

## La Regola Delta

---

di Enrico Nanni

**Keywords:** Adaline, Delta, LINear, AI, Intelligenza Artificiale

In questa sezione ci occupiamo di algoritmi di apprendimento, procedendo gradualmente attraverso la breve descrizione dei principali algoritmi sviluppati da vari autori per ottimizzare le prestazioni dei vari modelli connessionisti. La "regola delta", o "regola Adaline", rientra in quello che prende il nome di "apprendimento supervisionato" (*supervised learning*). Abbiamo visto che differenti pesi di una rete neurale producono differenti funzioni dell'input; per addestrare una rete neurale, possiamo somministrarle i pattern di input, e poi confrontare la risposta reale del sistema con la risposta *target*, ovvero ciò che la rete dovrebbe dire. La differenza tra questi due valori rappresenta l'**errore** (Floreano, 1996). Il ruolo dell'algoritmo di apprendimento è quello di aggiustare la matrice dei pesi in modo da ridurre l'errore che la rete compie sui pattern; in Figura 1 troviamo un esempio di come si presenta un algoritmo di apprendimento.

*Figura 1* - Esempio di algoritmo di apprendimento.

Il compito di un algoritmo di apprendimento è quello di aggiustare i pesi  $W$  in modo da ridurre la misura dell'errore  $err(p)$ , che rappresenta l'errore compiuto dalla rete sul pattern  $p$ . Vediamo che la matrice dei pesi  $W$ , all'istante di tempo, è rappresentata dalla matrice stessa all'istante di tempo precedente ( $W_{ij}(t)$ ), sommata al prodotto dell'errore compiuto sul generico pattern  $p$  per  $\eta$ ; (ricordiamo che  $\eta$  rappresenta il  $\eta$  learning rate o "tasso di apprendimento"). "Adaline" è l'acronimo per ADAPtive LINear Element. Questo algoritmo è stato sviluppato da Bernard Widrow e Marcian Hoff (Widrow & Hoff, 1960). La regola Adaline, conosciuta anche come regola Delta, minimizza l'errore utilizzando una procedura di discesa dal gradiente. Una volta che i

pattern sono stati presentati alla rete, la correzione da applicare ai pesi è proporzionale all'errore. Considerando una rete neurale di tipo feedforward, avente unità di output ad attivazione lineare. In formula

L'obiettivo, come abbiamo già visto, consiste nel rendere l'uscita effettiva della rete uguale all'uscita desiderata. Dato un gruppo di pattern di addestramento composto da  $M$  coppie formate dal vettore di ingresso  $x$  e dal vettore di risposta desiderata  $t$ , questo può essere visualizzato come

Le prestazioni generali della rete vengono quindi descritte da una funzione di errore  $E$ , o "funzione di costo":

questa rappresenta lo scarto quadratico medio tra la risposta desiderata e la risposta ottenuta per ciascuna unità della rete sommato su tutte le unità e su tutte le coppie di apprendimento. La frazione serve a semplificare alcuni calcoli nelle derivazioni successive. L'impiego di unità lineari permette di riscrivere la funzione di errore nel seguente modo:

L'errore diminuisce quanto più le risposte della rete si avvicinano al target. Il gradiente (e quindi la direzione di massima crescita) della funzione è individuato da un vettore di derivate parziali di  $E$  rispetto a ciascun peso sinaptico  $w_{ij}$ , il che sta a indicare che la variazione della funzione  $E$  dipende unicamente dal valore delle connessioni sinaptiche  $\mathbf{W}$ . L'obiettivo dell'algoritmo è quello di minimizzare lo scarto quadratico medio tra la risposta desiderata e la risposta ottenuta, ed è necessario quindi che la modifica dei pesi sinaptici avvenga **contro** il gradiente di  $E$ , nel seguente modo:

La derivata deve essere risolta per ogni unità, ottenendo la seguente formula per la modifica dei pesi sinaptici:

in cui la modifica dei pesi dipende dalla differenza tra la risposta desiderata e l'output della rete moltiplicata per l'attività presinaptica ( $x_j$ ). Generalizzando e aggiungendo il learning rate abbiamo

Ricordiamo inoltre che l'apprendimento può avvenire per cicli e per epoche; nel caso di apprendimento per cicli la modifica viene calcolata e addizionata ai pesi sinaptici per ogni coppia di addestramento; nel caso invece di apprendimento per epoche i pattern vengono presentati tutti alla rete e le modifiche progressivamente addizionate. Al termine della presentazione dell'intero training set la somma totale viene applicata ai pesi della rete. Il nome dell'algoritmo deriva dal fatto che esso è basato sulla differenza  $\Delta$  tra la risposta desiderata e la risposta ottenuta, e la formulazione matematica viene solitamente abbreviata in

Immaginando la funzione di errore come uno spazio n-dimensionale, in cui ciascuna dimensione rappresenta lo spazio di possibile variazione di ciascuna sinapsi, la superficie presenta un aspetto variegato caratterizzato da minimi e massimi (Floreano, 1996). Nel caso di pattern linearmente indipendenti, la funzione  $E_w$  per unità lineari appare come avente un minimo in cui  $E_w = 0$ . Nel caso invece di pattern non linearmente indipendenti la regola delta può portare a minimi in  $E_w$  è superiore a 0, trovando quindi una soluzione parziale.

### **Bibliografia**

Floreano, D. & Mattiussi, C. (1997). Manuale sulle Reti Neurali. Bologna: il Mulino.

Minsky, M. & Papert, S. (1969), Perceptrons. Cambridge, MA: MIT Press.

Nilsson, N.J. (1998). Intelligenza Artificiale. Milano: APOGEO.

Rosenblatt, F. (1962). Principles of Neurodynamics, New York, Spartan Books.

Widrow, B. & Hoff, M.E. (1960). Adaptive switching circuits, in IRE WESCON Convention Record, vol. IV, pp. 96-104, ristampato in Anderson e Rosenfeld (1988).

[permalink: http://www.neuroscienze.net/index.asp?pid=idart&cat=3&arid=331](http://www.neuroscienze.net/index.asp?pid=idart&cat=3&arid=331)