Cap. 4 Ricostruzione delle tracce con reti neurali

4.1 Introduzione alle Reti Neurali

4.1.1 Neuroni

Le reti neurali si ispirano al funzionamento dei neuroni biologici, le cellule alla base dell'elaborazione nel cervello animale.

Perfino il cervello degli insetti, molto più piccolo e più semplice di quello umano, contiene milioni di queste cellule elementari, mentre quello umano ne contiene centinaia di miliardi. Il cervello più piccolo in natura è probabilmente quello del verme nematode che ha circa 300 neuroni.

La caratteristica fondamentale del cervello è l'alto grado di connessione tra i neuroni: ognuno ha migliaia di "ingressi" e migliaia di "uscite"; l'immagazzinamento dell'informazione e la capacità elaborativa sono legate alle intensità di queste connessioni, cioè come si vedrà a dei pesi che modulano i segnali in ingresso al neurone. In Figura 67 è riportata una rappresentazione schematica di un neurone: gli ingressi sono in corrispondenza dell'albero dendritico, mentre le ramificazioni dell'assone sono le uscite; i neuroni sono quindi caratterizzati dall'unidirezionalità del flusso di informazione.

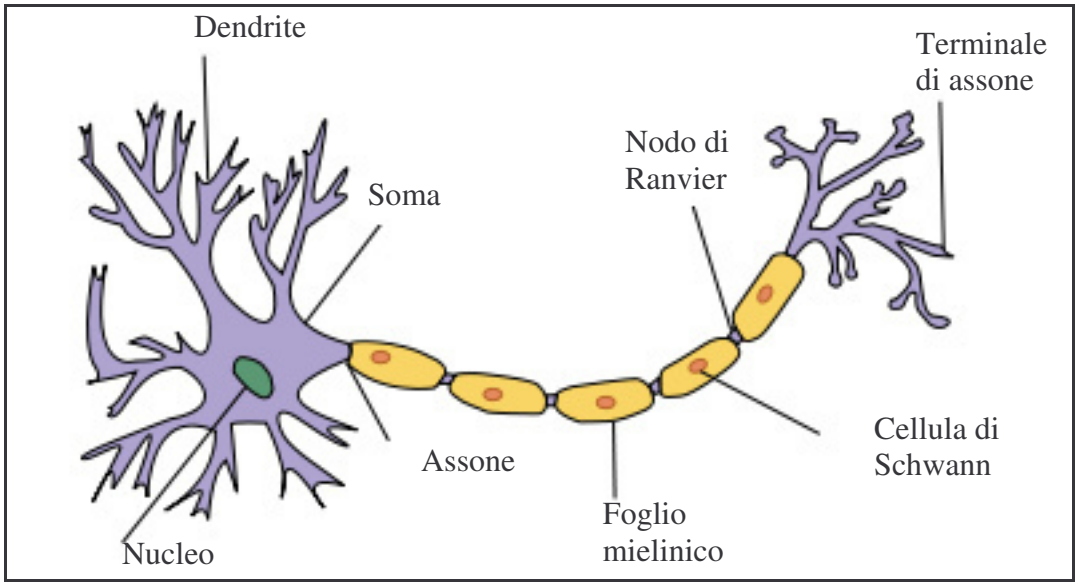

Figura 67 - Schema di un neurone.

Le uscite di un neurone vanno ad alimentare ingressi di altri neuroni, attraverso interfacce dette sinapsi, costituendo quindi una vera e propria rete. Ci sono neuroni afferenti, cioè legati a organi di senso, neuroni intermedi, che hanno per ingresso le uscite di altri neuroni e alimentano altri neuroni ancora, e ci sono neuroni efferenti, che trasmettono segnali alle cellule di organi motòri.

La trasmissione dell'informazione all'interno di un neurone avviene tramite impulsi elettrici, condotti attraverso la guaina mielinica. La trasmissione sinaptica invece è di tipo chimico; se il segnale elettrico che giunge all'assone supera il potenziale di azione vengono emesse in corrispondenza delle sinapsi molecole di neurotrasmettitori che attivano i dendriti dei neuroni successivi.

4.1.2 Elaborazione neurale

Un semplice modello di neurone, che ricalca quanto descritto precedentemente, è riportato in Figura 68

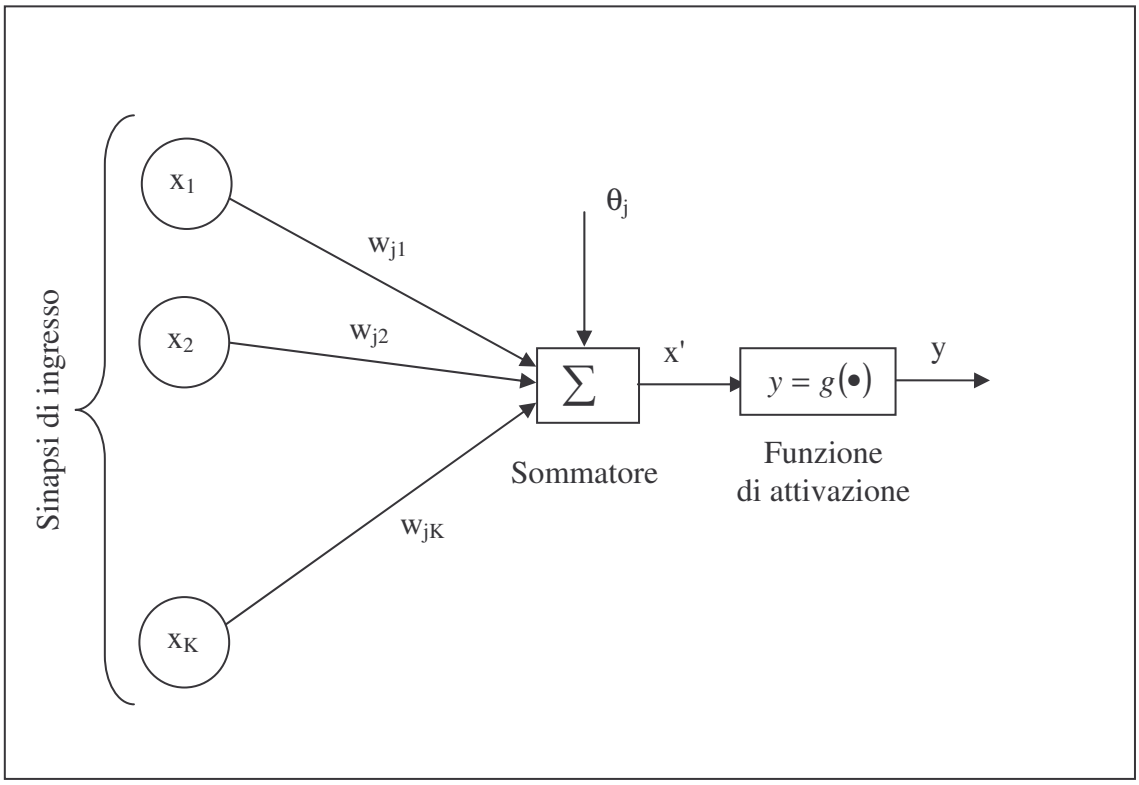

Figura 68 - Modello di neurone.

Questo è composto da:

- un insieme di sinapsi di ingresso x_1, x_2, ... x_K, ognuna caratterizzata da un'ampiezza o forza w_{j1}, w_{j2}, ... w_{jK};
- un sommatore di tutte le sinapsi di ingresso pesate con la propria ampiezza, con un eventuale offset (soglia)
- una funzione di attivazione (di solito non-lineare)

Nella figura l'uscita di un neurone k (x_k) è collegata all'ingresso del neurone rappresentato, e l'ampiezza della sinapsi è indicata dal peso w_{jk}.

Matematicamente l'uscita y_j del neurone j può essere espressa come:

$$y_j = g\left(\sum_{k=1}^{K} w_{jk} x_k - \theta_j\right)$$

In pratica ogni singolo neurone di una rete funziona come un interruttore che si accende se la somma pesata di tutti gli ingressi supera la soglia di attivazione [38].

4.1.3 Funzioni di attivazione

Le funzioni di attivazione utilizzate nelle implementazioni di reti neurali sono di vari tipi, ma tutte hanno due caratteristiche in comune:

- saturano ad un valore minimo e un valore massimo
- sono non-decrescenti

Alcune funzioni tipicamente usate in letteratura sono:

1. Funzione soglia, definita come

$$g(x) = \begin{cases} 1 & se \quad x \geq 0 \\ 0 & se \quad x < 0 \end{cases}$$

   che definisce quindi un interruttore ideale (Figura 69 a)

2. Funzione lineare a tratti, dove le due saturazioni sono raccordate da un tratto rettilineo (Figura 69 b):

$$g(x) = \begin{cases} 1 & se \quad x \geq 1/2 \\ x + 1/2 & se \quad -1/2 < x < 1/2 \\ 0 & se \quad x \leq -1/2 \end{cases}$$

3. Funzione sigmoide, una delle funzioni di attivazione più usate, anche perché la sua caratteristica di essere derivabile ovunque facilita la convergenza degli algoritmi di apprendimento

$$g(x) = \frac{1}{1 + e^{-a\,x}}$$

   L'effetto del parametro *a* è evidente nella Figura 69 (c).

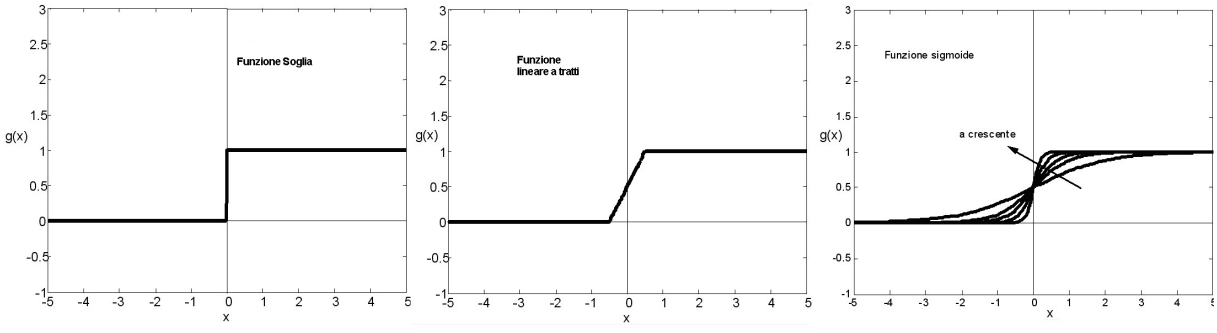

Figura 69 - Funzioni di attivazione

Questi tre esempi di funzione di attivazione sono definiti tra 0 ed 1; per alcuni scopi è comodo definire la funzione tra -1 e 1, e questo può essere fatto mantenendone la forma.

4.1.4 Architetture di reti neurali

Single layer Feed-forward

L'architettura più semplice per una rete neurale è composta da uno strato di N neuroni ad ognuno dei quali sono collegati K ingressi attraverso i pesi w_{nk} come rappresentato in Figura 70

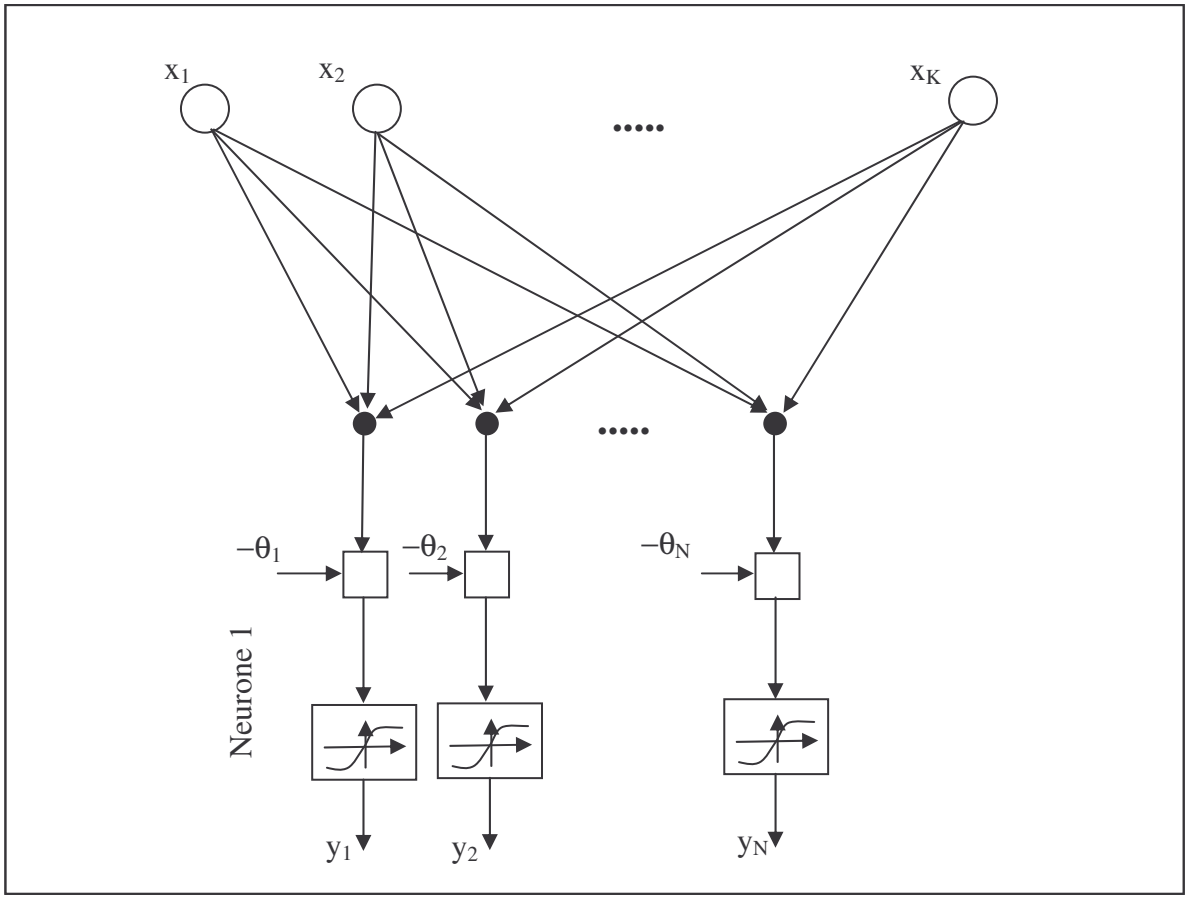

Figura 70 - Schema di rete single layer feed-forward.

Questa struttura è molto semplice e permette di rappresentare delle mappe cosiddette a classi linearmente separabili, ma fallisce su problemi non-linearmente separabili, di cui l'esempio più classico è la funzione XOR [40].

Multilayer feed-forward

Un'architettura più generale, in grado di superare le limitazioni viste per il Single Layer è naturalmente una struttura a più livelli, in cui oltre allo strato di ingresso e allo strato

di uscita, vi siano uno o più livelli intermedi (nascosti o hidden). Questo tipo di architettura è detta multilayer perceptron.

In effetti è stato dimostrato che un'architettura con due livelli, (uno hidden) e funzioni di attivazione non-lineari è in grado di implementare qualsiasi funzione di interesse pratico [41], e comunque aggiungere livelli interni (a parità di gradi di libertà) non aggiunge capacità di elaborazione alla rete.

Quindi uno schema sufficientemente generale di rete neurale Multi Layer Feed Forward NN è quello riportato in Figura 71

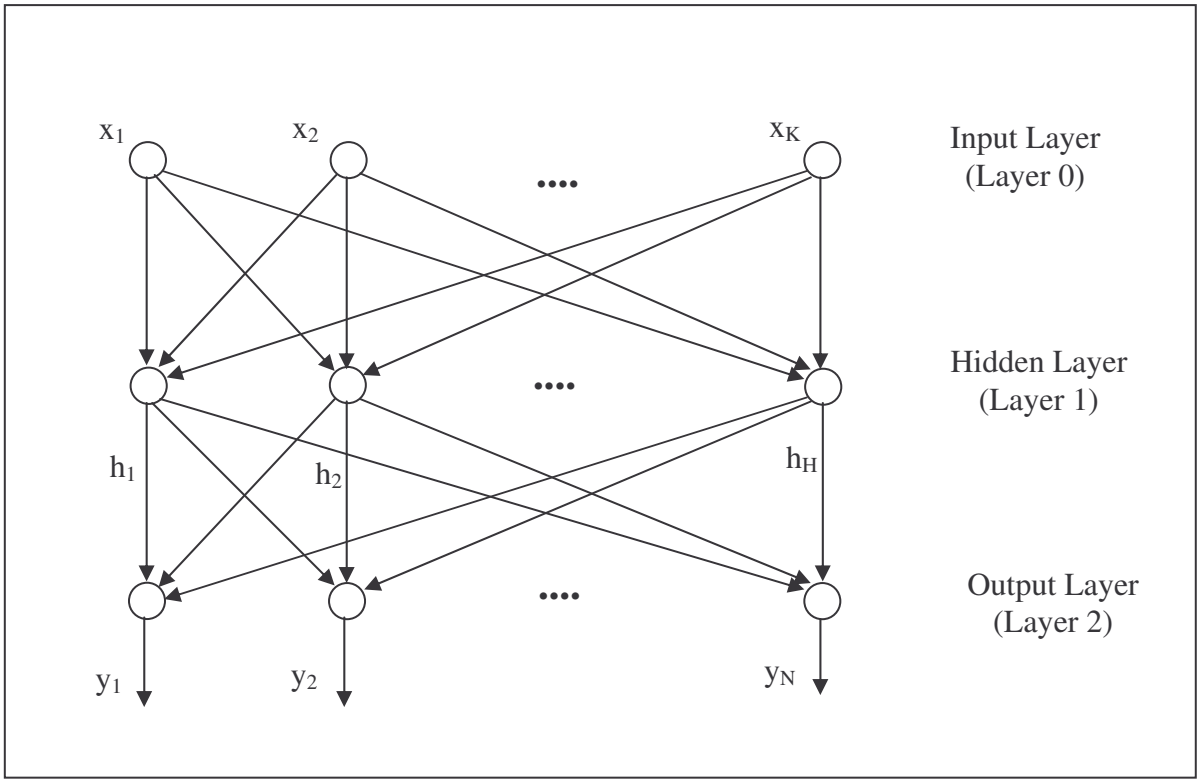

Figura 71 - Schema di rete Multi layer feed forward.

Qui è rappresentata una rete con K ingressi, H neuroni hidden e N uscite.

Questa architettura di rete neurale, sebbene fosse nota già dagli anni '60 e fosse già stato dimostrato che avrebbe superato i limiti della struttura a singolo layer, è stata utilizzata estensivamente solo dopo la scoperta di un efficace metodo per l'apprendimento: il cosiddetto back-propagation algorithm, degli anni '80 [42], di cui si parlerà in seguito.

Altri tipi di architettura

Esistono altri tipi di architetture per le reti neurali, ognuna con delle particolarità che la rendono più o meno utile a seconda dell'applicazione. In effetti la struttura classica Multi Layer Feed Forward è la più versatile, essendo buona per ogni tipo di applicazione.

Le Recurrent NN sono reti in cui è permesso il ritorno delle uscite verso gli ingressi; c'è praticamente la stessa differenza che corre tra i filtri FIR (Finite Impulse Response) e quelli IIR (Infinite Impulse Response).

Le Radial Basis Function NN sono più utili per applicazioni in cui si richiede l'approssimazione di una funzione più che la classificazione di un pattern; sono essenzialmente uguali alla multi layer, ma le funzioni di attivazione sono a simmetria radiale (per esempio una gaussiana), quindi i neuroni intervengono localmente (intorno al picco della gaussiana).

Le Adaptive Resonance Theory NN sono delle reti neurali con struttura più complessa, che permettono una certa adattabilità a nuove tipologie di ingressi, e quindi l'applicabilità a segnali non stazionari [43].

4.1.5 Metodologie di apprendimento

Le reti neurali hanno bisogno di essere addestrate per fornire le risposte corrette. L'addestramento viene condotto scegliendo un set di dati di ingresso, di cui è noto il risultato desiderato dell'elaborazione. I dati di training vengono presentati ad una rete inizializzata con valori random dei parametri, e le uscite ottenute vengono confrontate con le uscite desiderate. I pesi della rete vengono quindi corretti iterativamente per minimizzare l'errore di uscita.

Esistono tipicamente tre metodologie di apprendimento per una rete neurale: apprendimento supervisionato, apprendimento non supervisionato e apprendimento ibrido.

- Apprendimento supervisionato (supervised training)

La caratteristica di questo tipo di apprendimento è la presenza di un insegnante esterno alla rete. L'insegnante fornisce alla rete un set di dati di training di cui sono note le risposte esatte. I pesi della rete sono ottimizzati in base all'errore che questa commette in successive e reiterate prove.

- Apprendimento non supervisionato (unsupervised training)

Nell'apprendimento non supervisionato, non conoscendo le risposte ai dati di training, si deve individuare una misura della bontà della rappresentazione che la rete deve imparare. Alla fine del training la rete sarà in grado di raggruppare dati omogenei rispetto al set di parametri utilizzati per definire la rappresentazione.

- Apprendimento ibrido

In alcuni casi è conveniente utilizzare entrambi i metodi descritti o in cascata, per diminuire il set di apprendimento supervisionato, o in parallelo, per migliorare le prestazioni dell'apprendimento stesso.

Tutte queste tecniche possono essere implementate tramite l'algoritmo di retro-propagazione dell'errore, (Back-propagation algorithm [42]). Dato un insieme di ingressi di training **x** e le relative uscite desiderate **d**, si può calcolare la funzione d'errore

$$E(\mathbf{w}) = \sum_{1=1}^{N} (d_i - y_i)^2$$

dove **w** rappresenta la matrice dei pesi e y_i le uscite.
Ora, assumendo che la funzione d'errore sia differenziabile, si possono applicare le usuali tecniche di ottimizzazione, per esempio la tecnica del *gradient descent*. Si aggiornano i pesi di una quantità proporzionale al gradiente della funzione d'errore:

$$\Delta\mathbf{w} = -\eta \nabla E(\mathbf{w})$$

per poi procedere ad una successiva iterazione.
Oltre a MINUIT, anche in MatLAB esistono già implementate molte funzioni che applicano questa tecnica, con delle piccole varianti che ottimizzano alcuni aspetti [44].

Quando la dimensione del set di training è piccola (rispetto al numero di pesi totali da ottimizzare), la convergenza del gradient descent è molto veloce, ma in questo caso la rete non è in genere in grado di generalizzare (a meno di un criterio di riconoscimento molto semplice).
Una buona regola empirica per addestrare correttamente una rete è quella di utilizzare un campione di training di dimensione almeno 10 volte il numero totale di pesi indipendenti; questo implica che, una volta fissato il numero di ingressi e di uscite della rete, la scelta del numero di neuroni dello strato intermedio (H):

$$H = \frac{PT}{10(I+O)}$$

dove PT è la dimensione del set di training, I ed O sono il numero di ingressi e di uscite rispettivamente.

4.1.6 Applicazioni

Le reti neurali trovano applicazioni particolarmente appropriate nel campo di pattern recognition, ad esempio nel riconoscimento vocale, o di immagini, e in tutti i campi in cui sia necessaria una classificazione veloce di segnali complessi, in cui l'applicazione di algoritmi tradizionali sarebbe molto difficile.

4.2 Ricostruzione delle tracce di muoni con reti neurali

Si vuole costruire ed addestrare una rete neurale in grado di prendere in ingresso gli hit di un evento in un telescopio Čerenkov sottomarino (tempi e carica) e restituire in uscita la direzione della traccia del muone (ϕ, θ), come rappresentato in Figura 72. Ci si limita ai parametri (ϕ, θ) con l'intento di ottenere un prefit.

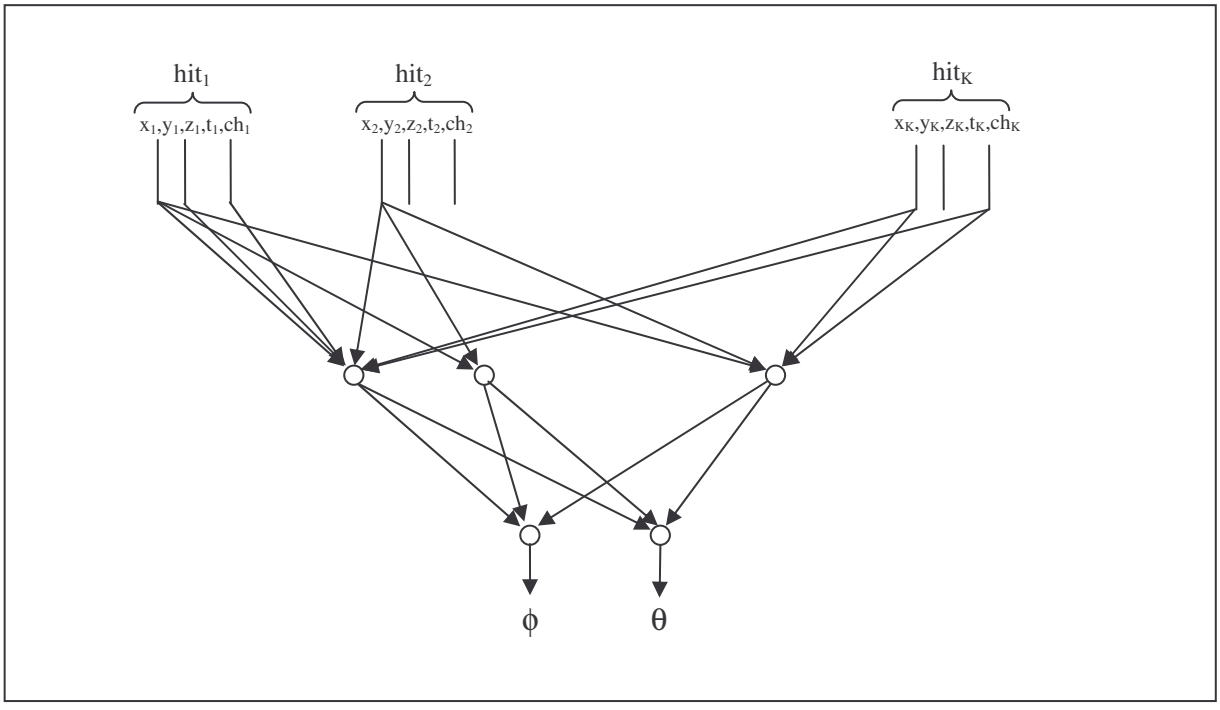

Figura 72 - Schema di principio della ricostruzione di una traccia con una rete neurale.

La scelta della topologia della rete e della modalità con cui fornire gli ingressi relativi agli eventi lascia una certa libertà. Bisogna indagare quali sono i vantaggi e gli svantaggi delle possibili scelte implementative.

Nel nostro caso un evento consiste di un certo numero di hit, da poche unità ad alcune centinaia, ogni hit è contraddistinto da una posizione nello spazio, da un tempo, e da una carica.

La rete più generale possibile probabilmente è quella che ha 2 N_{PMT} ingressi (N_{PMT} è il numero totale di PMT dell'apparato) a cui vanno i tempi e la carica di ogni relativo hit. Nei PMT che non hanno rivelato un segnale si può porre zero carica e un certo codice (p.es. -1) al tempo, posto che tutti gli hit abbiano tempo positivo.

In questa topologia non c'è bisogno di passare alla rete le coordinate degli hit, perché queste verrebbero apprese e memorizzate implicitamente nella rete al momento dell'addestramento; tuttavia il grande numero di detector nell'apparato NEMO (N_{PMT} = 5832) rende questa via poco percorribile, specialmente per la fase di addestramento.

Si possono allora fornire in ingresso alla rete solo gli hit, questa volta con la posizione e il tempo. In questo caso, poiché il numero di ingressi sarebbe variabile da evento a evento, si possono scegliere un certo numero di hit (quelli con carica maggiore) oppure si possono dare alla rete più sequenze di hit e poi mediare le uscite di sequenze appartenenti allo stesso evento come rappresentato in Figura 73. Qui le sequenze di hit sono lunghe K, e per ogni hit vengono passati alla rete le coordinate del PMT e il tempo di hit, quindi la rete avrà 4·K ingressi. Durante l'addestramento, per tutte le sequenze di hit dello stesso evento, il target (cioè le uscite desiderate) saranno i valori veri di ϕ e θ della traccia.

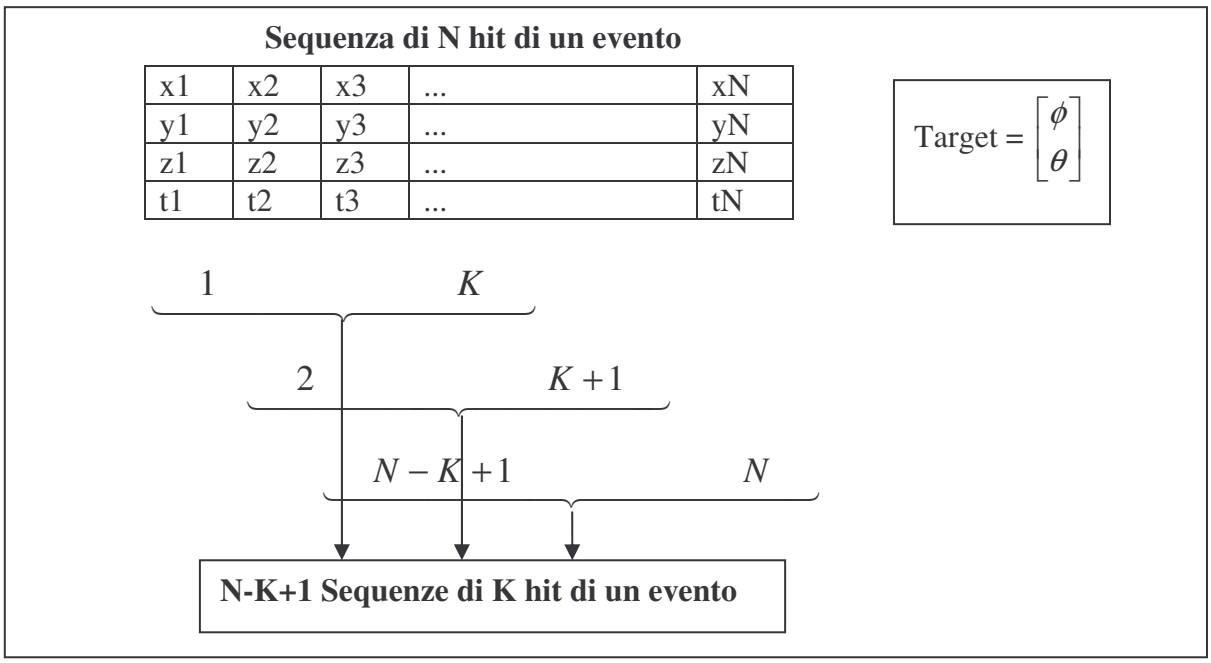

Figura 73 - Sequenze di hit in ingresso alla rete neurale.

Un altro approccio che è stato preso in considerazione è quello di dare in ingresso alla rete gli ingressi differenziali *dx/dt, dy/dt, dz/dt*, cioè le differenze di coordinate tra coppie di hit divise per le differenze di tempo d'arrivo. Anche in questo caso si possono selezionare gli hit con carica maggiore.

Il procedimento utilizzato per la simulazione (cfr. §4.3) è stato quello di generare un file di 1000 eventi con il programma OPNEMO, una parte del quale (da 100 a 200 eventi) è stata utilizzata per ricavare i dati di training, e la parte restante per simulare le ricostruzioni operative.

Le ricostruzioni sono state fatte al variare dei seguenti parametri:

- numero di hit di ogni sequenza in ingresso
- numero dei neuroni intermedi

e sono state generate reti per la stima di theta e phi e solo per la stima di theta (riconoscimento up/down). Il parametro di bontà utilizzato per i confronti è la mediana della distribuzione dell'angolo tra la traccia vera e la traccia ricostruita.

4.3 Risultati della ricostruzione con reti neurali

La ricostruzione mediante reti neurali, descritta in §4.2 , è stata realizzata con il programma Matlab. Inizialmente è stata implementata la ricostruzione con in ingresso le

coordinate e i tempi di arrivo di gruppi di M hit: (x_i, y_i, z_i, t_i). Valori possibili per M sono tra 10 e 30; infatti con pochi hit la ricostruzione è sempre più critica, mentre al crescere del numero di ingressi la memoria e il tempo di elaborazione crescono eccessivamente: cresce molto anche la dimensione del set di training necessario per un buon addestramento, che come detto (§4.1.5) è dell'ordine di $10 \cdot N_W$, dove N_W è il numero totale di parametri liberi della rete.

In Figura 74 è riportato l'istogramma dell'errore angolare per M = 20; la mediana è molto alta anche rispetto al prefit.

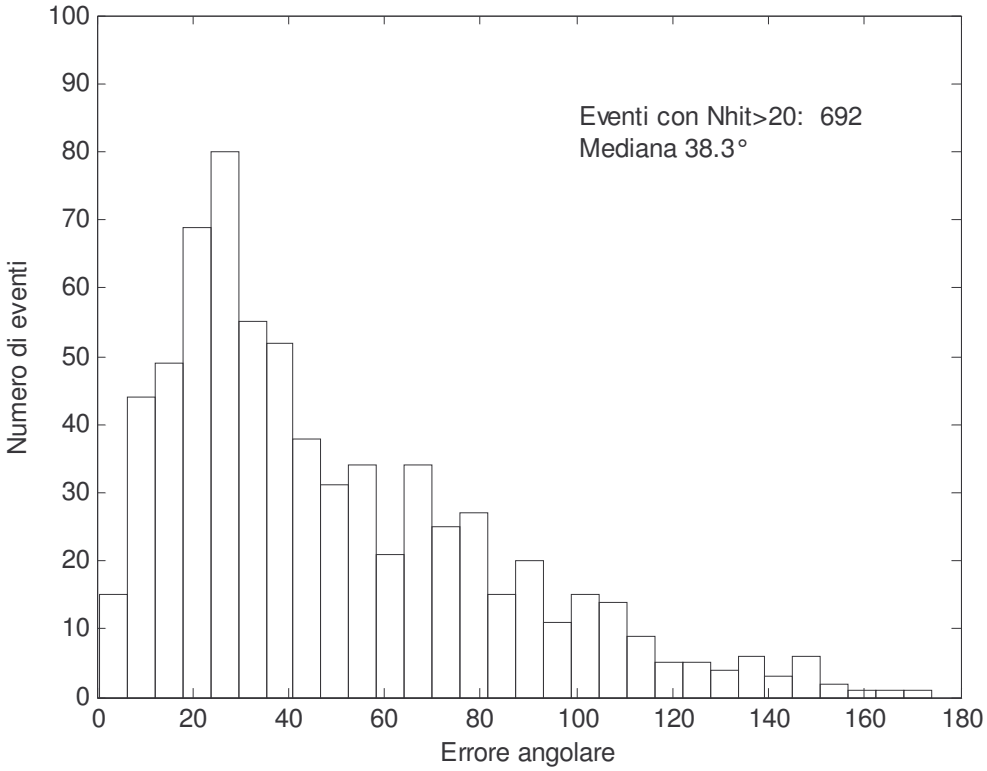

Figura 74 – Angolo nello spazio tra la direzione della traccia simulata e quella ricostruita (M=20).

Aumentando il numero di ingressi (in Figura 75 l'errore angolare per M=30) le performance non migliorano (mediana ≅ 50°), probabilmente perché il set di training non è sufficientemente ampio.

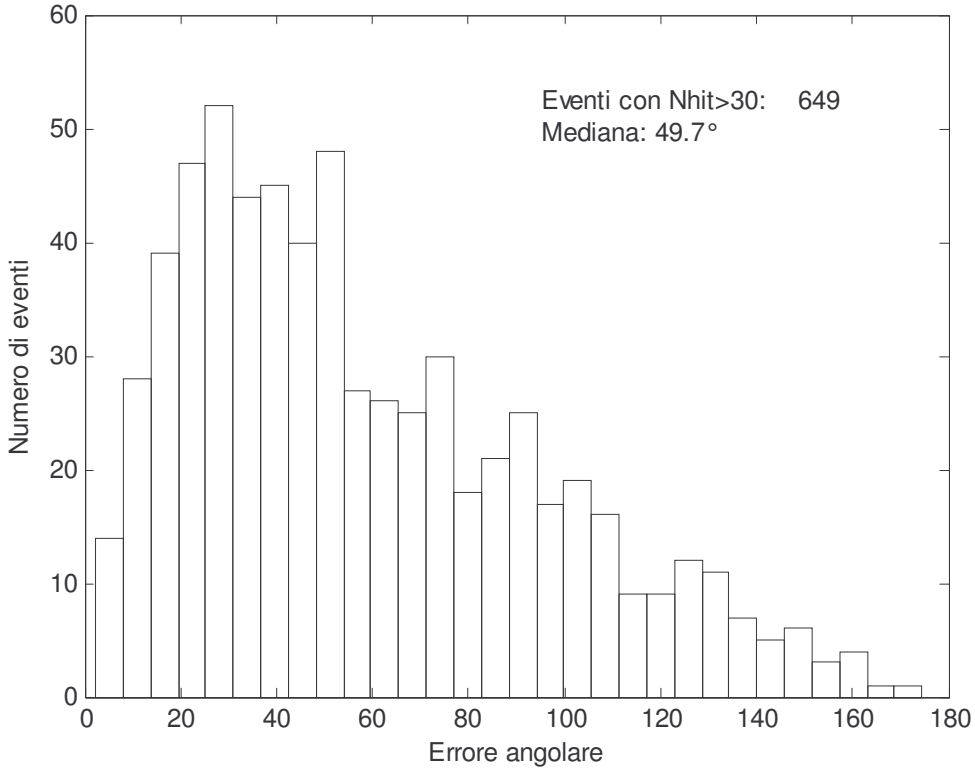

Figura 75 - Angolo nello spazio tra la direzione della traccia simulata e quella ricostruita (M=30).

Per alleggerire il compito della rete è stata tentata la ricostruzione della sola variabile angolare θ, cercando quindi di individuare solamente la direzione Up/Down. Le performance per M=10 e M=20 sono riportate in Figura 76 e in Figura 77 rispettivamente. I risultati sono migliorati, ma ancora sono peggiori del prefit (cfr. Figura 46).

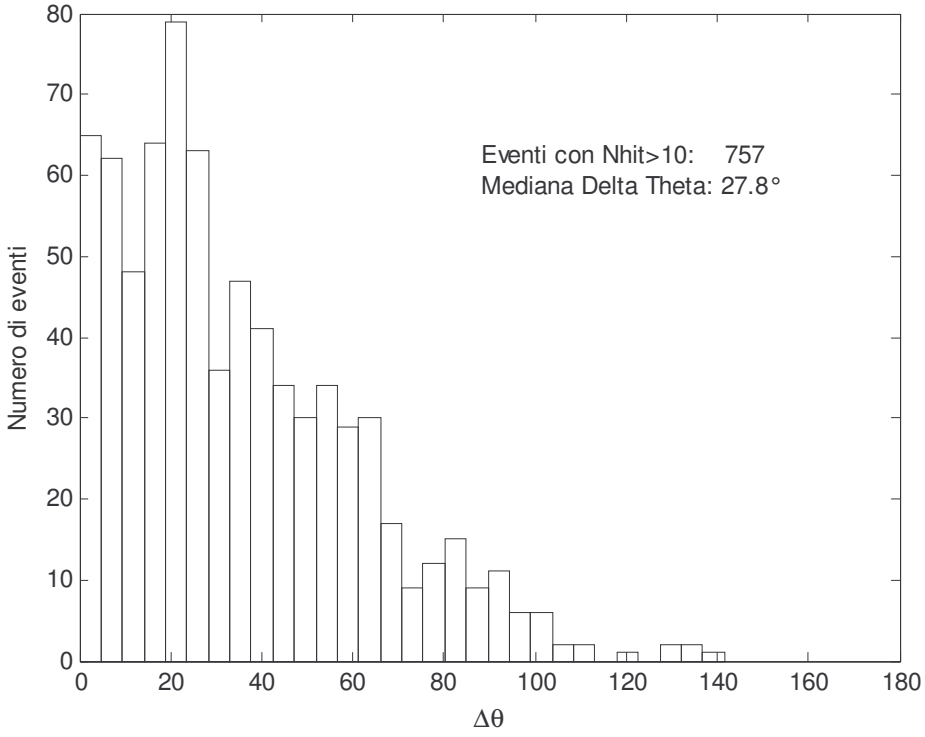

Figura 76 - Ricostruzione dell'angolo θ con rete neurale con M=10.

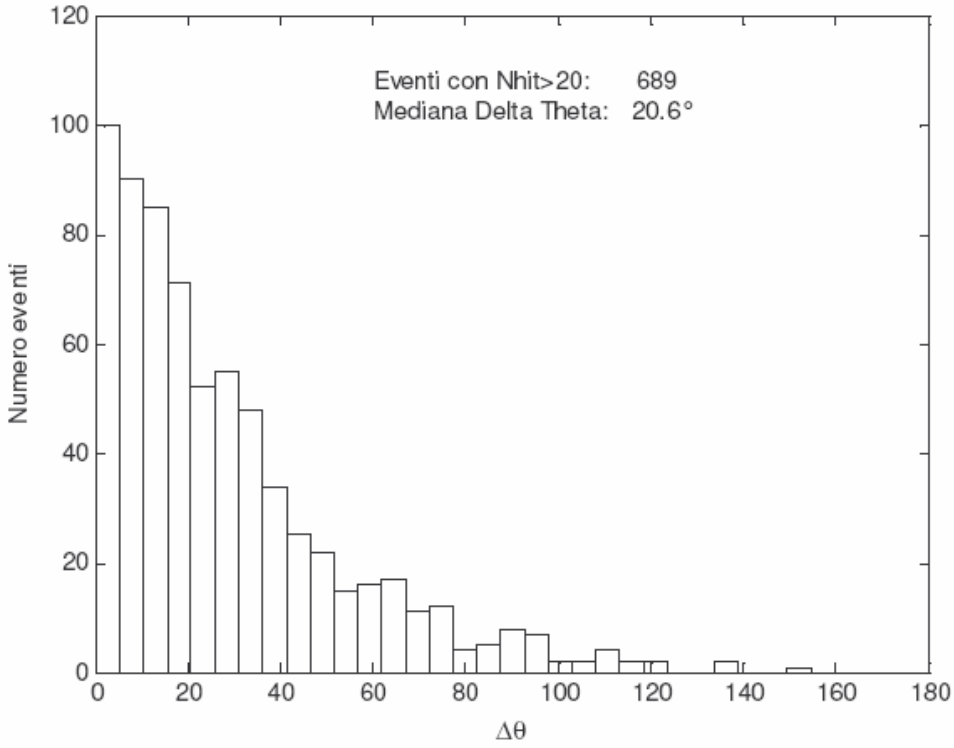

Figura 77 - Ricostruzione dell'angolo θ con rete neurale con M=20.

Un miglioramento ulteriore, che porta la mediana dell'errore in θ a 11.5°, è stato quello di considerare come ingressi alla rete le variabili dx_{ij}/dt_{ij}, dy_{ij}/dt_{ij}, dz_{ij}/dt_{ij}, dove i, j sono gli indici di alcune coppie di hit, opportunamente scelti. In questo caso sono stati scelti gli hit con carica misurata maggiore, e sono state fatte le differenze con tutti gli altri hit. L'istogramma dell'errore in θ è riportato in Figura 78, si ha una accuratezza confrontabile con un buon prefit.

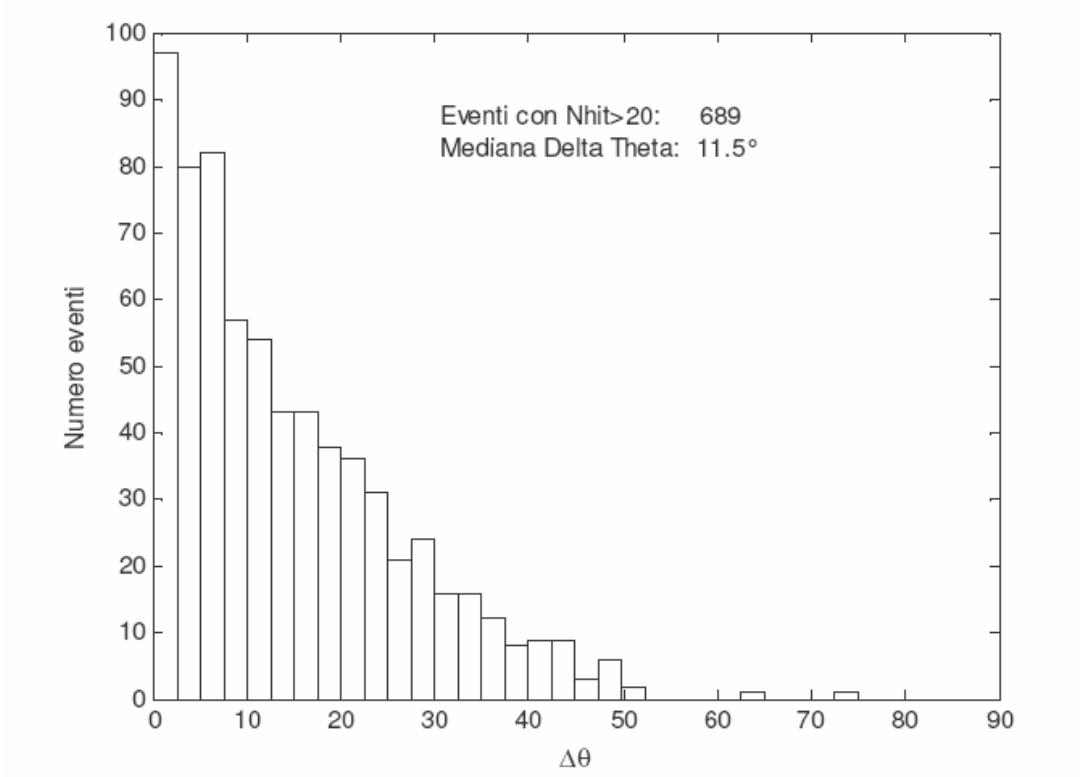

Figura 78 - Errore in θ della ricostruzione con rete neurale con ingressi differenziali e selezione degli hit tramite la carica.