Percettrone, back-propagation, MLP

- Cenni
- Principali concetti base
- Poca Matematica (a volte non formalmente corretta)
- Complicato, non complesso

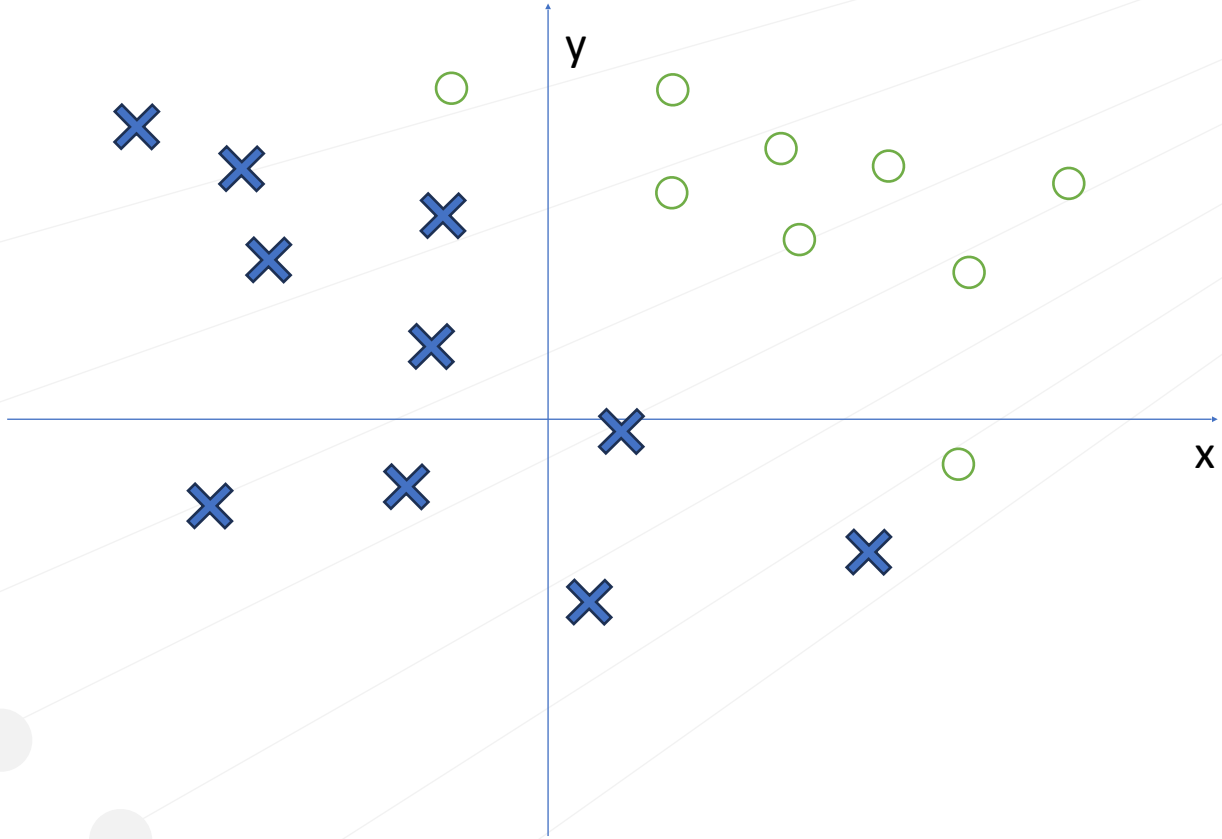
Uomo o Donna?

Un problema
di classificazione
(binaria)

Uomo o Donna?

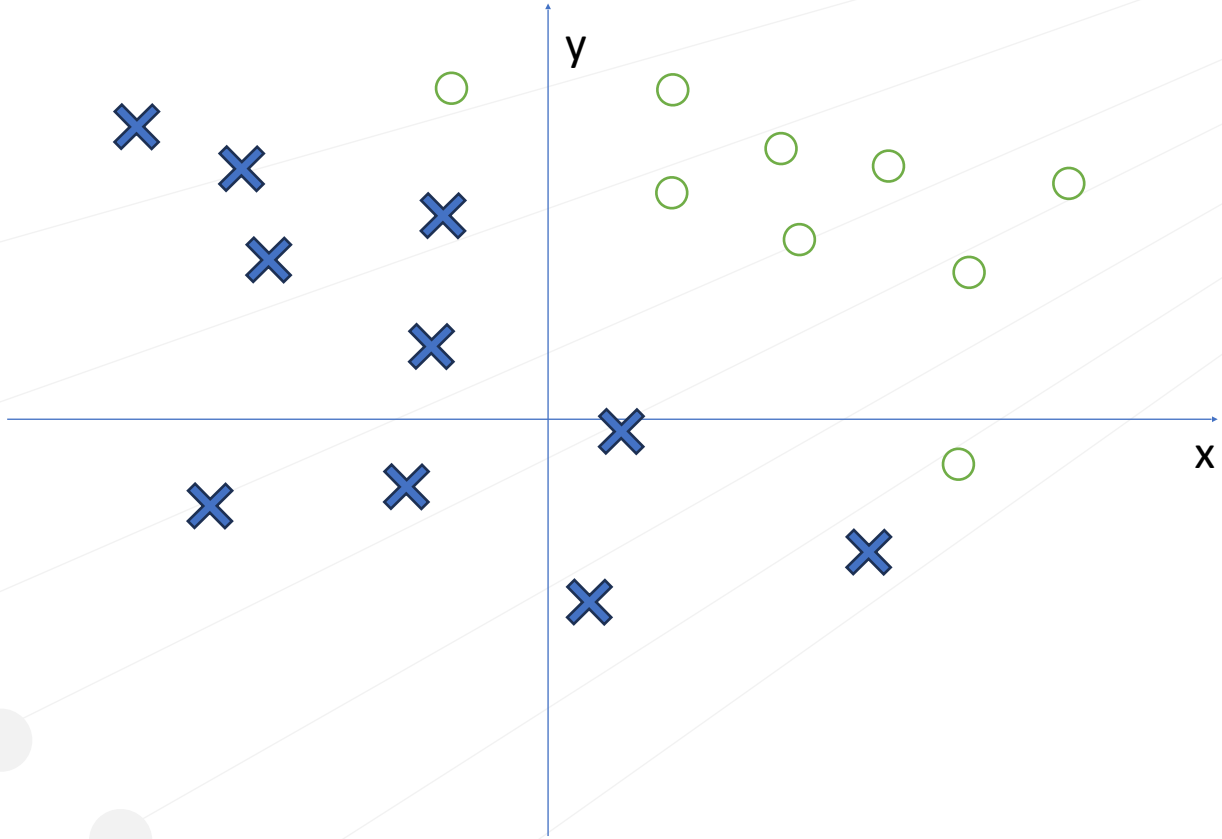
Un problema semplice di classificazione

Sei cerchio o sei croce ?



Un problema semplice di classificazione

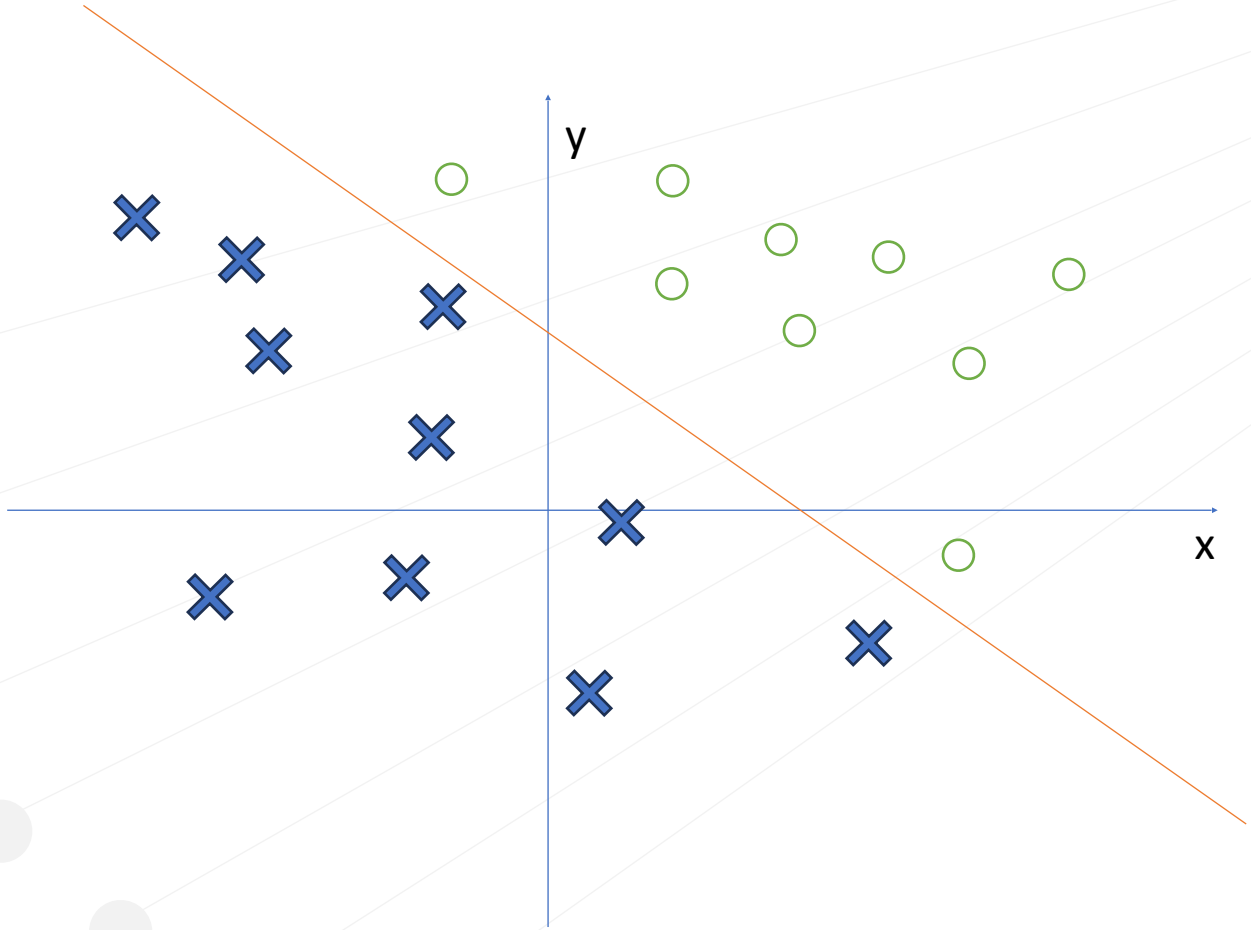
Sei cerchio o sei croce ?



Sei americano o messicano ?

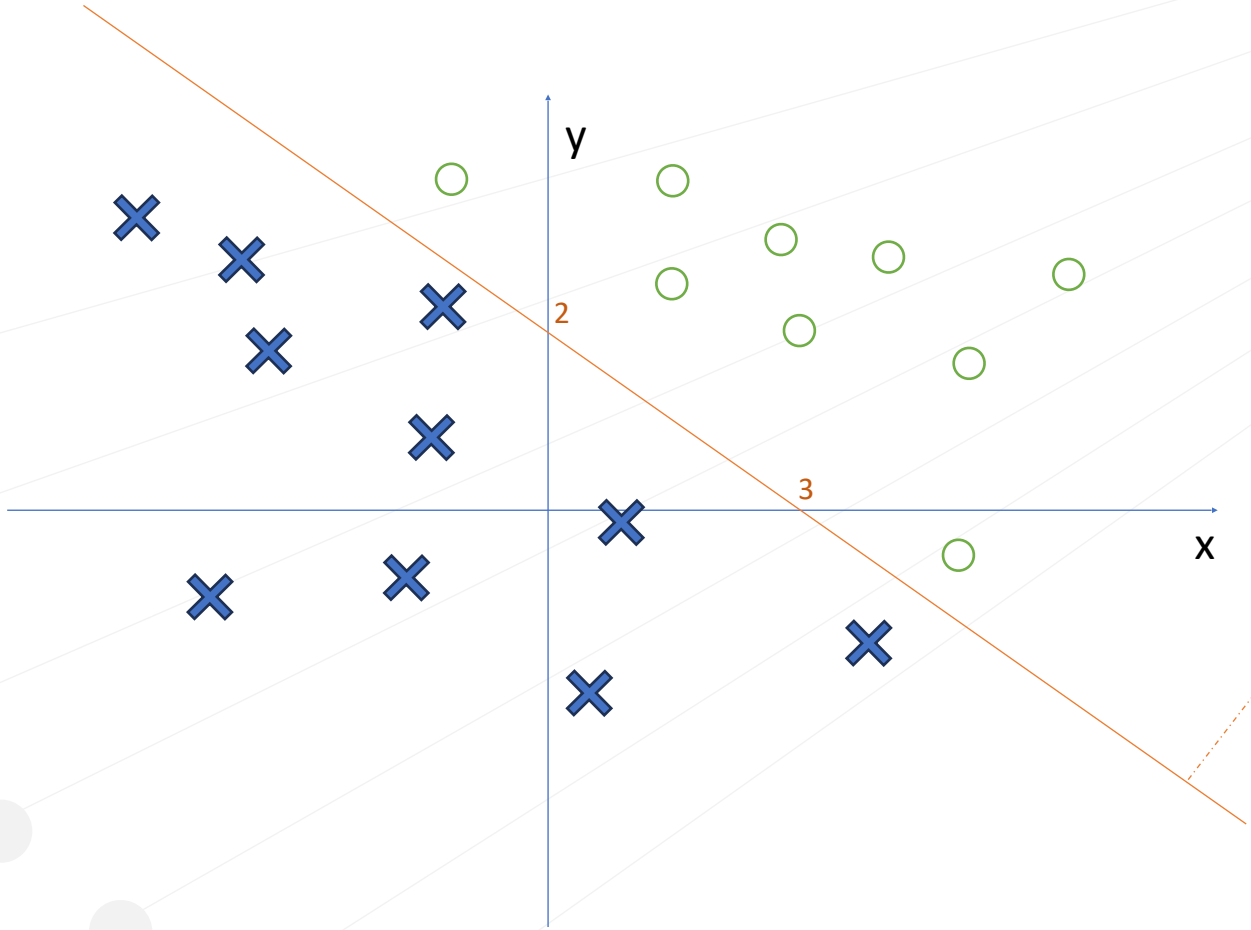
Un problema semplice di classificazione

Sei cerchio o sei croce ? La regola più semplice: tiro un riga…



Un problema semplice di classificazione

Sei cerchio o sei croce ? La soluzione più semplice: tiro un riga…



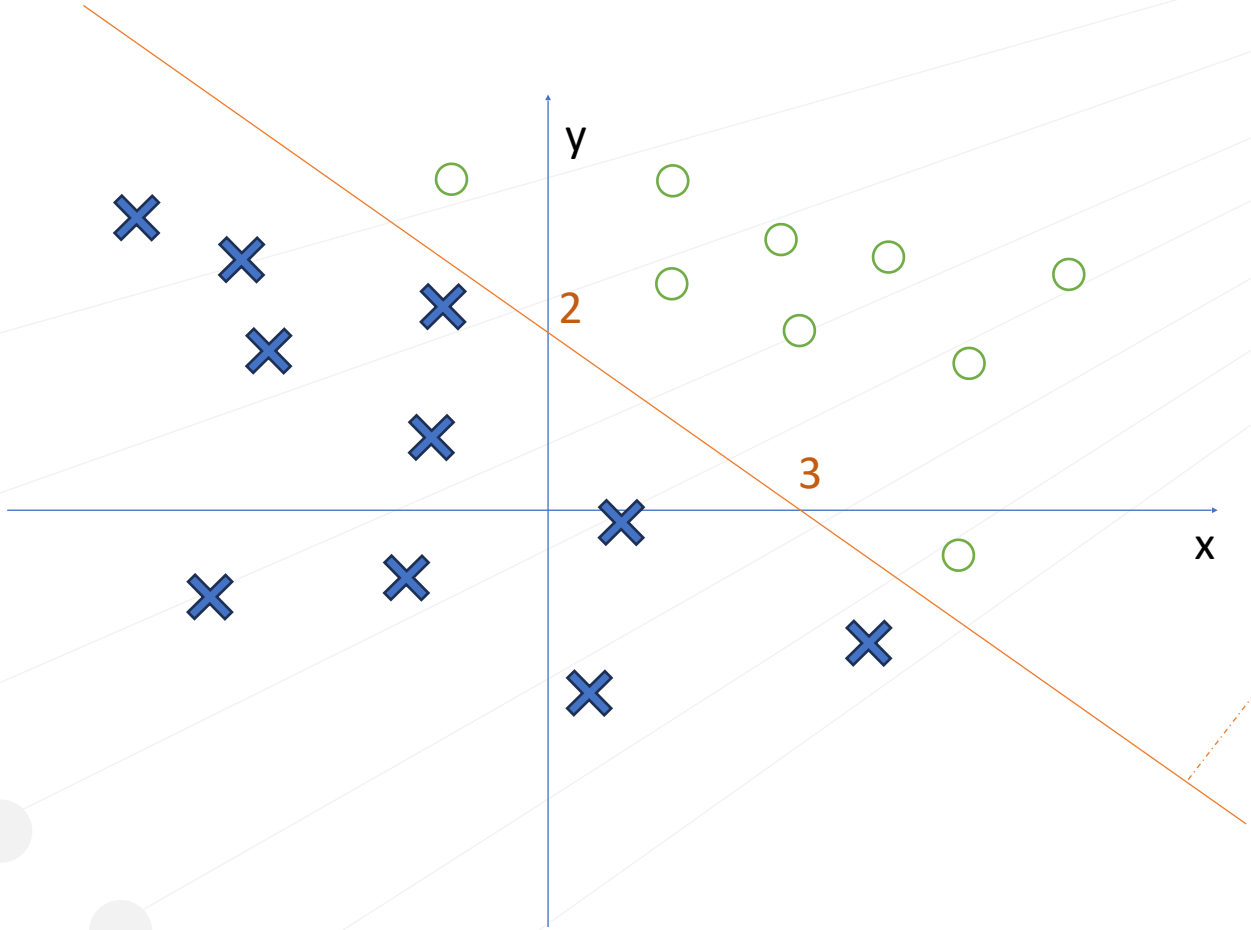
$2*x + 3*y = 6$ $\{ y = -\frac{2}{3}x + 2\}$

$2*x + 3*y > 6$ ⟶ ○

$2*x + 3*y < 6$ ⟶ ✕

Un problema semplice di classificazione

Sei cerchio o sei croce ? La soluzione più semplice: tiro un riga…

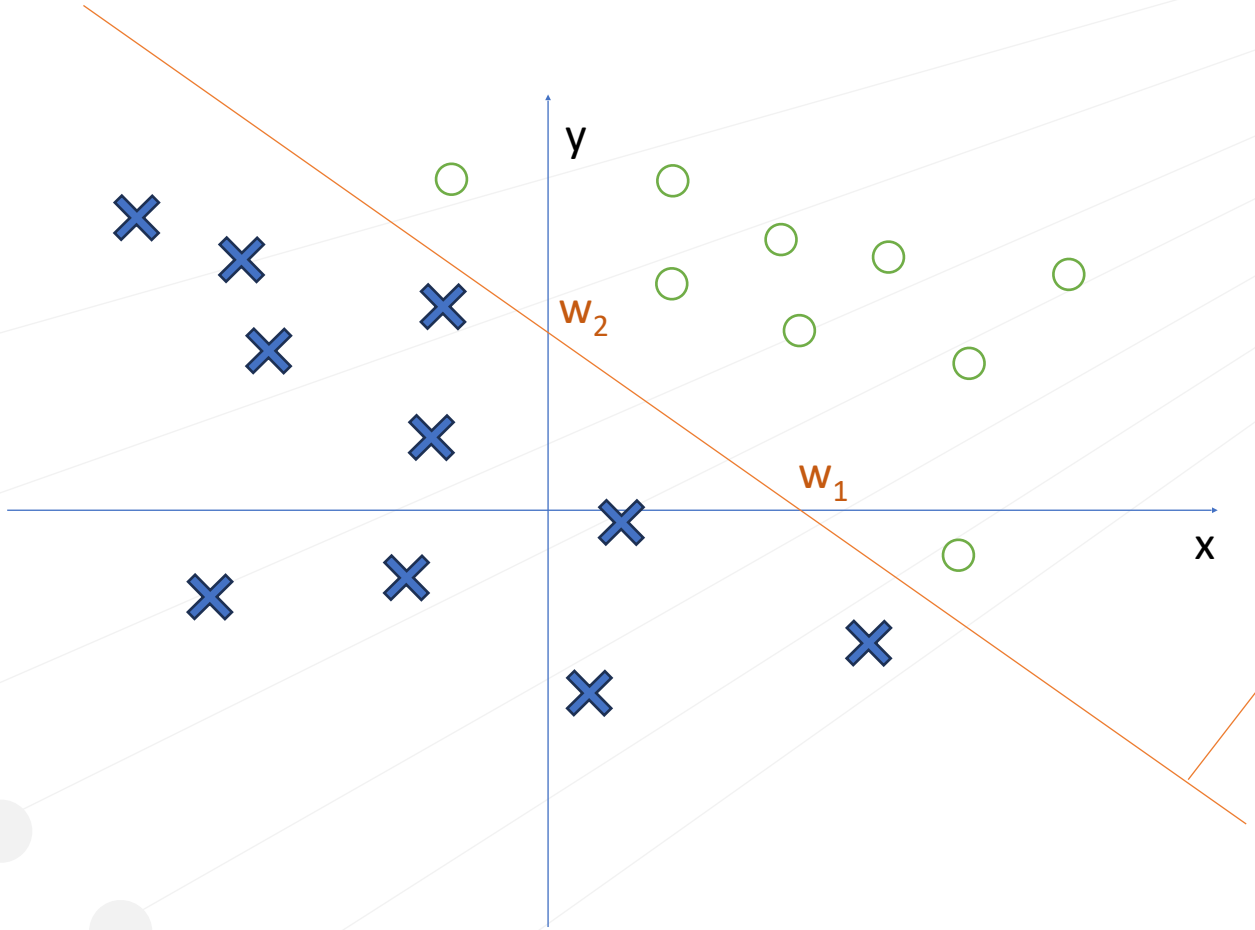


$2*x + 3*y = 6$

x, y	decisione
2, 2	○
5, 6	○
-1, 5	○
-2, 0	✕
-3, -1	✕
-2, 6	○
-2, 2	✕

Un problema semplice di classificazione

Generalizziamo un po’



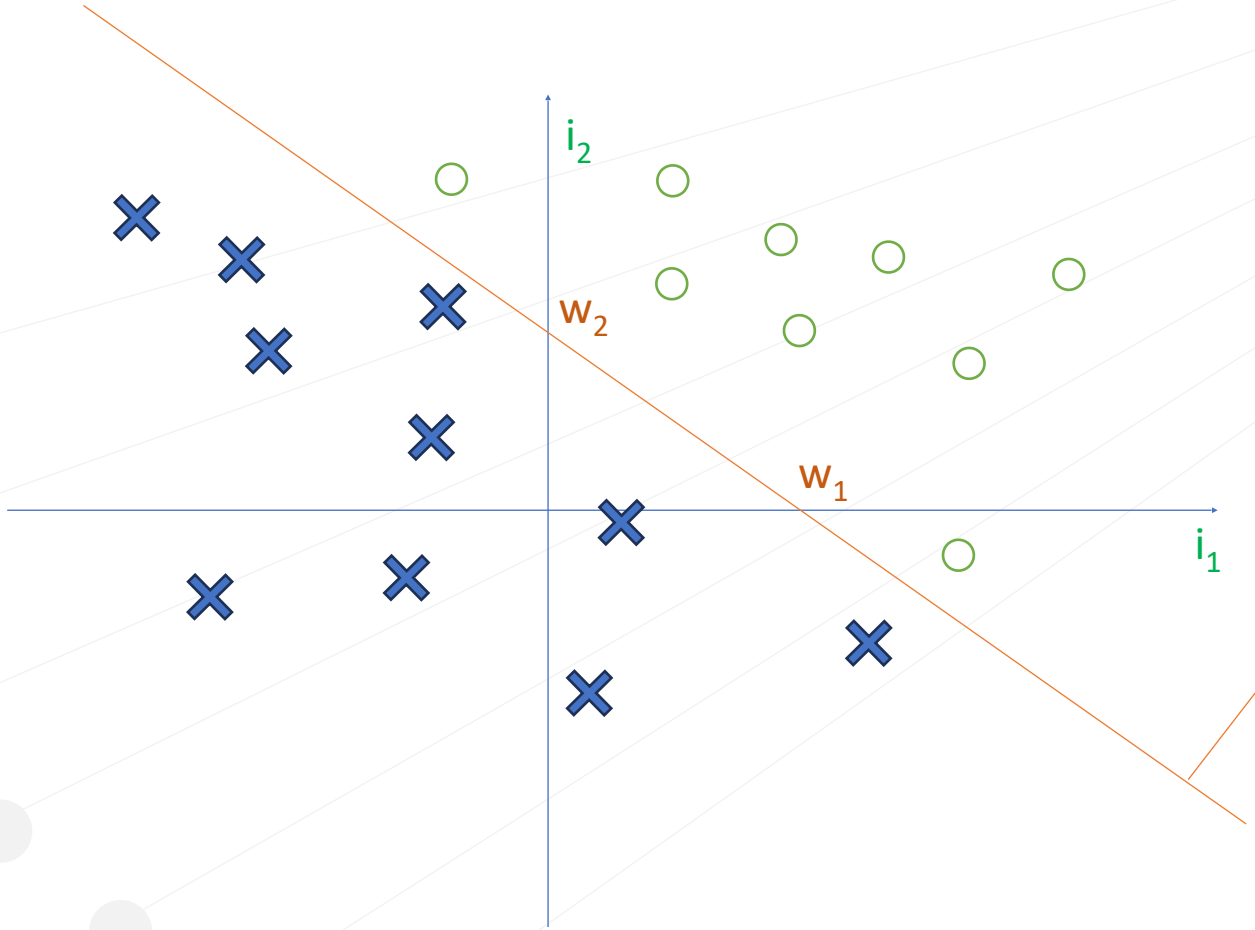
$w_1 * x + w_2 * y = b$

$w_1 * x + w_2 * y > b$ ⟶ ○

$w_1 * x + w_2 * y < b$ ⟶ ×

Un problema semplice di classificazione

Chiamiamo gli assi ‘input’



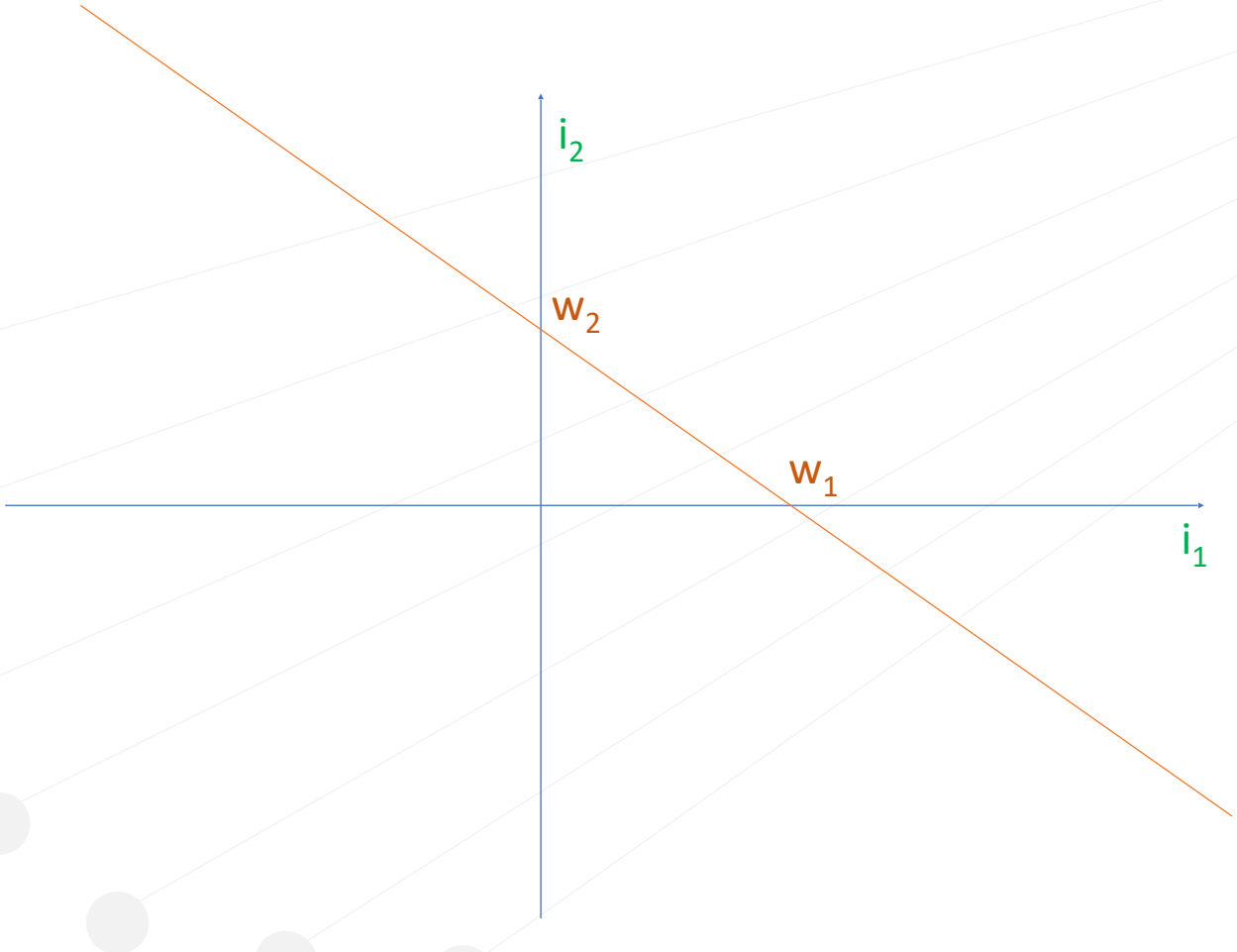
$$w_1 * i_1 + w_2 * i_2 = b$$

$$w_1 * i_1 + w_2 * i_2 > b \longrightarrow \bigcirc$$

$$w_1 * i_1 + w_2 * i_2 < b \longrightarrow \times$$

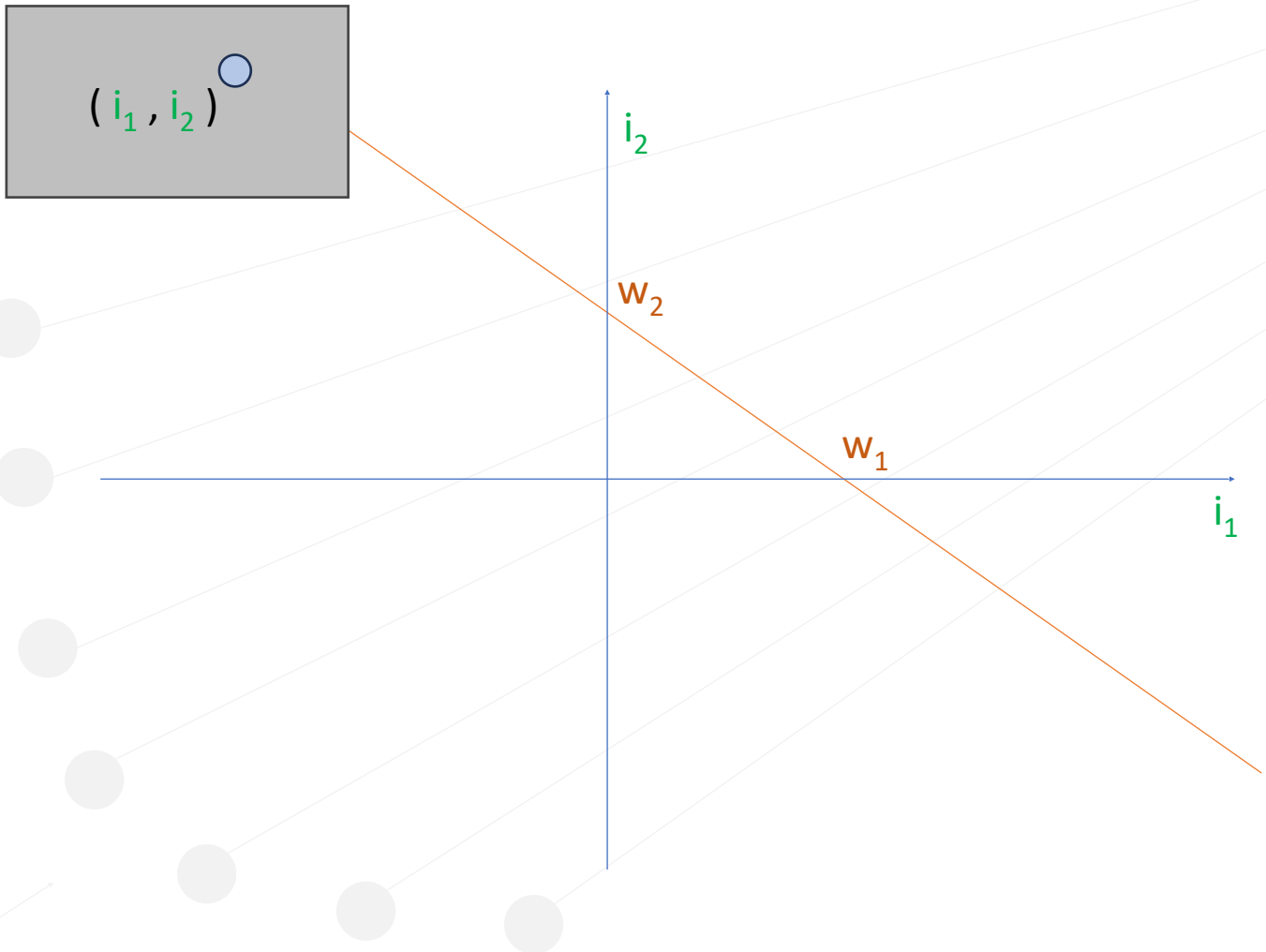
Un problema semplice di classificazione

Classifico un nuovo punto:



Un problema semplice di classificazione

Classifico un nuovo punto: (i_1 , i_2)

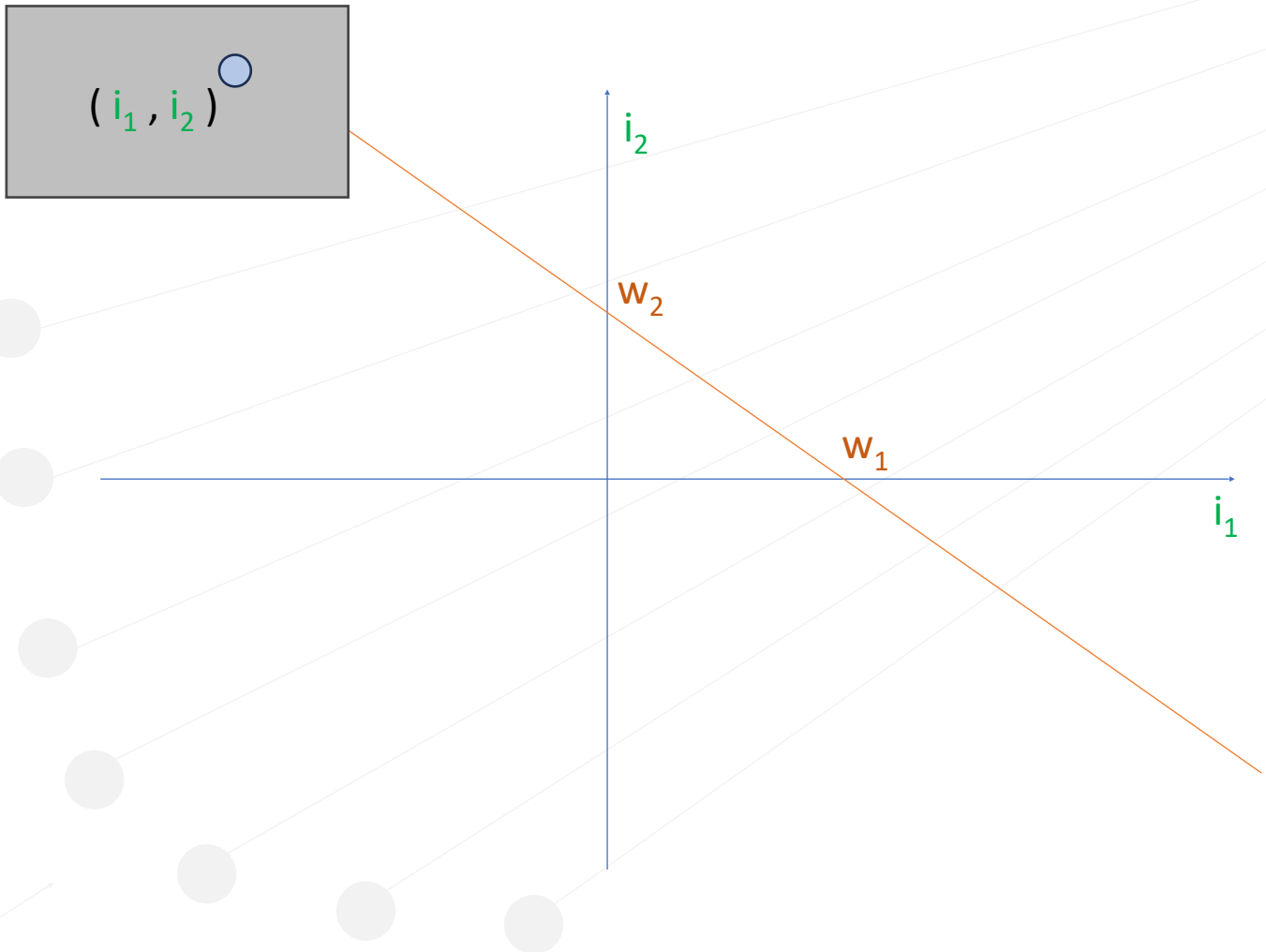


(i_1 , i_2)

i_1 i_2

Un problema semplice di classificazione

Classifico un nuovo punto: (i_1 , i_2)



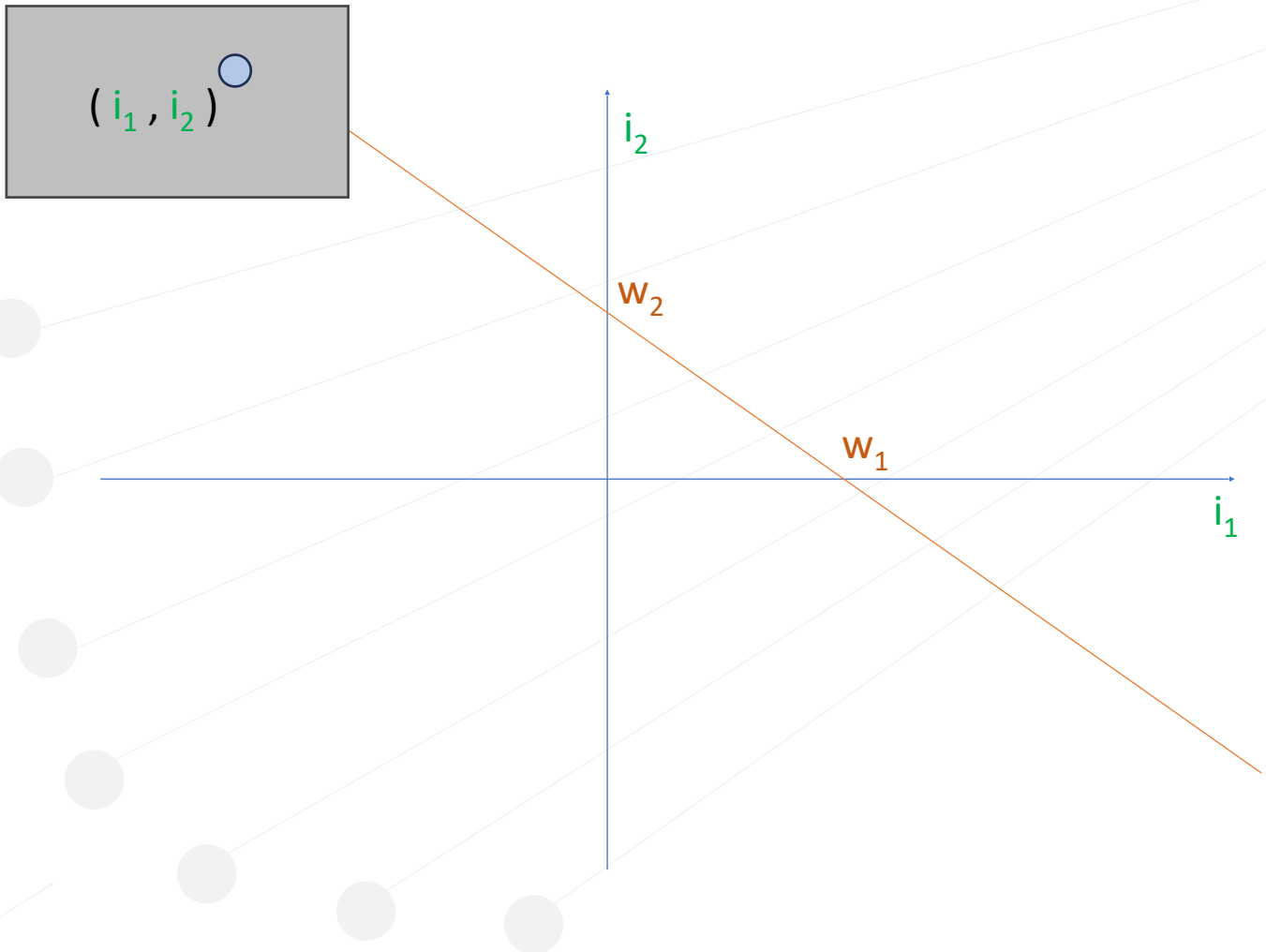
(i_1 , i_2) → * w_1

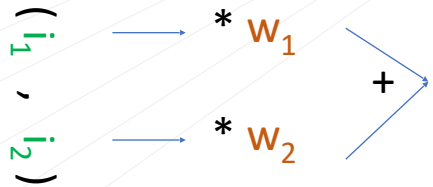
(i_1 , i_2) → * w_2

$w_1 * i_1$ $w_2 * i_2$

Un problema semplice di classificazione

Classifico un nuovo punto: (i_1 , i_2)

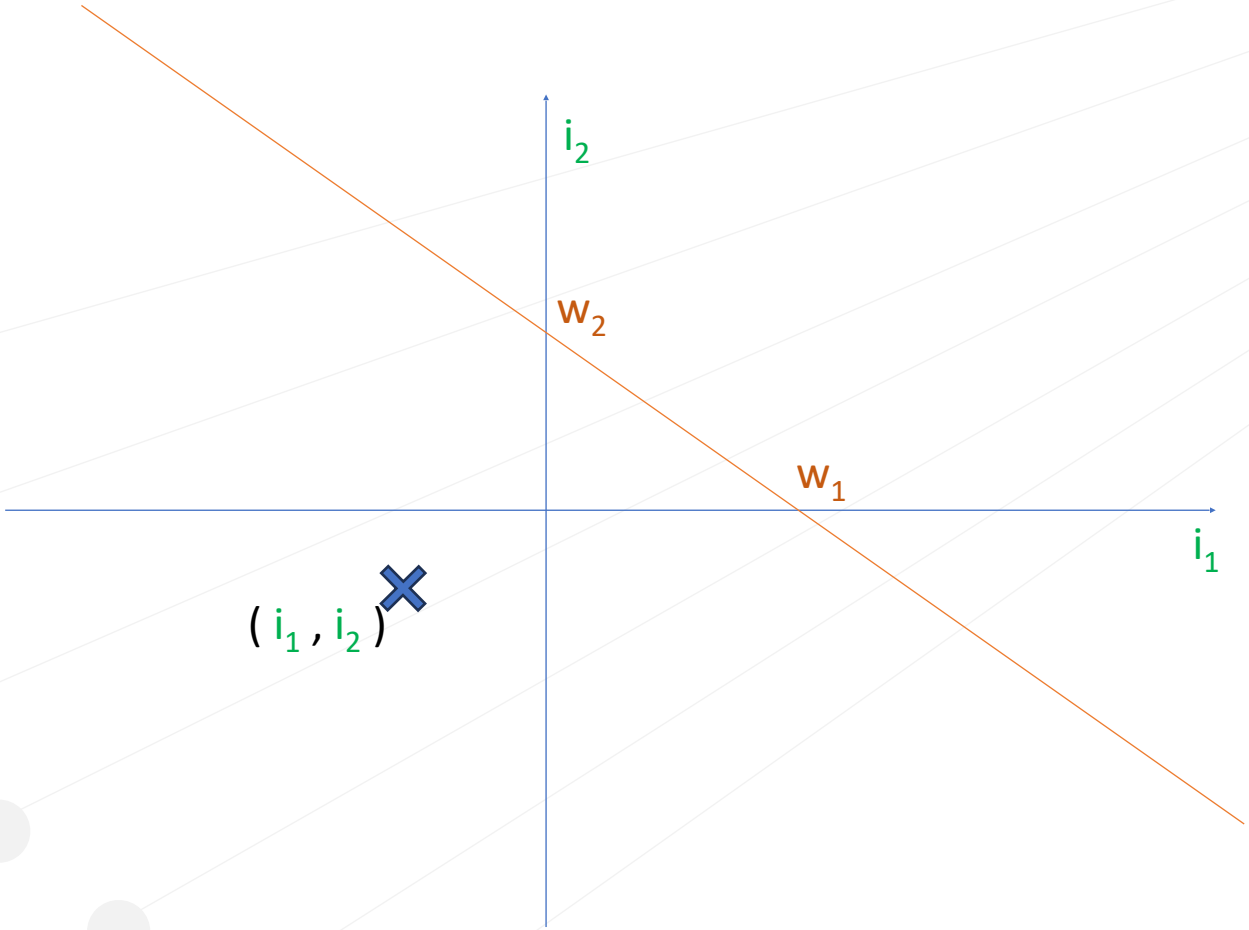


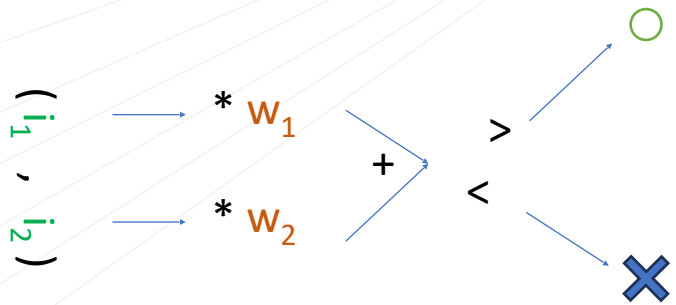


$w_1 * i_1 + w_2 * i_2$

Un problema semplice di classificazione

Classifico un nuovo punto: (i_1 , i_2)

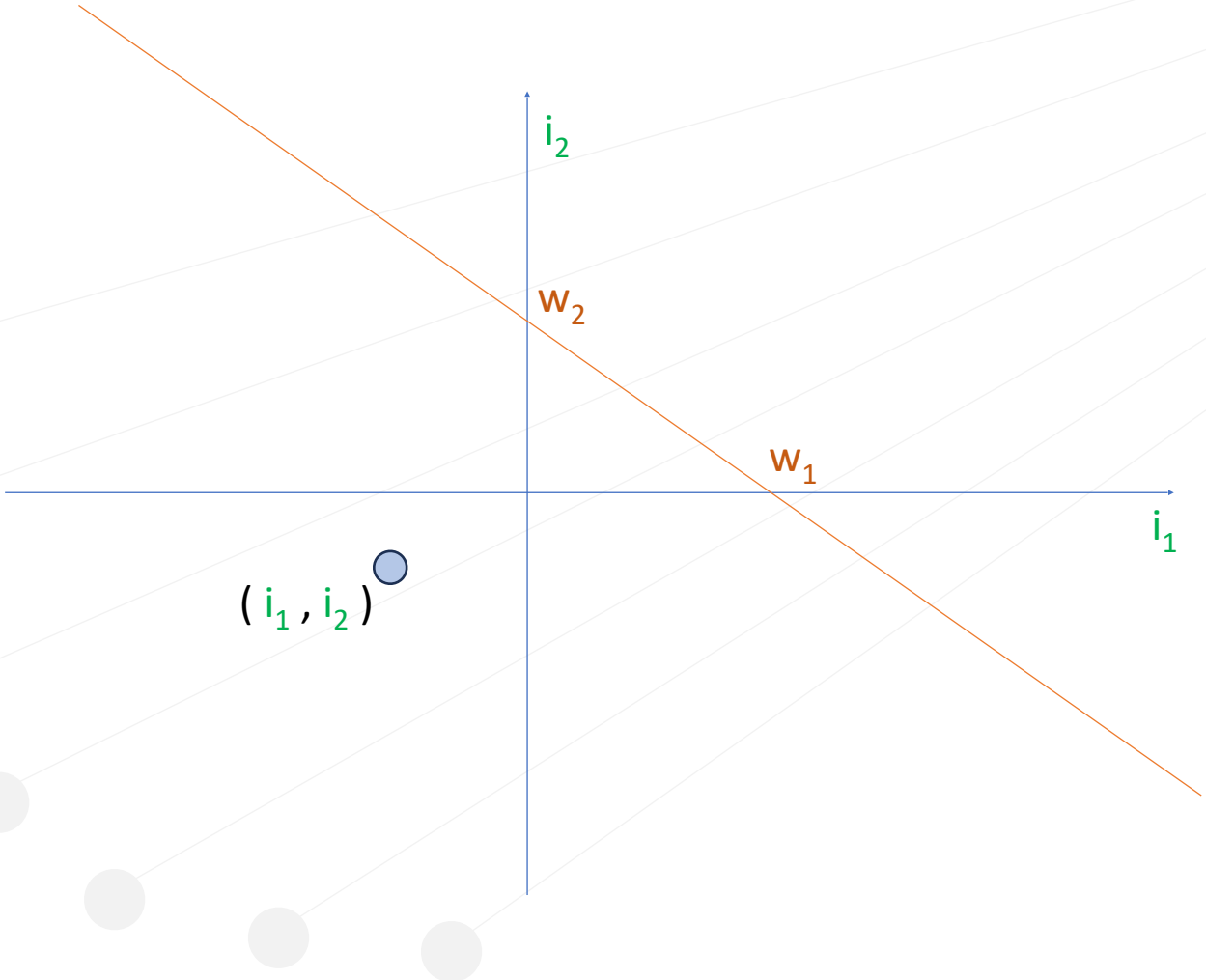


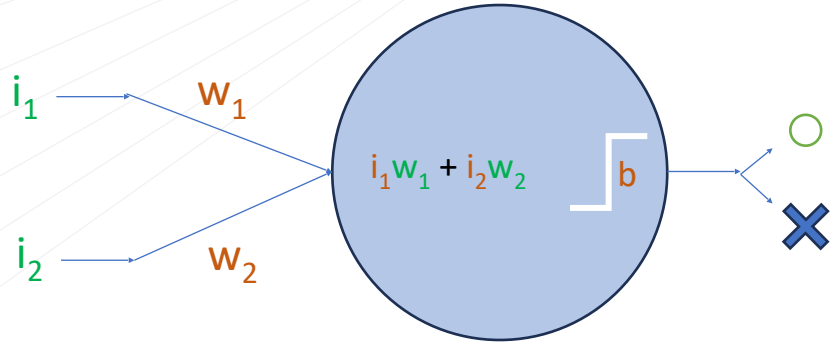


$$w_1 * i_1 + w_2 * i_2 \begin{matrix} > \\ < \end{matrix} b$$

Un problema semplice di classificazione

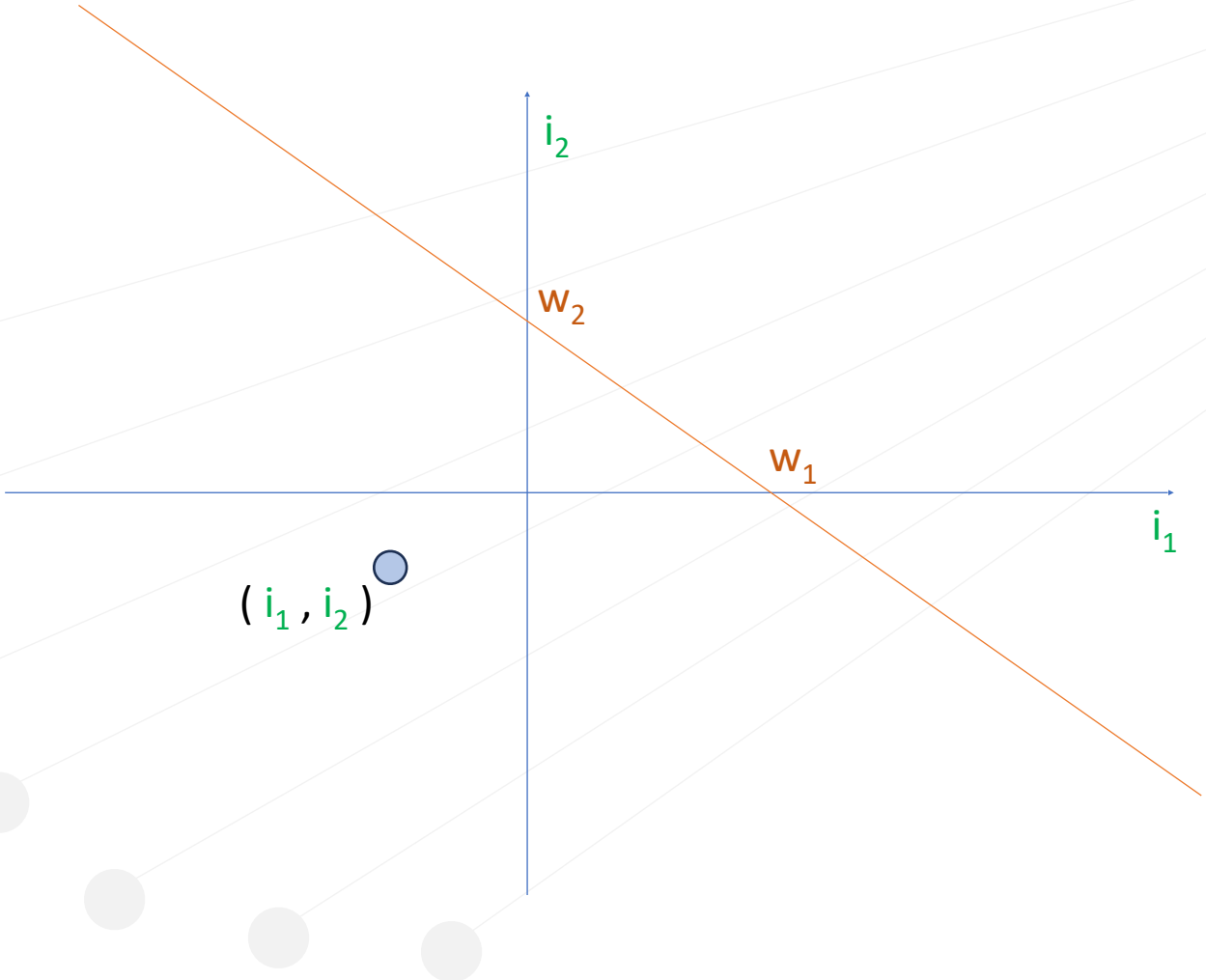
Classifico un nuovo punto: (i_1 , i_2)

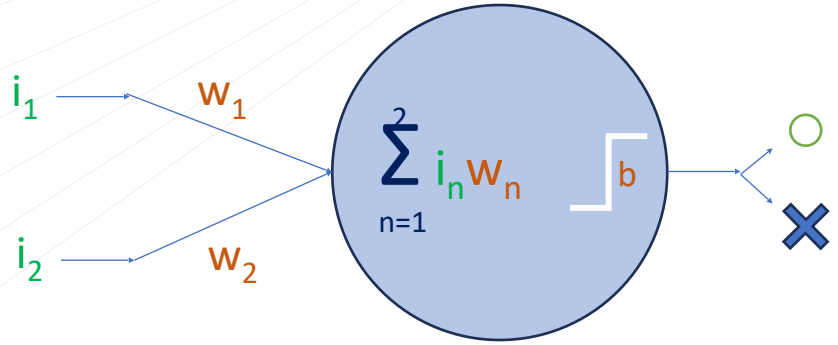




Un problema semplice di classificazione

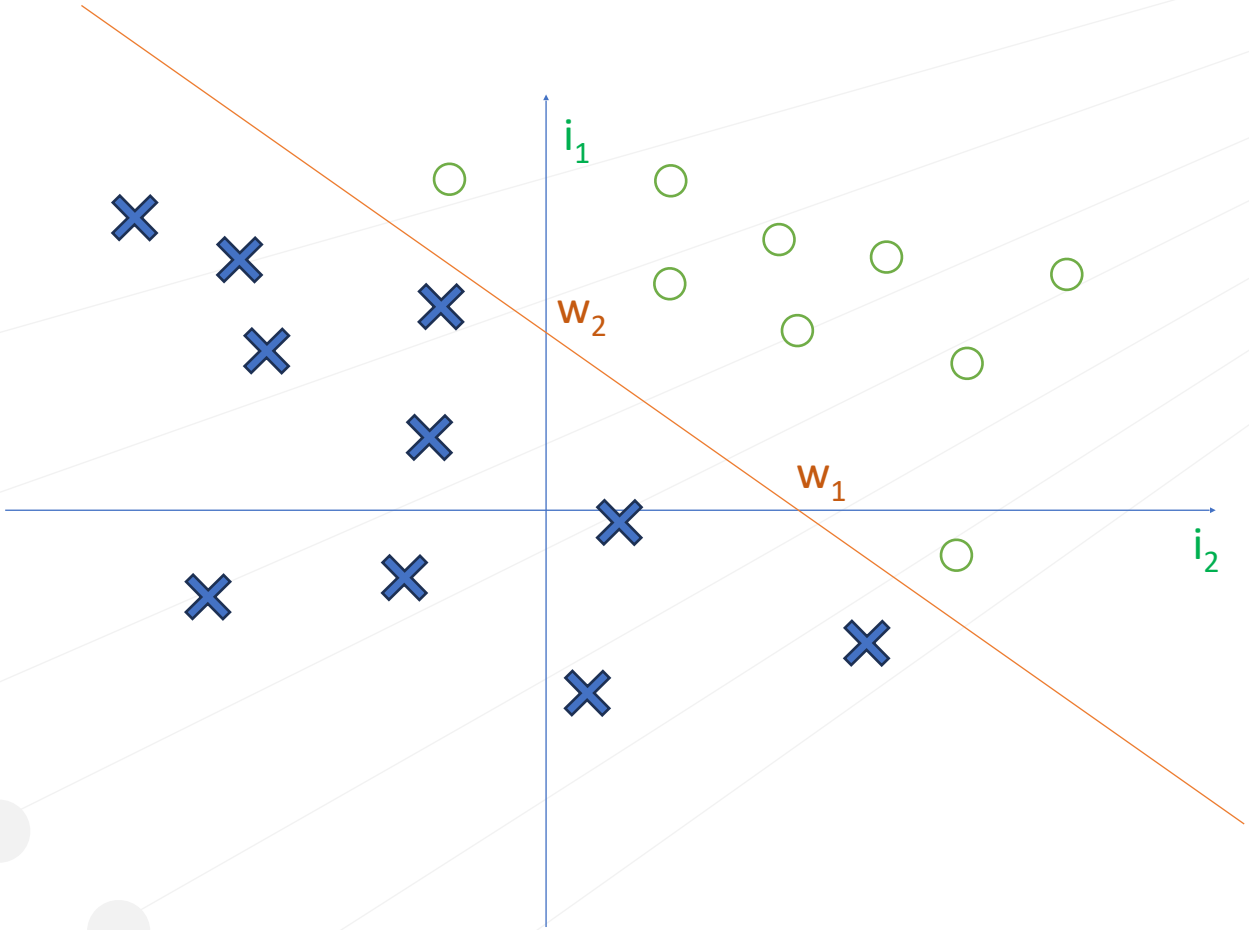
Classifico un nuovo punto: (i_1 , i_2)

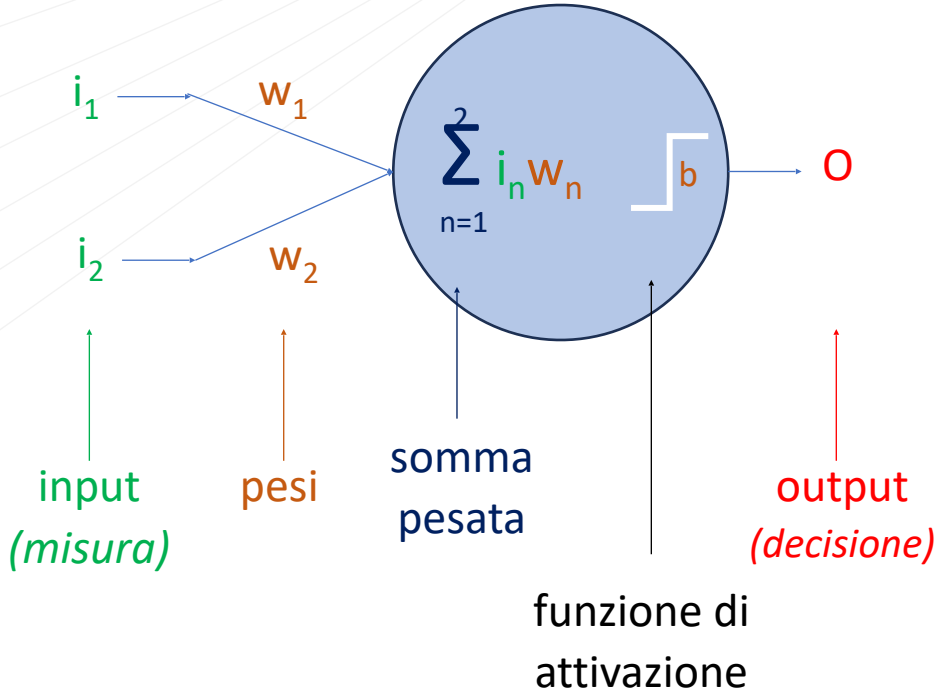




Il percettrone

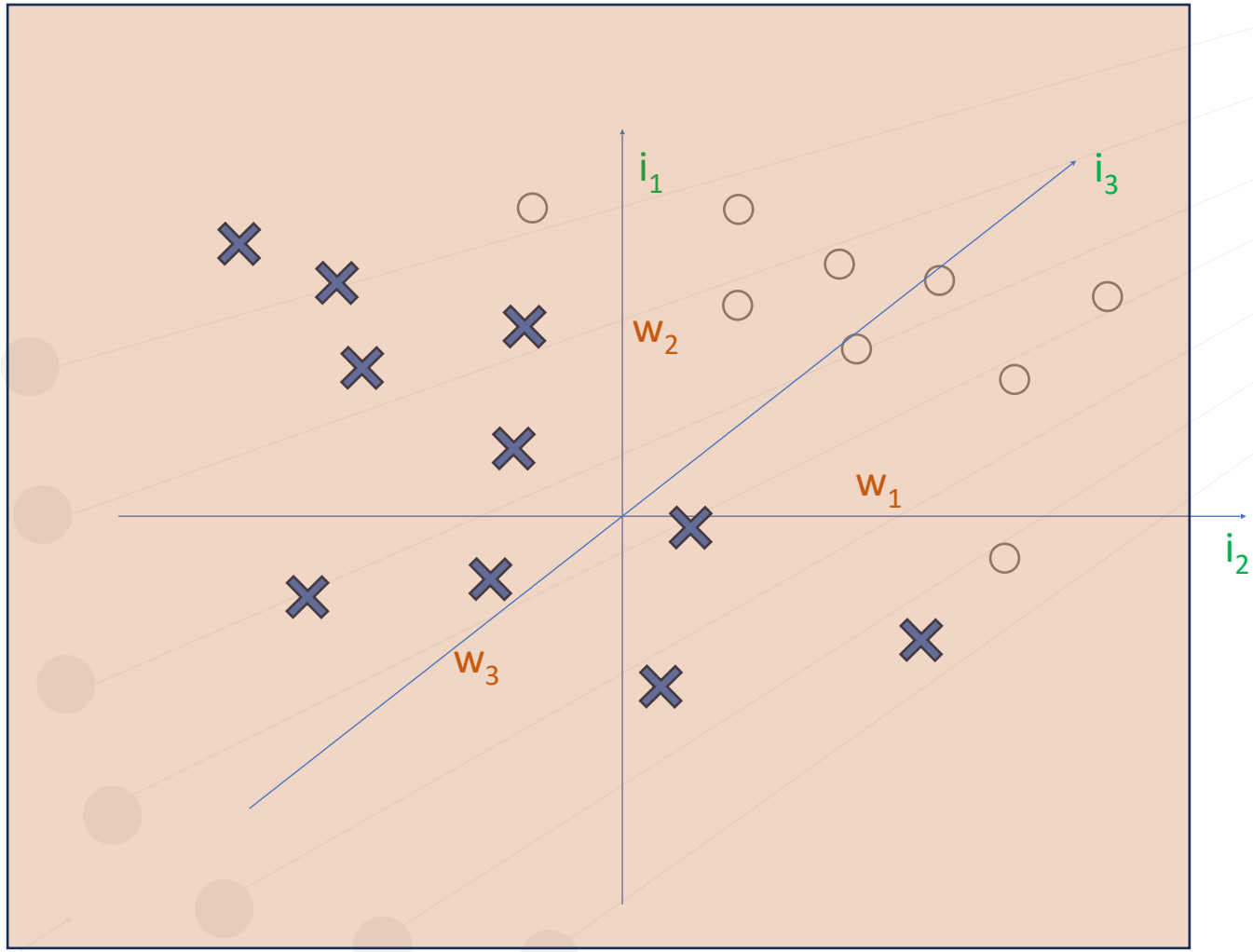
Classificatore lineare (binario in 2D)

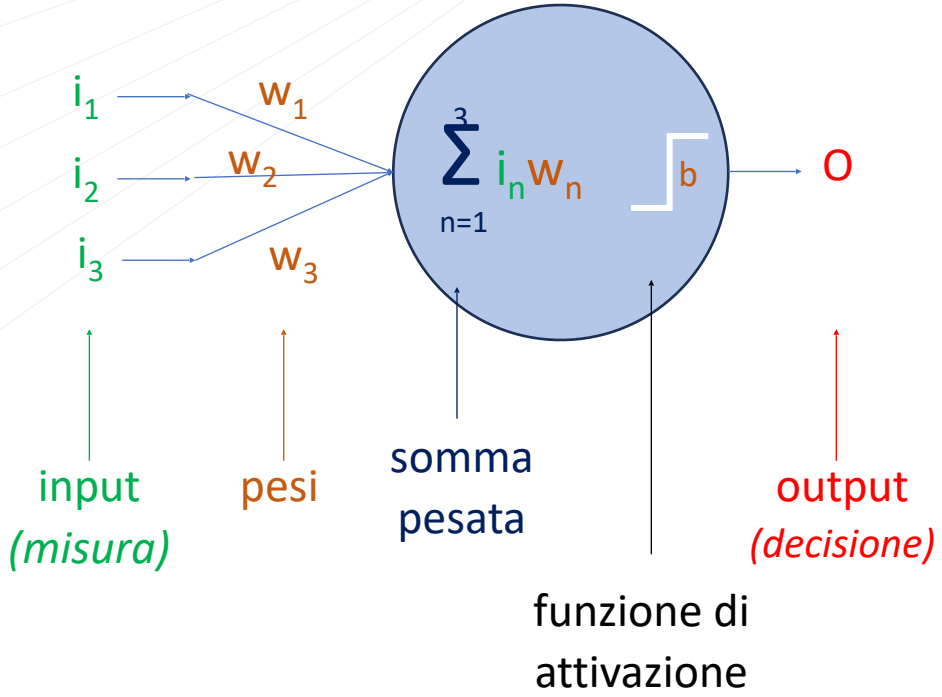




Il percettrone

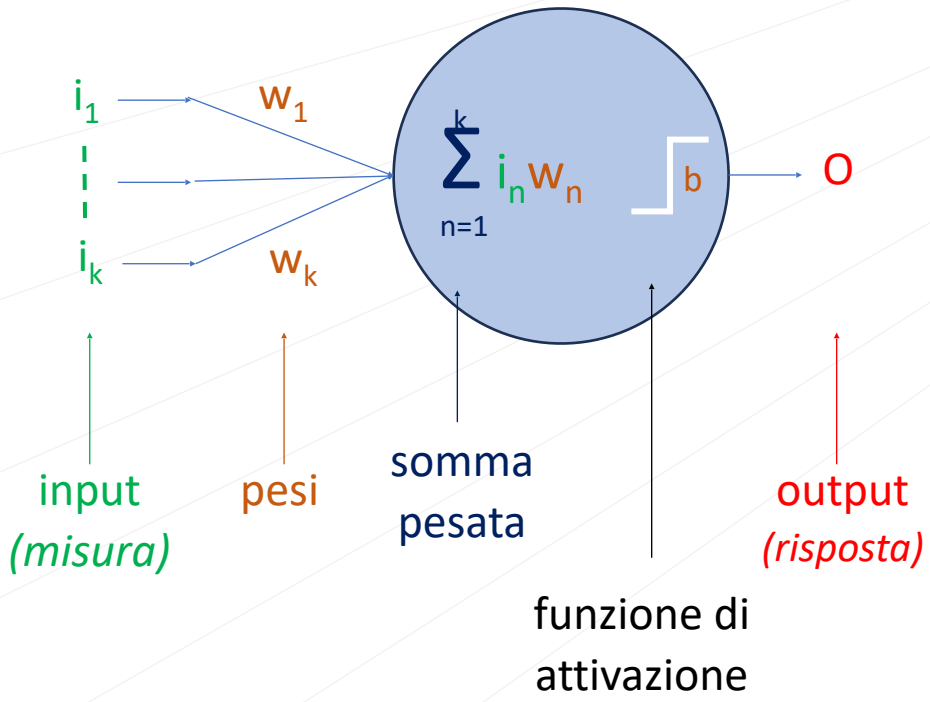
Classificatore lineare (binario in 3D)





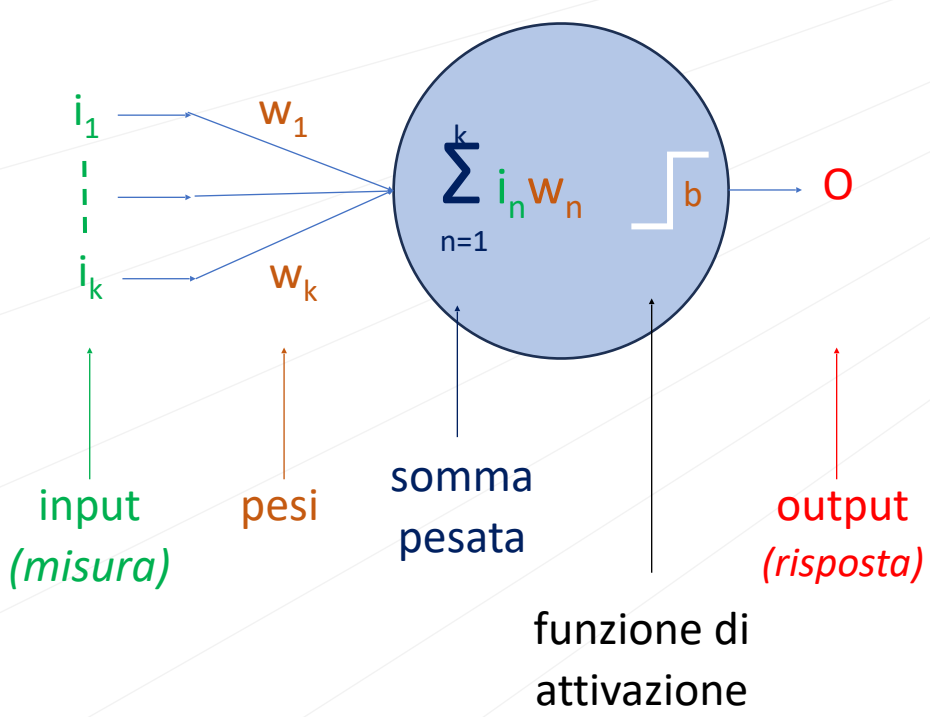
Il percettrone

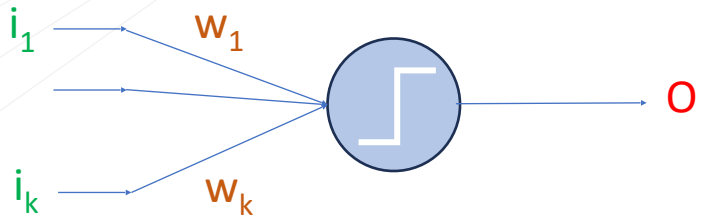
Classificatore lineare (binario in kD)



Il percettrone

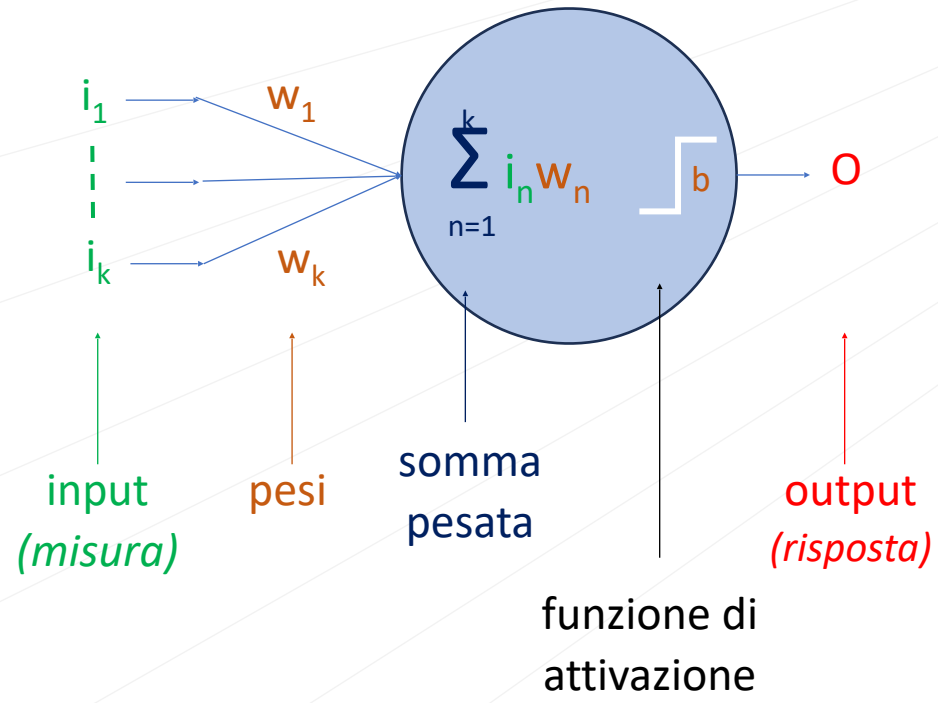
Classificatore lineare (binario in kD)

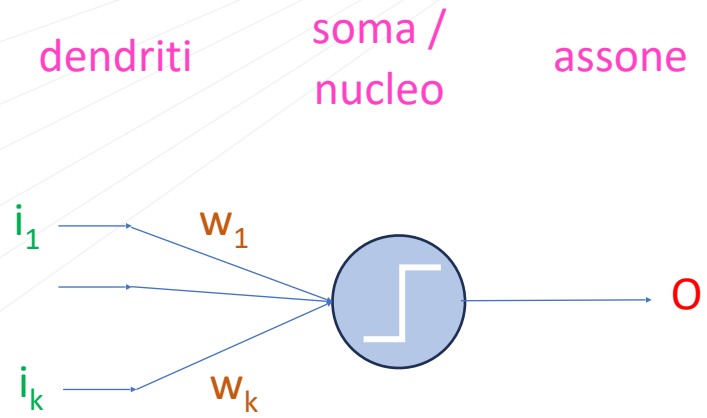


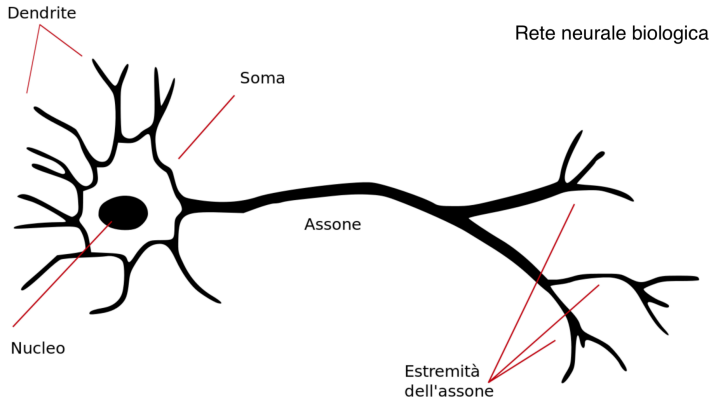


Il percettrone

Classificatore lineare (binario in kD)

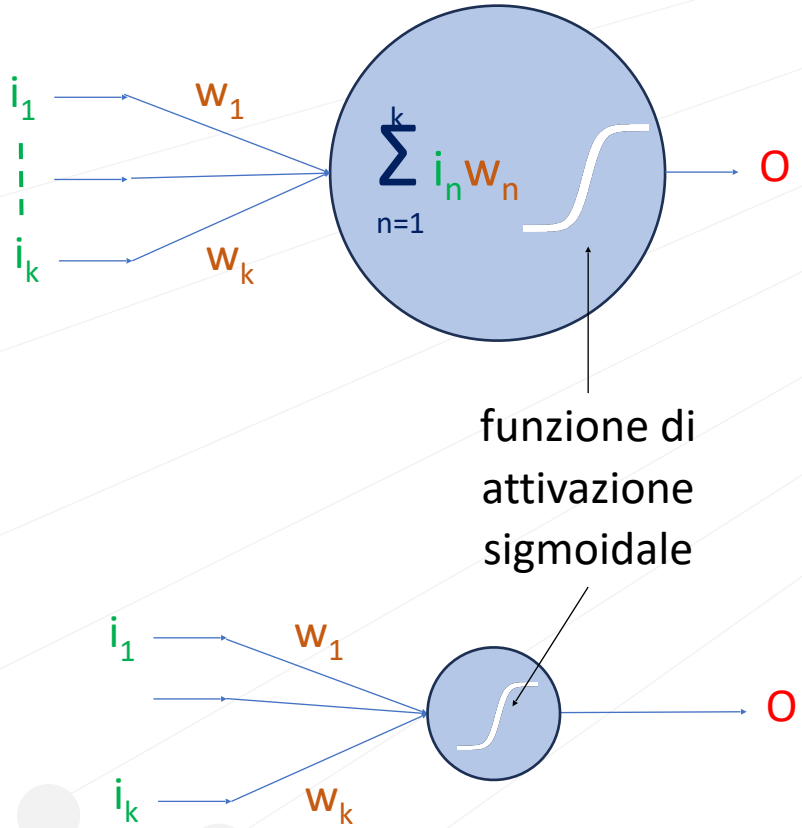






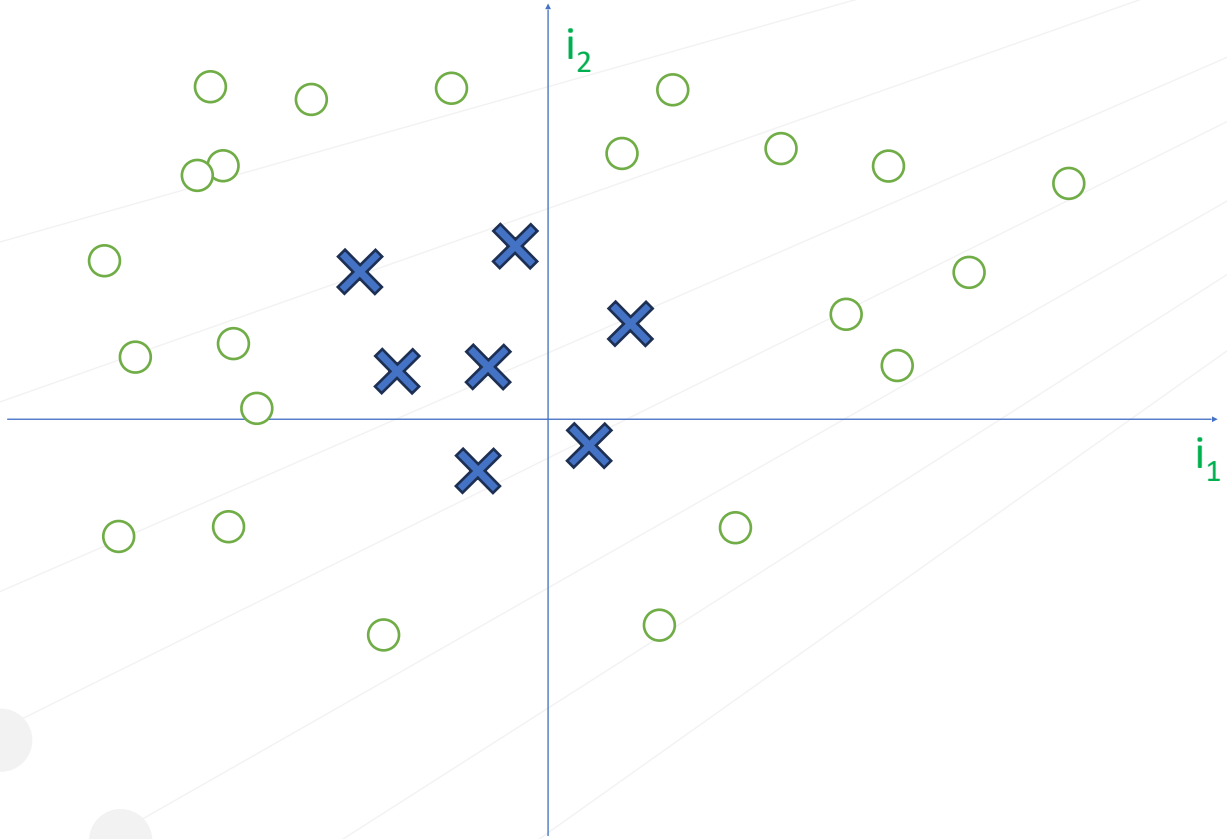
Funzione di attivazione non binaria

Classificatore lineare (probabilistico in kD)



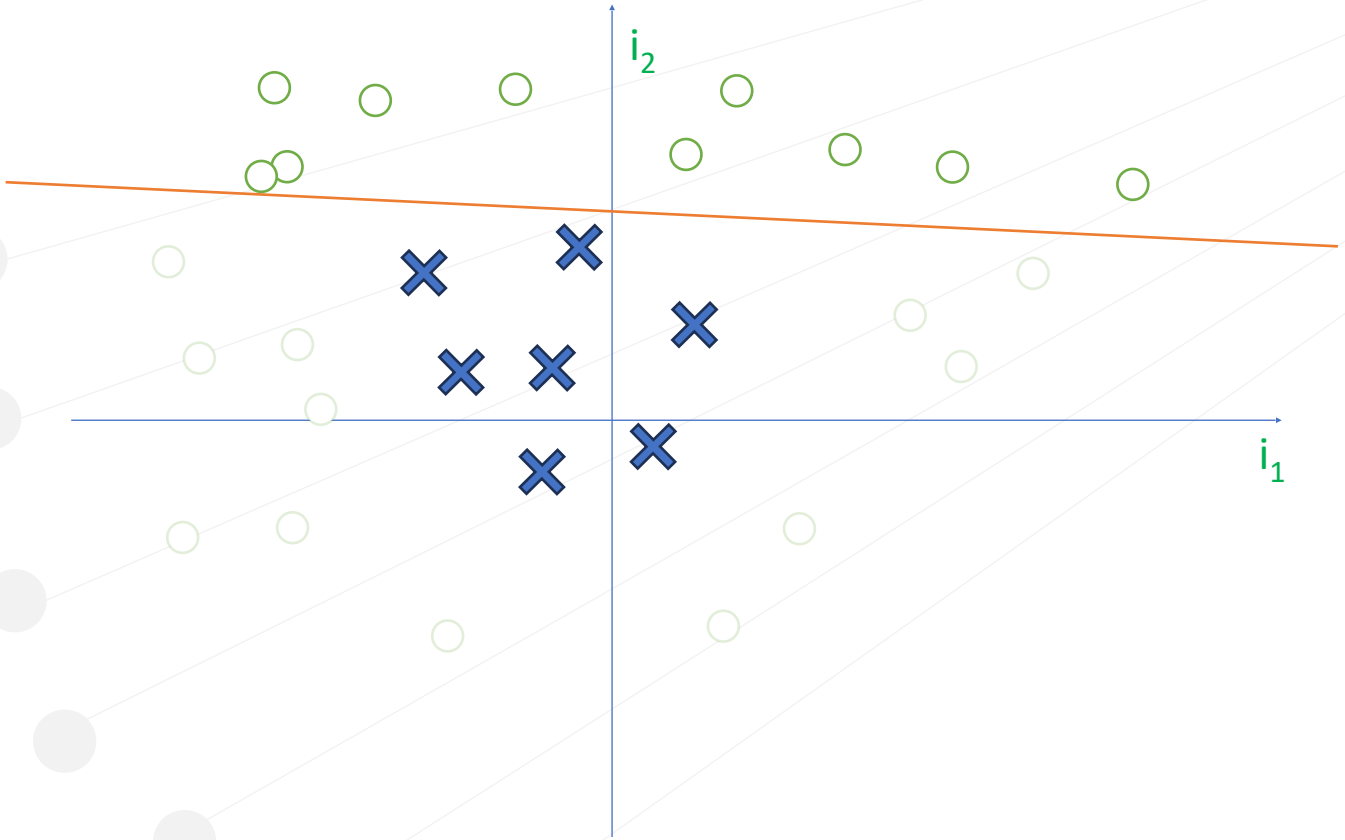
Un problema semplice di classificazione

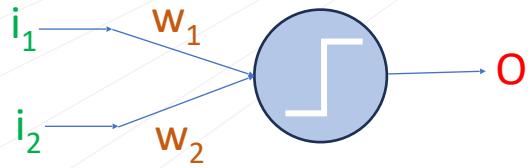
Leggermente più complesso… non basta un classificatore lineare



Un problema semplice di classificazione

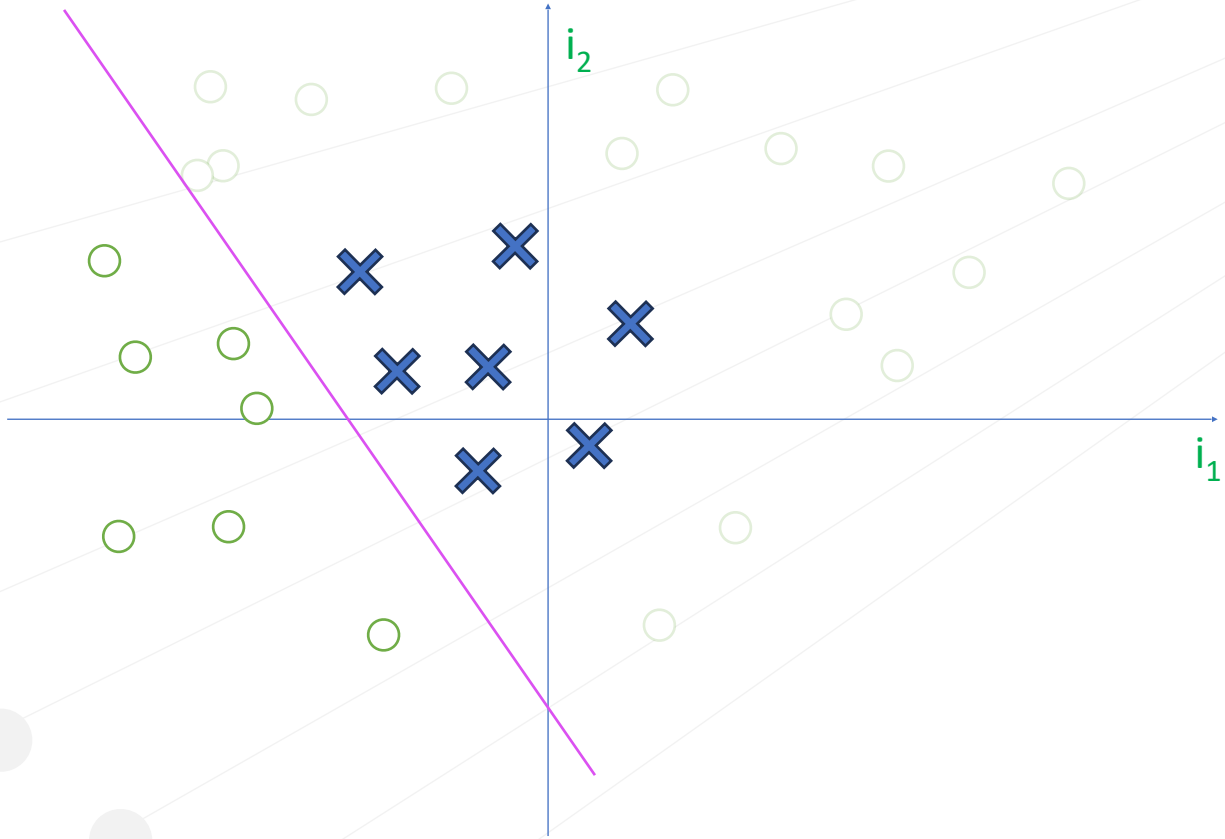
Primo classificatore

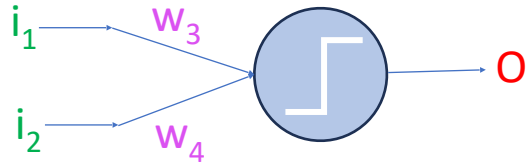




Un problema semplice di classificazione

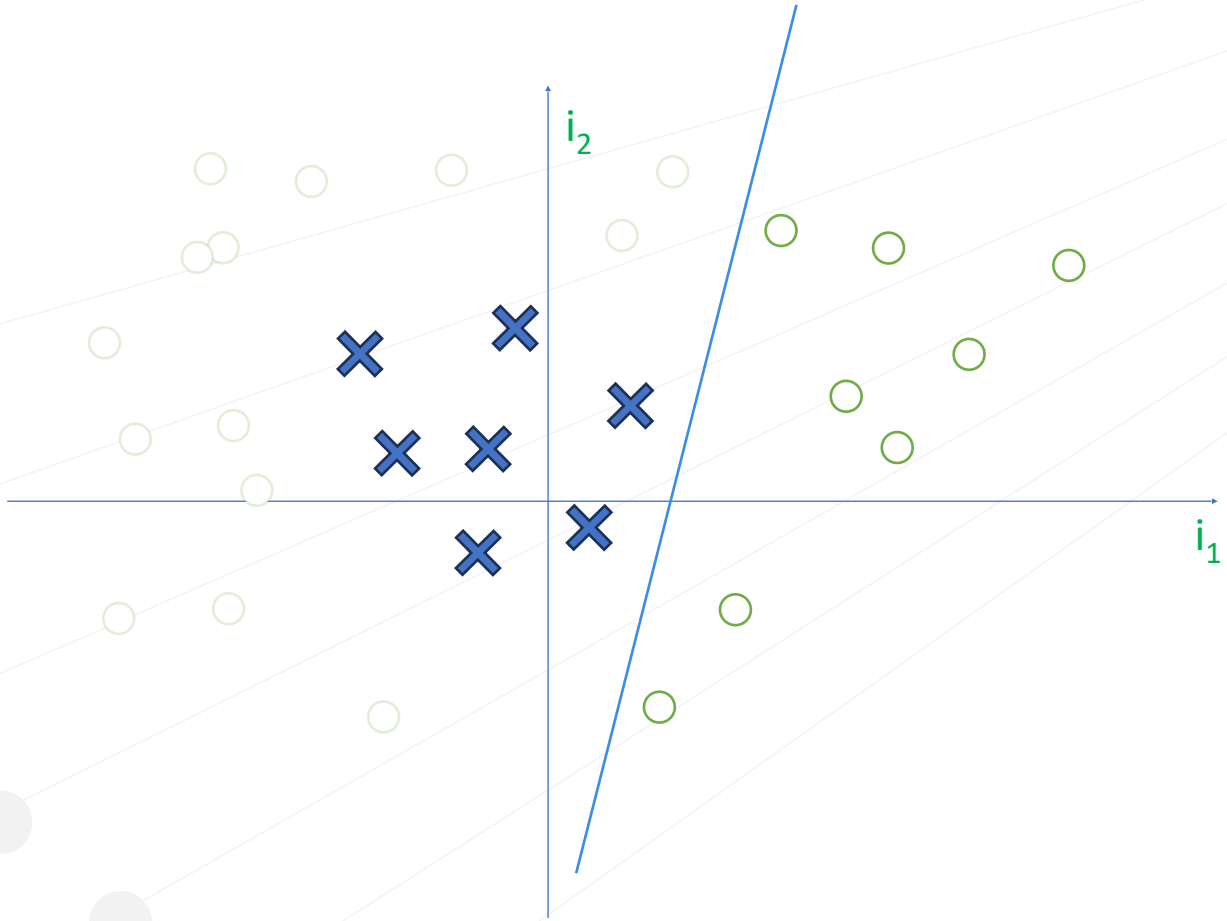
Secondo classificatore

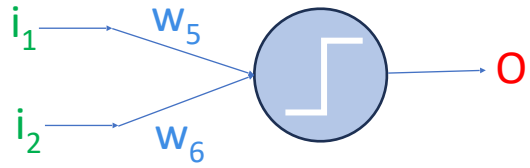




Un problema semplice di classificazione

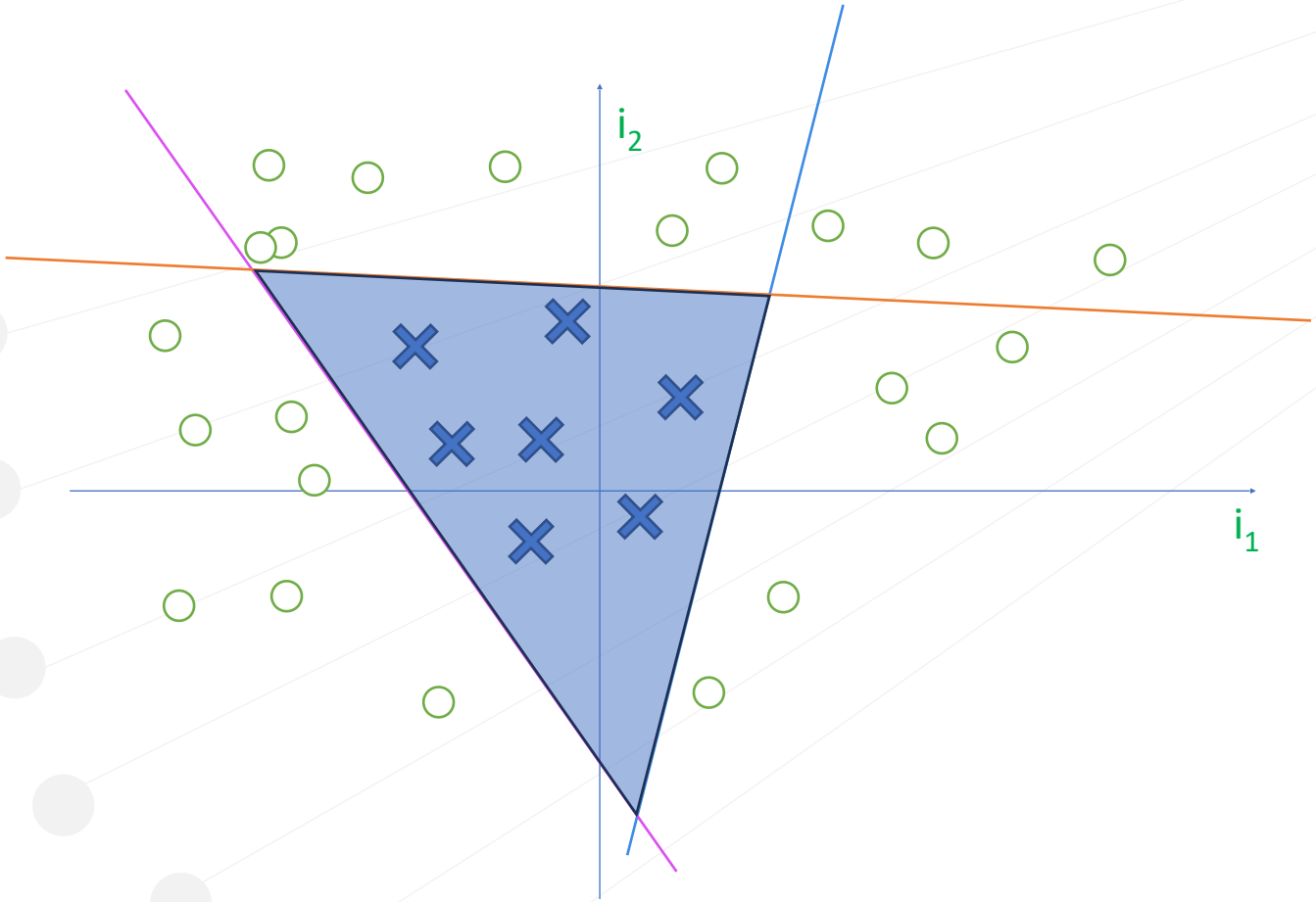
Terzo classificatore

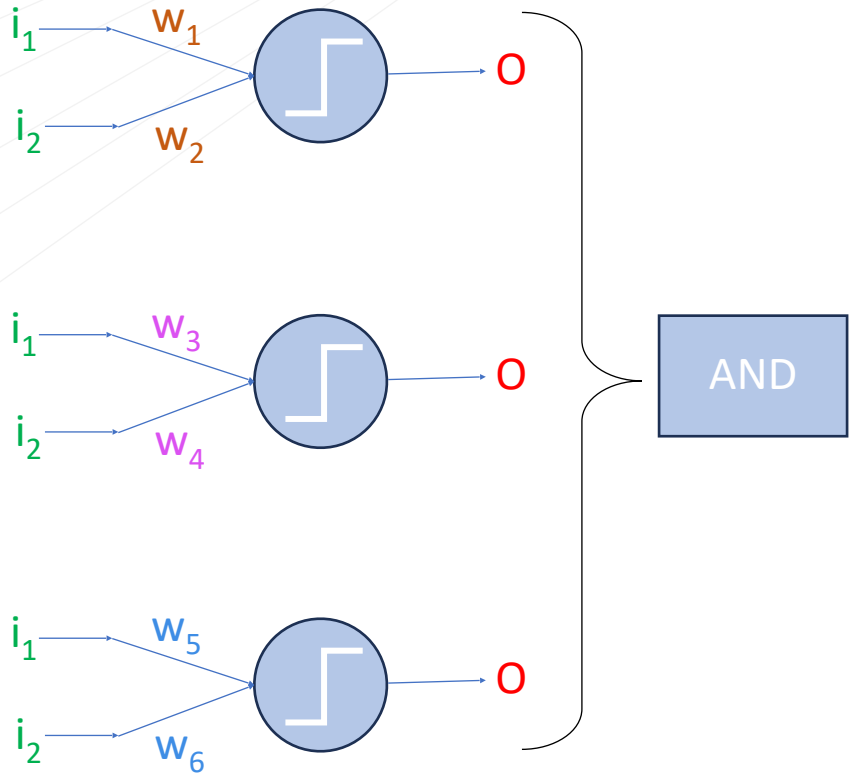




Un problema semplice di classificazione

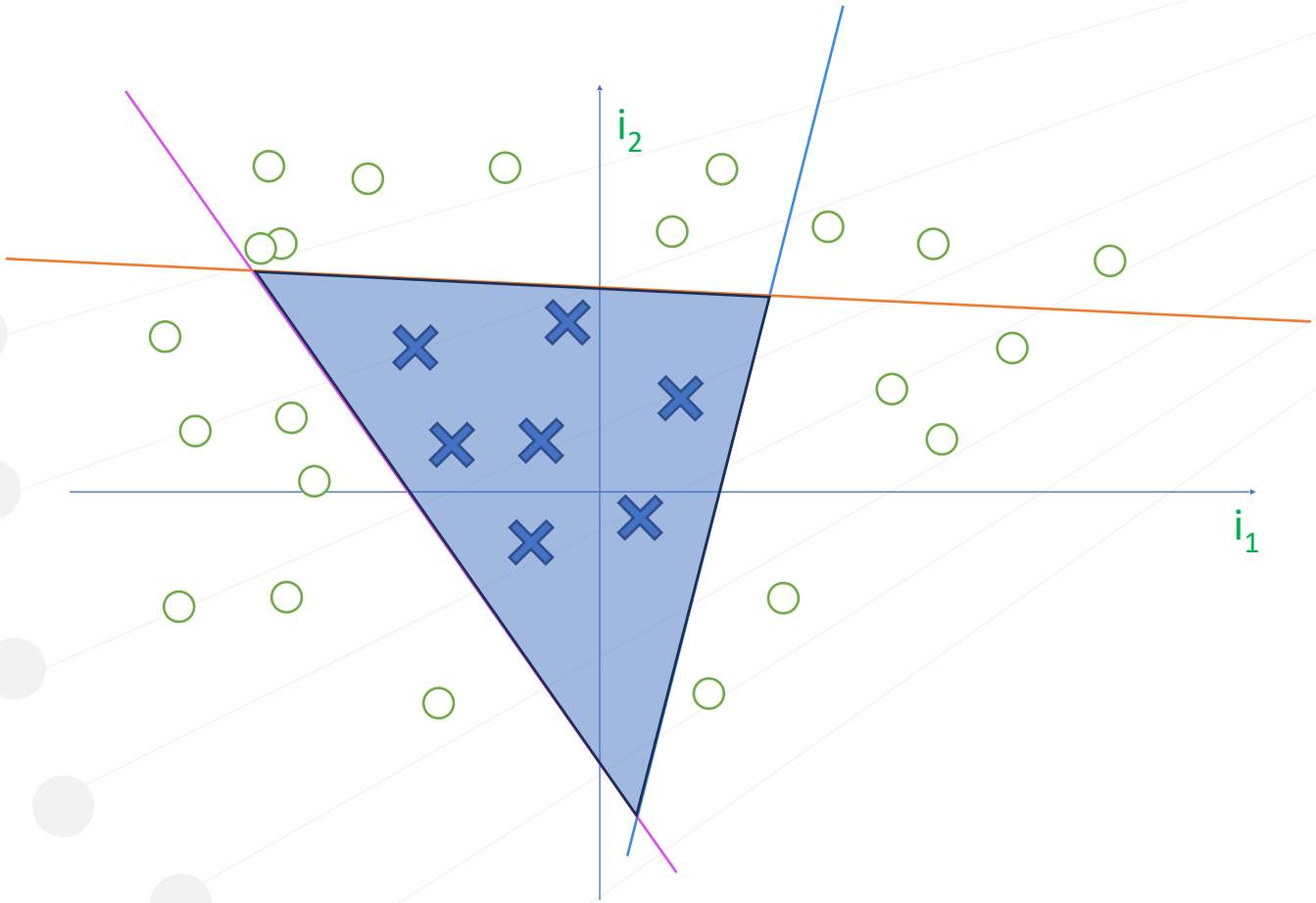
Tutto insieme

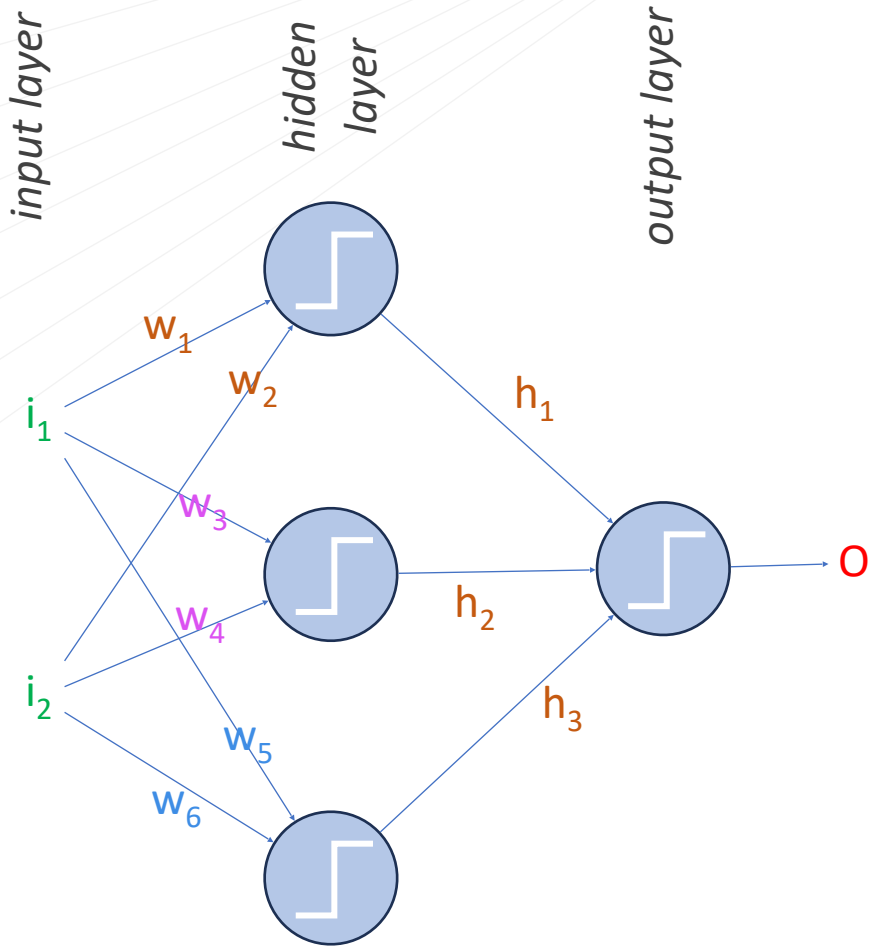




Un problema semplice di classificazione

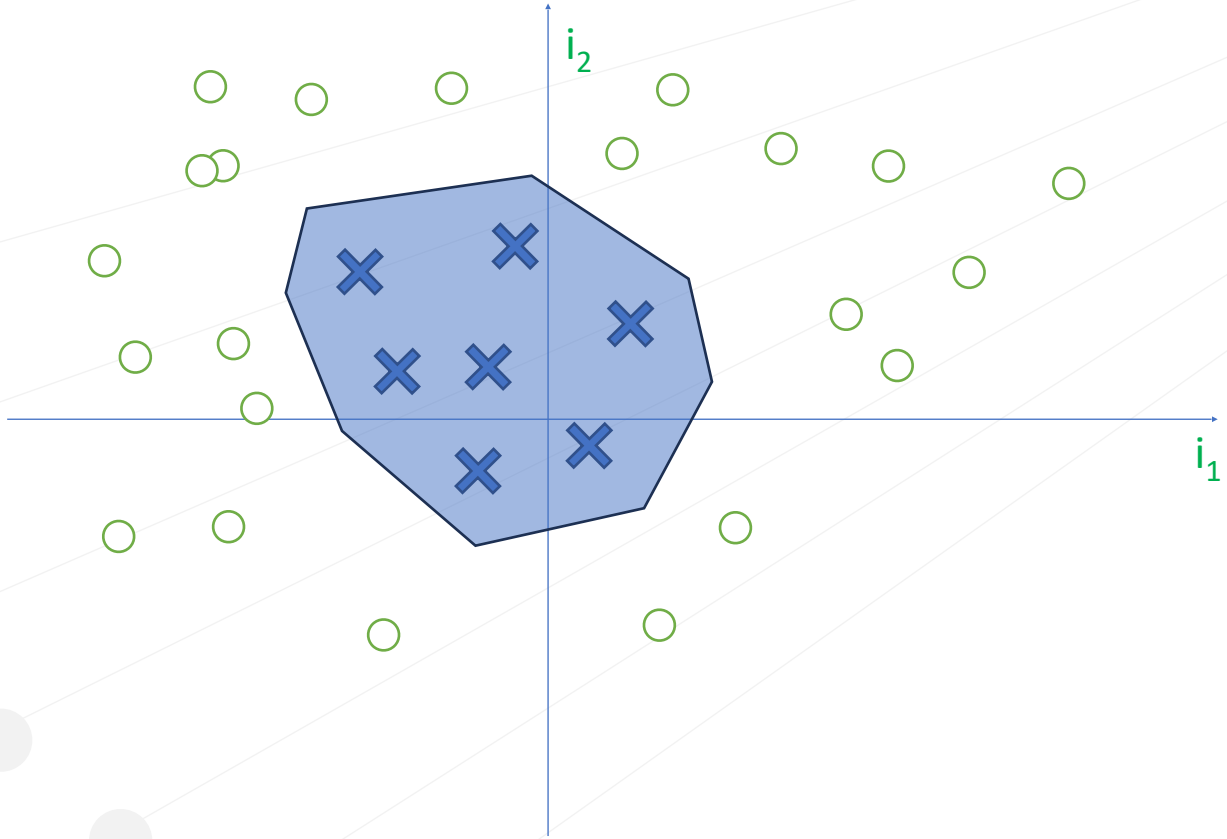
Multi Layer Perceptron

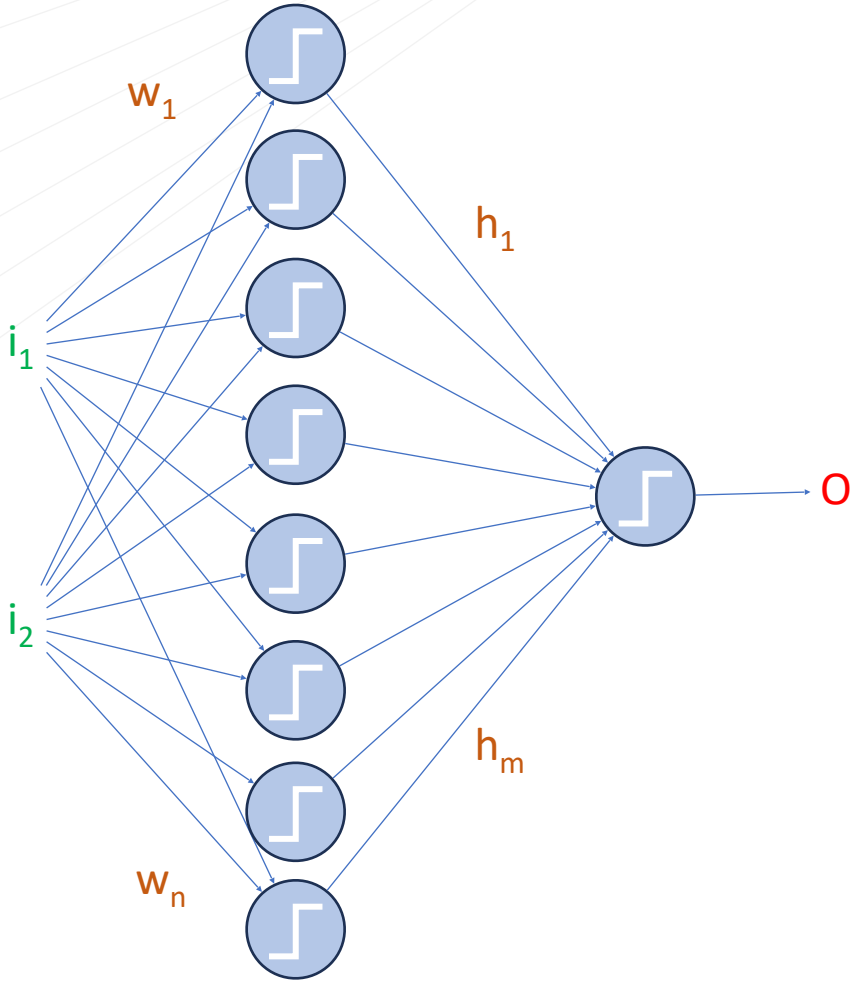




Multi Layer Perceptron

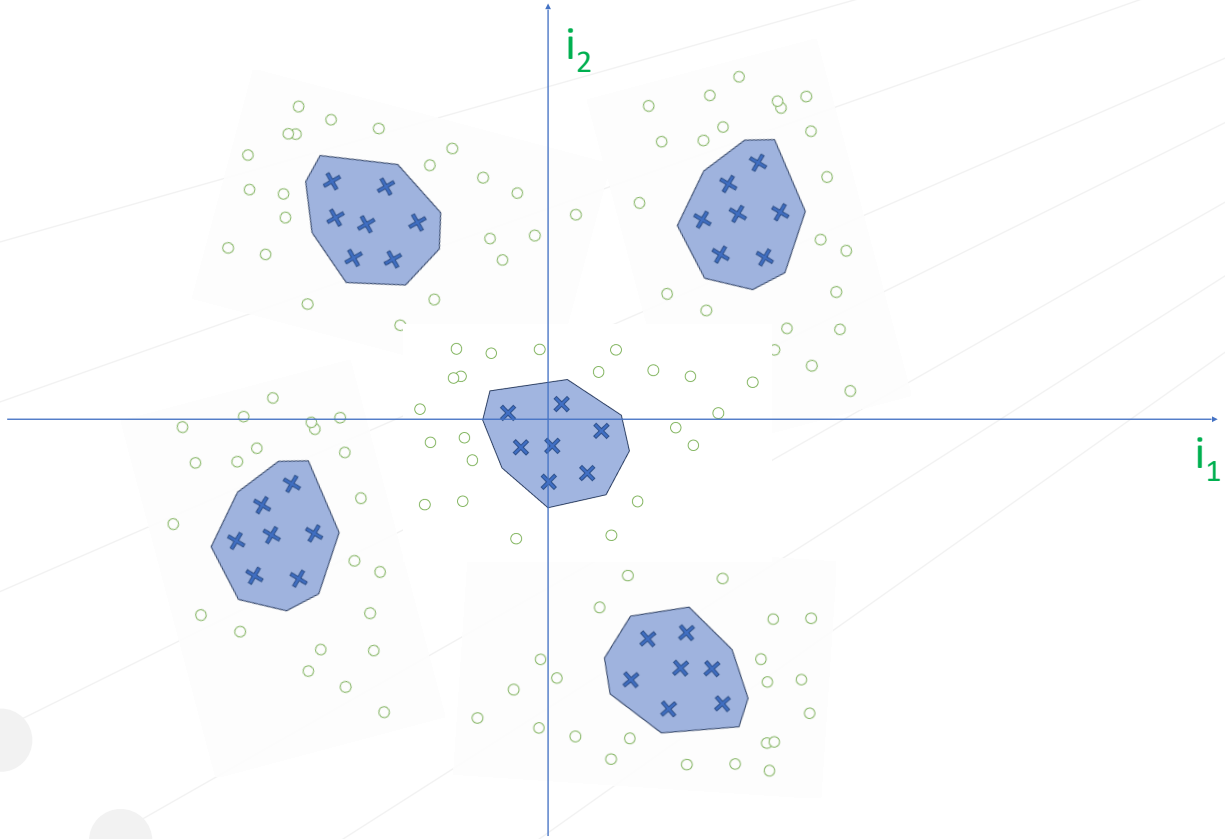
Circoscrivo meglio

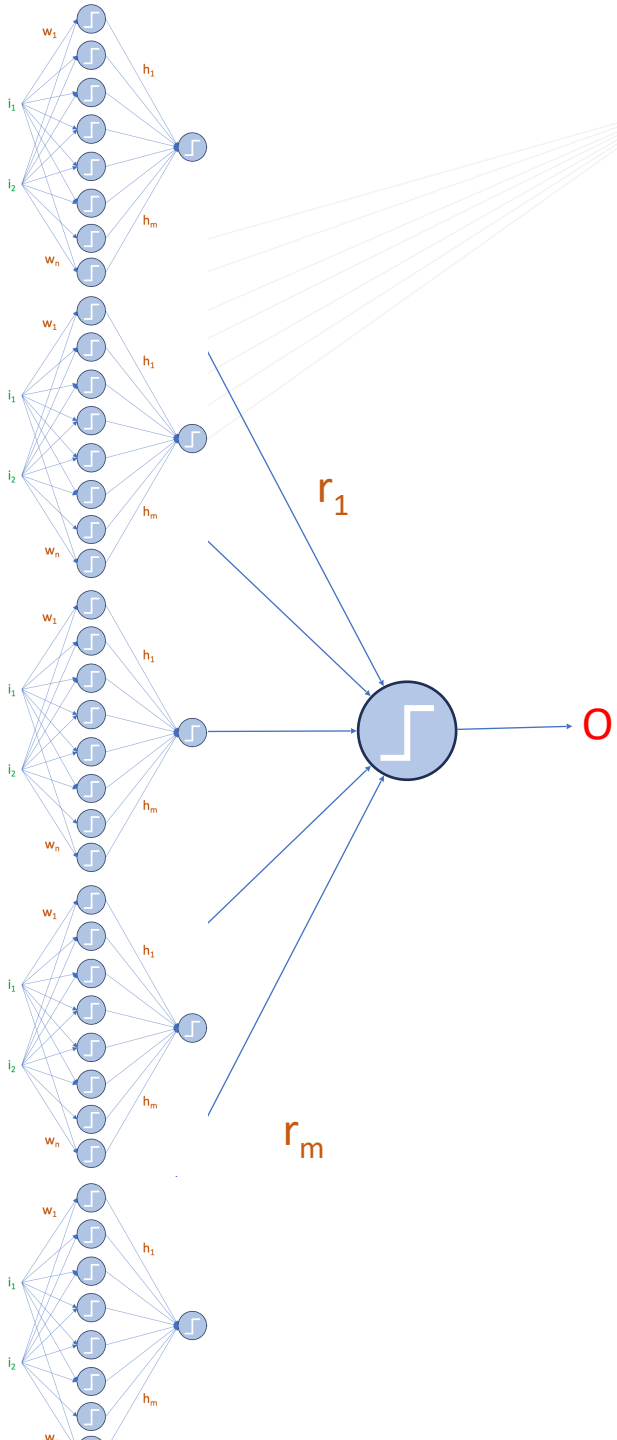




Multi Layer Perceptron

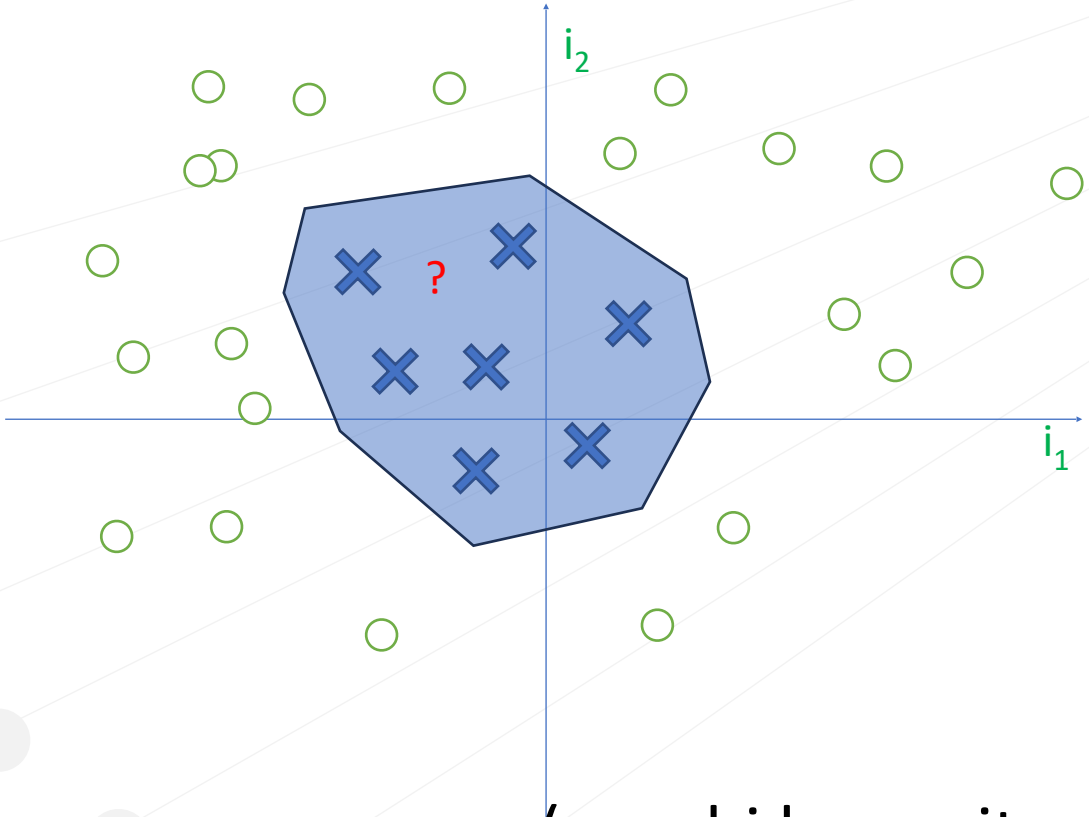
Un mondo un po’ più articolato



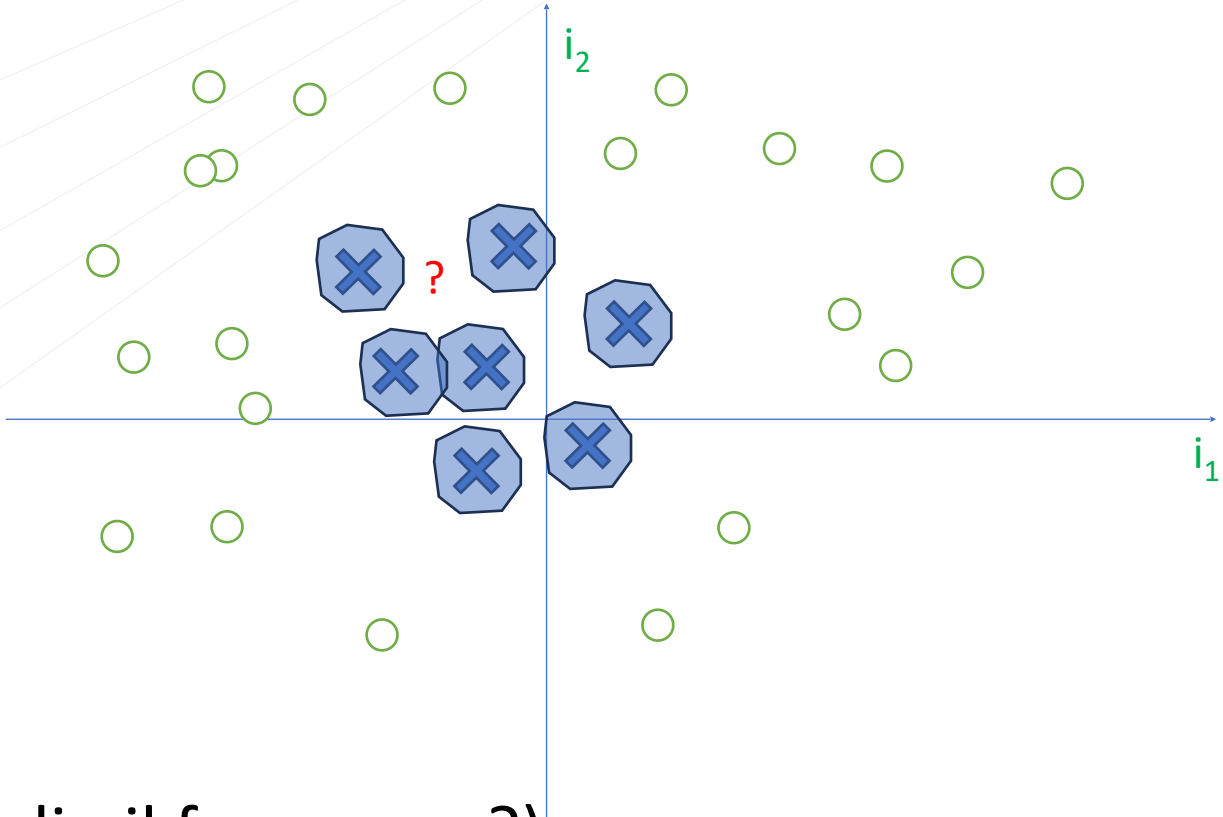


Topologie di MLP

8 neuroni



56 neuroni

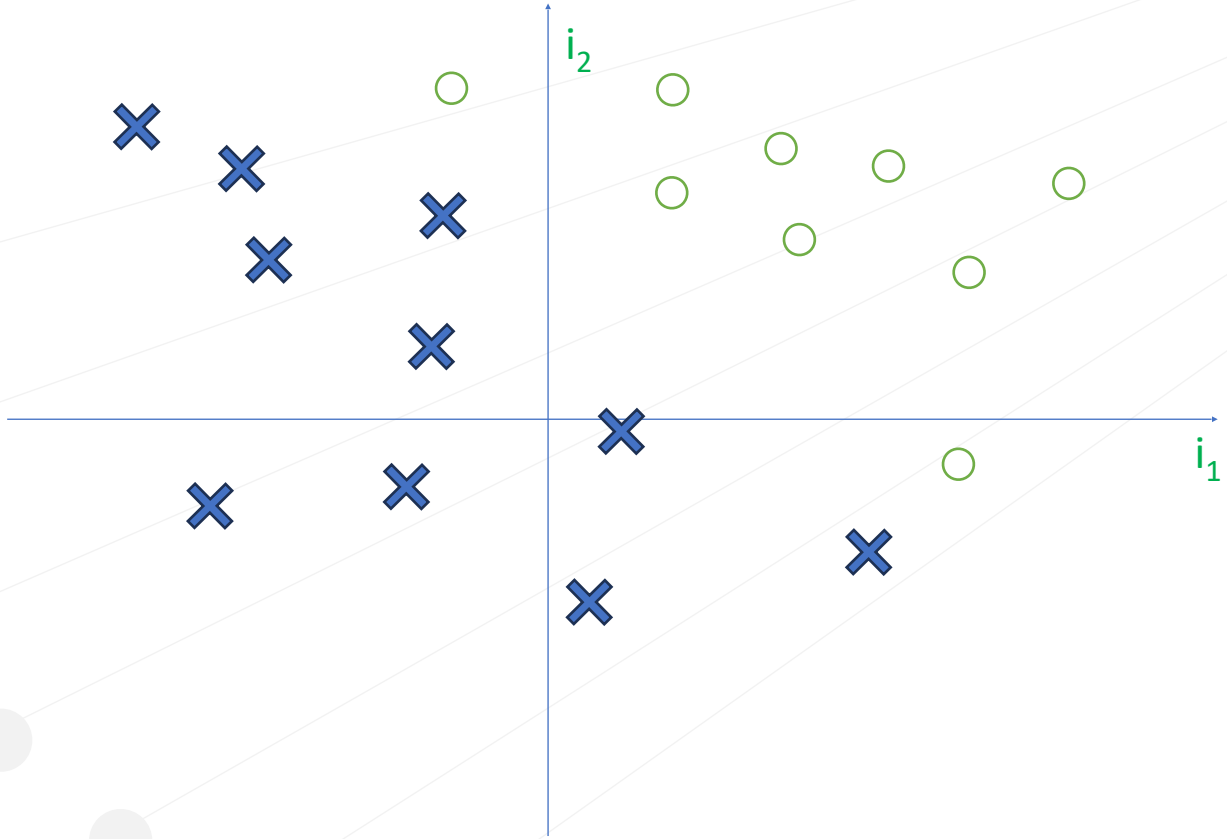


(ma chi ha capito meglio il fenomeno?)

BREAK

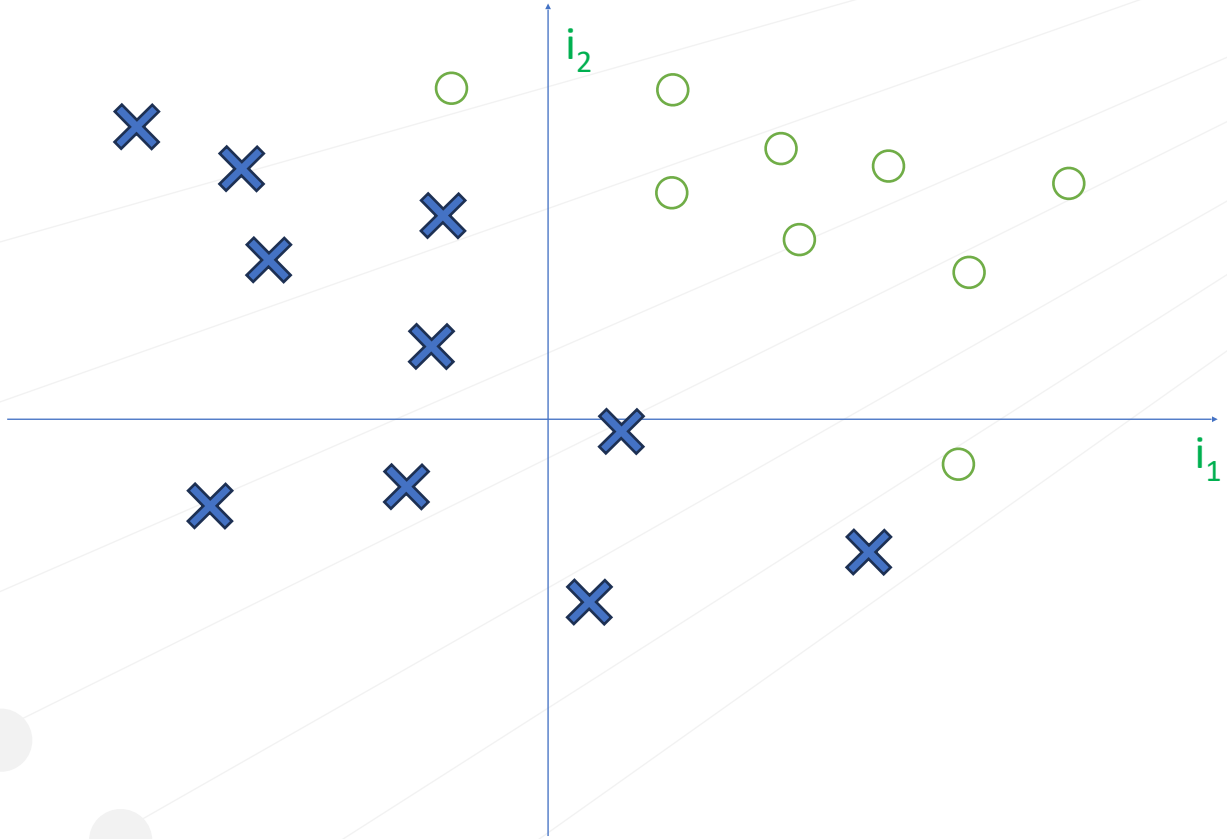
Addestramento

Come trovo la riga (iperpiano) ?



Addestramento

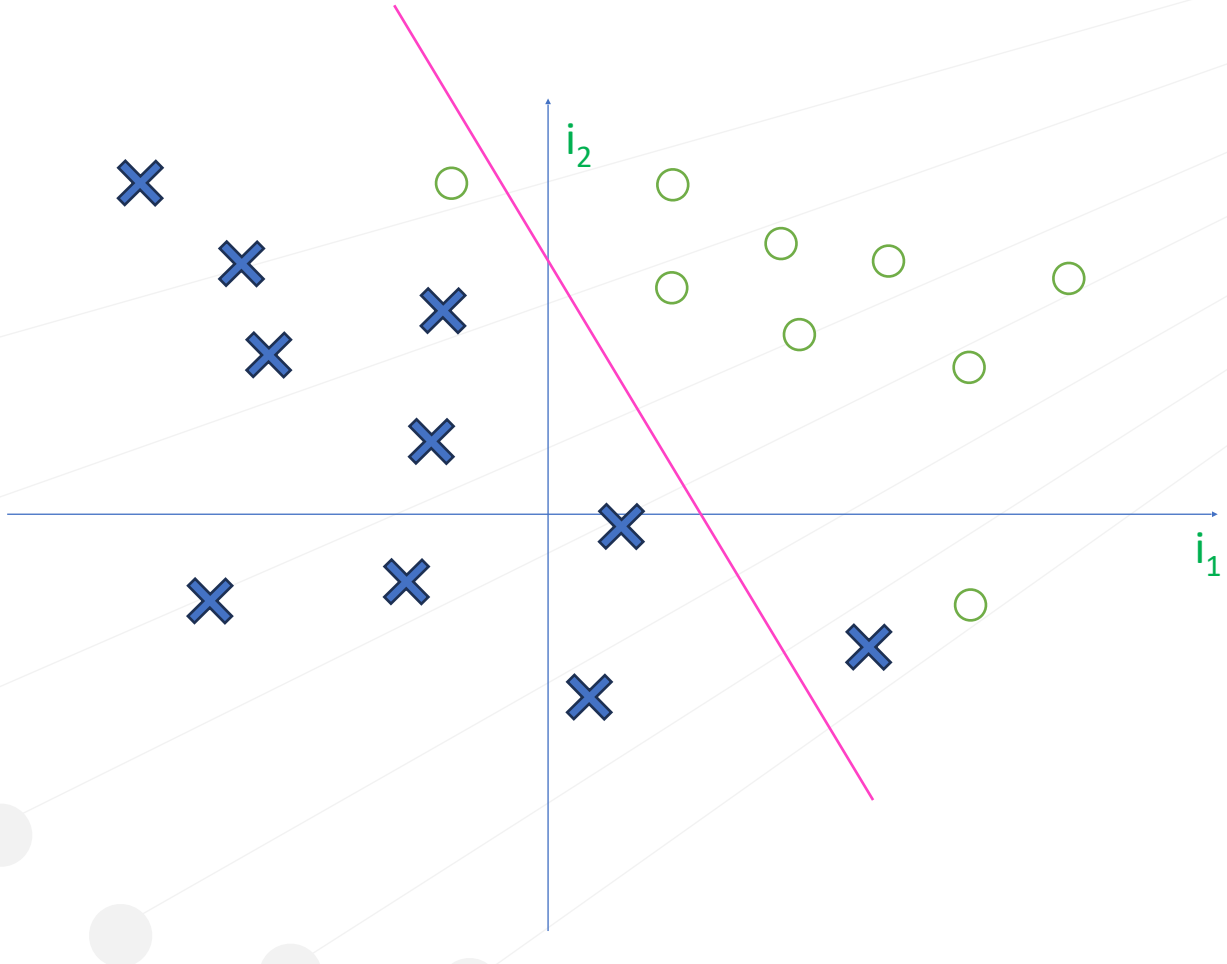
Osservo il fenomeno : creo un insieme di esempi (“Training set”)



i_1 , i_2	decisione
2, 2	○
5, 6	○
-1, 5	○
-2, 0	✕
-3, -1	✕
-2, 6	○
-2, 2	✕

Addestramento

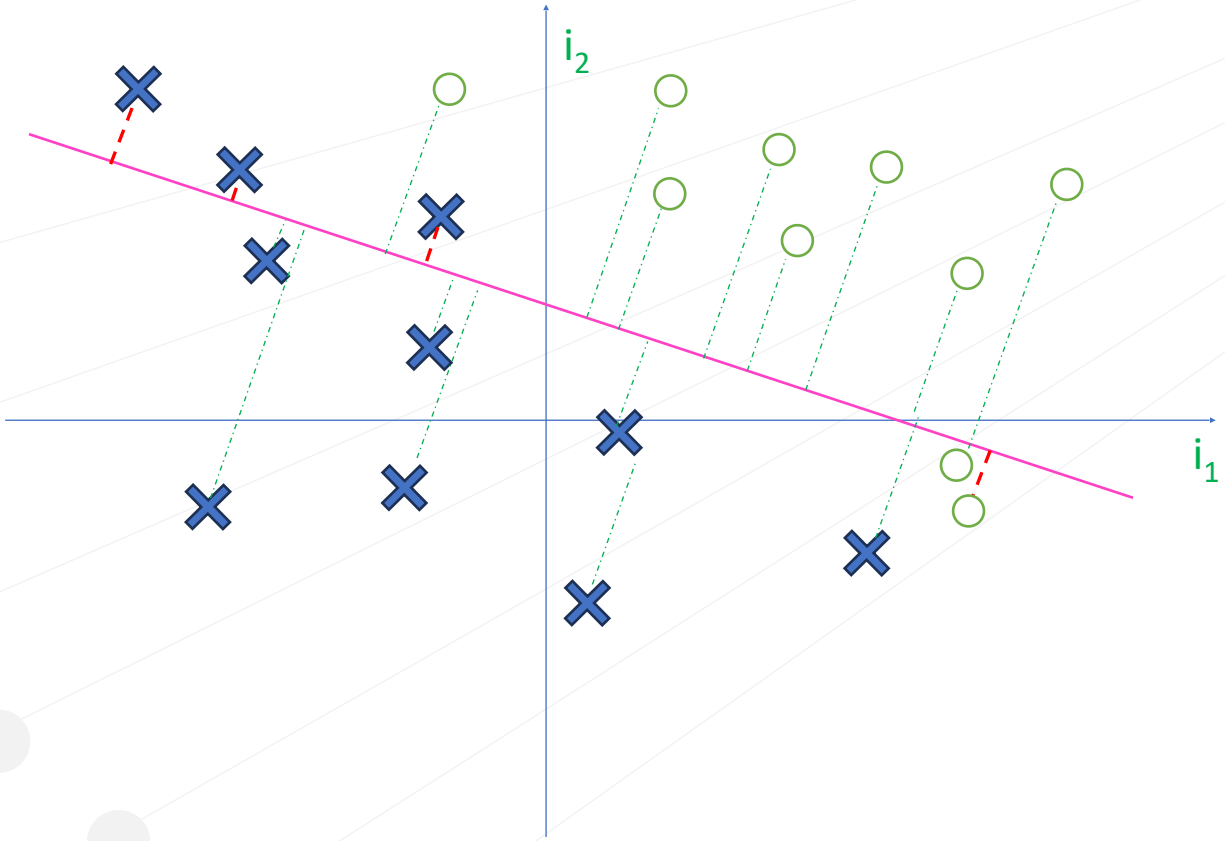
Se vado a tentativi



i_1 , i_2	decisione	
2, 2	○	✓
5, 6	○	✓
-1, 5	○	X
-2, 0	✕	✓
-3, -1	✕	✓
-2, 6	○	✓
4, -2	✕	X

Addestramento

Se vado a tentativi: come verifico la soluzione migliore ?

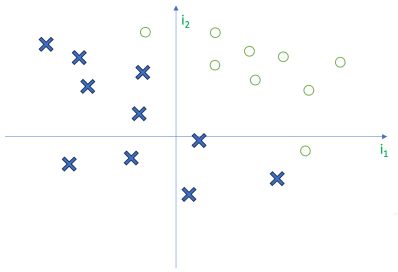


i_1 , i_2	decisione	
2, 2	○	✓
5, 6	○	✓
-1, 5	○	✓
-2, 0	✕	✓
-3, 4	✕	X
5, -1	○	X
4, -2	✕	✓

Addestramento

Cerco i pesi [w_1, …, w_n] che rendono minimo

l'errore tra i risultati del percettrone e il risultato "corretto"

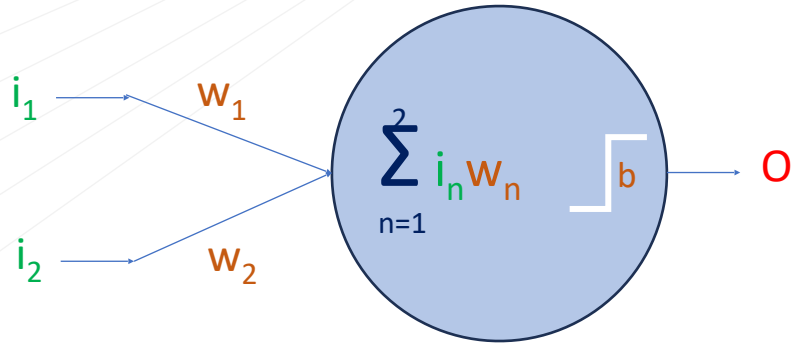


i_1 , i_2	decisione
2, 2	○
5, 6	○
-1, 5	○
-2, 0	✕
-3, -1	✕
-2, 6	○
-2, 2	✕

MIN (Funzione di costo)

$$\text{MIN}\left(\sum_{k=1}^{S}\left(\text{step}_k - T_k\right)^2\right)$$

$$\text{MIN}\left(\sum_{k=1}^{S}\left(\sum_{p=1}^{N} i_{pk} w_p - T_k\right)^2\right)$$



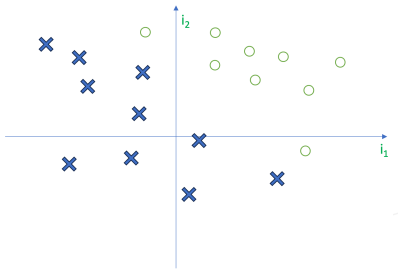
- i_p = input p-esimo
- w_n = peso n-esimo
- E_k = Esempio k-esimo
- i_{pk} = input p-esimo dell'esempio k-esimo
- T_k = Output dell'esempio k-esimo

N = numero di nodi S = numero di esempi

Addestramento

Cerco i pesi $[w_1, w_2]$ che rendono minimo

l'errore tra i risultati del percettrone e il risultato "corretto"



i_1 , i_2	decisione	
2, 1	○	(1)
-3, -1	✕	(0)

$$\text{MIN}\left(\sum_{k=1}^{2} \left(\text{step}_k - T_k \right)^2 \right)$$

$$\text{MIN}\left(\sum_{k=1}^{2} \left(\sum_{p=1}^{2} i_{pk} w_p - T_k \right)^2 \right)$$

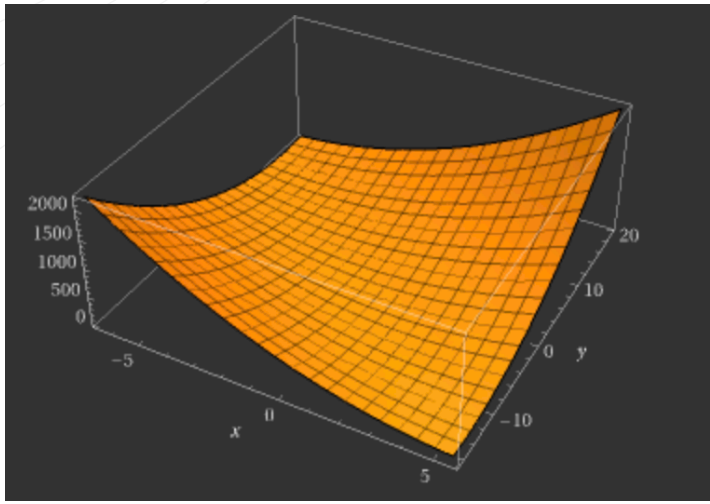
$$\text{MIN}\left((i_{11}w_1 + i_{21}w_2 - T_1)^2 + (i_{12}w_1 + i_{22}w_2 - T_2)^2 \right) =$$

$$\text{MIN}\left((2w_1 + 1w_2 - 1)^2 + (-3w_1 + -1w_2 - 0)^2 \right) =$$

$$\text{MIN}\left((2w_1 + w_2 - 1)^2 + (-3w_1 - 1w_2)^2 \right) =$$

$$\text{MIN}\left((4w_1^2 + w_2^2 + 1 + 4w_1w_2 - 2w_1 - w_2) + (9w_1^2 + w_2^2 + 6w_1w_2) \right) =$$

$$\text{MIN}\left(13w_1^2 + 2w_2^2 + 10w_1w_2 - 2w_1 - w_2 + 1 \right)$$



Addestramento

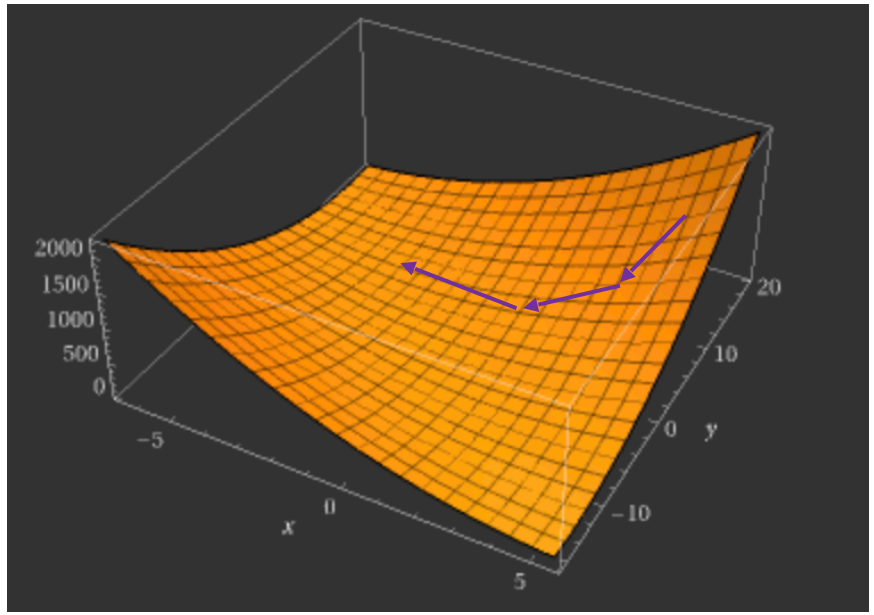
Discesa lungo il gradiente

Si parte da una configurazione arbitraria di $[w_1, w_2]$, tipicamente scelta in modo casuale, denotata con $W^{(0)}$.
I pesi vengono quindi aggiornati iterativamente secondo la formula:

$$W^{(k+1)} = W^{(k)} + \Delta W^{(k)}$$

dove

$$\Delta W^{(k)} = \underbrace{-\ \eta}_{\textit{ampiezza}} \ * \ \underbrace{\text{derivata(Funzione di costo)}}_{\textit{direzione}}$$



Addestramento

Discesa lungo il gradiente - operativamente

$W^{(k+1)} = W^{(k)} + \Delta W^{(k)}$

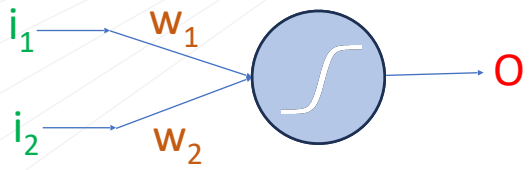
dove

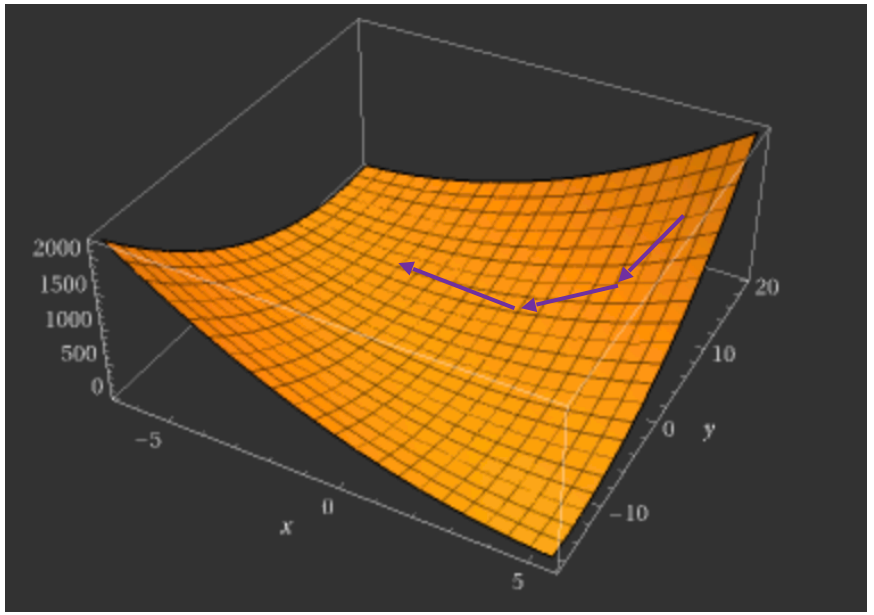
$\Delta W^{(k)} = - \eta * \text{derivata(Funzione di costo)}$

$\Delta w_1^{(k)} = - \eta * (\sigma - T^{(k)}) * i_1^{(k)}$

$\Delta w_2^{(k)} = - \eta * (\sigma - T^{(k)}) * i_2^{(k)}$

errore in uscita all'iterazione k-esima

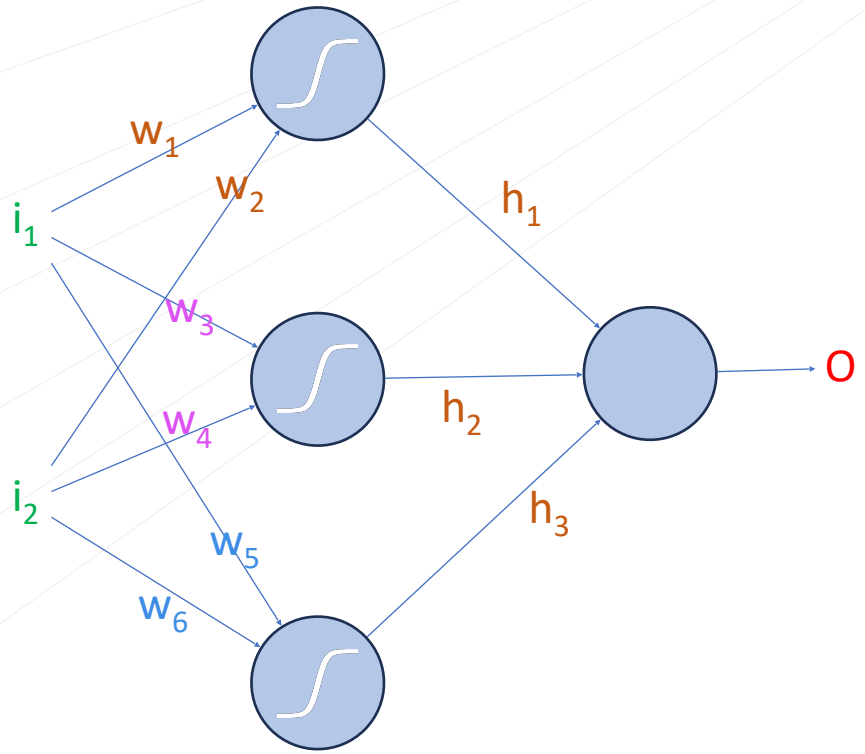




Discesa lungo il gradiente

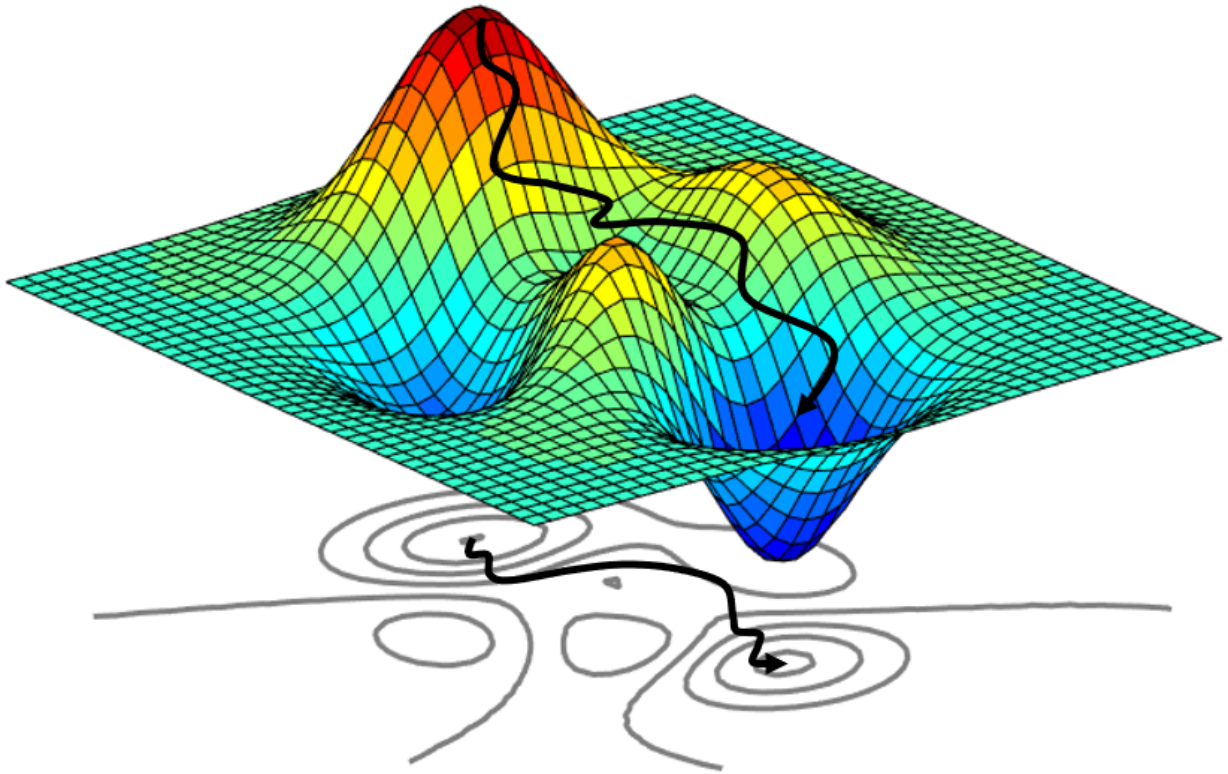
La regola di aggiornamento dei pesi è una regola locale:

- l'algoritmo necessita solamente delle informazioni presenti alle due estremità della connessione;
- questo permette un'alta parallellizzazione del processo di addestramento della rete neurale.

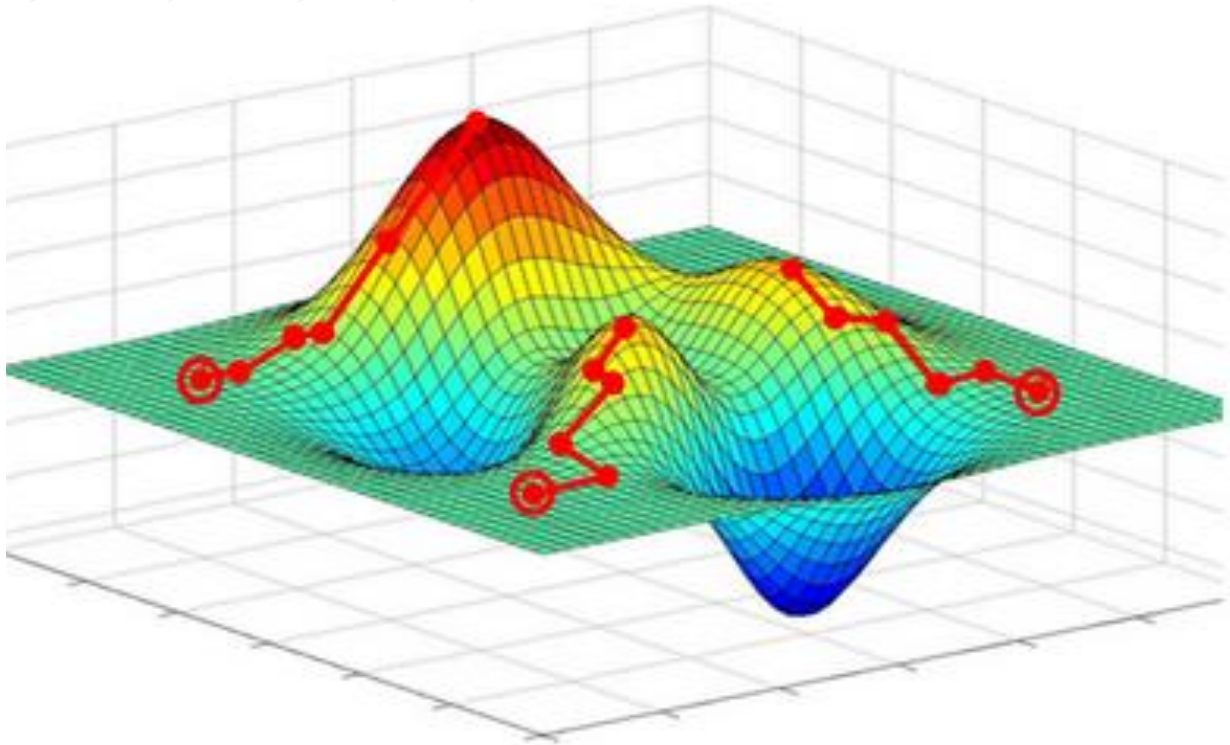


Discesa lungo il gradiente

Cosa fare

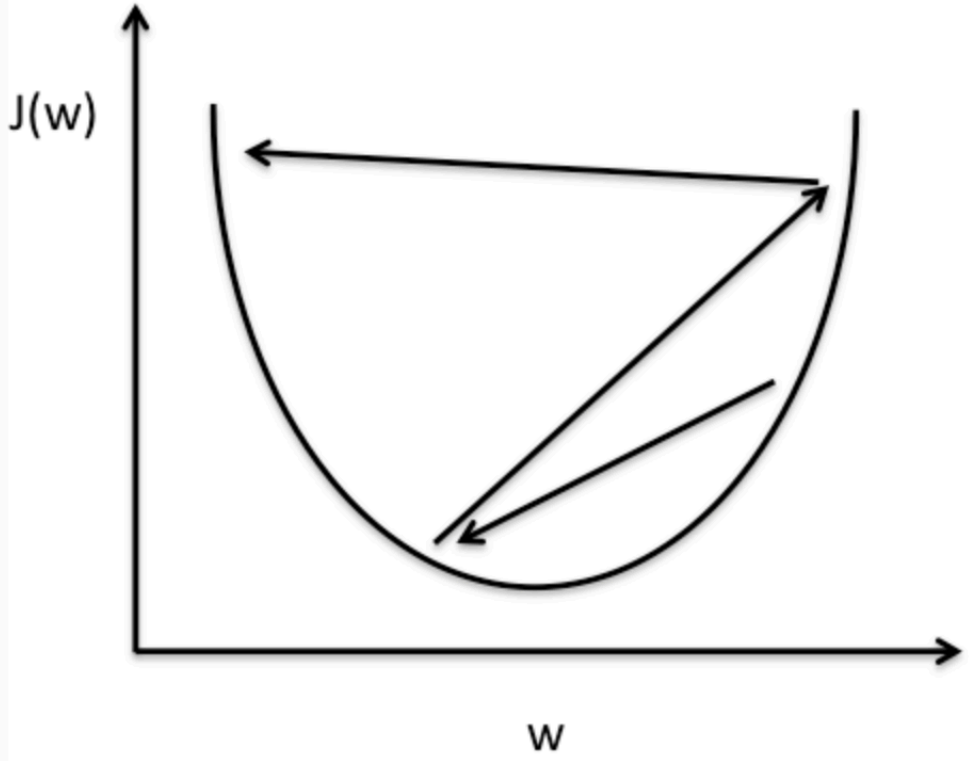


Cosa evitare

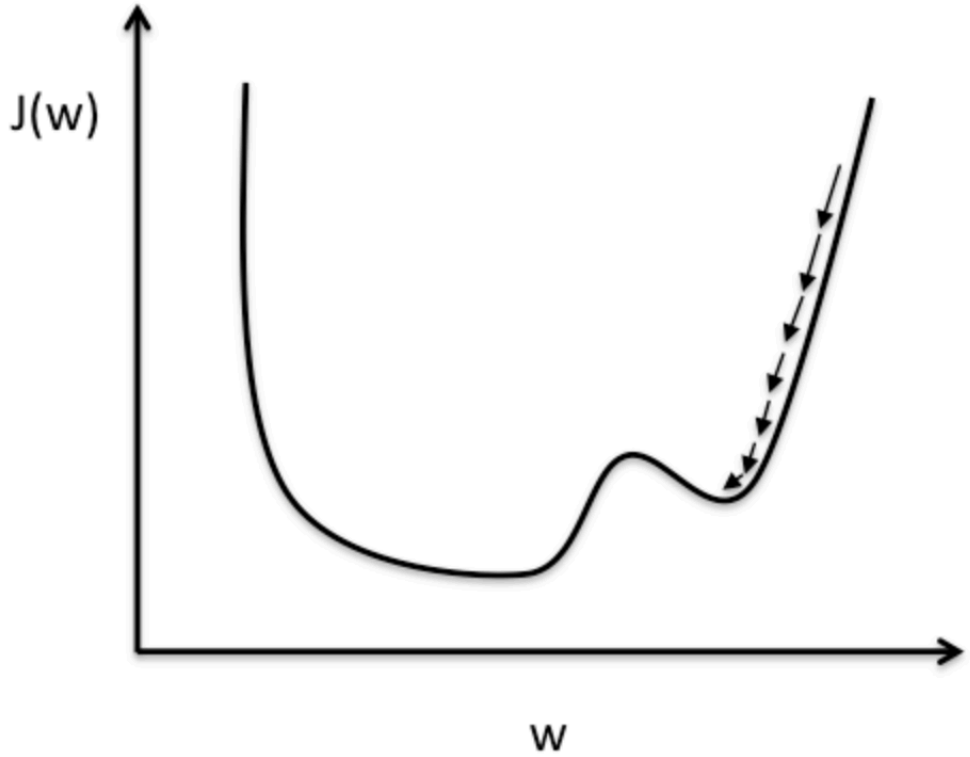


Discesa lungo il gradiente

“Troppa” ampiezza

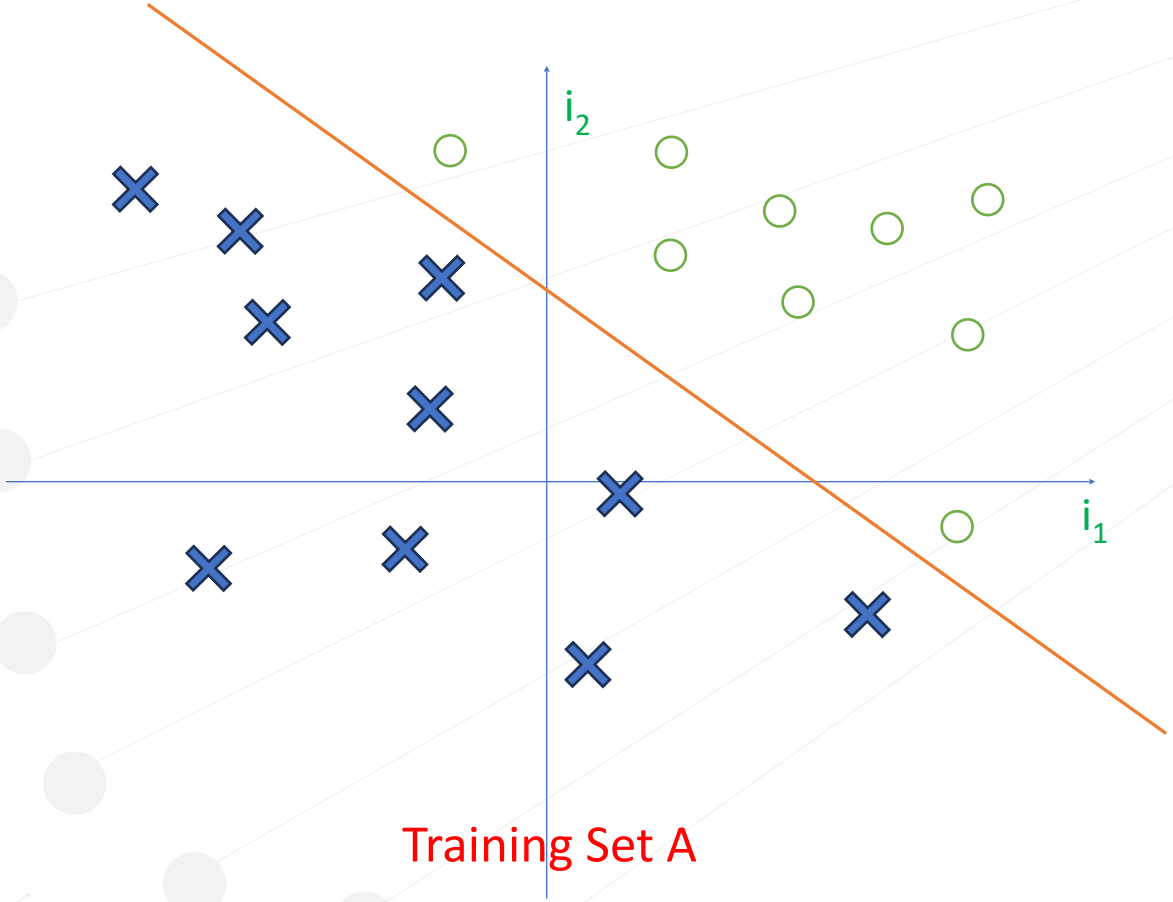


“Poca” ampiezza

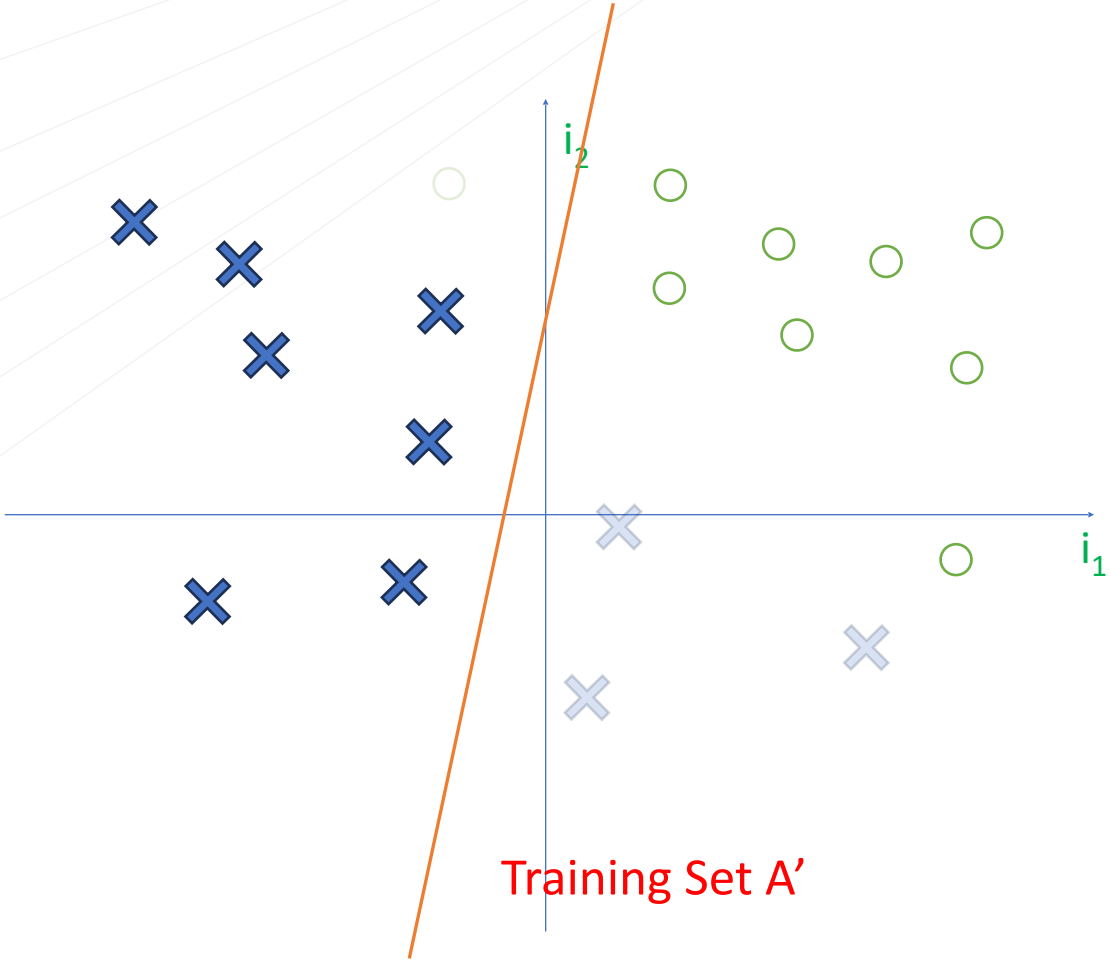


Training set

Ammesso che discendo al meglio il gradiente, cosa ottimizzo ?

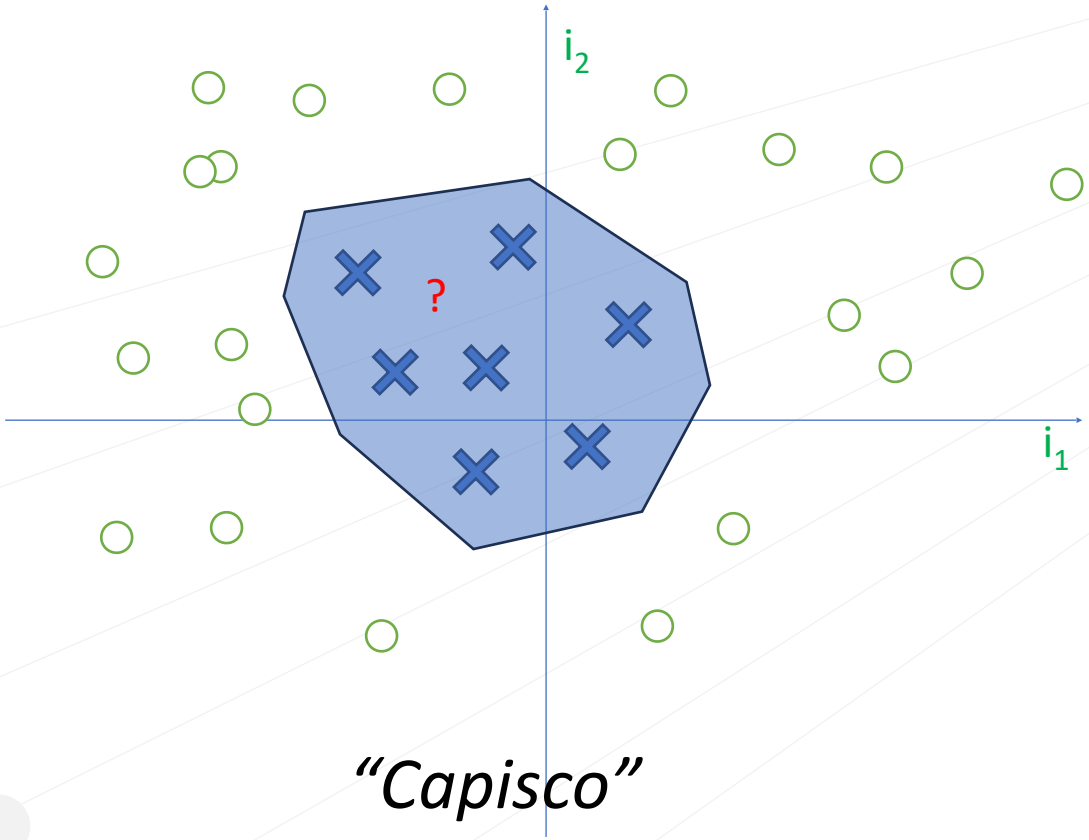


Training Set A

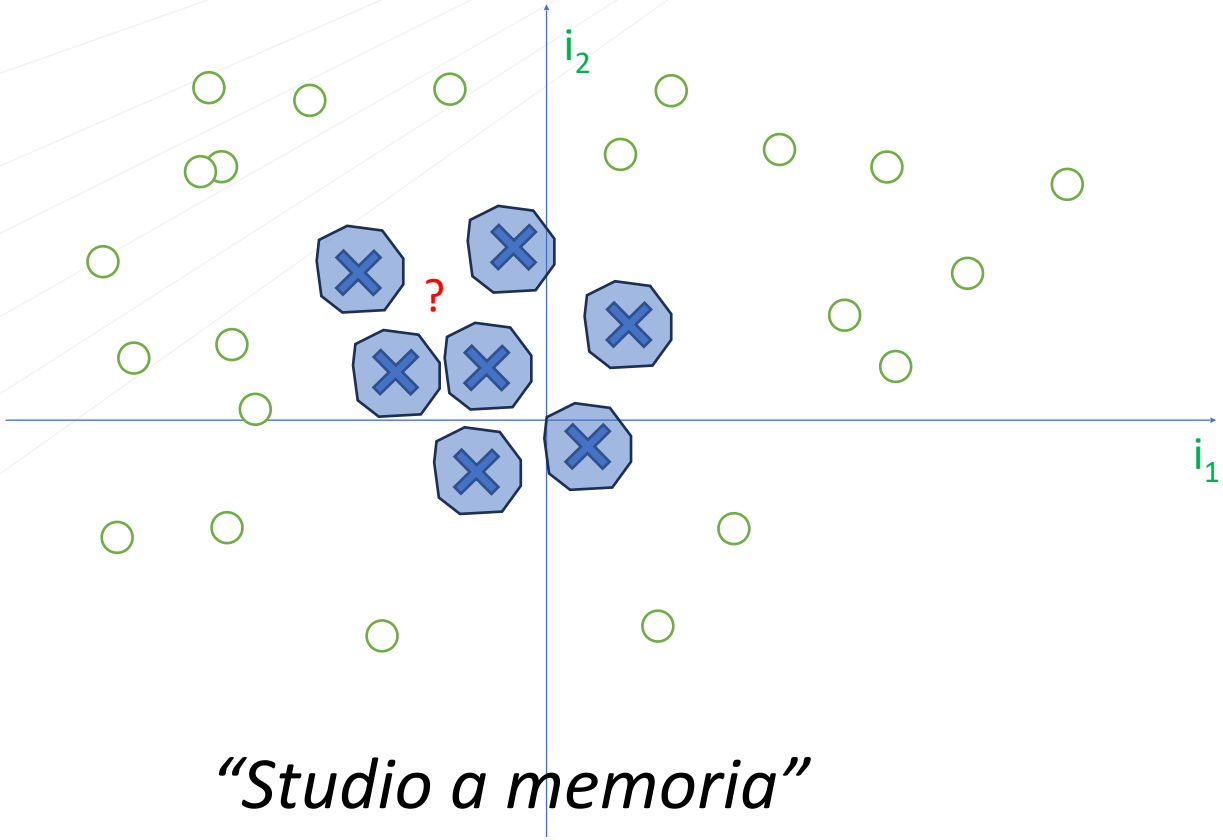


Training Set A’

Overfitting



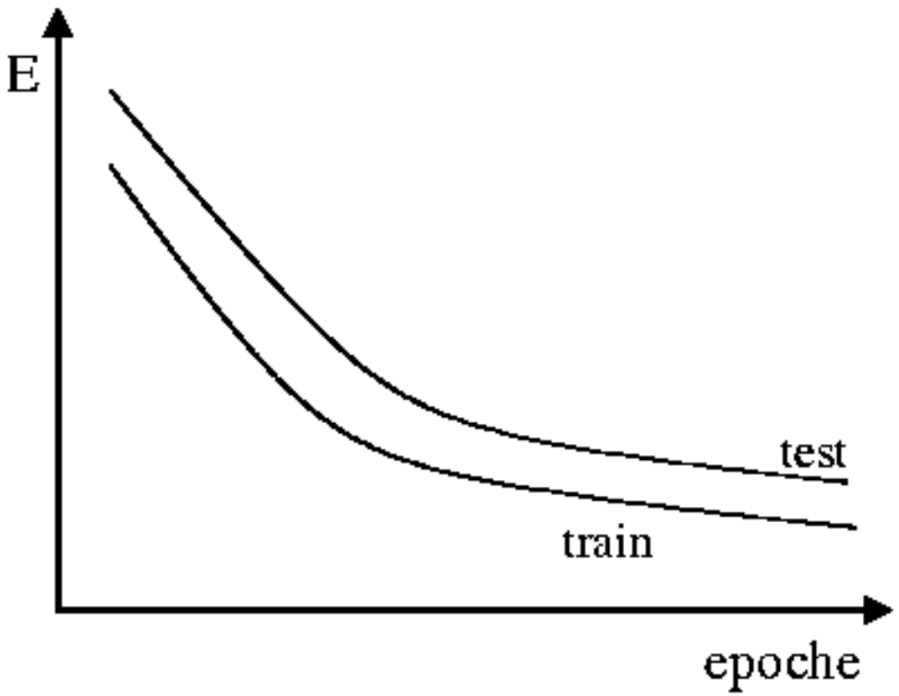
“Capisco”

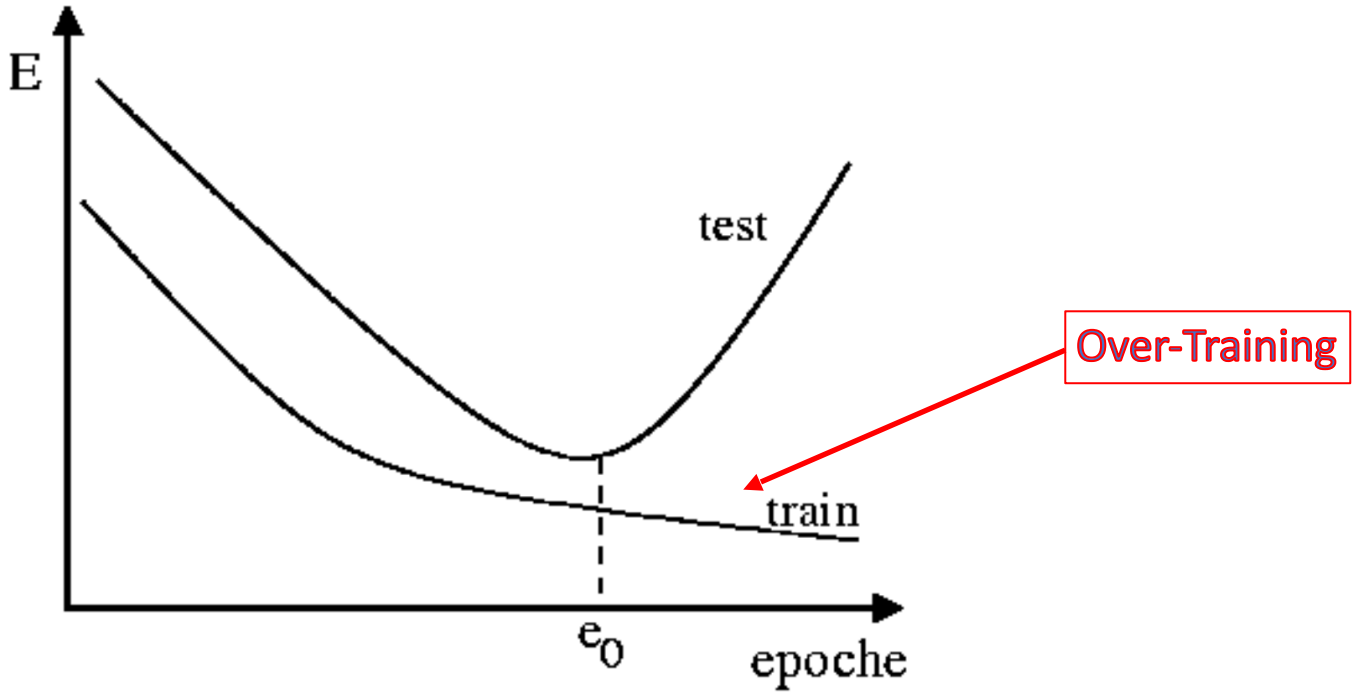


“Studio a memoria”

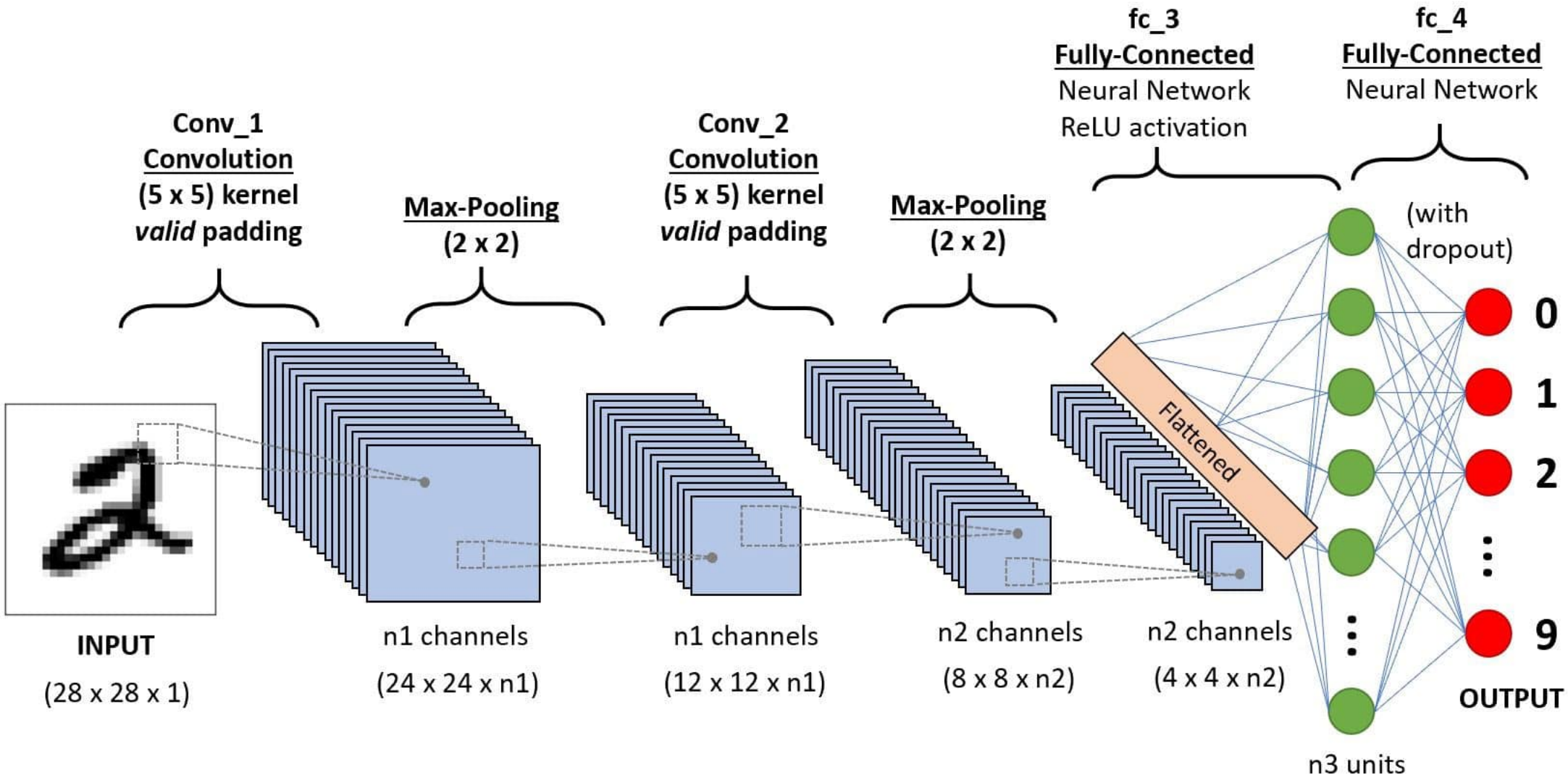
Overfitting

Training test, validation test





Simboli (e non numeri)



Reti neurali non feed forward