Computational Intelligence

Samaneh Hosseini

Isfahan University of Technology

Outline

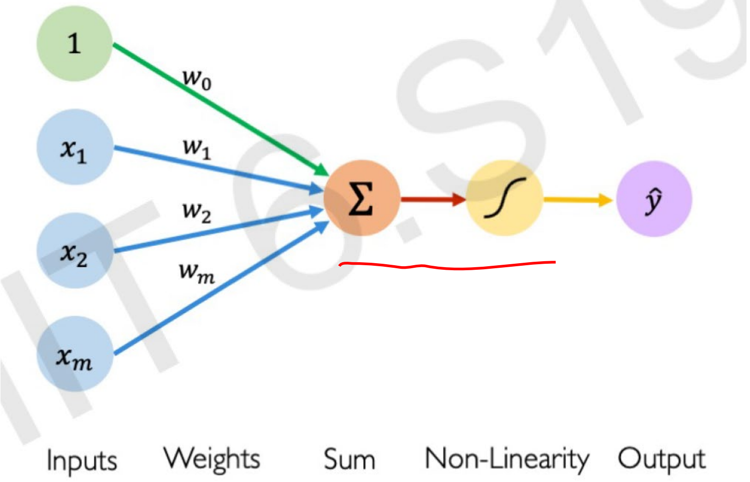
- Budling Neural Networks with Perceptron
- Representing Logical Functions Using Perceptron
- Limitations of Perceptron

Budling Neural Networks with Perceptron

The Perceptron

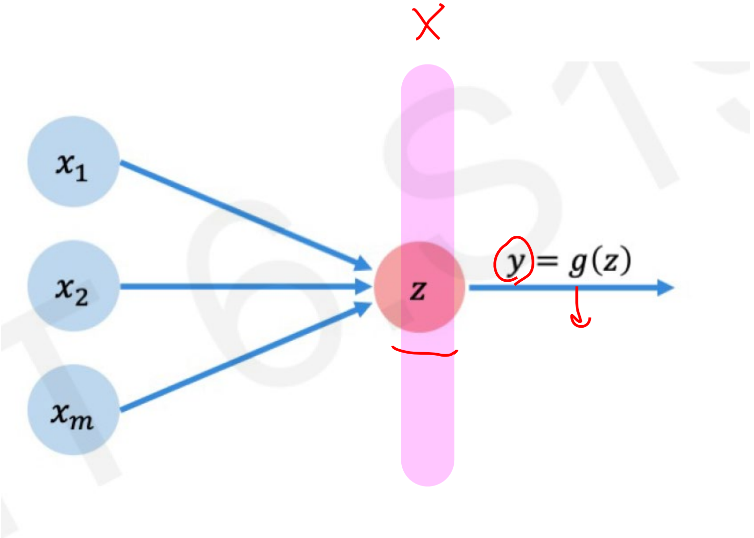
- Single-layer feed-forward neural network
- The inputs are connected directly to the outputs.
- Can represent logical functions, e.g. AND, OR, and NOT.

$$\hat{y} = g(\, w_0 + \boldsymbol{w}^T \boldsymbol{X} \,)$$



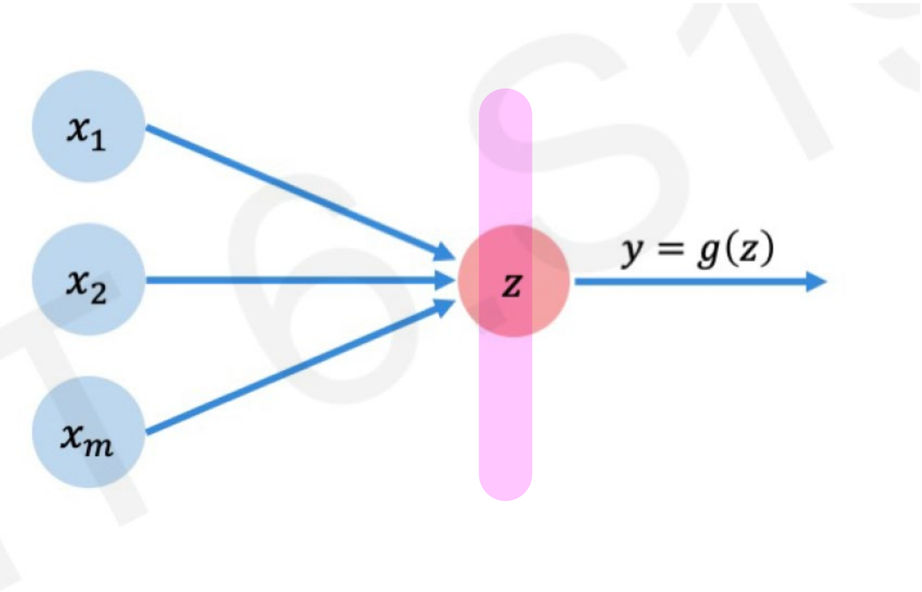
The Perceptron: Simplified

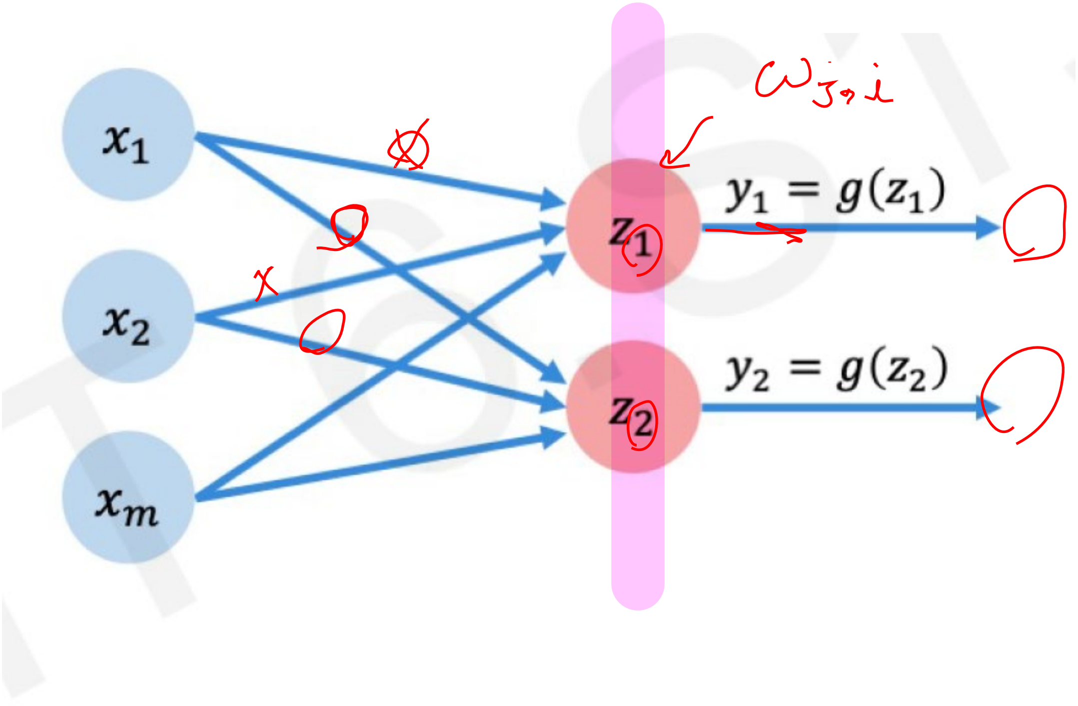
- Single-layer feed-forward neural network
- The inputs are connected directly to the outputs.
- Can represent logical functions, e.g. AND, OR, and NOT.



$$z = w_0 + \sum_{j=1}^{m} x_j w_j$$

Multi Output Perceptron





$$z = w_0 + \sum_{j=1}^{m} x_j\, w_j$$

$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j\, w_{j,i}$$

شماره خروجی

Representing Logical Functions Using Perceptron

CQ: What does the perceptron compute?

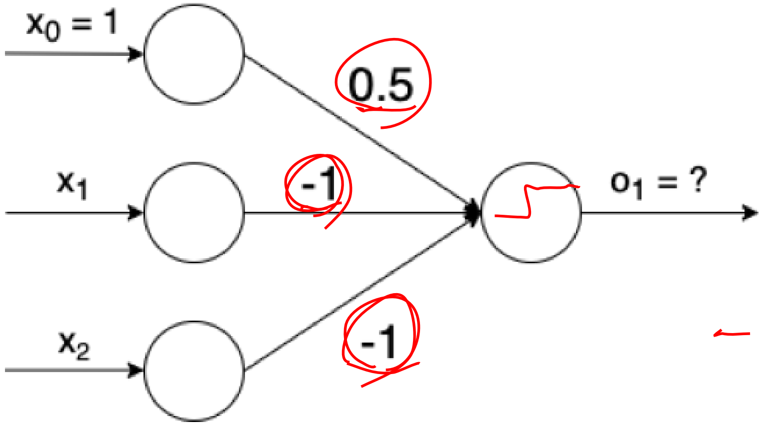
CQ: Consider the following perceptron, where the activation function is the step function. ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.). Which of the following logical function does the perceptron compute?

(A) $x_1 \wedge x_2$

(B) $\neg(x_1 \wedge x_2)$

(C) $x_1 \vee x_2$

(D) $\neg(x_1 \vee x_2)$



x_1	x_2	o_1
0	0	1
1	0	0
0	1	0
1	1	0

$-1 - 1 + 0.5 = -1.5$

$o_1 = 0$

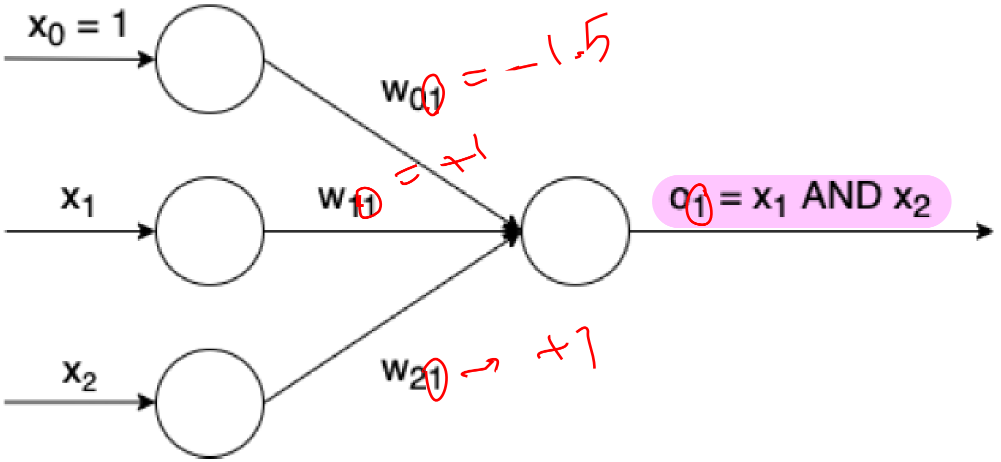
CQ: Learning a perceptron for the AND function

CQ: Consider the perceptron below where the activation function is the step function ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.). What should the weights w_{01}, w_{11} and w_{21} be such that the perceptron represents an AND function?

w_{ji}

اندیس ورودی

اندیس خروجی



x_1	x_2	o_1
0	0	0
0	1	0
1	0	0
1	1	1

$$y = -x + 1.5$$

$$x_2 = -x_1 + 1.5$$

$$x_2 + x_1 - 1.5 = 0$$

$$1 + 1 - 1.5 = 0.5$$

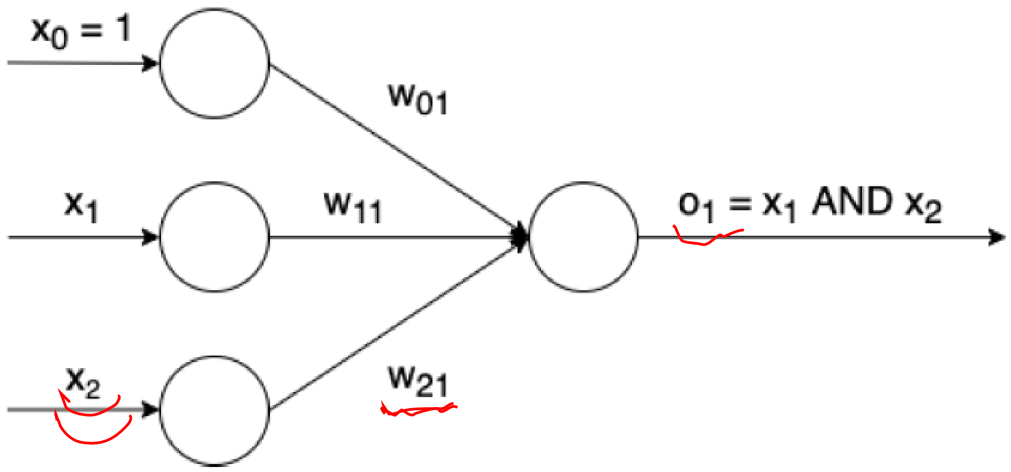
$$\downarrow$$

$$1$$

$$\underset{\downarrow \; 1 \; \downarrow \; w_{21}}{1} x_2 + \underset{\downarrow \; 1 \; \downarrow \; w_{11}}{1} x_1 \; \underbrace{-1.5}_{\downarrow \; -1.5 \; \downarrow \; w_{01}} > 0$$

CQ: Learning a perceptron for the AND function

CQ: Consider the perceptron below where the activation function is the step function ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.).
How do we learn the weights w_{01}, w_{11} and w_{21} such that the perceptron represents an AND function?



x_1	x_2	o_1
0	0	0
0	1	0
1	0	0
1	1	1

X_0	X_1	X_2	w_{01}	w_{11}	w_{12}	O_{actual}	O_{Expect}
1	1	1	0	0	0	0	1
1	1	0	0.2	0.2	0.2	1	0
1	0	1	0	0	0.2	1	0
1	1	1	-0.2	0	0	0	1
1	1	0	0	0.2	0.2	1	0
1	1	1	-0.2	0	0.2	0	1
1	0	1	0	0.2	0.4	1	0
			-0.2	0.2	0.2		

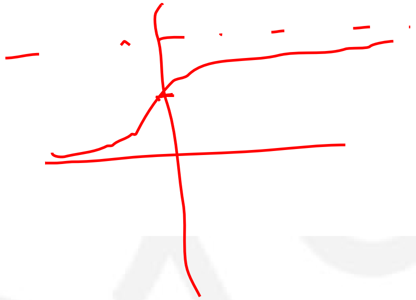
A perceptron representing OR

A question for you:

Consider a perceptron with three inputs x_0, x_1 and x_2 where the activation function is the step function ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.).

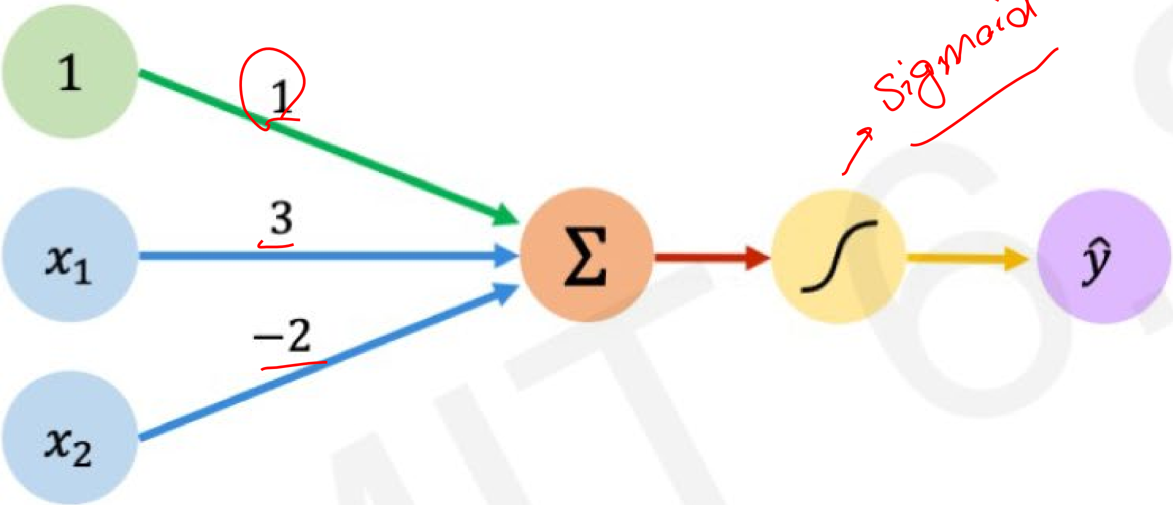
- What should the weights w_{01}, w_{11} and w_{21} be such that the perceptron represents an OR function?
- How do we learn these weights?

The Perceptron: Example



K=1

Sigmoid



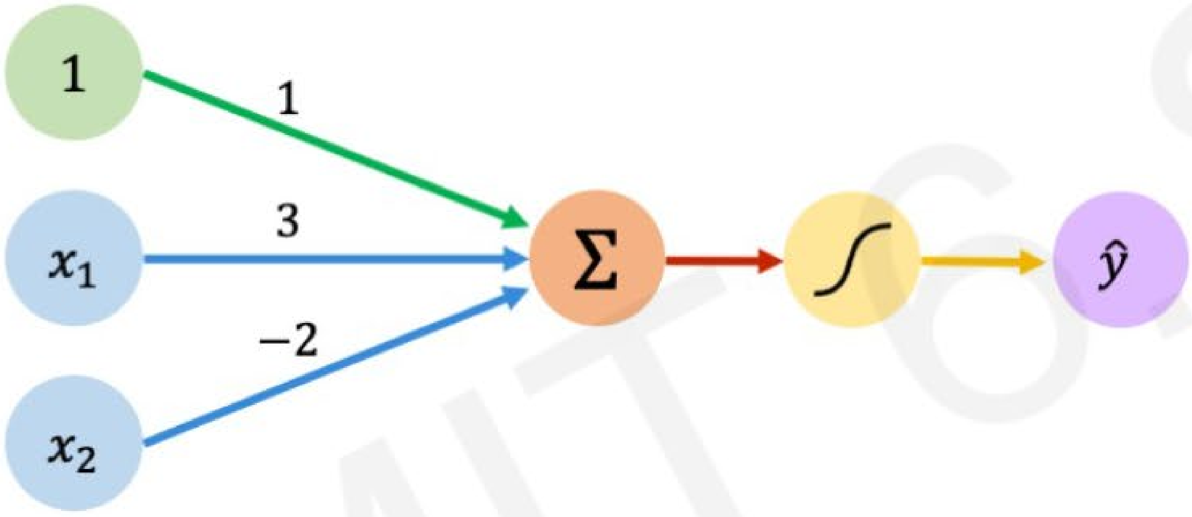
We have: $w_0 = 1$ and $\boldsymbol{W} = \begin{bmatrix}3\\-2\end{bmatrix}$

$$\hat{y} = g(w_0 + \boldsymbol{w}^T \boldsymbol{X})$$

$$= g\left(1 + \begin{bmatrix}3\\-2\end{bmatrix}^T \begin{bmatrix}x_1\\x_2\end{bmatrix}\right)$$

$$\hat{y} = g(\underbrace{1 + 3x_1 - 2x_2}_{\text{This is just a line in 2D!}})$$

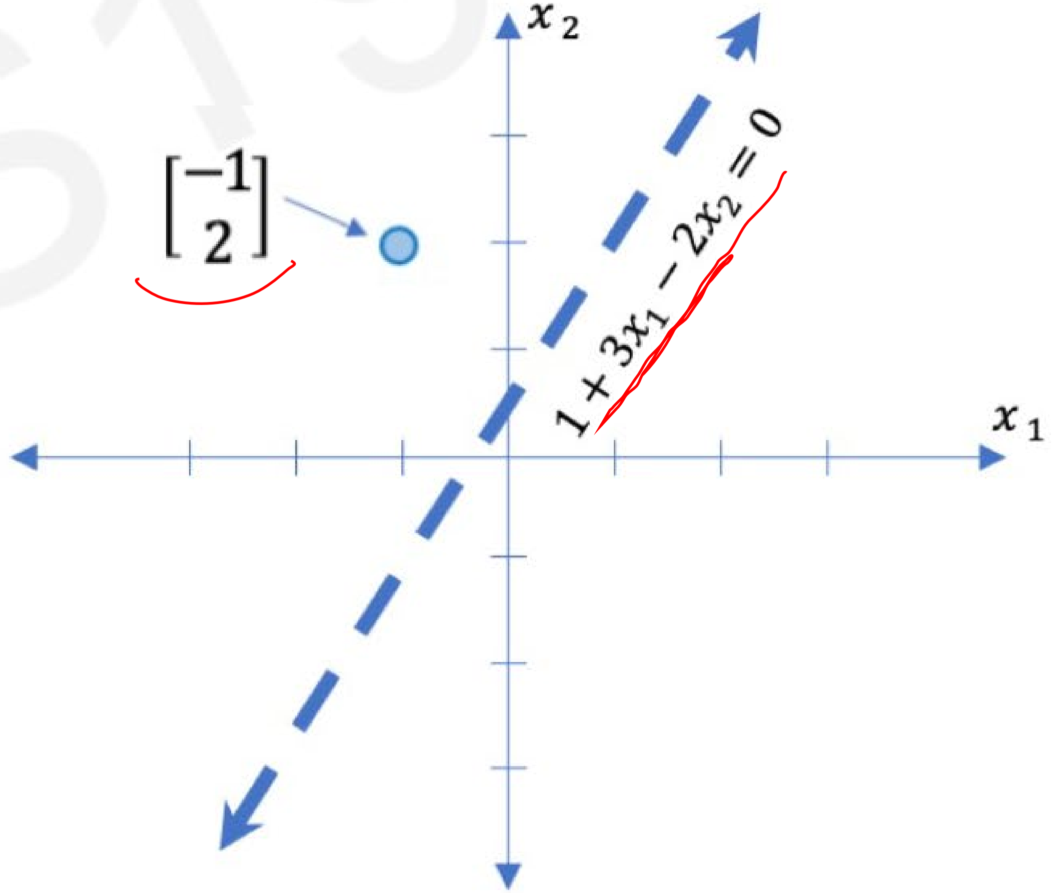
The Perceptron: Example



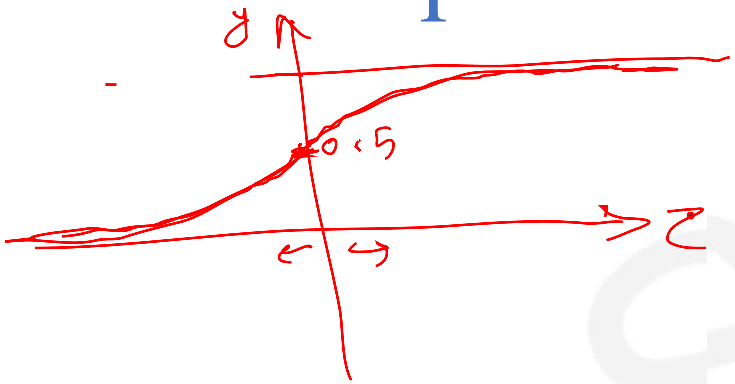
Assume we have input: $\boldsymbol{X} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$

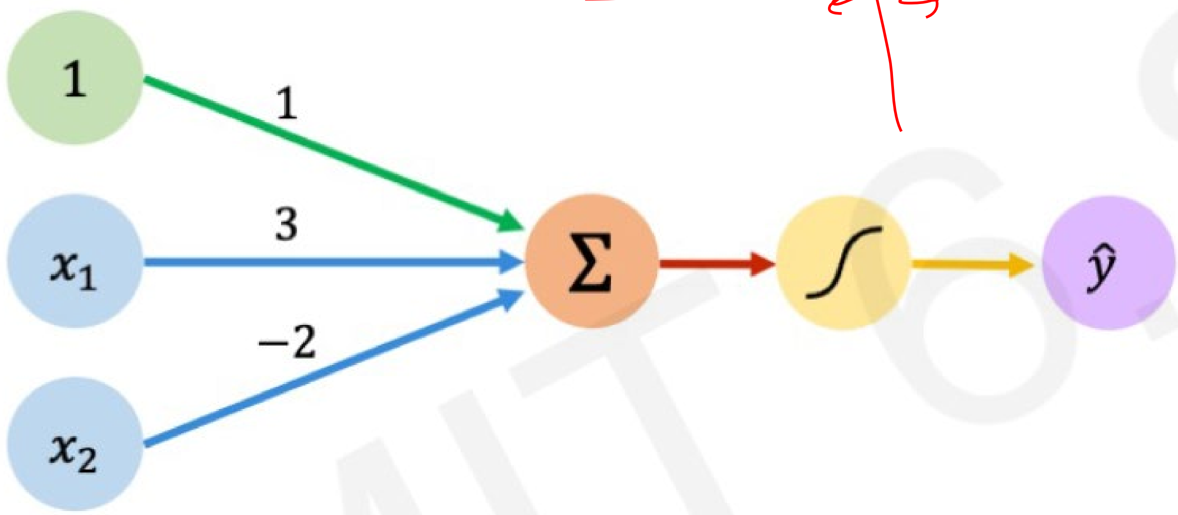
$$\hat{y} = g\big(1 + (3 * -1) - (2 * 2)\big)$$
$$= g(-6) \approx 0.002$$

$$\hat{y} = g(1 + 3x_1 - 2x_2)$$

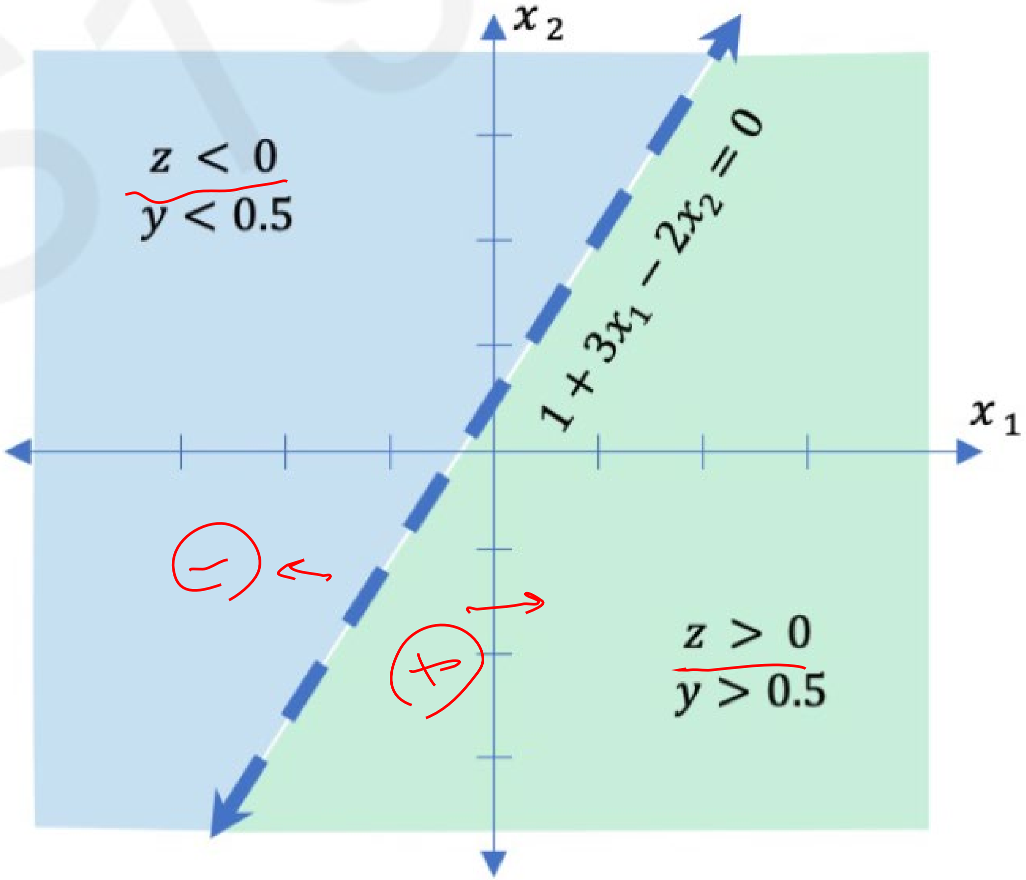


The Perceptron: Example





$$\hat{y} = g(1 + 3x_1 - 2x_2)$$



Limitations of perception

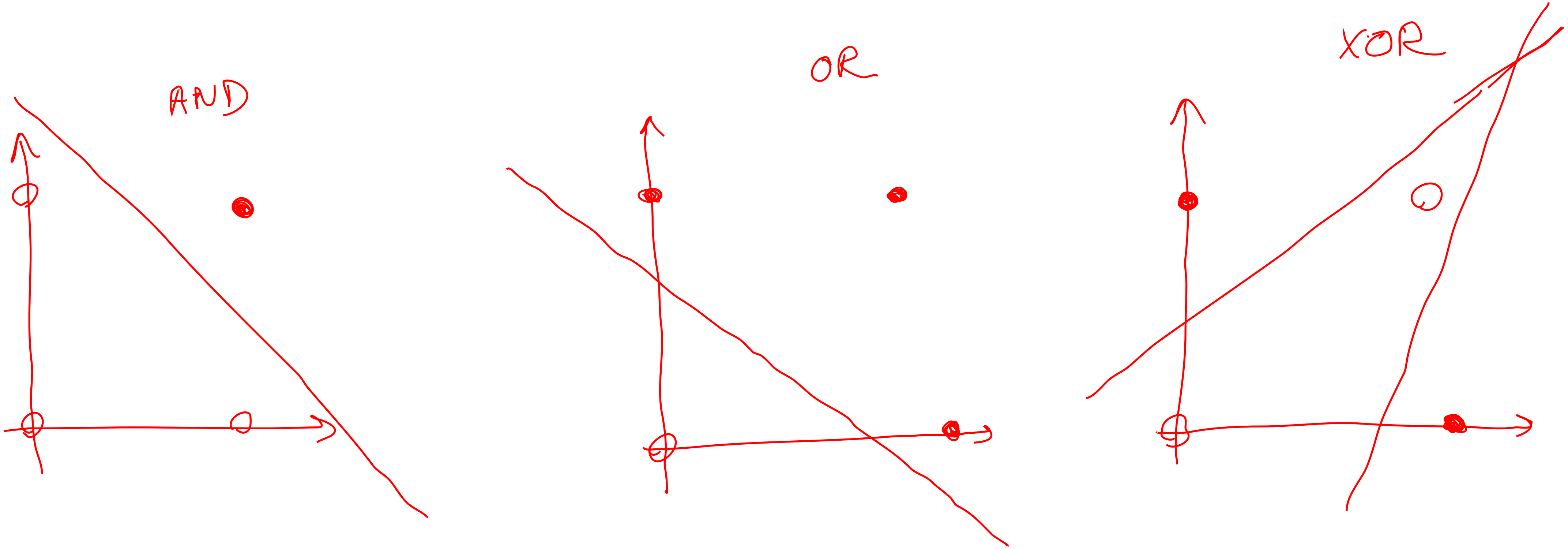
Limitations of perceptions

- Perceptrons: An introduction to computational geometry. Minsky and Papert. MIT Press. Cambridge MA 1969.
- Results:
  - XOR cannot be represented using perceptrons. We need a deeper network.
  - No one knew how to train deeper networks.
- Led to the first AI winter.

CQ: Why can't a perceptron represent XOR?

a perceptron is a linear classifier

XoR is not linearly separable

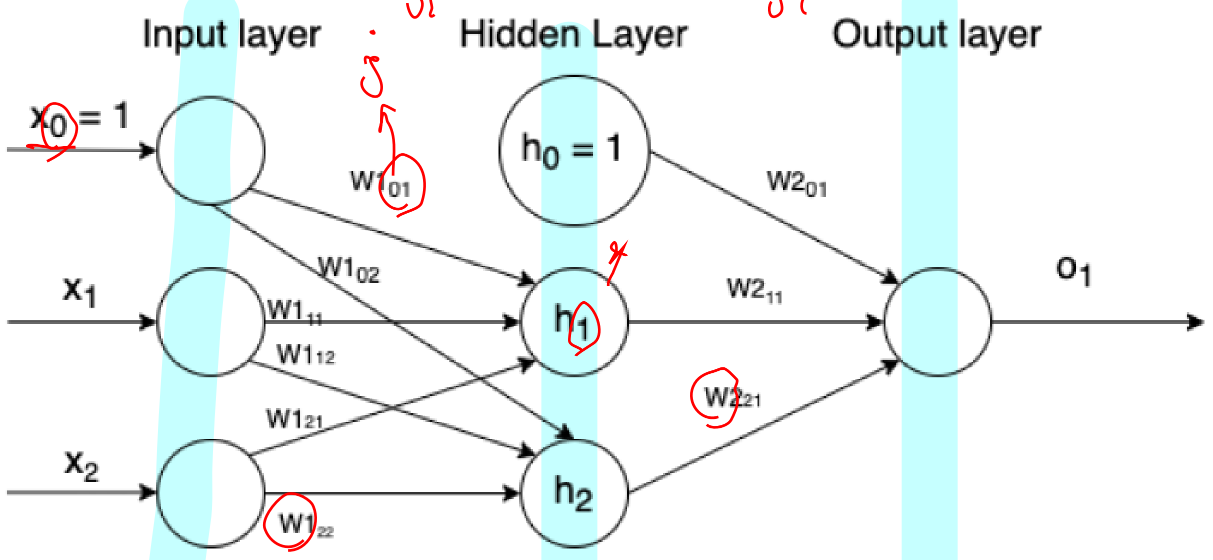


x

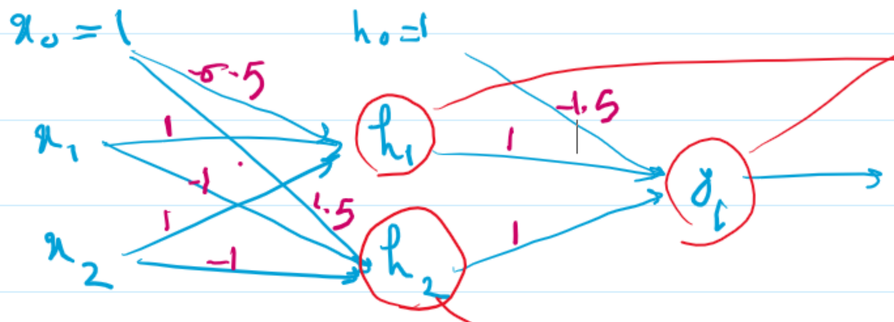
XOR as a 3-Layer Neural Network

Can you come up with the weights such that
the following network represents the XOR function?

$w^{[1]}_{ji}$ $w^{[2]}_{ji}$



$XOR = x_1 \; XOR \; x_2 \equiv (x_1 \vee x_2) \wedge \overline{(x_1 \wedge x_2)}$



$$y_1 = [(x_1 \vee x_2) \wedge (\neg (x_1 \wedge x_2))]$$

$$= (x_1 \text{ XOR } x_2)$$

$$h_1 = g(x_1 + x_2 - 0.5)$$

x_1	x_2	h_1
0	0	0
0	1	1
1	0	1
1	1	1

$$h_1 = (x_1 \vee x_2)$$

$$h_2 = g(-x_1 - x_2 + 1.5)$$

x_1	x_2	h_2
0	0	1
0	1	1
1	0	1
1	1	0

$$h_2 = \neg (x_1 \wedge x_2)$$

$$y_1 = g(h_1 + h_2 - 1.5)$$

h_1	h_2	y_1
0	0	0
0	1	0
1	0	0
1	1	1

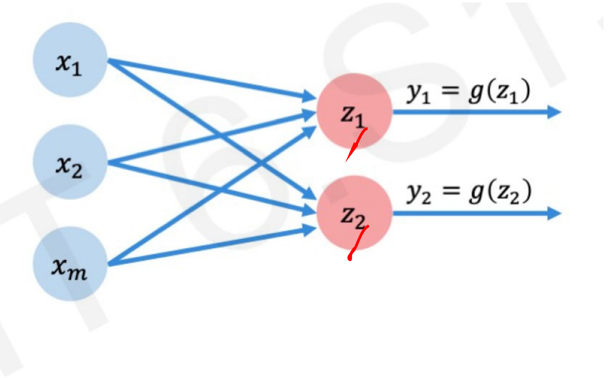
$$y_1 = h_1 \wedge h_2$$

Core Foundation Review

The Perceptron

Calculating logical functions

Limitations of Perceptron



$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j \, w_{j,i}$$