di **Alberto Montanari**, Consigliere e Coordinatore Commissione Industria 4.0 Federmanager Bologna - Ravenna

INTELLIGENZA ARTIFICIALE: COME FUNZIONANO LE RETI NEURALI

In un precedente articolo (cfr. FD2/2019 pagg. 9-12) si è affrontato il problema dell'Intelligenza Artificiale ed i rischi connessi, ma non abbiamo analizzato lo strumento maggiormente usato allo scopo: la rete neurale.

Mi scuso con i lettori più tecnici perché nella spiegazione effettuerò alcune semplificazioni per agevolare la lettura e considereremo solo il tipo più semplice di rete.

Le **reti neurali** sono approssimatori universali e funzionano molto bene per catturare le associazioni o scoprire le regolarità all'interno di un insieme di modelli dove il volume, il numero di variabili o la diversità dei dati è molto grande, dove le relazioni tra le variabili non sono chiare oppure dove sono difficili da descrivere adeguatamente con gli approcci convenzionali.

Concetti di base

L'idea di base dietro una rete neurale è quella di **simulare cellule cerebrali all'interno di un computer** in modo da potergli far imparare cose, riconoscere i modelli e prendere decisioni in modo autonomo. Ciò che sorprende di una rete neurale è che non è necessario programmarla per imparare, lo fa da sola dopo aver impostato i parametri di riferimento.

Le reti neurali sono simulazioni software e sono realizzate programmando computer comuni che lavorano in modo tradizionale. Le reti neurali prodotte in questo modo sono chiamate **reti neurali artificiali (ANN).**

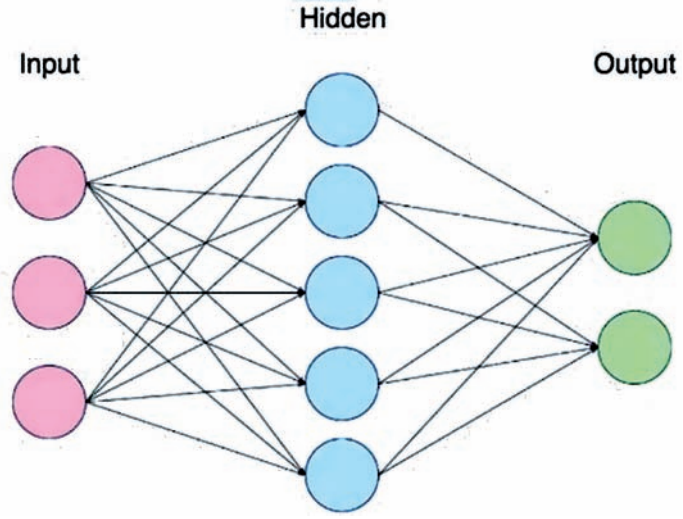
Struttura generale

Una rete neurale artificiale è costituita da 3 componenti, a loro volta costituiti da livelli (layer):

Layer di Input

Layer nascosti (Hidden) di calcolo (uno o più…)

Layer di Output



Tutte le figure sono tratte da www.medium.com

Per definire una rete occorre decidere quanti neuroni ci saranno per ogni livello (layer) virtuale intermedio nascosto ed il numero degli stessi layer nascosti, considerando che la quantità complessiva dei livelli e neuroni ha ovviamente un grosso impatto sui tempi di calcolo.

Se inseriamo dei dati di input in un primo layer, ogni singolo valore avrà il suo neurone dedicato, pensiamo ad una cella di Excel, come dall'esempio precedente. I neuroni sono collegati da sinapsi (collegamenti virtuali) alle quali vengono assegnati dei pesi (parametri numerici virtuali). I pesi sono il tramite con cui le ANN imparano; regolandoli la ANN decide in quale misura i segnali vengono ritrasmessi. Quando si "allena" la rete, in realtà si lavora sulla regolazione dei pesi.

Una schematizzazione

Immaginiamoci una cartella di lavoro Excel con tanti fogli al suo interno; prendiamo il foglio (layer) 1, ogni riga potrebbe ad esempio essere un tipo di auto sul mercato, con ogni colonna che rappresenta una caratteristica ad esempio cilindrata, potenza, peso e consumo.

I layer nascosti potrebbero essere i fogli numerati 2, 3…sui quali trasferiamo i dati del foglio 1 in successione moltiplicandoli per certi fattori (pesi) e introducendo delle formule (attivazione) che ci daranno i nuovi valori; alla fine sul foglio n avremo il nostro risultato.

Questa è una massima semplificazione per visualizzare qualcosa che sta nei componenti di un computer.

Ora immettiamo i dati di un'altra autovettura di cui conosciamo tutto ma non il consumo e desideriamo valorizzarlo, quindi introdurremo i nuovi dati e il sistema ci stimerà il consumo sulla base degli esempi precedentemente forniti ed appresi; è comprensibile

come il risultato dipenda molto dalla qualità e quantità dei dati disponibili.

Reti di feedforward e di feedback

Ci sono diversi tipi di reti neurali, generalmente classificate in reti di feedforward e di feedback; al loro interno sono realizzate tutte le possibili connessioni tra i neuroni.

In una **rete feedforward** i segnali possono viaggiare solo in una direzione, in avanti. Ogni neurone si basa sulla somma ponderata dei suoi ingressi del livello inferiore. I risultati diventano i valori di input che alimentano a loro volta il livello successivo. Questo continua attraverso tutti i livelli e determina l'output finale.

Una **rete di feedback** invece ha segnali che viaggiano in entrambe le direzioni. Questo tipo di rete diventa un sistema dinamico non lineare che evolve continuamente fino a raggiungere uno stato di equilibrio.

Propagazione feedforward

I nostri dati inseriti sono x1 e x2, si parte dalla inizializzazione casuale dei pesi w1, w2 e w3 (pallini colorati viola, gialli e arancioni della fig. sottostante).

I dati al livello di input vengono moltiplicati per i pesi corrispondenti per passare al livello successivo.

I pesi rappresentano le connessioni fra una unità e l'altra; il peso può essere positivo se eccita il neurone successivo o negativo se lo inibisce; ricordiamo che tanto maggiore è il peso, tanto maggiore è la sua influenza.

Ogni unità riceve gli ingressi da quelle alla sua sinistra e li somma tutti, se la somma è superiore ad un certo valore di soglia, l'unità "spara" e attiva i neutroni a cui è collegata alla sua destra.

h1 = (x1 * w1) + (x2 * w1)

h2 = (x1 * w2) + (x2 * w2)

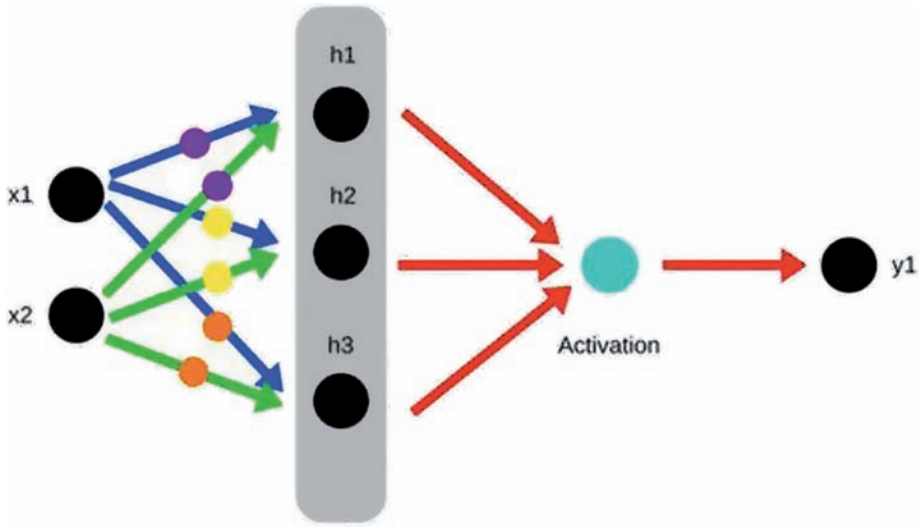
h3 = (x1 * w3) + (x2 * w3)

L'uscita dei livelli nascosti y1 è calcolata attraverso una funzione non lineare f di h1, h2, h3, nota anche come "funzione di attivazione" (pallino azzurro).

y1 = f (h1, h2, h3)

Propagazione feedback

L'errore totale è calcolato valutando la differenza tra il valore atteso 'y' (valore conosciuto) e il valore osservato 'y1' (valore ottenuto dalla propagazione in avanti) attraverso una "funzione di costo". La rete utilizza questa funzione per valutare le prestazioni del modello, espresse come la differenza tra il valore reale e quello previsto, in pratica l'errore che si ha nella valutazione. L'informazione risale all'indietro (feedback) con l'obiettivo di minimizzare la funzione.



Dall'apprendimento ai risultati

L'apprendimento è un processo che avviene ogni volta che alla rete vengono sottoposti nuovi dati. Quando ad una rete neurale viene presentato un nuovo set di dati, essa fa una "supposizione" introducendo pesi casuali. Successivamente calcola di quanto la sua risposta sia differente da quella reale attesa ed effettua un'adeguata regolazione dei suoi pesi di collegamento con una serie anche lunga di ricalcoli (feedback).

Una volta che una rete neurale è stata "addestrata", può essere utilizzata come strumento di analisi su nuove informazioni salvando il modello. Per fare questo l'utente non specifica più alcuna sessione di "allenamento" e permette invece alla rete di lavorare solo in modalità di propagazione in avanti (feedforward). I nuovi input sono presentati al modello dove vengono filtrati e processati dagli strati intermedi come se si stesse svolgendo un training, tuttavia a questo punto il risultato dell'output viene mantenuto e non si verifica alcun feedback.

Una volta creata l'architettura e permesso alla rete di lavorare, non ab-

biamo la necessità (e la possibilità) di comprendere tutte le regole al suo interno.

La gerarchia di concetti

La rete crea una gerarchia di concetti. In questa gerarchia ogni livello trasforma i propri dati di input in una rappresentazione sempre più complessa. Ciò significa che per un'immagine di un viso, ad esempio, l'input potrebbe essere una matrice di pixel, ad ogni pixel corrisponde un valore, quindi un neurone. Il primo livello potrebbe codificare i bordi e comporre i pixel. Lo strato successivo potrebbe comporre una disposizione dei bordi. Quello dopo potrebbe codificare il naso e gli occhi. Il livello seguente potrebbe riconoscere che l'immagine contiene una faccia, e così via. Per dare un ordine di grandezza, un'immagine di piccole dimensioni (268 x 268 pixel) a tre colori RGB richiede in ingresso (i nostri x1, x2….) circa 215.000 neuroni.

Che cos'è l'apprendimento profondo (Deep Learning)?

L'apprendimento profondo, come si potrebbe intuire dal nome, è l'uso di molti livelli per estrarre progressivamente caratteristiche superiori dai dati che inviamo alla rete neurale, quindi l'uso di più livelli nascosti per migliorare le prestazioni dei nostri modelli neurali. L'esempio per eccellenza di un modello di apprendimento profondo è il "feedforward deep network" o **"Multilayer Perceptron" (MLP).**

L'apprendimento profondo eccelle nei settori in cui gli ingressi sono analogici, quindi quantità non in formato tabellare ma immagini di pixel, documenti di testo o file di dati audio e con grandissime quantità di dati di input.

Gli sviluppi

Vi sono molti tipi di reti neurali e continuamente ne vengono sviluppate con risultati eccezionali, particolarmente nell'autoapprendimento; reti alle quali vengono forniti molti dati senza alcuna indicazione e loro autonomamente creano nuove regole, anche estremamente complesse, dotandosi quindi di una sorta di reale "intelligenza". Tipi attualmente molto usati sono la **"Convolutional neural networks (CNNs)"** e la **"Generative adversarial networks (GANs)"**, particolarmente adatti per il riconoscimento delle immagini.

Il **"reinforcement learning"** è una tecnica di apprendimento automatico che punta ad avere sistemi in grado di apprendere ed adattarsi alle mutazioni del loro ambiente tramite una "ricompensa" che consiste nella valutazione delle loro prestazioni.

Gli input al sistema possono provenire dai più svariati sensori, ad esempio, nel caso di un robot che deve imparare a muoversi all'interno di un percorso, i segnali possono essere forniti da sensori di prossimità.

Uno degli obiettivi dell'algoritmo è quello di imparare regole di comportamento.

Qualche considerazione

L'argomento trattato è molto divisivo. Soprattutto negli USA vi sono molte discussioni sull'uso e sul possibile controllo sociale dato dalle reti neurali; per alcuni, come l'imprenditore e visionario Elon Musk, l'intelligenza artificiale rappresenta una minaccia; per altri, come Mark Zuckerberg, al contrario, saranno le macchine per l'apprendimento a rendere la nostra vita migliore e più sicura in futuro. Nella storia dell'umanità, passando per le varie rivoluzioni tecnologiche, non si è mai assistito a nulla di così innovativo e siamo solo all'inizio. Le superpotenze e le grandi aziende tecnologiche, ovviamente a scopi anche militari, stanno sviluppando algoritmi e macchine sempre più potenti, potremmo arrivare alle macchine che si auto progettano. Chi sarà in grado di gestirle? E quale etica avranno?

Si stanno avverando scenari da film di fantascienza di pochi anni fa: ricordiamoci di "*2001: Odissea nello spazio*", con la presenza HAL 9000, un computer dotato di sentimenti e di una "psiche" instabile e malata.

Un po'di matematica…

La funzione di costo

Vi sono varie funzioni di costo ma una delle più utilizzate è la cosiddetta "radice dell'errore quadratico medio", in inglese "root mean square error", RMSE; per minimizzarla si usa una tecnica chiamata "gradient descent".

In pratica si scende lungo la pendenza del grafico per arrivare al minimo della funzione.

La rete impara quindi riducendo progressivamente la differenza tra l'output effettivo e quello previsto fino al punto in cui i due elementi coincidono.

Le derivate parziali

Il gradiente di una funzione è definito come il vettore che ha come compo-

nenti le derivate parziali della funzione.
Le derivate parziali rispetto ad ogni peso sono gli elementi che compongono il vettore di gradiente della nostra funzione di costo e sono la misura dei contributi di ciascun peso. La backpropagation ci permette quindi di calcolare l'errore attribuibile ad ogni neurone, calcolando le derivate parziali e infine il gradiente in modo da utilizzarne la discesa.
Le derivate parziali sono conteggiate derivando l'errore in funzione di ogni peso, moltiplicandole per un numero chiamato "learning rate" η (tasso di apprendimento); il risultato è sottratto dal peso di riferimento.
w1 = w1 - (η * ∂(err) / ∂(w1))
w2 = w2 - (η * ∂(err) / ∂(w2))
w3 = w3 - (η * ∂(err) / ∂(w3))

Le distorsioni e i tassi di apprendimento

Anche se inizializziamo pesi casuali per poi gradualmente adattarli fino a ricavare il risultato atteso, in realtà nel calcolo vengono inserite delle distorsioni ("bias"), che sono pesi aggiunti ai layer nascosti. Mentre il ruolo del livello nascosto è quello di mappare la forma della funzione contenuta nei dati, il ruolo dei bias è quello di spostare la funzione appresa in modo che si sovrapponga il più possibile alla funzione originale.
Mentre la rete neurale è usata per automatizzare la caratterizzazione, vi sono alcuni parametri che dobbiamo ancora inserire manualmente come il tasso di apprendimento e la funzione di attivazione.
Il tasso di apprendimento η (learning rate) è un parametro cruciale. Se il tasso è troppo basso, anche dopo un lungo periodo di lavoro della rete, saremo lontani dai risultati ottimali, invece, se il tasso è troppo alto, salteremo alle conclusioni troppo presto e con scarsa precisione.

La funzione di attivazione

Con la funzione di attivazione decidiamo la soglia con la quale i neuroni saranno attivati, quindi quali informazioni saranno passate agli altri livelli. Senza funzioni di attivazione, le reti perdono il loro potere di apprendimento e diventano una sorta di regressioni lineari.
Tecnicamente la rete realizza un'analisi predittiva utilizzata per misurare la relazione tra la variabile dipendente (ossia ciò che vogliamo prevedere) e una o più variabili indipendenti (le nostre caratteristiche), stimando delle probabilità tramite una funzione di attivazione.
Queste probabilità vengono trasformate in valori binari per poter fare una previsione, assegnando il risultato della previsione stessa alla classe di appartenenza.
Se ad esempio otteniamo un valore 0,8 dopo aver applicato la funzione, diremo che l'input ha generato una classe positiva e verrà assegnato alla classe 1 (viceversa se avesse ottenuto valore < 0,5).
Le funzioni di attivazione più usate sono la sigmoide , la tangente iperbolica o le funzioni a rampa e servono sia per introdurre la non linearità nel modello sia per assicurarsi che determinati segnali rimangano all'interno di specifici intervalli.
La non linearità consente di generalizzare problemi di alta dimensionalità, ovvero con dati molto numerosi, e ricondurli a dimensioni inferiori.