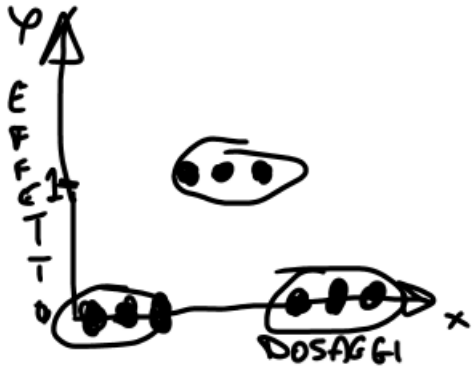
Data Mining e Bioimmagini

Interpolazione Lineare, Artificial Neural Network, Percetor

Prof. Pierangelo Veltri – 30/10/2023- Autori: Accetturo

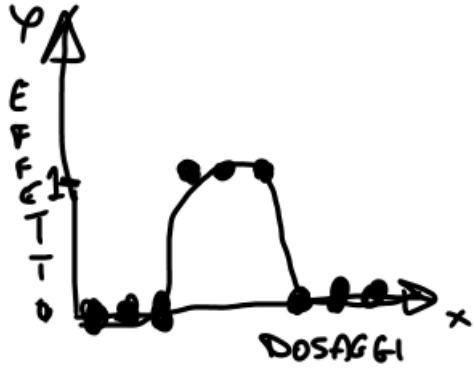
INTERPOLAZIONE LINEARE

L'interpolazione lineare è un metodo matematico che ci consente di stimare valori sconosciuti all'interno di un intervallo basandoci su punti noti. L'idea di base è quella di collegare questi punti noti con una linea retta, supponendo che la relazione tra i valori noti sia approssimativamente lineare. Tuttavia, è importante notare che l'interpolazione lineare funziona in modo efficace solo su intervalli o range di valori specifici. Se il range delle variabili è ampio e le x crescono notevolmente, l'interpolazione lineare può perdere di precisione e ridurre tutto nuovamente a zero.



Per spiegare meglio questo concetto, consideriamo un esempio con un farmaco. Immaginiamo di voler studiare il dosaggio e il meccanismo di somministrazione di questo farmaco. L'obiettivo non è solo capire se il farmaco funziona o no, ma anche come varia l'effetto in base al dosaggio. Per fare ciò, possiamo creare un grafico con l'asse delle x che rappresenta il dosaggio del farmaco e l'asse delle y che rappresenta l'effetto ottenuto.

Se iniziamo con dosaggi molto bassi, noteremo che l'effetto sperimentato è nullo, come evidenziato dai punti corrispondenti a dosaggi bassi sull'asse x. Man mano che aumentiamo gradualmente il dosaggio, osserveremo un incremento dell'effetto, che si avvicina a un valore di uno. Qui, uno rappresenta un effetto significativo. Tuttavia, la relazione tra dosaggio ed effetto non è lineare e dipende dalla gamma di dosaggi studiata.



La curva di risposta può variare, ma l'idea fondamentale è quella di identificare come il dosaggio influenza l'effetto, il che ci permette di calibrare la somministrazione del farmaco in modo efficace. Questo tipo di analisi è utile per determinare il dosaggio ideale del farmaco in base alle risposte osservate. Il comportamento di questa curva potrebbe sembrare "strano", in quanto inizia da zero, aumenta gradualmente con l'incremento dei dosaggi e poi diminuisce nuovamente fino a tornare a zero. Possiamo chiamare questa funzione "sempre segnato x", poiché sembra seguire costantemente la stessa tendenza di aumento e diminuzione.

Infine, quando affrontiamo problemi più complessi e abbiamo bisogno di approssimare relazioni tra variabili in modo più accurato rispetto all'interpolazione lineare, possiamo utilizzare una rete neurale artificiale. Questo è un meccanismo di classificazione interno che sfrutta la capacità di apprendimento di una rete neurale per risolvere problemi complessi. In breve, mentre l'interpolazione lineare è utile per stime approssimative su intervalli limitati, le reti neurali artificiali sono strumenti più potenti quando si tratta di approssimare relazioni complesse tra variabili.

ARTIFICIAL NEURAL NETWORK

Un'artificial neural network si basa sul funzionamento biologico di un cervello, utilizzando concetti simili a quelli dei neuroni e dei percettori biologici. L'obiettivo è apprendere da dati di input e creare una funzione basata su questo apprendimento. La costruzione di una rete neurale coinvolge elementi che operano su valori di input, e più elementi sono coinvolti, migliori sono i risultati ottenuti.

Durante la fase di addestramento, ogni elemento di input riceve un valore e fornisce una risposta basata su questo input. Immaginate di avere diversi studenti, ognuno dei quali reagisce in modo diverso all'input, rappresentando così diverse funzioni con diverse risposte.

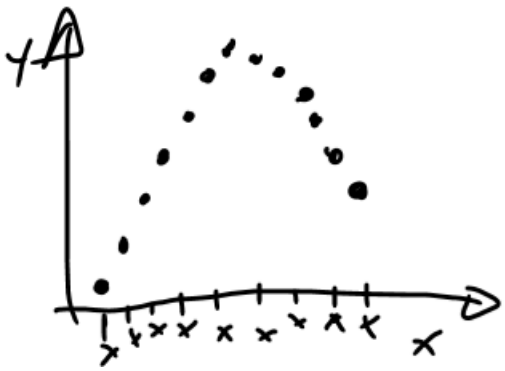
L'idea principale delle reti neurali è quella di simulare il comportamento dei neuroni biologici. Un neurone biologico è caratterizzato da un nucleo e terminazioni nervose. La struttura è rilevante poiché alcune funzioni possono reagire a segnali specifici, mentre altre rimangono indifferenti. La tipologia del segnale, la sua intensità e la soglia di attivazione sono importanti.

Ad esempio, immaginate un insegnante con molti studenti in classe. Alcuni studenti potrebbero essere distratti da fattori come la fame o l'accompagnamento di un amico, mentre altri potrebbero essere più attenti a ciò che li interessa. Questa variazione di interesse è simile al comportamento dei neuroni che rispondono a vari stimoli.

In generale, le reti neurali si concentrano sulla commutazione tra attivazione e inibizione, poiché ci sono caratteristiche specifiche che svolgono diverse funzioni. Questo concetto è analogo alle funzioni muscolari scheletriche e alle funzioni legate alla digestione. Inoltre, il tempo di reazione e il tempo di elaborazione sono importanti nei neuroni. Un neurone può trasmettere informazioni in modo eccitatorio o inibitorio, simile a come un farmaco antidolorifico inibisce il segnale del dolore. Anche se il farmaco non risolve il problema alla radice, inibisce la percezione del dolore. Quando l'effetto del farmaco svanisce, il dolore può tornare.

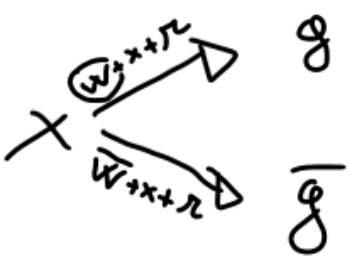
Quindi, il processo funziona nel seguente modo:
- ciascun valore numerico di input è elaborato dalle funzioni interne della rete neurale.
- Ogni funzione interna fornisce una risposta in base all'input fornito, e questi valori sono successivamente aggregati in un unico valore.
- In sostanza, il processo di allenamento della rete neurale comporta l'apprendimento da una serie di esempi.

Tornando all’esempio del farmaco



La rete neurale cerca di catturare queste variazioni per rappresentare al meglio il comportamento del dosaggio sull'effetto.

L'elaborazione del dosaggio avviene in questo modo:
- inizialmente, viene preso un valore di input, rappresentato da "x".
- Questo valore viene quindi moltiplicato per un peso specifico, ottenendo così "peso moltiplicato x". È importante notare che ci sono due funzioni distinte coinvolte, ognuna con il proprio peso. Quindi, otteniamo due valori separati: "peso moltiplicato x" per una funzione e "peso moltiplicato x" per un'altra funzione.



- Successivamente, entrambi questi valori, che differiscono tra di loro, vengono introdotti come input a una funzione denominata "funzione G segnato". Ciò significa che per uno stesso valore di "x", questa funzione "G segnato" restituirà un valore di "y". Tuttavia, questi due valori di "y" derivati dalle due funzioni di cui sopra sono distinti.
- Alla fine, l'obiettivo è quello di ottenere un unico valore di "y" da questa funzione "G segnato", che rappresenti il risultato desiderato.

In sintesi, l'elaborazione coinvolge l'input di un dosaggio, la moltiplicazione di "x" per pesi differenti, e la conseguente generazione di valori di "y" distinti da due funzioni. Questi valori di "y" sono quindi combinati tramite la funzione "G segnato" per produrre un singolo risultato.

STRUTTURA E PROBLEMI RISOLVIBILI

La progettazione di una rete neurale coinvolge diverse fasi, tra cui la scelta del numero e del tipo di unità, la determinazione della struttura morfologica, la codifica degli esempi di addestramento in termini di ingressi e uscite dalla rete, l'inizializzazione e l'addestramento per determinare i pesi delle interconnessioni tramite il set di addestramento.

Una rete neurale artificiale è composta da un insieme di nodi, pesi e soglie. La definizione di questi parametri, la codifica degli ingressi e la selezione delle funzioni per ciascun nodo dipendono dalla natura del problema da risolvere. In genere, gli ingressi rappresentano dati noti, poiché il processo di addestramento richiede un insieme di dati noti.

Durante il processo di addestramento, una serie di dati di input è utilizzata per allenare le funzioni dei nodi. La complessità delle funzioni di addestramento può variare a seconda dei dati di input. Ad esempio, nell'addestramento di un modello per il riconoscimento facciale, è necessario utilizzare dati di input che rappresentano variazioni nelle posizioni facciali, consentendo al modello di riconoscere volti da diverse angolazioni e condizioni di luce.

Questo processo di addestramento migliora la capacità del modello di generalizzare e riconoscere oggetti o pattern simili anche in diverse condizioni. Un modello di riconoscimento facciale addestrato con una varietà di immagini di volti sarà in grado di riconoscere volti da diverse angolazioni e condizioni di luce, rendendolo più affidabile.

Le reti neurali presentano diverse caratteristiche, tra cui la rappresentazione di istanze mediante molte feature con valori reali, la funzione obiettivo a valori reali, la presenza di esempi rumorosi, tempi di addestramento lunghi e la necessità di valutare rapidamente la rete appresa. Inoltre, non è sempre cruciale comprendere la semantica della funzione attesa, ma piuttosto ottenere risultati precisi.

Problematiche affrontare per costruire il modello robusto

Nella costruzione di un modello preciso e robusto, l'obiettivo è caratterizzare gli elementi in modo accurato. Questo processo coinvolge l'interazione tra diverse persone, ognuna delle quali ha comunicato informazioni relative alla linearità di un processo. L'obiettivo è ottenere una funzione che vada oltre una semplice distinzione tra "bravo" e "non bravo", ma che fornisca dettagli più specifici.

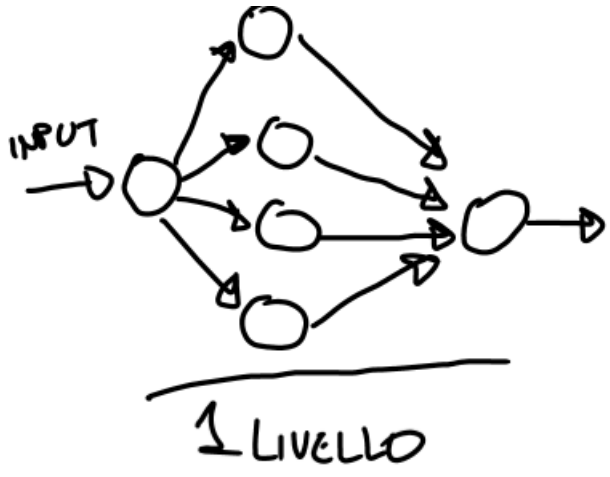
Per raggiungere questo obiettivo, si utilizzano più modelli intermedi durante il processo di allenamento. L'input viene preso da un insieme di dati diversi, e questo comportamento, come precedentemente discusso, non è rappresentabile da una funzione lineare. Invece, si genera una funzione "segnato X", che rappresenta il risultato complessivo del training.

Per costruire tale modello, si devono affrontare due problemi principali. In primo luogo, si devono scegliere le funzioni GE utilizzate durante il training della rete interna. Queste funzioni, spesso comuni tra i neuroni, possono variare solo in termini di peso e rumore. In secondo luogo, è necessario determinare il valore di peso e rumore per ciascuna funzione. Questa fase richiede l'utilizzo di algoritmi di calcolo dei pesi ottimali e implica prove iterative per raggiungere i valori migliori.

Una volta determinate le funzioni, i pesi e i rumori, è possibile definire la rete e iniziare il training. Il training della rete comporta l'allenamento delle funzioni all'interno dei neuroni per produrre una funzione di classificazione basata sulla rete. Questa funzione sarà utilizzata per valutare i nuovi dati in ingresso.

In conclusione, il processo comporta la scelta delle funzioni, la definizione dei pesi e dei rumori, il training della rete e la generazione di una funzione di classificazione basata sulla rete, che è in grado di catturare il comportamento non lineare dei dati in ingresso.

ESEMPIO IMMAGINI 3D



Un'architettura a singolo livello consiste di input, pesi, rumore e funzioni. Queste funzioni vengono utilizzate per calcolare un output che rappresenta una combinazione di punti.

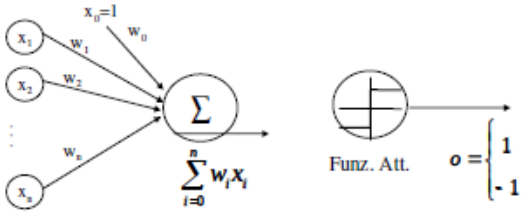
Un'idea importante è quella di utilizzare il feedback degli studenti per migliorare la precisione del sistema. Se il giudizio degli studenti non viene utilizzato direttamente come valore di output, ma piuttosto come input per colleghi di un secondo livello, si può ottenere una maggiore precisione. Questi colleghi possono arricchire l'output iniziale con ulteriori informazioni, ottenendo un output più preciso. Questo processo consente di coprire una vasta gamma di casi e migliorare l'accuratezza del sistema.

L'approccio può essere esteso a più livelli, in cui le funzioni di output dei livelli precedenti vengono combinate con pesi appropriati per ottenere un'uscita finale che è ulteriormente arricchita con informazioni dettagliate. Questa strategia aumenta la precisione dell'output generale.

Nel caso delle immagini 3D, più attributi vengono utilizzati per rappresentare un punto nello spazio, come le coordinate XYZ e i valori di colore. Questo approccio permette di ricostruire immagini complesse con maggiore precisione.

In generale, questo concetto è alla base degli algoritmi di intelligenza artificiale utilizzati per la ricostruzione delle immagini 3D, consentendo di ottenere risultati più accurati e dettagliati. La capacità di adattare il modello in base alle esigenze specifiche è fondamentale per il successo dell'applicazione. TensorFlow fornisce gli strumenti per sperimentare con diverse configurazioni e trovare la soluzione migliore.

PERCETTRONE



La rete neurale più semplice è basata sul percettrone, che utilizza una funzione di attivazione molto elementare. La funzione di attivazione è una semplice funzione che prende in input dei valori e restituisce un risultato basato su una condizione. Nel caso più semplice di un percettrone, questa condizione è che la sommatoria dei valori di ingresso moltiplicati per dei pesi sia maggiore o uguale a zero. La funzione restituisce uno se questa condizione è verificata e meno uno altrimenti.

Questo tipo di rete è progettato per la classificazione. L'obiettivo è assegnare una delle due classificazioni possibili in base al risultato della funzione di attivazione. La fase di addestramento coinvolge l'uso di dati di input noti e l'associazione di una funzione di attivazione che risponde correttamente a tali dati.

La rappresentazione di queste funzioni è fondamentale per la fase di apprendimento. In generale, se si utilizzano più di un perceptron, si ottiene una rappresentazione più complessa delle funzioni. Ad esempio, utilizzando tre perceptrons con differenti pesi e funzioni di attivazione, è possibile sviluppare una funzione di apprendimento che risponde a tre ingressi diversi e restituisce un risultato in base a una condizione specifica.

I pesi associati a ciascun ingresso svolgono un ruolo cruciale nella definizione del comportamento della funzione. Ogni peso è moltiplicato per il corrispondente valore di ingresso, e la somma di queste moltiplicazioni determina se la condizione della funzione di attivazione è soddisfatta.

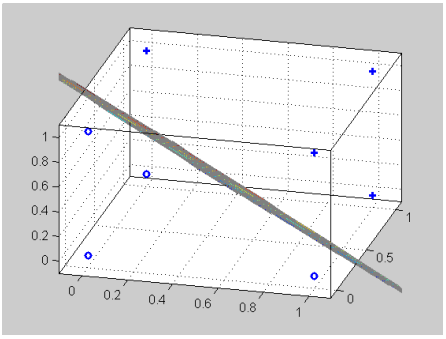
In sostanza, il perceptron più semplice è un modello di base che può essere utilizzato per problemi di classificazione. È un primo passo nell'implementazione di reti neurali più complesse che possono affrontare problemi più sfaccettati.

Addestramento percettrore

1. Inizializza i pesi *(w0, w1, …, wd)*
2. Ripeti: (Per ciascun esempio del training set (xi, yi))
    a. Calcola
    b. Calcola l'errore
    c. Aggiorna i pesi
3. Fino a quando la condizione di uscita e verificata

La fase di addestramento, come accennato in precedenza, è un processo iterativo.
L'obiettivo è aggiornare i pesi in modo che l'errore sia il più vicino possibile a una soglia predefinita, fino a ottenere un valore di uscita che sia in linea con il valore desiderato.
Questa fase di aggiornamento dei pesi è una parte fondamentale del processo di apprendimento.

Poiché f(w,x) è una combinazione lineare di variabili di input, la separazione è definita da un iperpiano.
Per i problemi separabili non linearmente, l'algoritmo di apprendimento del percettrone fallirà perché nessun iperpiano lineare può separare perfettamente i dati



Esempio

$\lambda = 0.1$

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	-0.2	0	0
2	0	0	0	0.2
3	0	0	0	0.2
4	0	0	0	0.2
5	-0.2	0	0	0
6	-0.2	0	0	0
7	0	0	0.2	0.2
8	-0.2	0	0.2	0.2

Epoch	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	0	0.2	0.2
2	-0.2	0	0.4	0.2
3	-0.4	0	0.4	0.2
4	-0.4	0.2	0.4	0.4
5	-0.6	0.2	0.4	0.2
6	-0.6	0.4	0.4	0.2

Nel processo di addestramento, si calcola l'errore confrontando la funzione di uscita prevista (calcolata utilizzando i pesi assegnati ai singoli input) con l'uscita desiderata che è già conosciuta. Questo passaggio consiste nel determinare la discrepanza tra la risposta simulata e il valore desiderato. In termini più semplici, si confronta ciò che il modello predice con ciò che dovrebbe predire. Questa differenza è l'errore.

Quindi, in un ciclo di apprendimento, quando un input attiva la funzione di uscita e genera un'informazione, questa informazione viene corretta in base all'errore tra il valore previsto e il valore desiderato. Si compara il risultato calcolato (che è una stima) con il risultato effettivo (che è noto), e la differenza tra di essi è l'errore. Questo processo di confronto e correzione aiuta a regolare i pesi e migliorare il modello.