

# Foundations of Deep Learning



ALF

Alfredo Canziani

 @alfcnz

# A simple program

2 bit addition with logic gates

# Programming

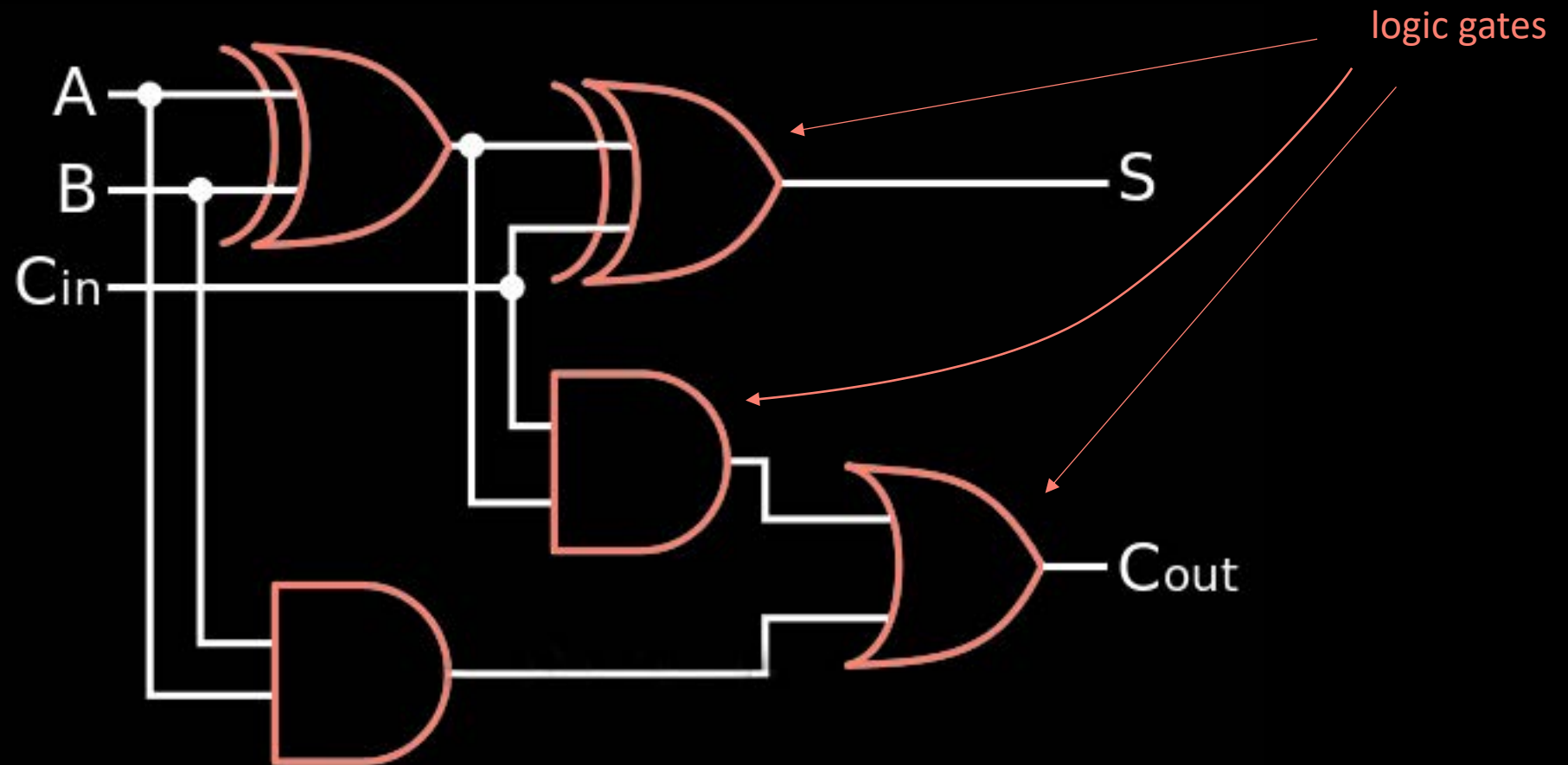


- Algorithm: addition (half-adder)
- Data: 2 binary digits (bits)
- Answer: sum

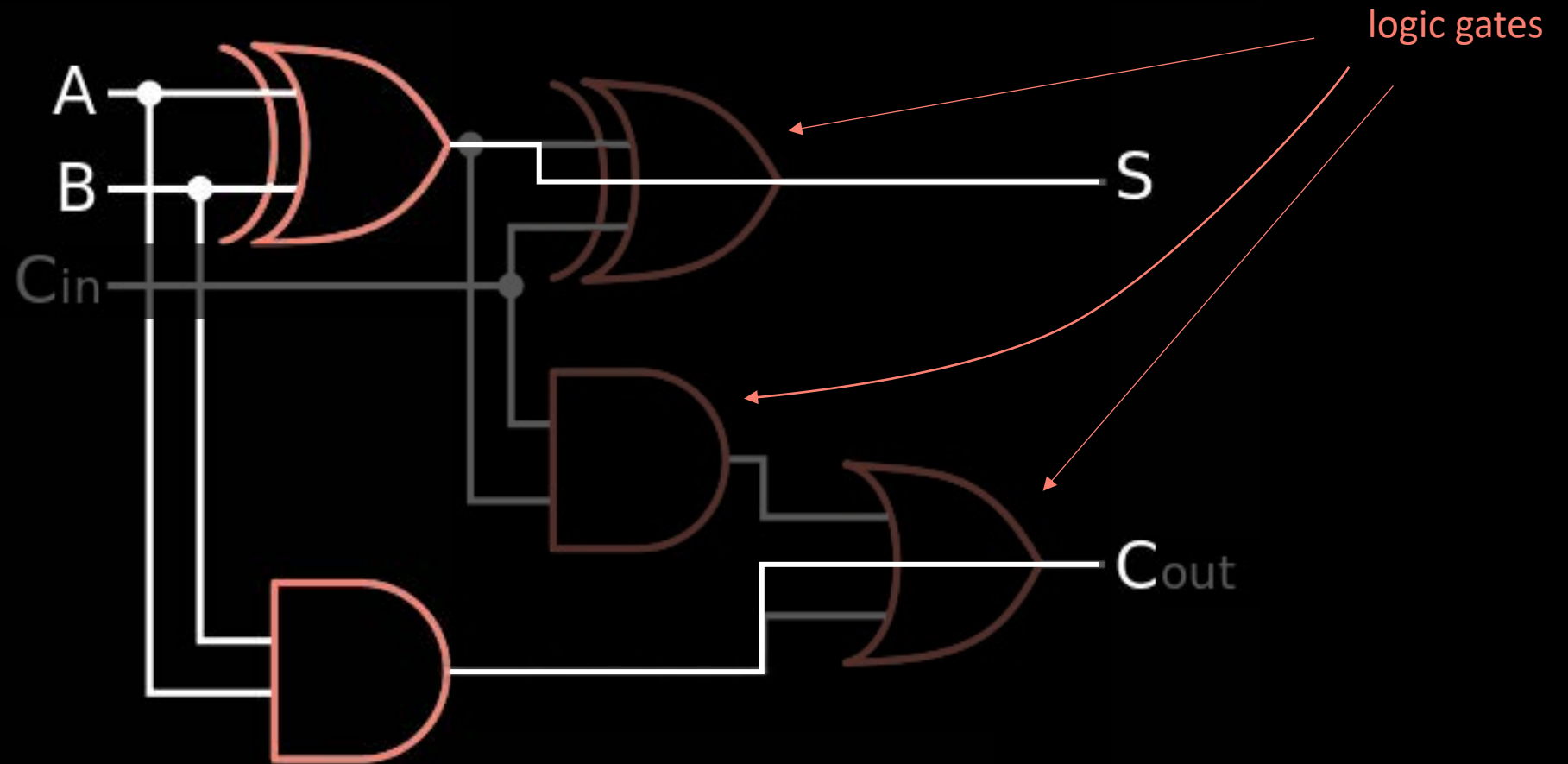
# Number base

0	1:00	1:00:00	1d	1d + 0h	0 <sub>2</sub>	0 <sub>16</sub>
1	1:01	1:00:01	1d + 1h	1d + 1h	1 <sub>2</sub>	1 <sub>16</sub>
2	1:02	1:00:02	1d + 2h	1d + 2h	10 <sub>2</sub>	2 <sub>16</sub>
⋮	⋮	⋮ ⋮	⋮	⋮	11 <sub>2</sub>	⋮
58	59:58	23:59:58	1d + 22h	6d + 22h	100 <sub>2</sub>	E <sub>16</sub>
59	59:59	23:59:59	1d + 23h	6d + 23h	101 <sub>2</sub>	F <sub>16</sub>
1m	1h	1d	2d	1w	110 <sub>2</sub>	10 <sub>16</sub>
1:01	1:00:01	1d + 1s	2d + 1h	1w + 1d	111 <sub>2</sub>	11 <sub>16</sub>

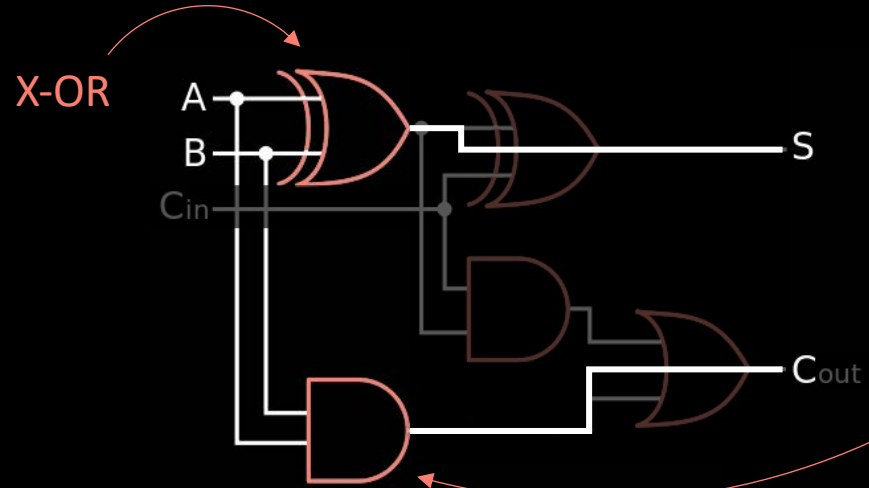
# Adder



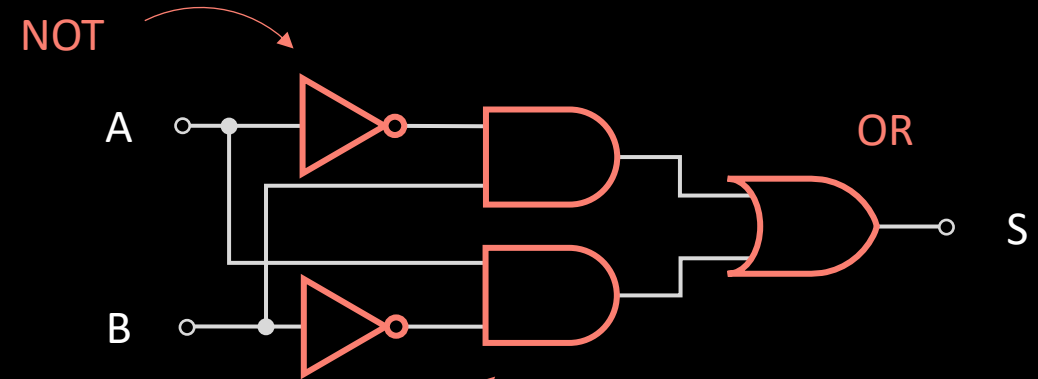
# Half-Adder



# Half-Adder

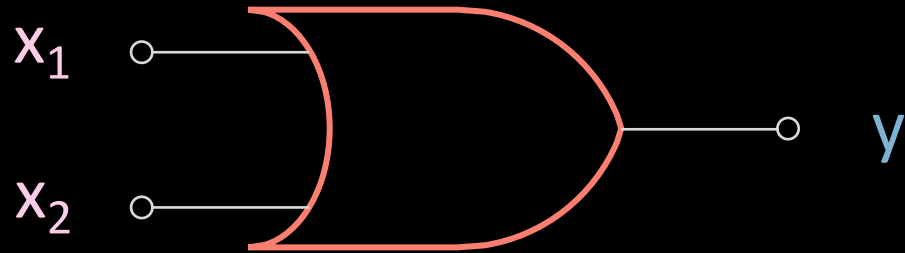


# X-OR



A		B	C	S
0	+	0	0	0
0	+	1	0	1
1	+	0	0	1
1	+	1	1	0

# OR gate



	$x_1$	$x_2$	$y$	
	0	0	0	false
false	0	1	1	true
true	1	0	1	true
	1	1	1	true

Diagram illustrating the truth table for an OR gate. The inputs  $x_1$  and  $x_2$  are shown, along with the output  $y$ . The output is 0 (false) only when both inputs are 0 (false). Otherwise, the output is 1 (true).

# AND gate

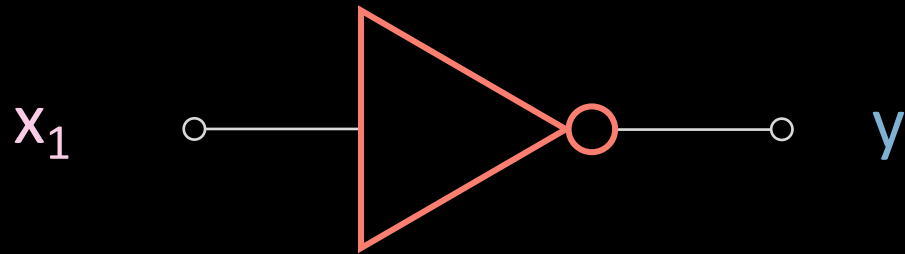


$x_1$	$x_2$	$y$
0	0	0
0	1	0
1	0	0
1	1	1

Diagram illustrating the truth table for an AND gate. The inputs  $x_1$  and  $x_2$  are shown, along with the output  $y$ . The output is 1 (true) only when both inputs are 1 (true). Otherwise, the output is 0 (false).

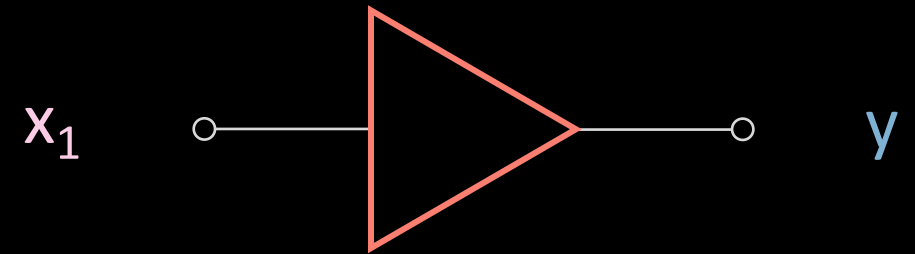


## NOT gate



$x_1$	$y$
0	1
1	0

## BUFFER gate

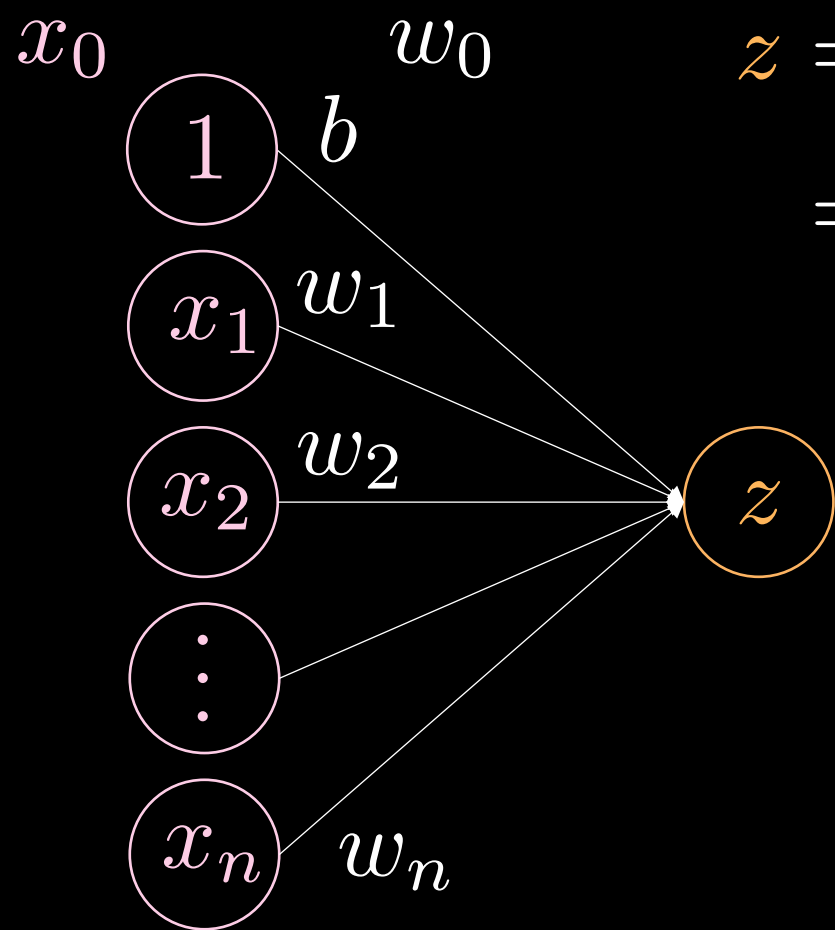


$x_1$	$y$
0	0
1	1

# The perceptron

AKA non-linear neuron

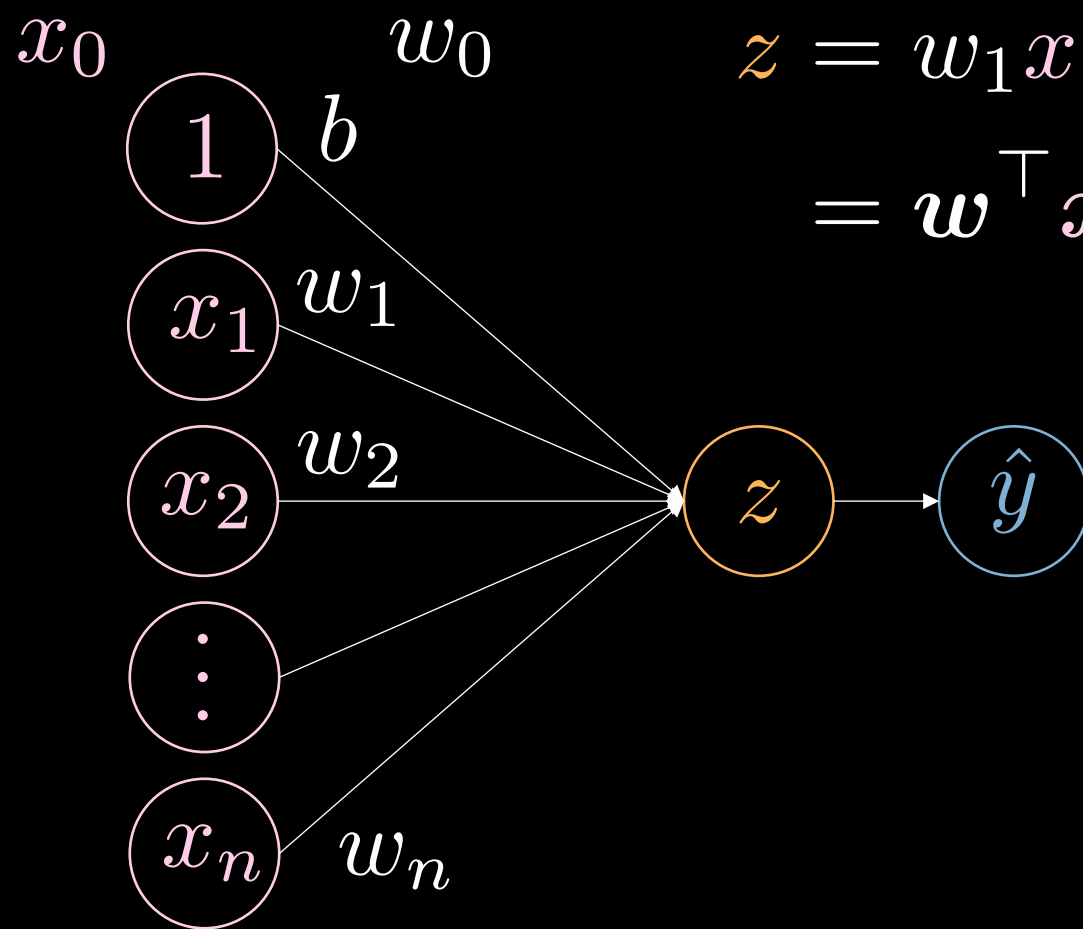
# Linear neuron



$$\begin{aligned} z &= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \\ &= \boldsymbol{w}^\top \boldsymbol{x} + b = \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}} \end{aligned}$$

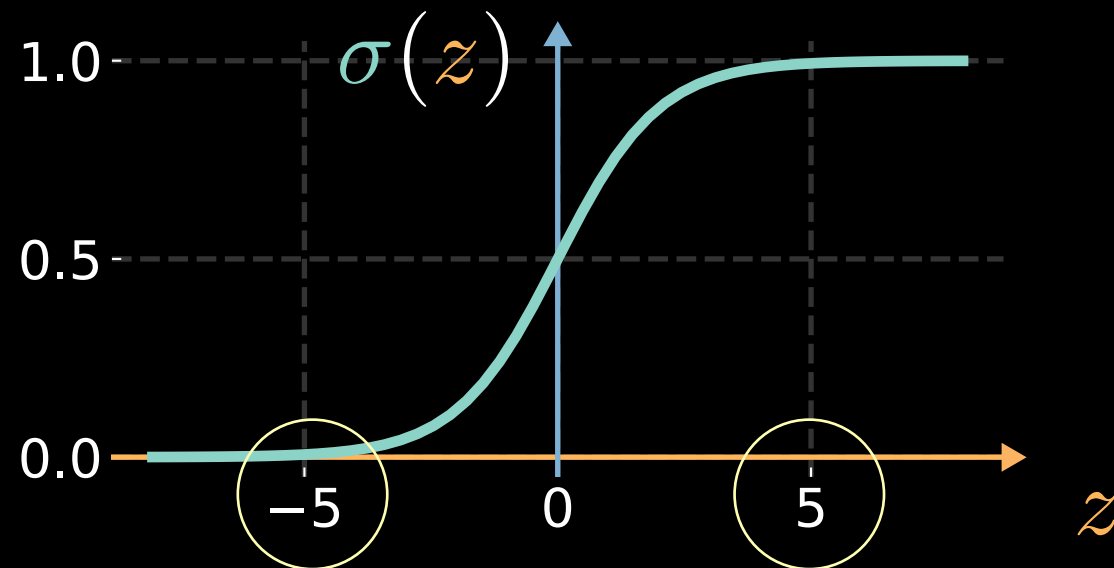
# Neuron / perceptron

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

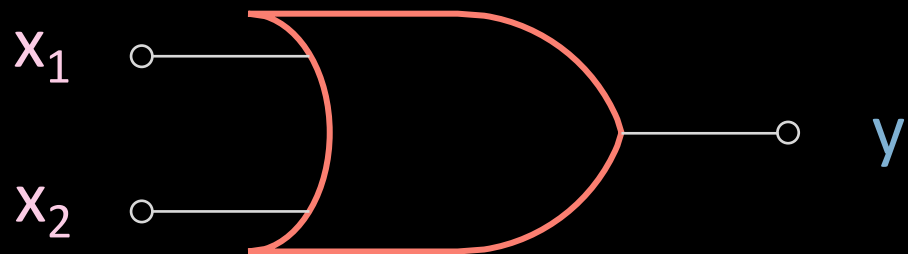


$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$
$$= \mathbf{w}^\top \mathbf{x} + b = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}$$

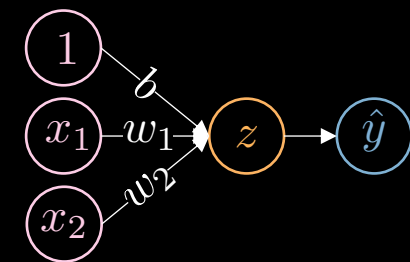
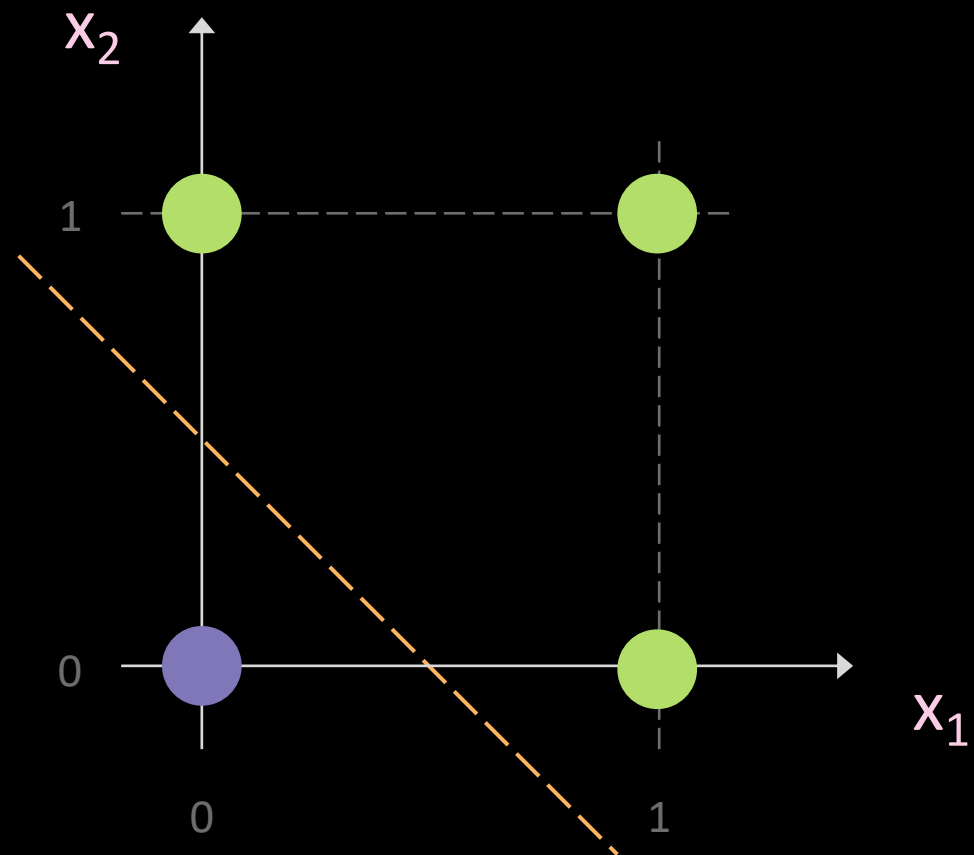
$$\hat{y} = \sigma(z)$$

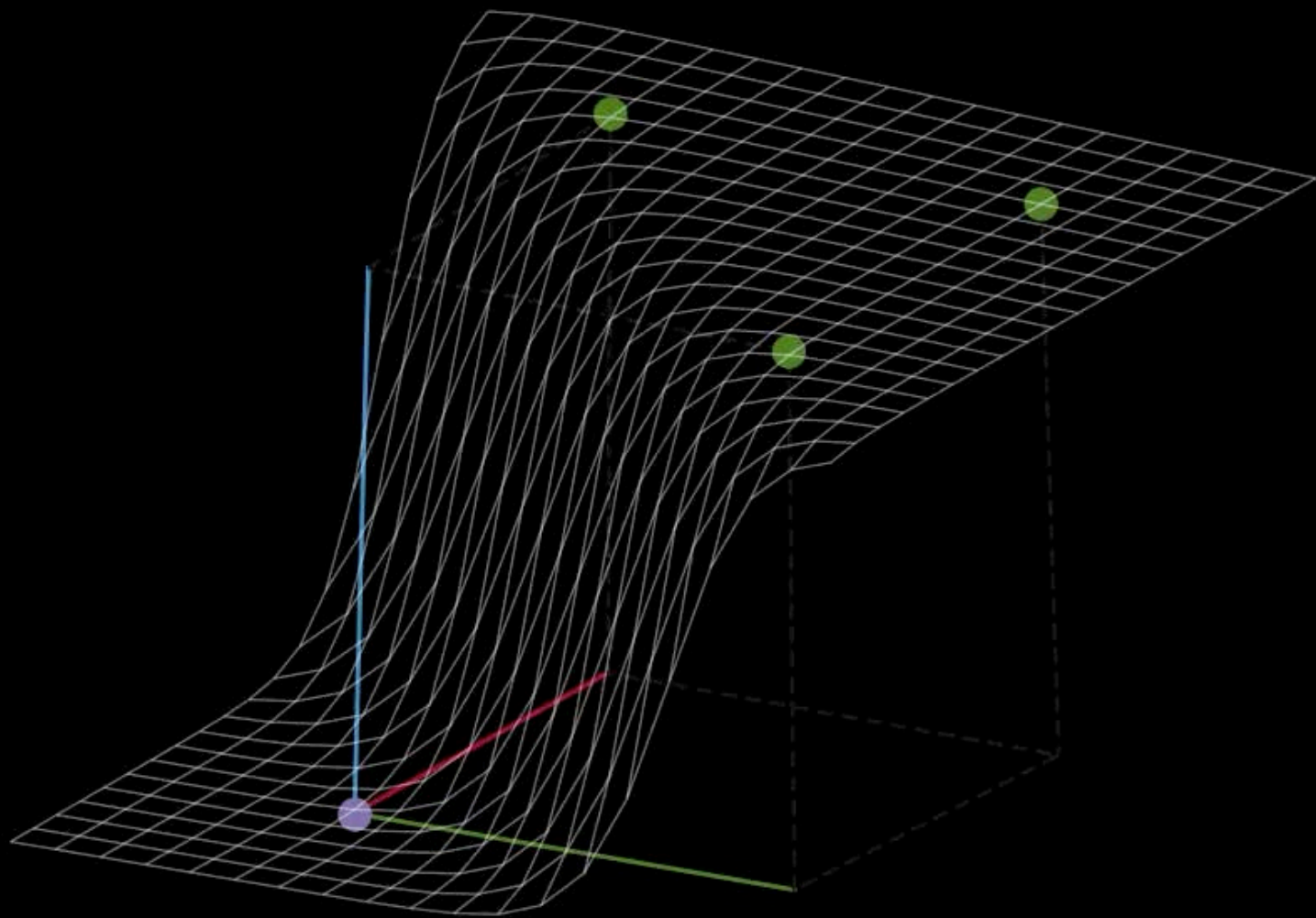


# OR neuron



$x_1$	$x_2$	$y$	$z$
0	0	0	$< -5$
0	1	1	$> +5$
1	0	1	$> +5$
1	1	1	$> +5$

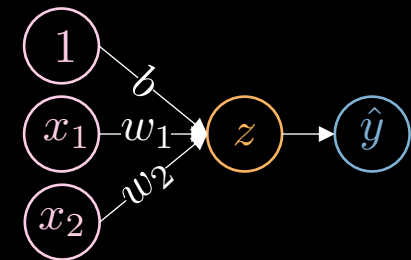
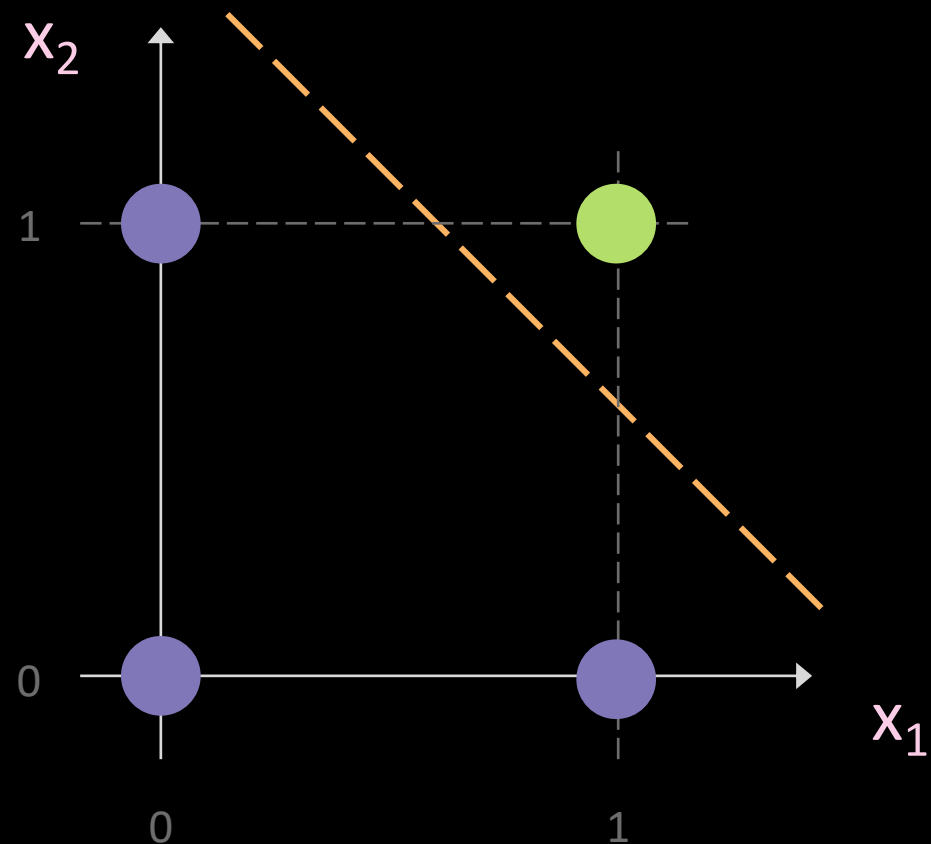


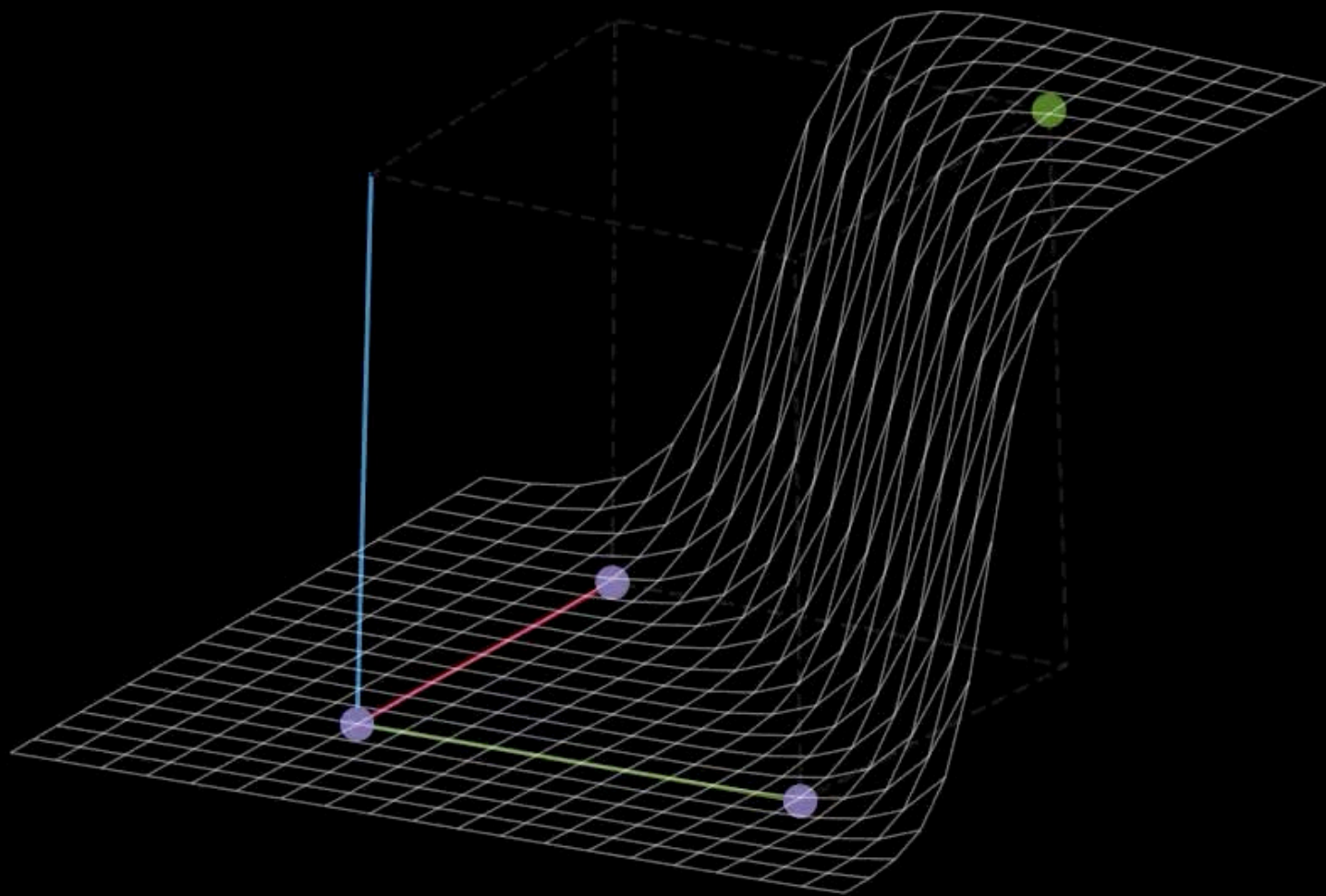


# AND neuron



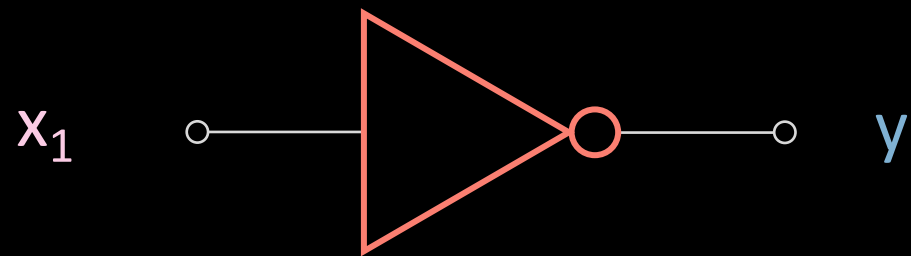
$x_1$	$x_2$	$y$	$z$
0	0	0	$< -5$
0	1	0	$< -5$
1	0	0	$< -5$
1	1	1	$> +5$



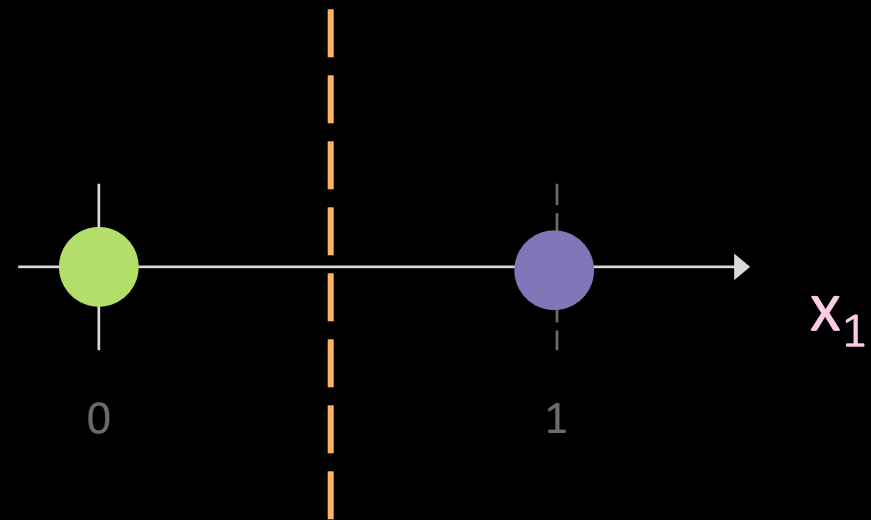
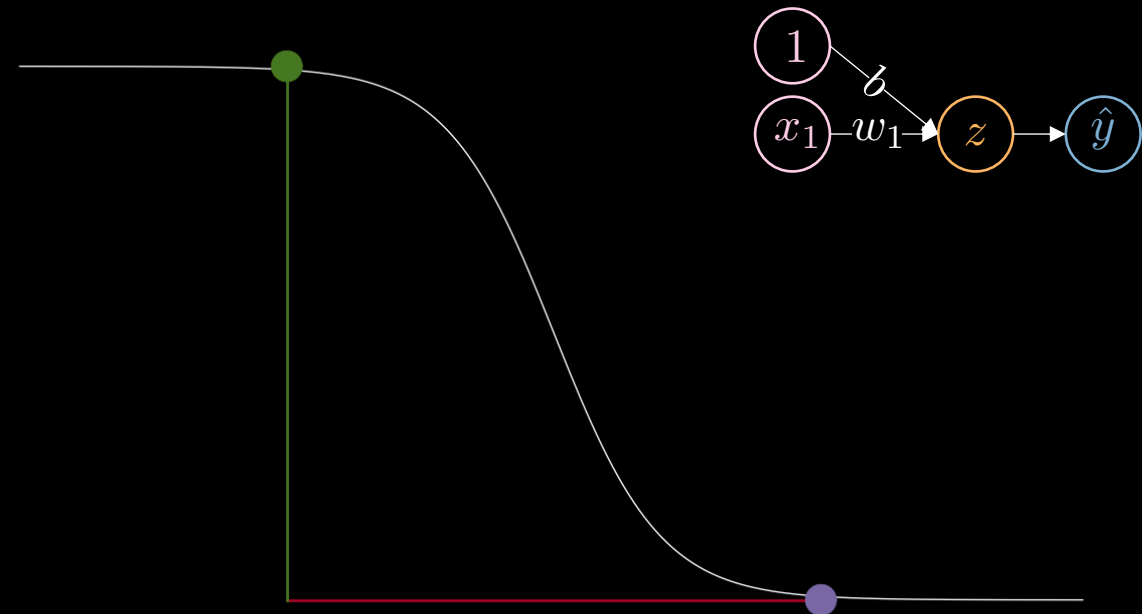




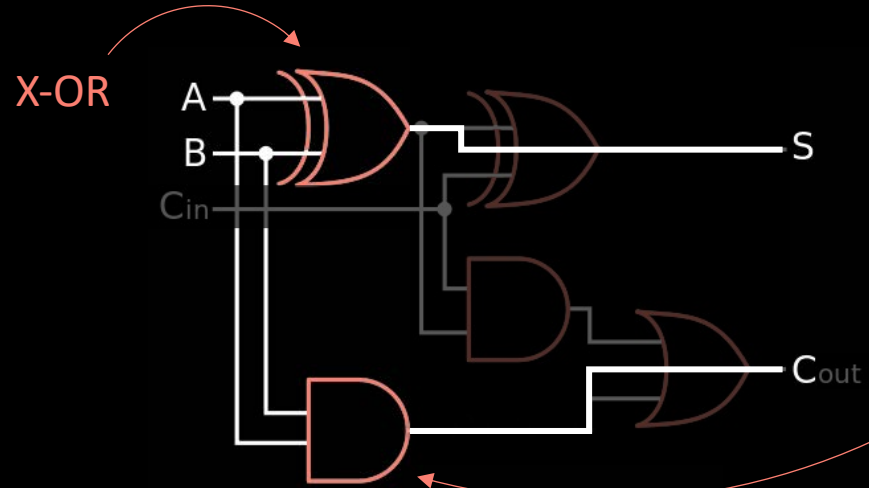
# NOT neuron



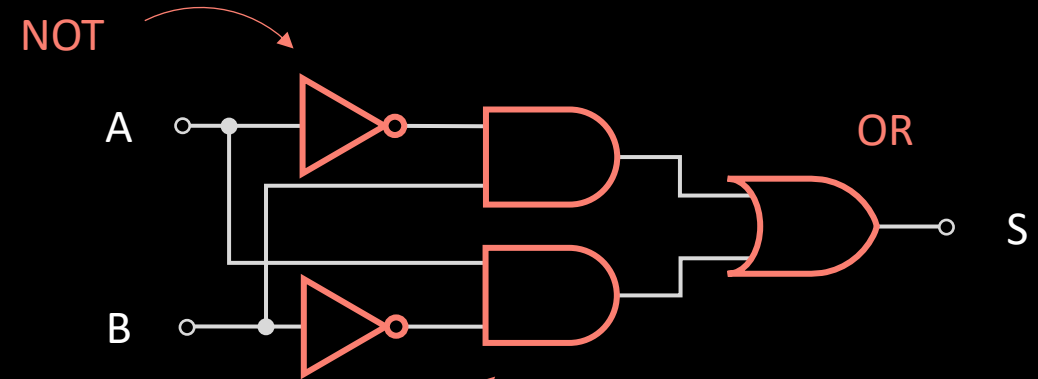
$x_1$	$y$	$z$
0	1	$> +5$
1	0	$< -5$



# Half-Adder



# X-OR



A		B	C	S
0	+	0	0	0
0	+	1	0	1
1	+	0	0	1
1	+	1	1	0