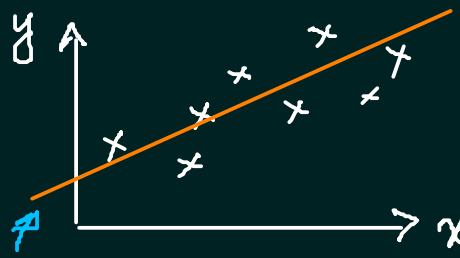
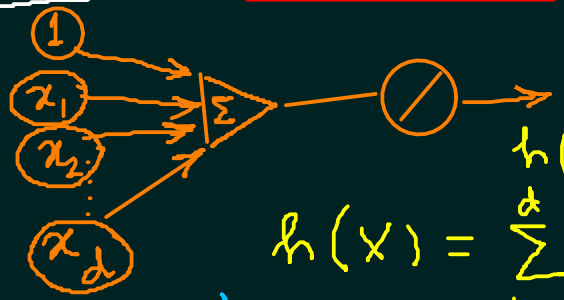
SUMMARY

Linear Regression:



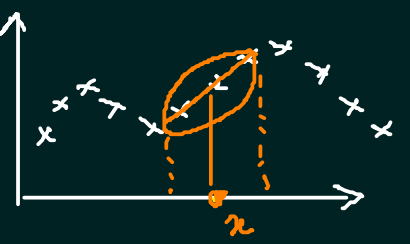


$$h(x) = \sum_{i=0}^{d} w_i x_i$$

$$E_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \sum_{i=0}^{d} w_i x_{ni} \right)^2$$

Minimize $E_{in}(w) \Rightarrow$ $\boxed{w = (X^T X)^{-1} X^T y}$ ← One step learning ← Optimization Problem (Sum Sqr. error)

(Extn.) Locally weighted Regression

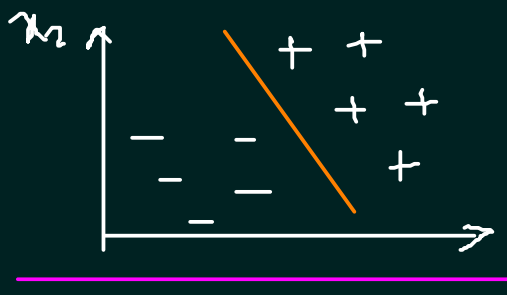


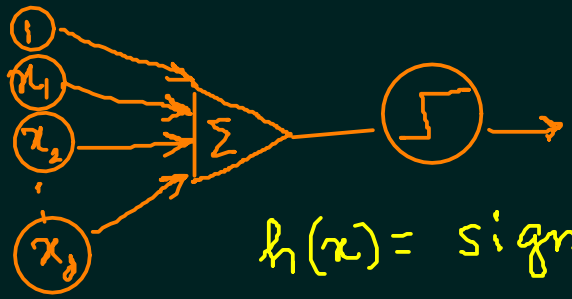
$$h(x) = \sum_{i=0}^{d} \alpha_i w_i x_i$$

where, $\alpha_i = e^{-\frac{(x_i - x)^2}{2\tau^2}}$

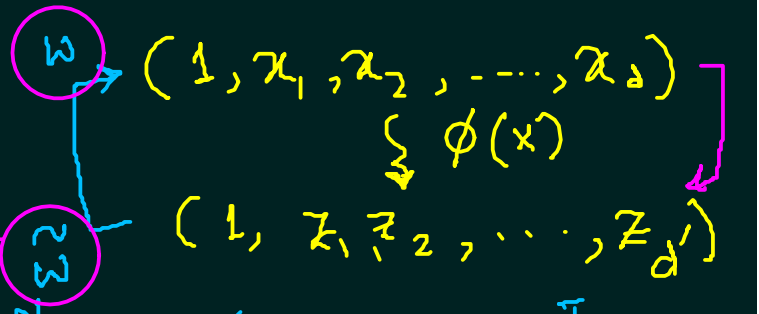
Linear classification:

$f : X \to Y = \{-1, +1\}$ (Extn.) Non-linear Transform

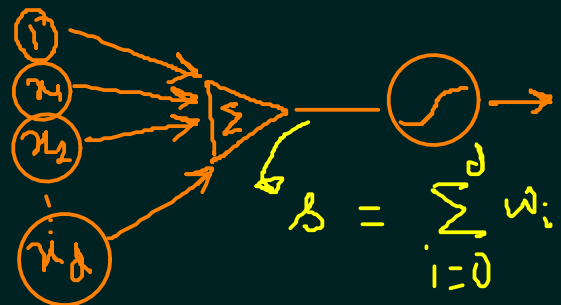




$$h(x) = \text{sign}\left(\sum_{i=0}^{d} w_i x_i \right)$$



Logistic Regression:



$$s = \sum_{i=0}^{d} w_i x_i$$

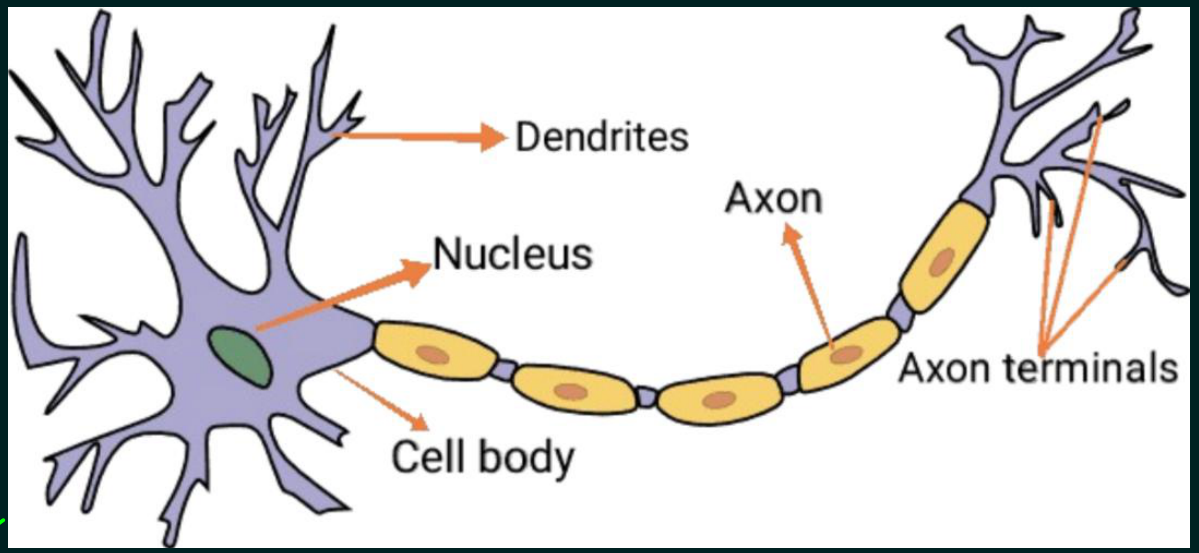
$$h(x) = \theta(s) = \frac{1}{1 + e^{-s}}$$ [Sigmoid]

$$E_{in}(w) = \frac{1}{N} \sum_{i=1}^{N} \ln\left(1 + e^{-y_n w^T x_n}\right)$$

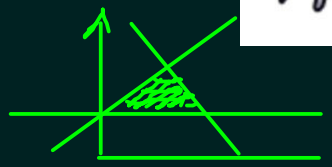
[Cross-entropy error] $e(h(x_n), y_n)$

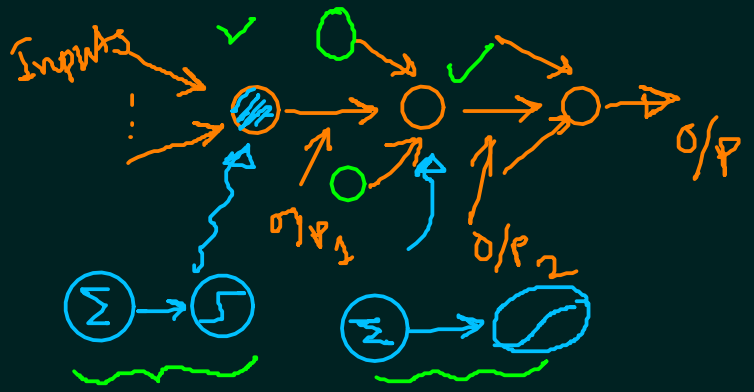
⇒ Gradient Descent → Batch / One step; → Stochastic / Incremental

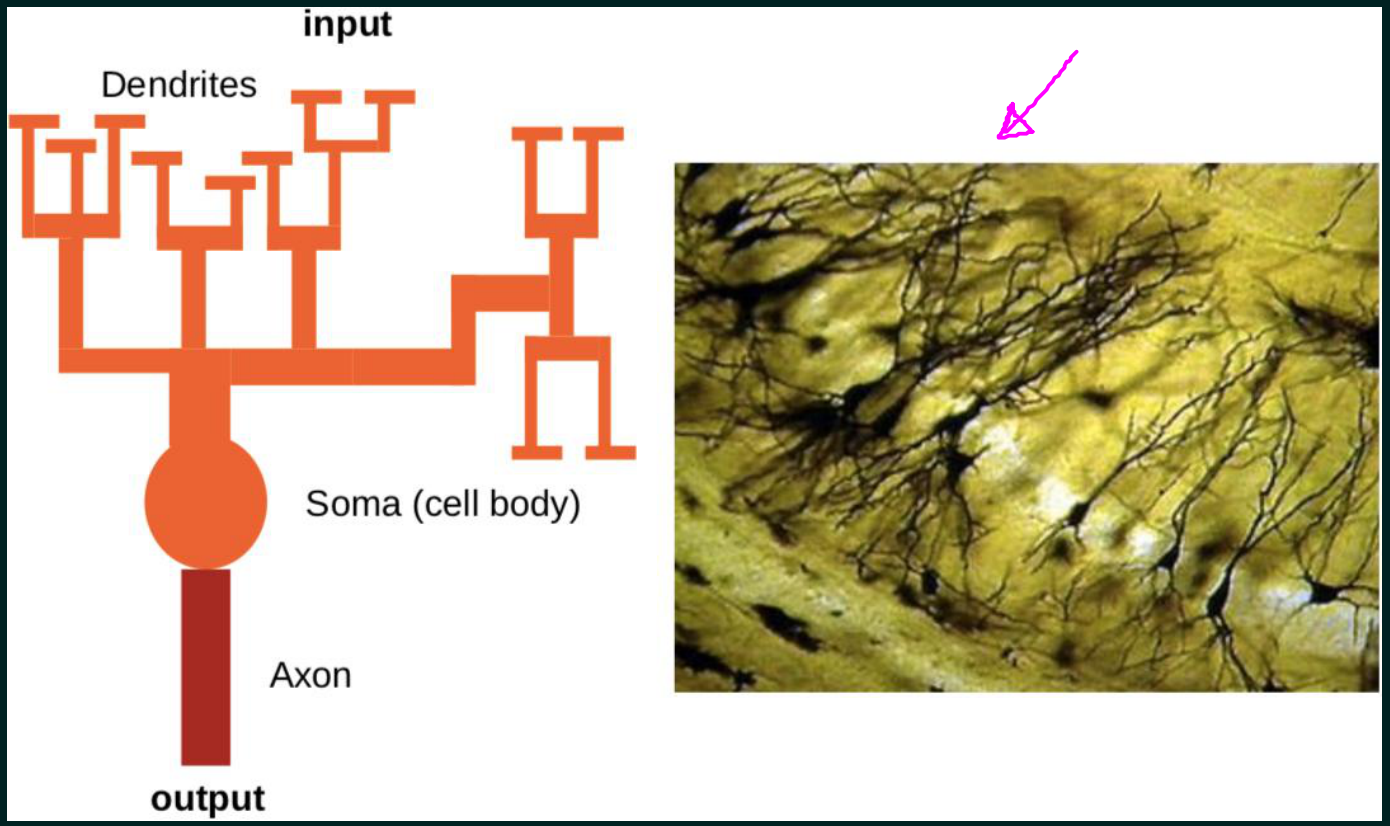
- Draws the analogy from Biology
  - 10^{11} neurons
  - → each neuron → 10^4 others (Connected)



- Recognize
  - → 10^{-1} sec
- → 10^{-3} switching time
- ⩾ 100 neurons (used)







▷ Linear classification : Perceptron (linear discriminator)

$f : X \to Y \in \{+1, -1\}$

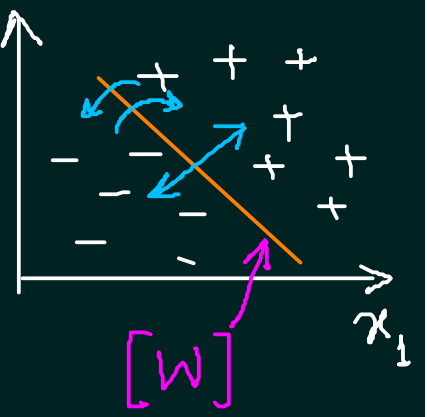
$s = \sum_{i=0}^{d} w_i x_i \quad (x_0 = 1)$

x → $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ TEs ($t_1, t_2, \ldots$)

Goal: $h(x) = \underline{\text{sign}}\left(\sum_{i=0}^{d} w_i x_i\right)$

$x^{new} = [x_1^{new} \; \cdots \; x_d^{new}]$

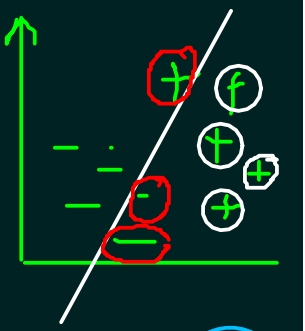
$y^{new} = + \; / \; -$



← classify

(x_n, y_n) → $[t_n]$ Min

$E_{in}(w) = \frac{1}{2} \sum_{i=1}^{N} \left[y_n - \text{sign}\left(\sum_{i=0}^{d} w_i x_{ni}\right) \right]^2$



$0 = \nabla E_{in}(w) = \left[\frac{\partial E_{in}}{\partial w_0}, \frac{\partial E_{in}}{\partial w_1}, \ldots, \frac{\partial E_{in}}{\partial w_d} \right]$

$\frac{\partial E_{in}}{\partial w_k} = \sum_{n=1}^{N} \left(y_n - \text{sign}\left(\sum_{i=0}^{d} w_i x_{ni}\right) \right) \left(-\text{sign}'(x_{nk}) \right)$

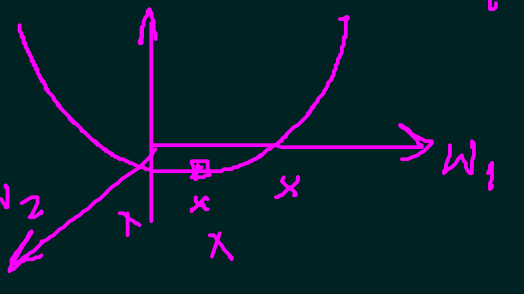
(t_n, o_n, $-x_{nk}$)

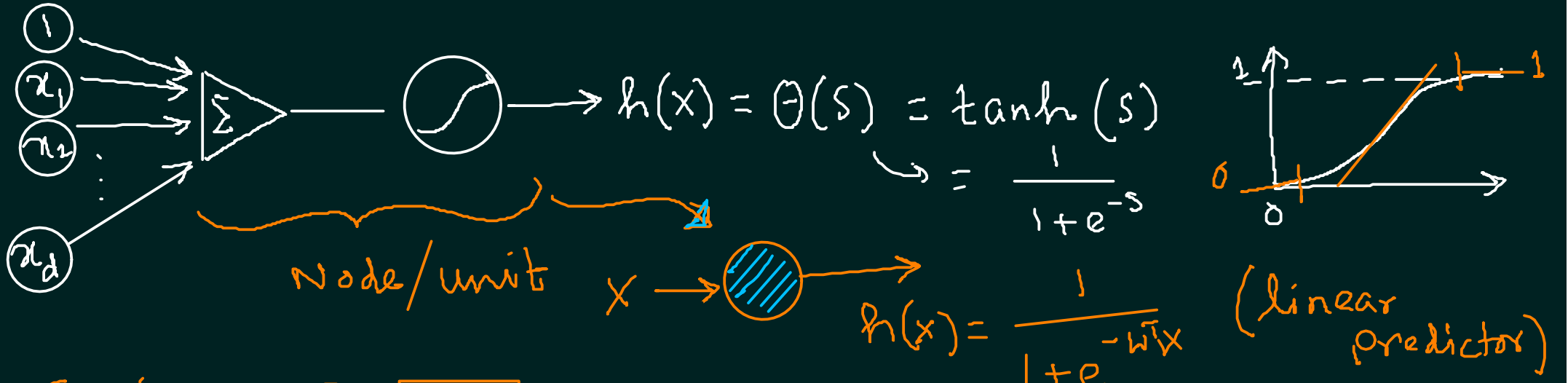
(Batch) G.D.

$\left[w_k \leftarrow w_k + \eta \sum_{n=1}^{N} (t_n - o_n)(x_{nk}) \right]$ (Delta)

$\left[w_i \leftarrow w_i + \Delta w_i \right]$

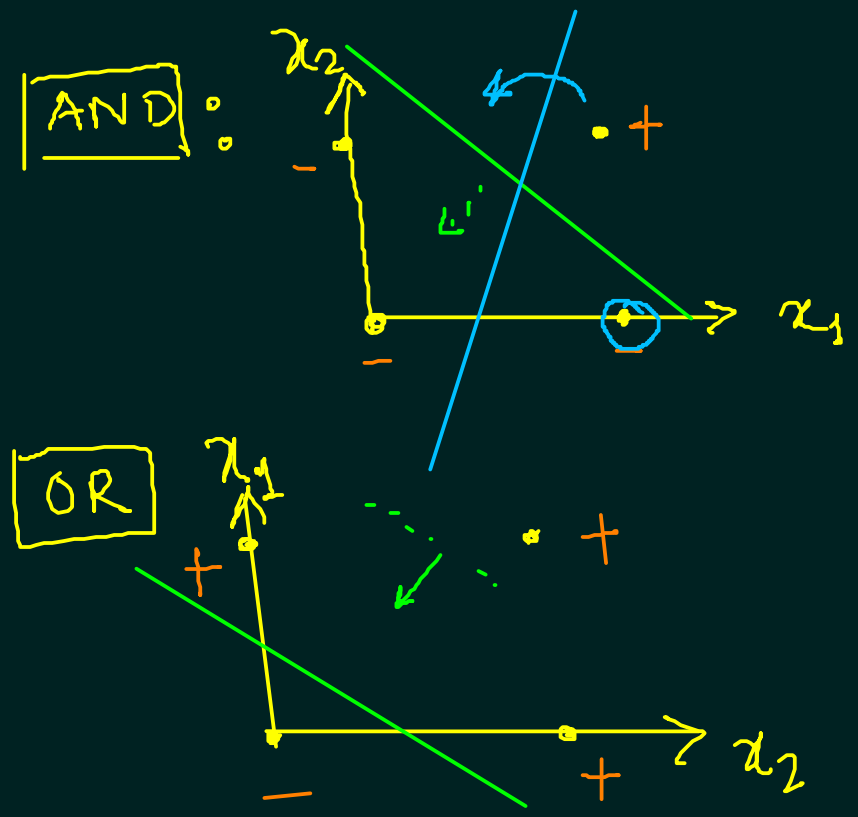
$\Delta w_i = -\eta \frac{\partial E_{in}}{\partial w_i}$

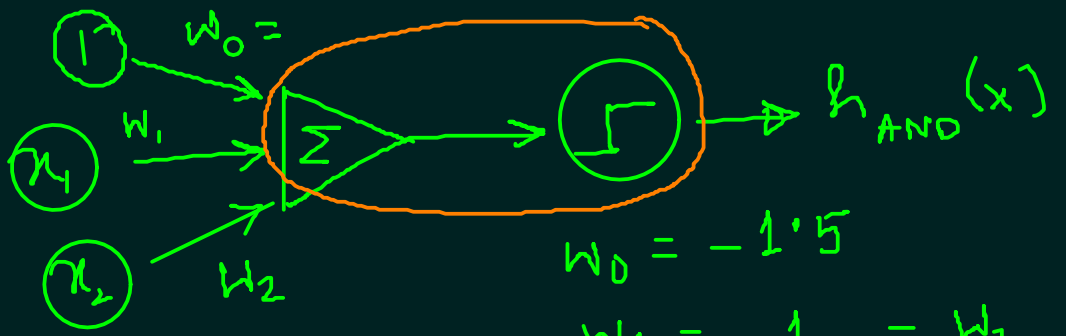




$\nabla E_{in}(w) = ?$ H/W

(Artificial Neural Node)





$w_0 = -1.5$

$w_1 = 1 = w_2$

$h_1 = (w_0 + w_1 x_1 + w_2 x_2) \gtrless \begin{matrix} + \\ - \end{matrix}$

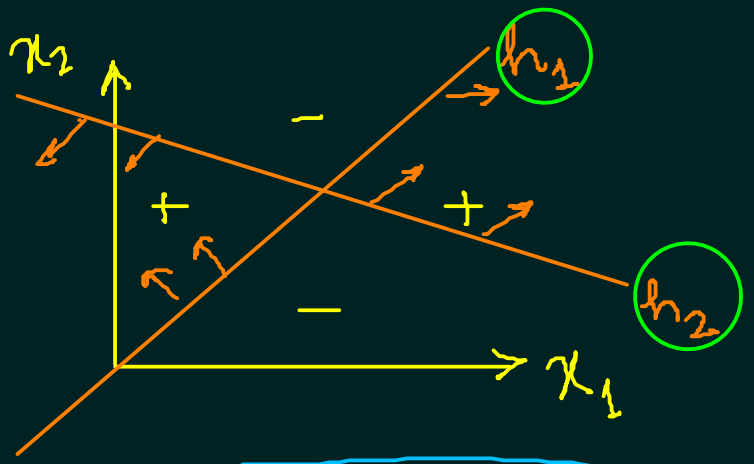
$h_2 = (w_0 + w_1 x_1 + w_2 x_2)$

$w_0 = -0.5$

$w_1 = 1 = w_2$

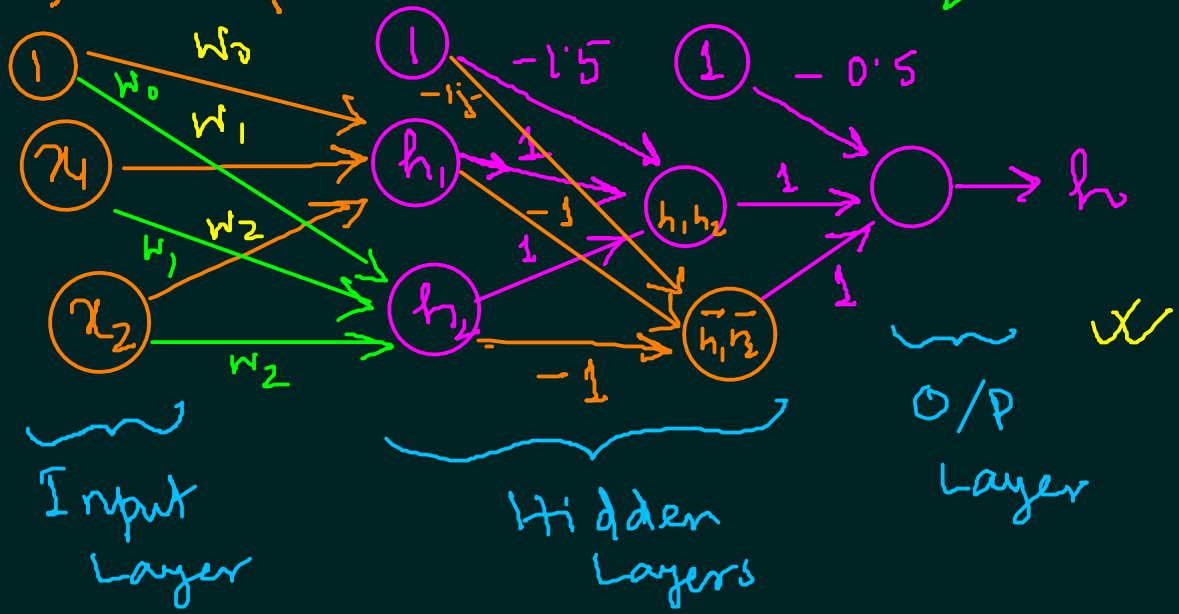
$+$ $-$

$w_i \leftarrow w_i - \eta \frac{\partial E_{in}}{\partial w_i}$



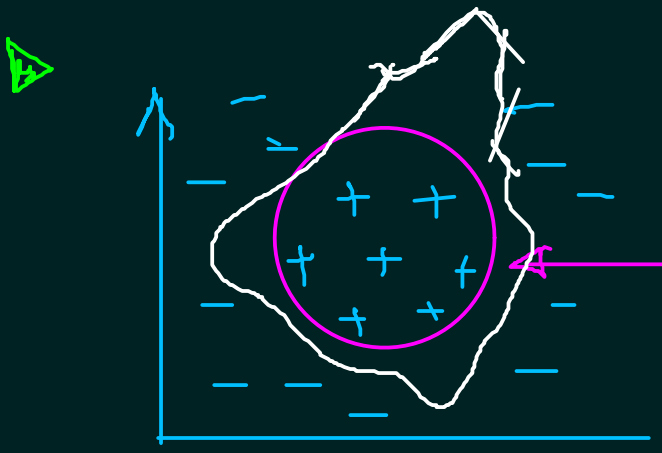
$$h = \overline{h_1 \cdot h_2} + \overline{h_1 \cdot h_2} \quad \text{OR}$$

(+ sign, − sign)



MULTI-LAYER PERCEPTRON

⇒ Neural Network (feed-forward NN)



Question: How to categorize +'s from −'s ??

(Think!)

NN →