



Fig. 1

卷五 五

Grazie PAINT

*Se non puoi passare attraverso una montagna, giraci intorno;
se non puoi girarci intorno, passaci sopra;
se non puoi passarci sopra, siediti un attimo
e chiediti se raggiungere l'altro lato sia davvero così importante.*

Se lo è comincia a scavare una galleria.

Indice

Introduzione	6
1 Presentazione delle reti neurali	7
1.1 Cenni storici ed origini	7
1.2 Analogie con il sistema nervoso ed applicazioni delle ANN . .	10
1.3 Il neurone biologico	12
1.4 Modello del neurone artificiale	14
1.5 Attivazione di un neurone e funzioni di attivazione	18
1.5.1 Funzione a soglia	18
1.5.2 Funzione lineare	20
1.5.3 Funzione lineare a tratti	21
1.5.4 Funzione sigmoide	21
1.6 Architettura di una rete neurale	22
2 L'apprendimento delle reti neurali	26
2.1 Addestramento e generalizzazione	26
2.2 Paradigmi di apprendimento	28
2.3 Processi di apprendimento	29
2.3.1 Delta rule	29
2.3.2 Apprendimento con correzione di errore o metodo di- scesa del gradiente	30
2.3.3 Apprendimento di Hebbian	34
2.3.4 Apprendimento competitivo	34

2.4	Attività di apprendimento	36
2.4.1	Associazione di pattern e memoria associativa	36
2.4.2	Riconoscimento di pattern	37
2.4.3	Approssimazione di funzioni	38
2.5	Un esempio di apprendimento supervisionato: il percettrone elementare	39
3	Ricorrenti: Reti di Hopfield	42
	Riferimenti bibliografici	43

Introduzione

prova

Capitolo 1

Presentazione delle reti neurali

1.1 Cenni storici ed origini

Le reti neurali artificiali, dall'inglese *Artificial Neural Network (ANN)*, e la conseguente computazione neurale, sono state costruite ispirandosi ai sistemi neurali biologici, con l'obiettivo di modellarne il comportamento e la struttura e di simularne le funzioni basilari e fondamentali.

Una definizione semplice e formale di tali strutture è fornita dall'inventore del primo neurocomputer, il Dr. Robert Hecht-Nielsen che le definì come:

«...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs»

(da Neural Network Primer: Parte uno, Maureen Caudill, 1989)

Tradotto in italiano

«... un sistema di calcolo costituito da una serie di semplici, altamente interconnessi elementi di elaborazione, che elaborano le informazioni attraverso il loro stato dinamico rispondendo agli input esterni.»

Le basi per lo studio di tali reti furono poste dallo psichiatra Warren McCulloch e del matematico Walter Pitts, i quali riuscirono a riprodurre una rete neurale utilizzando semplici circuiti elettrici collegati tra loro. Questa collaborazione portò alla luce l'analogia che sussiste tra le reti neurali e la macchina di Turing ed in tal modo si capì che qualsiasi operazione eseguita da una rete neurale poteva essere eseguita anche da un computer; infatti le reti che furono prodotte risultavano essere automi a stati finiti in grado di realizzare la logica delle proposizioni e di formulare ipotesi sulla natura dei meccanismi cerebrali, il tutto equivalentemente ad un programma per computer. Il frutto di tale lavoro fu reso noto nel 1943 con la pubblicazione del libro "*A logical calculus of the ideas immanent in nervous activity*", nel quale fu schematizzato un combinatore lineare a soglia con dati binari multipli in entrata e un singolo dato binario in uscita.

Ciò che quindi McCulloch e Pitts riuscirono a fare è presentare un modello di neurone formale dimostrando che reti formate da tali neuroni riuscivano a computare funzioni della logica del primo ordine.

Un punto di svolta per lo studio delle reti neurali si ebbe successivamente alla pubblicazione del lavoro dello psicologo Donald Hebb, "*The Organization of Behavior*" nel 1949. Hebb trovò una correlazione tra la psicologia del comportamento umano e la fisiologia del sistema nervoso, donando un grande contributo alla teoria sull'apprendimento associativo, teoria che risultò essere alla base dei metodi di apprendimento delle reti neurali, e che si basava sulla nota legge di Hebb: *«Se un neurone A è abbastanza vicino ad un neurone B da contribuire ripetutamente e in maniera duratura alla sua eccitazione, allora ha luogo in entrambi i neuroni un processo di crescita o di cambiamento metabolico tale per cui l'efficacia di A nell'eccitare B viene accresciuta»*.

Il decennio che va dagli anni cinquanta agli anni sessanta fu totalmente influenzato dalla legge di Hebb a tal punto che numerosi gruppi di ricerca condussero esperimenti e test sulle funzionalità del cervello, fino a porre le basi per la nascita dell'intelligenza artificiale (AI).

Nel 1958 Frank Rosenblatt introdusse il primo schema di rete neurale che designò con il termine "***perceptron***", in italiano percettrone, allo scopo di

fornire un'interpretazione dell'organizzazione generale dei sistemi biologici attraverso un modello mirato all'analisi, in forma matematica, di funzioni quali ad esempio l'immagazzinamento delle informazioni. Il perceptrone fu il primo modello di apprendimento supervisionato e presupponeva uno strato di ingresso ed uno di uscita, discriminando gli ingressi in due insiemi linearmente separabili e basandosi su una regola di apprendimento che si appellava alla minimizzazione dell'errore.

Il perceptrone ancora oggi viene utilizzato in varie applicazioni e risultò essere un modello più efficace rispetto a quello binario di McCulloch e Pitts, poichè i suoi pesi sinaptici sono variabili e quindi in grado di apprendere.

Nonostante, però, l'iniziale successo di tale modello e l'interesse mostrato dalla comunità scientifica, tale rete neurale non risultò abbastanza potente; le reti a due strati basate sui perceptron avevano limiti operativi, infatti non riuscivano a risolvere tutte le classi di problemi, in particolare quelli non caratterizzati dalla separabilità lineare delle soluzioni come ad esempio l'operatore *XOR*, ovvero la funzione *OR* esclusivo, che discrimina gli ingressi in modo non linearmente separabile.

Solo negli anni ottanta, con il matematico Paul Werbos, si superarono i limiti del perceptrone di Rosenblatt. Quest'ultimo introdusse uno o più livelli intermedi all'interno delle reti neurali creando una classe chiamata "***Multi-Layers Perceptron***", ovvero perceptrone multistrato, il cui metodo di addestramento principale era l'*error backpropagation*, ovvero l'algoritmo di retropropagazione dell'errore, che permetteva la modifica sistematica dei pesi delle connessioni, in modo da rendere la risposta della rete quanto più vicina a quella desiderata.

Tale algoritmo, proposto nel 1986 da David E. Rumelhart, G. Hinton e R. J. Williams consentì di superare le problematiche legate al perceptrone di Rosenblatt e permise di risolvere il problema della separabilità non lineare delle soluzioni, rendendo quindi possibile calcolare la funzione *XOR* e segnando il definitivo rilancio delle reti neurali.

1.2 Analogie con il sistema nervoso ed applicazioni delle ANN

Come accennato nel paragrafo precedente, una rete neurale è un sistema computazionale costruito basandosi sui processi biologici naturali, il cui obiettivo è la riproduzione delle attività tipiche del cervello umano, ad esempio la comprensione del linguaggio, la percezione di immagini, il riconoscimento di forme

In altre parole, lo scopo di una rete neurale artificiale è l'emulazione del sistema nervoso animale, in particolar modo di quello umano, il quale presenta numerose caratteristiche che risultano essere ottime per la riproduzione di un sistema computazionale che lo imiti: è flessibile poiché si adatta ad ogni tipologia di situazione imparando, è robusto in quanto le cellule nervose muoiono ogni giorno senza avere effetti significativi sulla performance del sistema, è resistente, piccolo e dissipa poca energia.

Tale riproduzione non deve essere intesa come atto alla costruzione di un cervello artificiale. Infatti le caratteristiche delle reti neurali biologiche, riprese dalla computazione neurale artificiale, sono in esigua minoranza: gli stessi neuroni artificiali sono solo un'approssimazione dei neuroni biologici e sono in grado di riprodurre solo tre dei circa centocinquanta processi che sono tipici dei neuroni del cervello umano.

Il sistema nervoso umano può essere pensato, quindi, come una grande struttura computazionale formata da milioni di unità fortemente interconnesse tra loro in modo parallelo e che riesce a trasformare continui input in output ragionevoli. Le reti neurali rappresentano una riproduzione significativa di tale struttura, in particolar modo degli algoritmi di apprendimento e di ottimizzazione, basati su un modello *connessionistico* di calcolo: le operazioni responsabili dello scambio di informazioni avvengono per mezzo dell'interazione tra le unità elementari. Sono sistemi altamente paralleli: fornendo i dati del problema alle unità di input, la computazione si propaga in parallelo nella rete fino alle unità di output che producono il risultato.

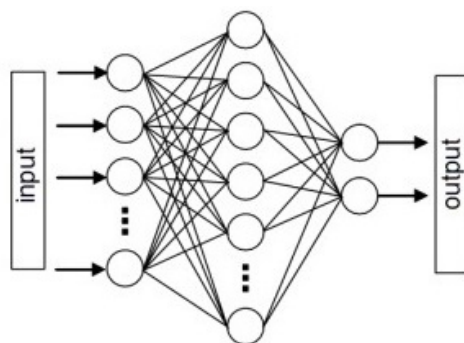


Figura 1.1: Esempio di rete neurale

Il campo tipico di applicazione delle reti neurali è quello dell'individuazione di legami di **ingresso-uscita** all'interno di sistemi complessi in cui i dati a disposizione sono molto numerosi o contenenti poca informazione e quindi non risulta chiaro quali relazioni deterministiche esistano tra le diverse variabili che caratterizzano il problema. Si tratta di campi in cui l'analisi statistica di tutte le variabili risulta difficoltosa o dispendiosa in termini di calcolo.

Ecco un breve elenco di applicazioni note:

- Elaborazione delle immagini (compressione e miglioramento di immagini in tempo reale);
- Elaborazione del suono (compressione e miglioramento in tempo reale della voce e della musica, aiuti audio e protesi ecc...);
- Rivelazione, verifica e identificazione di caratteri e oggetti (codici a barre, impronte digitali, firme, facce ecc...);
- Riconoscimento della voce (elaborazione di parole dettate, identificazione di persone per via vocale, selezione telefonica ecc...);
- Applicazioni basate su ingressi di origine sensoriale (ingresso tattile, visivo, acustico, olfattivo);
- Archivi (musica, video, immagini ecc...);
- Sintesi e miglioramento dati (fax, grafica, cancellazione di rumore, apparecchiature televisive ad alta risoluzione ecc...);

- Robot autonomi.

1.3 Il neurone biologico

Per convincersi a fondo sulla connessione tra le reti neurali artificiali e le reti neurali del sistema nervoso umano, e le numerose analogie che ne derivano, è bene dare una breve descrizione del secondo sistema, in modo da far emergere le principali caratteristiche ed i principali costituenti che sono utili al fine della comprensione delle reti neurali.

Il sistema nervoso è diviso in tre stadi:

- I *recettori*, che convertono gli stimoli esterni in impulsi elettrici che vengono inviati alla rete neurale;
- La *rete neurale*, la quale riceve gli impulsi elettrici provenienti dai recettori e li immagazzina come informazioni utili per poter prendere delle decisioni che vengono inviate, anch'esse sotto forma di impulsi elettrici, agli attuatori;
- Gli *attuatori* responsabili della trasformazione degli impulsi in risposte per l'ambiente esterno.

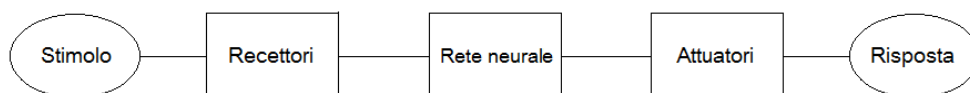


Figura 1.2: Stadi del sistema nervoso

Nel sistema nervoso centrale sono presenti circa 10^{11} cellule nervose, i *neuroni*. Tali cellule sono connesse strettamente tra loro attraverso numerosi collegamenti che nel loro insieme vanno a formare la rete neurale.

Un neurone è formato da un corpo centrale, identificato con il nome di

soma, all'interno del quale è presente il nucleo, e da molti prolungamenti citoplasmatici, detti *neuriti*, che si distinguono in:

- *dendriti*, organizzati con diramazioni ad albero che costituiscono il *ramo dendritico*;
- *assone*, la cui parte finale prende il nome di *bottone sinaptico*.

I dendriti ricevono segnali dai neuroni afferenti e li propagano verso il nucleo. L'assone conduce, invece, il segnale verso altre cellule grazie alla presenza del bottone sinaptico alla sua estremità, che risulta essere un'ulteriore ramificazione: quest'ultima va a formare i terminali attraverso i quali i segnali elettrici vengono trasmessi.

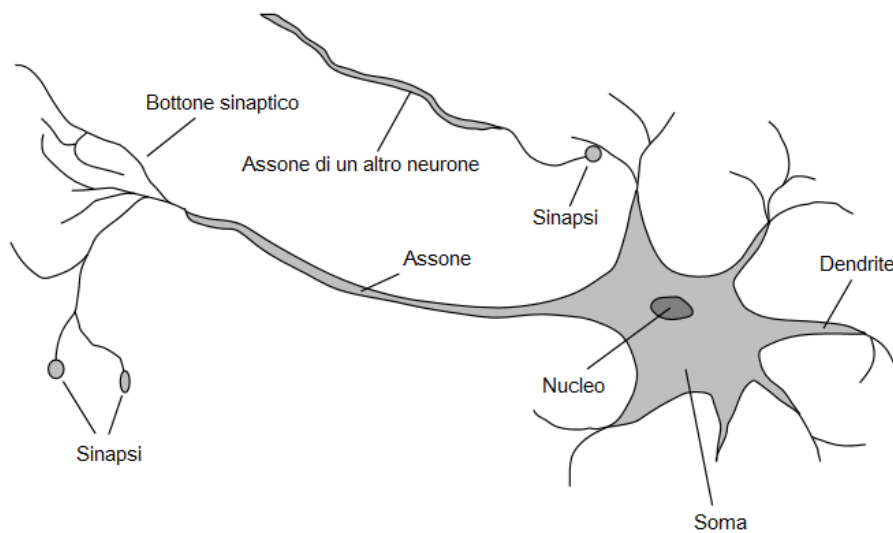


Figura 1.3: Neurone biologico

Tra un terminale di un assone e la cellula ricevente, esiste uno spazio che viene superato dagli impulsi elettrici per mezzo dei *neurotrasmettitori*, sostanze chimiche che guidano le informazioni tra le cellule e che si suddividono in *eccitatori* se favoriscono la creazione dell'impulso e *soppressori* se inibiscono l'impulso.

Il punto di connessione tra il terminale di un neurone ed il dendrite di un altro costituisce una struttura altamente specializzata che prende il nome di *sinapsi* e che risulta quindi essere la responsabile delle interazioni. La sinapsi svolge due processi: un processo *presinaptico* in cui viene liberato il neurotrasmettitore ed un processo *postsinaptico*, azionato dal neurotrasmettitore, che rigenera il segnale elettrico. Quindi una sinapsi converte un segnale elettrico in chimico durante il processo postsinaptico per poi riconvertirlo in elettrico durante la postsinapsi.

Un neurone trasmette un impulso elettrico lungo il suo assone nel momento in cui si verifica una differenza di potenziale elettrico tra l'interno e l'esterno della cellula, provocando la liberazione di un neurotrasmettitore di cui sopra.

1.4 Modello del neurone artificiale

La breve illustrazione del sistema neurale umano nel paragrafo precedente suggerisce uno schema per delineare l'organizzazione delle reti neurali artificiali: queste ultime sono formate da un elevato numero di unità computazionali, che possono essere equiparate ai neuroni umani, capaci di eseguire una somma pesata. Tali unità sono collegate tra loro attraverso delle connessioni, così come le sinapsi collegano i neuroni nella rete umana.

Consideriamo una generica unità j costituita da n canali di ingresso x_1, x_2, \dots, x_n . Gli input provenienti da strati precedenti o direttamente dall'esterno entrano nel neurone tramite tali canali.

Sulle connessioni sono presenti dei *pesi sinaptici* w_i , numeri reali che denotano l'*efficacia sinaptica*, ovvero la forza della connessione. Se $w_i > 0$ il canale è detto *eccitatorio*, se $w_i < 0$ il canale è *inibitorio*.

I segnali in entrata, pesati dalle rispettive sinapsi, sono convogliati nel *soma* del neurone artificiale, all'interno del quale vengono sommati producendo una combinazione lineare così definita:

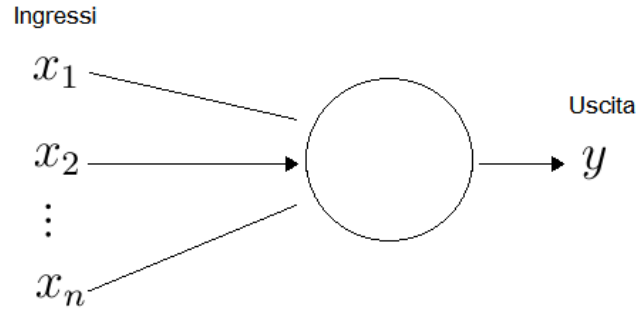


Figura 1.4: Canali di ingresso in un neurone

$$\sum_{i=1}^n w_i x_i$$

(1.1)

La somma pesata degli ingressi viene indicata con la parola *net* ed il segnale con cui il neurone trasmette la sua attività all'esterno è calcolato applicando una *funzione di attivazione* φ che limita l'ampiezza dell'output; si assume per comodità che le ampiezze degli output appartengono all'intervallo $[0, 1]$ oppure $[-1, 1]$.

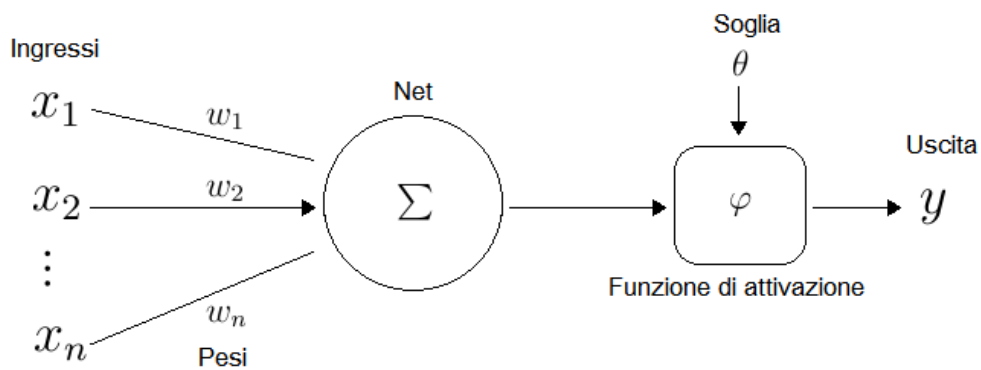


Figura 1.5: Modello neurone

Il modello neuronale include anche un valore *soglia* che ha l'effetto, a seconda della sua positività o negatività, di aumentare o diminuire il valore in ingresso alla funzione di attivazione.

L'output finale sarà allora:

$$y = \varphi\left(\sum_{i=1}^n w_i x_i\right) \quad (1.2)$$

E se indichiamo con θ il valore di soglia, la (1.2) diventerà:

$$y = \varphi\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (1.3)$$

Interpretando la soglia come il peso associato ad un ulteriore canale di ingresso x_0 , e quindi $w_0 = \theta$, potremmo anche scrivere :

$$y = \varphi\left(\sum_{i=0}^n w_i x_i\right) \quad (1.4)$$

Il modello finale sarà:

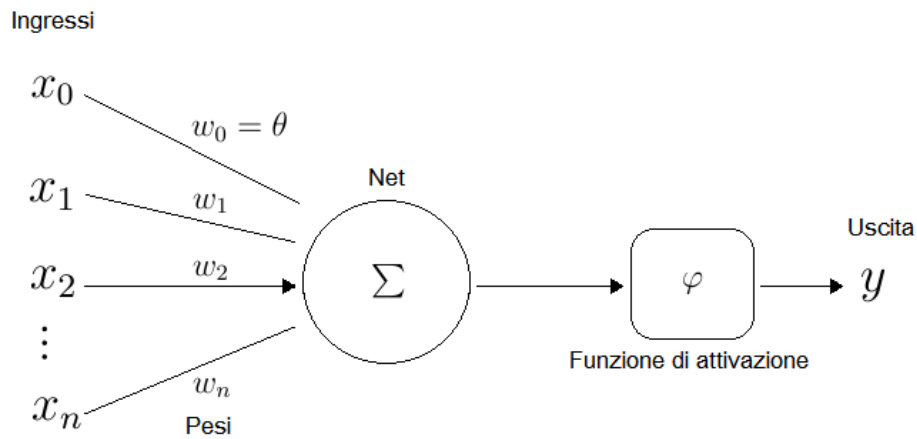


Figura 1.6: Modello neurone con soglia

L'effetto di un segnale x_i sul neurone è quindi uguale al prodotto $w_i \cdot x_i$ dove w_i è il peso attribuito alla sinapsi corrispondente ed il potenziale di attivazione è dato dalla somma algebrica dei prodotti di tutti i segnali di ingresso e dei valori dei pesi corrispondenti.

Schematizzando, ed individuando il neurone artificiale come l'unità di calcolo fondamentale della rete neurale, gli elementi base che lo rappresentano sono:

- Un insieme di connessioni;
- Un sommatore;
- Una funzione di attivazione;
- Un valore di soglia.

1.5 Attivazione di un neurone e funzioni di attivazione

Nelle reti neurali si può avere una *attivazione asincrona*, in cui i neuroni si attivano uno per volta, ed una attivazione sincrona in cui tutti i neuroni si attivano contemporaneamente. L'attivazione di un neurone dipende solo dall'informazione ricevuta ed è indipendente dalle operazioni svolte dagli altri nodi.

La funzione di attivazione determina il tipo di risposta che un neurone è in grado di emettere. Definisce, quindi, l'uscita di un neurone in funzione del livello di attivazione. L'uscita può essere un numero reale, un numero reale appartenente ad un intervallo, oppure un numero appartenente ad un insieme discreto.

1.5.1 Funzione a soglia

Imponendo $a = \sum_{i=0}^n w_i x_i$, il valore di uscita di un neurone assunto tramite una funzione a soglia é:

$$y = \varphi(a) = \begin{cases} 1 & \text{se } a \geq 0 \\ 0 & \text{se } a < 0 \end{cases} \quad (1.5)$$

ed il grafico relativo

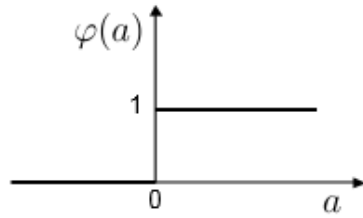


Figura 1.7: Funzione di attivazione a soglia

Se si trascuria il contributo dell'ingresso x_0 dal livello di attivazione e si ha quindi $a = \sum_{i=1}^n w_i x_i$, si avrà

$$y = \varphi(a) = \begin{cases} 1 & \text{se } a \geq \theta \\ 0 & \text{se } a < \theta \end{cases} \quad (1.6)$$

Il rispettivo grafico sarà

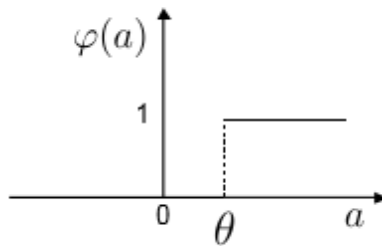


Figura 1.8: Funzione a soglia senza contributo x_0

A volte è opportuno che la funzione di attivazione assuma valori tra -1 e $+1$ ed in questo caso la funzione a soglia, che diviene la ben nota *funzione segno* è ridefinita così :

$$y = \varphi(a) = \begin{cases} 1 & \text{se } a > 0 \\ 0 & \text{se } a = 0 \\ -1 & \text{se } a < 0 \end{cases} \quad (1.7)$$

1.5.2 Funzione lineare

Se $a = \sum_{i=0}^n w_i x_i$, si avrà:

$$y = \varphi(a) = a \quad (1.8)$$

Il grafico rispettivo

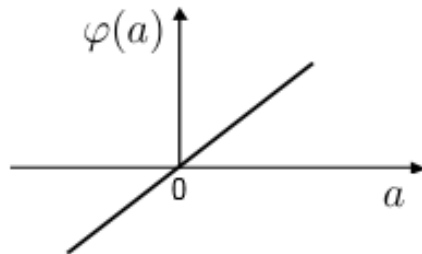


Figura 1.9: Funzione lineare

1.5.3 Funzione lineare a tratti

Un esempio di funzione di attivazione lineare a tratti è:

$$y = \varphi(a) = \begin{cases} 1 & \text{se } a \leq -0.5 \\ a + 0.5 & \text{se } -0.5 < a < 0.5 \\ 0 & \text{se } a \geq 0.5 \end{cases} \quad (1.9)$$

Che rappresentata è

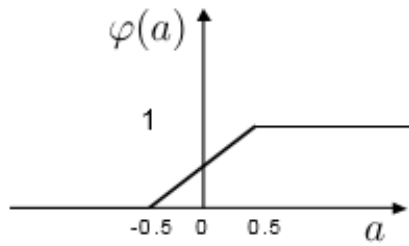


Figura 1.10: Funzione a tratti

1.5.4 Funzione sigmoide

La funzione sigmoide appartenente alla famiglia di funzioni continue non lineari ed è tra le più utilizzate.

Tra queste funzioni riportiamo la *funzione logistica* così definita:

$$y = \varphi(a) = \frac{1}{1 + e^{-a}} \quad (1.10)$$

imponendo sempre $a = \sum_{i=0}^n w_i x_i$.

Essa assume questa forma

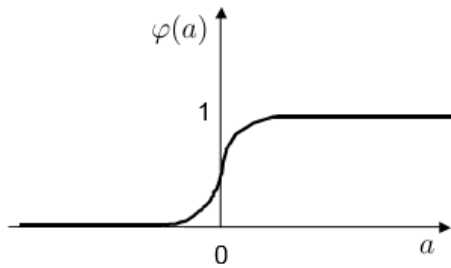


Figura 1.11: Funzione sigmoide

1.6 Architettura di una rete neurale

L'architettura di una rete neurale artificiale è caratterizzata da:

- Numero di strati di sinapsi;
- Numero di neuroni presenti nell'*input layer*, ovvero lo strato di ingresso;
- Numero di neuroni nell'*output layer*, lo strato di uscita.

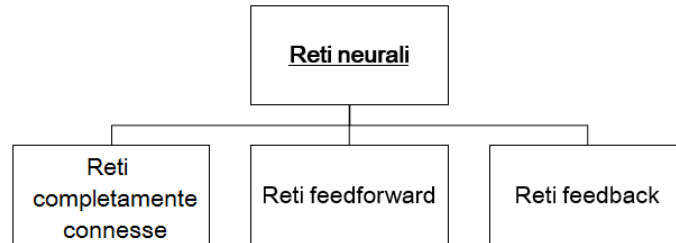


Figura 1.12: Suddivisione delle reti neurali

Le reti neurali si suddividono principalmente in due grandi classi: le reti *feedforward* e le reti *feedback* o *ricorrenti*. Si ha inoltre un'altra tipologia di reti che sono le *reti completamente connesse*.

Reti completamente connesse

Nelle reti completamente connesse ogni neurone è connesso con tutti gli altri. Le connessioni tra i neuroni sono bidirezionali e possono essere rappresentate per mezzo di una matrice quadrata W , di dimensione pari al numero di neuroni. Un suo generico elemento $w_{i,j}$ rappresenta il peso della connessione tra il neurone i ed il neurone j .

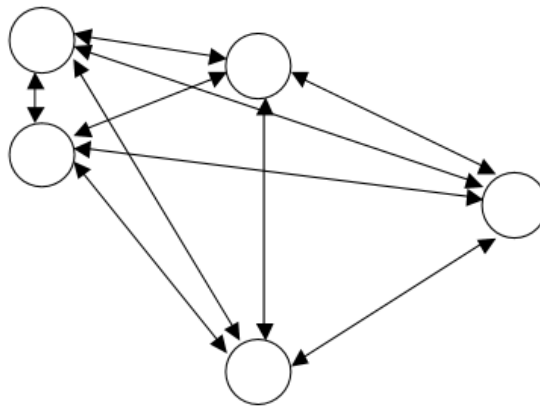


Figura 1.13: Rete completamente connessa

Reti feedforward a strati

In queste reti i segnali viaggiano dallo strato di ingresso verso lo strato di uscita e pertanto vengono anche chiamate **reti feedforward**. Nelle reti stratificate non esistono connessioni tra i neuroni all'interno di uno stesso strato, nè tra neuroni di strati non limitrofi. Ogni neurone in un generico strato è connesso con tutti quelli dello strato successivo ed i neuroni dello strato di ingresso hanno come unico compito quello di trasmettere i segnali ricevuti allo strato successivo, all'interno di essi non avviene alcuna computazione.

Le reti feedforward a strati si distinguono in base al numero di strati che presentano, numero che dipende dallo specifico problema che si intende risolvere.

- *Reti feedforward ad uno strato* : questa è una forma semplice di reti a strati. Il segnale nella rete si propaga in avanti senza cicli, **NON** ci sono connessioni che tornano indietro e nemmeno connessioni trasversali nel layer di output.

- *Reti feedforward a più strati* : le reti a più strati sono anche dette *reti multilivello* , in inglese **Multi-Layer Perceptron, MLP**. Tra l'input layer e l'output layer sono presenti uno o più strati di neuroni nascosti, si parla quindi di *hidden layers*. Nelle MLP non esistono connessioni nè tra neuroni di uno stesso strato nè tra neuroni di strati non limitrofi. Ogni strato ha connessioni entranti dal precedente strato e uscenti in quello successivo, quindi la propagazione del segnale avviene in avanti in modo aciclico e senza connessioni trasversali.

Tali reti vengono utilizzate per superare problemi che possono sorgere nella discriminazione dei segnali: attraverso i neuroni nascosti si ottengono delle rappresentazioni interne dei segnali di input che consentono il riconoscimento di forme più complesse, facilitando il compito della rete. Nonostante tutto, però, l'aggiunta di ulteriori strati nascosti non ottimizza le abilità di discriminazione della rete; questo è possibile solo se la funzione di attivazione è non lineare (una rete multistrato a neuroni lineari è sempre riconducibile ad una rete con due soli strati).

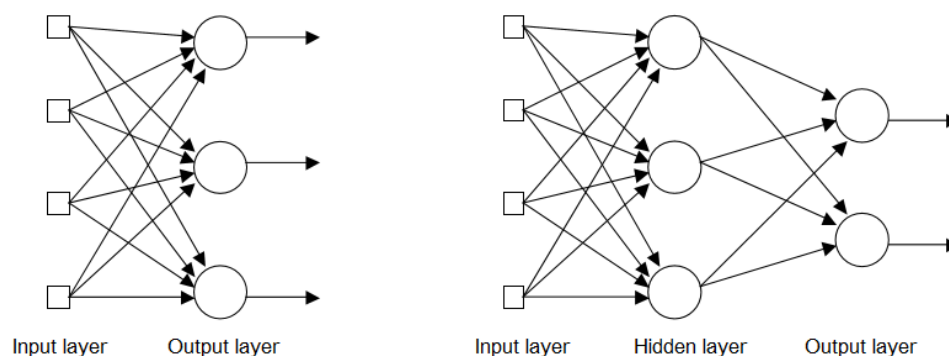


Figura 1.14: Rete ad uno strato (a sinistra), rete multilivello (a destra)

Reti ricorrenti

Una rete neurale ricorrente (RNN) si distingue dalle precedenti nel fatto che è ciclica. Il sistema è dinamico, le unità di output sono connesse con quelle intermedie e con quelle di input, si ha una *retroazione*. Dato un determinato stimolo, la risposta della rete ricorrente non viene dettata soltanto dai caratteri strutturali della rete stessa, come si verifica nella rete feedforward, ma varia in funzione del precedente contesto in cui si è manifestato lo stimolo. L'output non è determinato solo dall'input, ma anche da una *cronologia* di input che fornisce una forma di memoria a breve termine. Queste reti sono adatte per dati strutturati temporalmente ed hanno anche la possibilità di avere un'attività in assenza di una stimolazione input, poiché gli impulsi che viaggiano sulle vie ricorrenti possono essere sufficienti a mantenere il sistema in attività.

Capitolo 2

L'apprendimento delle reti neurali

2.1 Addestramento e generalizzazione

Per costruire una rete neurale efficiente, un passo fondamentale è individuare un **insieme di apprendimento** ed un **algoritmo di apprendimento**.

Per insieme di apprendimento si intende una collezione di esempi, chiamata (**cioè un**) **training set**, dai quali la rete può attingere per raggiungere lo scopo per il quale è stata progettata; per algoritmo, invece, un procedimento che permetta di prelevare le informazioni dall'insieme di apprendimento e di fissare dei parametri che vengano poi modificati attraverso operazioni iterative interfaccianti con l'ambiente.

L'apprendimento avviene sempre grazie ad un certo numero di esempi prelevati dal mondo reale ed opera in due fasi distinte:

- Fase di apprendimento o addestramento nota come **learning** o **training**;
- Fase di generalizzazione nota come **recall**.

Durante la fase di learning si cerca di far imparare alla rete tutte le informazioni contenute nel training set, ottenendo un modello che verrà poi utilizzato nella fase di generalizzazione per analizzare nuovi ingressi.

L'architettura della rete neurale gioca un ruolo fondamentale per l'efficienza nella fase di apprendimento. Non bisogna però immaginare che con l'aumentare del numero di unità, cresce il potere computazionale della rete. Infatti, in tal caso, ci si troverebbe di fronte ad una situazione di *overfitting*, un adattamento forzato della rete dato che risulterebbe avere un numero eccessivo di parametri rispetto al numero di esempi forniti. Questa situazione tende a far diminuire la capacità della rete di generalizzare su nuovi esempi originando una sorta di principio di indeterminazione dell'apprendimento che può essere affrontato con la limitazione del numero degli ingressi.

Per **generalizzazione** di una rete, si intende la sua capacità di fornire le risposte appropriate a pattern di input che non sono mai stati incontrati. Essa risulta essere un punto di forza delle reti neurali e dipende dal numero di unità nascoste, dai pesi sinaptici e dal numero di training record.

Una rete neurale “generalizza bene” se la mappa input/output che essa genera è corretta per esempi di test mai presentati in fase di training (si assume che gli esempi di test siano tratti dalla stessa popolazione usata per generare il training set). In questi termini essa produce mappature input/output corrette anche se l'input è lievemente differente dagli esempi usati in fase di training; se però la rete viene addestrata con troppi esempi, si rischia che quest'ultima memorizzi il training set trovandosi nella condizione di *over-training*.

Per avere una struttura ottimale della rete dal punto di vista della capacità di generalizzazione una guida adatta è il **Principio di Occam**, che suggerisce molto sulla complessità delle reti e sulla loro adeguatezza e che in relazione a queste ultime può essere enunciato nel seguente modo:

“Date due reti che soddisfano l'insieme di apprendimento, la rete di minore complessità è quella che si comporta meglio su esempi non visti, cioè ha la migliore capacità di generalizzazione.”

2.2 Paradigmi di apprendimento

Le reti neurali si ispirano al tratto caratteristico del sistema nervoso, ovvero la capacità di acquisire esperienza da esempi del mondo reale: per questo, oltre che di apprendimento, si parla di *addestramento* delle reti neurali attraverso dei *paradigmi di apprendimento*.

I paradigmi di apprendimento si suddividono in:

- *apprendimento supervisionato*
- *apprendimento non supervisionato*

Apprendimento supervisionato

Tale paradigma presuppone un *training set* cioè un set di esempi nel quale sono presenti coppie del tipo (x_k, y_{dk}) dove la prima variabile indica il k -esimo ingresso e la seconda la k -esima uscita desiderata. Con y_k si indica l'uscita reale e la si confronta con l'uscita desiderata: l'obiettivo è modificare i pesi affinché si minimizzi la differenza tra le due uscite. Il training set iniziale viene proposto ripetutamente finché $y_k \approx y_{dk}$, ovvero l'uscita reale sia il più simile possibile a quella desiderata, il tutto modificando i pesi in base alla legge di apprendimento scelta.

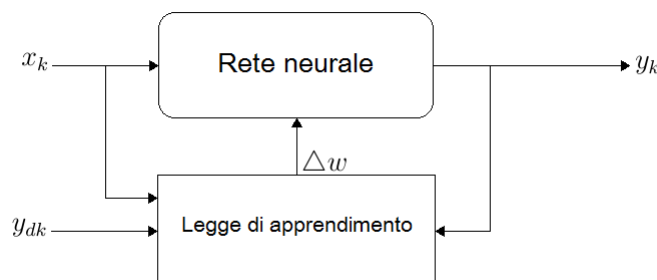


Figura 2.1: Apprendimento supervisionato

Apprendimento non supervisionato

Nell'apprendimento non supervisionato la rete modifica i pesi autonomamente, si auto-organizza. Viene fornito solo il training set senza precisare le uscite.

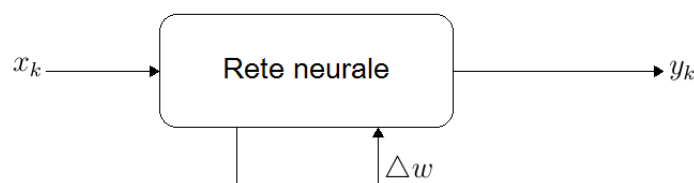


Figura 2.2: Apprendimento non supervisionato

2.3 Processi di apprendimento

L'addestramento è un processo ad hoc dipendente dallo specifico problema trattato. Riportiamo alcuni processi comunemente usati .

2.3.1 Delta rule

Con regola delta, o *delta rule*, si intende un apprendimento con correzione dell'errore. Siano:

$\bar{x}_k = [x_1, x_2, \dots, x_n]^T$ il vettore k -esimo degli ingressi,

$\bar{W}_k = [w_1, w_2, \dots, w_n]^T$ il vettore dei pesi al k -esimo ingresso,

e siano y_k ed y_{dk} rispettivamente l'uscita ottenuta e l'uscita desiderata.

Definiamo l'*errore* come:

$$\delta_k = y_k - y_{dk} \quad (2.1)$$

Allora se consideriamo la variazione del generico vettore dei pesi \overline{W}_k si ha che:

$$\Delta \overline{W}_k = \eta \overline{x}_k \delta_k \quad (2.2)$$

Il numero η che compare nell'equazione (2.2) è compreso tra 0 ed 1, prende il nome di **learning rate** ed indica la velocità di apprendimento del neurone. Questa regola modifica in maniera proporzionale solo i pesi delle connessioni che hanno contribuito all'errore.

L'algoritmo è il seguente:

- 1 - se $y_k = y_{dk}$ nessuna modifica dei pesi
- 2 - se $y_k \neq y_{dk}$ allora $\Delta \overline{W}_k = \eta \overline{x}_k \delta_k$.

2.3.2 Apprendimento con correzione di errore o metodo discesa del gradiente

Se indichiamo con y_k una risposta generata da un segnale di stimolo x in un tempo t ed indichiamo con y_{dk} la risposta desiderata, avremo un *segnale di*

errore che può essere così espresso:

$$\delta = y_{dk} - y_k \quad (2.3)$$

Tale segnale dà il via ad un meccanismo di controllo che va ad applicare una sequenza di modifiche ai pesi sinaptici del neurone interessato, al fine di avvicinare la risposta ottenuta a quella desiderata.

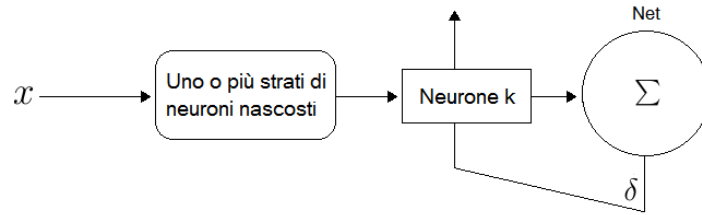


Figura 2.3: Apprendimento con correzione

Questo processo di ricerca dei pesi migliori si basa sulla scelta di pesi che minimizzano una funzione errore $E(\overline{W})$, costruita al variare dei pesi stessi. Per la scelta di tali pesi si sfruttano le informazioni fornite dal gradiente locale della funzione errore costruita.

Se con $\nabla_w E(\overline{W})$ indichiamo il gradiente della funzione, allora:

$$\Delta W \propto \nabla_w E(\overline{W}) \quad (2.4)$$

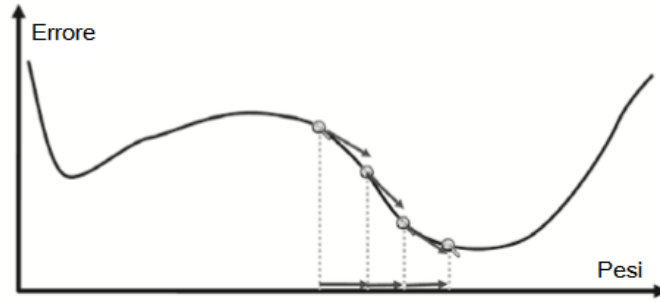


Figura 2.4: Metodo della discesa del gradiente

La ricerca risulta quindi essere guidata in modo proporzionale e dato che:

$$\frac{dE(\bar{w})}{d\bar{w}} = \nabla_w E(\bar{w}) = \left(\frac{\partial E(\bar{w})}{\partial w_1}, \frac{\partial E(\bar{w})}{\partial w_2}, \dots, \frac{\partial E(\bar{w})}{\partial w_n} \right)^T \quad (2.5)$$

si avrà:

$$\Delta \bar{W}_k = \eta \frac{dE(\bar{w})}{d(\bar{w})} \quad (2.6)$$

Se consideriamo la k -esima uscita y_k e la rispettiva uscita desiderata y_{dk} ed indichiamo con g la funzione di attivazione:

$$\begin{cases} y_k = g(net_k) \\ \delta_k = y_k - y_{dk} \end{cases}$$

$$(2.7)$$

Si definisce l'**errore quadratico medio**:

$$E_k = \frac{1}{2}(y_k - y_{dk})^2 \quad (2.8)$$

Ma allora se consideriamo l' i -esimo peso del k -esimo input si avrà che:

$$\Delta w_i = \eta \frac{\partial E_k}{\partial w_i} \quad (2.9)$$

e sviluppando:

$$\begin{aligned} \frac{\partial E_k}{\partial w_i} &= \frac{\partial}{\partial w_i} \left[\frac{1}{2} (y_k - y_{dk})^2 \right] = (y_k - y_{dk}) \frac{\partial (y_k - y_{dk})}{\partial w_i} = (y_k - y_{dk}) \frac{\partial y_k}{\partial w_i} = \\ &= (y_k - y_{dk}) \frac{\partial g(net_k)}{\partial w_i} = (y_k - y_{dk}) \frac{\partial g(net_k)}{\partial net_k} \frac{\partial net_k}{\partial w_i} = \delta_k g'(net_k) x_i \end{aligned} \quad (2.10)$$

Quindi il peso w_i sarà modificato:

$$\Delta w_i = \eta \frac{\partial E_k}{\partial w_i} = \eta \delta_k g'(net_k) x_i \quad (2.11)$$

2.3.3 Apprendimento di Hebbian

Come nella delta rule, definiamo:

$\bar{x}_k = [x_1, x_2, \dots, x_n]^T$ il vettore k -esimo degli ingressi,

$\bar{W}_k = [w_1, w_2, \dots, w_n]^T$ il vettore dei pesi al k -esimo ingresso,

e siano y_k ed y_{dk} rispettivamente l'uscita ottenuta e l'uscita desiderata.

Nell'apprendimento di tipo hebbiano viene modificato ogni valore del peso di una connessione in questo modo:

$$\Delta \bar{W}_k = \eta \bar{x}_k y_{dk} \quad (2.12)$$

con η learning rate.

I passi dell'algoritmo sono i seguenti:

- 1- se $y_k = y_{dk}$ nessuna modifica pesi
- 2- se $y_k > y_{dk}$ allora $\Delta \bar{W}_k = -\eta \bar{x}_k y_{dk}$
- 3- se $y_k < y_{dk}$ allora $\Delta \bar{W}_k = \eta \bar{x}_k y_{dk}$

2.3.4 Apprendimento competitivo

Tale apprendimento si basa su una vera e propria competizione tra i neuroni di uscita di una rete neurale per attivarsi in seguito ad uno stimolo. In un certo momento t può attivarsi un solo neurone che viene denominato **winners-takes-all**.

In tale processo è importante avere quindi un meccanismo che permetta ai neuroni di competere ed un valore che indichi il limite della “forza” di ciascun neurone.

Nella forma più semplice, la rete neurale ha un solo strato di neuroni di uscita, completamente connessi ai nodi di input e che possono essere connessi tra loro.

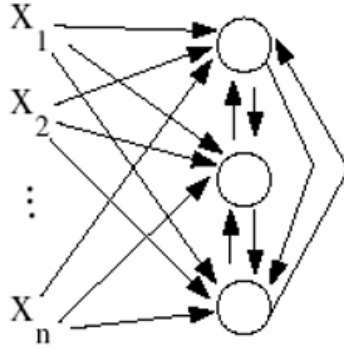


Figura 2.5: Struttura rete con apprendimento competitivo

Il neurone k che si attiva è quello con input netto ν_k più alto per un dato input x , con ν_k combinazione lineare di tutti gli input. Il suo segnale di output y_k sarà 1, mentre quello degli altri neuroni rimasti inattivi è 0.

$$y_k = \begin{cases} 1 & \text{se } \nu_k > \nu_j \quad \forall j, j \neq k; \\ 0 & \text{altrimenti} \end{cases} \quad (2.13)$$

Ora se indichiamo con $w_{k,j}$ i pesi tra il neurone k e l'input j , supponendo che ogni neurone abbia un ammontare di peso sinaptico fisso $\sum_j w_{kj} = 1, \forall k$, il neurone che vince apprende spostando i pesi dagli input inattivi agli input attivi utilizzando tale regola:

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & \text{se il neurone } k \text{ vince;} \\ 0 & \text{altrimenti} \end{cases} \quad (2.14)$$

2.4 Attività di apprendimento

La scelta del paradigma di apprendimento, e quindi la metodologia di addestramento di una rete neurale, avviene in base all'attività che la rete deve svolgere, ovvero il problema che deve risolvere. Le reti neurali possono essere utilizzate come *classificatori*, in grado cioè di riconoscere oggetti appartenenti a diverse categorie o possono essere responsabili dell'approssimazione di funzioni. Data una funzione nota per punti, di cui non si conosce la forma analitica, il compito della rete neurale è quello di approssimarla nel modo migliore possibile, calcolandone il valore anche in punti diversi da quelli proposti in ingresso.

2.4.1 Associazione di pattern e memoria associativa

Per memoria associativa intendiamo un criterio di memorizzazione e recupero di informazioni attraverso l'associazione. Completamente ispirata al metodo di memorizzazione del cervello umano, consente il recupero dell'informazione sulle basi di una conoscenza parziale del suo contenuto senza conoscerne la locazione di memoria.

La memoria associativa si articola in due fasi:

- Fase di memorizzazione: la rete viene addestrata per fare in modo che dei *pattern*, vettori di numeri reali, vengano memorizzati ed associati;

- Fase di richiamo: si richiama dalla rete un pattern memorizzato a seguito della presentazione di una versione parziale o distorta di un pattern chiave.

Le *associazioni* possono essere di due forme:

- *Autoassociazioni*: generalmente utilizzate nell'apprendimento non supervisionato, la rete neurale memorizza un insieme di pattern che vengono ripetutamente presentati. Tra essi ne viene presentato in particolare uno che risulta essere distorto o in forma parziale. L'obiettivo è che la rete riconosca il pattern e ne richiami la versione completa/originale precedentemente memorizzata.
- *Eteroassociazioni*: ogni pattern è associato ad un altro pattern. Se x_k è un generico pattern chiave associato al pattern memorizzato y_k , allora x_k opera in modo tale da richiamare y_k .

2.4.2 Riconoscimento di pattern

Tale processo si articola in due fasi: la prima fase consiste in una sessione di addestramento in cui alla rete vengono presentati ripetutamente un insieme di pattern specificando per ognuno la classe di appartenenza; nella seconda fase alla rete viene presentato un pattern mai mostrato precedentemente, ma appartenente ad una categoria di pattern che la rete ha memorizzato. La rete neurale sarà in grado di riconoscerne la categoria e classificare il pattern grazie ai dati precedentemente acquisiti.

Il sistema è diviso in tre spazi:

- Lo *spazio dei dati* costituito da tutti i pattern, un generico pattern x identifica un punto dimensionale di tale spazio;
- Lo *spazio delle caratteristiche* all'interno del quale sono presenti dei punti y ;

- Lo *spazio decisionale* multidimensionale suddiviso in regioni determinate dalla rete nella fase di addestramento, ognuna delle quali è associata ad una classe.

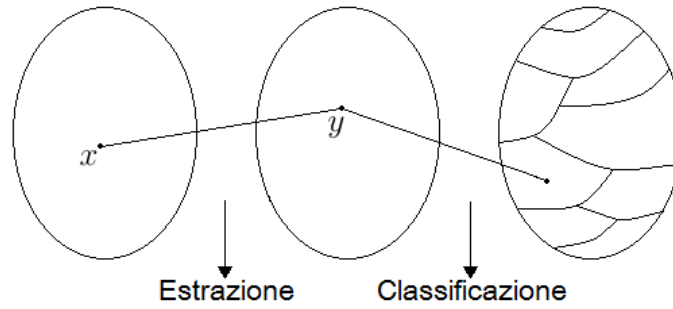


Figura 2.6: Classificazione di pattern

Il processo di riconoscimento inizia con l'estrazione delle *features* ovvero le caratteristiche che un generico pattern nello spazio dei dati deve rispettare. L'*estrazione* è una trasformazione che collega il pattern x con un punto y nello spazio delle caratteristiche. Infine un'ultima trasformazione detta *classificazione* mappa dal punto y in una regione dello spazio decisionale.

2.4.3 Approssimazione di funzioni

Si prenda in esame una rappresentazione non lineare input/output descritta da $y = f(x)$ dove x è l'input e y è l'output e si consideri un insieme di esempi (x_i, y_i) per $i = 1, 2, \dots, n$. L'incognita è la funzione f ed il problema consiste nel creare una rete capace di generare una rappresentazione F quanto più vicina alla funzione f . In senso euclideo bisogna soddisfare la condizione

$$\|F(x) - f(x)\| < \varepsilon \quad \forall x$$

dove ε è un numero positivo piccolo, che risulterà essere sempre più piccolo con l'accrescere delle dimensioni e dei parametri liberi della rete.

2.5 Un esempio di apprendimento supervisionato: il percettrone elementare

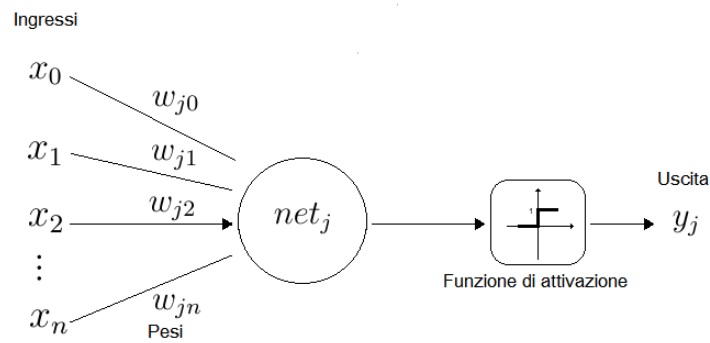


Figura 2.7: Il percettrone

Proposto da Rosenblatt, il percettrone è la forma più semplice di rete neurale e coincide con la struttura del neurone formale. Esso è costituito da un singolo neurone in cui confluiscono più ingressi i cui pesi sinaptici sono modificabili. La funzione di attivazione è una funzione a gradino detta *funzione di Heavside*:

$$g(net_j) = \begin{cases} 1 & \text{se } net_j \geq 0 \\ 0 & \text{se } net_j < 0 \end{cases} \quad (2.15)$$

Il percettrone divide lo spazio di ingresso in due regioni mediante l'iperpiano in \mathfrak{R}^n di equazione:

$$w_0 + \sum_{i=1}^n w_i x_i = 0 \quad (2.16)$$

ed è quindi in grado di risolvere solo problemi *linearmente separabili*.

Supponiamo di voler classificare in due classi distinte, C_1 e C_2 , oggetti rappresentati mediante punti nel piano e di sfruttare $n = 2$ ingressi. Se le due classi sono linearmente separabili, l'iperpiano è una retta che funge da separatore lineare tra le due classi. Un oggetto sarà quindi classificato in base alla parte di semipiano in cui il suo punto di rappresentazione cadrà.

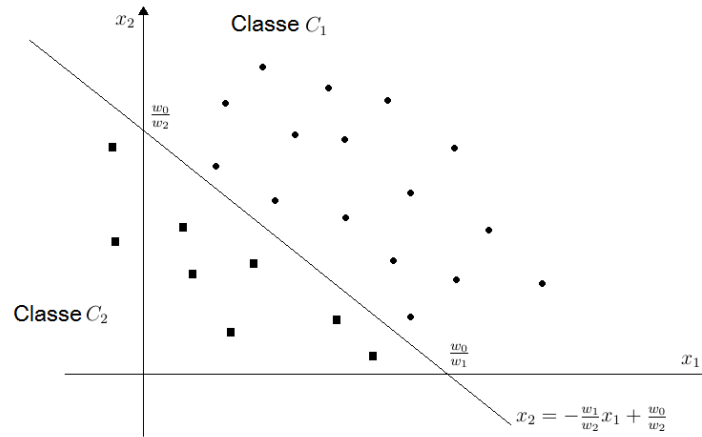


Figura 2.8: Separazione lineare dello spazio di ingresso di un Perceptron

Nel piano (x_1, x_2) degli ingressi della rete le classi C_1 e C_2 sono rappresentate da due semipiani separati dalla retta $x_2 = -\frac{w_1}{w_2}x_1 + \frac{w_0}{w_2}$ e possiamo classificare gli stimoli in questo modo:

$$\begin{cases} x \in C1 & \text{se } y = 1 \\ x \in C2 & \text{se } y = 0 \end{cases}$$

(2.17)

L'apprendimento utilizzato nel perceptron è di tipo supervisionato e la learning rule attraverso i quali si modificano i singoli pesi in modo appropriato è la delta rule, definita in forma generale come:

$$\overline{W}_{k+1} = \overline{W}_k + \Delta \overline{W}_k$$

(2.18)

dove $\overline{W}_k = [w_1, w_2, \dots, w_n]^T$ è il vettore dei pesi al k -esimo ingresso.

I passi da eseguire sono:

- 1 - Si inizializzano i pesi con valori casuali
- 2 - Si presenta un ingresso x_k ed il valore di uscita desiderato y_{dk}
- 3 - Si calcola la risposta y_k e si aggiornano i pesi attraverso la delta rule
- 4 - Si ripete il ciclo dal passo 2 finché non si ottiene una risposta soddisfacente

Con riferimento alla figura 2.5, si osservi che modificando i pesi w_1, w_2, w_0 si modifica la posizione e la pendenza della retta di separazione tra le due classi. Il processo termina quando la retta separa correttamente le due classi.

Capitolo 3

Ricorrenti: Reti di Hopfield

Bibliografia

- [1] Prof.ssa Lazzarini Beatrice, (2015), *Introduzione alle reti neurali*.
- [2] Prof. Gambosi Giorgio, (2010), *Reti neurali, note dal corso di Machine Learning*.
- [3] Prof.ssa Labonia Laura, *Storia delle reti neurali artificiali*.
- [4] Prof. Bicego Manuele, *Riconoscimento e recupero dell'informazione per bioinformatica, reti neurali*.
- [5] Ing. Pioggia Giovanni, (2009), *Modelli di sistemi fisiologici*.