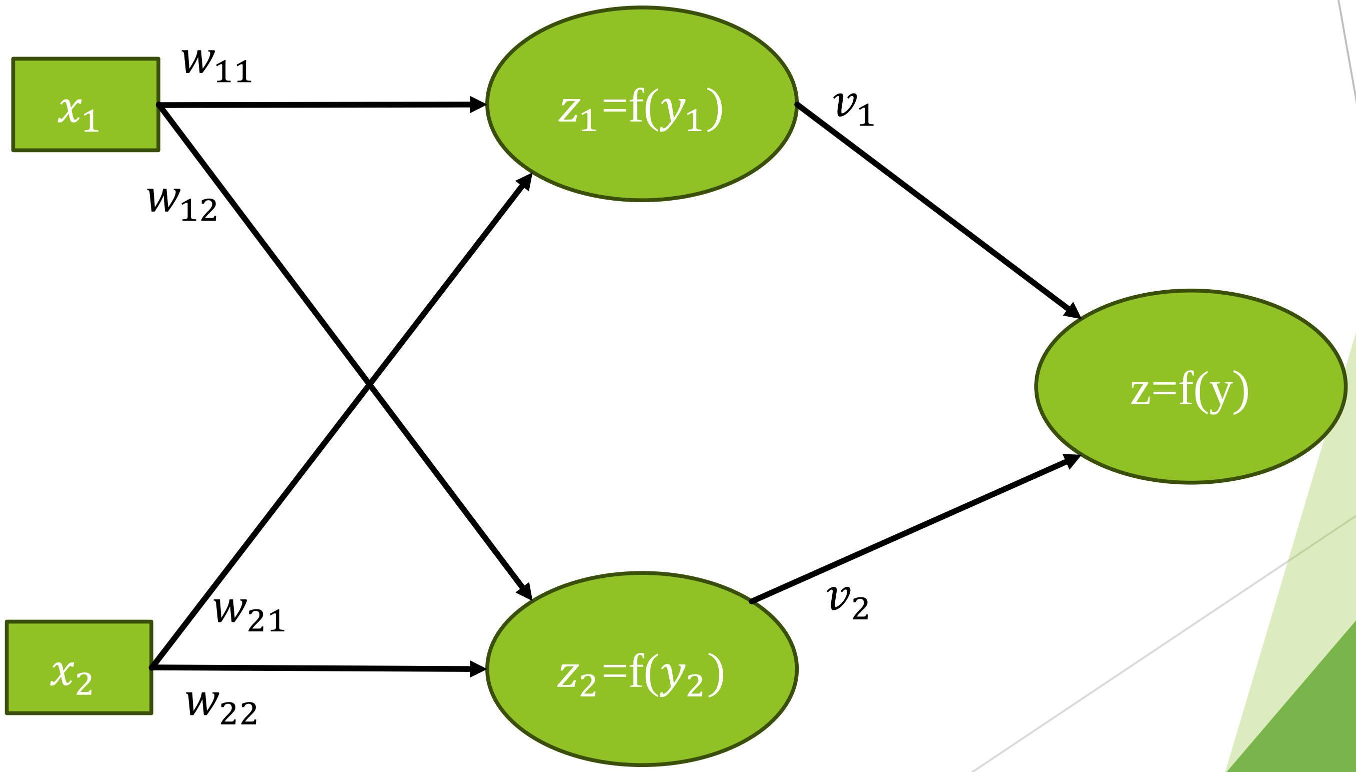
Why a single perceptron can't solve XOR

- A single perceptron draws one straight line (a linear decision boundary) in the (x_1, x_2) plane. For XOR, the 1's are at (0,1) and (1,0), while the 0's are at (0,0) and (1,1). No single line can separate those two classes hence non-linear separability. (Geometric intuition: the positive points sit on opposite corners; any straight line that groups them together also includes at least one negative point.).



How a Multi-Layer Perceptron (MLP) solves XOR

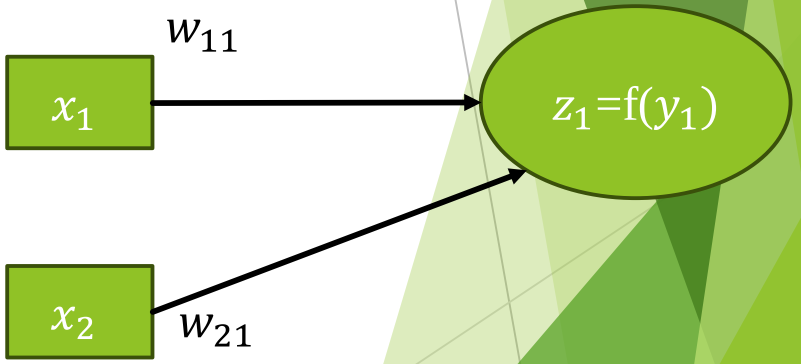
- Build XOR from simpler gates that a perceptron **can** do (like AND/OR/NOT):
- z_1=f(y_1) $= x_1\overline{x_2}$
- z_2= f(y_2) $= \overline{x_1}\ x_2$
- $Z\ = f\,(y) =\ z_1\ or\ z_2$
=> $f\,(y) = v_1 z_1 + v_2 z_2$



z_1=f(y_1) $= x_1\overline{x_2}$

Input 1 (x_1)	Input 2 (x_2)	$\overline{x_2}$	$z_1 = x_1\overline{x_2}$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

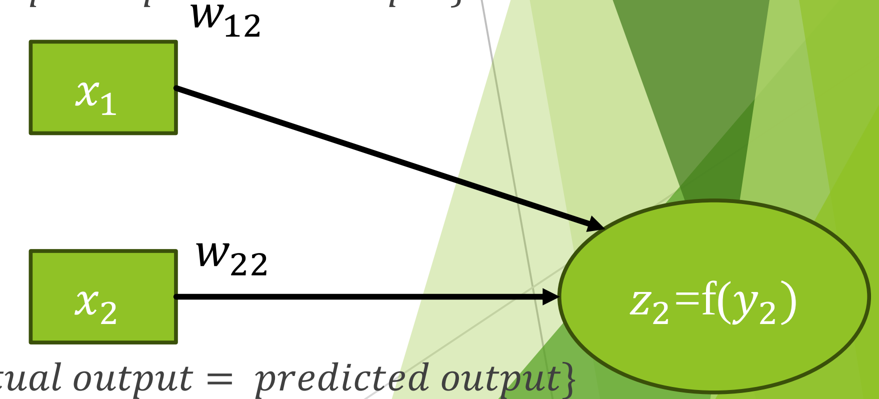
- Assume the initial weights $w_{11} = w_{21} = 1$
- $z = f(y) = \begin{cases} 1, & y \geq 1 \\ 0, & y < 1 \end{cases}$ (Here the threshold of activation function is 1)
- Learning rate ($n = 1.5$)
- (x_1, x_2) = (0,0) → $y_1 = w_i x_i$ = (1*0)+(1*0) = 0 => z = f(y_1) $= f(0) = 0$ { $actual\ output = predicted\ output$}
- Hence no weights update.
- (x_1, x_2) = (0,1) → $y_1 = w_i x_i$ = (1*0)+(1*1) = 1 => z = f(y_1) $= f(1) = 1$ { $actual\ output \neq predicted\ output$}
- Hence weights update
- $w^2 = w^1 + n(actual\ output - predicted\ output)* x_i$
- $w_{11}^2 = 1 + 1.5(0 - 1)$*0 = 1
- $w_{21}^2 = 1 + 1.5(0 - 1)$*1 = -0.5
- Now weights $w_{11} = 1\ and\ w_{21} = -0.5$
- (x_1, x_2) = (0,0) → $y_1 = w_i x_i$ = (1*0)+(-0.5*0) = 0 => z = f(y_1) $= f(0) = 0$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (0,1) → $y_1 = w_i x_i$ = (1*0)+(-0.5*1) = -0.5 => z = f(y_1) $= f(-0.5) = 0$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (1,0) → $y_1 = w_i x_i$ = (1*1)+(-0.5*0) = 1 => z = f(y_1) $= f(1) = 1$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (1,1) → $y_1 = w_i x_i$ = (1*1)+(-0.5*1) = 0.5 => z = f(y_1) $= f(0.5) = 0$ { $actual\ output = predicted\ output$}



Input 1 (x_1)	$\overline{x_1}$	Input 2 (x_2)	$z_2=\overline{x_1}x_2$
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0

z_2= f(y_2) $= \overline{x_1}\, x_2$

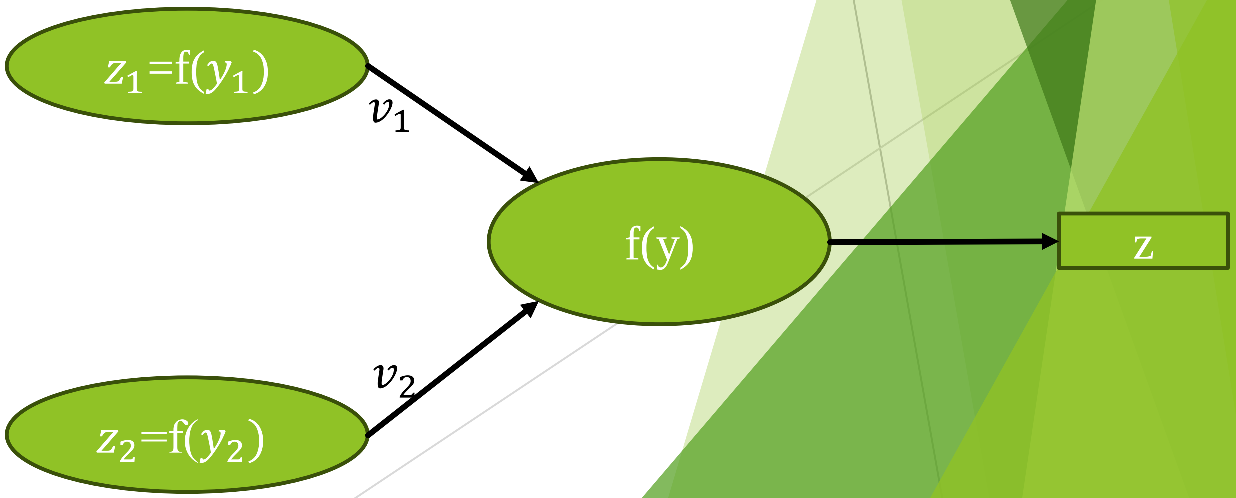
- Assume the initial weights $w_{11} = w_{21} = 1$
- $z = f(y) = \begin{cases} 1, & y \geq 1 \\ 0, & y < 1 \end{cases}$ (Here the threshold of activation function is 1)
- Learning rate ($n = 1.5$)
- (x_1, x_2) = (0,0) → $y_2 = w_i x_i$ = (1*0)+(1*0) = 0 => z = f(y_2) $= f(0) = 0$ { $actual\ output = predicted\ output$}
- Hence no weights update.
- (x_1, x_2) = (0,1) → $y_2 = w_i x_i$ = (1*0)+(1*1) = 1 => z = f(y_2) $= f(1) = 1$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (1,0) → $y_2 = w_i x_i$ = (1*1)+(1*0) = 1 => z = f(y_2) $= f(1) = 1$ { $actual\ output \neq predicted\ output$}
- Hence weights update
- $w^2 = w^1 + n(actual\ output - predicted\ output) * x_i$
- $w_{12}^2 = 1 + 1.5(0-1)$*1 = -0.5
- $w_{22}^2 = 1 + 1.5(0-1)$*0 = 1
- Now weights $w_{12} = -0.5$ and $w_{22} = 1$
- (x_1, x_2) = (0,0) → $y_2 = w_i x_i$ = (-0.5*0)+(1*0) = 0 => z = f(y_2) $= f(0) = 0$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (0,1) → $y_2 = w_i x_i$ = (-0.5*0)+(1*1) = 1 => z = f(y_2) $= f(1) = 1$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (1,0) → $y_2 = w_i x_i$ = (-0.5*1)+(1*0) = -0.5 => z = f(y_2) $= f(-0.5) = 0$ { $actual\ output = predicted\ output$}
- (x_1, x_2) = (1,1) → $y_2 = w_i x_i$ = (-0.5*1)+(1*1) = 0.5 => z = f(y_2) $= f(0.5) = 0$ { $actual\ output = predicted\ output$}



$z = f(y) = z_1 \text{ or } z_2 \Rightarrow f(y) = v_1 z_1 + v_2 z_2$

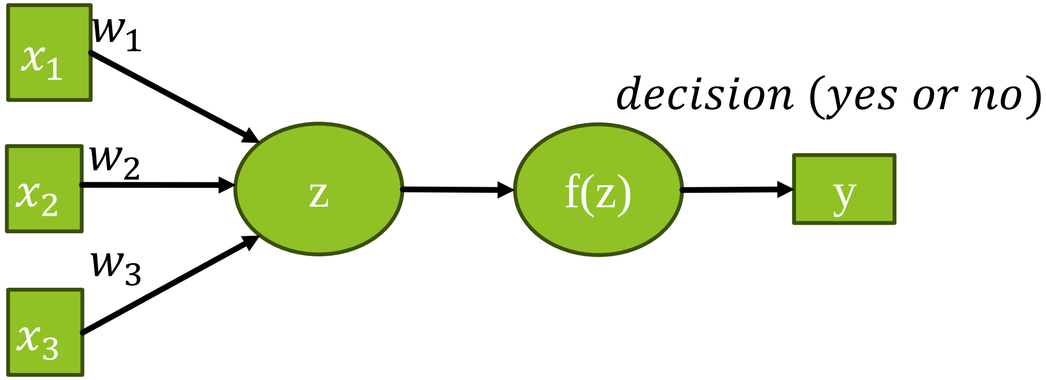
- Assume the initial weights $v_1 = v_2 = 1$
- $z = f(y) = \begin{cases} 1, & y \geq 1 \\ 0, & y < 1 \end{cases}$ (Here the threshold of activation function is 1)
- Learning rate $(n = 1.5)$
- $(x_1, x_2) = (0,0) \rightarrow y_2 = w_i x_i = (1*0)+(1*0) = 0$ => z = f(y) $= f(0) = 0$ $\{$ *actual output = predicted output* $\}$
- $(x_1, x_2) = (0,1) \rightarrow y_2 = w_i x_i = (1*0)+(1*1) = 1$ => z = f(y) $= f(1) = 1$ $\{$ *actual output = predicted output* $\}$
- $(x_1, x_2) = (1,0) \rightarrow y_2 = w_i x_i = (1*1)+(1*0) = 1$ => z = f(y) $= f(1) = 1$ $\{$ *actual output = predicted output* $\}$
- $(x_1, x_2) = (1,1) \rightarrow y_2 = w_i x_i = (1*1)+(1*1) = 0$ => z = f(y) $= f(0) = 0$ $\{$ *actual output = predicted output* $\}$

Input 1 (x_1)	Input 2 (x_2)	z_1	z_2	z
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0



Practice Problem

- **Initial Conditions**
- Initial Weights:
- w1 = 0.75
- w2 = 0.5
- w3 = -0.6
- Bias (b) = 0
- Learning Rate (α) = 0.2



Input Dataset

x1	x2	x3	y
1.0	1.0	1	1
9.4	6.4	1	-1
2.5	2.1	1	1
8.0	7.7	1	-1
0.5	2.2	1	1
7.9	8.4	1	-1
7.0	7.0	1	-1
2.8	0.8	1	1
1.2	8.0	1	1
7.8	6.1	1	-1

Solution

- **Input #1**
- $x = [1.0, 1.0, 1]$, $y = 1$
- $f(net) = f(0.75*1.0 + 0.5*1.0 - 0.6*1) = f(0.65) = 1$
- Since predicted = actual, no weight update
- $W^2 = W^1$
- **Input #2**
- $x = [9.4, 6.4, 1]$, $y = -1$
- $f(net) = f(0.75*9.4 + 0.5*6.4 - 0.6*1) = f(9.65) = 1$
- Mismatch: predicted $\neq$ actual $\Rightarrow$ apply learning rule
- $W^2 = [0.75, 0.5, -0.6] - 0.4*[9.4, 6.4, 1] = [-3.01, -2.06, -1.0]$

Solution

- **Input #3**
- $x = [2.5, 2.1, 1]$, $y = 1$
- $f(net) = f(-3.01*2.5 - 2.06*2.1 - 1.0*1) = f(-12.851) = -1$
- Mismatch $\Rightarrow$ apply learning rule
- $W^3 = [-3.01, -2.06, -1.0] + 0.4*[2.5, 2.1, 1] = [-2.01, -1.22, -0.6]$
- **Input #4**
- $x = [8.0, 7.7, 1]$, $y = -1$
- $f(net) = f(-2.01*8.0 - 1.22*7.7 - 0.6) = f(-26.074) = -1$
- Predicted = actual $\Rightarrow$ No update
- $W^4 = W^3$

Solution

- **Input #5**
- $x = [0.5, 2.2, 1]$, $y = 1$
- $f(net) = f(-2.01*0.5 -1.22*2.2 -0.6) = f(-4.289) = -1$
- Mismatch ⇒ apply learning rule
- $W^5 = [-2.01, -1.22, -0.6] + 0.4*[0.5, 2.2, 1] = [-1.81, -0.34, -0.2]$
- **Input #6**
- $x = [7.9, 8.4, 1]$, $y = -1$
- $f(net) = f(-1.81*7.9 -0.34*8.4 -0.2) = f(-17.355) = -1$
- Predicted = actual ⇒ No update
- $W^6 = W^5$

Solution

- **Input #7**
- x = [7.0, 7.0, 1], y = -1
- f(net) = f(-1.81*7.0 -0.34*7.0 -0.2) = f(-15.25) = -1
- Predicted = actual ⇒ No update
- $W^7 = W^6$
- **Input #8**
- x = [2.8, 0.8, 1], y = 1
- f(net) = f(-1.81*2.8 -0.34*0.8 -0.2) = f(-5.54) = -1
- Mismatch ⇒ apply learning rule
- W^8 = [-1.81, -0.34, -0.2] + 0.4*[2.8, 0.8, 1] = [-0.69, -0.02, 0.2]

Solution

- **Input #9**
- x = [1.2, 8.0, 1], y = 1
- f(net) = f(-0.69*1.2 -0.02*8.0 + 0.2) = f(-0.788) = -1
- Mismatch ⇒ apply learning rule
- W^9 = [-0.69, -0.02, 0.2] + 0.4*[1.2, 8.0, 1] = [-0.21, 3.18, 0.6]
- **Input #10**
- x = [7.8, 6.1, 1], y = -1
- f(net) = f(-0.21*7.8 +3.18*6.1 +0.6) = f(18.36) = 1
- Mismatch ⇒ apply learning rule
- W^{10} = [-0.21, 3.18, 0.6] - 0.4*[7.8, 6.1, 1] = [-3.33, 0.74, 0.2]
- Final Weights = [-3.33, 0.74, 0.2]