

Reti Neurali

Corso di AA, anno 2016/17, Padova

Fabio Aiolli

2 Novembre 2016

unipd_logo.png

Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano



Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello



Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello
 - riproduzione fedele fenomeni neuro-fisiologici



Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello
 - riproduzione fedele fenomeni neuro-fisiologici
 - verifica sperimentale (il modello proposto riproduce i dati biologici?)



unipd_logo.png

Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello
 - riproduzione fedele fenomeni neuro-fisiologici
 - verifica sperimentale (il modello proposto riproduce i dati biologici?)
- Estrazione dei principi fondamentali di calcolo usati dal cervello umano



Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello
 - riproduzione fedele fenomeni neuro-fisiologici
 - verifica sperimentale (il modello proposto riproduce i dati biologici?)
- Estrazione dei principi fondamentali di calcolo usati dal cervello umano
 - non importa riprodurre il cervello ma solo evincere quali sono i principi fondamentali di calcolo utilizzati



unipd_logo.png

Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello
 - riproduzione fedele fenomeni neuro-fisiologici
 - verifica sperimentale (il modello proposto riproduce i dati biologici?)
- Estrazione dei principi fondamentali di calcolo usati dal cervello umano
 - non importa riprodurre il cervello ma solo evincere quali sono i principi fondamentali di calcolo utilizzati
 - semplificazioni ed astrazioni permesse, anzi, strumento principale di lavoro



unipd_logo.png

Reti Neurali Artificiali: Generalità

Due motivazioni diverse hanno spinto storicamente lo studio di Reti Neurali Artificiali (RNA):

- Riproduzione (e quindi comprensione) del cervello umano
 - modelli affidabili di tutto o almeno parte del cervello
 - riproduzione fedele fenomeni neuro-fisiologici
 - verifica sperimentale (il modello proposto riproduce i dati biologici?)
- Estrazione dei principi fondamentali di calcolo usati dal cervello umano
 - non importa riprodurre il cervello ma solo evincere quali sono i principi fondamentali di calcolo utilizzati
 - semplificazioni ed astrazioni permesse, anzi, strumento principale di lavoro
 - produrre un sistema artificiale, eventualmente diverso dal cervello, che riproduca le sue funzioni, magari più veloci ed efficienti (metafora del volo “aereo vs uccello”)

A noi interessa la seconda motivazione!

unipd_logo.png

A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)



A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)
- Apprendimento non supervisionato (clustering, data mining, self-organized maps, memorie associative, ...)



A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)
- Apprendimento non supervisionato (clustering, data mining, self-organized maps, memorie associative, ...)



unipd_logo.png

A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)
- Apprendimento non supervisionato (clustering, data mining, self-organized maps, memorie associative, ...)

Tali modelli possono differire per:

- Topologia della rete



A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)
- Apprendimento non supervisionato (clustering, data mining, self-organized maps, memorie associative, ...)

Tali modelli possono differire per:

- Topologia della rete
- Funzione calcolata dal singolo neurone




A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)
- Apprendimento non supervisionato (clustering, data mining, self-organized maps, memorie associative, ...)

Tali modelli possono differire per:

- Topologia della rete
- Funzione calcolata dal singolo neurone
- Algoritmo di apprendimento



unipd_logo.png

A noi interessa la seconda motivazione!

In questo caso i modelli proposti sono molteplici e con scopi diversi:

- Apprendimento supervisionato (classificazione, regressione, serie temporali, ...)
- Apprendimento non supervisionato (clustering, data mining, self-organized maps, memorie associative, ...)

Tali modelli possono differire per:

- Topologia della rete
- Funzione calcolata dal singolo neurone
- Algoritmo di apprendimento
- Modalità di apprendimento (utilizzo dei dati di apprendimento)



Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità



Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)



unipd_logo.png

Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta



unipd_logo.png

Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta
- Accettabile avere tempi lunghi di apprendimento e richiesta una veloce valutazione della funzione appresa



unipd_logo.png

Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta
- Accettabile avere tempi lunghi di apprendimento e richiesta una veloce valutazione della funzione appresa
- La soluzione finale **NON** deve essere compresa da un esperto umano (“black box problem”)



unipd_logo.png

Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta
- Accettabile avere tempi lunghi di apprendimento e richiesta una veloce valutazione della funzione appresa
- La soluzione finale **NON** deve essere compresa da un esperto umano (“black box problem”)



unipd_logo.png

Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta
- Accettabile avere tempi lunghi di apprendimento e richiesta una veloce valutazione della funzione appresa
- La soluzione finale NON deve essere compresa da un esperto umano (“black box problem”)

Esempi di campi applicativi:

- Riconoscimento del parlato (speech recognition)



Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta
- Accettabile avere tempi lunghi di apprendimento e richiesta una veloce valutazione della funzione appresa
- La soluzione finale NON deve essere compresa da un esperto umano (“black box problem”)

Esempi di campi applicativi:

- Riconoscimento del parlato (speech recognition)
- Classificazione di immagini (image classification)



Quando usare una RNA?

Caratteristiche del problema:

- Input: discreto e/o a valori reali, alta dimensionalità
- Output: vettore di valori discreti (classificazione) o reali (regressione)
- I dati (input e/o output) possono contenere rumore e la forma della funzione target totalmente sconosciuta
- Accettabile avere tempi lunghi di apprendimento e richiesta una veloce valutazione della funzione appresa
- La soluzione finale NON deve essere compresa da un esperto umano (“black box problem”)

Esempi di campi applicativi:


- Riconoscimento del parlato (speech recognition)
- Classificazione di immagini (image classification)
- Predizione su serie temporali (time series prediction)



Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi




unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni



unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi




unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi



unipd_logo.png


Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi

Considerando che:

- Per riconoscere il contenuto di una scena un umano impiega ~ 1.1 secondi



unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi

Considerando che:

- Per riconoscere il contenuto di una scena un umano impiega ~ 1.1 secondi
- Non può effettuare piú di ~ 100 calcoli seriali [$0.1/0.001 = 100$]



unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi

Considerando che:

- Per riconoscere il contenuto di una scena un umano impiega ~ 1.1 secondi
- Non può effettuare piú di ~ 100 calcoli seriali [$0.1/0.001 = 100$]
- Ne consegue che..



unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi

Considerando che:

- Per riconoscere il contenuto di una scena un umano impiega ~ 1.1 secondi
- Non può effettuare piú di ~ 100 calcoli seriali [$0.1/0.001 = 100$]
- Ne consegue che..



unipd_logo.png

Confronto con il cervello umano

Le RNA si ispirano al cervello umano:

- Il cervello è costituito da $\sim 10^{11}$ neuroni fortemente interconnessi
- Ogni neurone è connesso con un altri $\sim 10^4$ neuroni
- Il tempo di risposta di un neurone è ~ 0.001 secondi

Considerando che:

- Per riconoscere il contenuto di una scena un umano impiega ~ 1.1 secondi
- Non può effettuare piú di ~ 100 calcoli seriali [$0.1/0.001 = 100$]
- Ne consegue che..

Il cervello umano sfrutta pesantemente il **calcolo parallelo**.



unipd_logo.png

Singolo neurone - Perceptron

Consideriamo lo spazio degli iperpiani in \mathbb{R}^n (n dimensione dell'input):



Singolo neurone - Perceptron

Consideriamo lo spazio degli iperpiani in \mathbb{R}^n (n dimensione dell'input):

$$\mathcal{H} = \{f_{(\mathbf{w},b)}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) : \mathbf{w}, \mathbf{x} \in \mathbb{R}^n, b \in \mathbb{R}\}$$



unipd_logo.png

Singolo neurone - Perceptron

Consideriamo lo spazio degli iperpiani in \mathbb{R}^n (n dimensione dell'input):


$$\mathcal{H} = \{f_{(\mathbf{w}, b)}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) : \mathbf{w}, \mathbf{x} \in \mathbb{R}^n, b \in \mathbb{R}\}$$

che possiamo riscrivere come:

$$\mathcal{H} = \{f_{\mathbf{w}'}(\mathbf{x}') = \text{sign}(\mathbf{w}' \cdot \mathbf{x}') : \mathbf{w}', \mathbf{x}' \in \mathbb{R}^n\}$$

effettuando le seguenti trasformazioni:

$$\mathbf{w}' = [b, \mathbf{w}], \quad \mathbf{x}' = [1, \mathbf{x}]$$



unipd_logo.png

Singolo neurone - Perceptron

Consideriamo lo spazio degli iperpiani in \mathbb{R}^n (n dimensione dell'input):

$$\mathcal{H} = \{f_{(\mathbf{w}, b)}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) : \mathbf{w}, \mathbf{x} \in \mathbb{R}^n, b \in \mathbb{R}\}$$

che possiamo riscrivere come:

$$\mathcal{H} = \{f_{\mathbf{w}'}(\mathbf{x}') = \text{sign}(\mathbf{w}' \cdot \mathbf{x}') : \mathbf{w}', \mathbf{x}' \in \mathbb{R}^n\}$$

effettuando le seguenti trasformazioni:

$$\mathbf{w}' = [b, \mathbf{w}], \quad \mathbf{x}' = [1, \mathbf{x}]$$

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

unipd_logo.png

Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori `or`, `and`, `not`



Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori `or, and, not`
- Può un Perceptron implementare la `or`? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$



unipd_logo.png

Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori **or, and, not**
- Può un Perceptron implementare la **or**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- Può un Perceptron implementare la **and**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -n + 0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$

unipd_logo.png

Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori **or, and, not**
- Può un Perceptron implementare la **or**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- Può un Perceptron implementare la **and**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -n + 0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- La **not** si realizza banalmente con un Perceptron avente una singola connessione (fare per esercizio)



unipd_logo.png

Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori **or, and, not**
- Può un Perceptron implementare la **or**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- Può un Perceptron implementare la **and**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -n + 0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- La **not** si realizza banalmente con un Perceptron avente una singola connessione (fare per esercizio)

unipd_logo.png

Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori **or, and, not**
- Può un Perceptron implementare la **or**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- Può un Perceptron implementare la **and**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -n + 0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- La **not** si realizza banalmente con un Perceptron avente una singola connessione (fare per esercizio)

Quindi una qualsiasi funzione booleana può essere realizzata come combinazione di Perceptron!

unipd_logo.png

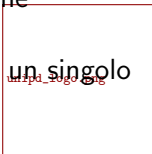
Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori **or, and, not**
- Può un Perceptron implementare la **or**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- Può un Perceptron implementare la **and**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -n + 0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- La **not** si realizza banalmente con un Perceptron avente una singola connessione (fare per esercizio)

Quindi una qualsiasi funzione booleana può essere realizzata come combinazione di Perceptron!

Esiste una funzione booleana semplice NON REALIZZABILE da un singolo Perceptron?



Funzioni booleane e Perceptron

Consideriamo input binari e funzioni booleane.

- Ogni funzione booleana può essere rappresentata tramite gli operatori **or, and, not**
- Può un Perceptron implementare la **or**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- Può un Perceptron implementare la **and**? SI!
Es. $\mathbf{x} \in \{0, 1\}^{n+1}$, $\mathbf{w}'_0 = -n + 0.5$, $\{\mathbf{w}_i = 1\}_{i=1, \dots, n}$
- La **not** si realizza banalmente con un Perceptron avente una singola connessione (fare per esercizio)

Quindi una qualsiasi funzione booleana può essere realizzata come combinazione di Perceptron!

Esiste una funzione booleana semplice NON REALIZZABILE da un singolo Perceptron?

SI, vista quando abbiamo parlato di VC-dim dello spazio di iperpiani in \mathbb{R}^n


Algoritmo di apprendimento del Perceptron

Assumiamo di avere esempi di apprendimento **linearmente separabili**, ovvero tali che esiste un iperpiano che li possa separare.

Perceptron

Input: Insieme di apprendimento $\mathcal{S} = \{(\mathbf{x}, t)\}$, $\mathbf{x} \in \mathbb{R}^{n+1}$, $t \in \{-1, +1\}$,
 $\eta \geq 0$ (learning rate)

- 1 Inizializza il valore dei pesi \mathbf{w} ad un vettore random



unipd_logo.png

Algoritmo di apprendimento del Perceptron

Assumiamo di avere esempi di apprendimento **linearmente separabili**, ovvero tali che esiste un iperpiano che li possa separare.

Perceptron

Input: Insieme di apprendimento $\mathcal{S} = \{(\mathbf{x}, t)\}, \mathbf{x} \in \mathbb{R}^{n+1}, t \in \{-1, +1\}$,
 $\eta \geq 0$ (learning rate)

- 1 Inizializza il valore dei pesi \mathbf{w} ad un vettore random
- 2 Ripeti



unipd_logo.png

Algoritmo di apprendimento del Perceptron

Assumiamo di avere esempi di apprendimento **linearmente separabili**, ovvero tali che esiste un iperpiano che li possa separare.

Perceptron

Input: Insieme di apprendimento $\mathcal{S} = \{(\mathbf{x}, t)\}$, $\mathbf{x} \in \mathbb{R}^{n+1}$, $t \in \{-1, +1\}$,
 $\eta \geq 0$ (learning rate)

① Inizializza il valore dei pesi \mathbf{w} ad un vettore random

② Ripeti

(a) Seleziona (a caso) uno degli esempi di apprendimento (\mathbf{x}, t)



Algoritmo di apprendimento del Perceptron

Assumiamo di avere esempi di apprendimento **linearmente separabili**, ovvero tali che esiste un iperpiano che li possa separare.

Perceptron

Input: Insieme di apprendimento $\mathcal{S} = \{(\mathbf{x}, t)\}$, $\mathbf{x} \in \mathbb{R}^{n+1}$, $t \in \{-1, +1\}$,
 $\eta \geq 0$ (learning rate)

- 1 Inizializza il valore dei pesi \mathbf{w} ad un vettore random
- 2 Ripeti

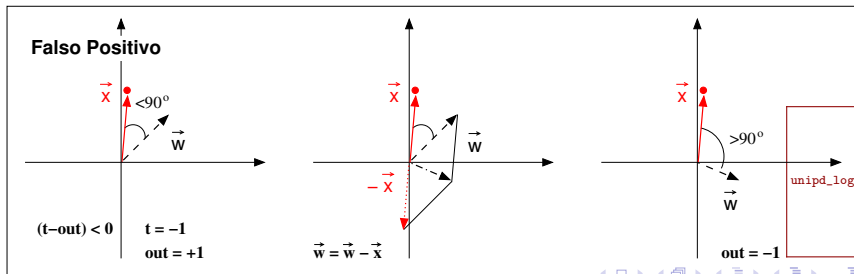
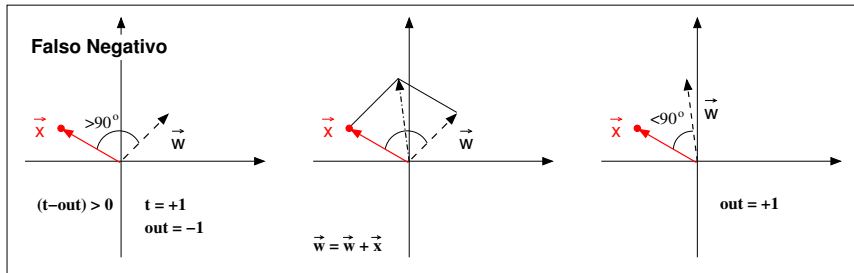
- (a) Seleziona (a caso) uno degli esempi di apprendimento (\mathbf{x}, t)
- (b) Se $o = \text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq t$, allora

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(t - o)\mathbf{x}$$




unipd_logo.png

Interpretazione Geometrica



Osservazioni

- Non necessariamente un singolo passo di apprendimento (passo 2(b)) riuscirà a modificare il segno dell'output, potrebbero servire più passi



unipd_logo.png

Osservazioni

- Non necessariamente un singolo passo di apprendimento (passo 2(b)) riuscirà a modificare il segno dell'output, potrebbero servire più passi
- Il coefficiente η **learning rate** serve per rendere più stabile l'apprendimento, ovvero evita che il vettore dei pesi subisca variazioni troppo “violente” quando il passo 2(b) viene eseguito



unipd_logo.png

Osservazioni

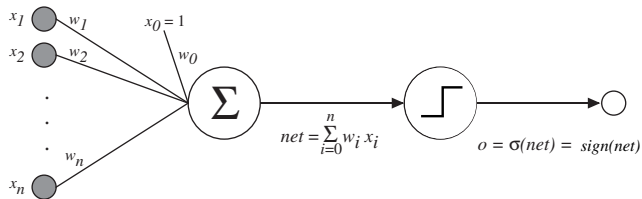
- Non necessariamente un singolo passo di apprendimento (passo 2(b)) riuscirà a modificare il segno dell'output, potrebbero servire più passi
- Il coefficiente η **learning rate** serve per rendere più stabile l'apprendimento, ovvero evita che il vettore dei pesi subisca variazioni troppo “violente” quando il passo 2(b) viene eseguito
- 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 finito di passi**



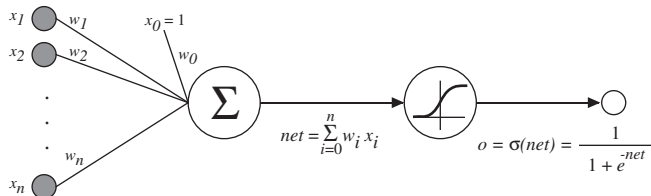
unipd_logo.png

Neurone Artificiale

Alternativa 1: Hard-threshold




Alternativa 2: Neurone con sigmoide (derivabile)



Reti Neurali Multistrato

Una RNA multistrato (multi-layer NN) è un sistema costituito da unità interconnesse che calcolano funzioni (numeriche) non-lineari:

- Le unità di **input** rappresentano le variabili in ingresso



unipd_logo.png

Reti Neurali Multistrato

Una RNA multistrato (multi-layer NN) è un sistema costituito da unità interconnesse che calcolano funzioni (numeriche) non-lineari:

- Le unità di **input** rappresentano le variabili in ingresso
- Le unità di **output** rappresentano le variabili di uscita



unipd_logo.png

Reti Neurali Multistrato

Una RNA multistrato (multi-layer NN) è un sistema costituito da unità interconnesse che calcolano funzioni (numeriche) non-lineari:

- Le unità di **input** rappresentano le variabili in ingresso
- Le unità di **output** rappresentano le variabili di uscita
- Le unità **nascoste** (se ve ne sono) rappresentano variabili interne che (dopo l'apprendimento) codificano le correlazioni tra le variabili di input, relativamente al valore di output che si vuole generare



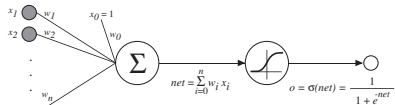
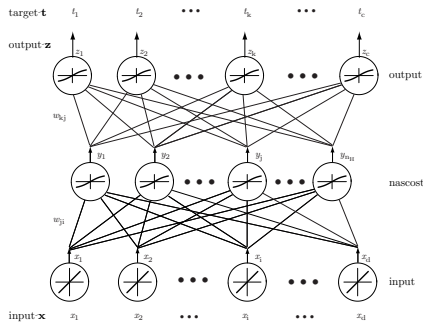
Reti Neurali Multistrato

Una RNA multistrato (multi-layer NN) è un sistema costituito da unità interconnesse che calcolano funzioni (numeriche) non-lineari:

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

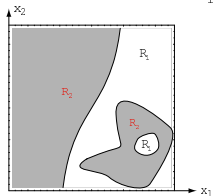
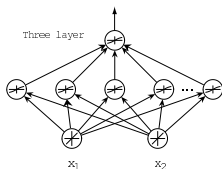
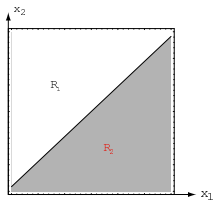
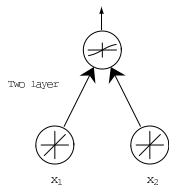
unipd_logo.png

Reti Neurali Multistrato



unipd_logo.png

Superfici di decisione



unipd_logo.png