Lezione11

Table of contents

Reti Neurali Ricorrenti e di Hopfield

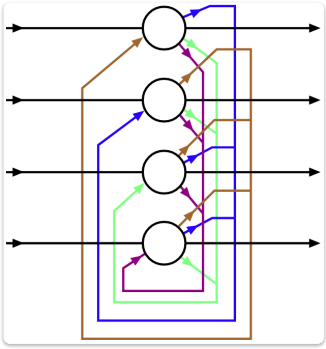
Fino adesso abbiamo visto le reti feed forward.
Se lo scopo dell'AI è quello di simulare al meglio il cervello umano, ci rendiamo conto che forse non stiamo seguendo la strada corretta: la nostra mente funziona piuttosto come un grafo connesso con caratteristiche locali per ogni neurone.

Cambiamo ora percorso, prendendo un rete neurale che abbia un struttura ad *anelli*.
Una rete non sarà più una foresta ma magari un output rientrante, sempre con i suoi pesi, con la differenza sostanziale che gli strati sono scomparsi.
Un qualche neurone, ha output che tornerà indietro, complicando le cose: l'output diventa input, il valore d'ingresso dei neuroni collegati a questo oggetto anche loro assumeranno cambiamento e così via.

L'ipotesi di questa rete che stiamo analizzando, è da seguire con cautela perché neuroni con questo comportamento quasi caotico, potrebbero:

- portare a una *divergenza* della funzione di attivazione (`ReLU`, o lineare)
  una variazione positiva porta a una crescita positiva indefinita
- generare oscillazioni
  un output porta a una crescita, il nuovo input è in calo, il nuovo output risale e così via
- stabilizzarsi su un output asintotico
  che se studiato, potrebbe comunque dare una uscita finale.

John Joseph Hopfield (1933)



Una rete di Hopfield

Di queste *reti ricorrenti* ne studieremo una in particolare che è la **rete di Hopfield**, popolarizzate dal fisico americano John Hopfield. Sono state molto famose agli inizi degli anni '80 e hanno portato, proprio come al giorno d'oggi, a una ricerca approfondita dell'AI. Una rete di Hopfield è una rete neurale ricorrente la cui funzione di attivazione dei neuroni è la funzione **signum**

$$\mathbf{sgn}(x)\begin{cases}+1 & \text{se} \quad x>0\\ 0 & \text{se} \quad x=0\\ -1 & \text{se} \quad x<0\end{cases}$$

(Osserviamo la rete nella tabella) Avremo un insieme di neuroni, tutti collegati gli uni con gli altri, senza auto-anelli (niente neuroni che collegano se stessi). L'output di ogni neurone viene calcolato con la funzione segno, in funzione dell'input che fissato un istante di tempo prende il nome di **istante zero**. Il valore di bias sembra non esserci, in verità c'è e può essere fisso per tutti i neuroni. Nell'istante di tempo, gli output possono essere di 3 tipi, per come è strutturata la funzione segno. Dopo alcuni cicli per lasciare che la rete si stabilizzi su di un valore asintotico, avremo il nostro risultato.

La rete di Hopfield può essere vista come una *memoria associativa*, realizzata nei nostri cervelli: la memoria non è indirizzata, ma abbiamo bisogno di un'informazione parziale per poter risalire alle informazioni complete (di norma non ci ricordiamo cosa abbiamo mangiato una settimana fa a una tal ora). Questo va notato perché questo tipo di rete è lo stesso principio che segue la natura del *riconoscimento delle immagini*.

Questo tipo di sistemi dinamici non lineari è complicato da realizzare dal punto di vista matematico, perché ci è difficile visualizzare un risultato che è praticamente istantaneo. Quella che piuttosto viene fatta è una *simulazione sincrona a tempo discreto*: definiti istanti di simulazione, per ciascuno andiamo a ricalcolare i nuovi output. Semplificare ci permette di realizzare la simulazione come semplici cicli `for`, anche se ci stiamo distaccando dalla vera simulazione di cervello, in quanto gli assoni hanno una lunghezza, traversati dalle cariche.

L'output del singolo neurone i-esimo, viene calcolato con la funzione segno applicata all'input dove questo è input della rete moltiplicato scalarmente con i vettori dei pesi.
Chiamando s **stato della rete** in un istante di tempo, l'uscita di tutti i neuroni, questa va a finire come input di ogni neurone, sottraendo bias.

$$o_i=\mathbf{sgn}(\mathbf{w}_i\cdot\mathbf{s}-b_i)=\mathbf{sgn}\left(\sum_{j=1}^{n}w_{i,j}s_j-b_j\right)$$

La matrice dei pesi $W=(w_{i,j})_{n,n}$, compresa da tutti i pesi dei neuroni della rete, sarà una con la diagonale principale esclusivamente composta da 0, siccome gli auto-anelli non sono permessi.

In un istante di tempo siamo in grado di associare alla rete una sua energia, detta **energia della rete**. Come vedremo, è molto importante andare a calcolare l'energia della rete perché come in tutti i sistemi dinamici, la convergenza si ha ai minimi dell'energia. Fissato lo stato della rete $\mathbf{s}\in\mathbb{R}^n$, fissata la matrice dei pesi W e fissato il vettore di bias $\mathbf{b}$, con

$$E(\mathbf{s})=-\frac{1}{2}\mathbf{s}^{\mathsf{T}}W\mathbf{s}+\mathbf{b}\cdot\mathbf{s}$$

calcoleremo la sua energia.

> ⓘ **Essendo un fisico e avendo studiato il ferromagnetismo, Hopfield creerà l'energia della rete, come somiglianza diretta dell'energia nel campo della meccanica.**

Teorema di Convergenza delle Reti Neurali di Hopfield

Teorema

Fissata l'espressione sopra, possiamo generare un teorema che dia un senso alla rete.

Presa una rete di Hopfield formata da $n\in\mathbb{N}_+$ neuroni, costruendo una matrice dei pesi $W_{n\cdot n}$ con diagonale a 0, con vettore di pesi bias $\mathbf{b}$, se W è simmetrica (peso neurone i-esimo è uguale al peso neurone j-esimo)

allora indipendentemente dallo stato iniziale tutti i neuroni della rete, tenderanno a uno stato stazionario che corrisponde a uno dei minimi locali dell'energia della rete.

La rete convergerà a un valore asintotico.
Se sappiamo quali sono i minimi locali, sappiamo dove converge la nostra rete.
Se siamo capaci di trovare pesi in modo da ottenere una rete che trovi minimi dove vogliamo noi, siamo anche capaci di trovare una rete che converga dove vogliamo noi.

Da questo teorema capiamo come mai la rete di Hopfield non sia in addestramento supervisionato: a noi non interessa comunicare, dato input, quale sia l'output collegato; ci interessa dire soltanto quali sono i pesi per raggiungere i minimi e quindi la convergenza. Partendo da uno stato della rete, questa converge al minimo locale più vicino.

Dimostrazione

Vediamo la dimostrazione del teorema, per capire meglio la situazione.

Considerando un passo di simulazione asincrono (istante di tempo in cui output diventa input), prendiamo l'indice di un neurone a e andiamo a vedere cosa succede a questo. $\mathbf{s}$ sarà lo stato della rete prima della simulazione (istante di campionamento), $\mathbf{s}'$ sarà lo stato dopo la simulazione. Vediamo la differenza dell'energia tra le due situazioni.

$$E(\mathbf{s}') - E(\mathbf{s}) = -\frac{1}{2}\left(\sum_{i=1}^{n} w_{i,a}s'_i s'_a + \sum_{j=1}^{n} w_{a,j}s'_a s'_j - \sum_{i=1}^{n} w_{i,a}s_i s_a - \sum_{j=1}^{n} w_{a,j}s_a s_j\right)$$
$$+b_a(s'_a - s_a)$$

La formula può essere semplificata siccome $w_{a,a}$ usa come ipotesi la simmetria.

$$E(\mathbf{s}') - E(\mathbf{s}) = -\left(\sum_{i=1}^{n} w_{i,a}s'_i s'_a + \sum_{i=1}^{n} w_{i,a}s_i s_a\right) + b_a(s'_a - s_a)$$

Sfruttando la diagonale nulla, riscriviamo.

$$E(\mathbf{s}') - E(\mathbf{s}) = -(s'_a - s_a)\left(\sum_{i=1}^{n} w_{i,a}s_i - b_a\right)$$

3 sono i casi possibili che si generano dal calcolo di questa equazione:

1. lo stato del neurone a (s_a) non è cambiato a causa dello stato di aggiornamento, quindi $(s'_a - s_a) = 0$, l'energia prima del campionamento è uguale all'energia dopo il campionamento, l'energia non cambia
$$E'(\mathbf{s}) = E(\mathbf{s})$$
2. s_a passa da -1 a $+1$, quindi $(s'_a - s_a) = 2$, con cambio di segno
$$E'(\mathbf{s}) \leq E(\mathbf{s})$$
3. s_a passa da $+1$ a -1, quindi $(s'_a - s_a) = -2$, con cambio di segno
$$E'(\mathbf{s}) \leq E(\mathbf{s})$$

Stiamo dicendo che viene fatta una *discesa del gradiente*: a un certo punto l'energia raggiungerà una situazione in cui può soltanto crescere quando non può e quindi rimanere dov'è. Un grafo orientato, con pesi reali completamente connesso senza auto-anelli e matrice di pesi annessa, ha punti di convergenza che non dipendono dallo stato iniziale.

Se vogliamo utilizzare la rete di Hopfield come una memoria associativa, quello che dobbiamo fare per i ricordi che vogliamo memorizzare, è che questi siano associati ai minimi dell'energia della rete. Per farlo non ci serve il supervisore che ci dica quale è l'output, piuttosto che ci dica quali sono i punti da utilizzare come ricordi. La rete di Hopfield non è per rinforzo perché non è detto il training set venga riproposto; non è nemmeno supervisionata siccome non prevede output corretti per dati input.

Teorema di Addestramento con la Regola di Hebb

Teorema

La regola di Hebb è uno dei modi utilizzati per costruire i pesi per una rete di Hopfield.

Prevede di avere un training set formato da m vettori che vogliamo memorizzare all'interno della rete, vettori che non scegliamo liberamente, richiediamo che tra di loro siano mutualmente ortogonali e privi di componenti nulle:

- se andiamo a vedere i vettori da memorizzare, questi comprenderanno sempre $\{-1, +1\}$ e mai 0, anche se proprio non è un problema siccome non c'è ritorno;
- presi due vettori del training set, il loro prodotto scalare deve valere 0 (o molto vicino).

In uno spazio di dimensione $\mathbb{R}^2$, stiamo dicendo che questo avrà sempre e soltanto 2 vettori mutualmente ortogonali; ci servirà un numero di neuroni abbastanza elevato per poter memorizzare tutti gli elementi del training set. Seguendo queste ipotesi, la regola ci calcola la matrice dei pesi.

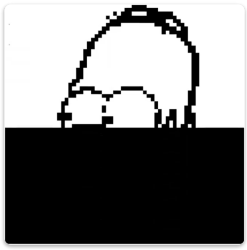
$$W = \frac{1}{m} \sum_{k=1}^{m} \mathbf{x}_k \mathbf{x}_k^{\mathsf{T}} - I_n$$

- sottraendo la matrice identità I di dimensione n, stiamo eliminando la diagonale principale, soddisfando così la prima delle ipotesi viste sopra;
- per come sono fatti i prodotti, ci accorgiamo che la matrice è simmetrica, l'ipotesi del prodotto scalare nullo viene (circa) soddisfatta.

Il vettore di bias viene impostato a 0, siccome questo non viene sfruttato in alcun modo dalla regola di Hebb.

Dimostrazione

Consideriamo una rete di Hopfield per cui è stato utilizzato il vettore sottostante per costruire la matrice di pesi.

	
Input originale addestrante la rete	Nuovo input dopo addestramento

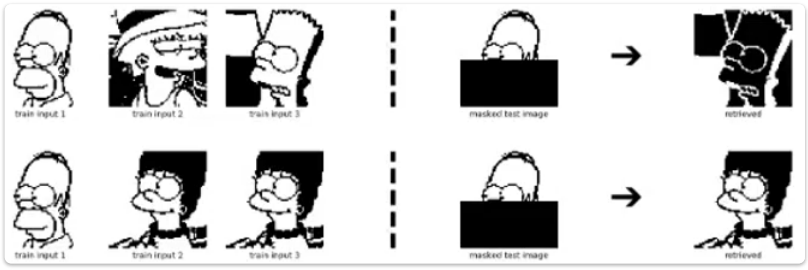
Il vettore che stiamo prendendo in considerazione è uno binario $\{\text{nero} = 1, \text{bianco} = 0\}$ per una dimensione di 128×128. Necessitiamo di una rete di Hopfield compresa da questo numero di neuroni. Con la regola di Hebb, costruiamo la matrice dei pesi con la cui, fornito input, fornirà output originale.

Se tra l'input e il minimo locale c'è una distanza significativa e quindi potenzialmente con altri minimi locali nel mezzo, quando la rete converge non è assolutamente detto che converga nel punto esatto ma potrà convergere in qualche immagine diversa. La rete la utilizziamo non tanto per ottenere l'output "pulito", quanto piuttosto per capire se l'input fornito è abbastanza simile.

3 sono le osservazioni che devono essere fatte.

1. Proprio come il nostro cervello, la rete cerca quei dettagli significativi abbastanza per poter riconoscere e arrivare all'immagine originale. Qual'ora i dettagli mancano, allora la rete non avrà un comportamento propriamente aspettato.
2. Costruire una rete che generi output di grandezza via via maggiore, vuole significare costruire una matrice che abbia un numero maggiore di minimi locali che faranno sì che per input abbastanza simili, la rete non restituisca proprio l'immagine aspettata.
3. Non vogliamo mai avere effettivamente dei vettori che siano mutualmente ortogonali tra di loro, perché questo significherebbe che nessuno di loro a qualcosa in comune. Le piccole variazioni del prodotto, sono le significatività.

Servirà un compromesso da realizzare per il numero di neuroni da memorizzare per la rete e il numero di neuroni della rete ($m \leq n$), dipeso dalle caratteristiche dei vettori che stiamo considerando. Il test set ci farà vedere il comportamento effettivo.



(Osserviamo l'immagine) Dati 3 vettori di input, usando la regola di Hebb otteniamo il vettore di pesi con minimi locali. Notare come nel primo caso evidentemente, il vettore con maggior numero di pixel neri, sia più vicino in termini di energia rispetto alla prima immagine delle 3 (Lisa e Bart contengono più pixel neri di Homer).
Anche nel secondo caso, l'immagine input viene associata con quella con maggior numero di pixel neri (Marge contiene più pixel neri di Homer).

(Osserviamo l'immagine) Notare per come questo nuovo vettore d'input fornito alla rete, l'immagine che si ottiene si distacca di moltissimo rispetto al risultato aspettato, siccome i minimi locali sono molto numerosi. L'evento finale, dato un'immagine mascherata, non è mai avvenuto: la rete sta "sognando".

Gli errori che otteniamo, va da precisare, non sono dipesi dal numero grande di vettori entranti la rete, quanto piuttosto dalla loro dipendenza lineare (notare il quantitativo di pixel neri in somiglianza per ciascun vettore).

30/03/2023