3. Introduzione alla Neuro Computazione

di PK-Lab dicembre 1994

Attorno alla neuro computazione ruotano discipline molto differenti e, spesso, come discusso nell'introduzione, finiscono col prendere posizioni fortemente contrastanti, invece di collaborare nello studio di questi sistemi. Vi sono tuttavia, casi in cui psicologi, neurologi, filosofi tentano di studiare alcune problematiche del sistema nervoso, attraverso la simulazione di neuroni artificiali. Un esempio interessante è offerto dalla realizzazione della Retina di Silicio [Le Scienze 275-1991] condotto da Carver Mead, professore di Scienza dei Calcolatori al Caltech e da Misha A.Mahowald, biologa al Caltech. La Retina di Silicio è un chip in grado di simulare le cellule nervose della retina, tanto da essere soggetto alle illusioni ottiche tipiche dell'occhio umano.

3.1.Le basi biologiche.

Solo in questi anni si sta cominciando a capire come la mente dell'uomo sia in grado di funzionare. Tuttavia, i problemi da affrontare sono molti, si veda ad esempio: la memoria, i sentimenti, l'invecchiamento celebrale, la coscienza ecc..
L'elemento responsabile della nostra mente sembra essere proprio il cervello, che viene visto come un vero e proprio strumento di calcolo al nostro servizio.
In questa ottica, si può pensare al cervello come ad un calcolatore al quale affluiscono gli stimoli dal mondo esterno, attraverso le terminazioni nervose e, dal quale, partono le reazioni a tali stimoli, sempre attraverso i canali del sistema nervoso.
Elemento centrale del nostro cervello sembrano essere le cellule in esso contenuto, note con il nome di neuroni.Queste cellule sono in gran numero nel nostro cervello (circa 100 miliardi) e sono tra loro fittamente collegate. Si pensa che la capacità di calcolo del nostro cervello stia proprio nelle connessioni di tali neuroni e che, l'elaborazione degli stimoli provenienti dal sistema nervoso, avvenga quasi alla cieca, facendo passare, attraverso l'insieme dei neuroni, tali stimoli.
In questo modo, gli stimoli nervosi vengono prelevati dai neuroni che potremmo chiamare "accettori", i quali comunicano ad altri neuroni quello che hanno ricevuto. Questa catena di "passa-parola" si protrae fino a quando non vengono raggiunte delle terminazioni che potrebbero essere chiamate "esecutrici". La diffusione sembra essere un'elaborazione dello stimolo nervoso, che infatti giunge alle terminazioni come una risposta. In questo modo, si spiegano quelle zone del nostro cervello dedicate allo svolgimento di compiti precisi, come la corteccia visiva primaria.
Quindi, la propagazione dello stimolo nervoso tra un neurone ed un altro, provoca un'alterazione elementare dello stesso stimolo. L'insieme di tali deformazioni elementari rappresenta il calcolo operato dal cervello.

Per capire come avvengono le trasformazioni elementari è necessario capire come in realtà comunicano due neuroni.
Un neurone è composto da un nucleo contenuto in un corpo cellulare. A questo nucleo giungono delle terminazioni chiamate dendriti che ricevono lo stimolo nervoso e lo trasmettono al nucleo. Dal nucleo, inoltre, parte un assone che nella parte finale si ramifica e presenta delle sinapsi.
La connessione tra due neuroni è tra la sinapsi di uno e la dendrite dell'altro.
Quando un neurone viene eccitato trasmette un potenziale di azione lungo l'assone e, quando questo raggiunge le sinapsi, viene trasformato in segnale chimico.

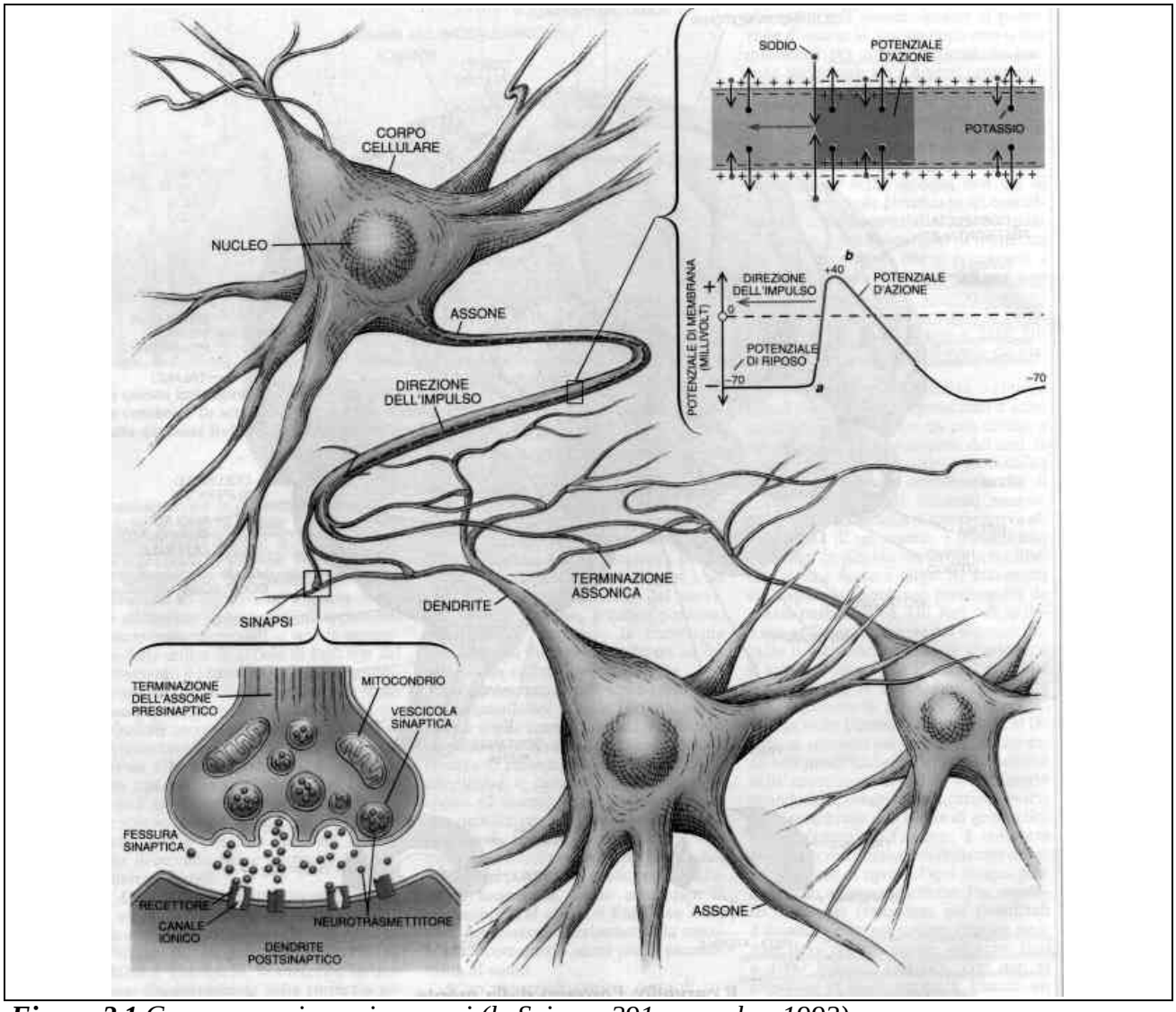

Figura 3.1 *Come comunicano i neuroni.(le Scienze 291 novembre 1992)*

A riposo, la membrana esterna del corpo cellulare mantiene un potenziale di circa -70mV rispetto alla parte interna. In questa situazione, la membrana è più permeabile al potassio che al sodio e si crea un equilibrio alla barriera di potenziale. Quando la cellula riceve lo stimolo nervoso, sotto forma di segnale ionico, accade che la situazione precedente si inverte. Si ha un aumento della permeabilità al sodio, il quale entrando nella cellula, carica la membrana in quel punto

positivamente di circa 100 mV, provocando un impulso elettrico che si propaga lungo l'assone. Quando l'impulso raggiunge la terminazione sinaptica, provoca la liberazione di molecole di neurotrasmettitori, le quali vengono in parte recepite dalla dendrite del neurone successivo. Questa è dotata di canali ionici in grado di catturare i neurotrasmettitori che vengono trasportati fino al corpo cellulare, dove può avvenire la generazione di un potenziale d'azione. Per dare un'idea della velocità massima di un neurone, si può immaginare che esso è in grado emettere al massimo 100 impulsi al secondo[1].

Il meccanismo sopra descritto è relativo ad un neurone ideale[2]. Inoltre, esistono diversi tipi di cellule neurali ed ognuna si distingue per delle particolari caratteristiche, quindi, la descrizione precedente è relativa ad un ipotetico neurone medio.

In ogni caso si possono distinguere almeno due fasi.

1- Il neurone riceve lo stimolo ionico dalle sue dendriti. Se questo è alto può scatenare il potenziale d'azione.

2- La sinapsi riceve il potenziale d'azione ed, in base alla sua struttura, emette una certa quantità di neurotrasmettitori che sono in grado di indurre un potenziale ionico nella dendrite del neurone successivo.

Queste due fasi sono state messe in evidenza poiché sono proprio quelle imitate con le reti neurali artificiali.

3.2 . Perceptron

E' solo nel 1958 che Frank Rosembalt, in un articolo [Rosembalt 1958] presenta quello che ormai è noto con il nome di *perceptron.* Rosembalt riprende il neurone artificiale formalizzato da McCulloch e Pitts ed afferma inoltre che una rete con pesi regolabili può essere utilizzata per classificare un certo gruppo di modelli.

Il neurone formale. a cui sia McCulloch e Pitts che Rosembalt fanno riferimento è quello mostrato in figura .0.2. Sono stati evidenziati, in modo schematico, gli ingressi ad un nodo. Questo, trasferisce la somma degli ingressi ad un operatore funzionale che manipola l'uscita del nodo che viene chiamata *attivazione* . del neurone. Il ruolo dell'operatore funzionale F è quello di funzionare

---

[1]Sulla frequenza di lavoro dei neuroni esistono due teorie a riguardo del metodo di trasimssione adottato da queste cellule. Alcuni pensano che l'altezza dell'impulso non è sempre costante e di conseguenza si avrebbe una sorta di modulazione d'ampiezza. Diversamente, altri sostengono che l'altezza dell'impulso è sempre di circa 100 mV e che, al massimo, è presumibile una modulazione in frequenza.

[2]Per una descrizione più dettagliata, soprattutto da punto di vista delle relazioni analitiche, si vedano le equazioni di Hodgkin e Huxly, publicate in "*Jurnal of Psycology"* nel 1952.

come un interruttore che regola l'eccitazione o l'attivazione del nodo. Il nodo rappresenta il corpo cellulare e, di conseguenza, gli ingressi al nodo sono proprio le dendriti. Infine, i segnali provenienti sulle dendriti vengono pesati con i pesi W che svolgono la funzione di sinapsi.

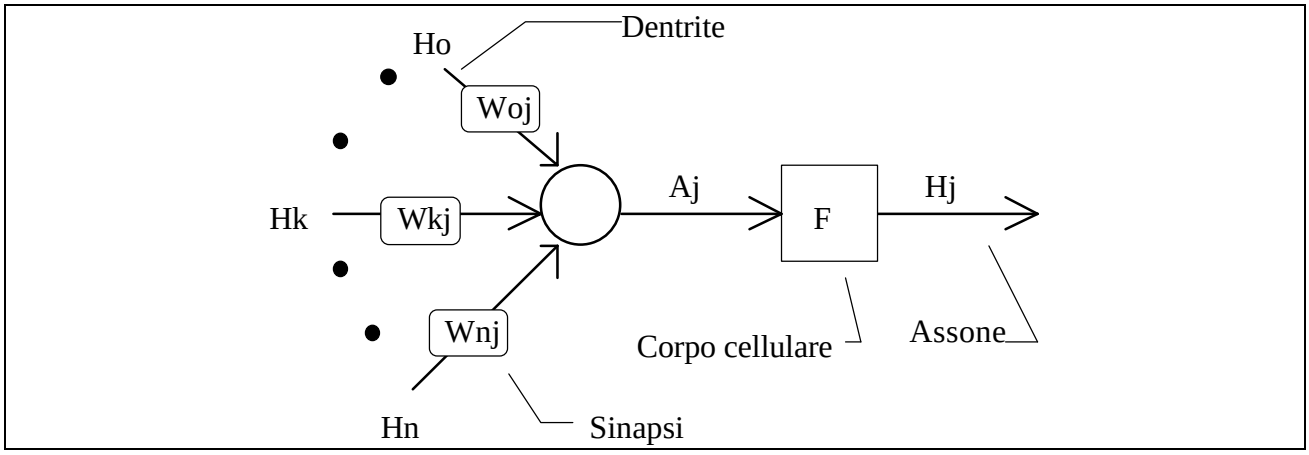

Figura .0.2 *Neurone Formale*

Analiticamente, tale modello è rappresentabile dalla seguente relazione:

$$H_j = F(\Sigma_k w_{kj} * H_k)$$

Le analogie di questo modello con quello descritto al paragrafo precedente sono molto evidenti e rappresentano il motivo per cui questi modelli sono stati identificati come *reti neurali artificiali*. Successivamente sono stati sviluppati molti modelli di reti neurali. Si è variato il tipo di funzione F, la topologia della stessa rete e soprattutto il suo apprendimento. In ogni caso, c'è ancora confusione sull'argomento e non esiste una metodologia precisa che permetta di affrontare scientificamente il problema. Spesso tutto è lasciato all'iniziativa personale del ricercatore. Infine, in quest'era di rapide evoluzioni si cerca di ottenere da questi metodi delle applicazioni funzionanti, ma, in realtà, non sono ancora chiari i metodi per il loro sviluppo.

3.3.Apprendimento di una Rete Neurale.

La fase in cui la rete impara a trattare il problema per cui è stata progettata viene chiamata apprendimento. Questa fase, diversamente da quanto accade nella mente umana, è nettamente distinta dalla fase di esecuzione ed, inoltre, la precede.

Le reti neurali si distinguono in due grandi categorie in funzione dell'apprendimento. Troviamo, infatti, reti ad apprendimento con supervisore e senza supervisore. Nel primo caso, durante la fase di addestramento è necessario fornire alla rete degli esempi di cui si conosce il risultato esatto, nel secondo caso è sufficiente fornire solo gli esempi ed è compito dell'apprendimento trovare il legame tra essi..

La fase di apprendimento si distingue da quella di esecuzione perché, nella prima, la rete cambia qualcosa nella sua struttura con l'obiettivo di minimizzare un errore.

3.4.Reti ed Esempi.

Il compito di una rete è di trovare una serie di numeri (pesi) che, opportunamente combinati (la rete), sono in grado di ripetere il legame tra l'esempio ed il suo risultato. Il suo funzionamento è fortemente condizionato dai seguenti fattori:

- Diversificazione degli esempi. Infatti, se venissero dati molti esempi tutti uguali, la rete troverebbe immediatamente una serie di numeri in grado di calcolare perfettamente il risultato dell'esempio. Ma sbaglierà tutte le volte che viene utilizzata per casi da lei mai visti.
- Numero degli esempi. Anche se vengono forniti una serie di esempi caratteristici, la rete ne deve avere un numero sufficiente per potere rendere generale la sua conclusione. In effetti, questo problema è fortemente legato alla diversificazione, che deve essere sufficientemente fitta.

La rete riesce a risolvere casi da lei mai visti, perché questi rispettano la stessa regola degli esempi visti in fase di apprendimento.

QUINDI, LA FASE DI ADDESTRAMENTO DELLA RETE RICHIEDE UNA SERIE NUMEROSA DI ESEMPI. QUESTI, DEVONO ESSERE SIGNIFICATIVI PER IL PROBLEMA CHE LA RETE DEVE AFFRONTARE. DI CONSEGUENZA, LA SELEZIONE DEGLI ESEMPI DEVE FAR PARTE DELLA FASE DI PROGETTO DELLA RETE.

La selezione degli esempi è tanto più critica quanto meno numerosa è la serie degli esempi. Inoltre, non è possibile scendere al di sotto di un certo numero di esempi a causa della totale deficienza della rete.

Infatti, se per risolvere un problema semplice come la funzione di una retta, si fornisse alla rete un solo punto, questa al massimo potrebbe calcolare una delle infinite rette che passano per il punto indicato, ma la probabilità che la retta sia quella cercata è nulla.

I "problemi reali", anche se non hanno una regola formale, ne hanno una intrinseca, che può essere astratta dall'insieme degli esempi.

VIENE CERCATA LA RELAZIONE CHE LEGA UNA SERIE DI ESEMPI AD UNA SERIE DI RISULTATI. SUCCESSIVAMENTE, SI PRESUPPONE CHE TALE RELAZIONE SIA VERA E VALGA PER TUTTI I MODELLI APPARTENENTI AD UNA DATA CLASSE.

E' da notare come il concetto di "appartenere ad una stessa classe" sia molto astratto e che questa distinzione dipende da noi. La rete sarà tanto più perfetta quanto più gli esempi dati hanno un'effettiva correlazione tra di loro: le sue risposte saranno tanto più affidabili quanto più il modello posto in ingresso ha una correlazione con la famiglia di esempi.

In questa analisi la rete viene vista come una vera e propria funzione. Dato un ingresso, la rete sarà in grado di fornire un'uscita più o meno accettabile. Non è sempre noto quale sia la funzione tantomeno come sia fatta, tuttavia si può pensare alla rete come un'operatore funzionale.

3.5.La Rete come Operatore Funzionale.

La rete è dunque definita come una funzione f di cui $\underline{X}$ rappresenta la variabile in ingresso e $\underline{Y}$ la variabile in uscita per cui:

$$\underline{Y} = f(\underline{X})$$

$\underline{X}$ e $\underline{Y}$ sono vettori e rappresentano l'ingresso e l'uscita. In questa configurazione, dunque $\underline{X}$ e $\underline{Y}$ sono strettamente legate al problema che si intende trattare. Il primo problema che si incontra nella realizzazione di una rete è quello di:

stabilire quali sono le grandezze in ingresso e quelle in uscita.

Si tratta di stabilire l'interfaccia mondo macchina e macchina mondo. Tali grandezze sono fortemente condizionate dalle apparecchiature tecnologiche che si utilizzano, da come viene rappresentata l'informazione ed ancora da quale parte dell'informazione si vuole elaborare[3]. Comunque, dati i vettori $\underline{X}$ e $\underline{Y}$ si determina solo la funzione f di cui fisso l'insieme di definizione:

$$\underline{X} = (x_1,x_2,...,x_n) \in A \subseteq R^n$$

$$\underline{Y} = (y_1,y_2,...,y_m) \in R^m$$

$$f : \underline{X} \in A \subseteq R^n \rightarrow R^m$$

e cioè, la funzione associa un valore di $\underline{X}$ che appartiene all'insieme di definizione A, sotto insieme dello spazio vettoriale ad n componenti di numeri reali, un vettore ad m componenti di numeri reali. Ad ogni vettore $\underline{X} = (x_1,x_2,...,x_n)$, la funzione fa corrispondere un vettore $\underline{Y} = (y_1,y_2,...,y_m,)$ per cui:

$$(y_1,y_2,...,y_m) = [f_1(x_1,x_2,...,x_n) ,f_2(x_1,x_2,...,x_n) ,...,f_m(x_1,x_2,...,x_n)] \text{ quindi}$$

$$y_i = f_i(x_1,x_2,...,x_n) = f_i(\underline{X})$$

da quest'ultima relazione risulta che la funzione cercata è del tipo VETTORIALE, non è detto però che essa sia LINEARE, che sia cioè del tipo:

---

[3]In realtà, questo problema sarebbe secondario dato che si potrebbe delegare alla rete il compito di estrarre l'informazione utile, però questo graverebbe troppo sulla dimensione della rete e di conseguenza sulla fase di addestramento.

$y_i = a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n$ con i coefficienti a_{ij} costanti

L'INSIEME DEGLI ESEMPI B, E' UN SOTTO INSIEME DELL'INSIEME DI DEFINIZIONE A PER LA FUNZIONE f. INOLTRE, DEVE ESSERE NOTO L'INSIEME C OTTENUTO DA B ATTRAVERSO LA TRASFORMAZIONE f.

$\underline{Xd} \in B \subset A$

$\underline{Yd} \in C \subset R^m$: $\underline{Yd} = f(\underline{Xd})$

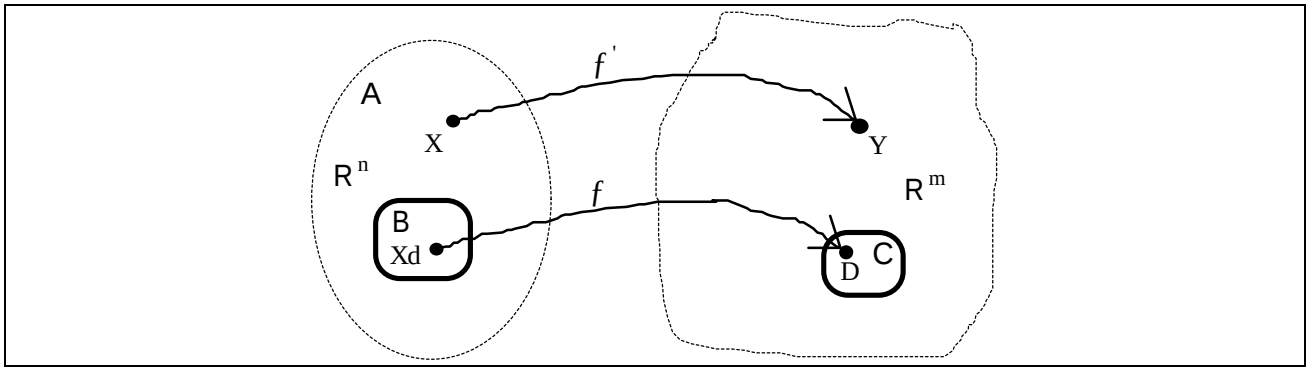

***Figura 3.3. Insiemi** di definizione per la rete*

Nella formulazione rigorosa non è presupposta la conoscenza di f, bensì si vuole che l'insieme degli esempi sia tale da appartenere alla stessa categoria a cui appartiene il problema.

Lo scopo della rete è proprio quello di scoprire la funzione che lega $\underline{Xd}$ *a* $\underline{Yd}$ *per poi generalizzare e rendere valida, per tutto l'insieme* A*, la funzione trovata valida per l'insieme* B.
B *è solo una parte di* A *che invece raccoglie tutti i casi del problema.*

Quindi, in realtà, la rete trova una funzione f' che è la migliore approssimazione determinata in base agli esempi forniti. A questo proposito esistono due diversi problemi di precisione:

- La funzione f' ha un errore[4] ϕ calcolato sugli esempi. Cioè, l'errore che la rete commette calcolando da sola il valore $\underline{Yd}$ da $\underline{Xd}$. Tale errore è dovuto al fatto che difficilmente si riesce a trovare un'unica funzione in grado di riprodurre esattamente tutti gli esempi, quindi si cerca una funzione che meglio riesca ad approssimare la serie dei campioni. Questo sarà chiamato ERRORE FISIOLOGICO (ϕ) della funzione.
- La funzione f' non è esattamente f, bensì è una sua approssimazione fatta su un campione di esempi B che è solo una parte dei valori che si intende trattare. C'è da aspettarsi dunque che per valori sempre più prossimi a quelli contenuti in B la rete tende ad un errore vicino a ϕ, ma

[4]L'errore non è stato ancora detrminato analiticamente. Ad esempio potrebbe essere lo scarto quadratico medio. Ma questo problema vale la pena di affrontarlo in modo approfondito.

che in generale sia soggetta anche ad un errore α dovuto alla differenza tra la funzione reale e quella che l'insieme degli esempi ha lasciato capire. Se l'insieme degli esempi è significativo, (al limite tutti i casi possibili), α tende a zero. Questo verrà indicato come ERRORE DI ASTRAZIONE (α) del problema.

L'errore generale è dunque:

$$e = \phi + \alpha$$

- ϕ dipende esclusivamente dai modelli matematici utilizzati per determinare la funzione *f*'. Potrebbe tendere a zero; comunque, il suo valore è noto e costante.
- α dipende solo dalla dimensione e dalla selezione degli elementi dell'insieme B. Concettualmente tende a zero solo quando $B = A$. α dipende molto da come vengono presi i campioni, oltre che dal numero degli stessi; infatti, un numero infinito di esempi non è sufficiente a garantire $\alpha=0$, poiché questi devono essere anche ben distribuiti in A.

La scelta degli esempi permette di regolare l'errore di astrazione α che è sempre incerto.
L'apprendimento della rete ha lo scopo di portare l'errore fisiologico ϕ al minimo.
Una rete affidabile deve avere un errore generale controllabile e quindi molto vicino a ϕ.

$e \cong \phi \Rightarrow \alpha \cong 0$ errore controllabile

$\phi \rightarrow 0$ errore minimo

3.6.Tipologie di rete neurale artificiale.

A partire dal modello perceptron di Rosembalt, sono stati sviluppati diversi modelli di reti neurali anche molto differenti tra loro. La prima grande classificazione che si può fare è in funzione delle modalità di apprendimento. Un'altra caratteristica molto importante è la funzione a cui la rete meglio si adatta, oltre, ovviamente, alla sua capacità.

E' importante comunque sottolineare che le reti neurali sono pensate per dei circuiti appositi. Infatti, uno dei maggiori pregi delle reti neurali, è l'alto grado di parallelismo insito[5]. Spesso anche con dei circuiti appositamente realizzati, non si riesce a sfruttare del tutto il calcolo distribuito tipico delle reti neurali e quindi, si è sempre costretti ad una serializzazione. Questo problema si fa molto più evidente quando si parla di simulazione su computer, dove nella stragrande maggioranza dei casi non esiste nessun parallelismo e spesso, si satura immediatamente la capacità di calcolo e di memoria. Difatti, anche la memoria rappresenta un problema tipico di una rete neurale. E'

[5]Infatti, il nostro cervello pur lavorando ad una frequenza inferiore ai cento hetz, ha una velocità di risposta molto superiore.

necessario conservare i pesi di ogni connessione ed, in genere, per una rete con N neuroni, il numero dei pesi è circa N^2, e poiché di solito il numero N è molto grande e i pesi sono normalmente numeri reali, ci si può rendere conto delle grosse risorse necessarie.
Attualmente sono allo studio reti neuro-ottiche, dove la capacità di calcolo è svolta da "calcolatori ottici" Tuttavia, le conoscenze nel campo sono ancora troppo poche per poter ottenere dei risultati pratici soddisfacenti (per maggiori informazioni riferirsi a: [Abu-Mostafa 1987]).

3.6.1. Reti Associative.

Le reti associative si distinguono per le loro capacità di funzionare come memorie "indirizzabili per contenuto", cioè come memorie associative. Il padre di questa tipologia di reti è Hopfield, [Hopfield 1982],dal quale prendono il nome. Le caratteristiche principali di questa tipologia di reti è l'ingresso binario, l'alta velocità di apprendimento, l'oscillazione in fase di esecuzione.
Normalmente, in questa rete, i valori binari sono del tipo -1 +1 e, la funzione operata da ogni nodo, è del tipo a scalino centrato sullo 0. La peculiarità di questa rete è che il segnale in ingresso viene continuamente fatto circolare nella rete fino a quando questa non si stabilizza. Lo stato stabile viene identificato dal fatto che dopo un certo numero di oscillazioni, i nodi in uscita mantengono sempre il loro valore, cioè fino a quando la rete non cambia più le sue unità. Tali oscillazioni avvengono in fase di esecuzione, ma non è garantito che siano in grado di trovare un punto stabile. Questo è solo uno dei problemi della rete associativa. Questo tipo di topologia, infatti, presenta grossi problemi di capacità, ossia, di quantità di informazione memorizzabile [McEliece 1987]. Inoltre, essa confonde configurazioni di ingresso.

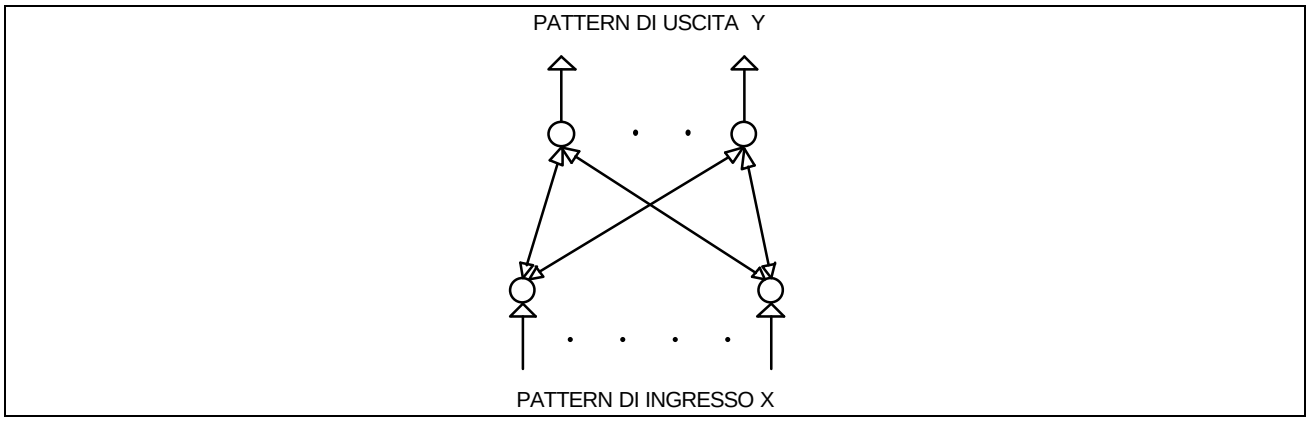

***Figura 3.4**. Schema di una memoria associativa distribuita.*

Quest'ultima questione è risolvibile con una diagonalizzazione dell'ingresso, tuttavia permangono notevoli incertezze sulla capacità e sulla scarsa resistenza al rumore.
Per questa serie di motivi, le reti associative pur avendo un grande interesse scientifico, trovano scarso interesse pratico.

Per chi volesse approfondire le reti associative si trova un'ottima descrizione in [Koso 1987] e in [Kohonen 1977], in cui cercano di fornire una metodologia di studio basata sulla teoria dei sistemi.

Box 1. *L'algoritmo di HOPFIELD:*

1 Assegnare i pesi alle connessioni:

$$w_{ij} = \left\{ \begin{array}{l} \sum_{s}^{M-1} x_i^s y_j^s, i \neq j \\ 0 i = j i \geq 0 j \leq M-1 \end{array} \right\}$$

w_{ij} è il peso della connessione tra il nodo i ed il nodo j. Xi è l'i-esimo elemento del vettore X che ha N componenti. M è il numero dei componenti di Y di cui yj è un elemento.

2.Inizializzare con vettori in ingresso sconosciuti:

$$\mu_i(0) = x_i \quad 0 \leq i \leq N-1$$

il termine $\mu_i(0)$ è l'uscita del nodo i all'istante 0;

3.Fino alla convergenza cioè fino a quando non è $\mu_j(t+1) = \mu_j(t)$

$$m_j(t+1) = f_h\left(\sum_{i=0}^{N-1} w_{ij} m_i(t) \right) 0 \leq j \leq M-1$$

la funzione f_h è una funzione limitata tra -1 e +1 è potrebbe essere, ad esempio, la funzione segno oppure il gradino centrato in 0.

3.6.2. Reti Feedforward.

Questo tipo di rete neurale è una generalizzazione del perceptron di Rosembalt. Letteralmente feedforward significa non ricorrente, cioè la direzione dell'impulso nella rete è unidirezionale. Questo, la distingue dalla rete di Hopfield proprio perché, quest'ultima, possiede dei collegamenti nella direzione tra l'uscita e l'ingresso. Inoltre, questo tipo di rete neurale tratta valori reali, anche se è utile normalizzare tali valori all'interno dell'ipercubo di lato unitario.

L'apprendimento di questa rete avviene con l'algoritmo di retropropagazione dell'errore, meglio noto come back propagation.

Questa tipologia di rete si presta bene alla classificazione di modelli e denota una certa capacità di astrazione: è abile a classificare bene anche modelli che non ha mai visto.

I problemi principali di questa rete sono dovuti all'algoritmo di apprendimento. Infatti, l'algoritmo di back propagation tende a minimizzare l'errore che viene messo in relazione con i pesi delle connessioni della rete. L'errore della rete è dunque funzione dei suoi pesi. In questo modo, il problema di minimizzare l'errore è il problema di minimizzare una funzione di n variabili dove n è il numero totale delle connessioni presente nella rete. Per risolvere tale minimo viene utilizzata una tecnica nota come *discesa del gradiente,* in cui si cerca con piccole variazioni dei pesi di raggiungere uno stato di minimo.

La discesa del gradiente è caratterizzata da due problemi. Innanzitutto, richiede tempi lunghi per dare una risposta. Ma, soprattutto, non garantisce che il minimo trovato sia un minimo assoluto e che la stabilizzazione avvenga in un minimo locale. Inoltre, essa non è in grado di determinare se esiste un minimo, quindi, potrebbe anche non convergere mai.
Nel caso di reti neurali, la non convergenza implica l'impossibilità di trovare soluzione per la rete.
Nonostante questi grossi inconvenienti, la rete feedforward è quella più utilizzata nelle applicazioni pratiche. Inoltre, il teorema di [Hecht-Nielsen 1989] afferma che qualsiasi funzione $\underline{Y}=F(\underline{X})$ può essere accuratamente computata da una rete non ricorrente a soli tre livelli. Questo teorema in realtà ha più valore teorico che pratico, infatti non viene spiegato come dimensionare tali livelli per ottenere l'accuratezza desiderata. Infine, il numero delle unità per ogni livello è, in generale molto grande.
Quello che di fatto accade è che, scegliendo bene gli esempi, è molto probabile che si trovi una situazione stabile. Non è garantito però che sia la soluzione ottimale e che la rete non si sia stabilizzata in un minimo locale[6].
Il vero e proprio problema delle reti feed forward è il tempo di apprendimento, dovuto essenzialmente all'algoritmo di back propagation. In più, per avere dei risultati significativi la rete deve vedere un grande numero di esempi e, questo, oltre ad aumentare il tempo per l'apprendimento, pone problemi pratici di reperimento degli stessi esempi.

3.6.3. Reti Stocastiche.

L'elemento che contraddistingue questa tipologia di rete è che l'algoritmo di apprendimento non è deterministico, bensì i termini utilizzati sono probabilistici. La rete stocastica per eccellenza è la *Macchina di Boltzman* [Hilton-Sejnowsky 1986].
In queste reti la regola per l'addestramento stocastico è del tipo:

$$p_j = \frac{1}{1+e^{-\Delta E_j/T}}$$

che è la probabilità che il nodo *j* assuma lo stato *1*. Il termine *T* rappresenta la "temperatura assoluta" delle rete, mentre DE_j rappresenta l'energia (l'input) totale ricevuto dallo stesso nodo.
La probabilità p_j è un gradino addolcito meglio nota come sigmoide. Questo gradino è tanto più addolcito quanto più è alta la temperatura *T*. Tale situazione implica che, un nodo il cui input è nullo, ha comunque una certa probabilità di essere attivo. Essa è tanto più alta quanto più alta è la temperatura *T*. Questo fenomeno è analogo allo stato di agitazione degli elettroni soprattutto dei metalli, quando aumenta la temperatura. L'idea dei metodi stocastici è quella di regolare i pesi delle

---

[6]La situazione di minimo locale o assoluto deve essere valutata solo in funzione della risposta fornita dalla rete. Se questa è soddifacente, non ha grossa rilevanza il tipo di minimo che la rete ha trovato.

connessioni in modo che nodi vicini siano tra loro in buone relazioni, proprio come accade per gli elettroni. Infatti, lo stato stabile di una rete neurale è praticamente il minimo nella superficie d'energia della funzione errore. L'idea è quindi di riuscire a fare in modo che, come accade in natura, la rete si assesti nel suo stato di energia minima. Per ottenere ciò si opera il meccanismo della "tempra simulata", ovvero dell'algoritmo di *simulated anneling* [Kirkpatrick 1983].Così come accade per i metalli, si alza notevolmente il parametro *T* della temperatura, provocando una grossa agitazione dei nodi della rete, che spesso sono attivi anche quando non dovrebbero esserlo. Successivamente, si opera un "congelamento", cioè la temperatura viene portata a valori bassissimi, di conseguenza rimangono attivi solo i nodi che hanno effettivamente lo stato di eccitazione. Questo meccanismo mette in ordine i nodi della rete, così come il metallo diventa più compatto. Infatti, se due nodi sono in buona relazione tra di loro, vuole dire che il loro stato di energia è minimo.
Una trattazione analitica delle reti stocastiche si può trovare in [Amari 1972] con una particolare attenzione alle connessioni eccitatorie ed inibitorie.

3.6.4. Reti Auto Organizzanti.

Il padre di questa interessante tipologia di reti è Kohonen [Kohonen 1982], dal quale prendono il nome. Il concetto di queste reti è quello di auto organizzarsi in strati o zone che diventano specializzate nello svolgere una determinata area del problema generale. Questa situazione è molto simile a quelle che accade nella nostra corteccia celebrale. Infatti, il nostro sistema di elaborazione dei segnali sonori è organizzato tonotopicamente, nel senso che vi sono zone dedicate a trattare con determinate frequenze piuttosto che con altre. Questo significa che il suono da noi percepito viene scomposto e suddiviso nelle diverse aree di competenza.
Inoltre, questa famiglia di reti neurali si distingue per essere caratterizzata da apprendimenti senza supervisione.

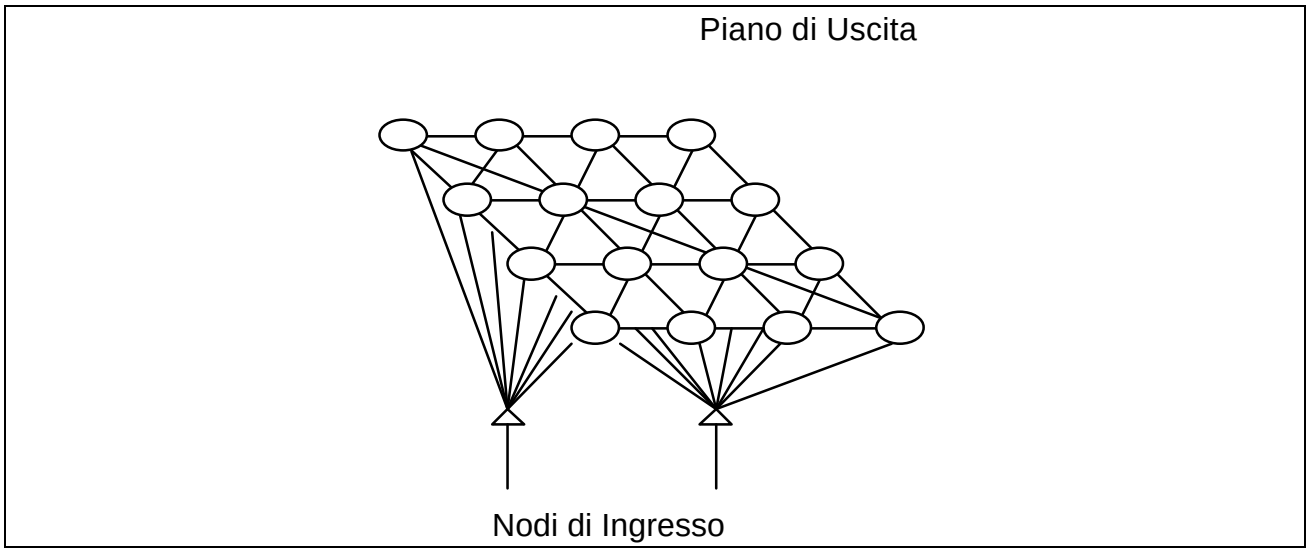

Figura 0.3*. Schema di una memoria auto organizzata. I nodi in uscita sono organizzati in un array bidimensionale, tutti i nodi vicini sono interconnessi. Inoltre, ogni nodo di input ha una connessione pesata con tutti i nodo di out.*

E' interessante notare come la distribuzione dei pesi nella griglia di uscita tende a ricopiare la distribuzione degli esempi. Si è infatti osservato che nella rete rappresentata in figura 0.3, fornendo ai due ingressi valori distribuiti, rispettivamente sui due assi cartesiani, dopo un considerevole numero di esempi (circa 5000), la matrice dei pesi relativi alle connessioni dei due nodi con il piano di uscita, assume una rappresentazione a griglia regolare, che copre l'intera area da cui sono stati tratti gli ingressi, presi in modo casuale. Questo implica che la rete si costruisce internamente la mappa delle caratteristiche, cercandole nel vettore di input, da cui scaturisce l'apprendimento senza supervisione. I pesi, in questo caso, rappresentano una quantizzazione dello spazio rappresentato dall'input. Ogni quanto raggruppa gli esempi simili tra loro, in funzione dell'insieme analizzato. Questo modello di rete si presta bene ad effettuare una classificazione di modelli, in cui non è ben chiaro come elaborare le informazioni ad essi associati. Tuttavia, richiede un numero molto elevato di connessioni e molti esempi, che tra l'altro devono coprire tutto il problema.

3.6.5. Reti Genetiche.

Sono modelli derivati teoria dell'evoluzione di Darwin. Elemento chiave di questo approccio è una "mutazione genetica", intesa come piccolo cambiamento di alcuni pesi e come aggiunta o rimozione di un nodo [Nolfi 1989]. Attualmente, questi metodi non hanno avuto grossi successi, soprattutto a causa delle eccessive risorse di calcolo richieste.

4. Feedforward Networks

4.1.Topologia di una Feedforward.

Una rete Feedforward è composta da tre elementi principali: i due livelli di ingresso ed uscita ed un numero variabile di livelli nascosti o interni.
Ogni livello è composto da un certo numero di unità elementari del tipo perceptron descritto nel paragrafo 3.2. In questa tipologia di rete, le unità di due livelli consecutivi sono completamente connessi, mentre non esiste nessuna connessione tra le unità dello stesso livello.

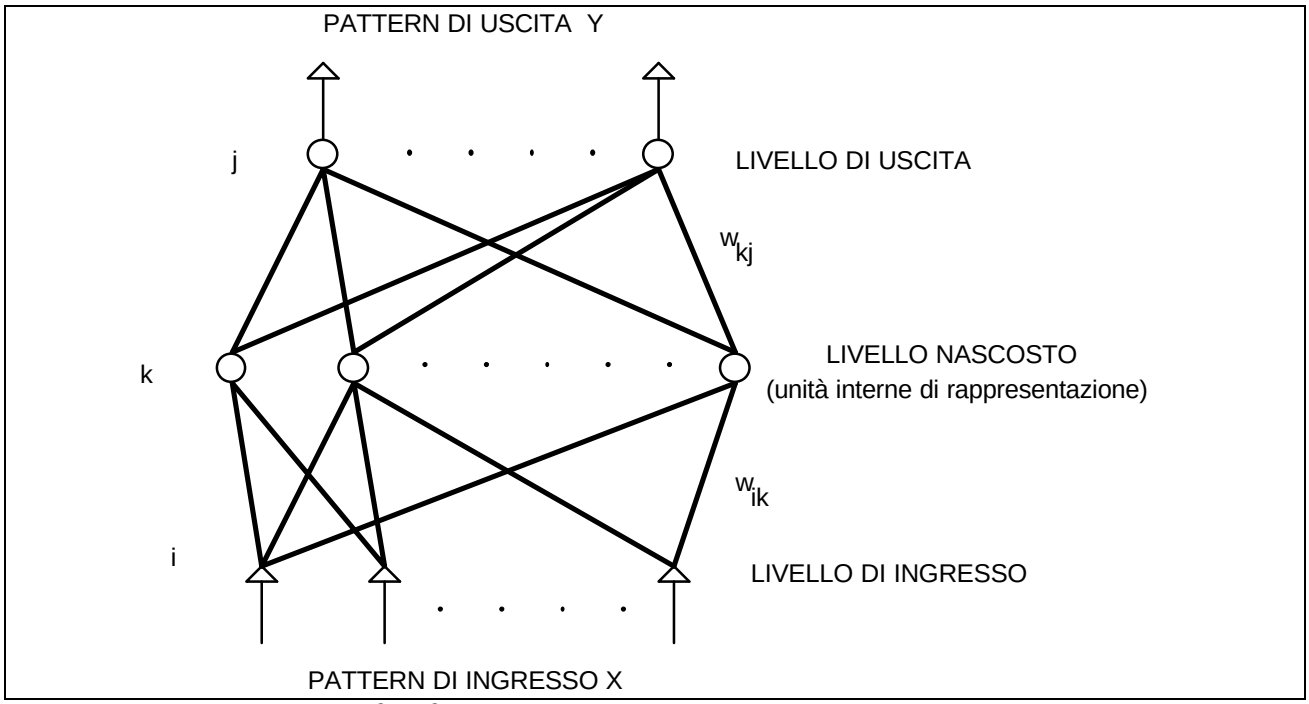

Figura 0.1 *Esempio di rete feedforward ad un livello nascosto.*

Il compito preciso delle unita della rete e di ogni livello della stessa non è precisamente noto, tuttavia in seguito si cercherà di dare un'interpretazione di tali elementi. In generale, il livello in ingresso serve per "catturare" la configurazione da classificare. Questo pattern è organizzato in una forma che può essere non utilizzabile direttamente dalla rete. La rete deve dunque organizzarsi tale informazione. Questa operazione viene eseguita dai livelli interni che vengono chiamati nascosti.

Inoltre, queste unità nascoste hanno il compito di trasferire il segnale preso dal livello di ingresso, al livello di uscita.
I livelli di ingresso e di uscita configurano il modello da calcolare, in un uno spazio vicino al nostro modo di pensare al problema. I livelli interni devono riorganizzare l'informazione ed elaborarla. Viene istintivo pensare di fornire in ingresso una rappresentazione del modello già organizzata per l'elaborazione, quindi eliminare i livelli intermedi. Di fatto, questa è una operazione possibile e quando si riesce a trovare una buona rappresentazione del modello in uno spazio adatto all'elaborazione della rete, questa risulterebbe sicuramente più semplice. Concettualmente non cambia nulla, dato che lo studio fatto a monte nel tentativo di cambiare la rappresentazione del modello corrisponde ad aver fatto precedere un "livello umano" alla rete analitica.
Questa osservazione suggerisce due conclusioni, una pratica ed una teorica:
1. Quando è possibile studiare una rappresentazione del modello in modo formale, è bene farlo per semplificare il compito della rete[7].
2. Se operando una trasformazione a monte delle rete possiamo semplificarla, allora è probabile che complicando la rete semplifichiamo il nostro compito. Questo è vero solo in parte, dato che le reti sono già normalmente complesse. Ciò che è più realistico è il collegamento in serie di reti, pensate per svolgere operazioni elementari, facenti parte di un'unica operazione complessa. In questa ottica una rete può essere vista come un componente di un circuito elettronico e, la funzione operata dal circuito, rappresenta il problema affrontato. Personalmente penso che questo modo di guardare alle reti potrebbe renderle più efficienti e potrebbe anche semplificare l'apprendimento.

4.2.Modello Analitico

Per ridurre al minimo l'errore di astrazione, come spiegato nel paragrafo 3.5, è necessario agire sull'insieme degli esempi. L'errore fisiologico dipende invece dai modelli matematici utilizzati per la determinazione della funzione f', approssimazione della funzione esatta f.
La rete feedforward utilizza come nodo elementare il perceptron descritto nel paragrafo 3.2. Questo componente elementare è regolato dalle seguenti relazioni:

$$A_{j,(l+1)} = \Sigma_k \, w_{kj} * H_{k,l} \qquad (1)$$

$$H_{j,(l+1)} = f(A_{j,(l+1)}) \qquad (2)$$

dove w_{kj} sono dei pesi associati al collegamento dal nodo $H_{k,l}$ al nodo $H_{k,l+1}$.

---

[7]Questo non è sempre possibile dato che le reti neurali vengono solitamente utilizzate quando il problema è poco formalizzabile e molto variabile.

La funzione f determina il tipo di funzione operata dalla rete. In particolare, se la funzione f è non lineare lo sarà anche la funzione generale della rete. Ogni nodo della rete funziona da "interruttore" comandato dai segnali in ingresso $\underline{H_{,l}}$ che vengono opportunamente pesati. La funzione f deve "lasciare aperto il canale" se $A_{j,(l+1)}$ è al di sopra di una certa soglia e chiuderlo se non lo è. Potrebbe essere dunque un gradino centrato sulla soglia, tuttavia per poter operare bene con grandezze reali si utilizza una funzione sigmoide (anche per facilità nei calcoli) del tipo seguente:

$$H_{j,l} = \frac{1}{1+e^{-A_{j,l}}} \qquad (3)$$

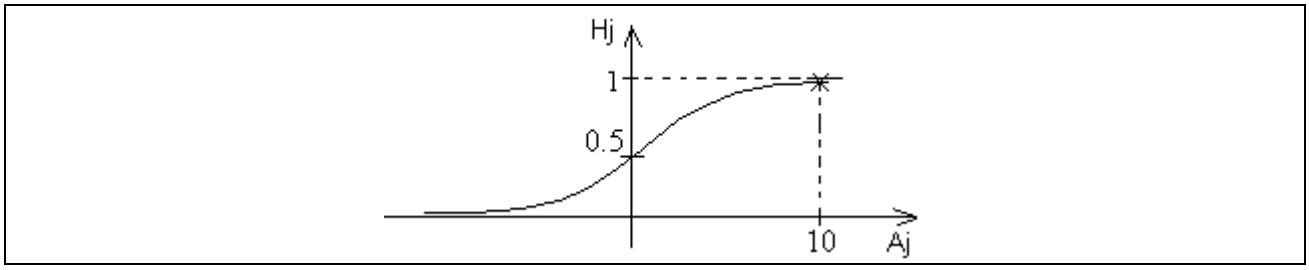

Figura 0. 2*- Funzione sigmoide.*

Da questa funzione si nota che l'uscita di ogni nodo è limitata tra 0 e 1 e che solo per ingressi compresi tra -10 e 10 l'uscita è diversa dai valori limite. L'interruttore è completamente aperto se al suo ingresso c'è un valore superiore a 10, diversamente è completamente chiuso quando all'ingresso vi giunge un segnale minore di -10. Al contrario, l'interruttore è più o meno aperto e, questo, dà alla rete un comportamento graduale.

Si potrebbe dire che il gradino centrato sullo zero sia la soluzione esatta, mentre la sigmoide ha un range di imprecisione che fornisce alla rete una certa resistenza al rumore.

In generale ogni nodo deve poter avere una soglia centrata su un valore che dipende da diversi fattori. Per rendere generica la relazione precedente:

$$A_{j,(l+1)} = \sum_k w_{kj} * H_{k,l} - B_{j,(l+1)} \qquad (4)$$

dove $B_{j,(l+1)}$ è il valore su cui è centrata la sigmoide del nodo j al livello $l+1$.

Questo permette di lasciare invariata l'espressione della sigmoide.

4.3.Determinazione delle grandezze.

Nella presentazione del singolo nodo della rete sono state messe in evidenza diverse grandezze qui di seguito esplicitate.

4.3.1.L'indice *l*.

Ha a che fare con la topologia della rete. Infatti, la rete è realizzata con una serie di nodi collegati in cascata ed l è l'indice del livello in tale cascata. Il numero di livelli può essere indeterminato, in

generale *L* indica il numero di livelli. Casi particolari sono *l=0* in cui si ha il livello in ingresso ed *l=L* che rappresenta il livello di uscita.

4.3.2.La grandezza $H_{k,l}$

E' il valore in uscita dal nodo i che appartiene al livello l. Si distinguono 3 casi in funzione di l:

l = 0, in questo caso il valore $H_{i,0}$ è proprio X_i, infatti il livello *l=0* è il livello di ingresso ed i nodi all'ingresso ricevono lo direttamente stimolo sensoriale dal *vettore* $\underline{X}$.

0<l<L. In questi casi, $H_{k,l}$ è la grandezza prodotta dai nodi intermedi. Si determina applicando la funzione del perceptron a partire da $\underline{H}_{,l-1}$.

l = L. Questo caso si distingue dal precedente solo perché $H_{i,L}$ rappresenta il risultato della rete, cioè *Yi*. Quindi, $\underline{H}_L$ rappresenta il vettore $\underline{Y}$.

4.3.3. Il peso w_{ik}

E' il peso per la connessione tra il nodo i ed il nodo j appartenete al livello di rete immediatamente successivo. Viene determinato durante l'apprendimento ed è l'unica grandezza che si modifica durante tale fase. La conoscenza della rete sta proprio nei pesi dei collegamenti.

4.3.4. La soglia $B_{j,l}$

Rappresenta la soglia per la *j*-esima unità al generico livello l. Cioè, indica dove è centrata la sigmoide data dall'equazione (3). La soglia è quel valore che, in assenza di stimoli, il nodo tende ad assumere. Ogni nodo ha una propria soglia in funzione dell'applicazione, quindi anche questa deve essere appresa dalla rete.

Se si considera la soglia come un peso di una connessione del nodo con una unità fittizia di valore sempre unitario, si ottiene che l'apprendimento della soglia si riconduce all'apprendimento di un ulteriore peso, pertanto il livello superiore sarà composto da N unità come in precedenza descritte ed una a valore unitario per un totale di *N+1* unità. Di conseguenza, ogni nodo mantiene la sua sigmoide centrata sullo zero e si ha che:

$$A_{j,(l+1)} = \sum_k w_{kj} * H_{k,l} - B_{j,(l+1)} \xrightarrow{H_{N+1}=1} = \sum_k^{N+1} w_{k,j} H_{k,l}$$

4.4.L'Algoritmo di Apprendimento: Back Propagation.

La fase di apprendimento in una rete back propagation avviene propagando all'indietro l'errore che la rete commette, valutando un esempio d della funzione che si vuole programmare. Tale errore viene determinato confrontando il vettore $\underline{Yd}$ che la rete fornisce con il vettore $\underline{Dd}$ che è il risultato

esatto che la rete dovrebbe fornire per l'esempio d. Pertanto, l'errore globale commesso dalla rete viene definito come segue:

$$E = \frac{1}{2}\sum_{j}(D_j - Y_j)^2 \qquad (5)$$

Obbiettivo dell'apprendimento è quello di ridurre al minimo l'errore E che dipende esclusivamente dal vettore $\underline{Y}$ che, a sua volta, dato il vettore $\underline{X}$ all'ingresso, dipende solo dai pesi w_{kj} che attraversano tutta la rete. L'errore è dunque funzione dei pesi ed il problema dell'apprendimento si riconduce a quello di minimizzare una funzione ad n variabili dove n è il numero dei pesi. E' da notare che n solitamente è un numero molto grande, dato che corrisponde al numero delle connessioni che normalmente sono totali tra due livelli consecutivi.

Per minimizzare la funzione errore si utilizza la tecnica di discesa del gradiente, attraverso la minimizzazione delle derivate parziali. Quindi, preso un generico livello della rete, si ha che il peso di un link k-j viene modificato secondo la regola:

$$\Delta w_{kj} = -\varepsilon \frac{\partial E}{\partial w_{kj}} \qquad (6)$$

dove ε è una costante detta tasso di apprendimento con valore compreso tra 0.1 e 0.9.

Per la determinazione Δw_{kj} si procede come segue:

$$\Delta w_{kj} = -\varepsilon \frac{\partial E}{\partial A_{j,l}} \frac{\partial A_{j,l}}{\partial w_{kj}} \qquad (7)$$

$$\Delta w_{kj} = \varepsilon \delta_{j,l} \frac{\partial A_{j,l+1}}{\partial w_{kj}} \qquad (8)$$

$$\delta_{j,l} = -\frac{\partial E}{\partial A_{j,l}} = -\frac{\partial E}{\partial H_{j,l}} \frac{\partial H_{j,l}}{\partial A_{j,l}} \qquad (9)$$

per la relazione (8) la derivata parziale si risolve ricordando la formula di attivazione (1);

$$A_{j,(l+1)} = \sum_{k}^{N+1} w_{kj} * H_{k,l}$$
$$\frac{\partial A_{j,(l+1)}}{\partial w_{kj}} = H_{k,l} \qquad (10)$$

L'equazione (9) viene risolta nelle due derivate parziali ricordando sempre le equazioni che legano le variabili interessate e, precisamente, la seconda derivata è la derivata inversa della funzione sigmoide (3) che vale:

$$H_{j,l} = \frac{1}{1+e^{-A_{j,l}}} \Rightarrow A_{j,l} = \ln\left(\frac{H_{j,l}}{1-H_{j,l}}\right)$$
$$\frac{\partial H_{j,l}}{\partial A_{j,l}} = H_{j,l}(1-H_{j,l}) \qquad (11)$$

La prima parte dell'equazione (9) viene calcolata a partire dal livello di uscita, quindi i pesi considerati sono quelli tra il livello *L-1* verso *L*[8].Ricordando l'equazione (5) relativa all'errore della rete si ha che:

$$-\frac{\partial E}{\partial Y_j} = (D_j - Y_j) \qquad (12)$$

pertanto componendo i termini della equazione (9) si ha:

$$\delta_{j,L} = (D_j - Y_j) * Y_j(1 - Y_j) \qquad (13)$$

e da cui si ottiene l'espressione completa per la (8)

$$\Delta w_{kj} = \varepsilon \delta_{j,L} H_{k,(L-1)}$$

Per un livello generico, il ragionamento è identico partendo dalla scomposizione della prima derivata dell'equazione (9)

$$\frac{\partial E}{\partial H_{k,l}} = -\sum_j \frac{\partial E}{\partial A_{j,(l+1)}} \frac{\partial A_{j,(l+1)}}{\partial H_{k,l}}$$

ma, ricordando la (9) si osserva che il primo termine di tale equazione vale

$$\frac{\partial E}{\partial A_{kj,(l+1)}} = -\delta_{j,(l+1)}$$

ed il secondo si ottiene ricordando la formula di attivazione (1)

$$\frac{\partial A_{j,(l+1)}}{\partial H_{k,l}} = w_{kj}$$

pertanto si ha:

$$\frac{\P E}{\P H_{k,l}} = -\sum_j d_{j,(l+1)} w_{kj}$$

$$d_{k,l} = -\left(\sum_j d_{j,(l+1)} w_{kj} \right) * H_{k,l}(1 - H_{k,l})$$

A questo punto è possibile calcolare per ogni unità di ogni livello la variazione da fornire al peso della connessione con i nodi del livello successivo. E' da notare per risolvere il livello l è necessario risolvere il livello l+1, quindi il primo passo è partire da livello *l=L*.

[8]E' da ricordare che $H_{j,l}$ quando *l=L* viene chiamato Y_j, così come quando *l=0* vale X_j.

Box 2. *L'algoritmo Back-Propagation*

1. Inizializzare tutti i pesi e le soglie con dei numeri casuali compresi tra -0.5 e 0.5
2. Prendere N esempi distinti e rappresentativi del problema;
3. Per tutti gli *N* esempi
   3.1. Presentare il prossimo esempio $\underline{X}$ specificando $\underline{D}$ che è il vettore $\underline{Y}$ desiderato.
   3.2. Utilizzare la funzione 1 con la funzione 3 su tutti i nodi della rete per calcolare $\underline{Y}$
   3.3. Aggiustare i pesi secondo la relazione:

$$\Delta w_{kj} = \varepsilon \delta_{j,l} H_{k,(l-1)}$$

   dove se l = L cioè livello di uscita:

$$\delta_{j,L} = (D_j - Y_j) * Y_j (1 - Y_j)$$

   altrimenti per tutti i livelli precedenti:

$$d_{k,l=} - \left(\sum_j d_{j,(l+1)} w_{kj} \right) * H_{k,l} (1 - H_{k,l})$$

   3.4. Tornare al passo 3.1 fino alla fine degli esempi
4. Tornare al passo 3 fino a quando l'errore dato dalla relazione (5) è basso quanto basta.

4.4.1. Epoche di Apprendimento.

Un'epoca è il passo fondamentale dell'apprendimento, in cui vengono presentati tutti gli esempi alla rete. L'algoritmo di back propagation regola i pesi per piccoli passi ciclando più volte sugli esempi. Ogni volta che l'algoritmo ha elaborato tutti gli esempi si conclude un'epoca. Nel box 2 un'epoca comincia al punto 3 e termina al punto 3.4. L'algoritmo di apprendimento cerca di determinare i pesi che vanno bene per tutto l'insieme degli esempi. Ogni variazione ai pesi, operata al punto 3.3 del box 2, tiene conto solo dell'errore commesso per l'esempio corrente. Per trovare una soluzione valida per tutto l'insieme degli esempi, l'algoritmo li deve elaborare tutti durante ogni epoca. Questo vuol dire che, per aggiungere "conoscenza" ad una rete già addestrata, è necessario aggiungere al vecchio insieme di esempi il nuovo insieme e rioperare un addestramento generale con l'insieme così ottenuto. Se non si operasse in questo modo, la regolazione dei pesi non potrebbe tener conto degli esempi dell'insieme precedente e la rete "cancellerebbe la vecchia conoscenza per fare posto alla nuova".

4.4.2. Errore di un'Epoca.

Ogni esempio analizzato dalla rete ha il valore di un errore associato, dato dalla relazione (5). Poiché in un'epoca vengono analizzati tutti gli esempi, per essere precisi l'errore di un'epoca non è un numero scalare, bensì un vettore di dimensione pari al numero degli esempi contenuti nell'insieme di addestramento. Tuttavia, si considera l'errore di un'epoca come il massimo errore verificatosi durante quell'epoca. Di conseguenza, tale valore è l'errore massimo che la rete commette su tutto l'insieme degli esempi. Per cui:

$$\underline{R} = \{(\underline{X}_0, \underline{D}_0), .. (\underline{X}_N, \underline{D}_N)\} \underline{Y}_k = F(\underline{X}_k)$$
$$E_k = \frac{1}{2}\sum_{j=0}^{m}(D_{k,j} - Y_{k,j})^2 k = 0,1...N \qquad (14)$$
$$Ee = \max(E_K)$$

Dove $\underline{R}$ è l'insieme delle N coppie (esempio, risultato esatto) .*F* è la funzione operata dalla rete, m il numero delle unità del livello di uscita, ed *Ee* è l'errore dell'epoca.

- Durante la fase di apprendimento l'errore istantaneo della rete è l'errore dell'epoca corrente.
- Ad apprendimento completato l'errore della rete è l'errore dell'ultima epoca operata dall'algoritmo di back propagation.
- L'insieme degli errori di ogni epoca danno la curva di apprendimento in funzione dell'epoca.

4.5.La configurazione dell'Input $\underline{X}$.

L'ingresso per la rete è dunque un vettore di numeri reali che si è indicato con $\underline{X}$. Inoltre, il vettore X deve essere normalizzato in un intervallo. Questa necessità è dovuta dalla reazione della sigmoide (3), utilizzata nella rete. Infatti, come si può notare nella figura 0.2, la curva della sigmoide è molto presto asintotica. Questo implica che la funzione assume valori rappresentativi solo in un piccolo range di valori, pertanto per mantenere questa significatività, è necessario normalizzare l'ingresso. Una buona regola è quella di normalizzare il vettore $\underline{X}$ all'interno di un ipercubo di lato unitario.

4.6.La configurazione dell'Output $\underline{Y}$.

I nodi al livello in uscita contengono il risultato della rete. Il vettore *Y* è un vettore di numeri reali ed è normalizzato nell'intervallo in cui è limitata la funzione sigmoide (3), quindi tra (0,1).
Inoltre, *Y* non può contenere la risposta in forma "umana", ma è necessario pensare ad una sorta di codifica. Il tipo di codifica dipende dal tipo di problema da risolvere. Per problemi in cui si desidera un valore numerico, una soluzione potrebbe essere la normalizzazione di tale valore nell'intervallo (0,1). Una possibile alternativa, potrebbe essere quella di fornire in uscita la codifica binaria del risultato numerico. In questo caso, bisogna disporre, al livello di uscita, un numero di nodi pari al numero di bit necessari a rappresentare il massimo risultato possibile.
In ogni caso, la scelta della configurazione in uscita è fortemente legata alla scelta della regola di decisione. Questo è dovuto al fatto che la rete non fornisce risposte esatte, infatti, nel caso della rappresentazione binaria, mai nessun nodo avrà valore 1 e tantomeno alcun nodo potrà valere 0[9].

[9]La funzione sigmoide che determina i valori del vettore Y è asintotica tra (0,1) esclusi

Quindi, ad esempio, è necessario decidere il minimo valore da interpretare come 1, il massimo da interpretare come 0, la soglia di incertezza e così via.
Un metodo adottato nei problemi di classificazione è quello di configurare l'uscita Y, come un vettore avente m componenti dove m è il numero delle classi a cui un dato modello in ingresso può appartenere. In questa rappresentazione, tutti gli elementi di Y valgono 0 tranne quello relativo alla classe esatta che vale 1. Anche con questa rappresentazione è necessario stabilire una regola di decisione che in prima approssimazione può essere quella indicata con la relazione () già discussa nel paragrafo relativo al pattern recognition.

4.7.Livelli nascosti e unità per livello.

Uno dei maggiori problemi relativi alla progettazione di una rete neurale è probabilmente il dimensionamento del numero di livelli e, soprattutto, di quante unità per ogni livello. Infatti, mentre i livelli di ingresso ed uscita, vengono determinati dalla rappresentazione stessa dell'esempio e del risultato, la struttura dei livelli interni della rete è in un certo senso arbitraria. Tuttavia, la capacità di astrazione della rete, la sua robustezza e, non ultimo, il suo tempo di apprendimento, dipendono fortemente dal numero di livelli, nascosti e dal numero delle unità per ognuno di questi.
Il teorema di [Hecht-Nielsen 1989][10] afferma che qualsiasi funzione $\underline{Y}=F(\underline{X})$ può essere accuratamente computata da una rete non ricorrente a soli tre livelli e con un adeguato numero di unità per ogni livello.
In questo teorema si afferma che ogni funzione continua di n variabili, definita nell'intervallo [0,1], può essere implementata esattamente da una rete neurale a 3 livelli, avente: nel livello in ingresso n unità, nel livello intermedio 2n+1 elementi e m unità al livello di uscita, dove m è la dimensione dell'insieme immagine.

$$Y = \Phi(X) \quad \Phi: I^n \subset \Re^n \rightarrow \Re^m$$

$$z_k = \sum_{i=1}^{n} \lambda^k \psi(x_i + \varepsilon k) + k$$

$$y_j = \sum_{k=1}^{2n+1} g_j(z_k)$$

z_k è il valore del k-esimo nodo al livello interno. l è un numero reale costante mentre y è una funzione reale monotona crescente. l y sono indipendenti da F . e è un numero razionale. Infine, g_j è una funzione reale e continua e dipende da F e da e. Questo teorema non dice molto su come

[10]In realtà il terorema è di Kolgmorov e risale al 1957 [Sprecher 1965]e Hecht-Nielsen lo hanno diffuso nella loro publicazione.

devono essere le funzioni g e y , inoltre non è specificato se la funzione sigmoide normalmente usata va bene. Il teorema dunque finisce per avere più valore teorico che pratico.
Da un altro punto di vista, si può ragionare in termini di regioni di decisione, viste nel paragrafo 2.3 In quella sezione, si è parlato di funzione di separazione lineare) che traccia un iperpiano del dominio di definizione della funzione vettoriale. Successivamente, nel paragrafo si è introdotta la formula di attivazione (4) relativa ad un nodo di rete. E' facile notare che le due relazioni sono praticamente identiche, infatti nella (4) è stato solo esplicitato il prodotto vettoriale tra il vettore colonna dei pesi ed il vettore riga degli ingressi.
Questa possibile interpretazione suggerisce che un livello nascosto all'interno di una rete è in grado di tracciare degli iperpiani di confine tra classi differenti, quindi tra differenti risultati della rete. Seguendo questo ragionamento, con un solo livello nascosto, è possibile solo una separazione lineare. L'inserimento di un successivo livello di rete potrebbe dare la possibilità di tracciare delle separazioni lineari per ogni nodo al secondo livello nascosto. In questo modo, variando il numero delle unità nei due livelli nascosti, dovrebbe essere possibile "sagomare" le superfici di decisione come desiderato.
Anche questo ragionamento è puramente teorico, dato che non c'è una regola per determinare quale è il numero di livelli ottimale e quale il numero delle unità. Tuttavia, è indicativo il compito di suddivisione dell'iperspazio della funzione da parte dei livelli interni.