

# Dalla biologia all'intelligenza artificiale: le reti neurali

Pubblicato da Gabriele Pucciarelli il giorno 17 febbraio 2012

Può una macchina pensare come un uomo? Ad oggi, la risposta sembrerebbe essere “no, non può”. **Ma non è così**: il cervello umano non può ancora essere riprodotto in toto, in tutti i suoi campi, ma esistono software che possono simulare il pensiero umano (e non) in alcune delle sue funzioni e campi di applicazione; un esempio? Molti di voi conosceranno **Siri**, famosissima app per iPhone di riconoscimento vocale (ve ne abbiamo parlato [qui](#)). Vi siete mai chiesti cosa c’è dietro applicazioni come Siri? Oggi parleremo proprio di questo, parleremo di **reti neurali**.

## Che cos’è una rete neurale

---

Qual’è la differenza tra un cervello, ad esempio quello umano, e il software che fa girare il vostro videogioco preferito?

Più volte nel corso della giornata voi stessi venite bombardati da segnali (visivi, uditivi, tattili etc.) nuovi, di cui non avete assolutamente esperienza: il vostro cervello è strutturato proprio per far fronte a queste occasioni. Il cervello umano, sebbene si **rinforzi** sulla base dell’esperienza vissuta, di fronte ad un nuovo stimolo funziona per **analogia**. Su un cartellone pubblicitario vedete una tonalità di blu che probabilmente non avete mai visto, ma il vostro cervello sa già dirvi che quello è blu e non rosso. Sentite il cinguettare di un uccello: non l’avete mai visto, non avete mai sentito quel canto, ma basta questo al vostro cervello per sapere che, tra le migliaia di segnali uditivi che vi giungono all’orecchio, quello è **probabilmente** il canto di un -qualche- uccello.

Cosa succederebbe se invece riusciste a trovare un bel bug nel vostro videogioco preferito nella quale, ad esempio, il vostro personaggio si spostasse da uno scenario ad un altro solo e soltanto cadendo da un dirupo? Molto probabilmente il gioco andrebbe in crash, perché i programmati non hanno dato al software **l’esperienza necessaria per gestire quell’evento** (oppure potreste usare quel bug a vostro favore! 😊). E’ un errore, e il software non sa cosa fare di fronte ad esso.

La differenza principale tra una rete neurale (biologica, come il cervello, o artificiale) ed un normale software è proprio questa: sapere gestire situazioni, eventi e dati **sulla base** dell’esperienza passata anche senza essere stato a conoscenza, fino a quel momento, dell’eventualità di quel segnale. In definitiva, la vera **forza** di una rete neurale sta

nel **gestire** situazioni impreviste, **impararle** e -magari-**prevederle**. Tutto questo grazie, principalmente, alla **struttura fisica**, all'impalcatura della rete neurale.

Il **cervello animale** è formato principalmente da neuroni, collegati tra loro a formare una fittissima rete. Un neurone è quindi una cellula collegata ad uno o più neuroni dalla quale **prende** informazioni, e collegata ad altri neuroni alla quale **passa** informazioni (sotto forma di segnale elettrico) tramite sinapsi. La prima idea di rete neurale artificiale deriva proprio da questo concetto: **simulare la struttura biologica di un cervello animale**.

Una rete neurale è quindi una struttura composta principalmente da neuroni artificiali, suddivisi in strati (uno strato di **input**, uno o più strati **nascosti** ed uno strato di **output**) e collegati tra loro in modo che ogni neurone dello strato precedente sia collegato a tutti i neuroni dello strato successivo.

Nella concezione più classica e più semplice delle reti neurali artificiali, qualsiasi neurone della rete -diciamo così- “contiene” un valore, principalmente numerico. Anche ogni sinapsi (quelle che nell’immagine qui sopra sono raffigurate dalle frecce che collegano un neurone a quelli successivi) “contiene” un numero, che chiameremo **peso**.

Quindi ci saranno tanti neuroni di input quanti sono i valori che vorremmo dare in pasto alla nostra rete, un numero impreciso di strati e neuroni nascosti e tanti neuroni di output quanti sono i valori che alla fine vorremo ottenere. Ad esempio, se vogliamo una rete neurale che, dati tre numeri  $a, b$  e  $c$  esegua la funzione  $x=(a+b)*c$  creeremo una rete con 3 neuroni di input e 1 di output.

## Come funziona una rete neurale

---

Creata l’impalcatura della nostra rete, il suo funzionamento è semplice. In natura, ad un neurone arriva uno stimolo dai neuroni precedenti. Il segnale viene elaborato e, se supera una certa soglia, viene spedito ai neuroni successivi, che a loro volta lo elaborano, e così via.

Nella nostra rete artificiale siamo noi stessi che diamo dei valori ai neuroni di input. Ricordate che ogni neurone di input è collegato, tramite un **peso**, al neurone dello strato nascosto? Bene, il **valore del primo** neurone input viene moltiplicato per il **peso** che lo collega al **primo** neurone nascosto, e il prodotto di questa moltiplicazione ( $\text{input1} \times \text{peso1}$ ) diventa il nuovo valore del primo neurone nascosto.

Adesso tocca al **secondo** neurone di input: il suo **valore** viene moltiplicato per il **peso** che lo collega al **primo** neurone nascosto. Il prodotto di questa operazione ( $\text{input2} \times \text{peso2}$ ) viene sommato a quello che era già stato inserito nel primo neurone nascosto, ossia il prodotto della prima moltiplicazione ( $\text{input1} \times \text{peso1}$ ).

E così via per ogni neurone di input. Molto più semplice spiegarlo con un’immagine che spiegarlo a parole

In definitiva, il valore di ogni neurone nascosto è rappresentato dalla sommatoria dei valori dei neuroni di input moltiplicato il rispettivo peso.

Allo stesso modo, il valore di ogni neurone di output è rappresentato dalla sommatoria dei valori dei neuroni nascosti moltiplicato il rispettivo peso.

Ad onor del vero, la funzione “moltiplicazione” è solo la più semplice utilizzata. In alcuni casi, parlando di reti neurali più complesse, il valore di un neurone non viene semplicemente moltiplicato il proprio peso, ma vengono utilizzate funzioni più complesse.

E allora -direte voi- dov’è il famoso **apprendimento** della rete neurale? Quello di cui vi abbiamo parlato fin’ora è il funzionamento di una rete neurale già funzionante, visto che abbiamo dato per scontato che i **pesi** erano già stati determinati. La vera forza di una rete neurale sta proprio nei pesi, e nel saperli automaticamente modificare a seconda dell’output desiderato: i valori di input infatti sono decisi dall’utente, i valori di output devo essere dati dalla macchina, i valori dei neuroni nascosti sono soltanto il risultato di semplici sommatorie. Cosa rimane di variabile all’interno della rete? I **pesi delle sinapsi**! Essi sono l’elemento chiave, veramente modificabile dalla rete stessa. E sono essi che la rete deve impostare per coprire ogni possibilità richiesta dall’utente.

## La rete neurale impara!

---

In origine tutti i valori e tutti i pesi sono 0 (zero). La rete neurale è vuota, e ovviamente inutilizzabile.

Dobbiamo allora cominciare a dare delle informazioni alla nostra rete neonata: informazioni che formeranno poi la vera e propria **esperienza** del nostro “cervello” artificiale. Si parte dunque da esempi (o *pattern*) conosciuti, situazioni di cui sappiamo dare con certezza input ed output. Esempi magari storici, come ad esempio il valore di oggi in borsa (output) di una certa azione data l’oscillazione del suo valore nel corso dell’anno passato (input); oppure la situazione meteo di oggi (output) date le situazioni meteo dei giorni precedenti (input). In soldoni, nella **fase di apprendimento**, dobbiamo dare alla rete una tabella che comprende i valori di input ma anche i valori di output. Più valori daremo alla rete, più precisa sarà in futuro.

Da qui, utilizzando un algoritmo di *backpropagation* (o “propagazione all’indietro”): la rete imposta dapprima dei **pesi casuali** su tutta la rete, e poi comincia ed eseguire **in ciclo**:

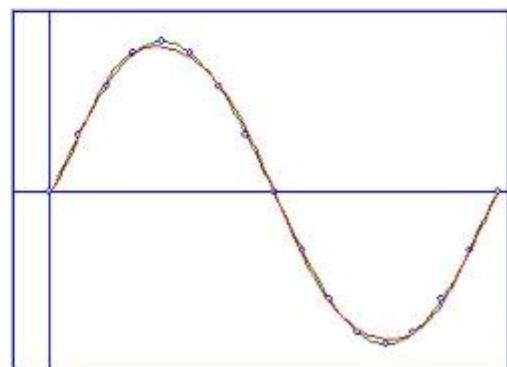
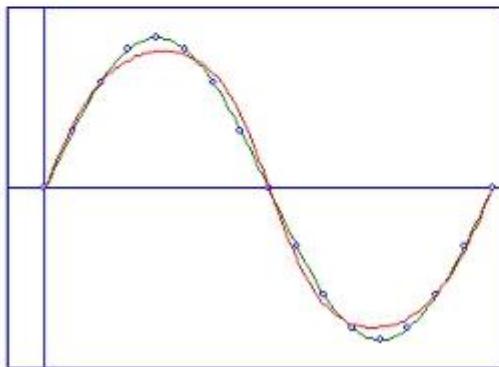
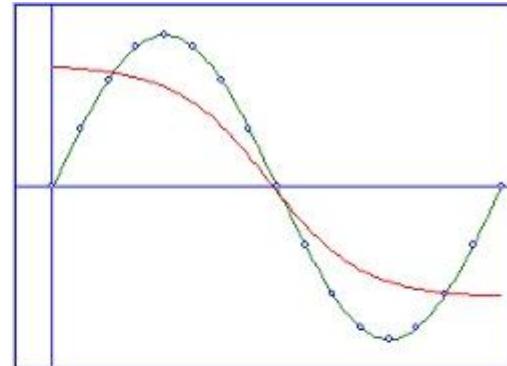
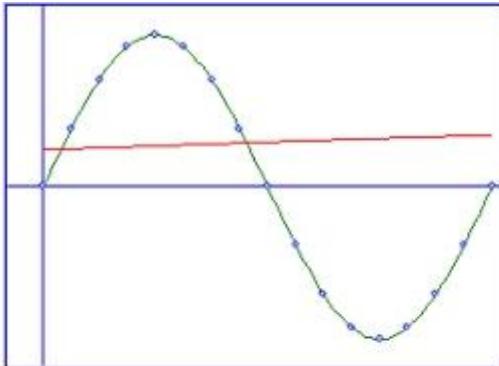
- calcolo del valore di output, dati valori di input dalla tabella di apprendimento (l’output sarà quasi sicuramente lontano da quello richiesto)
- calcolo dell’errore tra l’output della rete e l’output richiesto, ossia quello della tabella di apprendimento

- piccola modifica dei pesi

fin quando l'errore tra l'output della rete e l'output richiesto dalla tabella non è molto basso. Quindi la rete ricomincia con i prossimi valori dalla tabella, e così via, fin quando i pattern dati dalla rete e quelli richiesti dalla tabella non sono **tutti** molto vicini. Per farvi un esempio più concreto, è come se voi doveste imparare a **lanciare la palla nel cesto**: la prima volta sarà troppo lunga, la seconda troppo corta, ma probabilmente dopo decine e decine di tentativi riuscirete a fare canestro.

A questo punto sapremo che la nostra rete, per ogni valore di input preso dalla tabella di apprendimento, ci consegnerà con un piccolissimo errore il valore di output atteso. Nel caso in cui i valori di input saranno “nuovi”, ossia se la rete neurale non avrà un’esperienza diretta con quel pattern di dati, l’errore sarà comunque molto molto basso: il risultato sarà quindi quello statisticamente **più probabile**. (Siri, per farvi un esempio, non ha problemi a riconoscere voci anche totalmente diverse, perché riesce a gestire l’errore dando come risultante la frase più probabile)

Qui sotto trovate un esempio concreto di addestramento di una rete neurale artificiale. E’ stato richiesto alla rete di imparare la funzione  $\text{sen}(x)$ . Nella prima immagine, in alto a sinistra, la linea verde è quello che ci aspettiamo dalla rete, la linea rossa è l’output della rete senza addestramento. In alto a destra, vediamo la rete dopo 1.000 cicli, in basso a sinistra dopo 10.000 cicli, in basso a destra dopo 100.000 cicli.



addestramento di una rete neurale alla funzione  $\text{sen}(x)$

Come potete vedere dall’immagine, la rete dopo 100.000 cicli di apprendimento riesce con pochissimo errore a simulare la funzione, allontanandosi maggiormente dalla linea richiesta solo nei 2 picchi, massimo e minimo.

Ovviamente la rete neurale potrà essere successivamente addestrata anche sui suoi stessi errori: basterà aggiungere alla tabella di apprendimento i dati mancanti e la rete diventerà sempre più accurata e precisa.

## Campi di applicazione e vantaggi

---

Le reti neurali, proprio per la caratteristiche di malleabilità, basso errore, e apprendimento su dati storici vengono applicate in ogni campo dove si deve fare delle previsioni (sismologia, mercato finanziario, meteorologia, scommesse), laddove, pur avendo degli schemi di base, non si può non tenere conto della variabilità dei dati e dell’errore (sistemi di riconoscimento vocale e facciale, OCR, riconoscimento della scrittura manoscritta), in robotica, in statistica, nel controllo di qualità dei prodotti industriali, in medicina, nei videogiochi, etc.

Le reti neurali hanno il pregio di poter lavorare con enormi quantità di dati. Anzi, essendo a ben vedere un enorme e sofisticato **sistema statistico**, le reti neurali lavorano meglio con ingenti quantità di esempi di apprendimento. Inoltre, facendo richiesta di output su dati “limite” o eccezioni, la rete neurale difficilmente avrà un calo di prestazioni paragonabile ad un software classico.

## Difetti e svantaggi

---

Anche se molto efficienti, le reti neurali hanno ricevuto la critica di essere delle *black box*: hai creato la rete, hai creato l’algoritmo di apprendimento, hai creato la tabella di esempi, ma dopo l’addestramento vero e proprio non sai come la rete stia funzionando. Se ci fosse un problema in una singola unità della rete, difficilmente si riuscirebbe a capire su quale singolo neurone o sinapsi stia il problema. Una vera e propria *scatola nera*.

Un’altra arma a doppio taglio è quello della “dimensione” della tabella di apprendimento: è vero che la nostra rete funzionerà meglio con una grossa quantità di dati, ma è vero, dall’altra parte, che una piccola quantità di dati renderà inutilizzabile la rete o, ancor peggio, errata. La difficoltà sta proprio nel capire la **quantità minima di dati** di cui abbia bisogno la nostra rete prima di essere perfettamente funzionante.

Inoltre, non esistono teoremi e/o regole che permettano di definire la rete ottimale, con il **giusto numero**, ad esempio, di neuroni nascosti.

Infine, anche se l’idea di rete neurale deriva dalla struttura biologica del cervello, anche se in alcuni campi dà risultati strabilianti o può dare persino l’impressione di avere a che fare

con una persona reale (vedi Siri), ad oggi nessuna rete neurale è paragonabile con quello che è il **pensiero o il ragionamento** umano: una rete artificiale può “parlare”, prevedere dati e riconoscere facce od oggetti, ma nel cervello umano tutto questo è messo in moto o mette in moto una serie di reazioni e meccanismi come possono essere, ad esempio, le emozioni. Il pensiero umano, poi, non ha necessariamente bisogno di dati di input per funzionare.

## Qualche esempio di rete neurale

---

- [20q.net](http://20q.net), fantastico sito che indovina un oggetto, un animale, etc. che avete in mente con 20 domande
- Siri, sistema di riconoscimento vocale per iPhone
- [Grouper](http://grouper.csail.mit.edu), applicazione sperimentale che riesce, dati alcuni punti di colore diverso su una tavolozza, a riconoscere i colori e a suddividere la tavolozza in diverse aree.
- [GNU Backgammon](http://backgammon.gnu.org), applicazione per giocare a Backgammon contro il pc.

Provatevi (soprattutto il primo, è fantastico) e fateci sapere!

*\*Un ringraziamento speciale a Erika Gherardi per l'attenta revisione.\**

Attenzione: l'articolo e la rete neurale descritta sono volutamente semplicistici. Questo per poter introdurre l'argomento anche ai profani