Foundations of Deep Learning

ALF

Alfredo Canziani

@alfcnz

A simple program

2 bit addition with logic gates

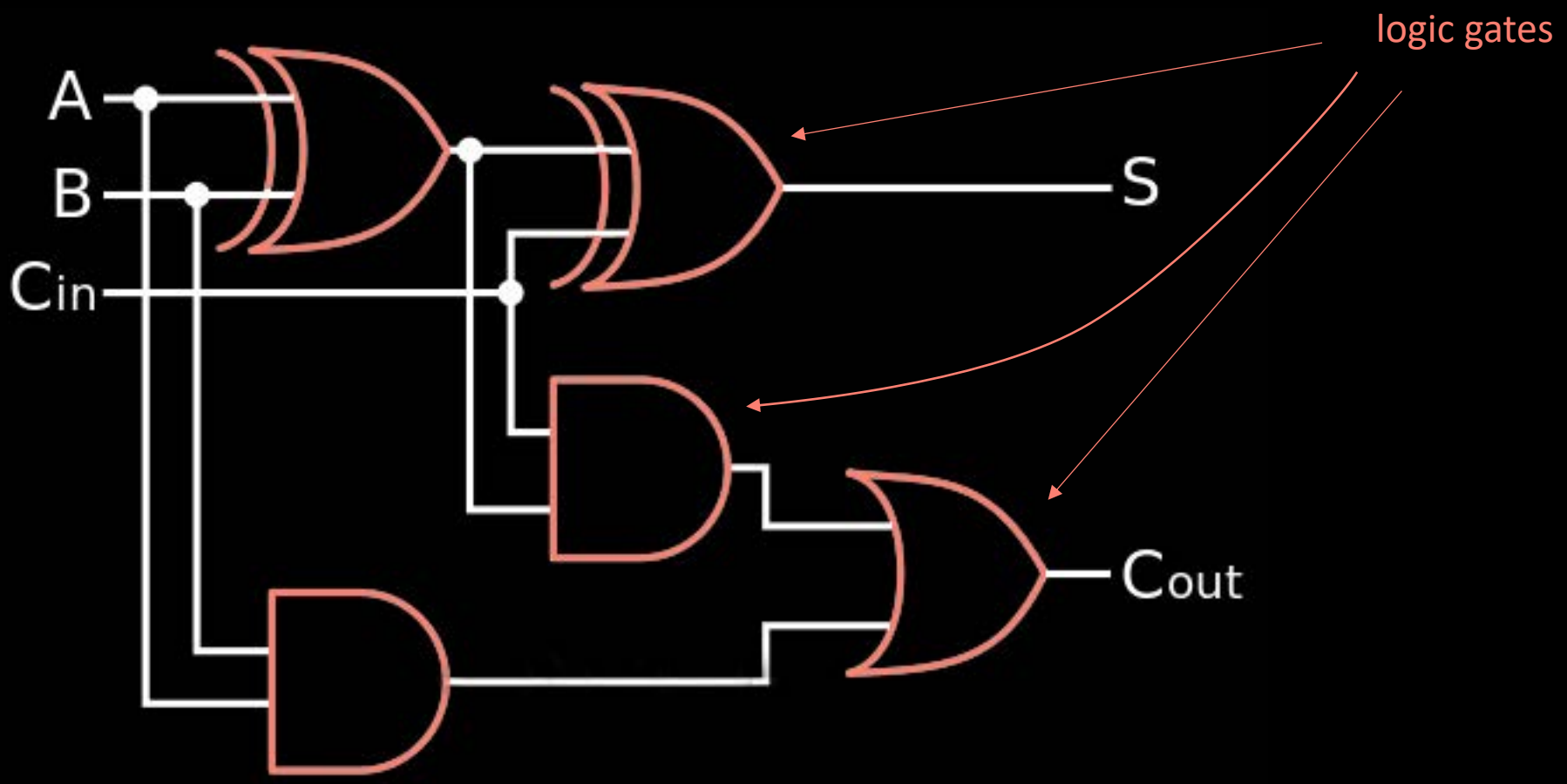
Programming

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

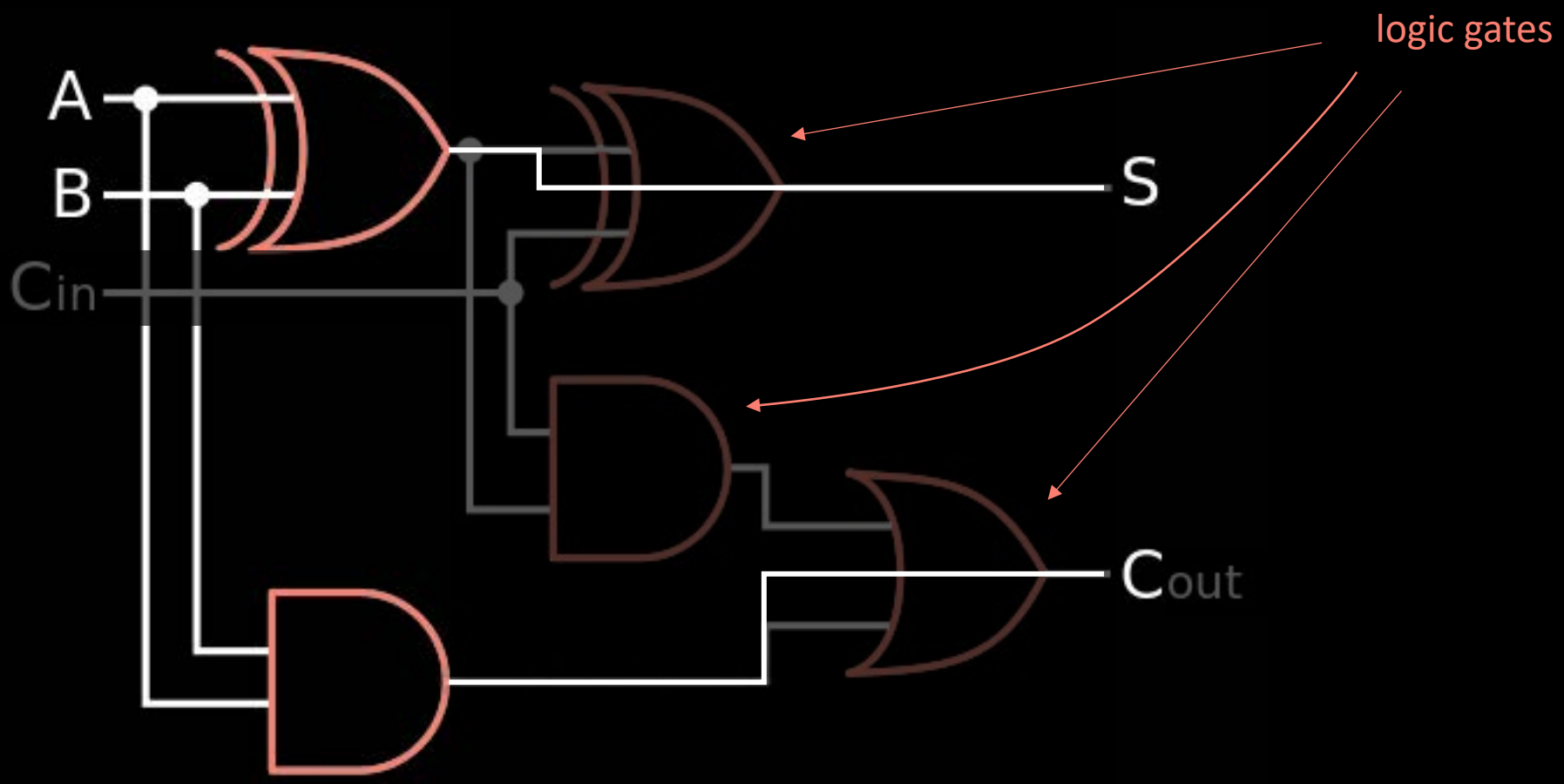
Number base

0	1:00	1:00:00	1d	1d + 0h	0_2	0_{16}
1	1:01	1:00:01	1d + 1h	1d + 1h	1_2	1_{16}
2	1:02	1:00:02	1d + 2h	1d + 2h	10_2	2_{16}
⋮	⋮	⋮ ⋮	⋮	⋮	11_2	⋮
58	59:58	23:59:58	1d + 22h	6d + 22h	100_2	E_{16}
59	59:59	23:59:59	1d + 23h	6d + 23h	101_2	F_{16}
1m	1h	1d	2d	1w	110_2	10_{16}
1:01	1:00:01	1d + 1s	2d + 1h	1w + 1d	111_2	11_{16}

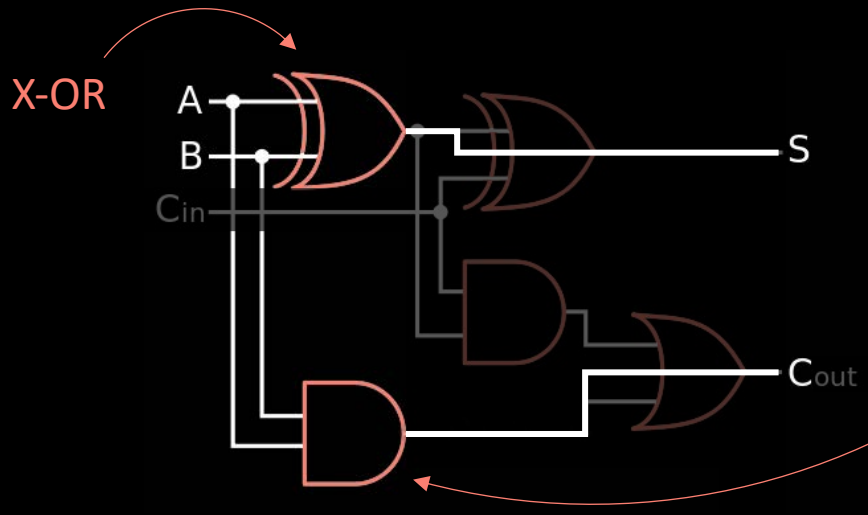
Adder



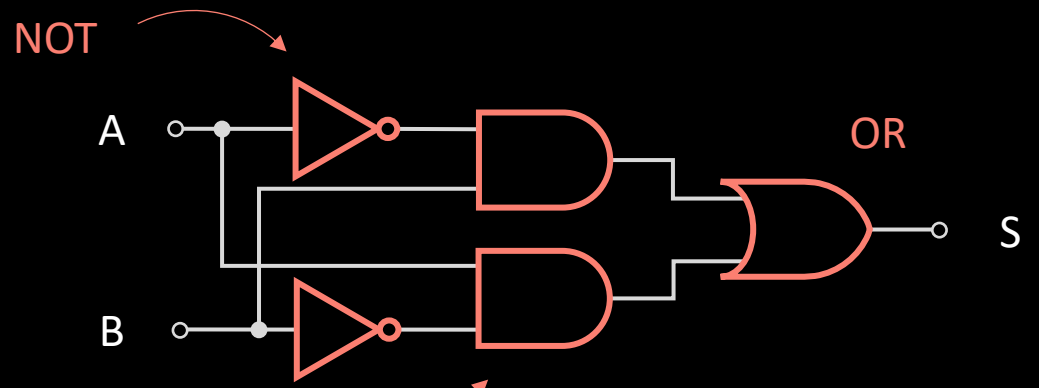
Half-Adder



Half-Adder

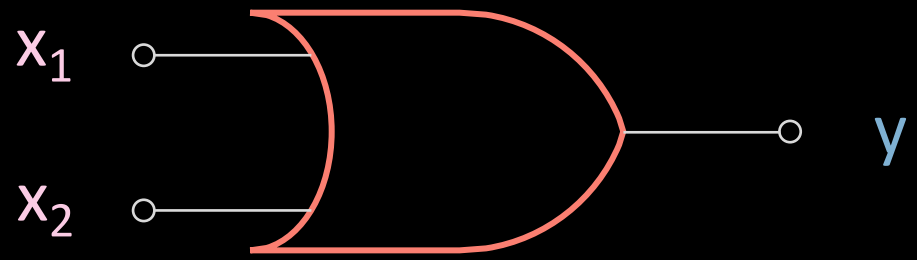


X-OR



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

OR gate



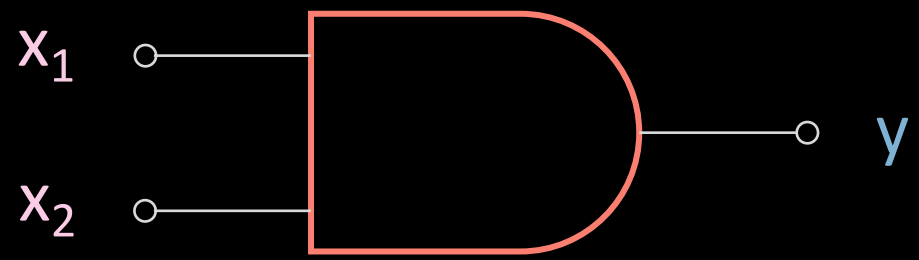
false

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

false

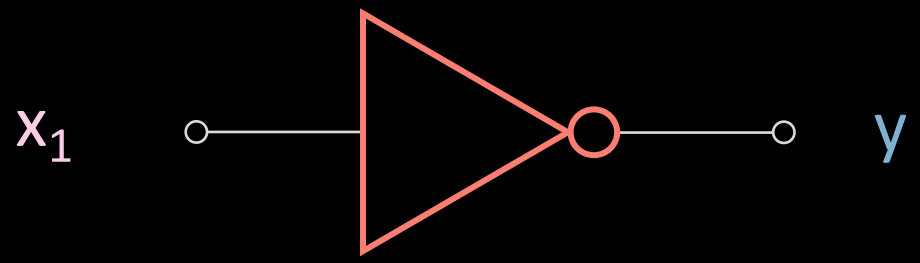
true

AND gate



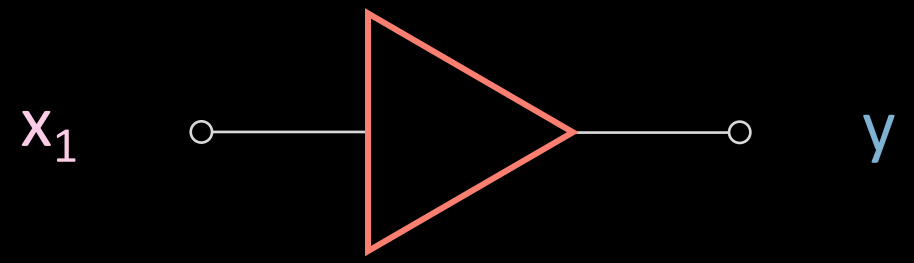
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

NOT gate



x_1	y
0	1
1	0

BUFFER gate

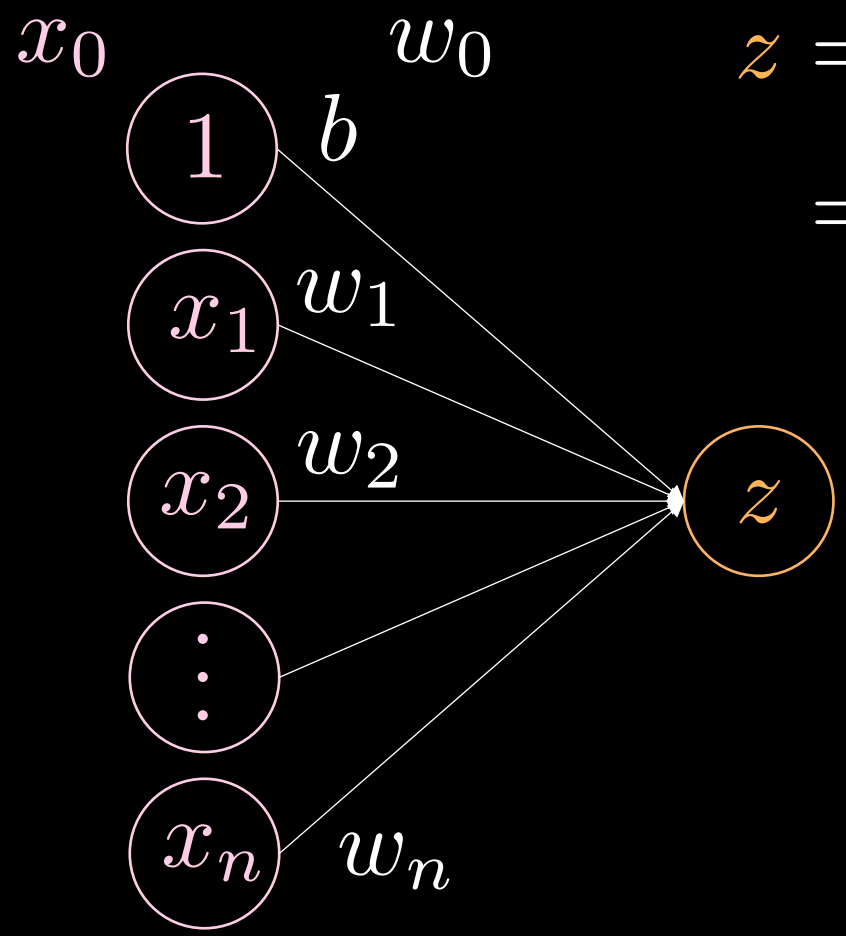


x_1	y
0	0
1	1

The perceptron

AKA non-linear neuron

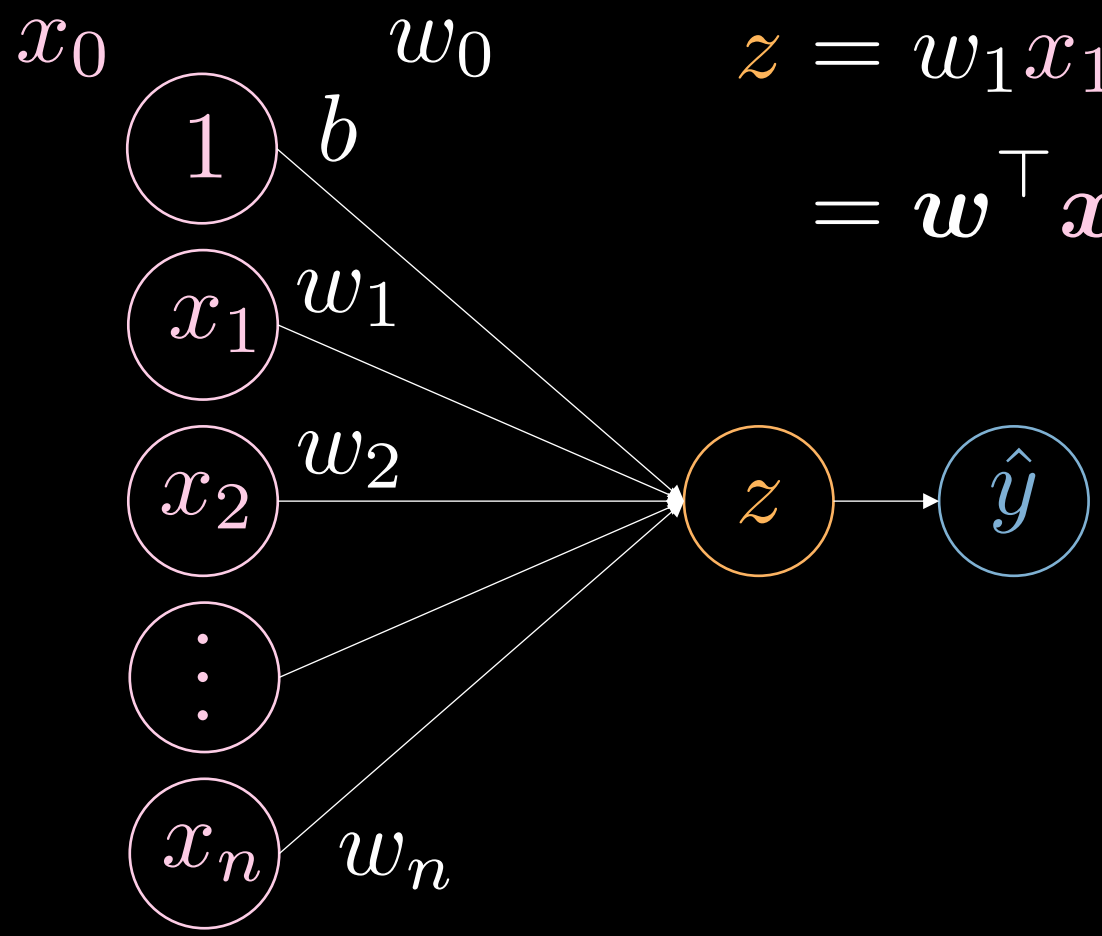
Linear neuron



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

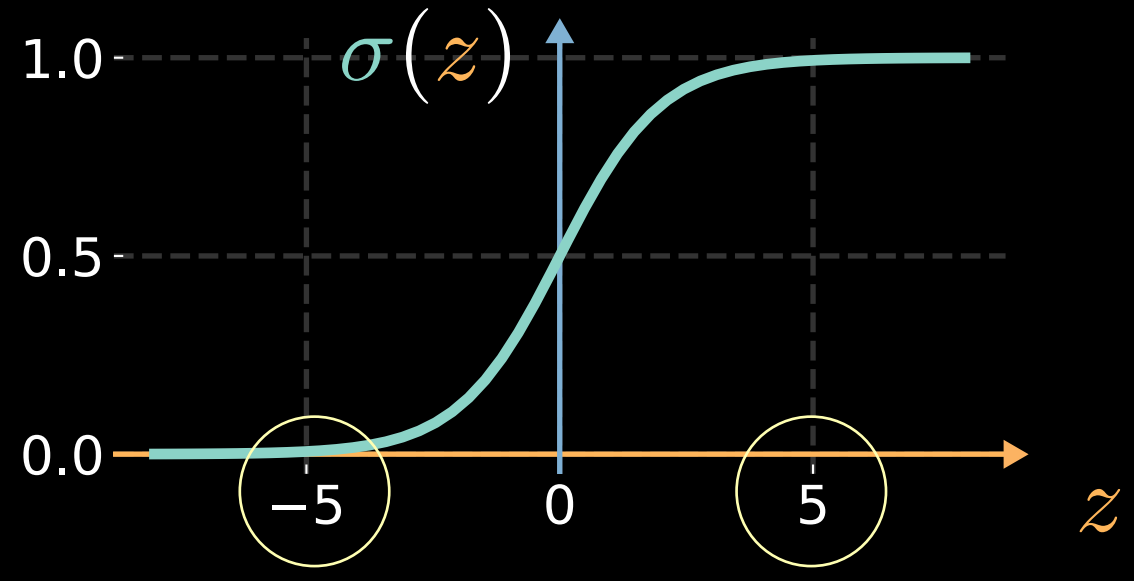
Neuron / perceptron

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

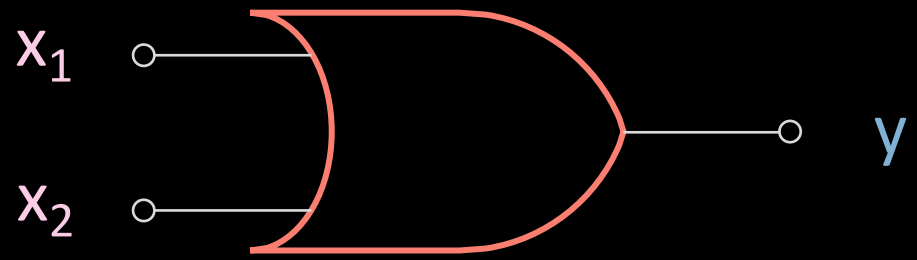


$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$
$$= \boldsymbol{w}^\top \boldsymbol{x} + b = \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}}$$

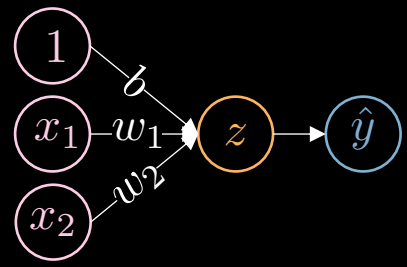
$$\hat{y} = \sigma(z)$$

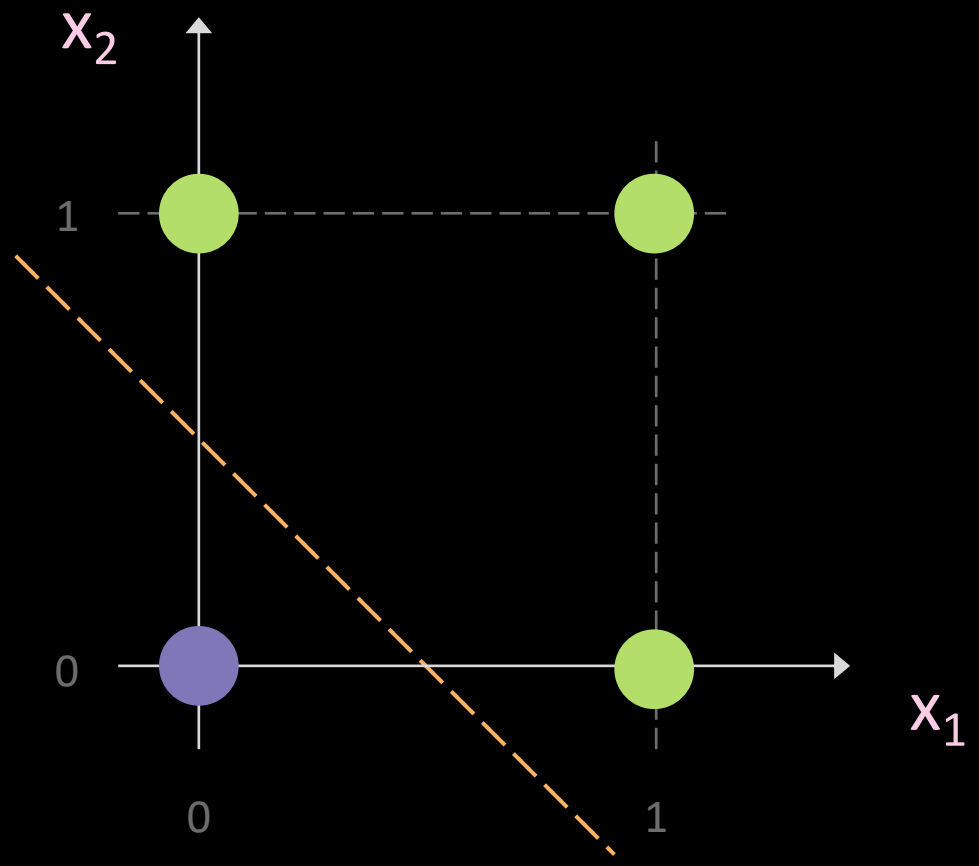


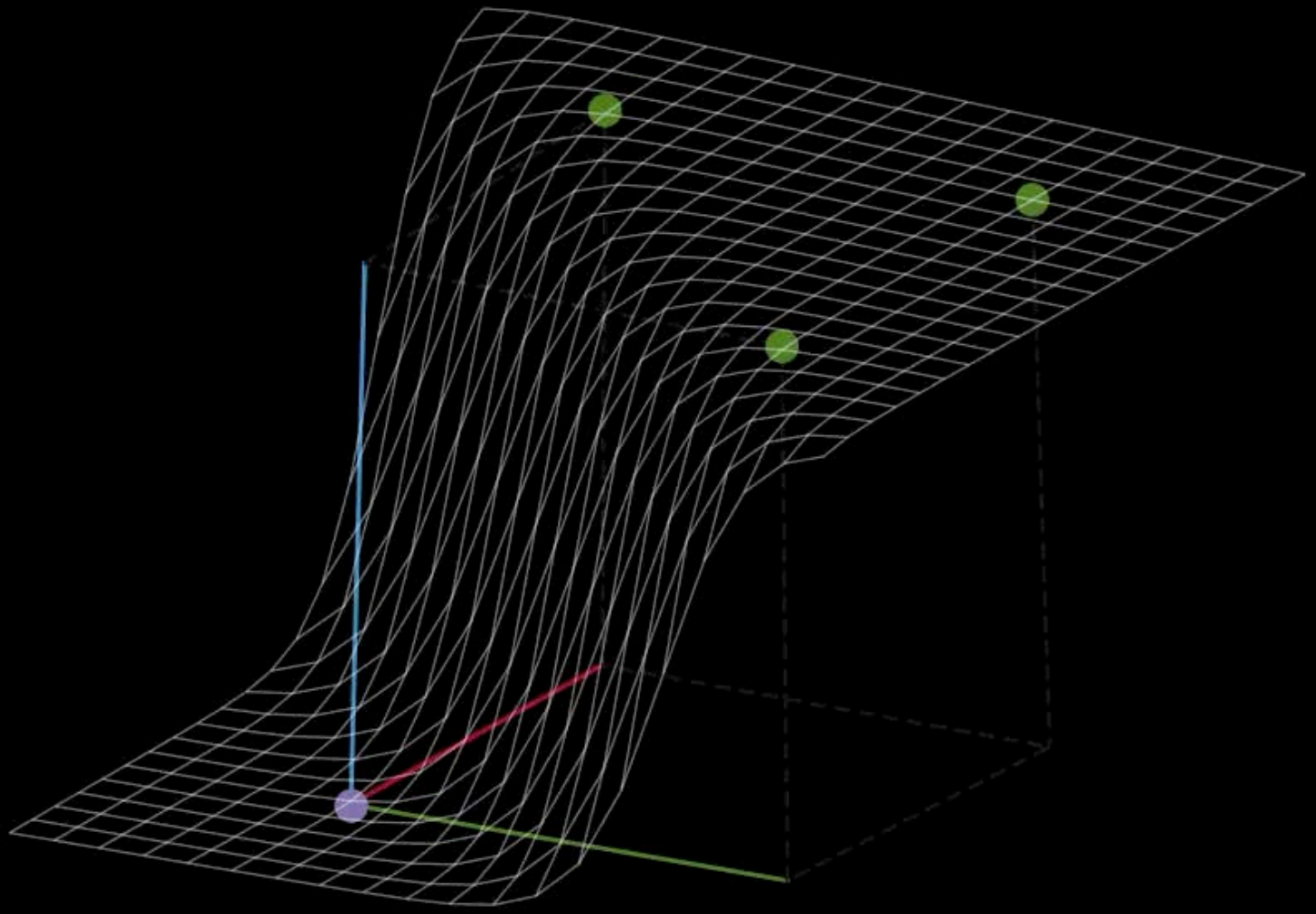
OR neuron



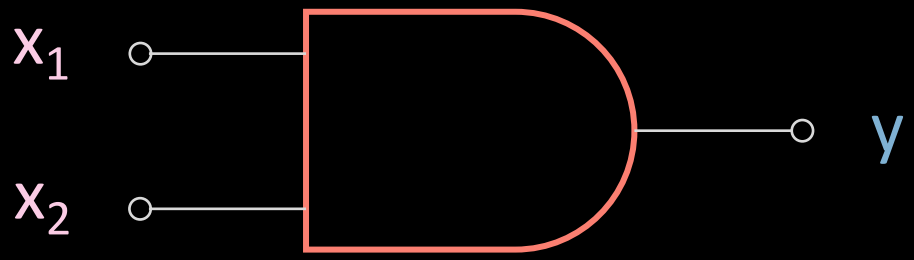
x_1	x_2	y	z
0	0	0	< –5
0	1	1	> +5
1	0	1	> +5
1	1	1	> +5



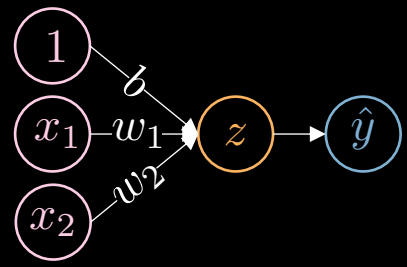


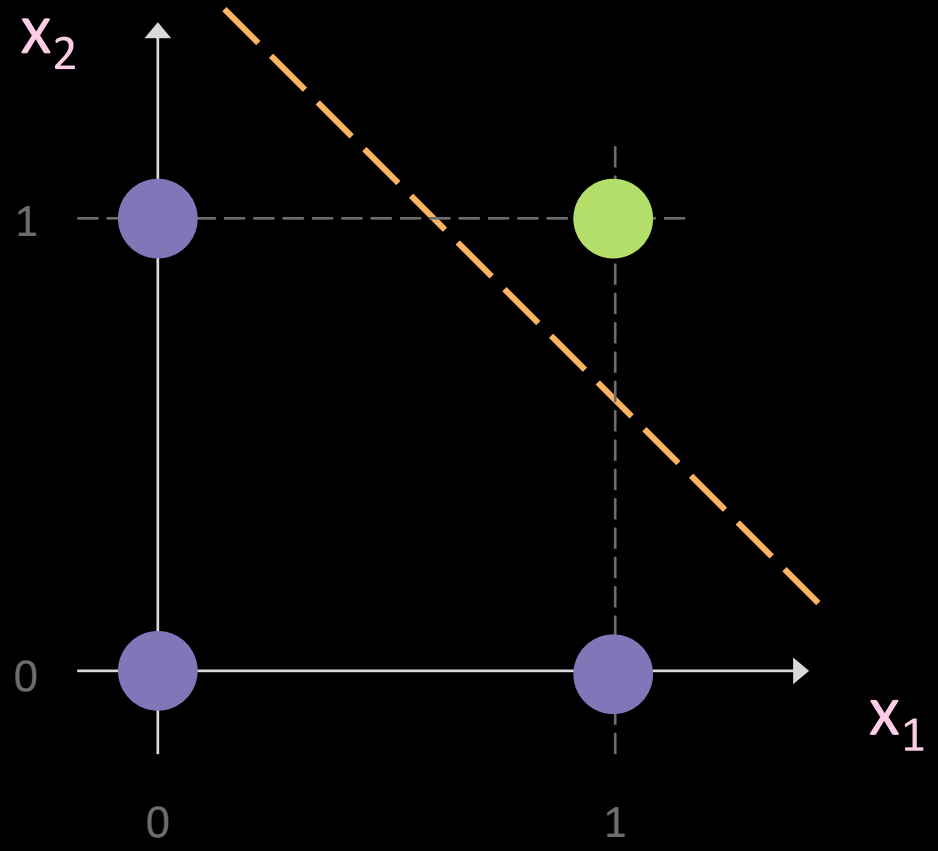


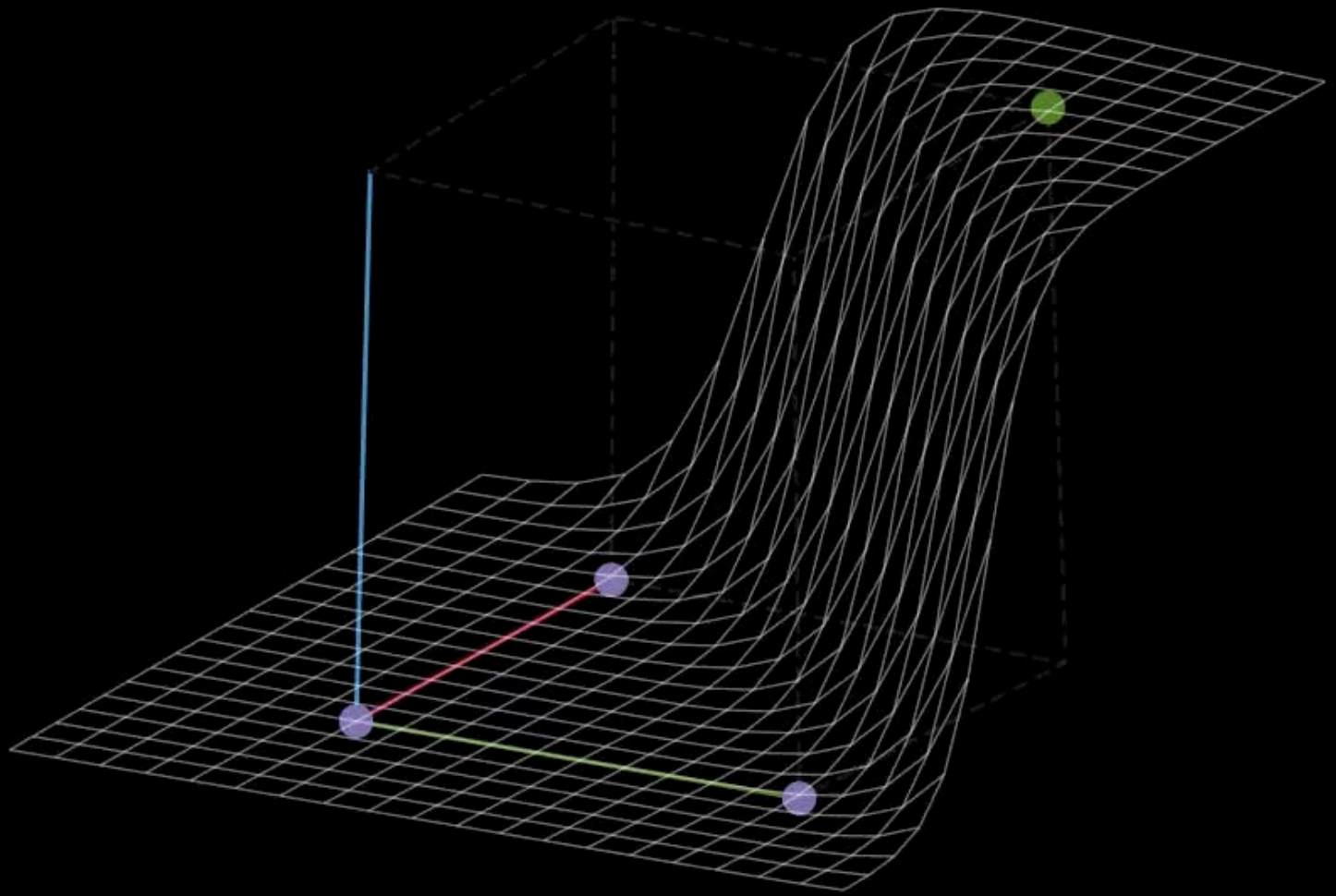
AND neuron



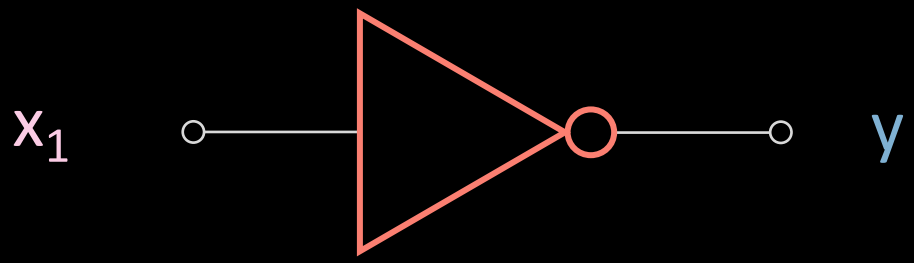
x_1	x_2	y	z
0	0	0	< –5
0	1	0	< –5
1	0	0	< –5
1	1	1	> +5

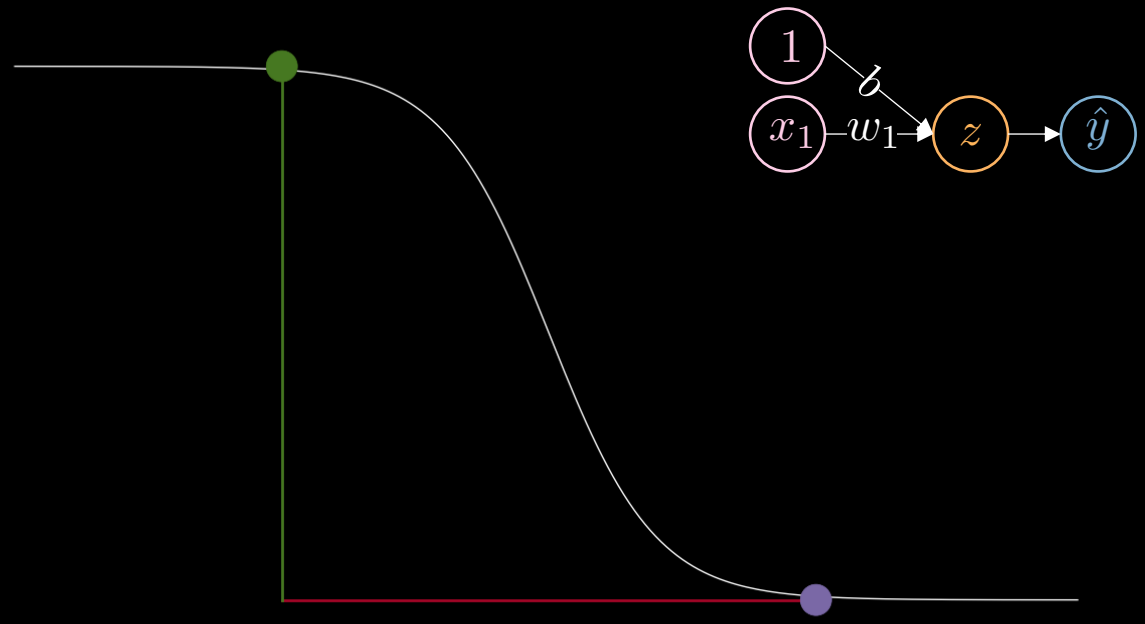




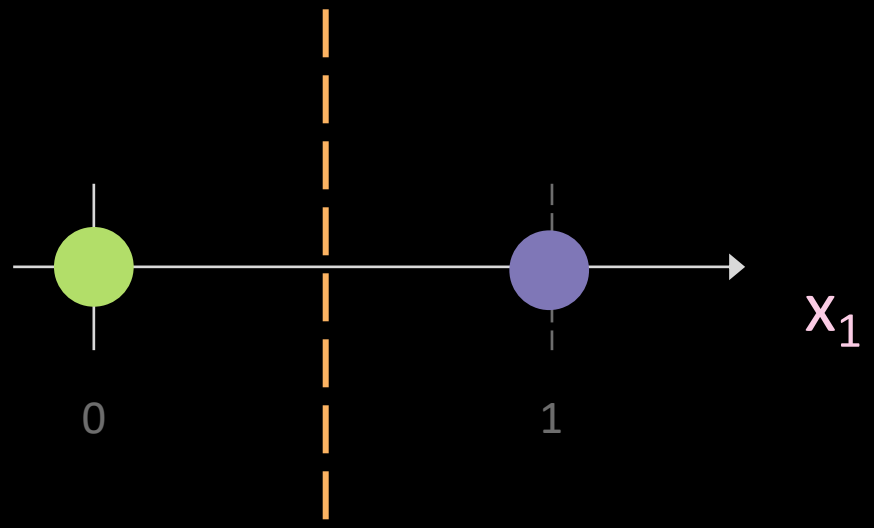


NOT neuron

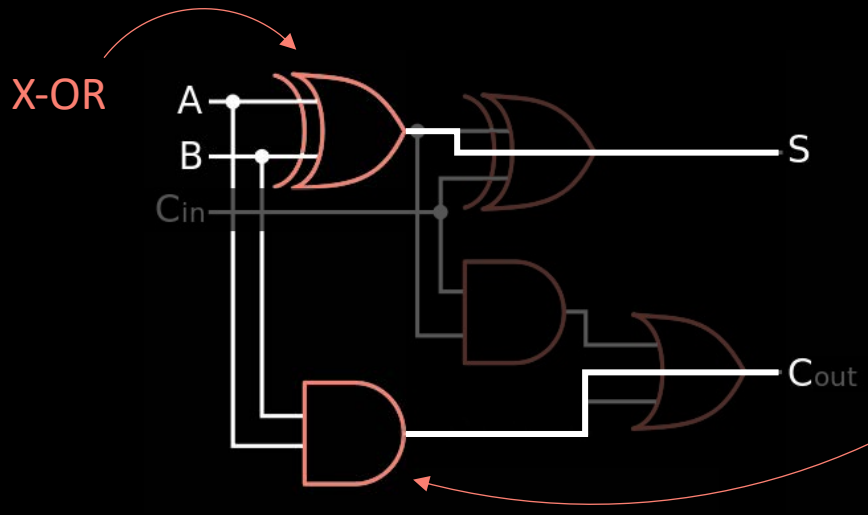




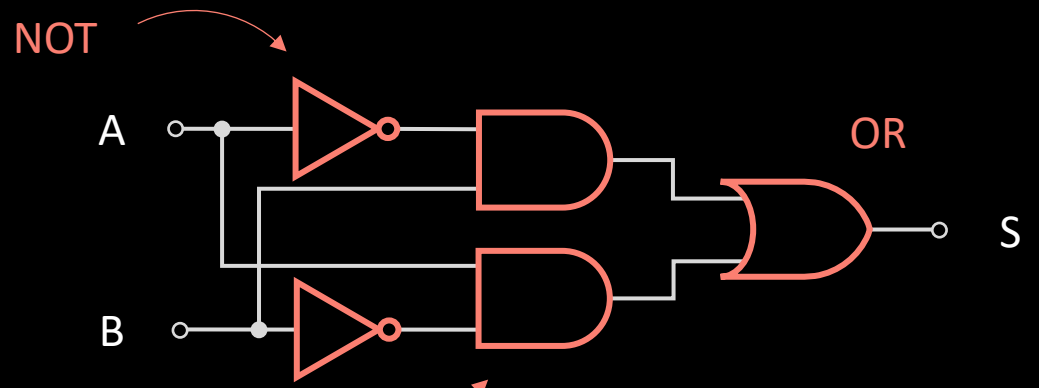
x_1	y	z
0	1	> +5
1	0	< –5



Half-Adder



X-OR



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