

Settimana 7 – Apprendimento non supervisionato e supervisionato



Apprendimento

- Agenti che possono migliorare il loro comportamento mediante lo studio dell'esperienza passata e predizioni del futuro.

- machine learning: un computer osserva dei dati, costruisce un modello basandosi su quei dati sia come ipotesi circa il mondo osservato che come un software che può risolvere problemi.

- Noi abbiamo sviluppato il ML perché non è possibile sviluppare software deterministico per affrontare i problemi che un agente intelligente deve affrontare nel mondo reale.

Definizioni di Apprendimento Automatico

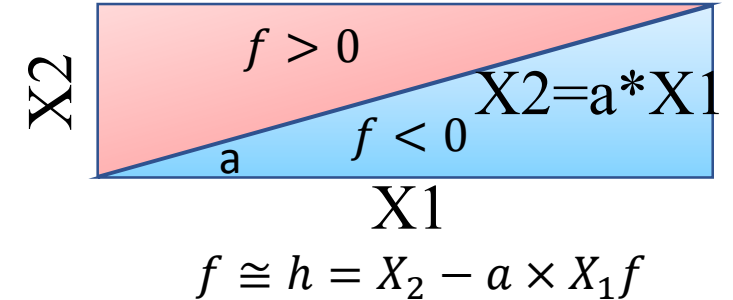
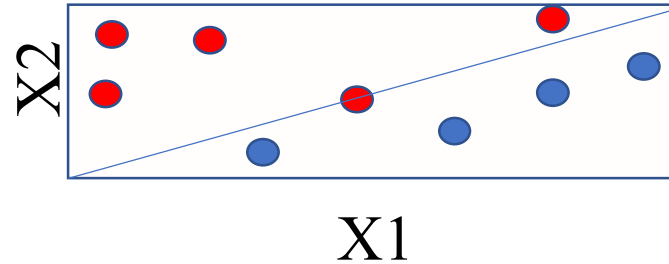
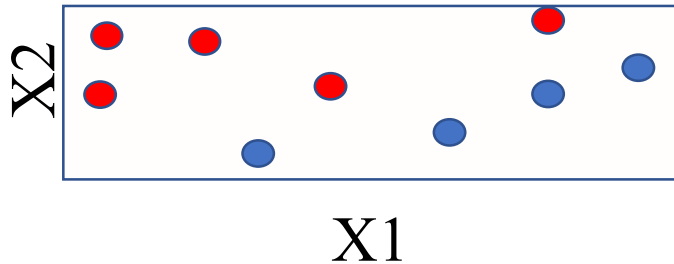
- Apprendere = migliorare la capacità di esecuzione di un certo compito, attraverso l'esperienza
 - Migliorare nel task T
 - Rispetto ad una misura di prestazione P
 - Basandosi sull'esperienza E
- E = esempi di comportamenti “positivi” o “negativi” forniti da un istruttore, oppure un sistema di “ricompense”.
- In A.I. : Tecniche che consentono ad agenti (o sistemi automatici) di migliorare il proprio comportamento attraverso l'analisi delle proprie esperienze.

Scelte nella definizione di un sistema di apprendimento

1. ***Componenti Migliorabili***: decidere quali sono le componenti sulle quali “lavorare”
2. *Scelta e Rappresentazione* di una “***funzione obiettivo***” che rappresenti il compito
3. Scelta di un ***algoritmo di apprendimento***
4. ***Modalità di training***: supervisionato, guidato da obiettivi

funzione obiettivo

- Funzione obiettivo: espressione formale della conoscenza appresa. Viene usata per determinare le prestazioni del programma di apprendimento.
 - Esempi di funzioni obiettivo: polinomi, alberi di decisione, reti neurali, insiemi di regole....



funzione obiettivo e funzione appresa

- Nota: Difficilmente un sistema riesce ad apprendere perfettamente una funzione obiettivo f (l'esempio della slide precedente è un caso tipico).
- In genere viene appresa una funzione, avente la forma prescelta (polinomio, regole,..), che rappresenta una stima, o IPOTESI (indicata con h), per la funzione obiettivo.
- L'obiettivo è di apprendere una h che approssimi f “il meglio possibile”.

algoritmo di apprendimento

- Scelta dell'algoritmo e della funzione obiettivo ovviamente correlate
- Metodi statistici
 - Bayesian learning
 - Markov models
- Metodi algebrici
 - Gradient descent
 - Support Vector Machines
- Metodi knowledge-based
 - Alberi di decisione
 - Regole di associazione

Modalità di training

- Basato sull'informazione a disposizione:
 - Supervisionato (esempi disponibili)
 - Con rinforzo (premi a fronte di comportamenti corretti)
 - Non supervisionato (senza esempi)
- Basato sul ruolo dell'apprendista (learner)
 - Apprendimento passivo (apprende da esempi a-priori)
 - Apprendimento attivo (apprende anche durante il funzionamento)

Training Supervisionato

- Si ha a disposizione un insieme di esempi classificati
- $(X,o): \langle x_1, x_2, \dots, x_n, o \rangle$
- dove vi sono valori delle variabili di ingresso x_i , ed o è l'uscita. Pendono anche il nome di attributi o features. Implica l'esistenza di un istruttore che conosce la risposta corretta;
Si apprende una funzione obiettivo
- $h: X \rightarrow \tilde{o}$
- La misura della prestazione consiste nel minimizzare l'errore di approssimazione della funzione obiettivo sugli esempi a disposizione

$$\min(e) = \min|h - f|$$

Training Con rinforzo

- L'apprendista interagisce con l'ambiente e riceve una ricompensa (numerica) positiva o negativa (es. se un robot che fa goal il “peso” della sequenza di azioni che lo ha portato a fare goal viene aumentato)
- Cosa si apprende: una strategia di comportamento (un “piano” o sequenza di azioni)
- Misura della prestazione: si cerca di massimizzare “a lungo termine” la ricompensa complessivamente ottenuta.
- Percepisco la valutazione ma non conosco l'azione

Training Non supervisionato

- Si percepisce solo l'evoluzione delle percezioni e non la bontà delle azioni.
- L'esperienza di apprendimento è rappresentata da dati non classificati
(ad esempio, scontrini fiscali, cartelle cliniche, pagine web).
- Cosa si apprende: regole associative fra i dati.

Training attivo e passivo

- Apprendimento passivo:
 - L'apprendista può apprendere solo dai dati che vengono messi a disposizione (E)
- Apprendimento attivo:
 - L'apprendista può fare domande ed esperimenti

Apprendimento supervisionato



Apprendimento supervisionato

- Vogliamo che il nostro agente AI sia in grado, dato un insieme di informazioni di allenamento composto da dati di ingresso X_a e risposte corrette Y_a per un certo problema applicativo, di apprendere la regola sottostante i dati di ingresso e uscita che gli forniamo.

Apprendimento supervisionato

- In particolare, il nostro agente AI deve essere in grado di «imparare» dall'insieme di dati di allenamento e cercare/costruire in qualche modo una funzione ipotesi degli ingressi $h(x)$ che per ogni x_a dia un risultato pari a y_a o molto simile.

$$y_a \cong h(x_a)$$

Apprendimento supervisionato

- Vediamo un caso univariato.
- Immaginiamo che abbiamo a disposizione i dati di temperatura corporea di un certo numero di pazienti e che per gli stessi pazienti si sia visto che alcuni hanno una certa patologia P e altri no;
- Quindi la nostra AI dovrebbe imparare dai dati disponibili e trovare una funzione che data la temperatura corporea di un nuovo paziente stabilisca se il paziente ha o meno la patologia P

Apprendimento supervisionato

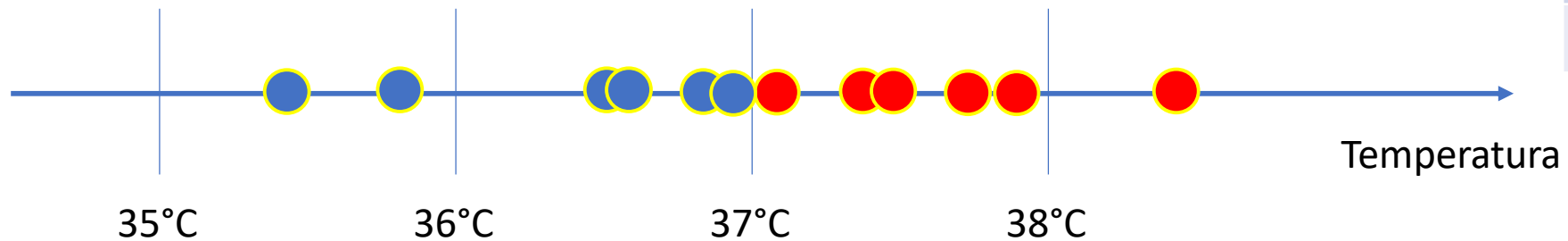
- Ad esempio i dati disponibili potrebbero essere questi.
- X_a è la temperatura corporea
- Y_a è la patologia P che è uguale a 0 in assenza di patologia oppure 1 in presenza di patologia

X_a	Y_a
35,5	0
35,8	0
36,5	0
36,6	0
36,8	0
36,9	0
37,1	1
37,4	1
37,5	1
37,6	1
37,8	1
38,6	1

I dati sono riportati nel file temperature.csv

Apprendimento supervisionato

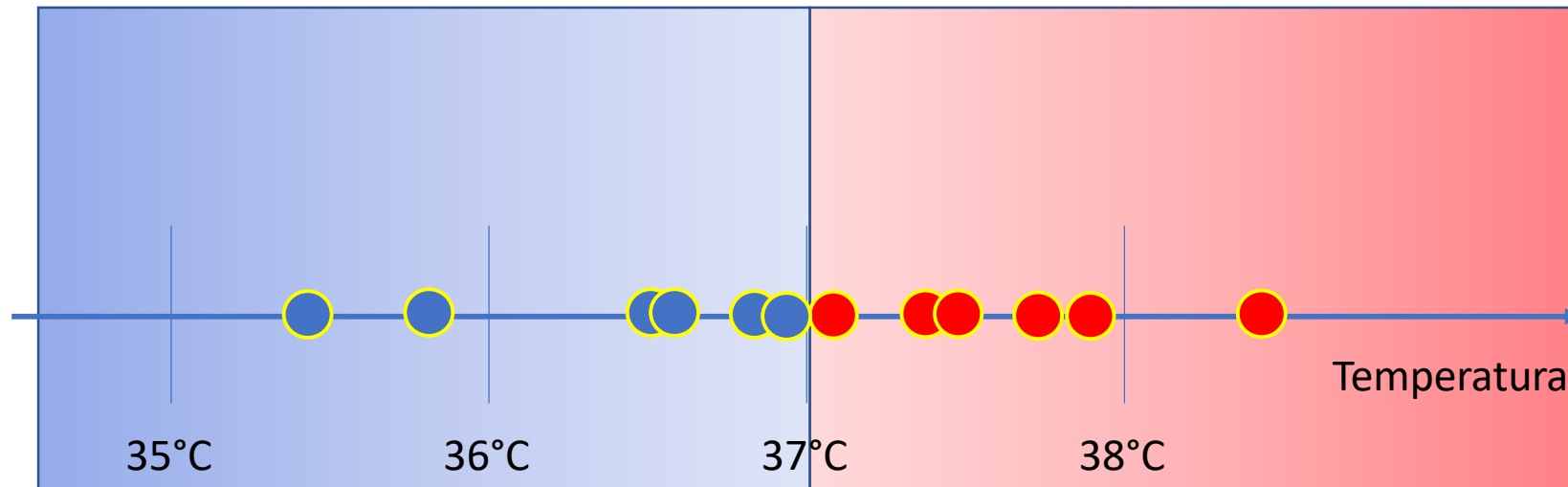
- Possiamo fare un grafico unidimensionale dei nostri dati;
- Rappresentiamo i pazienti con patologia P li rappresentiamo con un pallino rosso e quelli che non manifestano la patologia li rappresentiamo in blu.



X_a	Y_a
35,5	0
35,8	0
36,5	0
36,6	0
36,8	0
36,9	0
37,1	1
37,4	1
37,5	1
37,6	1
37,8	1
38,6	1

Apprendimento supervisionato

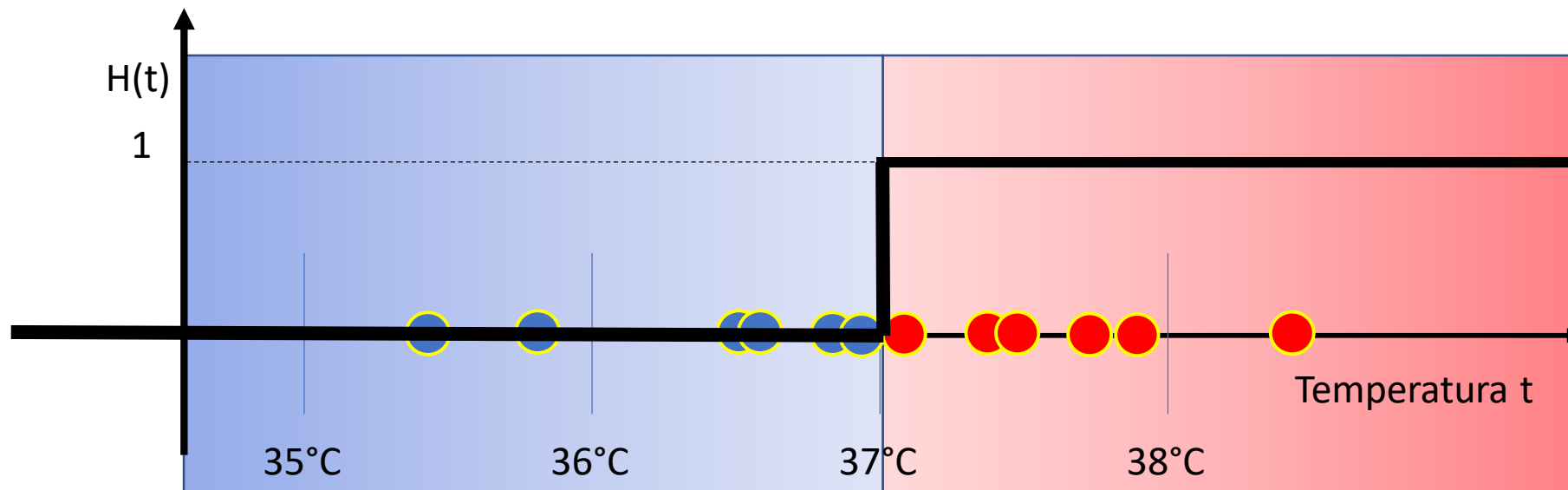
- In questo caso molto semplice basta osservare i dati per scoprire che tutti i pazienti con patologia $P=1$ hanno una temperatura corporea maggiore di 37°C



X_a	Y_a
35,5	0
35,8	0
36,5	0
36,6	0
36,8	0
36,9	0
37,1	1
37,4	1
37,5	1
37,6	1
37,8	1
38,6	1

Apprendimento supervisionato

- Quindi la funzione $h(x)$ che cerchiamo è una funzione pari a 0 per $t \leq 37$ e pari a 1 per $t > 37$
- Questa funzione si chiama gradino con soglia pari a 37



X_a	Y_a
35,5	0
35,8	0
36,5	0
36,6	0
36,8	0
36,9	0
37,1	1
37,4	1
37,5	1
37,6	1
37,8	1
38,6	1

Apprendimento multivariato

Apprendimento supervisionato

- Nearest-Neighbor Classification
- K-Nearest-Neighbor Classification
- Perceptrons
- Neural Networks

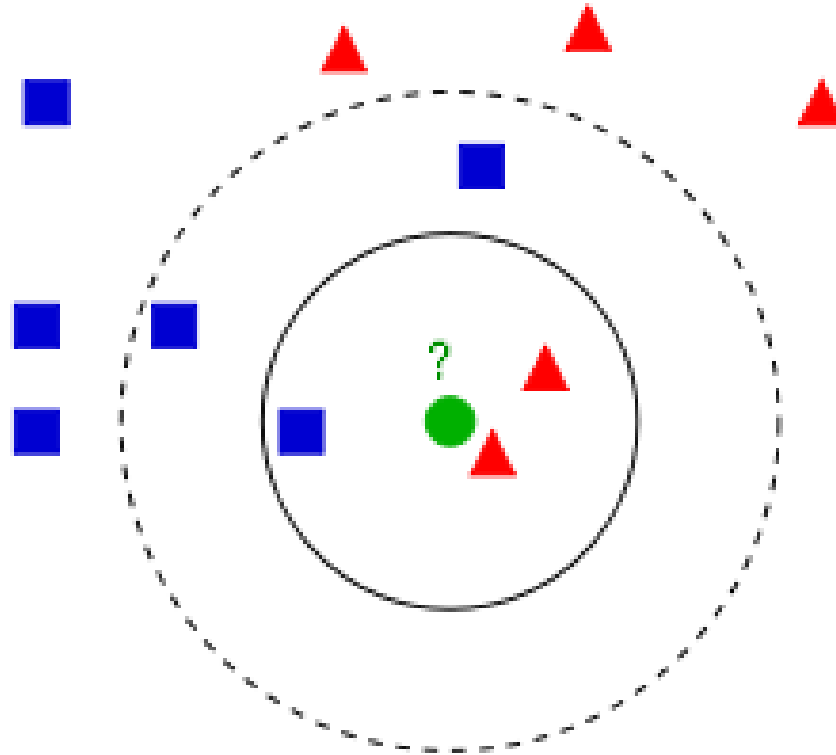
Nearest-Neighbor Classification



Nearest-Neighbor Classification

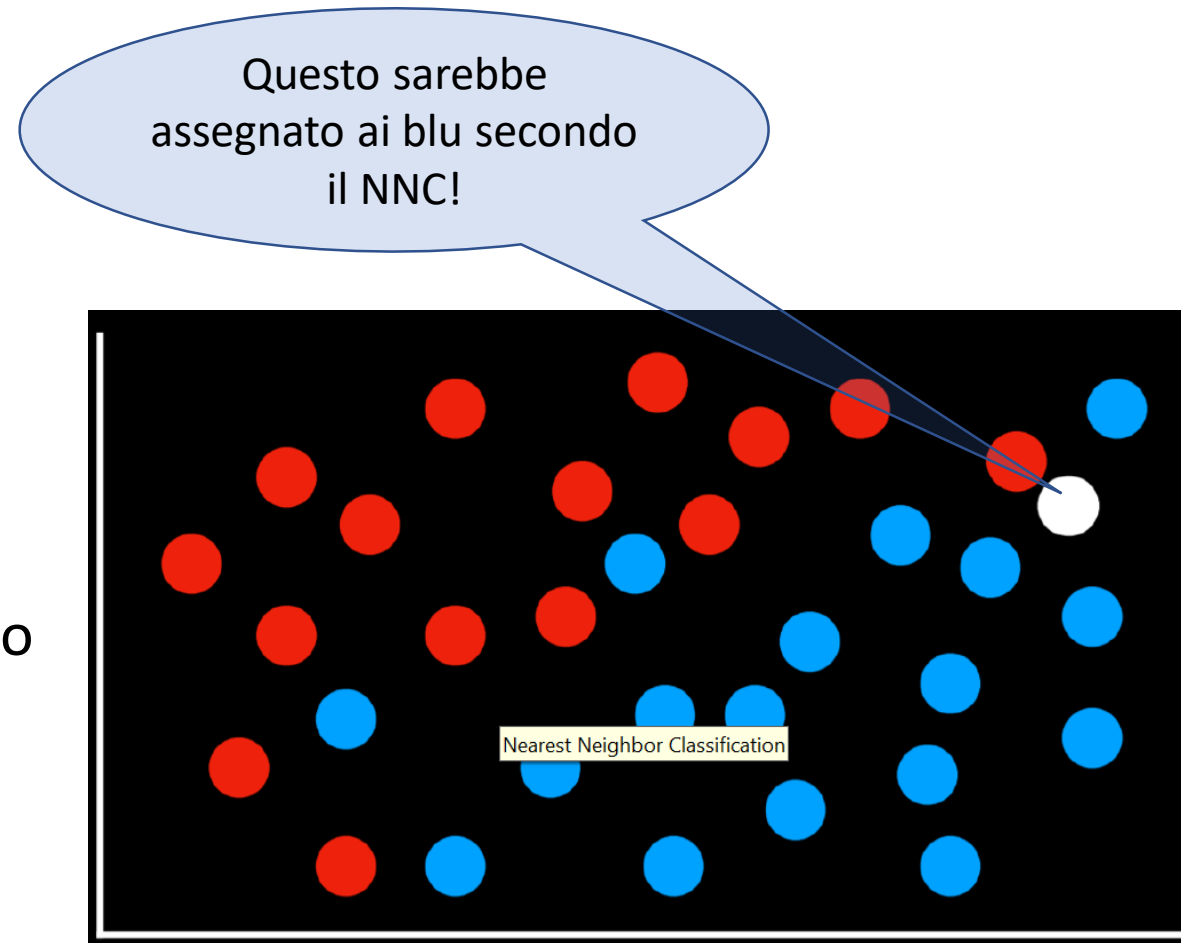
- Un approccio multivariato all'apprendimento supervisionato è il Nearest-Neighbor Classification che funziona così:
- Parto da un dataset di apprendimento con dei dati già classificati, ad esempio, a due categorie A e B.
- Un nuovo dato multidimensionale in arrivo lo classifico come A o B a seconda se si trova più vicino a uno dei punti di A o di B

Nearest-Neighbor Classification



Nearest-Neighbor Classification

- Vantaggi:
 - Semplice implementazione anche se molto pesante.
- Svantaggi:
 - Non sempre funziona bene!

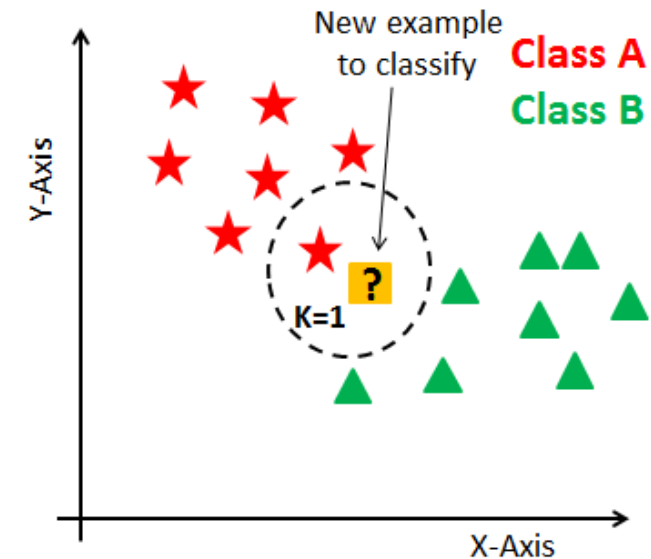


K-Nearest-Neighbor Classification

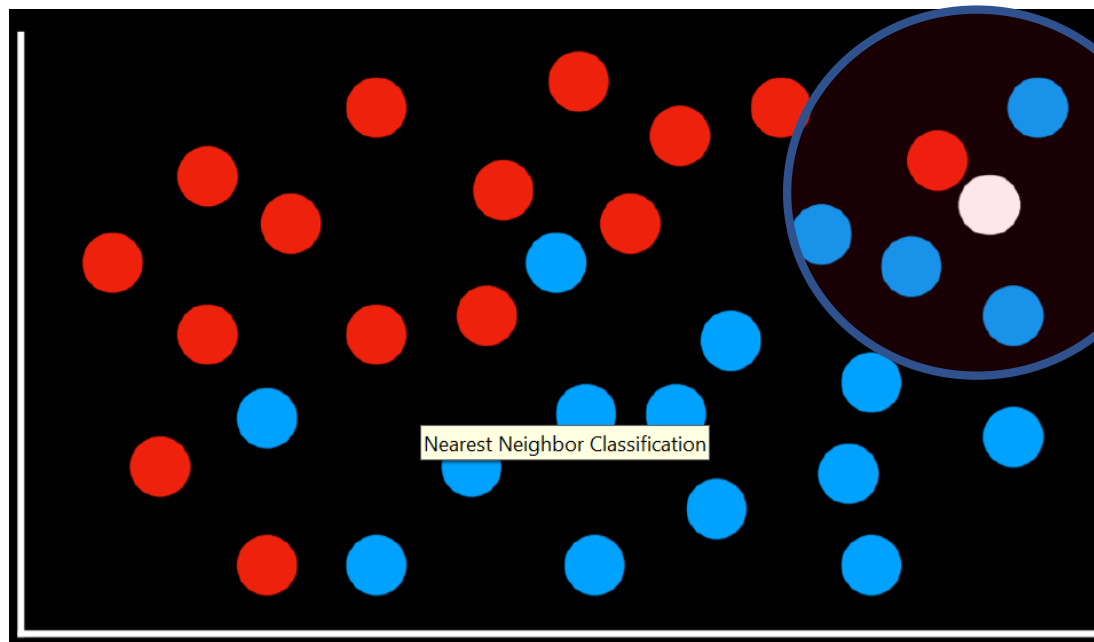


K-Nearest-Neighbor Classification

- Invece di assegnare il nuovo campione alla classe dell'elemento più vicino lo assegno alla classe degli elementi più numerosi tra i k più vicini;
- Quindi il Nearest-Neighbor Classification sarebbe il caso con $k=1$



K-Nearest-Neighbor Classification



In questo caso con $k > 1$ possiamo fare una scelta migliore circa la classe di appartenenza del nuovo arrivato.

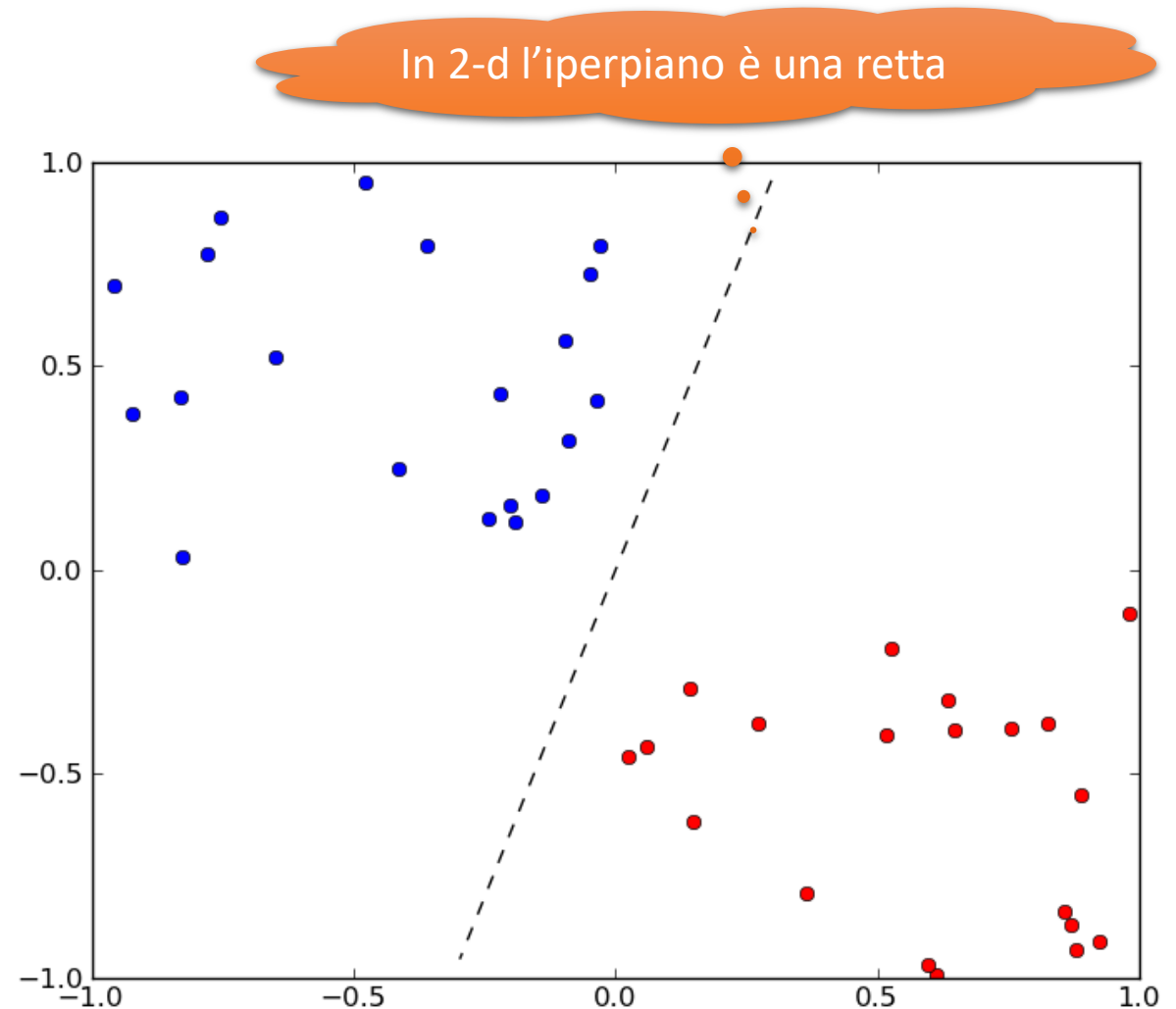
Ad esempio con $K = 4$ in questo caso possiamo decidere che il nuovo arrivato è un blu.

Perceptrons



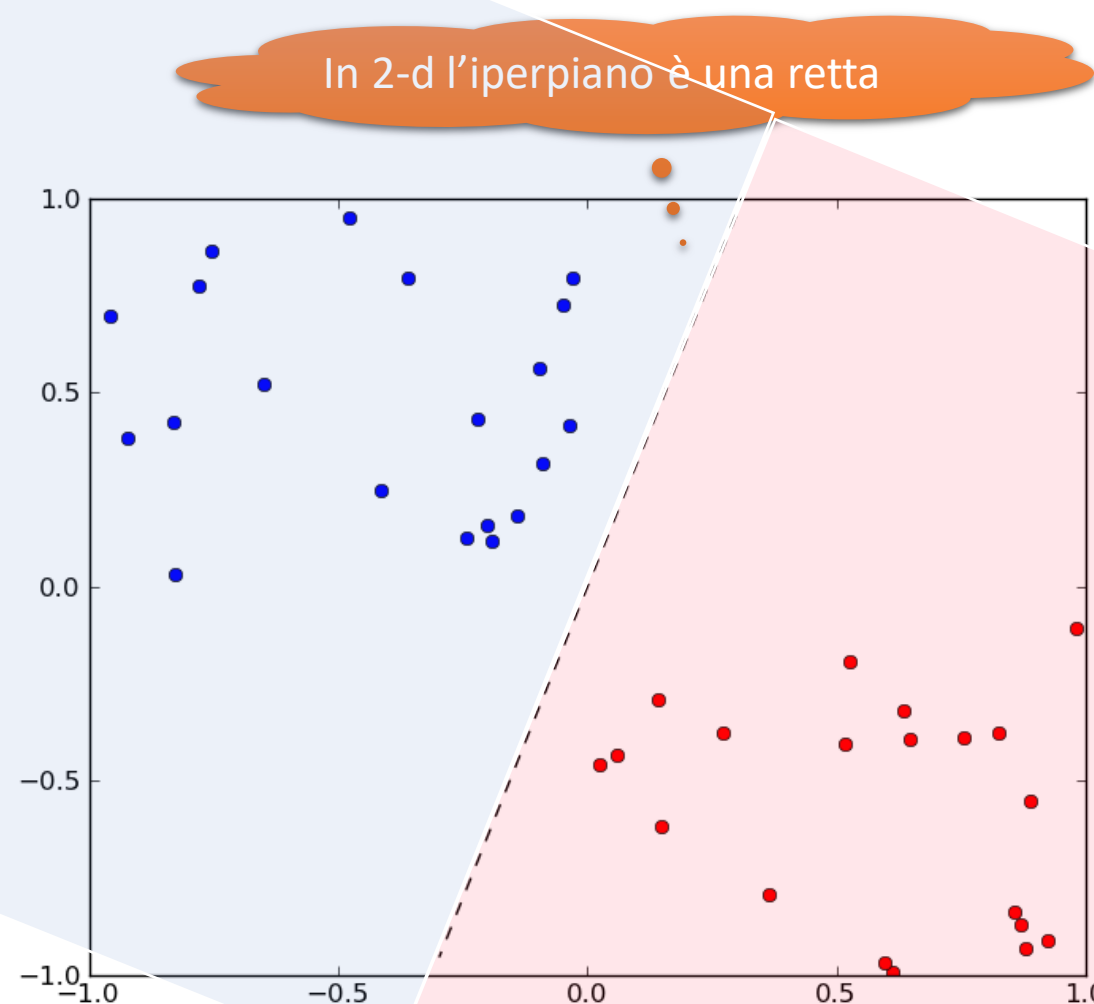
Perceptrons

- Ci sono casi in cui i dati di apprendimento sono così ben separati tra le classi disponibili che è possibile tracciare un iperpiano che separa le classi



Perceptrons

- Ad esempio, in 2-D e in caso di due classi possiamo tracciare una retta che separa i dati delle due classi.



Perceptrons

- Dalla geometria sappiamo che una retta ha una equazione del tipo

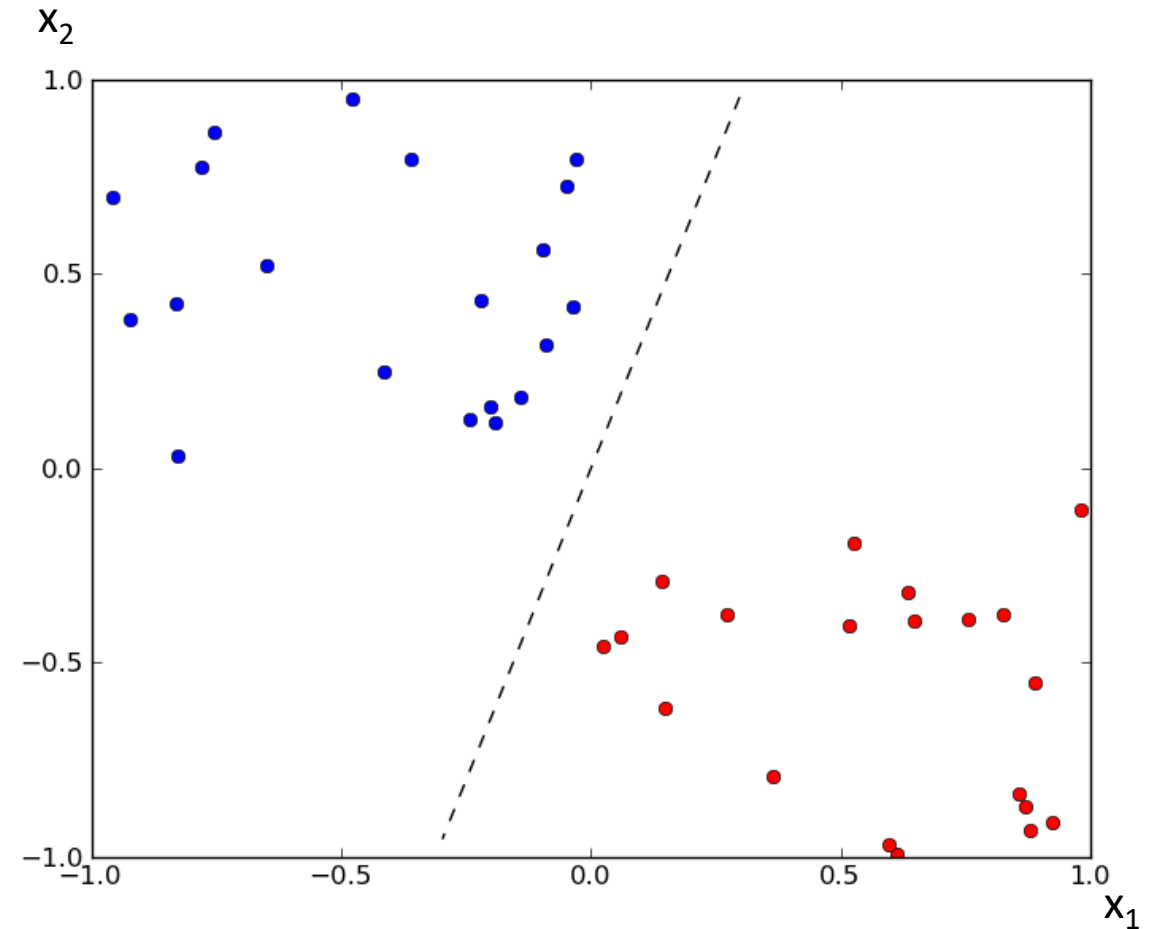
$$a_1x_1 + a_2x_2 + a_3 = 0$$

- Inoltre, i punti (x_1, x_2) che si trovano da un lato della retta sono tali che

$$a_1x_1 + a_2x_2 + a_3 > 0$$

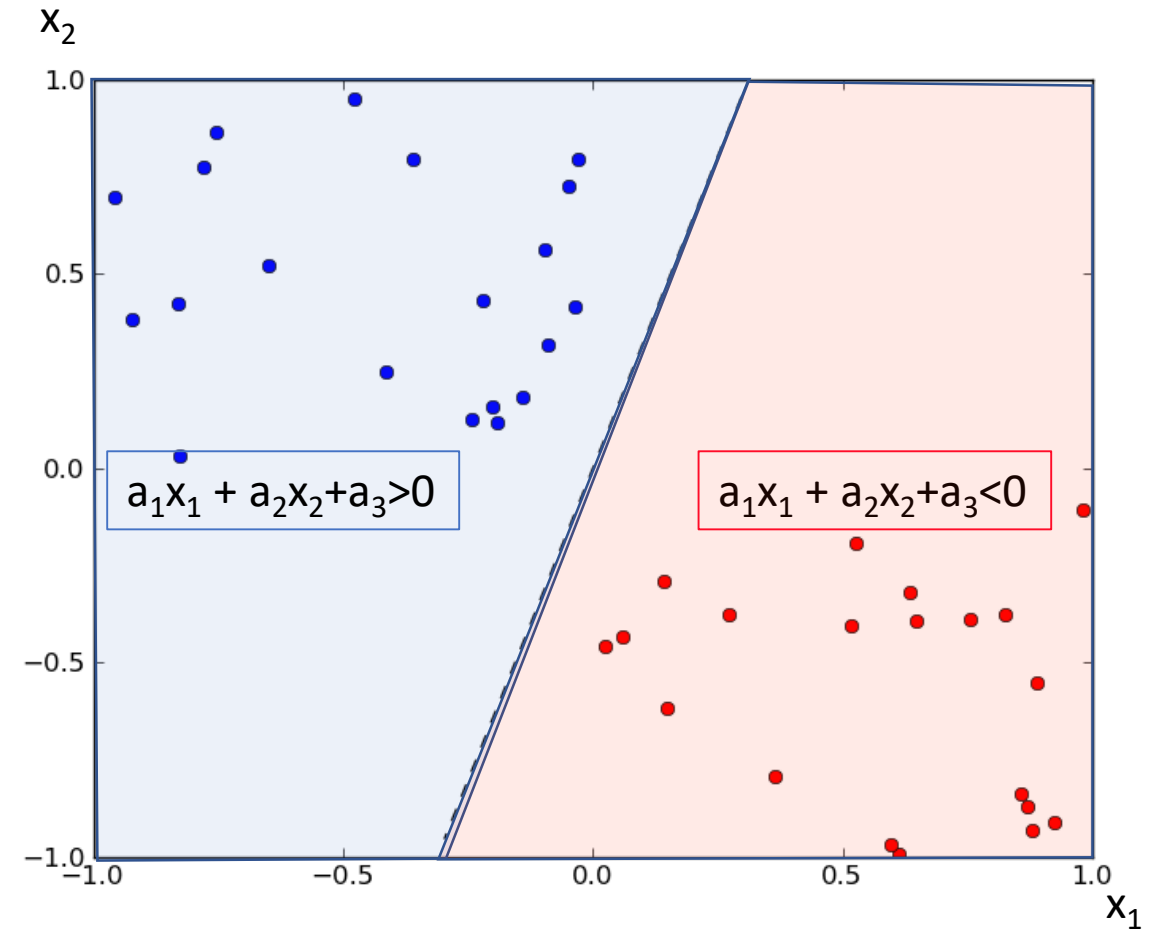
- E, i punti (x_1, x_2) che si trovano dall'altro lato della retta sono tali che

$$a_1x_1 + a_2x_2 + a_3 < 0$$



Perceptrons

- Questa circostanza ci consente di separare il piano in due semipiani.
- Uno per $a_1x_1 + a_2x_2 + a_3 > 0$ (blu)
- Uno per $a_1x_1 + a_2x_2 + a_3 < 0$ (rosso)



Perceptrons

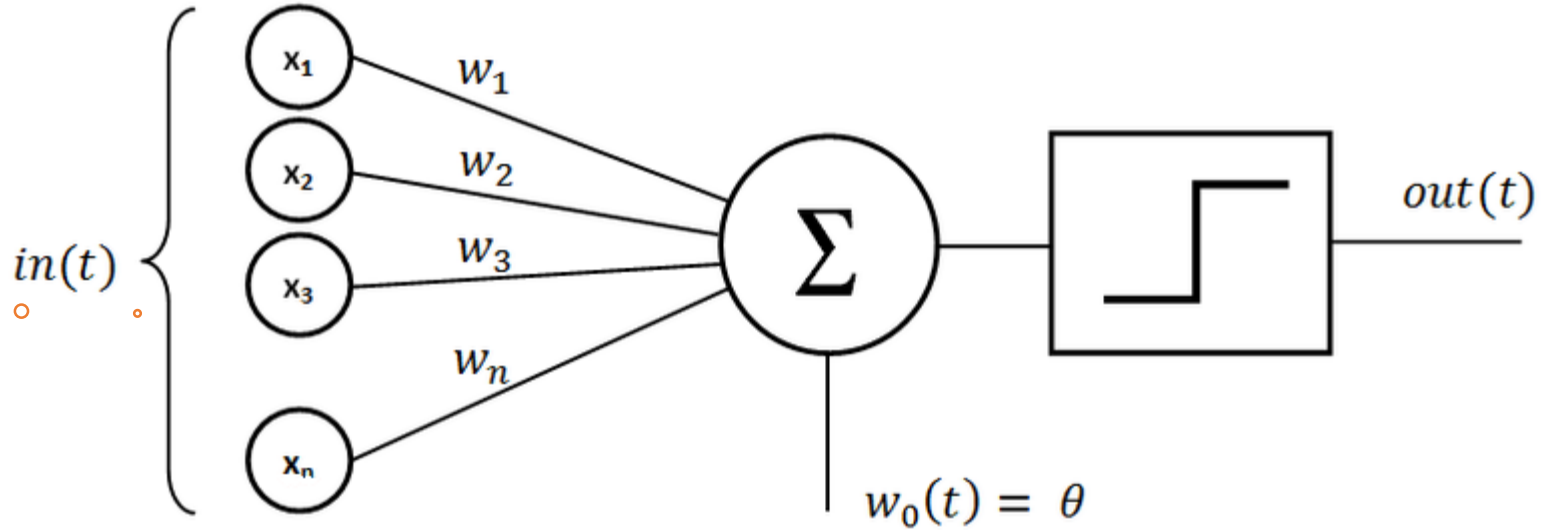
- Quindi una buona funzione ipotesi h è proprio l'equazione della retta (in 2D) o dell'iperpiano (in n -D)
- $y = h(x_1, x_2, \dots, x_n) = \sum_1^n a_i x_i + a_{n+1}$
- In questo modo, se $y > 0$ il punto (x_1, x_2, \dots, x_n) appartiene alla classe 1 mentre se $y < 0$ allora (x_1, x_2, \dots, x_n) appartiene alla classe 2

Perceptrons

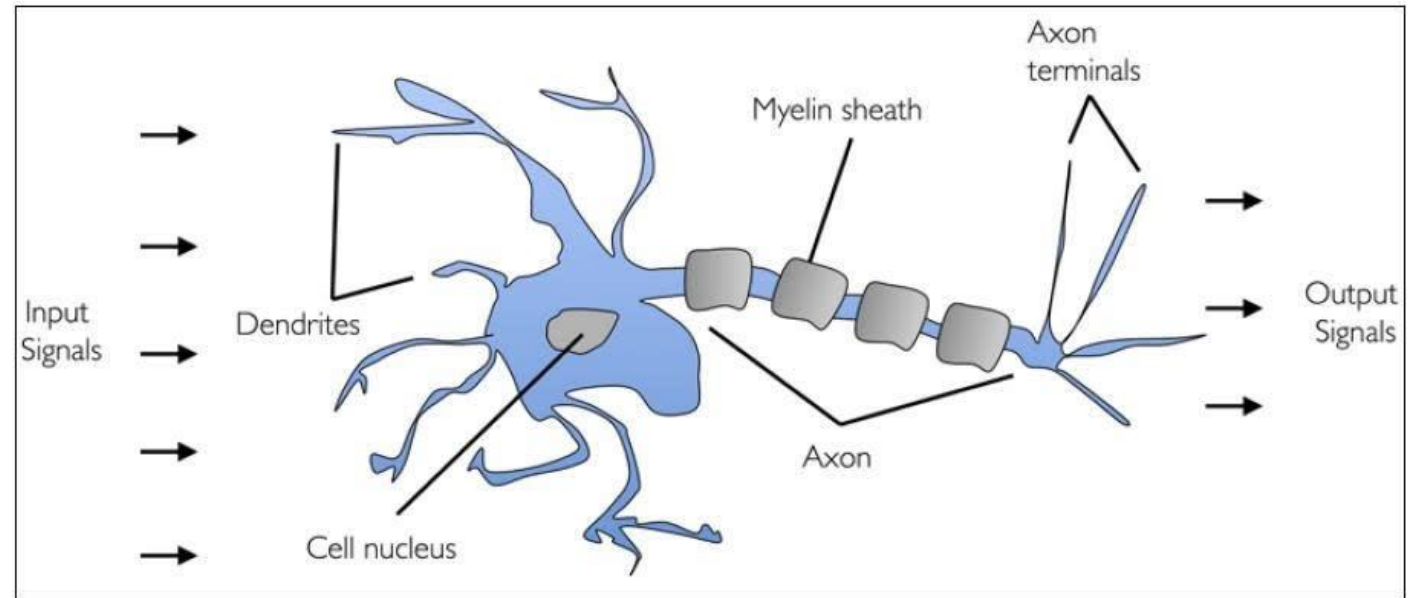
- Il problema di classificazione si riconduce quindi al problema di trovare i coefficienti a_i della formula dell'iperpiano.
- Poiché la letteratura sul tema è tipicamente in lingua Inglese si troverà spesso indicati i coefficienti con w_i (weight = peso) i coefficienti a_i , che possiamo immaginare come dei «pesi» che applichiamo agli ingressi x_i per ottenere la risposta y .

Perceptrons

La nostra AI



una cellula
nervosa.

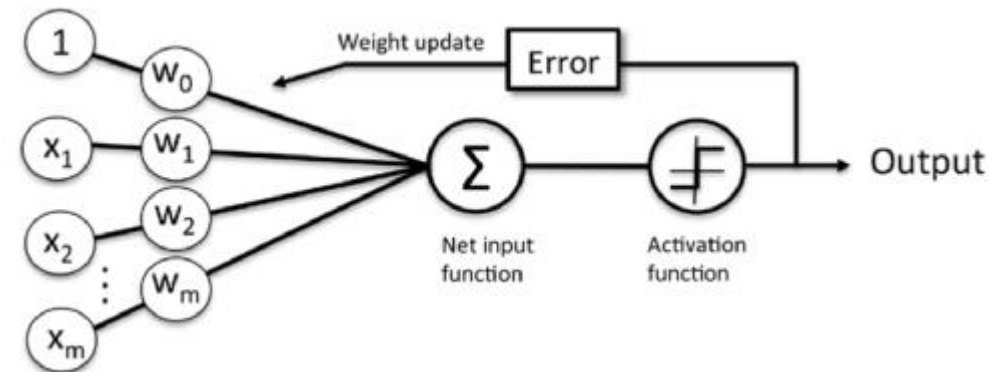


Perceptrons

Addestramento di un perceptrone

- Per calcolare i pesi w si può utilizzare un approccio iterativo che applica un insieme di coefficienti w iniziali e per ogni campione del dataset di addestramento calcola la differenza tra $h(x)$ e y e la usa per «aggiustare» i coefficienti w fino ad una possibile convergenza verso valori di w ottimali:

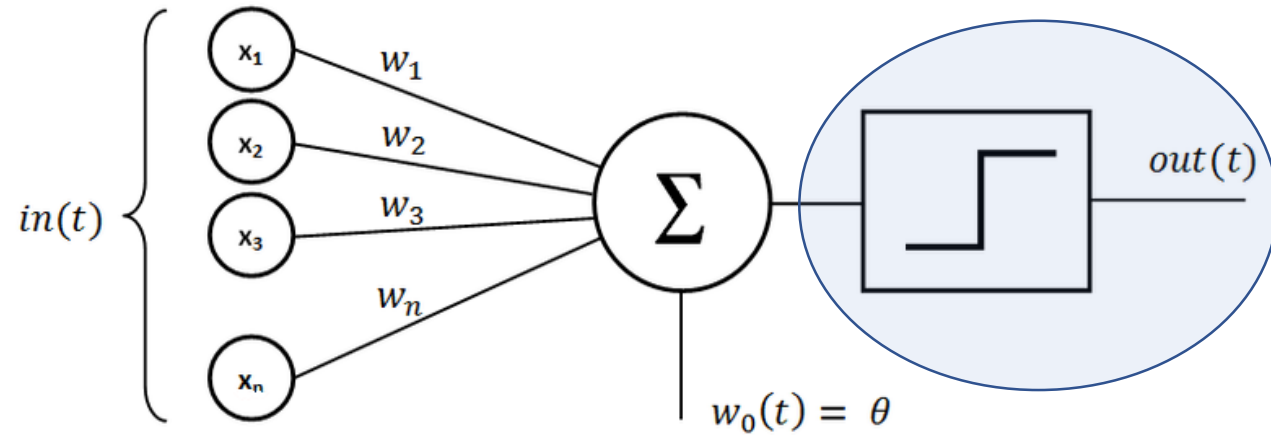
- $w_i = w_i + \alpha(y - h_w(X)) x_i$



Perceptrons

- Al di là dei metodi di stima dei pesi del perceptrone quello che ci interessa è che se i dati di addestramento mostrano la possibilità di suddividere i campioni appartenenti a due classi nei due semispazi che un iperpiano divide nell'iperspazio allora possiamo usare un perceptrone come classificatore che ha una funzione ipotesi $h(X)$ pari all'equazione dell'iperpiano separatore.

Perceptrons



- Inoltre è comodo aggiungere dopo la funzione h una funzione a gradino. In tal modo, l'uscita out sarà pari a 0 per i punti appartenenti a una classe e 1 per i punti appartenenti all'altra classe

Apprendimento non supervisionato

