

Appunti di Intelligenza Artificiale

A.A. 2023/2024

Rosso Carlo

Indice

1	Introduzione	2
1.1	Che cosa sono le Artificial Neural Networks (ANN)	2
1.2	Caratteristiche delle ANN	5
2	Fondamenti	5
2.1	Il neurone artificiale	5
2.2	Funzioni di attivazione	6
	Funzione "a gradino"	7
	Funzione lineare	7
	Funzione sigmoide	8
2.3	Rappresentazione	8
	Codifica locale	9
	Codifica distribuita	9
	Codifica grezza	9
	Campi recettivi sovrapposti	10
	Preparazione	10
	Normalizzazione	10
2.4	Apprendimento	11
2.5	Analisi vettoriale di un neurone artificiale	12
2.6	Apprendimento hebbiano	13
	La regola di Hebb	13
	La regola postsinaptica	13
	La regola presinaptica	14
	La regola della covarianza	14

1 Introduzione

I programmi tradizionali elaborano l'informazione in modo radicalmente diverso rispetto ai sistemi nervosi biologici. Un computer tradizionale è composto sostanzialmente da un'unità di calcolo (processore, CPU), che esegue in successione un altissimo numero di operazioni; e da tre tipi di memoria: una che contiene le istruzioni necessaria a svolgere le operazioni, una temporanea da cui vengono letti i dati necessari e salvati i risultati dei calcoli effettuati e una permanente in cui questi dati rimangono registrati. L'architettura e i principi di funzionamento del computer seriale sono stati utilizzati dal cognitivismo come metafora della mente; questa scelta presenta due vantaggi:

1. **Formalismo:** la scienza dell'informazione fornisce un formalismo scientifico e universalmente accettato con cui è possibile descrivere il funzionamento della mente in modo univoco;
2. **Modelli:** rende possibile l'impiego di modelli della mente per la creazione di nuove tecnologie con caratteristiche di intelligenza artificiale.

Il sistema nervoso può essere paragonato a un'immensa società (Cervellopoli). Ciascun abitante conosce quasi tutti gli abitanti del proprio paese o quartiere e passa il proprio tempo a parlare con tutti loro. Alcuni abitanti possiedono anche relazioni con individui che vivono in zone più distanti e mantengono così la propria comunità continuamente aggiornata su quello che succede. La comunicazione è spesso caratterizzata da ripetizioni, rumore e interruzioni.

L'elaborazione nei sistemi nervosi avviene in *parallelo* (mentre nei calcolatori tradizionali avviene in successione). L'elaborazione nei sistemi nervosi è *distribuita* su molti elementi. Durante lo svolgimento dei compiti, si attivano molti neuroni, a volte organizzati in gruppetti locali, altre volte distribuiti a macchie in zone diverse nel cervello. I sistemi nervosi non devono essere programmati per svolgere un compito, bensì imparano autonomamente in base all'esperienza o con l'aiuto di un insegnante esterno. Si ritiene che l'apprendimento consista soprattutto nella modifica della "forza" delle connessioni attraverso cui i neuroni comunicano.

1.1 Che cosa sono le Artificial Neural Networks (ANN)

Le reti neurali artificiali sono dei sistemi di elaborazione dell'informazione, il cui funzionamento trae ispirazione dai sistemi nervosi biologici. Una rete neurale artificiale possiede molte semplici unità

di elaborazione variamente connesse tra di loro. Alcune unità ricevono informazioni dall'ambiente (input), altre emettono risposte nell'ambiente (output), altre ancora comunicano solamente con le unità all'interno della rete (hidden).

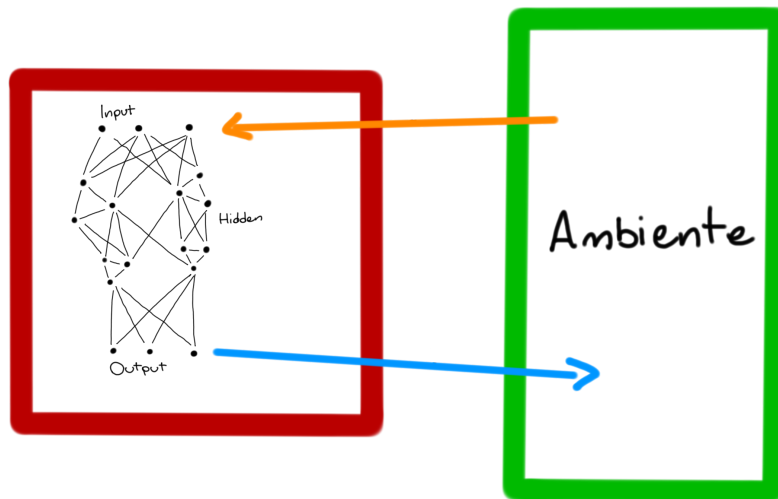


Figura 1: Schema di una rete neurale artificiale

Ciascuna unità simula il ruolo di un neurone (infatti sono anche chiamate neuroni o percettori (perceptrons)). Ciascuna unità diventa attiva se la quantità totale di segnale che riceve supera la propria soglia di attivazione. Ciascun punto di connessione agisce come un filtro che trasforma il messaggio ricevuto in un segnale inibitorio o eccitatorio, aumentandone o diminuendone l'intensità. Poiché il loro ruolo consiste in effetti nel pesare l'intensità dei segnali trasmessi, essi vengono definiti anche con il nome di pesi sinaptici o semplicemente pesi.

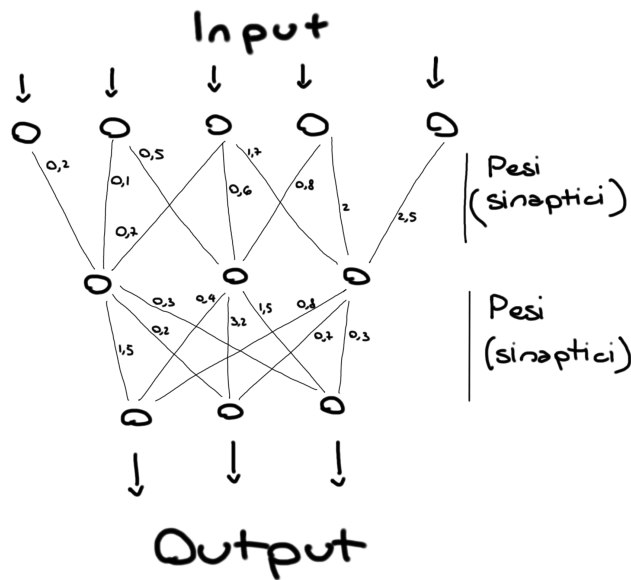


Figura 2: Schema di una rete neurale artificiale

Formalmente, il segnale di risposta emesso da un nodo n , è uguale a

$$n_i = \phi\left(\sum_{j \in \text{input}} w_{ij} \cdot n_j - \theta_i\right) \quad (1)$$

Dove:

- n_j è l'output del nodo j ;
- w_{ij} è il peso della connessione tra i e j ;
- θ_i è la soglia di attivazione del nodo i ;
- ϕ è una generica funzione di attivazione (ne studieremo alcune all'interno del corso).

Quando uno stimolo viene applicato ai neuroni d'ingresso della rete neurale i segnali viaggiano in parallelo lungo le connessioni attraverso i nodi interni fino ai nodi di uscita la cui attivazione rappresenta la risposta della rete neurale. La configurazione delle connessioni e i valori delle sinapsi artificiali determinano in gran parte il comportamento e la risposta della rete. Per questo motivo si dice che le sinapsi rappresentano la conoscenza o memoria a lungo termine della rete neurale. Nelle reti neurali la memoria è totalmente distribuita e non richiede l'utilizzo di componenti aggiuntivi poichè è una proprietà intrinseca del sistema stesso.

La regola di apprendimento di un modello neurale indica semplicemente le condizioni locali e le

modalità in cui le sinapsi si modificano a prescindere dal tipo di compito per cui la rete verrà utilizzata.

1.2 Caratteristiche delle ANN

- **Robustezza:** una ANN è resistente al rumore, ovvero è in grado di continuare a dare una risposta corretta anche se alcune delle sue connessioni vengono eliminate o se viene aggiunto del rumore al segnale di ingresso;
- **Flessibilità:** una ANN può essere impiegata per un grande numero di finalità diverse: non ha bisogno di conoscere le proprietà del dominio specifico di applicazione perché le apprende in base all'esperienza. Questa caratteristica rappresenta un vantaggio perché permette di affrontare molti problemi di cui non sono note le soluzioni analitiche. Tuttavia, si rischia di rinunciare a cercare di comprendere a fondo la natura di un problema e di rifugiarsi in una soluzione neurale che non aumenta la nostra conoscenza;
- **Generalizzazione:** una ANN allenata su un numero limitato di esempi è in grado di produrre una risposta adeguata a dei pattern di ingresso che non ha mai visto in precedenza, ma che presentano qualche somiglianza con gli esempi presentati durante l'allenamento. In effetti, una ANN estrae caratteristiche invarianti dei pattern di ingresso piuttosto che memorizzare ciascun singolo pattern;
- **Recupero in base al contenuto:** le ANN sono in grado di recuperare le proprie memorie in base al contenuto partendo da dati incompleti, simili o corrotti da rumore.

2 Fondamenti

2.1 Il neurone artificiale

La descrizione degli elementi fondamentali dei circuiti neurali artificiali che segue si riferisce a proprietà generali dei modelli presentati nei capitoli successivi.

Un neurone artificiale è caratterizzato da un insieme di sinapsi che corrispondono ai terminali di altri neuroni, da una soglia e da una funzione di attivazione. L'input netto o potenziale di attivazione

A_i di un neurone i è la somma algebrica dei prodotti fra tutti i segnali di ingresso x_j e i valori dei pesi corrispondenti w_{ij} :

$$A_i = \sum_{j=1}^n w_{ij} x_j$$

A cui solitamente si sottrae il valore di soglia θ_i del neurone:

$$A_i = \sum_{j=1}^n w_{ij} x_j - \theta_i$$

La risposta del neurone y_i viene calcolata sottoponendo il potenziale di attivazione (A_i) così ottenuto all'azione di una funzione di attivazione $\phi(A)$:

$$y_i = \phi(A_i) = \phi\left(\sum_j w_{ij} x_j - \theta_i\right)$$

Nella maggior parte dei modelli il peso w_{ij} può assumere valori positivi o negativi continui (non interi, ovvero con la virgola) ed è il valore che si modifica durante la fase di apprendimento. Ora risulta vantaggioso analizzare il sistema in notazione vettoriale. Dato che il potenziale di attivazione di un neurone è una funzione lineare di segnali di ingresso, il potenziale di attivazione di un intero strato di neuroni $A^T = \{A_1, A_2, \dots, A_n\}$ è riscrivibile come:

$$A = W \cdot x$$

Dove:

- x^T è il vettore dei segnali d'ingresso e possono essere sia valori di input esterno (dall'ambiente) che valori di attivazione di uno strato inferiore di unità;
- $W = \{w_{ij}\}$ è la matrice dei pesi sinaptici, con w_{ij} che rappresenta il peso della connessione tra l'unità i e l'unità j ;

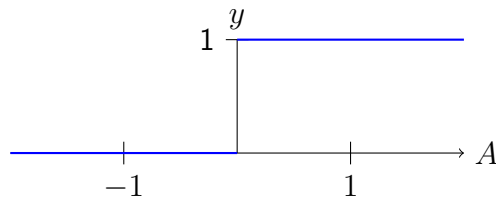
2.2 Funzioni di attivazione

La funzione di attivazione determina il tipo di risposta che un neurone può emettere. Di seguito sono elencati alcuni esempi di funzioni di attivazione.

Funzione "a gradino"

$$\phi(A) = \begin{cases} 1 & \text{se } A \geq \theta \\ 0 & \text{se altrimenti} \end{cases}$$

Dove θ è la soglia del neurone. Ponendo $\theta = 0$ si ottiene:



Alternativamente, l'output può essere bipolare:

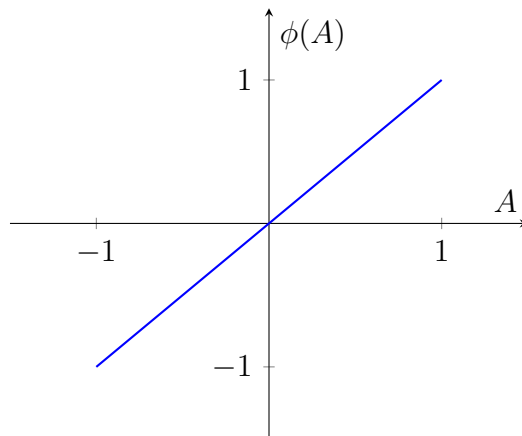
$$\phi(A) = \begin{cases} 1 & \text{se } A \geq \theta \\ -1 & \text{se altrimenti} \end{cases}$$

Funzione lineare

Maggiori informazioni sono trasmesse se si utilizza una funzione continua lineare:

$$\phi(A) = kA$$

dove k è una costante. Le funzioni continue permettono al neurone di trasmettere una gradazione di segnali di varia intensità che può essere opportunamente sfruttata dai neuroni riceventi.

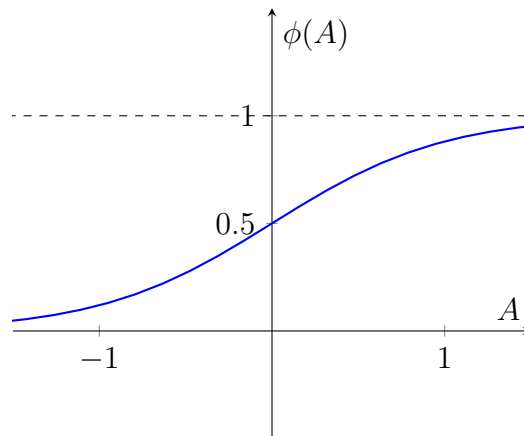


Funzione sigmoide

In alcune situazioni risulta utile forzare l'output del neurone ad assumere valori compresi in un intervallo, per esempio $[0, 1]$ o $[-1, 1]$. Infatti, in questo modo si può interpretare l'output come una probabilità o come una misura di confidenza. La funzione sigmoide o logistica è definita come:

$$\phi(A) = \frac{1}{1 + e^{-kA}}$$

Dove k è la costante che controlla l'inclinazione della curva; per $k \rightarrow \infty$ la funzione si comporta come una funzione a gradino. Le rette $y = 0$ e $y = 1$ sono asintoti orizzontali della funzione sigmoide.



Una funzione simile è la tangente iperbolica ($\tanh(kA)$) che ha le rette $y = -1$ e $y = 1$ come asintoti orizzontali.

Nella maggior parte dei modelli tutti i neuroni eccetto i neuroni di input (recettori) utilizzano la medesima funzione di attivazione.

2.3 Rappresentazione

Riporto alcune riflessioni sulle rappresentazione dei dati di input.

In generale le codifiche bipolari si rivelano vantaggiose rispetto alle codifiche binarie. Infatti, permette di trattare dati incompleti in modo neutro, al posto di usare -1 o 1 , il dato mancante può essere rappresentato con 0 , che non ha alcun effetto sulle operazioni di somma e prodotto (quindi la tangente iperbolica è una funzione di attivazione più adatta rispetto alla sigmoide).

Codifica locale

Nella codifica locale, ogni unità di input rappresenta un determinato oggetto. Si tratta di una rappresentazione molto semplice, ma ha diversi svantaggi:

1. richiede un alto numero di unità di input: ovvero n unità per rappresentare n oggetti;
2. richiede di conoscere il numero di oggetti da rappresentare in anticipo;
3. è molto fragile alla perdita di un'unità di input, che corrisponde alla perdita dell'oggetto che rappresenta.

Codifica distribuita

Nella codifica distribuita, gli oggetti sono rappresentati da un codice binario di n nodi che rappresentano 2^n oggetti. Anche questa rappresentazione ha alcuni svantaggi:

1. mancanza di flessibilità: non è possibile rappresentare nuovi oggetti senza modificare la struttura della rete;
2. fragilità: la perdita di un nodo di input può portare alla perdita di un'informazione significativa.

Codifica grezza

Una soluzione proposta codifica le caratteristiche degli oggetti. Per cui ciascun oggetto è descrivibile da un insieme di caratteristiche (e.g. peso, colore, altezza, ecc.). Ciascuna unità di input codifica la presenza o il valore di una certa caratteristica. Questa rappresentazione ha diversi vantaggi:

1. flessibilità: è possibile rappresentare nuovi oggetti senza modificare la struttura della rete;
2. robustezza: la perdita di un'unità di input, comporta la perdita di una caratteristica dell'oggetto, ma sono comunque presenti altre caratteristiche che permettono di identificare l'oggetto.

Le unità di input possono essere rappresentate come uno spazio multi-dimensionale con tante dimensioni quante sono le unità. Oggetti con caratteristiche simili saranno rappresentati da

punti vicini nello spazio multi-dimensionale. Per cui questa codifica facilita la classificazione e generalizzazione della rete neurale.

Campi recettivi sovrapposti

C'è un altro modo per ottenere una codifica grezza: si tassella l'input (e.g. l'immagine) in zone di dimensioni uguali e parzialmente sovrapposte. In questo caso, l'accuratezza della rappresentazione dipende dalla dimensione delle zone e dal numero di zone che coprono l'input. Approfondiamo le implicazioni che hanno le dimensioni e il numero di zone sulle informazioni che possono essere estratte:

- **campi recettivi piccoli e non sovrapposti**: sono usate per categorizzare le figure;
- **campi recettivi grandi e parzialmente sovrapposti**: sono usate per distinguere le singole figure. Infatti, larghi campi recettivi fanno sì che piccole caratteristiche possano essere codificate su un numero maggiore di nodi, permettendo una maggior discriminazione dei dettagli.

Preparazione

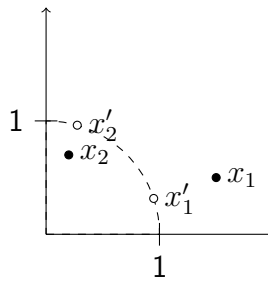
Prendendo spunto dalla biologia, risulta utile preparare i dati prima di essere dati in pasto alla rete neurale. Per esempio, si possono filtrare i dati, oppure si esegue la derivata sulle immagini per evidenziare i bordi e annullare i colori, senza perdere informazioni utili allo scopo del modello. In alternativa, si può usare una cascata di nodi con campi recettivi localmente ristretti in modo che l'ampiezza della finestra attraverso la quale ogni nodo osserva l'immagine diventa più grande man mano che si procede verso strati di elaborazione successiva.

Normalizzazione

Un'altra operazione che può essere eseguita sui dati è la normalizzazione. Questa operazione è utile per rendere i dati indipendenti dalla scala:

$$||x|| = \sqrt{\sum_{i=1}^n x_i^2}$$

In questo modo la lunghezza del vettore ottenuto sarà sempre 1.



2.4 Apprendimento

La risposta di una rete neurale è determinata dai valori sinaptici delle connessioni fra i nodi. Le reti neurali artificiali apprendono modificando gradualmente i propri valori sinaptici attraverso la presentazione ripetuta di una serie di esempi. Si distinguono tre tipi di apprendimento:

- **Apprendimento supervisionato:** la modifica dei valori sinaptici avviene impiegando una misura di errore tra la risposta fornita dalla rete neurale e la risposta desiderata per ogni vettore di input;
- **Apprendimento per rinforzo:** L'apprendimento supervisionato include anche una gamma di algoritmi che richiedono solamente una misura di bontà della risposta della rete neurale, piuttosto che la specificazione della risposta esatta per ogni pattern di input. Questi algoritmi sono denominati algoritmi di apprendimento per rinforzo;
- **Apprendimento non supervisionato:** sono anche chiamati apprendimento per auto-organizzazione. In questo caso, la rete neurale deve apprendere a riconoscere regolarità nei dati di input, in quanto la risposta desiderata è sconosciuta e deve essere individuata dalla rete medesima. Molti algoritmi di apprendimento senza supervisore derivano da una precisa formulazione dell'informazione che deve essere estratta dall'ambiente e richiedono dettagliate assunzioni sulla struttura dei pattern di input.

Approfondiamo ora le caratteristiche comuni a tutti i tipi di apprendimento:

1. **Stato iniziale:** i pesi iniziali sono assegnati in modo casuale entro un piccolo campo di variazione (e.g. $[-0.1, 0.1]$), oppure sono messi a zero;
2. **Iterazione:** l'apprendimento consiste nella presentazione ripetuta di una serie di vettori, detti anche pattern d'addestramento;

3. **Nuove conoscenze:** Gli algoritmi di apprendimento riguardano il calcolo di Δw_{ij} , la modifica dei pesi rispetto al valore corrente:

$$w_{ij}^t = w_{ij}^{t-1} + \Delta w_{ij}^t$$

4. **Tasso di apprendimento:** la velocità di apprendimento è regolata da una costante η :

$$w_{ij}^t = w_{ij}^{t-1} + \eta \Delta w_{ij}^t, \quad 0 < \eta < 1$$

5. **Test:** quando l'apprendimento è completato, i pesi sono "congelati" e si studia la risposta della rete neurale su dei vettori di test, che non sono stati usati durante l'apprendimento. Questo non è vero per gli algoritmi che operano all'interno della Teoria della Risonanza Adattiva.

2.5 Analisi vettoriale di un neurone artificiale

Per semplificare l'analisi consideriamo il caso di un singolo neurone i , tale che:

$$n_i = \phi\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right) = w \cdot x$$

La risposta dell'unità n_i è una misura della somiglianza tra il vettore di input ed il peso sinaptico. Come abbiamo già visto la norma (o lunghezza) di un vettore x è definita come:

$$||x|| = \sqrt{\sum_{j=1}^n x_j^2}$$

Il coseno dell'angolo θ tra due vettori x e w è definito come:

$$\cos(\theta) = \frac{w \cdot x}{||w|| \cdot ||x||}, \quad 0 \leq \theta \leq \pi$$

Quindi, il prodotto interno $n_i = w \cdot x$ è

$$n_i = ||w|| \cdot ||x|| \cdot \cos(\theta)$$

Questo significa che muovendo nello spazio i due vettori mantenendo costante la loro lunghezza, il loro prodotto interno sarà proporzionale al coseno dell'angolo θ tra i due vettori. Ovvero, minore è l'angolo e maggiore è il prodotto interno. In particolare il prodotto interno sarà massimo

quando i due vettori sono allineati, vale 1 quando sono ortogonali e sarà minimo quando sono opposti (e $\max = -\min$, al variare dell'angolo).

Infine, è importante notare che in una rete neurale è possibile giudicare quale unità possieda il vettore sinaptico più simile al vettore di input solo se i vettori sinaptici sono normalizzati.

2.6 Apprendimento hebbiano

La regola di modifica sinaptica di Hebb costituisce le fondamenta su cui si basano o derivano tutti gli algoritmi di apprendimento. Approfondiamo l'uso delle regole hebbiane in una rete etero-associativa feedforward con un singolo strato di sinapsi e con unità binarie. Consideriamo un algoritmo di apprendimento supervisionato.

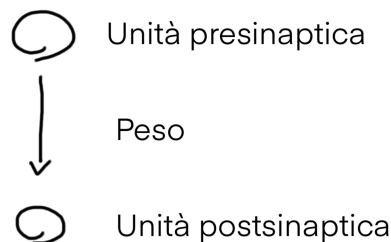


Figura 3: Rete neurale etero-associativa feedforward con un singolo strato di sinapsi e con unità binarie.

La regola di Hebb

Se due neuroni collegati tra di loro sono contemporaneamente attivi, il loro peso viene incrementato. La regola di Hebb prevede solamente l'incremento dei pesi, per cui la rete non è in grado di apprendere associazioni che presentano elementi in comune, ma che richiedono risposte differenti. Dunque la regola di Hebb permette di apprendere solamente pattern di input che sono ortogonali tra loro.

La regola postsinaptica

Questa regola è anche chiamata regola di Stent-Singer. Il peso viene incrementato ogni volta che l'unità postsinaptica e l'unità presinaptica sono entrambe attive; inoltre viene diminuito ogni volta che l'unità postsinaptica è attiva, ma quella presinaptica è inattiva. Questa regola migliora

la generalità, tuttavia, non fallisce l'apprendimento di pattern di input parzialmente sovrapposti sullo stesso output, perché tende a creare troppe sinapsi inibitorie.

La regola presinaptica

Il peso viene aumentato quando sia l'unità presinaptica che l'unità postsinaptica sono attive; il peso viene diminuito quando l'unità presinaptica è attiva, ma l'unità postsinaptica è inattiva. Questa regola permette di generalizzare molto meglio rispetto alle regole precedenti.

La regola della covarianza

Viene anche chiamata regola di Hopfield. Quando l'unità presinaptica e l'unità postsinaptica hanno lo stesso stato, il peso viene incrementato; quando hanno stati opposti, il peso viene diminuito.

Unità presinaptica	+	+	-	-
Unità postsinaptica	+	-	+	-
Hebb	+			
Postsinaptica	+		-	
Presinaptica	+	-		
Covarianza	+	-	-	+

Tabella 1: Riassunto del funzionamento delle regole hebbiane.