# Lezione 2
# Introduzione alle Reti Neurali

Vito Roberto
Dipartimento di Matematica, Informatica e Fisica (DMIF)
Università di Udine
E-mail: vito.roberto@uniud.it

## 1.-Reti Neurali

Le prestazioni cui si è accennato nella lezione precedente non sarebbero possibili se non si facesse riferimento all'hardware biologico: la struttura e alcune funzioni del cervello. Faremo un accenno al sistema visivo umano, dall'occhio alla corteccia cerebrale. Le analogie con i sistemi biologici sono un'altra affascinante caratteristica dell'IA. A partire dalla seconda metà del secolo scorso gli scienziati hanno studiato modelli matematici e realizzato programmi che al computer simulano la struttura del cervello, sebbene in modo molto semplificato rispetto al reale.

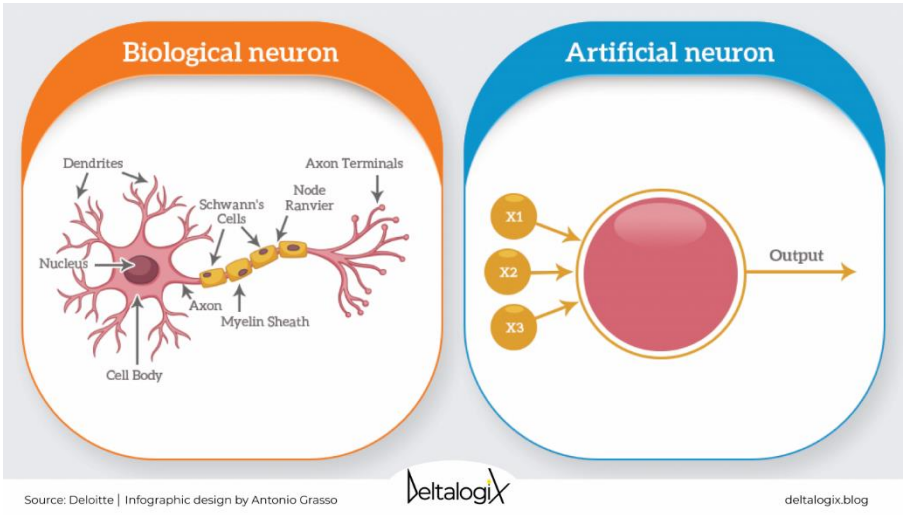


Figura 1.

Nella Figura 1 a sinistra riportiamo lo schema di un neurone biologico con alcune sue parti; nel cervello di un umano adulto vi sono circa 85 miliardi di neuroni, ciascuno dei quali connesso a migliaia di altri tramite gli *assoni.* A destra lo schema di un neurone artificiale: tre segnali di ingresso X1, X2, X3 – eventualmente provenienti da altri neuroni – entrano in un *nodo,* che è un'unità di calcolo in cui i segnali sono combinati e trasmessi in uscita *(output).* Possono essere inviati ad altri nodi tramite le *connessioni,* che sono l'analogo

degli assoni biologici. Lo schema del neurone artificiale dà luogo a configurazioni più complesse che servono a progettare programmi di calcolo, le *reti neurali artificiali (Artificial Neural Networks, ANN).*
Le reti neurali sono state studiate per realizzare l'apprendimento da parte dei computer. Dai primi anni duemila si sono verificati progressi importanti: nuovi modelli matematici delle ANN che danno origine ad architetture complesse, che affronteremo nel seguito.
L'apprendimento è un processo fondamentale anche nella visione umana. Gli occhi catturano sequenze di immagini dell'ambiente e, attraverso il nervo ottico ed altri canali biologici, raggiungono la corteccia cerebrale posteriore in un'area denominata V1. Alcune delle fasi di elaborazione della corteccia visiva del cervello sono state esplorate dai fisiologi D.H. Hubel ed T. Wiesel, premi Nobel 1981. Hanno scoperto che nell'area V1 - costituita da neuroni e assoni come l'intero sistema cerebrale – alcuni neuroni rispondono a stimoli di forma particolare. Ad esempio, reagiscono mostrando attività elettrica maggiore in risposta a stimoli esterni di luce disposti in direzione verticale. Siccome i neuroni sono intercomunicanti, si creano *aree sensibili specializzate (visual maps)* nella corteccia visiva, che trasmettono l'informazione ad altre aree simili, sensibili ad altri stimoli. In tal modo, neuroni di zone più profonde – delle quali sappiamo ancora poco – probabilmente ricombinano i dettagli della sequenza visiva come un puzzle; la registrano in memoria in modo che in tempi successivi la si possa recuperare *(retrieving)*.
Ci proponiamo di approfondire nel seguito le reti neurali, dandone una presentazione formale e impostando uno studio progettuale e realizzativo con riferimento alla Visione artificiale.

**2. - Modelli neurali: terminologia**
*Un modello (rete) neurale – Rete Neurale (RN), Neural Network (NN) - è un insieme di elementi chiamati neuroni o nodi (neurons, nodes), tra loro connessi da legami (links, arcs, connections).*
Una NN può essere *biologica*, costituita dai neuroni e dagli altri componenti biologici illustrati in Figura 1; oppure *artificiale,* progettata e realizzata con procedure software per risolvere problemi di Intelligenza Artificiale (AI), in analogia con la struttura e le funzioni del cervello umano.
Le reti neurali artificiali (ANN) sono *modelli di calcolo*, cioè schemi astratti che permettono di definire le *architetture di calcolo:* le quali, a loro volta, permettono di *progettare* gli elementi hardware e/o software per realizzare le computazioni stesse.
La caratteristica fondamentale è che una rete neurale può essere addestrata ad *apprendere dai dati* per raggiungere obiettivi e risolvere problemi.
Un modello richiede che i nodi siano identificati *(etichettati, labelled)* con numeri reali detti *valori di attivazione (activation values.* Anche le connessioni devono essere etichettate con *fattori peso (weight factors);* in particolare, un peso positivo rappresenta una *connessione eccitatoria;* un peso negativo una *connessione inibitoria.*
Sia le attivazioni che i pesi possono variare nel tempo.
I modelli ANN sono indicati anche con altri nomi, a seconda degli aspetti da evidenziare: *Modelli Connessionisti (CM); Processi Paralleli Distribuiti (PDP).*

Per specificare i modelli e le architetture si usano *rappresentazioni di tipo grafo, costituite da nodi e connessioni etichettate (labelled).* Alcuni modelli richiedono di organizzare nodi

e connessioni in sottoinsiemi. Esempi sono gli *strati (layer)*, che a loro volta possono essere organizzati geometricamente in *lineari, planari, volumetrici;* oltre a queste, molte altre configurazioni spaziali sono state proposte. La Figura 2 riporta un modello neurale strutturato in tre strati lineari.

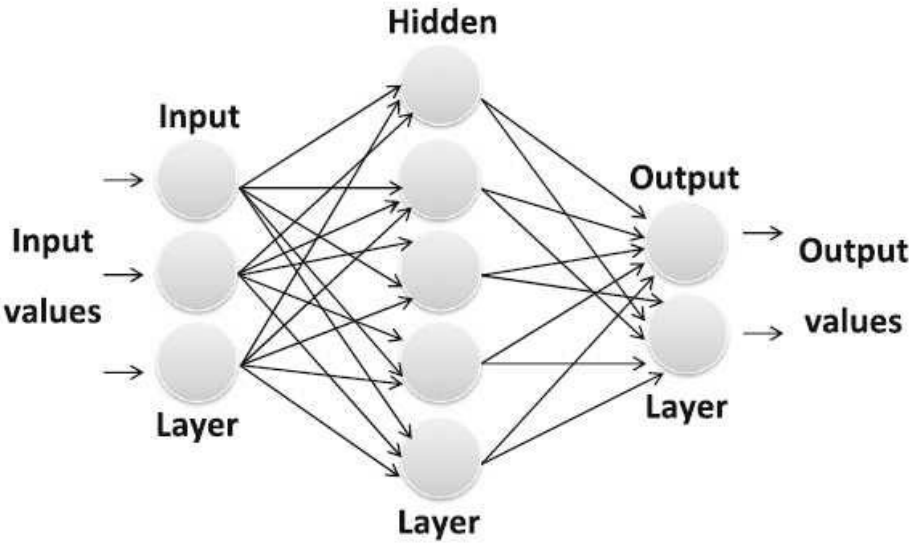


Figura 2. Esempio di modello connessionista in cui i nodi sono disposti in tre strati: strato di input; nascosto (hidden); di output.

L'organizzazione in strati permette di semplificare il modello per meglio studiarne le proprietà e progettarne il funzionamento. Uno strato *raggruppa i nodi in sottoinsiemi*: ciò introduce un *ordinamento spaziale*, che permette di indicizzare i nodi in modo più semplice. L'ordine spaziale può consentire anche *un ordine temporale:* è possibile *sincronizzare* il funzionamento dei nodi di uno stesso strato, ad esempio prevedendo che si aggiornino i rispettivi parametri nello stesso istante. In definitiva, l'organizzazione spazio-temporale dei nodi agevola la progettazione del *flusso dei dati (data flow)* nella rete, cioè la dinamica della computazione.
Supponiamo di indicizzare gli strati in Figura 1 da sinistra a destra in ordine crescente. Si dice che la rete opera in *modalità in avanti (feedforward mode)* quando il flusso dei dati segue l'ordinamento dallo strato di input a quello di output. Il flusso dei dati può seguire la direzione opposta: si dice che in tal caso la rete opera in modalità *propagazione all'indietro (backpropagation mode).* Naturalmente è possibile progettare flussi che non hanno un'unica direzione, ma possono alternare fasi di propagazione in avanti *(feedforward)* e all'indietro *(feedback).*

**3.- Tipi di modelli a strati**

(a) Riportiamo in Figura 3 un *modello monostrato (single-layer model).* I nodi di input sono indicati con simboli diversi (quadratini) rispetto alla Fig.2.

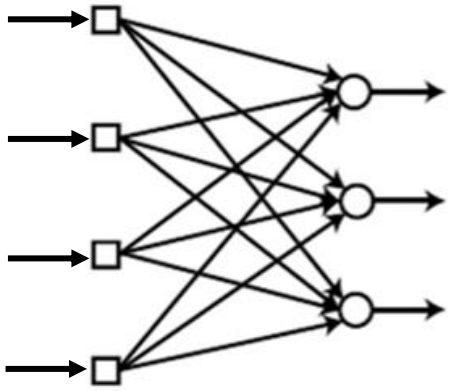

Figura 3. Un modello di rete monostrato.

(b) In Figura 4 riportiamo un *modello multistrato con un solo strato nascosto,* talvolta chiamato rete *superficiale (shallow NN)* o *vanilla network.* I nodi di input sono indicati come in Fig. 3.

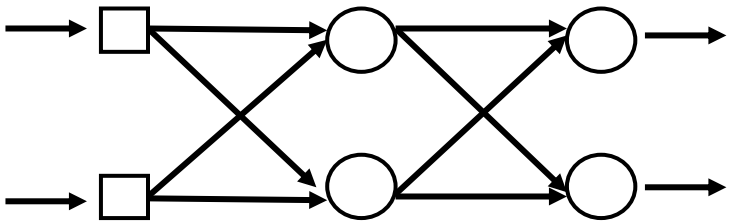

Figura 4. Una rete neurale superficiale.

(c) In Figura 5 riportiamo una rete con due strati nascosti, esempio di *modello profondo (deep neural network).*

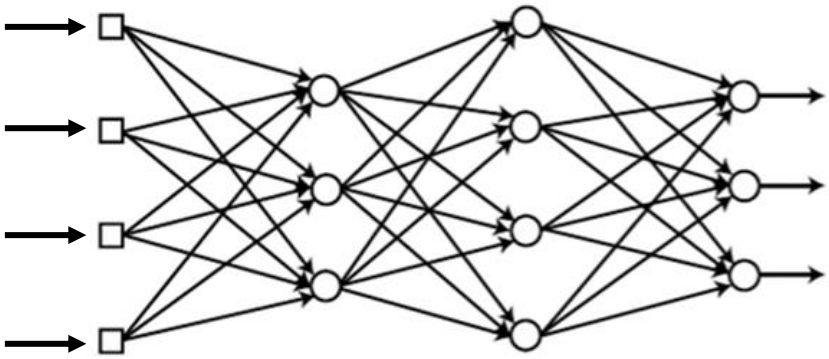

Figura 5. Modello di *deep neural network.*

## 4.- Cenni ai domini di applicazione

E'opportuna una osservazione. Un modello neurale mette in corrispondenza un vettore di dati di input – ad esempio nel caso in Fig.5 l'input è a quattro componenti - con un vettore di output – nell'esempio, tre componenti. Si tratta di *uno schema di funzione*, che mappa un m-pla di dati di input in una t-pla di output. Dunque, vi sono casi in cui non è possibile specificare la forma di una funzione né in termini analitici, né come tabella di corrispondenza tra variabili indipendenti e dipendenti: in questi casi sono utili i modelli neurali, perché con essi *la funzione può essere appresa* tramite una procedura automatica. Indicheremo due aree applicative delle reti neurali che sono tra le più importanti, pur tenendo conto che ve ne sono altre più specifiche.

*(a) Classificazione e riconoscimento di forme (pattern classification, recognition).*
In molte situazioni reali i dati di input sono in forma di segnali audio; immagini; sequenze video. Spesso si tratta di un grande ammontare di informazione, caratterizzato da variabilità, individualità, ambiguità, rumore di varia origine. Esempi fra i tanti sono la scrittura manuale o la firma di una persona su un documento; i messaggi vocali; le caratteristiche visive dei volti di persone. Si tratta di dati complessi, per i quali spesso si richiede di effettuare azioni di riconoscimento automatico: assegnare un'identità a una firma; comprendere un comando vocale e compiere azioni conseguenti; riconoscere il volto di una persona associandolo a dati anagrafici;….

*(b) Approssimazione di funzioni,* anche chiamata *analisi di regressione (regression analysis).* E'una disciplina di analisi dei dati quando questi consistono in una variabile dipendente e una o più variabili indipendenti. Lo scopo è stimare un'eventuale relazione funzionale tra di esse. Un caso molto comune è la regressione lineare, in cui si devono stimare i parametri di una funzione lineare dei dati d'ingresso: geometricamente i parametri della retta che meglio approssima l'andamento dei dati.

## 5.- Dai modelli alle architetture

Un'architettura di calcolo specifica nei dettagli le caratteristiche matematiche del modello per poi realizzarlo al computer. A titolo di esempio presentiamo l'architettura relativa al modello in Figura 2, che ora specifichiamo in Figura 6.
Presenteremo prima gli elementi e le loro notazioni, poi al successivo par.6 le operazioni previste dal modello in ciascun nodo.

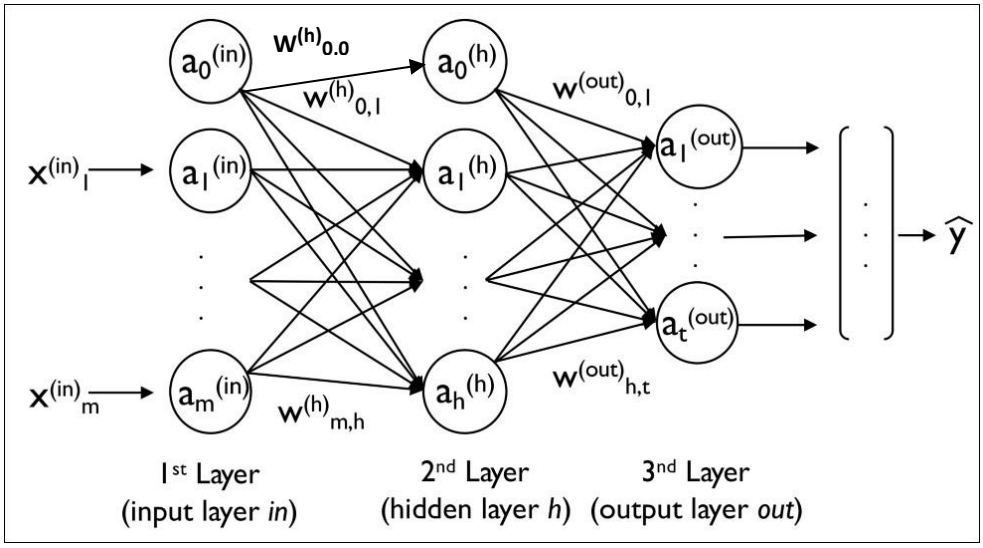


Figura 6. Architettura neurale con i nodi disposti in tre strati.

Gli elementi di un'architettura sono rappresentati matematicamente da numeri, vettori, matrici. Con riferimento alla Figura 6 gli elementi sono:

$\mathbf{x^{(in)}} = (x^{(in)}_1, x^{(in)}_2,\ldots,x^{(in)}_m)^T$
Vettore m-dimensionale degli input, ciascuno dei quali dipendente dal tempo. Rappresenta *gli stimoli provenienti dall'ambiente esterno,* che inizializzano la computazione. L'apice T, simbolo di trasposizione, indica che $\mathbf{x^{(in)}}$ è un vettore-colonna; le sue componenti non sono considerate nodi della rete e non svolgono elaborazioni;

$\mathbf{a^{(in)}} = (a_0^{(in)}, a_1^{(in)},\ldots..,a_m^{(in)})^T$
Vettore (m+1)-dimensionale dei valori di attivazione dei nodi dello strato di input. Sono identificati con i valori del vettore $\mathbf{x^{(in)}}$, a cui è aggiunto un nodo dal valore $a_0^{(in)}$. Sono talvolta compresi negli intervalli [0,1], oppure [-1,1];

$\mathbf{a^{(h)}} = (a_0^{(h)}, a_1^{(h)},\ldots\ldots..,a_h^{(h)})^T$
Vettore (h+1)-dimensionale dei valori dei nodi di uno strato nascosto. Sono in genere *funzioni del tempo;* la variabile tempo non è indicata esplicitamente per non appesantire la notazione;

$\mathbf{a^{(out)}} = (a_1^{(out)}, ..., a_t^{(out)})^T$
Vettore t-dimensionale dei valori dei nodi di output;

$\mathbf{y} = (y_1,\ldots y_t)^T$
Vettore t-dimensionale degli output, che si leggono direttamente dalle attivazioni dello strato di output (vettore $\mathbf{a^{(out)}}$). Rappresenta le risposte dell'intera rete agli stimoli di input, cioè i risultati che ci si attende dall'elaborazione;

**W = { $w_{ij}$ }** i = 0,…,m nodi di uno strato; j = 0,…,h nodi dello strato successivo. Matrice dei pesi delle connessioni; sono numeri reali sia positivi che negativi. Sono dipendenti dal tempo quando la rete opera in *modalità di apprendimento (training mode)*; sono costanti quando la rete opera altrimenti.

**6.- Architetture a strati: l'input ai nodi**

Specifichiamo ora l'architettura in termini di operazioni aritmetiche ai nodi; in questo modo definiremo matematicamente il flusso delle informazioni. Nella presentazione seguiremo l'ordinamento dall'input all'output, ipotizzando che la rete operi in modalità in avanti *(feedforward).*

*Calcolo degli input a un nodo.* Partendo dallo strato di input (Figure 2, 6), consideriamo le coppie di nodi appartenenti a due strati consecutivi tra cui vi sia un legame. Ad esempio, la coppia ($a_0^{(in)}$, $a_0^{(h)}$) del nodo 0 di input e 0 dello strato nascosto h**,** è connessa da un legame etichettato $w_{00}^{(h)}$

Il contributo all'attivazione al nodo 0 dello strato h, dovuto al nodo 0 di input, si calcola semplicemente con un prodotto e una somma tra numeri reali

$$a_0^{(h)} = w_{00}^{(h)} a_0^{(in)} + b_0 \qquad (1)$$

Nella (1) si è introdotto il numero reale costante $b_0$ detto 'bias' per tenere conto dei valori di fondo suggeriti dal contesto in cui opera la rete.

Il significato della (1) è che *il calcolo aggiorna, modifica (update, adjust) il valore di attivazione del nodo 0 dello strato h, per effetto dello stimolo (segnale) che a un certo istante proviene dal nodo 0 di input lungo la connessione che li collega.*

A sua volta, la (1) significa che il nodo 0 di input *influenza il nodo a cui è collegato*, cioè il nodo 0 dello strato nascosto h: *quanto lo influenzi* è determinato dal peso della connessione. Ad esempio, se nella (1) il peso $w_{00}^{(h)} = 0$, allora non vi è influenza del nodo sull'altro. In tal caso è il valore di bias $b_0$ a determinare il valore di attivazione del nodo nascosto, anche in assenza di contributo da quello di input ad esso collegato, cioè senza essere attivato: esprime quindi l'attività di fondo del nodo; se negativo, esprime il valore di soglia che l'input deve superare per poterne modificare l'attivazione.

Se nella (1) il peso $w_{00}^{(h)} > 0$, il nodo di input contribuisce positivamente all'attivazione di quello collegato *(nodo eccitatorio)* – cioè possibilmente, tenendo conto del bias - ne aumenta il valore di attivazione. La sua influenza è tanto maggiore rispetto agli altri nodi, quanto più alto è il peso $w_{00}^{(h)}$

Se nella (1) il peso $w_{00}^{(h)} < 0$, il nodo di partenza è *inibitorio:* influenza negativamente l'attivazione del nodo collegato, cioè possibilmente ne riduce il valore di attivazione.

A partire dalla (1) si calcola il contributo complessivo all'attivazione $a_0^{(h)}$, sommando algebricamente i contributi da parte dei singoli nodi dello strato di input che sono connessi collegati ad $a_0^{(h)}$ stesso.

$$a_0^{(h)} = \sum_i w_{i0}^{(h)} a_i^{(in)} + b_0 \qquad \text{i=0,…,m nodi di input}$$

Dunque, *la computazione aggiorna il valore di attivazione di un nodo, calcolando la combinazione lineare dei valori di attivazione dei nodi collegati in input, avendo come coefficienti i fattori peso corrispondenti a ciascun legame.*

Possiamo generalizzare il calcolo dei segnali d'ingresso a tutti i nodi dello strato nascosto h, a partire da tutti i nodi dello strato di input reciprocamente collegati.

$$a_j^{(h)} = \sum_i w_{ij}^{(h)} a_i^{(in)} + b_j \qquad (2)$$

…dove: i = 0,…,m nodi dello strato di input ; j = 0,…,h nodi dello strato nascosto
La (2) ha la forma di un prodotto matrice-vettore secondo l'algebra lineare; usando la notazione vettoriale per componenti si scrive:

$$\begin{pmatrix} a_0^{(h)} \\ a_1^{(h)} \\ \vdots \\ a_h^{(h)} \end{pmatrix} = \begin{pmatrix} w_{00}^{(h)} & w_{01}^{(h)} & \dots & w_{0m}^{(h)} \\ w_{10}^{(h)} & w_{11}^{(h)} & \dots & w_{1m}^{(h)} \\ \vdots & \vdots & \ddots & \\ w_{h0}^{(h)} & w_{h1}^{(h)} & \dots & w_{hm}^{(h)} \end{pmatrix} \begin{pmatrix} a_0^{(in)} \\ a_1^{(in)} \\ \vdots \\ a_h^{(in)} \end{pmatrix} + \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_h \end{pmatrix}$$

La matrice $\mathbf{w} = \{ w_{ij} \}$ denota in modo sintetico i valori dei pesi delle connessioni: il primo indice (i, di riga) indica la posizione del nodo di partenza appartenente allo strato di input; l'indice (j) di colonna indica la posizione del nodo di arrivo nello strato nascosto. Il prodotto matrice-vettore è del tipo righe per colonne. Il vettore-colonna $\mathbf{a_0^{(in)}}$ è quello delle attivazioni dei nodi di input; le righe della matrice contengono i pesi delle connessioni che collegano tutti i nodi dello strato di input con un singolo nodo dello strato successivo. Scrivendo in forma di operazioni tra matrici :

$$\mathbf{a^{(h)}} = \mathbf{w^{(h)} a^{(in)}} + \mathbf{b} \qquad (3)$$

Ricordiamo che i vettori di attivazione $\mathbf{a_0^{(h)}}$, $\mathbf{a_0^{(in)}}$ sono funzioni del tempo.
Nel seguito semplificheremo la notazione sopprimendo gli apici (h), (in) che indicano gli strati di appartenenza dei nodi. Dovremo comunque specificare l'architettura strato per strato, tenendo conto delle scelte particolari di ciascun progetto.

**7.- Architetture a strati: l'output dai nodi**

Ogni nodo nascosto riceve un input come specificato al paragrafo precedente, ma non si limita a ricevere e trasmettere, perchè *calcola* un segnale in uscita (output). Una *funzione di attivazione (activation function)* modifica (modula) il valore di attivazione appena aggiornato, che è la variabile **z** della funzione, in modo che questo rientri tra i limiti di ampiezza prefissati – ad es., ricada nell'intervallo [0, 1], oppure [0, 10].

Riportiamo due esempi molto comuni di funzioni di attivazione, di cui diamo i rispettivi diagrammi nella Figura 7.

(a) Sigmoide

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

(b) ReLU (Rectified Linear Unit)

$$R(z) = \max(0, z)$$

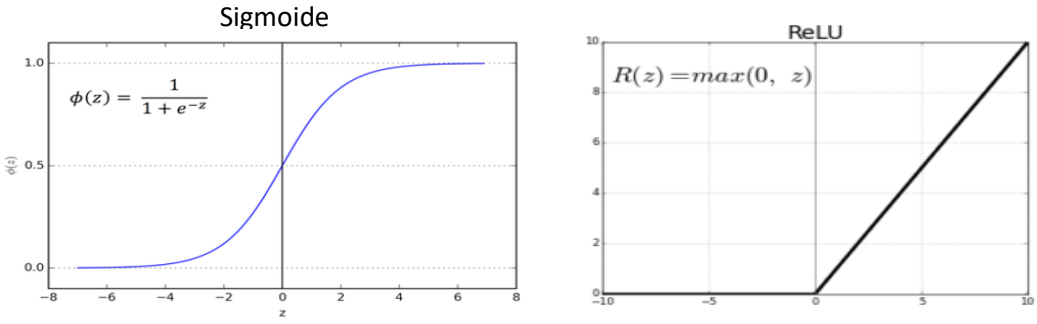


Figura 7. Diagrammi delle funzioni Sigmoide e ReLU.

Ricordiamo che la variabile **z** è il valore aggiornato delle attivazioni dei nodi di uno strato, calcolata secondo la (3). In forma semplificata:

$$\mathbf{z} = \mathbf{w}\,\mathbf{a} + \mathbf{b}$$

**8.- Conclusioni**

Dalla presentazione di un modello neurale e specifiche architetture ricaviamo alcune considerazioni. Visto come schema di una macchina di calcolo, il modello neurale non prevede la separazione della memoria e dell'elaborazione (processing) in unità distinte, a differenza di ciò che accade nel modello di macchina di Von Neumann. Per questo si parla anche di *architetture non-Von Neumann*. La computazione non è concentrata in singole unità, ciascuna con le proprie funzioni, ma distribuita tra i nodi, che agiscono sia come unità di memoria che come processori di informazione, ciascuno indipendente dagli altri. I legami tra i nodi operano come trasmettitori di informazione: non linee di trasmissione passive ma attive (sinapsi, *synapses)* in grado di modificare (modulare) l'ampiezza del segnale lungo la linea stessa.

Il flusso dei dati non segue un unico percorso ma cammini molteplici e distinti. Ciò giustifica la denominazione di *Processi Paralleli Distribuiti (PDP models)* data ai modelli neurali. Quanto alla memoria, è anch'essa distribuita nell'intera architettura. Parte di essa

consiste nei pesi assegnati ai legami, in analogia con la memoria a lungo termine dei sistemi biologici *(long-term memory).* Altre funzioni di memoria sono codificate nei valori di attivazione dei nodi: una memoria dipendente dal tempo, dinamica, che ha analogie con la memoria biologica a breve termine *(short-term memory).*
Dunque, la macchina neurale svolge le funzioni di elaborazione e memoria *distribuite su un insieme di nodi collegati*, i quali nel tempo svolgono ruoli molteplici. Le funzioni che svolge la macchina neurale *emergono da un comportamento collettivo e coordinato di elementi.*
Osserviamo infine che il modello neurale, per svolgere efficacemente le proprie funzioni dovrebbe essere realizzato su un supporto hardware coerente, cioè parallelo e distribuito anche questo. Data la ormai universale diffusione delle macchine di calcolo sequenziali, il modello neurale – e le architetture che ne derivano – generalmente sono *simulate su una macchina sequenziale,* come un PC da scrivania, un tablet, uno smartphone.

Nella prossima lezione ci occuperemo dell'assegnazione dei valori dei pesi ai legami della rete: come si è accennato al par.3, l'assegnazione può essere automatizzata tramite meccanismi di *apprendimento automatico*.