Reti Neurali

Reti Neurali in Generale

Bisogna distinguere due motivazioni diverse nello studiare Reti Neurali Artifi ciali:

1. riprodurre (e quindi comprendere) il cervello umano
   - modellare tutto o almeno parti del cervello umano in modo affi dabile
   - riprodurre fedelmente fenomeni neurofi siologici
   - verifi care sperimentalmente se il modello proposto riproduce i dati biologici
2. estrarre i principi fondamentali di calcolo utilizzati dal cervello
   - non importa riprodurre il cervello umano, ma solo evincere quali sono i principi fondamentali di calcolo che esso utilizza
   - semplifi cazioni ed astrazioni del carvello umano sono permesse, anzi, sono lo strumento principale di lavoro
   - produrre un sistema artifi ciale che sia eventualmente diverso dal cervello umano ma che riproduca alcune delle sue funzioni, magari in modo pi`u veloce ed effi ciente (metafora del volo: aereo “contro” uccello)

A noi interessa la seconda motivazione.

I modelli di Reti Neurali Artifi ciali proposti in questo ambito sono molteplici e con scopi diversi.

As esempio, esistono modelli per

- l’apprendimento supervisionato (Classifi cazione, Regressione, Serie Temporali, ...);
- l’apprendimento non supervisionato (Clustering, Data Mining, Self-Organization maps,...);
- realizzare Memorie Associative;
- ...

Tali modelli, in generale, differiscono per

- topologia della rete
- funzione calcolata dal singolo neurone
- algoritmo di apprendimento
- modalità di apprendimento (utilizzo dei dati di apprendimento)

Quando è opportuno utilizzare una Rete Neurale Artificiale ?

- l'input è ad alta dimensionalità (discreto e/o a valori reali)
- l'output è a valori discreti (classificazione) o a valori reali (regressione)
- l'output è costituito da un vettore di valori
- i dati possono contenere rumore
- la forma della funzione target è totalmente sconosciuta
- la soluzione finale non deve essere compresa da un esperto umano ("black-box problem")

Esempi di campi applicativi

- riconoscimento del parlato
- classificazione di immagini
- predizione di serie temporali in finanza
- controllo di processi industriali

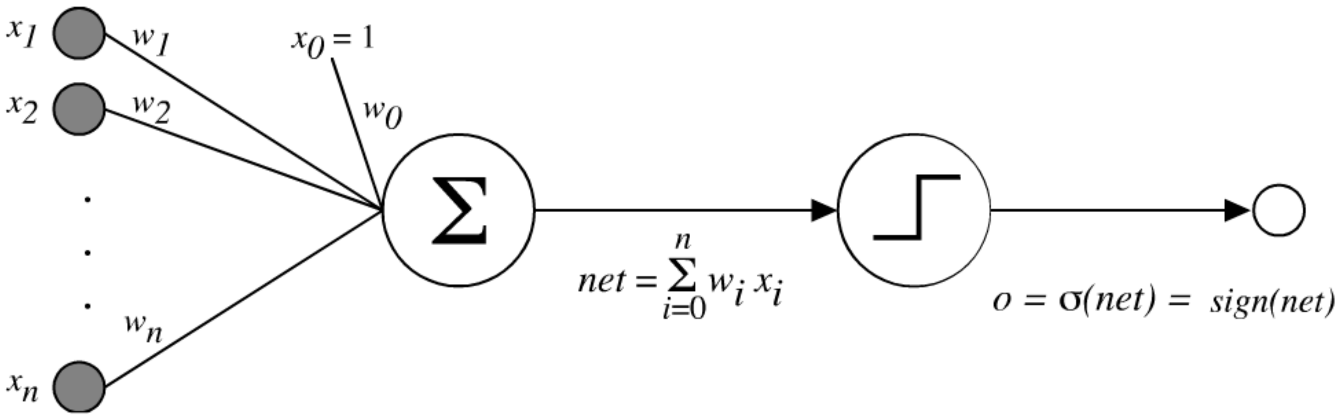
Reti Neurali Artificiali

Le Reti Neurali Artificiali si ispirano al cervello umano:

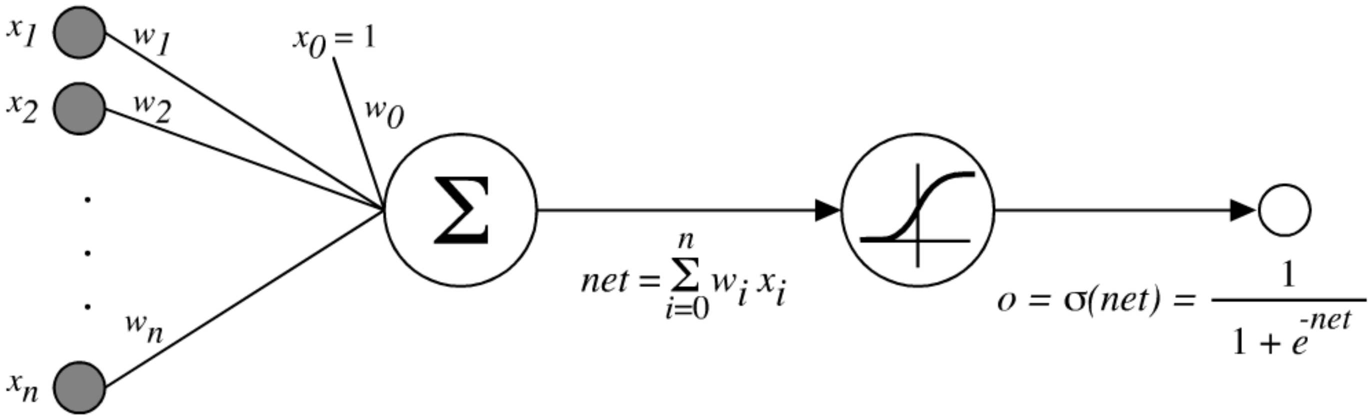
- Il cervello umano è costituito da circa 10^{10} neuroni fortemente interconnessi fra loro;
- Ogni neurone possiede un numero di connessioni che va da circa 10^4 a circa 10^5;
- Il tempo di risposta di un neurone è circa 0.001 secondi;
- Considerando che per riconoscere il contenuto di una scena un umano impiega circa 0.1 secondi, ne consegue che il cervello umano sfrutta pesantemente il calcolo parallelo: infatti, in questo caso, non può effettuare più di 100 calcoli seriali [0.1/0.001=100].

Neurone Artificiale

Alternativa 1: hard-threshold $\rightarrow$ iperpiano!!



Alternativa 2: neurone sigmoidale $\rightarrow$ funzione derivabile

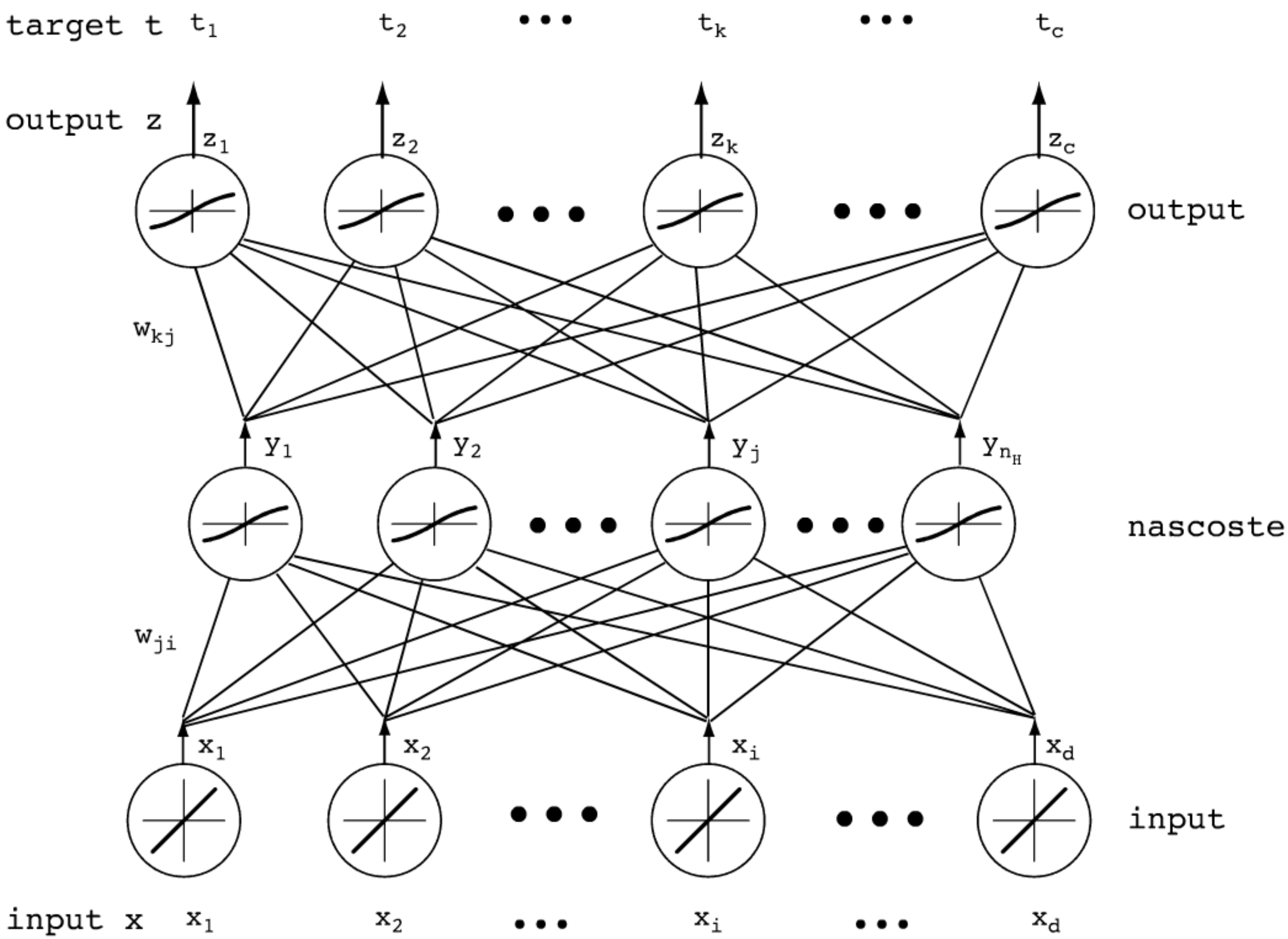


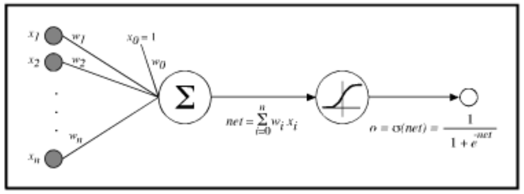
Reti Neurali Artificiali

Una Rete Neurale Artificiale è un sistema costituito da unità interconnesse che calcolano funzioni (numeriche) non-lineari:

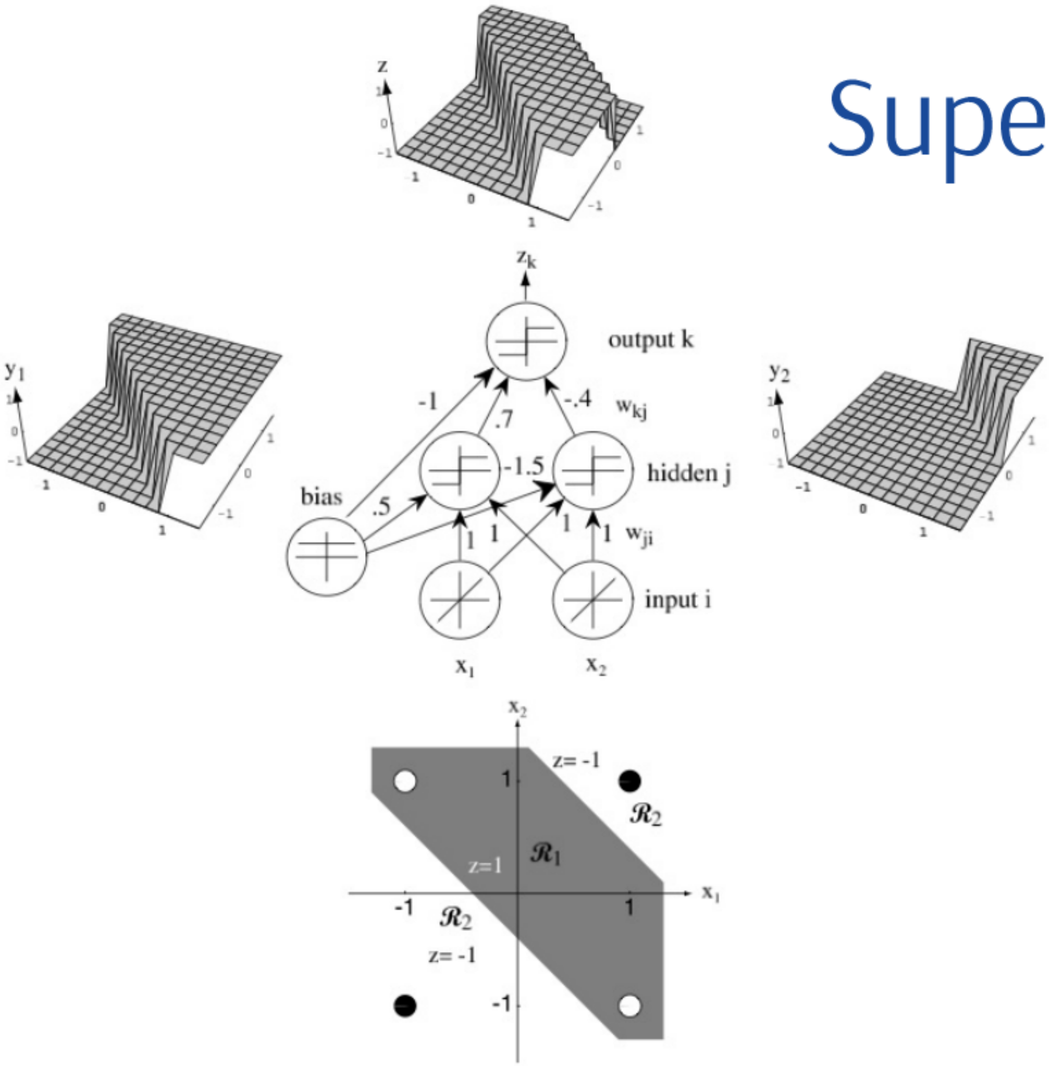
- le unità di *input* rappresentano le variabili di ingresso;
- le unità di *output* rappresentano le variabili di uscita;
- le unità di *nascoste* (se ve ne sono) rappresentano variabili interne che codificano (dopo l'apprendimento) le correlazioni tra le variabili di input relativamente al valore di output che si vuole generare.
- sulle connessioni fra unità sono definiti *pesi* adattabili (dall'algoritmo di apprendimento).

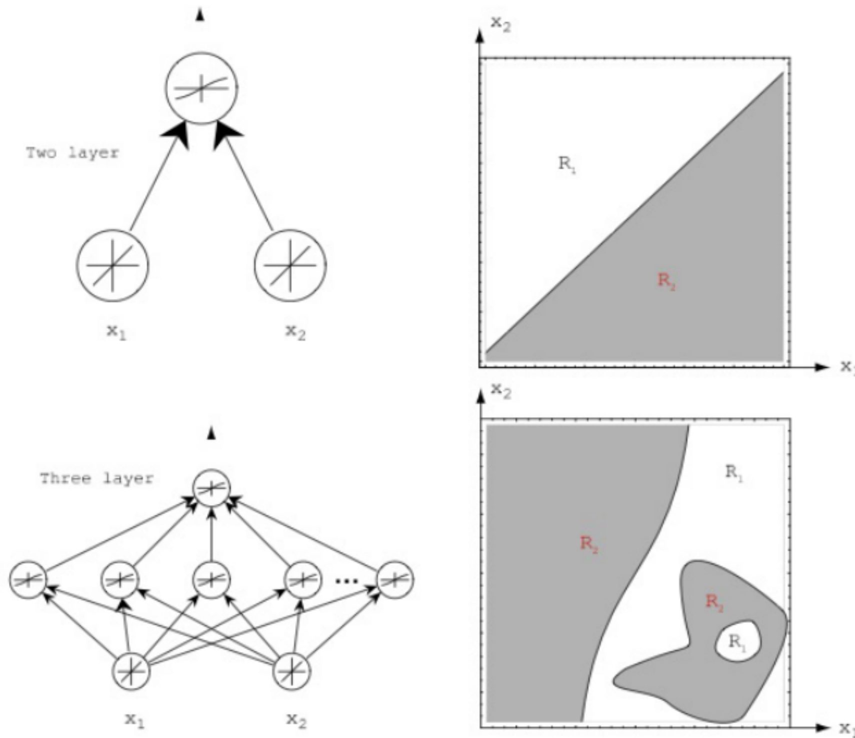
Reti Neurali Feed-forward (un solo livello nascosto)





Superficie di Decisione





Singolo Neurone con Hard-Threshold

Singolo Neurone con Hard-Threshold: iperpiani!

$$\mathcal{H} = \{f_{(\vec{w},b)}(\vec{y}) | f_{(\vec{w},b)}(\vec{y}) = sign(\vec{w} \cdot \vec{y} + b), \vec{w}, \vec{y} \in \mathbb{R}^n, b \in \mathbb{R}\}$$

che possiamo riscrivere come

$$\mathcal{H} = \{f_{(\vec{w}')}(\vec{y}') | f_{(\vec{w}')}(\vec{y}') = sign(\vec{w}' \cdot \vec{y}'), \vec{w}', \vec{y}' \in \mathbb{R}^{n+1}\}$$

se effettuiamo le segenti trasformazioni

$$\vec{w}' = [b, \vec{w}]^t \qquad \vec{y}' = [1, \vec{y}]^t$$

Faremo riferimento a tale neurone (e all'algoritmo di apprendimento associato) come Perceptron

Problemi: Quale è il potere computazionale di un Perceptron ?

Come definire un algoritmo di apprendimento ?

Funzioni Booleane

Consideriamo input binari e funzioni booleane.

Ogni funzione booleana può essere rappresentata tramite gli operatori or,and, not (in effetti, questo non è necessario se usiamo il nand o il nor)

Può un Perceptron implementare l'operatore or ? Si !

Es. $\vec{y}' \in \{0,1\}^{n+1}, w_0' = -0.5, w_i' = 1,\ i = 1,\ldots,n$

Può un Perceptron implementare l'operatore and ? Si !

Es. $\vec{y}' \in \{0,1\}^{n+1}, w_0' = -n + 0.5, w_i' = 1,\ i = 1,\ldots,n$

L'operatore not si realizza banalmente con un Perceptron con una singola connessione (fare per esercizio).

Esiste una funzione booleana semplice che non può essere realizzata da un Perceptron ?

(l'abbiamo già vista quando abbiamo calcolato la VC-dimension di $\mathcal{H}$!!)

Apprendimento di funzioni linearmente separabili

Assumiamo di avere esempi di funzioni linearmente separabili.

Algoritmo di Apprendimento per il Perceptron

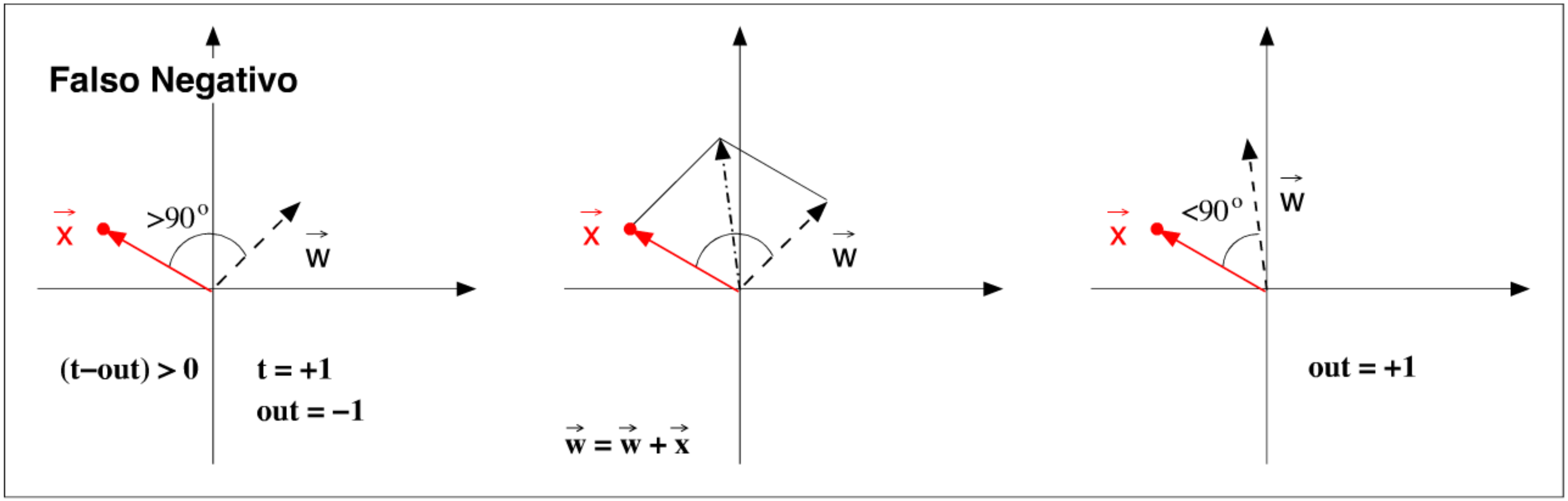
ingresso: insieme di apprendimento $Tr = \{(\vec{x}, t)\}$, dove $t \in \{-1, +1\}$

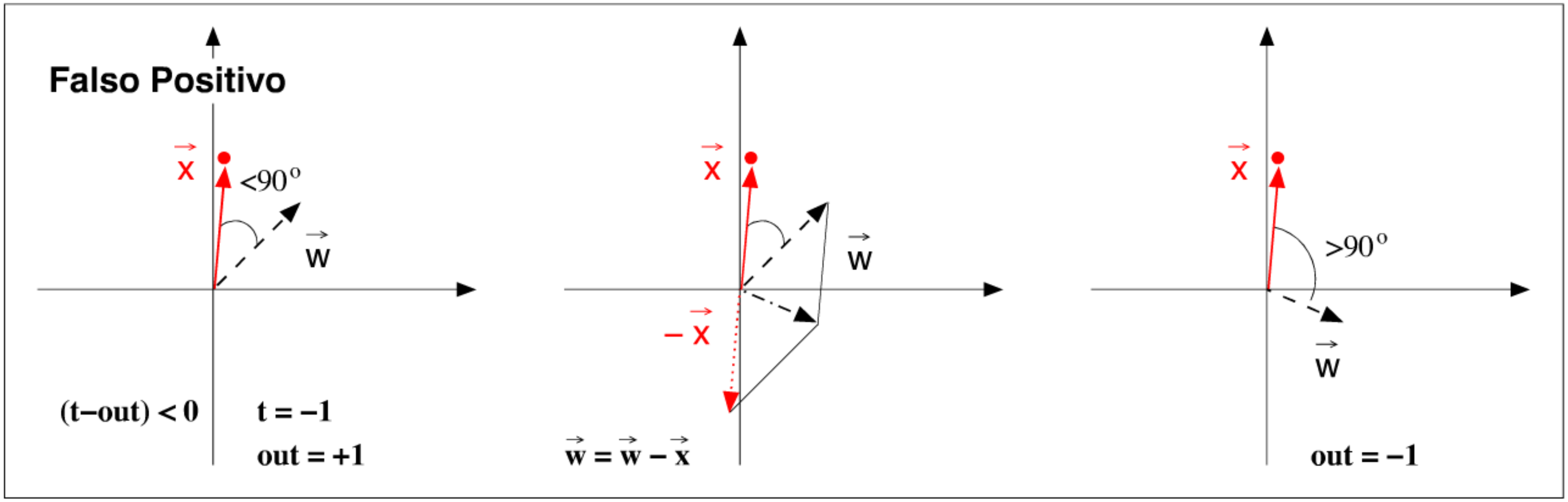
1. inizializza il vettore dei pesi $\vec{w}$ al vettore nullo (tutte le componenti a 0);
2. ripeti
   (a) seleziona (a caso) uno degli esempi di apprendimento $(\vec{x}, t)$
   (b) se $out = sign(\vec{w} \cdot \vec{x}) \neq t$ allora

$$\vec{w} \leftarrow \vec{w} + (t - out)\vec{x}$$

Apprendimento per Perceptron

Interpretazione geometrica





Apprendimento per Perceptron

Non necessariamente un singolo passo di apprendimento 2(b) riuscirà a modifi care il segno dell'output: potrebbero servire molti passi.

Per rendere pi`u stabile l'apprendimento si aggiunge un coeffi ciente di apprendimento η

$$\vec{w} \leftarrow \vec{w} + \eta(t - out)\vec{x}$$

con $\eta > 0$ e preferibilmente $\eta < 1$

Questo evita che il vettore dei pesi subisca variazioni troppo "violente" ogni volta che il passo 2(b) viene eseguito, e quindi cerca di evitare che esempi precedentemente ben classifi cati diventino classifi cati erroneamente a causa della forte variazione del vettore dei pesi.

Se l'insieme di apprendimento è LINEARMENTE SEPARABILE, si dimostra che l'algoritmo di apprendimento per il Perceptron termina con una soluzione in un numero fi nito di passi, altrimenti $\vec{w}$ CICLA in un insieme di vettori peso non necessariamente ottimali (cioè che commettono il minimo numero possibile di errori)