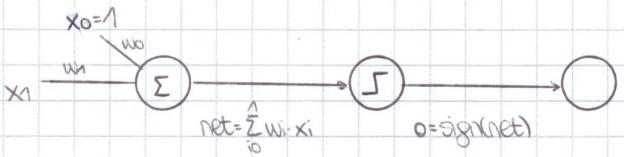
Lecture 07 Cristiano Panighel 2058351

- Give perceptron-based multi-layer networks with hard threshold and relative weights (without using learning) that implements simple Boolean functions such as A xor B.

NOT

I want to implement the formula $\Phi = \neg x_1$. x_1 can be 1 or 0.

x_1	$\neg x_1$
0	1
1	0



I assume that $sign(net) = \begin{cases} 1 & \text{if } net > 0 \\ 0 & \text{if } net \leq 0 \end{cases}$

Let's assign value 1 to w_0 and -1 to w_1, so I have $w_0 = 1$ and $w_1 = -1$. There are 2 cases:

- $x_1 = 1$ $\quad net = x_0 \cdot w_0 + x_1 \cdot w_1 = 1 \cdot (1) + 1 \cdot (-1) = 0$
  $sign(net) = sign(0) = 0$
- $x_1 = 0$ $\quad net = x_0 \cdot w_0 + x_1 \cdot w_1 = 1 \cdot (1) + 0 \cdot (-1) = 1$
  $sign(net) = sign(1) = 1$

So, setting $w_0 = 1$ and $w_1 = 1$, I have the correct weights and result for the NOT gate using only one perceptron.

XOR

I want to implement the formula $\Phi = x_1 \oplus x_2$

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

The xor operator can be expressed as $\Phi = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$.
x_1, x_2 can be 1 or 0.

This means that I need AND, NOT and OR gates to compose a XOR gate.