Lezione 4
Apprendimento Automatico: Approfondimenti

Vito Roberto
Dipartimento di Informatica, Università di Udine
E-mail: vito.roberto@uniud.it

1.- Richiami

Riportiamo in Figura 1 l'architettura di una generica rete multistrato senza specificare il numero di strati e nodi per strato. Ci proponiamo di progettare una procedura di apprendimento basata su una regola che generalizzi la regola a correzione di errore ricavata per le reti monostrato (Lez. 2, par.4).

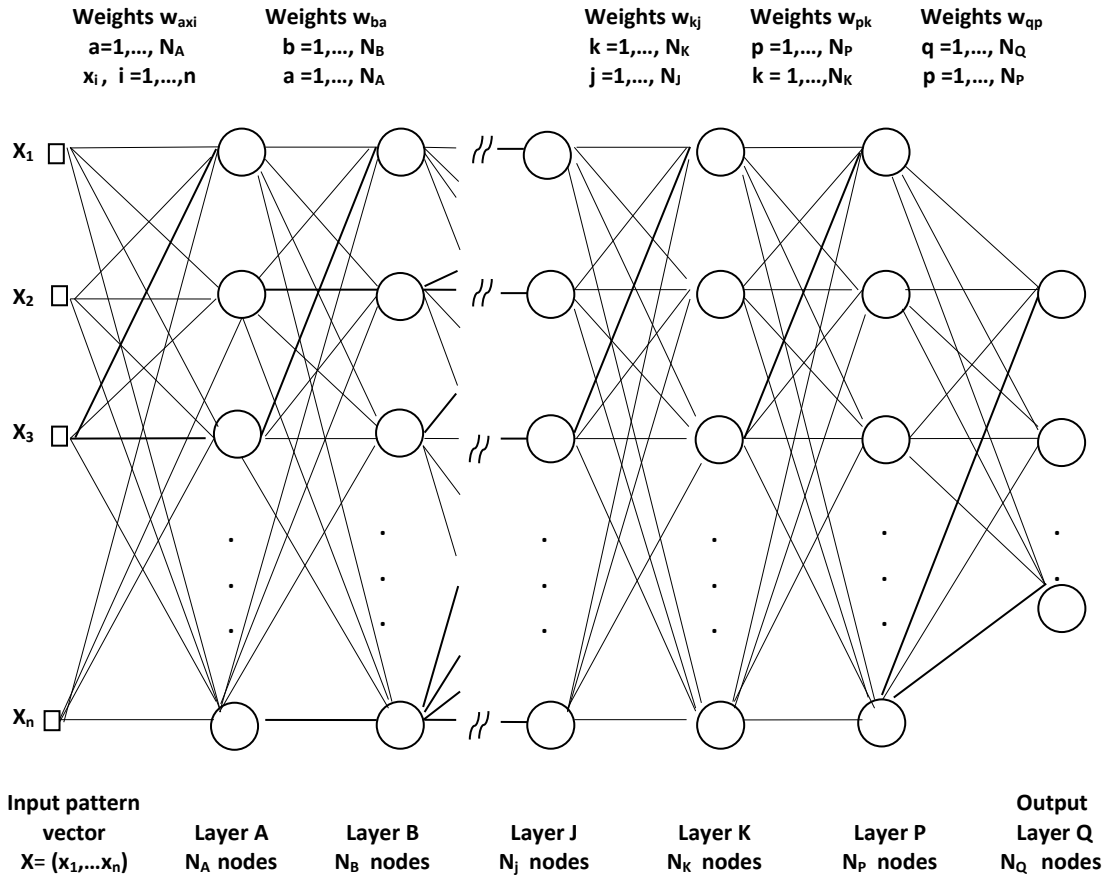

Figura 1

2.- La tecnica di backpropagation

Nel seguito faremo riferimento alla Figura 1, salvo diverso avviso. Procederemo alla valutazione degli errori adottando anche in questo caso la modalità supervisionata. Partiremo dallo strato di output e procederemo all'indietro attraverso gli strati intermedi (backpropagation).

2.1 Analisi dei valori di output

Consideriamo una rete multistrato come in Fig.1. Lo strato di output ha l'indice Q, ed è costituito da N_Q nodi denotati ciascuno con l'indice q.
L'errore quadratico complessivo su tutti i nodi dello strato di output si scrive:

$$E_Q = \frac{1}{2}\sum_{q=1}^{N_Q}(yc_q - O_q)^2 \qquad (1)$$

…dove yc_q e O_q sono rispettivamente la risposta corretta e quella osservata al nodo q; il fattore $\frac{1}{2}$ è usato per semplificare gli sviluppi successivi.
L'obiettivo è progettare una procedura supervisionata, a correzione di errore, analoga a quella basata sulla delta rule ricavata per un perceptron monostrato con funzione di attivazione a gradino (hard-limiter). Cercheremo una espressione più generale, in cui la funzione di attivazione abbia una forma generica $H_q(I_q)$ e non necessariamente la hard-limiter.
Richiamiamo la (2) della Lez.3, par.3: i pesi devono essere modificati in modo proporzionale alla derivata parziale dell'errore complessivo fatta rispetto ai pesi stessi.

$$\Delta w_{qp} = -\alpha \frac{\partial E_Q}{\partial w_{qp}} \qquad (2)$$

…dove lo strato P precede lo strato Q nell'architettura di rete in Fig.1; α è un numero reale positivo. L'errore E_Q è una funzione dei valori di output O_q ; questi, a loro volta, sono funzioni degli input ai rispettivi nodi q, che denotiamo I_q

Applicando la regola di derivazione delle funzioni composte *(chain rule):*

$$\frac{\partial E_Q}{\partial w_{qp}} = \frac{\partial E_Q}{\partial I_q}\frac{\partial I_q}{\partial w_{qp}} \qquad (3)$$

Richiamiamo qui il calcolo dell'input a un nodo q: esso è una combinazione lineare degli output dei nodi dello strato P che precede:

$$I_q = \sum_{p=1}^{N_p} w_{qp}\, O_p + b_q \qquad (4)$$

Allora, dalla (4)

$$\frac{\partial I_q}{\partial w_{qp}} = \frac{\partial}{\partial w_{qp}} \sum_{p=1}^{N_p} w_{qp}\, O_p = O_p \qquad (5)$$

Sostituendo la (5) nella (3) otteniamo:

$$\Delta w_{qp} = -\alpha \frac{\partial E_Q}{\partial I_q} O_p \qquad (6)$$

$$= \alpha \delta_q O_p \qquad (7)$$

…dove

$$\delta_q = -\frac{\partial E_Q}{\partial I_q} \qquad (8)$$

è il cosiddetto 'termine di errore'. Per calcolare il termine $\frac{\partial E_Q}{\partial I_q}$ usiamo nuovamente la chain rule

$$\delta_q = -\frac{\partial E_Q}{\partial I_q} = \frac{\partial E_Q}{\partial O_q} \frac{\partial O_q}{\partial I_q} \qquad (9)$$

Siccome per la (1)

$$\frac{\partial E_Q}{\partial O_q} = -(yc_q - O_q) \qquad (10)$$

…e ricordando che l'uscita da un nodo q è legata all'ingresso I_q dalla funzione di attivazione $H_q(I_q)$

$$O_q = H_q(I_q) \qquad (11)$$

$$\frac{\partial O_q}{\partial I_q} = \frac{\partial}{\partial I_q} H_q(I_q) = H'_q(I_q) \qquad (12)$$

Sostituendo la (10) e la (12) nella (9) si ricava

$$\delta_q = (yc_q - O_q) H'_q(I_q) \qquad (13)$$

….dove δ_q è il termine di errore. Tornando alla (7) e sostituendo

$$\Delta w_{qp} = \alpha \delta_q O_p \qquad (14)$$

$$= \alpha (yc_q - O_q) H'_q(I_q) O_p \qquad (15)$$

Una volta che sia stata specificata la funzione di attivazione $H_q(I_q)$**,** tutti i termini nella (15) sono noti, oppure si possono osservare nella rete. In particolare, dopo aver presentato in input alla rete un vettore (Fig.1) per l'apprendimento, conosciamo la risposta corretta yc_q che deve comparire sui nodi di output. Inoltre, si possono osservare: i valori di input I_q ai nodi dello strato Q; i valori di attivazione O_q sui nodi di output; i valori O_p sui nodi dello strato precedente P. Quindi sappiamo come modificare i pesi dei legami che connettono i nodi del penultimo strato P con quelli di output.

2.2 Analisi degli strati precedenti

Procediamo all'indietro per analizzare l'errore ai nodi del penultimo strato P. Con riferimento alla Figura 1 denotiamo il terzultimo strato con K, e i rispettivi nodi con l'indice k = 1,…,N_k

Procedendo allo stesso modo che al paragrafo precedente:

$$\Delta w_{pk} = \alpha \delta_p O_k \qquad (16)$$

$$= \alpha (yc_p - O_p) H'_p(I_p) O_k \qquad (17)$$

Nella (17) però non tutti i termini sono noti o si possono osservare nella rete. Infatti non disponiamo del valore corretto yc_p sui nodi interni, ma soltanto sui nodi di output, che ci danno la risposta corretta al problema di classificazione. Se li conoscessimo non ci sarebbe bisogno di ulteriori strati nell'architettura di rete. Perciò dobbiamo rielaborare il termine di errore δ_p per esprimerlo in termini che conosciamo o possiamo rilevare.

Scriviamo il termine di errore per lo strato P analogamente alla (9)

$$\delta_p = -\frac{\partial E_P}{\partial I_p} = \frac{\partial E_P}{\partial O_p}\frac{\partial O_p}{\partial I_p} \qquad (18)$$

Come nel paragrafo precedente,

$$\frac{\partial O_p}{\partial I_p} = \frac{\partial}{\partial I_p} H_p(I_p) = H'_p(I_p) \qquad (19)$$

L'altro termine nella (18)

$$\frac{\partial E_P}{\partial O_p} = -\sum_{q=1}^{N_Q} \frac{\partial E_P}{\partial I_q}\frac{\partial I_q}{\partial O_p} = \sum_{q=1}^{N_Q}\left(-\frac{\partial E_P}{\partial I_q}\right)\frac{\partial}{\partial O_p}\sum_{p=1}^{N_p} w_{qp}\, O_p$$

$$= \sum_{q=1}^{N_Q}\left(-\frac{\partial E_P}{\partial I_q}\right) w_{qp}$$

$$= \sum_{q=1}^{N_Q} \delta_q\, w_{qp} \qquad (20)$$

Otteniamo infine l'espressione del termine di errore che cercavamo:

$$\delta_p = H'_p\left(I_p\right)\sum_{q=1}^{N_Q} \delta_q\, w_{qp} \qquad (21)$$

Il termine di errore (21) ai nodi dello strato P può essere calcolato, perché i suoi componenti sono noti. Le due equazioni (19) e (20) definiscono completamente la regola di apprendimento per i nodi dello strato P. In particolare, osserviamo che il termine di errore stesso si calcola a partire da δ_q e dai pesi w_{qp}, cioè da termini che erano stati già calcolati per lo strato immediatamente successivo.

Allo stesso modo, dopo aver calcolato il termine di errore e i pesi per lo strato P, possiamo procedere al calcolo dei termini per lo strato precedente K. Dunque, si è trovato un modo per propagare all'indietro (backpropagation) il calcolo dell'errore lungo la rete, a partire dallo strato di output.

3.- La tecnica di backpropagation

Siamo ora in grado di individuare una procedura di apprendimento per architetture multistrato.

3.1 Uno schema di procedura

Riassumiamo la regola di apprendimento.

(a) Per due strati generici J e K, in cui J precede immediatamente K, si aggiornino i pesi w_{kj}, secondo la regola di modifica:

$$\Delta w_{kj} = \alpha\delta_k O_j$$

(b) Se lo strato K è di output, allora il termine di errore assume la forma:

$$\delta_k = (yc_k - O_k)H'_k(I_k) \qquad (22)$$

(c) Se lo strato K è interno, e P è lo strato immediatamente successivo, allora il termine di errore assume la forma:

$$\delta_k = H'_k(I_k)\sum_{p=1}^{N_P}\delta_p\, w_{kp} \qquad \text{per } k = 1,\ldots,N_k \qquad (23)$$

3.2 Uno sviluppo possibile

Svolgiamo il calcolo nel caso particolare della funzione di attivazione sigmoide, definita nella Lez.1, par.5,

$$\Phi(z) = \frac{1}{1 + e^{-z}}$$

..dove con le attuali notazioni la variabile z si riscrive

$$z = I_k = \sum_{j=1}^{N_J}(w_{kj}O_j + b_k)$$

…e la funzione di attivazione

$$H_k(I_k) = \Phi(I_k)$$

Perciò la sua derivata

$$H'_k(I_k) = O_k(1 - O_k) \qquad (24)$$

Quindi nel caso della funzione di attivazione sigmoide, avremo per il termine di errore nelle (22) e (23)

$$\delta_k = (yc_k - O_k)O_k(1 - O_k)$$

…per lo strato di output, e

$$\delta_k = O_k(1 - O_k)\sum_{p=1}^{N_P}\delta_p\, w_{kp} \qquad \text{per } k = 1,\ldots,N_K$$

…nel caso di K strato interno.

4.- Complessità computazionale: propagazione in avanti

Ai fini del calcolo della complessità è utile considerare separatamente le fasi di verifica *(test)* e addestramento (*training)*, ovvero della propagazione dell'informazione in avanti (*feedforward*) e all'indietro (*backpropagation*).

La propagazione in avanti richiede la stima del computo dei valori nei singoli nodi della rete, strato per strato. Riferendoci alla Fig. 1, per il generico strato *K* il calcolo degli input dei nodi è dato dalla (4). Ovvero riscrivendola in forma matriciale:

$$I_K = W_K O_{K-1} + b_K \qquad (25)$$

Si noti che nella (25) I_K , b_K sono vettori-colonna di dimensioni $K = 1,\ldots, N_K$;

W_K è una matrice con elementi $\{w_{K,K-1}\}$ dove: $k = 1,\ldots, N_K$; $(k\text{-}1) = 1,\ldots, N_{K-1}$

Gli output dei nodi si ottengono applicando la funzione di attivazione elemento per elemento; qui usiamo la notazione vettoriale relativa all'intero strato *K*

$$O_K = H(I_K)$$

Quindi, per ogni strato sono eseguite: la moltiplicazione di matrici $W_K O_{K-1}$; la somma vettoriale $(+ b_K)$; l'applicazione a ogni nodo della funzione di attivazione.

4.1 Un richiamo

Richiamiamo ora il calcolo di complessità del prodotto righe per colonne di due matrici. Date due matrici:

$$A = \{a_{rz}\} \qquad r = 1, \ldots, N_r \; ; \; z = 1, \ldots, N_z$$

$$B = \{b_{zc}\} \qquad z = 1, \ldots, N_z \; ; \; c = 1, \ldots, N_c$$

Il prodotto tra le due matrici

$$A \cdot B = \sum_{z=1}^{N_z} a_{rz} b_{zc}$$

Il calcolo della matrice prodotto richiede $N_r \times N_z \times N_c$ prodotti e $z - 1$ somme. Se passiamo in termini asintotici non distinguiamo le singole dimensioni, ovvero nulla cambia ponendo genericamente $N_r = N_z = N_c = n$ ed esprimendo il numero totale di operazioni come $n \times n \times n$. Diremo che la moltiplicazione di matrici è generalmente di ordine $O(n^3)$.

4.2 Complessità del calcolo

Nel caso della rete di Fig. 1 in un generico strato *K* avremo:

$$N_r = N_K \quad N_z = N_{K-1} \quad N_c = N_{K-2}$$

Quindi il numero totale n_{mul} di moltiplicazioni matriciali nella rete è:

$$n_{mul} = \sum_{K=2}^{N_{layers}} N_K N_{K-1} N_{K-2} + N_1 N_0 1$$

$$n_H = \sum_{K=1}^{N_{layers}} N_K$$

ottenendo complessivamente

$$tot_{time} = n_{mul} + n_H = \sum_{K=2}^{N_{layers}} N_K N_{K-1} N_{K-2} + N_1 N_0 1 + \sum_{K=1}^{N_{layers}} N_K$$

Passando ai termini asintotici

$$n_{mul} = N_{layers} \cdot n^3$$

Dunque:

$$n_{mul} = O(n \cdot n^3) = O(n^4)$$

Dato che l'applicazione della funzione di attivazione avviene singolarmente per ogni nodo avremo

$$n_H = N_{layers} \cdot n = O(n^2)$$

e quindi complessivamente

$$O(n^4 + n^2) \cong O(n^4)$$

5.- Complessità della propagazione all'indietro

Esaminiamo ora la fase di backpropagation, ricordando che il computo del termine di errore per uno strato generico *K* è dato dalle (22), (23):

$$\delta_k = \begin{cases} H'_k(I_k)(yc_k - O_k) & se\ k \in K = Q \\ H'_k(I_k) \sum_{p=1}^{Np} \delta_p w_{kp} & se\ k \in K < Q\ e\ P = K+1 \end{cases}$$

Per quanto ricavato al paragrafo 4.1 avremo:

$$H'_k(I_k)(yc_k - O_k) = O(n^2)$$

$$H'_k(I_k) \sum_{p=1}^{Np} \delta_p w_{kp} = O(n^3)$$

…e il tempo totale impiegato nel computo dei termini di errore è asintoticamente

$$\mathcal{O}(time_E) = n^2 + n^3(Q-1)$$

dove Q è il numero di strati della rete. Ponendo ogni dimensione pari a n si ottiene

$$time_E = \mathcal{O}(n^4)$$

Il calcolo dell'aggiornamento dei pesi per ogni strato, ottenuto moltiplicando le matrici dei pesi per i termini di errore, è:

$$\mathcal{O}(time_W) = \mathcal{O}(time_E) + n^3 = n^4$$

Considerando la procedura di correzione di errore basata sulla discesa lungo il gradiente (GD):

$$time_{GD} = n_{iterations} \cdot time_W$$

… e considerando asintoticamente n applicazioni della procedura:

$$\mathcal{O}(time_{GD}) = n \cdot n^4 = n^5$$

Quindi la complessità asintotica della procedura di backpropagation è $\mathcal{O}(n^5)$.