UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Corso di Laurea in Fisica

Tesi di Laurea

Reti neurali e modello di Hopfield

Relatore

Prof./Dr. Samir Suweis

Correlatore

Prof./Dr. Marco Baiesi

Laureando

Filippo Costa

Anno Accademico 2018/2019

Indice

Capitolo 1

Introduzione

1.1 La fine della Teoria

Pubblicato il 23 giugno 2008 da Chris Anderson su Wired.com, l'articolo *"The end of theory: the data deluge makes the scientific method obsolete"* propone, con una prevalente nota provocatoria, l'eliminazione del metodo scientifico per una nuova *epistemologia* basata sui dati, o meglio sui *Big Data*. Ma come può l'aumento dei dati in nostro possesso cambiare radicalmente il modo in cui conosciamo?

Internet è una rete globale che mette in connessione miliardi di abitanti nel nostro pianeta. Questa rete evolve, cresce e si ramifica in modo "auto-organizzato". Chi partecipa alla rete (si connette) scambia informazioni: messaggi mail, contenuti multimediali, consultazione di siti, etc. Tutto ciò significa scambiare e generare dati. Quanti dati scambiamo? Non è facile quantificarlo esattamente in quanto è un fenomeno in continua evoluzione, ma dal 2016 siamo entrati nella "Zetabyte era" (cfr. Cisco Systems 2017)

Questa crescita esponenziale di dati é ciò che va sotto il nome di *Big Data*.

La maggior parte di questa enorme quantità di dati viene utilizzata attraverso degli algoritmi noti come "reti neurali". Esse sono un esempio di ciò che più in generale viene chiamato apprendimento automatico (machine learning). Gli algoritmi di apprendimento automatico sono al cuore dell'intelligenza artificiale (IA), in quanto permettono alla macchina di "imparare" un certo compito, facendo sì che la sua performance migliori con il crescere della sua esperienza.

Ad esempio, supponiamo di avere un programma che osservi quali email vengono classificate dagli utenti come "spam" e quali no, e basandosi su questa informazione impari come meglio filtrare gli spam. Un algoritmo di machine learning è tale se la performance di tale filtraggio, ovverosia il numero di email che vengono classificate correttamente come spam, cresce con il numero di email dalle quali il programma impara.

Qual è la differenza rispetto a prima? In un approccio logico-deduttivo, ovvero nella creazione di un qualsivoglia modello, io debbo fornire preventivamente gli attributi caratterizzanti una email come "spam". Per esempio, potremmo dire che tipicamente gli spam presentano molti errori grammaticali o di lingua, oppure che il mittente non ha una email con nome e cognome. Sulla base di queste informazioni, la macchina dovrà quindi capire se la mail in arrivo è spam oppure no. Al contrario, in un approccio statistico, io non devo insegnare alla macchina niente su ciò che caratterizza gli spam. Semplicemente, "le mostro" milioni e milioni di email classificate come "spam" e altrettante come "non spam"; e la macchina, in una maniera che può essere o meno supervisionata, impara a distinguerle.

La conoscenza, a questo livello, è cambiata, passando da un sapere logico-deduttivo a uno unicamente statistico. Emblematica è la frase, riportata nell'articolo di Anderson, pronunciata da Peter Norvig,

direttore di ricerca di Google, alla O'Reilly Emerging Technology Conference: *"All models are wrong, and increasingly you can succeed without them."*

Dunque il nuovo paradigma consiste nell'affermare che per ottenere predizioni relative a un qualunque tipo di fenomeno, l'importante non è tanto sviluppare un modello che faccia emergere le relazioni causa-effetto, quanto avere a disposizione moltissimi dati esemplificativi del fenomeno da fornire ad una rete neurale.

La provocazione, procedendo verso la conclusione dell'articolo, diventa sempre più marcata, fino a giungere alla considerazione che, alla scala degli zetabytes, la correlazione tra eventi diventa sufficiente e l'approccio causale, in cui è il modello teorico a spiegare i fatti e ad attribuire loro senso, può essere abbandonato.

Un nuovo paradigma quindi, che porta sicuramente con se risultati straordinari che vanno dai progressi nella fisica delle particelle, fino ad "Amazon Alexa".

1.2 Black box e bias

Come anticipato, per gestire e sfruttare questa quantità immensa di dati, lo strumento più efficiente è la *rete neurale* o *neural network*, una struttura a uno o più livelli (*layers*) i cui costituenti fondamentali sono i nodi, elementi che si connettono tra loro per formare architetture più o meno complesse. Le connessioni tra nodi permettono la comunicazione tra di essi, e la condizione affinché questo sistema possa effettuare delle predizioni quanto più precise è che le connessioni siano adatte alla situazione in esame.

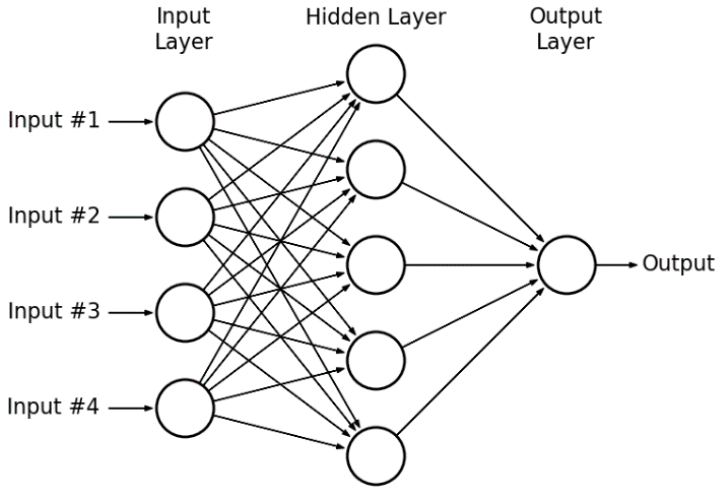

Figura 1.1: Neural network

L'utilizzo di un network di questo tipo è preceduto da una fase di *training*, in cui vengono forniti al sistema enormi quantità di dati di prova, con cui esso può "imparare" quali sono le connessioni migliori per la data situazione.

In questo modo, tuttavia, il programmatore del network non ha modo di capire quali siano le motivazioni per cui il sistema preferisce e rafforza alcune connessioni tra nodi, mentre invece ne trascura altre, e il compito del *data scientist* diventa, in questa situazione, quello di trovare empiricamente il numero di nodi e layers che fornisce i risultati migliori, trascurando però come e perchè si arrivi a quei risultati.

Abbiamo quindi a che fare con un modello a *black box*, in cui input e output sono collegati attraverso un rete di cui conosciamo la topologia e il meccanismo di attivazione dei nodi, ma non sappiamo il perchè dato un certo input vediamo proprio quell'output.

L'utilizzo delle reti neurali nei più svariati campi della conoscenza (dalla medicina, alla fisica delle particelle, da Google a Amazon), sebbene produca risultati mai raggiunti prima, presenta quindi molteplici problemi legati appunto alla non possibilità di comprensione dei risultati che le reti stesse producono. Uno, e tra i più rilevanti, problemi è quello noto come *bias*. Studi [1] mostrano infatti come le inferenze prodotte da questi sistemi contengano e amplifichino alcuni bias sociali/culturali e di altro tipo, come nell'esempio proposto in Figura 1.2[1].

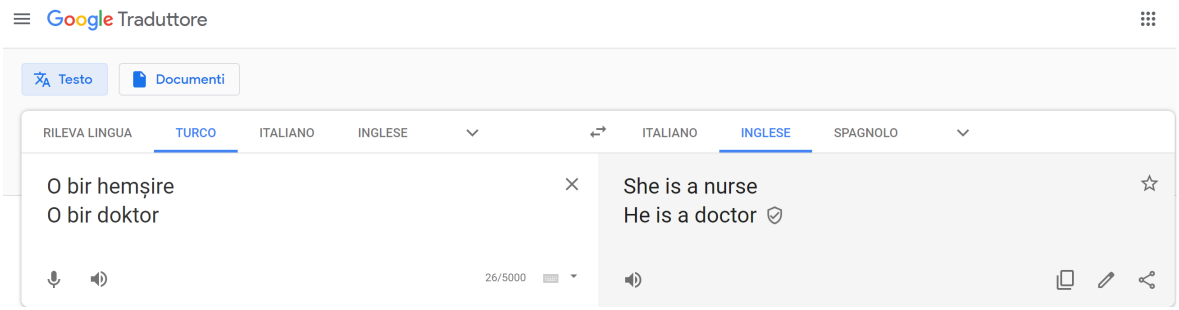

Figura 1.2: Bias sociali. Traduzione effettuata nell' agosto 2019

Sebbene in turco l'articolo "O" sia di genere neutro, l'algoritmo di Google associa la professione dell'infermiere/a al genere femminile, e quella di dottore/dottoressa a quello maschile. Questo perchè Google traduttore funziona con una rete neurale che usa come input miliardi di testi, e nei "dati" di input alcune professioni sono caratterizzate dalla predominanza di un certo sesso. Risulta evidente che questa polarizzazione tra attività classificate come maschili o femminili può provocare ingiustizie sociali, per esempio nel caso in cui questi networks con bias vengano utilizzati per una prima analisi di curriculum vitae inviati ad un'azienda, penalizzando alcuni candidati per il loro genere.

1.3 La fisica dei networks

Proprio per la vastità di applicazioni di queste reti neurali, e per la necessità di superare i problemi legati all'utilizzo delle reti come black box, negli ultimi anni è cresciuto moltissimo, o come vedremo tra breve è meglio dire tornato, l'interesse nella comunità dei fisici statistici per lo studio di queste reti neurali, cercando di comprendere più quantitativamente il processo di apprendimento della rete, e quindi di far emergere delle relazioni causali tra input e output.

Questi sistemi, infatti, composti da un grande numero di componenti elementari (i nodi) connesse tra loro e con una specifica dinamica di attivazione, possono essere visti come sistemi complessi con determinate proprietà emergenti.

L'utilizzo di termini quali "proprietà emergenti" e "sistemi complessi" è dovuto al fatto che conoscere la dinamica di un singolo nodo non permette automaticamente di dedurre il comportamento dell'intero sistema. L'analogia è quella con il cervello umano: conosciamo molto bene il funzionamento delle singole cellule che compongono il nostro cervello, i neuroni, ma sua la capacità computazionale non è deducibile da esse.

Uno dei grandi obiettivi della ricerca nell'ambito dell'intelligenza artificiale (ma anche nelle neuroscienze) risulta perciò quello di arrivare a conoscere la relazione tra la struttura topologica del sistema e le sue caratteristiche funzionali [2, 3].

In realtà questo obiettivo non è per nulla nuovo nella comunità dei fisici statistici. Negli anni 80, infatti, si era sviluppato un grande interesse per lo studio delle reti neurali artificiali, nel tentativo di sviluppare dei modelli semplificati per comprendere processi caratteristici del nostro cervello, quali la

[1] Dalla conferenza "Under the hood of Big Data and Artificial Embeddings", tenuta in data 9/11/18 dal Prof. Dott. L. Ballan al Dipartimento di Fisica e Astronomia dell'Università di Padova.

memoria. In questo ambito quindi si sono sviluppati svariati modelli che oggi possono essere di grande interesse, in quanto permettono di aiutare a capire il funzionamento di alcuni tipi di reti neurali.

In questa tesi, dunque, verrà presentato il funzionamento di un rete, detta "rete di Hopfield", presentata in un articolo [4] per la prima volta nel 1982 dal fisico John Hopfield. In essa, sebbene non sia presente la struttura a layers delle tipiche reti neurali, definite "deep", si evidenziano interessanti proprietà che legano topologia e funzione, in quanto la fase di *learning* é descritta da un algoritmo di cui conosciamo la soluzione in modo analitico. In altri termini, per la rete di Hopfield i pesi dei links della rete sono determinati in modo esatto a partire dall'input che si vuole "imparare".

Capitolo 2

Il network di Hopfield

2.1 Attractor Neural Network

La teoria alla base della struttura del network di Hopfield è denominata ANN (attractor neural network). Essa nasce con l'intento di fornire un modello della struttura cerebrale, focalizzandosi non sui singoli elementi, i neuroni, ma sull'insieme delle connessioni che questi creano tra loro. Nell' ANN, fisica, biologia e scienze cognitive interagiscono tra loro, garantendo ognuna un contributo essenziale. La prima fornisce il linguaggio matematico adatto, come vedremo meglio in seguito, la seconda le caratteristiche del modello, e la terza permette di confrontare questa teoria con i dati empirici derivanti dagli esperimenti.

Daniel J. Amit (1938-2007) costruisce per la prima volta una discussione organica sull'argomento [5], prendendo come riferimenti principali i lavori di John Hopfield e del neurospicologo Donald O. Hebb.

Il concetto di attrattore, caro alla fisica dei sistemi dinamici, e su cui ruota l'intero modello, è utilizzato per descrivere il funzionamento della memoria, il meccanismo che porta al riconoscimento di un elemento, sia esso un oggetto reale, un concetto o un' azione. Con attrattore, infatti, indichiamo un insieme (un punto, una curva o una varietà) verso il quale un sistema dinamico evolve dopo un tempo sufficientemente lungo, ed è per questo che nell'ambito dei modelli neurali può essere utilizzato in analogia con il concetto di memoria. Gli attrattori nascono dalle connessioni tra i neuroni, ed è infatti la struttura stessa del network a formare le memorie.

Secondo Amit, questo modello, per avere rilevanza dal punto di vista biologico e cognitivo, deve mostrare i seguenti caratteri:

1. *Associatività*

   Input diversi dello stesso oggetto devono portare alla stessa memoria.

2. *Comportamento emergente*

   Il network, da dinamiche locali, deve produrre proprietà emergenti che dipendono dalla tipologia di input.

3. *Plausibilità biologica*

   Il network non deve possedere caratteristiche che si discostano completamente dalla struttura biologica del cervello.

4. *Self-Organization*

   L'output del network non deve acquisire un significato attraverso agenti esterni, ma deve possederlo intrinsecamente.

5. *Potenziale di astrazione*

   Il network deve operare in modo simile con input diversi.

2.2 Basi biologiche

Prima di descrivere il modello è opportuna una descrizione del meccanismo fisiologico della comunicazione tra neuroni. Questo ci permetterà di mostrare come la rete neurale ideata da Hopfield soddisfi il principio della plausibilità biologica.

Il neurone, componente fondamentale del tessuto nervoso, è una cellula in cui possiamo identificare tre zone distinte. La prima è il *soma*, il corpo centrale in cui viene elaborato il segnale elettrico proveniente dagli altri neuroni e viene prodotto quello che invece verrà inviato, i *dendriti*, prolungamenti corti lungo i quali viene trasportato l'impulso nervoso proveniente dalle altre cellule, e l' *assone*, un prolungamento più lungo che trasporta invece il segnale, elaborato nel soma, verso altri neuroni.

I neuroni comunicano attraverso le sinapsi, zone tra l'assone del neurone pre-sinaptico, che trasmette il risultato dell'elaborazione del soma, e i dendriti del neurone post-sinaptico. Il segnale, arrivato alla coda dell'assone, causa la secrezione di neurotrasmettitori nella zona sinaptica che, giunti al neurone post-sinaptico, si legano ai recettori e portano alla formazione di una corrente ionica. Questi segnali possono eccitare o inibire il neurone, modificando la possibilità che questo produca un nuovo impulso da trasferire lungo l'assone. Infatti, se la somma degli stimoli che giungono al soma entro un breve periodo di tempo supera un valore di soglia, quel neurone produrrà un nuovo segnale elettrico. Il tempo che intercorre tra l'emissione di uno *spike* nel neurone pre-sinaptico e l'emissione di un altro impulso in quello post-sinaptico è di circa *1-2 ms*, cui segue un periodo di assoluta refrattarietà , in cui il neurone non può trasmettere nulla, e che ha sempre durata di pochi *ms*.

2.3 Basi fisiche

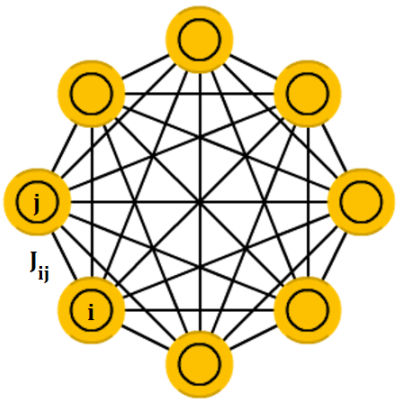

Figura 2.1: Hopfield network

Il network di Hopfield è costruito sulla base di queste conoscenze biologiche. La sua unità fondamentale è un *neurone computazionale* che assume valore ± 1. Ogni elemento è collegato con tutti gli altri neuroni del network attraverso sinapsi che portano un segnale di eccitazione o inibizione, e che nel modello si traducono in una matrice sinaptica caratterizzata da valori positivi (eccitazione) e negativi (inibizione). Questa matrice del network è detta anche matrice di adiacenza e la indicheremo con J, e con J_{ij}, l'entrata (i, j) della matrice, ovvero il valore della connessione sinaptica tra il neurone i e quello j.

Il modello richiede le seguenti ipotesi:

1. **La matrice sinaptica è simmetrica in quanto non consideriamo la direzione delle sinapsi**

$$J_{ij} = J_{ji} \tag{2.3.1}$$

2. **Il singolo neurone non ha memoria**

   il suo stato dipende solo dalla configurazione istantanea degli altri neuroni.

3. **I neuroni del network sono tutti dello stesso tipo**

   impiegano tutti lo stesso tempo per trasmettere l'informazione e per aggiornarsi.

4. **Il neurone i-esimo non si connette con se stesso**

$$J_{ii} = 0 \tag{2.3.2}$$

5. **La dinamica del network è asincrona**

I punti (2) e (5) richiedono una spiegazione dettagliata. Abbiamo detto che ogni neurone possiede uno stato ± 1, in cui il $+$ corrisponde ad un neurone attivo, il $-$ ad un neurone spento. Lo stato del neurone i-esimo viene determinato dagli input provenienti dagli altri N-1 neuroni (con N numero degli elementi del network), e la regola di aggiornamento dello stato S_i, se consideriamo una dinamica deterministica, é:

$$S_i(t+1) = sgn(\sum_j J_{ij} S_j(t) - \tau_i) \tag{2.3.3}$$

con τ_i il threshold per la produzione dello *spike*.

Lo stato all'istante $t+1$ del neurone S_i è determinato dal potenziale $U(t) = \sum_j J_{ij} S_j(t)$, generato dagli altri elementi del network all'istante t. Questo spiega l'assenza di memoria del neurone considerata nel punto 2.

Il punto 5 dell'elenco si spiega con il fatto che i neuroni del network non si aggiornano tutti nello stesso momento $t+1$ considerando la configurazione all'istante t, ma ogni elemento si aggiorna singolarmente, facendo evolvere il potenziale a step di durata $\delta t = \frac{1}{N}$.

L'eq. (2.3.3) corrisponde ad una dinamica deterministica, che tuttavia non rispecchia la reale situazione biologica, in quanto l'ambiente neurale presenta una situazione ben più ricca di quella modellizzata da Hopfield, e quindi la produzione del segnale da parte di un neurone è caratterizzata da variabili secondarie che nel network si traducono in una dinamica probabilistica, caratterizzata da un rumore più o meno intenso a cui assoceremo la variabile $\beta = T^{-1}$, con T la temperatura caratteristica del sistema.

In questo tipo di dinamica, la procedura di aggiornamento dello stato dell'elemento i-esimo é:

$$Pr(S_i) = \frac{1}{2}[1 + tanh(\beta h_i S_i)] \tag{2.3.4}$$

dove abbiamo definito

$$h_i = \frac{1}{2} \sum_{j, j \neq i}^{N} J_{ij} S_j. \tag{2.3.5}$$

Come si può notare, l'eq. (2.3.4) indica la probabilità del neurone i di essere nello stato S_i. Si noti che, nell'eq. (2.3.5), il threshold τ_i è stato posto a zero per semplificare la dinamica del sistema.

2.4 Dinamica e attrattori

2.4.1 Hamiltoniana del sistema

Diamo ora una derivazione dell' eq. (2.3.3) della sezione precedente. Possiamo associare una hamiltoniana al network di Hopfield che ha la seguente forma

$$H(\mathbf{S}) = -\frac{1}{2}\sum_{ij} J_{ij} S_i S_j, \tag{2.4.1}$$

dove $\mathbf{S}$ indica la configurazione completa di tutto il sistema. H($\mathbf{S}$) rappresenta quindi l'energia della rete in un certo stato $\mathbf{S}$. Introduciamo una dinamica dei neuroni basata sul principio di minimizzazione dell'energia. Pertanto i minimi $\mathbf{S}^\star$ di H($\mathbf{S}$) saranno degli *attrattori* della dinamica. Dato quindi un qualsiasi stato iniziale $\mathbf{S}_0$, esso evolverà verso l'attrattore più vicino, essendo quest' ultimo circondato da bacini di attrazione. Noi vogliamo che i minimi dell'energia corrispondano dunque alle memorie da recuperare. Indichiamo con il termine ξ_i^ν la componente i-esima della memoria ν.

La matrice sinaptica J deve quindi essere tale per cui determinati patterns $\boldsymbol{\xi^\nu}$ siano minimi di energia $\boldsymbol{S^\star}$. La procedura per la creazione di tale matrice prende il nome di *Regola di Hebb*. Se consideriamo $\nu = 1...P$ patterns $\boldsymbol{\xi^\nu}$, i termini J_{ij} della matrice sinaptica sono definiti come

$$J_{ij} = \frac{1}{N}\sum_{\nu=1}^{P} \xi_i^\nu \xi_j^\nu \tag{2.4.2}$$

Questa espressione soddisfa la minimizzazione dell'energia e spiega l'eq. (2.3.3). Infatti:

$$H(\mathbf{S}) = -\frac{1}{2}\sum_{ij} J_{ij} S_i S_j$$

$$= -\frac{1}{2}\sum_{ij}(\frac{1}{N}\sum_{\nu=1}^{P} \xi_i^\nu \xi_j^\nu) S_i S_j$$

$$= -\frac{1}{2N}\sum_{\nu=1}^{P}(\sum_i S_i \xi_i^\nu)(\sum_j S_j \xi_j^\nu) \tag{2.4.3}$$

L'Hamiltoniano quindi risulta minimizzato se lo stato del network diventa uguale a quello di uno specifico pattern, come indicato nell'eq. (2.3.3). L' eq. (2.4.3) mostra tuttavia altre importanti caratteristiche. L'hamiltoniano, infatti, risulta minimizzato anche se $\mathbf{S} = \text{-}\boldsymbol{\xi^\nu}$, ovvero nel caso, definito *reversed state*, in cui ogni elemento del network è l'opposto di quello definito nella memoria. Questo tipo di minimo, non essendo uno degli $\boldsymbol{\xi^\nu}$ definiti inizialmente, viene definito *minimo spurio*. Esistono inoltre altri tipi di minimi spuri. Una configurazione, infatti, può risultare un minimo di energia se

$$\mathbf{S} = c_1\, \boldsymbol{\xi^1} + c_2\, \boldsymbol{\xi^2} + ... + c_P\, \boldsymbol{\xi^P} \tag{2.4.4}$$

Questo tipo di minimo, combinazione lineare dei patterns $\boldsymbol{\xi^\nu}$, viene definito *minimo misto*.

Esiste infine un'altra tipologia di minimi spuri, definiti *stati di spin glass*, che, per essere spiegati, necessitano del concetto di *frustrazione*, esposto brevemente nella prossima sezione.

2.4.2 Frustrazione di un sistema fisico

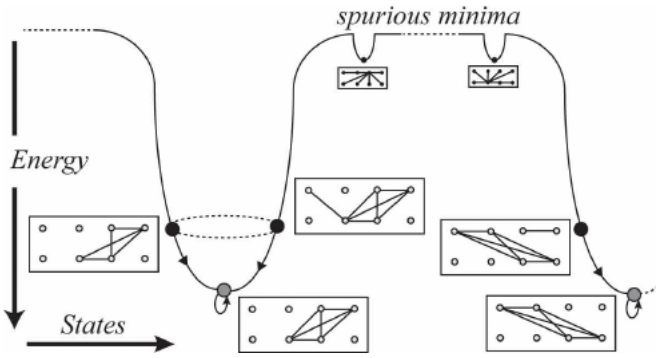

Figura 2.2: Energy landscape in un network di Hopfield

Come detto precedentemente, una memoria consiste in un minimo dell'energia. Tuttavia, aumentando i patterns memorizzati nel network, la configurazione energetica diventa sempre più complessa, arrivando alla formazione di minimi spuri, tra cui gli *stati di spin glass*. Questi ultimi si creano per un fenomeno detto *frustrazione*, ovvero l'impossibilità di un sistema di ottimizzare simultaneamente tutte le sue interazioni.

Il nome deriva dal fatto che la situazione che si viene a configurare è fisicamente simile a quella degli *spin glasses* [6], reticoli di spin in cui ogni elemento comunica con quelli a lui vicini, e la cui dinamica è legata alla minimizzazione dell'energia:

$$E(\boldsymbol{X}) = -\sum_{<kl>} J_{kl} S_k S_l - h \sum_k S_k \tag{2.4.5}$$

con $\boldsymbol{X}$ la configurazione del sistema, J la matrice di adiacenza, S_k lo stato di spin dell'elemento k-esimo e h il campo magnetico esterno.

Facciamo ora un semplice esempio di frustrazione e consideriamo, in 2D, quattro elementi di spin disposti come in figura, e cerchiamo la configurazione che minimizza l'energia.

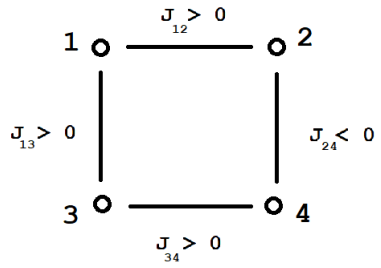

Figura 2.3: Esempio di frustrazione

Dovremmo scegliere gli stati degli spin 1,2,3,4 dello stesso segno ma, a causa del segno dei J_{kl}, per minimizzare l'energia lo spin 4 dovrebbe avere segno opposto rispetto a quelli di 2 e 3. Abbiamo quindi l'impossibilità di ottimizzare simultaneamente entrambe le condizioni, e il risultato di ciò è la creazione di due stati $\boldsymbol{X_1}$ e $\boldsymbol{X_2}$ nell'eq.(2.4.5), entrambi minimi di energia. Poiché la dinamica del modello di Hopfield definita precedentemente è simile a quella degli spin glasses, troveremo anche in questo caso minimi generati dalla frustrazione

Considerando poi che, nel nostro caso, la matrice sinaptica J ha solitamente una dimensione elevata, possiamo capire che nel sistema esistono una quantità considerevole di interazioni non ottimizzate, e che la presenza di minimi spuri dati dalla frustrazione è un fatto non trascurabile.

2.4.3 Capacità di immagazzinamento

Sono necessarie ulteriori considerazioni sull'efficacia dell'algoritmo di learning, ovvero sulla sua capacità di recuperare una memoria. Aumentando il numero di patterns immagazzinati, infatti, questa efficacia diminuisce, ed oltre una certa soglia il sistema cessa di funzionare. Analizziamo ora la condizione per cui una memoria coincida con un attrattore, che definiamo come *stabilità* di un determinato pattern $\boldsymbol{\xi^\nu}$. Essa può essere scritta [7] come

$$S_i = sgn(h_i^\nu) = \xi_i^\nu \tag{2.4.6}$$

dove, usando l'eq. (2.4.2), abbiamo che

$$h_i^\nu = \sum_j^N J_{ij}\xi_j^\nu \;=$$

$$= \frac{1}{N}\sum_{j=1}^{N}\xi_i^\nu\xi_j^\nu\xi_j^\nu + \frac{1}{N}\sum_{j=1}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{P}\xi_i^\mu\xi_j^\mu\xi_j^\nu \;=$$

$$= \xi_i^\nu + \frac{1}{N}\sum_{j=1}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{P}\xi_i^\mu\xi_j^\mu\xi_j^\nu \tag{2.4.7}$$

Il secondo termine di questa espressione è detto *crosstalk term* e contiene la sovrapposizione tra il pattern in esame e tutti i rimanenti.

Il nostro obiettivo è ottenere $S_i = \xi_i^\nu$, ma poichè $S_i = \pm 1$, come anche $\xi_i^\nu = \pm 1$, allora $S_i = \xi_i^\nu$ se e solo se $S_i\xi_i^\nu > 0$. Considerando poi l'eq. (2.4.6), abbiamo che $S_i\xi_i^\nu > 0$ è equivalente a $sgn(h_i^\nu)\xi_i^\nu > 0$. Ma se quest'ultima si verifica, allora è verificata anche

$$h_i^\nu\xi_i^\nu > 0. \tag{2.4.8}$$

Dato che la condizione di stabilità (2.4.6) è equivalente alla condizione (2.4.8), possiamo definire la quantità :

$$C_i^\nu = -\xi_i^\nu\frac{1}{N}\sum_{j=1}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{P}\xi_i^\mu\xi_j^\mu\xi_j^\nu. \tag{2.4.9}$$

In questo modo

$$h_i^\nu\xi_i^\nu = \xi_i^\nu\xi_i^\nu - C_i^\nu = 1 - C_i^\nu. \tag{2.4.10}$$

Se quindi $C_i^\nu < 0$ il *crosstalk term* ha lo stesso segno di ξ_i^ν e quindi la stabilità del pattern ν non viene alterata. Manteniamo la stabilità anche nel caso in cui $0 < C_i^\nu < 1$. Tuttavia, nel caso in cui $C_i^\nu > 1$ la condizione di stabilità viene meno e il sistema si allontanerà da questa memoria.

La probabilità per l'intero pattern $\boldsymbol{\xi^\nu}$ di essere stabile risulta pari a

$$Pr(\boldsymbol{\xi^\nu h^\nu} > 0) \approx 1 - \sqrt{\frac{\alpha}{2\pi}}exp(-\frac{1}{2\alpha}) \tag{2.4.11}$$

utilizzando la variabile $\alpha = P/N$, con P numero di patterns ed N numero di nodi.

Si può dimostrare inoltre che il valore $\alpha^\star$ di soglia, oltre il quale il network diventa inefficiente è:

$$\alpha^\star \approx 0.138 \tag{2.4.12}$$

Per la dimostrazione completa di questi ultimi due fatti si rimanda a [8]

2.5 Simulazioni

In questa sezione vengono mostrati i risultati delle simulazioni eseguite con un network di Hopfield al variare di N, numero di neuroni del sistema, e P, numero di patterns memorizzati nella matrice sinaptica J. Analizzeremo dei grafici in cui viene mostrato quanto il sistema si avvicina, nel corso del tempo, ai vari patterns $\boldsymbol{\xi^\nu}$, quantificando il suo *overlap*, indicato con m^ν, ovvero la somiglianza con le configurazioni salvate in memoria, e definito come:

$$m^\nu = \frac{1}{N}\sum_{i=1}^{N} \xi_i^\nu S_i \tag{2.5.1}$$

Il tempo è quantificato in cicli, cioè in numero di aggiornamenti completi del network.

2.5.1 Simulazioni a più patterns

Consideriamo configurazioni con un numero di nodi pari a 100, un numero di patterns pari a 5,10 e 20, e un overlap iniziale nullo. In questo modo osserviamo come il network si comporta se il landscape di energia si popola di una quantità considerevole di minimi spuri.

Mostriamo ora i risultati cui siamo giunti.

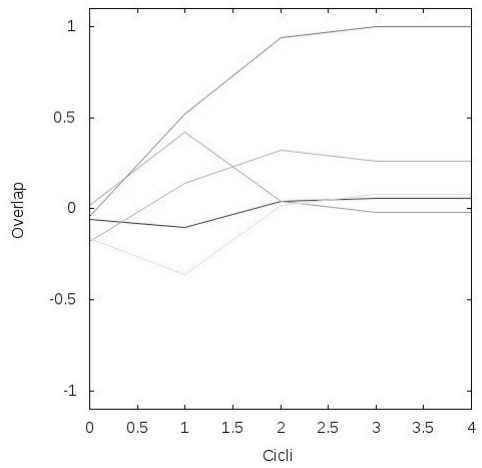

Figura 2.4: Risultato di una simulazione a 100 nodi, 5 patterns, ognuno rappresentato da una linea, e valore nullo di overlap iniziale. Nel caso presentato il network ha ricordato con successo uno dei patterns inseriti in memoria, giungendo ad un overlap con esso pari a 1. Abbiamo ripetuto 1000 simulazioni in questa configurazione di parametri, e questo tipo di risultato si ottiene nel 40% dei casi.

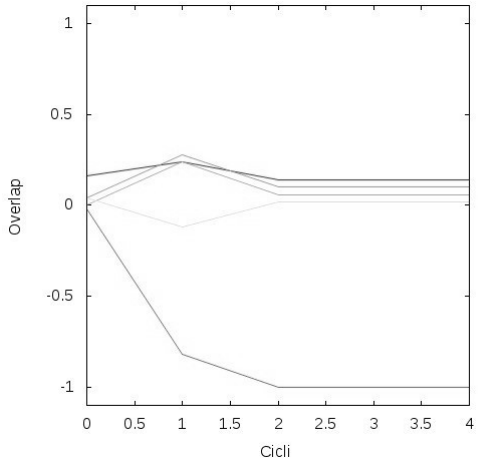

Figura 2.5: Risultato di una simulazione a 100 nodi, 5 patterns, ognuno rappresentato da una linea, e valore nullo di overlap iniziale. Il network ha concluso la sua evoluzione in un *reversed state* di un pattern della memoria (overlap -1), e quindi in un minimo spurio.

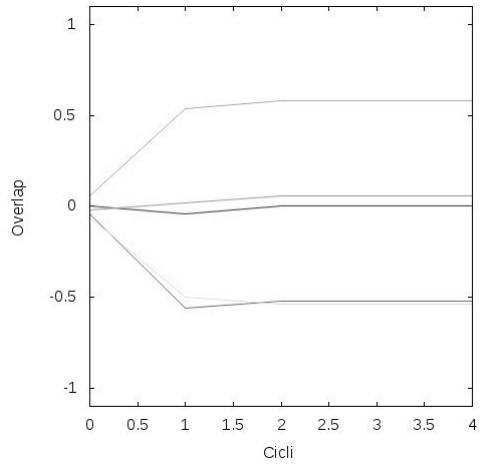

Figura 2.6: Risultato di una simulazione a 100 nodi, 5 patterns, ognuno rappresentato da una linea, e valore nullo di overlap iniziale. Dato che l'evoluzione del network si è conclusa senza il raggiungimento di una memoria, questo caso mostra una configurazione di minimo misto oppure di uno stato di spin glass.

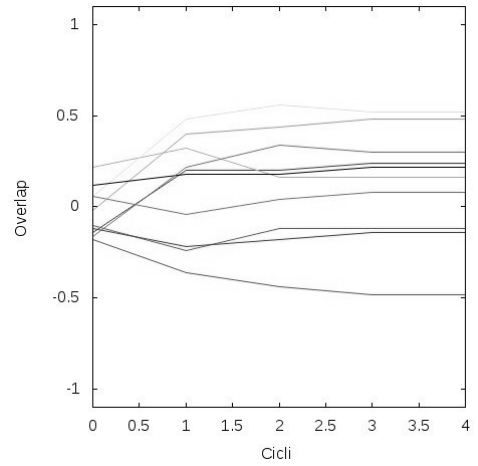

Figura 2.7: Risultato di una simulazione a 100 nodi, 10 patterns, ognuno rappresentato da una linea, e valore nullo di overlap iniziale. Anche in questo caso abbiamo eseguito 1000 simulazioni con questa configurazione di parametri, e la probabilità di ricordare correttamente un pattern della memoria scende al 15%. In figura è mostrata una delle evoluzioni più comuni.

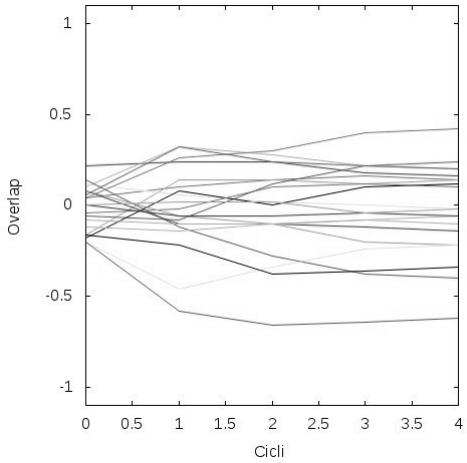

Figura 2.8: Risultato di una simulazione a 100 nodi, 20 patterns, ognuno rappresentato da una linea, e valore nullo di overlap iniziale. Eseguendo 1000 simulazioni con questa configurazione di parametri, la probabilità che l'evoluzione del network porti questo ad un pattern della memoria si avvicina allo 0%, e in figura viene mostrata la situazione più comune.

2.5.2 Simulazioni a singolo pattern

Esaminiamo ora l'evoluzione del network di Hopfield quando nella memoria è presente un solo pattern. Vengono eseguite simulazioni su due configurazioni differenti, a 20 e a 400 nodi, entrambe con un overlap iniziale m(0) = 0. In entrambi i casi, dopo 1000 simulazioni, si trova una probabilità di riconoscimento pari a circa 50%, mostrando quindi un' indipendenza dal numero totale di nodi.

Osservando poi l'andamento dell'overlap nelle singole simulazioni, si trovano solo due evoluzioni possibili. O il sistema riesce a ricordare correttamente il pattern in memoria, oppure si colloca nel *reversed state* di quest'ultimo.

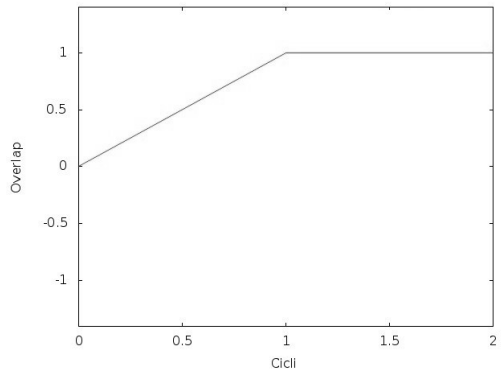

Figura 2.9: Risultato di una simulazione a 400 nodi, 1 pattern e valore nullo di overlap iniziale. In questa evoluzione il network è riuscito a ricordare (overlap 1) il pattern salvato in memoria.

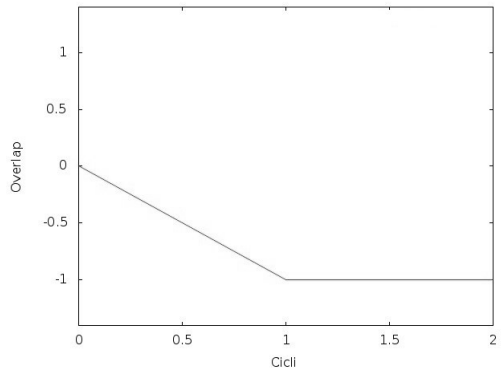

Figura 2.10: Risultato di una simulazione a 400 nodi, 1 pattern e valore nullo di overlap iniziale. Il network non è riuscito a ricordare il pattern salvato in memoria, ed ha concluso la sua evoluzione nell'unico minimo spurio possibile, ovvero nel *reversed state* (overlap -1) del pattern in memoria.

Notiamo inoltre che diminuendo il numero di connessioni all'interno del network, con modalità che verranno precisate nel capitolo seguente, le due configurazioni con overlap iniziale nullo producono gli stessi risultati di un network completamente connesso, e cioè una probabilità di riconoscimento sempre fissa al 50% .

Capitolo 3

Relazione topologia - funzione

Lo scopo di questo capitolo è quello di mostrare come la struttura del network di Hopfield influenzi il suo funzionamento. Lo studio che segue si concentra, in particolare, sul rapporto tra l'efficienza del network, ovvero la percentuale di volte in cui esso riesce a ricordare uno specifico pattern, e la sua *sparsity*. Con questo termine, nell'ambito della teoria dei grafi, si intende quanto il sistema sia connesso, considerando una quantità definita *connectance*, la cui formula è:

$$C = \frac{L}{\frac{N*(N-1)}{2}} \tag{3.1}$$

dove a numeratore abbiamo il numero effettivo L di links del sistema, mentre a denominatore abbiamo il numero di connessioni che il network può potenzialmente avere, non considerando connessioni di un nodo con se stesso, e indicando con N il numero di nodi del network.

3.1 Costruzione della simulazione

L'analisi delle relazioni tra la topologia e l'efficienza del network ha come base una serie di simulazioni. Il network di Hopfield utilizzato possiede parametri costanti, non modificabili runtime, e una variabile determinata stocasticamente, il cui valore varia durante l'esecuzione del programma.

Le costanti del modello sono N, il numero di neuroni computazionali utilizzati, P, il numero di patterns memorizzati, m(0), l'overlap iniziale tra lo stato del sistema e il primo pattern generato, D, la porzione di matrice sinaptica J che verrà cancellata prima dell'avvio della simulazione, equivalente al valore (1-C), con C il valore della connectance definita precedentemente.

L' elemento variabile è invece T, la temperatura cui è soggetto il sistema, indice della rumorosità del processo.

I patterns di memoria e lo stato iniziale del network sono generati casualmente e, a partire dai valori di P, viene creata la matrice J, che provoca l'update del network durante la simulazione. I valori degli elementi della matrice, ovvero i J_{ij}, si ottengono tramite l'eq. (2.4.2).

La temperatura è campionata da una distribuzione gaussiana $p(T)$, il cui scarto quadratico σ è scelto in modo tale da avere una configurazione biologicamente plausibile. La curva scelta per il campionamento ha infatti distanza 1σ da $T = 0.461$ e 2σ da $T = 1$. Il primo di questi valori è il livello sotto il quale gli stati spuri cominciano a essere progressivamente stabili, mentre il secondo rappresenta la temperatura critica, oltre la quale il sistema diventa ergodico, ostacolando quindi la produzione di minimi locali stabili di energia [5]. Sulla base di queste considerazioni, i parametri per $p(T) = N(\mu, \sigma)$, dove N è la distribuzione normale con media μ e scarto quadratico medio σ, sono $\mu = 0.640$ e $\sigma = 0.18$.

Queste considerazioni sulla temperatura valgono nel limite termodinamico $N \to \infty$, mentre per valori di $N << \infty$ la temperatura critica T_c assume valori più elevati. Dunque, nelle simulazioni seguenti, effettuate con un massimo di 200 nodi, il campionamento delle temperature con la distribuzione scelta serve unicamente per passare da una dinamica deterministica a una stocastica.

Prima dell'inizio della simulazione viene cancellata, randomicamente, una porzione di matrice J. Questo avviene ponendo a zero $D * N^2$ termini della matrice sinaptica.

La dinamica si suddivide in cicli. All'inizio di ogni ciclo viene campionata una temperatura, che si manterrà costante fino al termine di questo. Dopodichè avviene un update completo di tutto il network, con aggiornamento asincrono[1]. Se dopo 4 cicli completi il network non raggiunge lo stato del primo pattern, la simulazione si classifica come un insuccesso. Abbiamo scelto di analizzare un unico overlap, quello con il primo pattern, e abbiamo quindi inizializzato un valore $m(0) \neq 0$ solo con quest'ultimo.

La simulazione si considera avvenuta con successo se il network raggiunge lo stato corrispondente al primo pattern entro una precisione del 99.5%.

La dinamica dell'aggiornamento è stocastica, e il cambio di stato del singolo neurone segue la legge data dall'eq. (2.3.4)

L'idea per questo tipo di simulazione nasce dalla lettura di un articolo del 1993, in cui il deterioramento della memoria nel morbo di Alzheimer viene modellizzato da una variante del network di Hopfield in cui il numero di connessioni tra nodi diminuisce progressivamente [9].

L'intero progetto è disponibile su https://github.com/filippocosta/Hopfield-model-efficiency.

3.2 Risultati

Per poter analizzare in dettaglio il rapporto tra l'efficienza del network e la sua sparsity, abbiamo testato il nostro modello di rete partendo da diverse configurazioni iniziali.

Le combinazioni di parametri utilizzate sono:

- N = 200, P = 5, m(0) = 0.8
- N = 200, P = 10, m(0) = 0.8
- N = 200, P = 20, m(0) = 0.8
- N = 200, P = 5, m(0) = 0.6
- N = 200, P = 10, m(0) = 0.6
- N = 200, P = 20, m(0) = 0.6
- N = 100, P = 5, m(0) = 0.8
- N = 100, P = 10, m(0) = 0.8

Le analisi svolte nelle sezioni seguenti si basano sulla relazione tra D = (1-C), la porzione di matrice J cancellata, e R, il rate di successi su 1000 simulazioni svolte.

Segue il grafico con i dati raccolti per le simulazioni con N = 200 e m(0) = 0.8. [2]

[1] vedi sezione 2.3.

[2] Gli errori relativi ai dati raccolti verranno discussi successivamente.

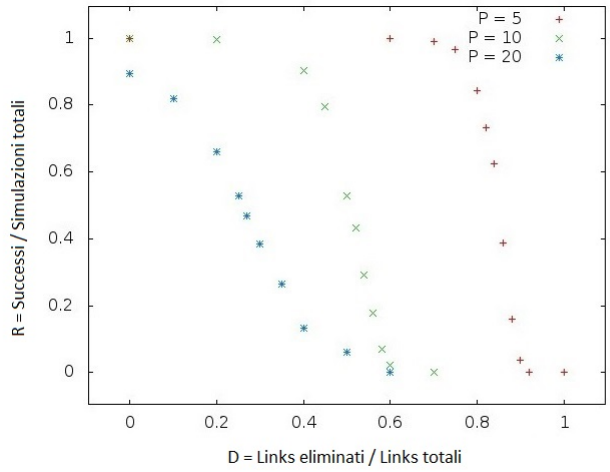

Figura 3.1: Risultati delle simulazioni con una configurazione a 200 nodi, valore di overlap con il primo pattern m(0) = 0.8, e numero P di patterns variabile.

Le simulazioni mostrano un andamento tipico di una funzione logistica. Preliminarmente, notiamo che il numero di patterns P salvati nella memoria influisce sulla posizione del punto di dimezzamento del rendimento, e sulla sua stessa velocità di discesa.

Prima di procedere con l'analisi delle curve, mostriamo un grafico con l'andamento di tutti i set di dati ottenuti dalle simulazioni.

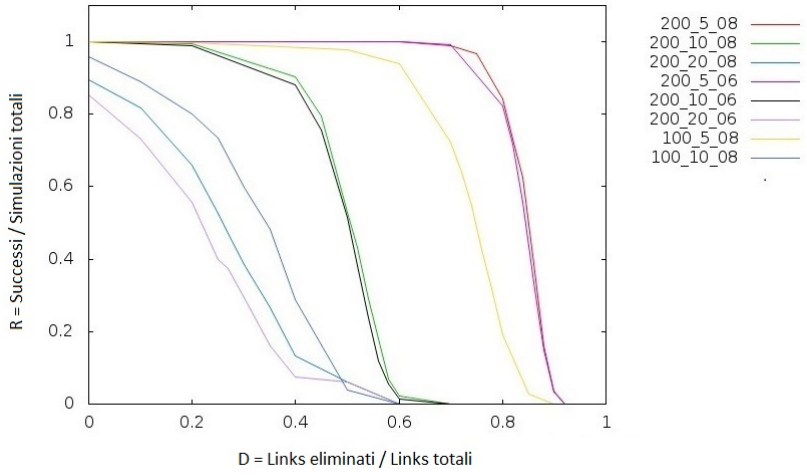

Figura 3.2: Andamento delle simulazioni. La legenda ha la struttura $N\_P\_m(0)$, con N numero di nodi, P numero di patterns nella memoria e m(0) valore di overlap iniziale con il primo pattern.

3.3 Analisi delle curve

Per analizzare e generalizzare i risultati delle simulazioni, vogliamo trovare i parametri delle curve logistiche che meglio descrivono i dati ottenuti.

La generica equazione di una funzione logistica con andamento decrescente ha la struttura:

$$R(D) = a\frac{1 + b\ e^{-\frac{(1-D)}{\alpha}}}{1 + d\ e^{-\frac{(1-D)}{\alpha}}} \tag{3.3.1}$$

con a, b, d, α da determinare.

Disponiamo inoltre dei seguenti risultati empirici:

1. R varia nel range $[0, 1]$;
2. A N fissato, aumentando P, la velocità di decrescita aumenta e il punto di dimezzamento del rendimento si trova ad un valore minore di D;
3. La variazione di $m(0)$ influisce maggiormente se il rapporto P/N aumenta;
4. L'aumento di $m(0)$ provoca un ritardo nella decrescita della funzione;
5. Quando $D = 1$, R deve necessariamente essere pari a 0;
6. Quando $D = 0$, R si avvicina all'unità , diminuendo all'aumentare di P;
7. Il rate di decrescita dipende dal rapporto P/N;
8. Il rate di decrescita non dipende da $m(0)$.

Le ultime due affermazioni derivano dall'analisi qualitativa del grafico. L'andamento di decrescita delle simulazioni a 100 nodi, 5 patterns e m(0) = 0.8 è simile a quello trovato nelle simulazioni a 200 nodi, 10 patterns e m(0) = 0.8 e di quelle a 200 nodi, 10 patterns e m(0) = 0.6. Lo stesso si può affermare per le simulazioni con 100 nodi, 10 patterns e m(0) = 0.8, quelle con 200 nodi, 20 patterns e m(0) = 0.8 e quelle con 200 nodi, 20 patterns e m(0) =0.6.

Sulla base di queste osservazioni, possiamo porre:

- $a = 1$, in modo da assegnare un valore unitario all'efficienza massima;
- $b = -1$, poichè l'aumento di D genera una diminuzione del rendimento e, quando D = 1, l'efficienza R deve essere pari a zero;
- $\alpha \approx P/N$, poichè questo parametro regola il rate di decrescita della curva, e per i punti 7 e 8, sappiamo che esso è proporzionale a P/N.

Infine, ridefiniamo il parametro di d con $d = e^q$, in modo da semplificare le considerazioni successive.

Allora possiamo riscrivere l'eq. (3.3.1) come:

$$R(D) = \frac{1 - e^{-\frac{(1-D)}{\alpha}}}{1 + e^{-\frac{(1-q-D)}{\alpha}}} \tag{3.3.2}$$

La ridefinizione dell' eq. (3.3.1) permette di esaminare con maggiore semplicità il grado di accordo tra la curva e i dati sperimentali.

3.3.1 Stima della deviazione standard di R

Per eseguire un corretto fit dei dati simulati, è necessario determinare la deviazione standard associata al termine R dell'eq. (3.3.2).

Le simulazioni forniscono, per una data configurazione di parametri e per un certo valore D, un valore R che corrisponde alla probabilità che la rete, attraverso il learning, raggiunga l'attrattore desiderato, ovvero la memoria. R è definito come $R = \frac{s}{n}$, con s numero di successi delle simulazioni, e con n numero totale di prove effettuate. Considerando il numero elevato di prove, mille per ogni punto, possiamo quindi considerare R come una probabilità.

Poichè i risultati delle singole simulazioni sono tra loro indipendenti, e che queste si classificano o come un successo o come un insuccesso, l'esito di queste costituisce un processo di Bernoulli, in cui ogni prova ha una probabilità R di successo, e una probabilità 1-R di insuccesso. Al valore di efficienza

R andrà quindi associata la deviazione standard $\sigma_R = \sigma_s \frac{\partial R}{\partial s}$, dove a σ_s è associata la deviazione standard di una distribuzione di Bernoulli con probabilità R di successo e 1-R di insuccesso, ovvero $\sigma_s = \sqrt{R(1-R)}$. Dunque avremo $\sigma_R = \sqrt{\frac{R(1-R)}{n}}$.

Tuttavia, dato l'elevato numero di simulazioni effettuate per ogni punto e data la forma stessa dell'errore, i punti con $R \approx 0$ possiedono una deviazione standard sottostimata. A questa considerazione si giunge eseguendo più set di simulazioni sui punti (D,R) con $R \approx 0$. Da queste simulazioni ripetute risulta una fluttuazione di R maggiore di quella stimata. É stato ritenuto opportuno quindi maggiorare la deviazione standard di R con il suo valore massimo $\sigma_R = \frac{1}{2\sqrt{n}}$, che si ottiene per $R = 0.5$.

Esiste un' ulteriore fluttuazione dei valori derivante dalla dinamica stocastica della simulazione, la quale dipende dalla temperatura, campionata all'interno di una distribuzione gaussiana con una $\sigma_T = 0.18$. Questo tipo di volatilità, tuttavia, risulta trascurabile rispetto a quello indicato precedentemente[3], e quindi, nelle analisi successive, considereremo unicamente $\sigma_R = \frac{1}{2\sqrt{n}}$.

3.3.2 Bontà dei fit

Analizziamo ora il grado di accordo tra i dati delle simulazioni e l'eq. (3.3.2). I fit sono stati svolti per i set di dati mostrati in Figura 3.2, ad eccezione di quelli con $m(0) = 0.6$, considerando che l'overlap iniziale, in prima approssimazione, rappresenta un fattore trascurabile, soprattutto se $\alpha << 1$.

Interpolando tutti i punti con l'eq. (3.3.2), per i set di dati simulati, la curva ottenuta non superava i test statistici di χ^2.

Tuttavia, eseguendo due interpolazioni differenti, una nell'intervallo $0 < D < D^\star$, e una in $D^\star < D < 1$, per un certo valore di $D^\star$, i risultati mostravano un netto miglioramento nel grado di accordo con i punti.

Si è notato poi che $D^\star$ assumeva un valore circa pari a $1 - q$, dove q si riferisce al parametro dell'eq. (3.3.2) che dà informazioni sul punto di dimezzamento di R.

La variabile α, indice della velocità di decrescita del rendimento, assume valori differenti nelle due interpolazioni. Per tutti i set di simulazioni, il valore ottenuto nell'intervallo $0 < D < D^\star$, denominato α_{sup}, risulta sempre doppio rispetto a quello dell'intervallo $D^\star < D < 1$, denominato α_{inf}. Nelle analisi successive verrà mostrato il grado di accordo tra α_{sup} e $2\alpha_{inf}$.

La funzione utilizzata per l'interpolazione dei dati simulati risulta quindi

$$R(D) = \begin{cases} \dfrac{1 - e^{-\frac{(1-D)}{\alpha_{sup}}}}{1 + e^{-\frac{(1-q-D)}{\alpha_{sup}}}} & se\ 0 < D < D^\star \approx 1 - q \\ \\ \dfrac{1 - e^{-\frac{(1-D)}{\alpha_{inf}}}}{1 + e^{-\frac{(1-q-D)}{\alpha_{inf}}}} & se\ 1 - q \approx D^\star < D < 1 \end{cases} \tag{3.3.3}$$

con R la probabilità di successo del network, D il numero di links eliminati su quello totale di connessioni possibili, α la velocità di decrescita di R e con $D^\star \approx 1 - q$ il punto di discontinuità della funzione.

L'interpolazione dei punti con l'eq. (3.3.3) è stata ottenuta eseguendo un primo fit nell'intervallo $0.5 < R < 1$, ottenendo un primo valore di q, seguito poi dal fit completo, utilizzando come stima di $D^\star$ il valore $1 - q$. Per ogni set di dati simulati, il procedimento è stato iterato variando l'intervallo

[3] La temperatura critica T_c assume valore 1 solo nel limite termodinamico di $N \to \infty$, mentre per un numero di nodi minore il valore di criticità risulta più elevato, e quindi l'errore relativo alla temperatura risulta trascurabile rispetto agli altri fattori.

per il fit superiore, in modo da ottenere il valore di q che permettesse di minimizzare il χ^2 delle due interpolazioni.

I risultati ottenuti mostrano che, in ogni configurazione di parametri, il valore $1-q$ ottenuto dal fit superiore è compatibile con quello del fit inferiore.

Per quanto riguarda invece il parametro α, possiamo notare come α_{sup} sia sempre compatibile con $2\alpha_{inf}$, e che l'ipotesi $\alpha \approx P/N$ si verifica con maggiore accordo per i set con P/N più elevato (vedi Figura 3.3 rispetto a Figura 3.5).

Seguono i risultati ottenuti dai fit e dai test statistici[4]

P/N	0.025
α_{sup}	0.034 ± 0.006
$(1-q)_{sup}$	0.857 ± 0.007
$\chi^2_{sup}(g.d.l = 5)$	$0.3 \rightarrow 99.5\%$
$2\alpha_{inf}$	0.031 ± 0.005
$(1-q)_{inf}$	0.853 ± 0.003
$\chi^2_{inf}(g.d.l = 4)$	$0.4 \rightarrow 97.5\%$
Compatibilità α	$0.4 \rightarrow$ ottima
Compatibilità $1-q$	$0.5 \rightarrow$ ottima

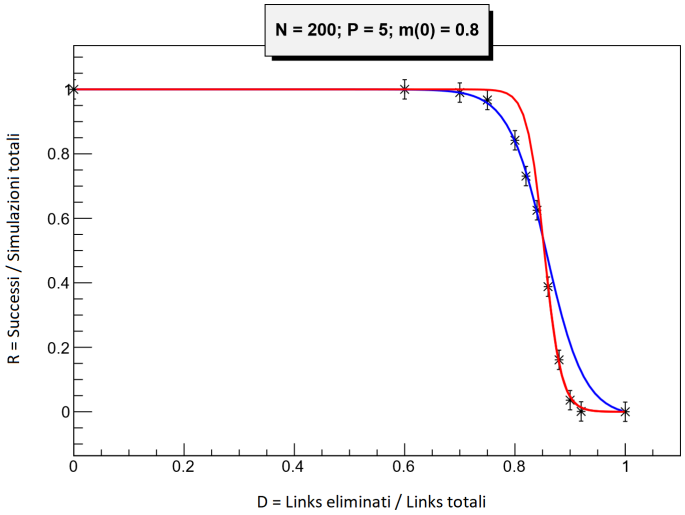

Figura 3.3: I risultati portano a confermare l'ipotesi di una velocità di decrescita doppia nella parte finale della simulazione. Tuttavia i valori di α, pur essendo compatibili tra loro, sono più distanti dei casi successivi dall' ipotesi $\alpha \approx P/N$. *In blu*: interpolazione dei dati in $0 < D < D^\star \approx 1-q$. *In rosso*: interpolazione dei dati in $1-q \approx D^\star < x < 1$.

P/N	0.05
α_{sup}	0.044 ± 0.006
$(1-q)_{sup}$	0.506 ± 0.006
$\chi^2_{sup}(g.d.l = 5)$	$0.42 \rightarrow 99.0\%$
$2\alpha_{inf}$	0.056 ± 0.006
$(1-q)_{inf}$	0.513 ± 0.004
$\chi^2_{inf}(g.d.l = 5)$	$1.4 \rightarrow 90.0\%$
Compatibilità α	$1.4 \rightarrow$ buona
Compatibilità $1-q$	$0.97 \rightarrow$ ottima

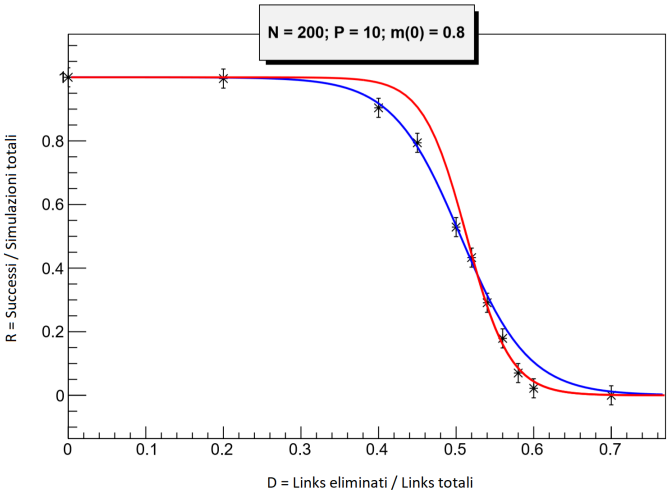

Figura 3.4: Questa simulazione supporta la tesi $\alpha \approx P/N$, e la composizione delle due logistiche nei modi descritti in precedenza è compatibile con i risultati ottenuti. *In blu*: interpolazione dei dati in $0 < D < D^\star \approx 1-q$. *In rosso*: interpolazione dei dati in $1-q \approx D^\star < x < 1$.

[4]la percentuale accanto al valore del test χ^2 mostra a che livello di confidenza si può rigettare l'ipotesi nulla. g.d.l. = gradi di libertà.

P/N	0.1
α_{sup}	0.12 ± 0.02
$(1-q)_{sup}$	0.27 ± 0.01
$\chi^2_{sup}(g.d.l = 2)$	$0.7 \rightarrow c.l < 90.0\%$
$2\alpha_{inf}$	0.15 ± 0.02
$(1-q)_{inf}$	0.262 ± 0.008
$\chi^2_{inf}(g.d.l = 5)$	$1.03 \rightarrow 95.0\%$
Compatibilità α	$1.06 \rightarrow$ buona
Compatibilità $1-q$	$0.6 \rightarrow$ ottima

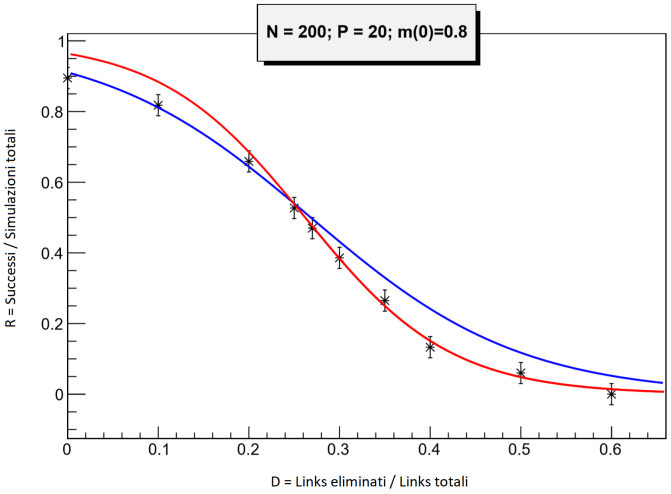

Figura 3.5: la tesi iniziale è convalidata anche nel caso di rapporti P/N più vicini al valore critico di $\alpha^\star \approx 0.138$, nell'ipotesi $\alpha \approx P/N$. *In blu*: interpolazione dei dati in $0 < D < D^\star \approx 1-q$. *In rosso*: interpolazione dei dati in $1-q \approx D^\star < x < 1$.

P/N	0.05
α_{sup}	0.06 ± 0.01
$(1-q)_{sup}$	0.752 ± 0.008
$\chi^2_{sup}(g.d.l = 5)$	$0.19 \rightarrow 99.5\%$
$2\alpha_{inf}$	0.065 ± 0.008
$(1-q)_{inf}$	0.752 ± 0.004
$\chi^2_{inf}(g.d.l = 4)$	$1.1 \rightarrow c.l < 90.0\%$
Compatibilità α	$0.4 \rightarrow$ ottima
Compatibilità $1-q$	$0 \rightarrow$ ottima

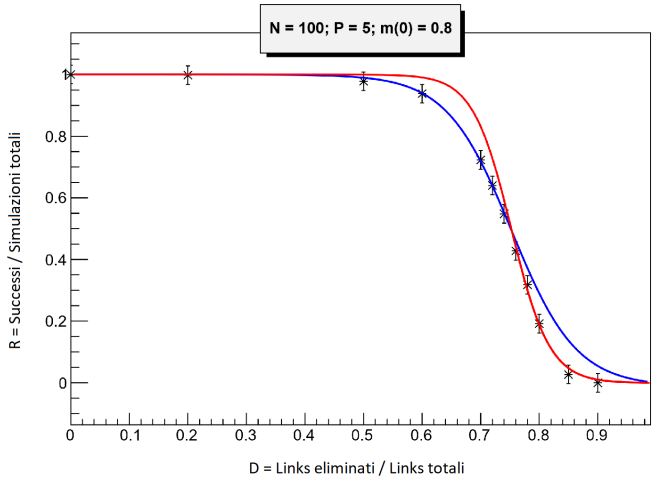

Figura 3.6: A parità di rapporto P/N, la configurazione con 100 nodi presenta delle stime di α nello stesso range di valori di quelle individuate precedentemente per N = 200. La differenza con il caso a 200 nodi riguarda principalmente la stima di $1-q$. *In blu*: interpolazione dei dati in $0 < D < D^\star \approx 1-q$. *In rosso*: interpolazione dei dati in $1-q \approx D^\star < x < 1$.

P/N	0.1
α_{sup}	0.11 ± 0.02
$(1-q)_{sup}$	0.35 ± 0.02
$\chi^2_{sup}(g.d.l = 5)$	$1.08 \rightarrow 95.0\%$
$2\alpha_{inf}$	0.08 ± 0.02
$(1-q)_{inf}$	0.36 ± 0.02
$\chi^2_{inf}(g.d.l = 2)$	$0.02 \rightarrow 97.5\%$
Compatibilità α	$1.06 \rightarrow$ buona
Compatibilità $1-q$	$0.4 \rightarrow$ ottima

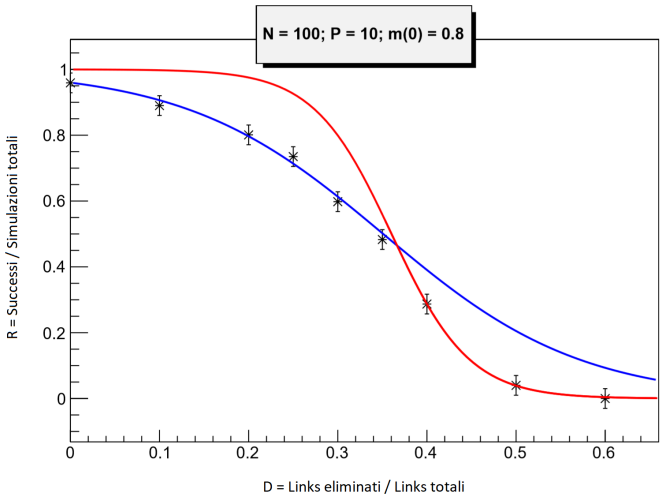

Figura 3.7: Anche in quest' ultima simulazione le stime di α hanno lo stesso range di valori del caso in Figura 3.5, mentre varia la stima di $1-q$. *In blu*: interpolazione dei dati in $0 < D < D^\star \approx 1-q$. *In rosso*: interpolazione dei dati in $1-q \approx D^\star < x < 1$.

Capitolo 4

Conclusione

In questa tesi abbiamo presentato le principali proprietà del network di Hopfield, un modello di rete neurale proposto nel 1982 dal fisico John Hopfield, la cui applicazione riguarda principalmente la realizzazione di memorie associative. Alla rete, infatti, si può associare un' hamiltoniana che presenta dei minimi proprio in quelle configurazioni che si vogliono "ricordare". Abbiamo in particolare studiato come la performance della rete dipende dalla sua connettività.

Nella prima parte del lavoro abbiamo mostrato come in questo sistema il learning possa essere espresso analiticamente attraverso la *regola di Hebb*, che permette di determinare i pesi della matrice sinaptica delle rete in modo da associare le memorie che si vogliono ricordare a minimi di energia $H(\mathbf{S}^\star)$ del sistema. Una volta definita l'hamiltoniana e la regola di Hebb, è stato possibile svolgere delle simulazioni, per mostrare le performance del sistema al variare del numero di neuroni N e del numero di memorie (patterns) da ricordare inserite nella matrice J. L'ultima parte della tesi è stata dedicata allo studio dei rapporti tra l'efficienza del network, ovvero la probabilità che questo riesca a "ricordare" una determinata memoria, e la connectance della rete, ovvero la quantità di connessioni tra i vari elementi rispetto al numero totale di links possibili. Attraverso l'utilizzo di simulazioni è stato possibile infatti ricavare una funzione che lega queste due proprietà della rete, ottenendo un buon grado di accordo con i dati simulati.

La possibilità di mettere in relazione in modo analitico le proprietà topologiche di un network e l'effetto di queste sul suo funzionamento ha una rilevanza sia dal punto teorico di comprensione del funzionamento del learning della rete, sia dal punto di vista dell'efficienza computazionale. Abbiamo visto in particolare che è possibile rimuovere una certa frazione di links senza deteriorare la performance del learning. Questo risultato può quindi permettere di risparmiare costo computazionale e, conseguentemente, costo economico nell'utilizzo del sistema in programmi con l'obiettivo di ricostruire un determinato pattern.

I risultati ottenuti valgono solo per questo specifico tipo di rete, la cui la dinamica è completamente determinata dalle equazioni date dalla regola di Hebb. L'obiettivo più ambizioso che ci prefiggiamo in futuro sarà la generalizzazione di questi risultati anche per reti per cui non conosciamo analiticamente la regola di apprendimento. Questo permetterebbe in parte *to unbox the black box* dei diversi algoritmi, seguendo la strada tracciata dai lavori di fisica statistica degli anni 80, e permettendoci di comprendere sempre meglio come imparano le macchine.

Questo è infatti un progresso necessario, perchè senza di esso non potremo mai gestire in modo pienamente conscio gli strumenti che stiamo iniziando a utilizzare, e che in futuro ricopriranno sempre più importanza nella vita di tutti.

Riprendendo infine l'articolo di Chris Anderson, citato nell'introduzione di questa tesi, risulta evidente che la correlazione, da sola, non può bastare, e che per un autentico progresso scientifico è necessario che l'intelligenza artificiale diventi pienamente comprensibile all'intelligenza umana.

Bibliografia

[1] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in neural information processing systems*, pp. 4349–4357, 2016.

[2] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature reviews neuroscience*, vol. 10, no. 3, p. 186, 2009.

[3] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, 2017.

[4] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[5] D. J. Amit, *Modeling brain function: The world of attractor neural networks*. Cambridge university press, 1992.

[6] H. Nishimori, *Statistical physics of spin glasses and information processing: an introduction*. No. 111, Clarendon Press, 2001.

[7] E. Orhan, "The hopfield model," tech. rep., 2014.

[8] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing infinite numbers of patterns in a spin-glass model of neural networks," *Physical Review Letters*, vol. 55, no. 14, p. 1530, 1985.

[9] D. Horn, E. Ruppin, M. Usher, and M. Herrmann, "Neural network modeling of memory deterioration in alzheimer's disease," *Neural Computation*, vol. 5, no. 5, pp. 736–749, 1993.