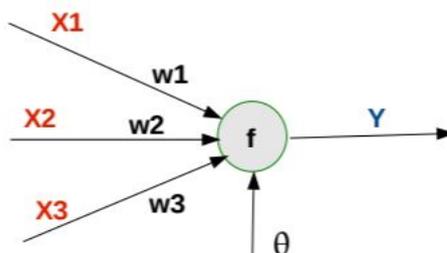


Percettrone:

Si tratta di una unità computazionale, ispirato alla struttura dei neuroni, a N ingressi (ognuno dotato di un certo peso) con una sola uscita Y. Il valore di y è calcolato da una funzione che prende in esame i valori di ingresso.



Nel dataset abbiamo vari dati (istanze) descritti da features (X1...Xn) associate ad un particolare valore Y (la categoria di quella istanza).

Il neurone elabora e produce un risultato che confronta con la Y associata a quel dato e usa la differenza, cioè l'errore, come segnale di apprendimento.

La funzione interna al neurone è sempre la stessa, quello che cambia (in base al dataset sul quale il neurone apprende) è il peso che viene dato ai singoli input.

I neuroni hanno due modalità di calcolo:

Forward run: il neurone produce il suo output (Y)

Backward run: in fase di apprendimento, cioè modifica i pesi (W)

Si rielabora il learning set più volte, anche centinaia, ogni passata prende il nome di epoca di apprendimento. Si può rappresentare il grado di apprendimento su un grafico in cui sulle ascisse c'è l'indice dell'epoca e sulle ordinate c'è l'errore medio. Al crescere delle epoche diminuirà il valore dell'errore. Il guadagno cresce epoca dopo epoca, quando la crescita inizia a essere minima si può interrompere l'apprendimento.

I nostri veri neuroni, si attivano quando l'input che ricevono superano una certa soglia, interna al neurone stesso.

Funzione di attivazione f(net): Spesso si usa la sigmoide. Restituisce valori compresi tra 0 e 1. α è un parametro passato, più è alto e più la funzione assumerà una forma a gradino. Essendo continua e derivabile.

$$net = \sum_{i=1}^n w_i X_i$$

Combinazione lineare

$$Y = f(net) = \frac{1}{1 + e^{-\alpha*(net - \theta)}}$$

Funzione a gradino, spesso una sigmoide

θ = soglia o bias

Un neurone istruito si attiverà (restituirà 1) solo quando incontrerà un'istanza di classe. Immaginiamo di porre le varie istanze da classificare (discriminare) su un piano cartesiano.

La funzione di attivazione sarà rappresentata da una retta che divide gli elementi (i valori appartenenti ad una certa categoria saranno sopra questa retta, gli altri al di sotto). I pesi corrispondono ai parametri di posizionamento di questa retta.

Inizialmente i pesi sono inseriti casualmente e la retta fa molti errori. Ad ogni errore il vettore dei pesi si muove nello spazio cartesiano verso lo spazio popolato dalle soluzioni, ovvero con la retropropagazione si va a modificare il peso associato a quel punto della retta per farne alzare o meno la sua pendenza, così da inglobare quel determinato punto (o viceversa).

Imparare = individuare la posizione corretta dell'iperpiano nello spazio

Algoritmo di apprendimento dei perceptron:

Se il vettore sbaglia la classificazione dando 0 al posto di 1, noi sommiamo al vettore dei pesi il vettore di input.

Viceversa se il neurone dà 1 al posto di 0, sottraiamo al vettore dei pesi il vettore di input.

L'input viene convertito in un vettore di feature, in cui ognuna descrive una determinata caratteristica dell' input.

Quando la somma pesata di queste feature (a cui viene aggiunto un ulteriore addendo, denominato Bias) supera una certa soglia, il neurone si attiva e classifica l'input come appartenente ad una certa categoria.

Limite: Sviluppato nel 1960 da Frank Rosenblatt venne stroncato nel 1969 da Minsky e Papert i quali ne esposero le principali limitazioni, spegnendo tutto l' interesse che questo strumento di calcolo aveva suscitato molto interesse fino ad allora.

Se il problema linearmente separabile (le istanze sono divisibili in classi) la retta converge, cioè troverà dei parametri che le permetteranno di distinguere tutti i vari punti nell' iperspazio.

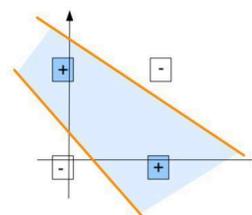
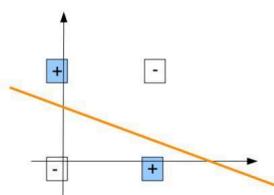
Osservazione: il perceptron definisce su un punto sta sopra o sotto la retta e basta, non ne prende in considerazione la distanza.

Critica ai neuroni: utilizzano rette che sono troppo limitate → Soluzioni: usare più rette per perfezionare la classificazione. Non si prende tutto ciò che è sopra o sotto una retta, ma ciò che è compreso tra rette, in questo modo posso anche catturare forme chiuse. **Rete neurale**

I limiti del perceptron furono evidenziati tramite un esempio semplicissimo: l'or esclusivo (XOR)

A1	A2	XOR
0	0	0
1	0	1
0	1	1
1	1	0

Valori degli attributi Output desiderato



Un'iperpiano da solo non ce la fa ma se ne avessi a disposizione 2 e avessi la capacità di combinare le loro risposte?

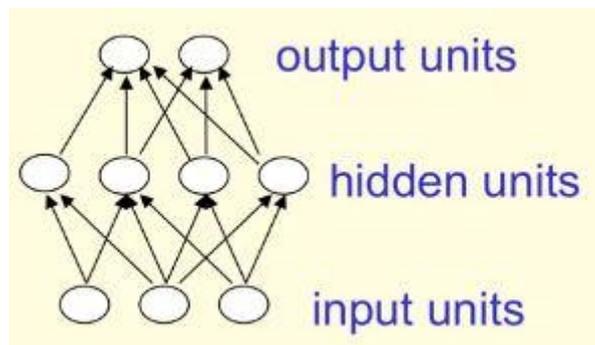


Il vero neurone però ha due tipi di memorie:

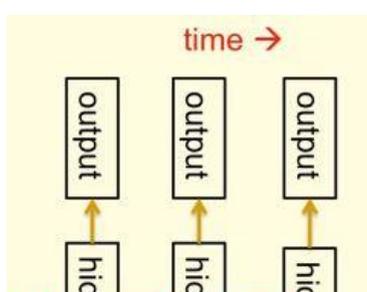
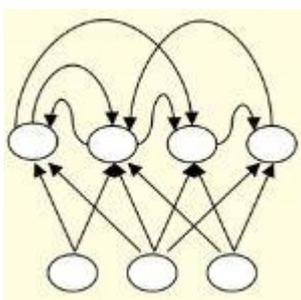
Memoria elettrica: a breve termine, si perde se si prende una forte scarica elettrica che va a corrompere proprio questa memoria

Memoria chimica: modifica proprio il funzionamento del neurone e mantiene la memoria a lungo termine (come i pesi del perceptron). Quindi come possiamo intuire i ricordi vengono continuamente affinati e si modificano nel tempo.

Feed-Forward neural network: composta da unità di input (nello strato inferiore) ed unità di output (nello strato superiore), connesse da unità nascoste negli strati centrali. Ovviamente, essendo reti "deep", possono esserci molteplici strati hidden.



Recurrent Neural Network: Sono tipi di reti più potenti delle Feed Forward. Hanno cicli nel loro grafo di connessione, cioè partendo da un nodo e attraversando gli archi del grafo è possibile incontrare lo stesso nodo di partenza. Sono difficili da addestrare ma sono molto potenti e vicine ai modelli reali presenti in biologia. A differenza delle reti "feed forward", in queste reti i neuroni hidden sono connessi tra loro.



Queste reti dividono il tempo in step, in ogni step la rete viene vista come una rete con un singolo neurone hidden, il quale calcola il valore dello stato di quello per lo step temporale successivo.

Due differenze fondamentali con le reti Feed Forward sono:

- Queste reti utilizzano gli stessi pesi per ogni step
- Ad ogni step vanno a rileggere gli input (potrebbero anche fornire un output a ogni fine step)

Esempio interessante di utilizzo: Llya Sutskever ha addestrato una rete neurale ricorrente attraverso le pagine inglesi di wikipedia, con lo scopo di insegnarle a predire la successiva lettera.

Questo può essere visto come un modo di generare testo in quanto basta iterare di continuo la scelta della lettera successiva fino a comporre intere frasi (e i risultati ottenuti sono stati molto positivi).

Symmetrically Connected Networks: un tipo di rete in cui i collegamenti tra neuroni hanno lo stesso peso in entrambe le direzioni (Reti di Hopfield, le vedremo nel dettaglio più avanti).

Thanks to: [Geoffrey Hinton \[Coursera\]](#)