

UNIVERSITÀ DEGLI STUDI DEL SALENTO

---



FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

CORSO DI LAUREA IN FISICA

ANNO ACCADEMICO 2008/2009

# **Le Reti Neurali: Isomorfismo tra i modelli di Hopfield e di Ising**

*Tesi di Laurea Triennale*

Relatore:

DOTT. DANIELE MARTELLO

Candidato:

MASSIMO STELLA

Lecce, 23 luglio 2009



# Indice

<b>Introduzione</b>	<b>ii</b>
<b>1 Introduzione alle Reti Neurali</b>	<b>1</b>
1.1 Cenni Introduttivi alla Teoria dei Grafi . . . . .	2
1.2 Connessioni, Complessità e Adattamento : le Reti Neurali Artificiali . . . . .	4
1.3 Analogie biologiche e breve storia delle ANN . . . . .	6
1.4 Funzioni Booleane Linearmente Separabili . . . . .	7
1.5 Le Reti Multistrato . . . . .	9
1.6 L'Algoritmo di Retropropagazione . . . . .	10
1.7 Tasso di Apprendimento e Momento . . . . .	12
1.8 Il problema dell'Apprendimento e sua Complessità . . . . .	13
<b>2 Reti ricorrenti e Reti di Hopfield</b>	<b>15</b>
2.1 Breve Introduzione alla Teoria di Lyapunov . . . . .	16
2.2 Reti di Hopfield . . . . .	17
2.2.1 Reti di Hopfield discrete . . . . .	18
2.2.2 Reti di Hopfield continue . . . . .	20
2.3 Reti di Hopfield e Sistemi Fisici: l'Emergenza di Proprietà Collettive . . . . .	21
2.4 La regola di Hebb e sua generalizzazione . . . . .	23
2.5 Reti di Hopfield ed il Problema del Commesso Viaggiatore . . . . .	25
<b>3 Simulated Annealing e Macchine di Boltzmann</b>	<b>29</b>
3.1 Annealing Simulato ed Energia Libera di Helmholtz . . . . .	30
3.2 L'Algoritmo di Metropolis, la Macchina di Boltzmann e la Matrice di Markov . . . . .	32
3.3 Convergenza asintotica del Simulated Annealing . . . . .	34
3.4 L'Algoritmo di Apprendimento delle Macchine di Boltzmann . . . . .	37
3.5 Dall'Energia Libera all'Entropia . . . . .	39
<b>4 Il Modello di Ising</b>	<b>41</b>
4.1 Modello di Ising: isomorfismo con il modello di Hopfield . . . . .	42
4.2 Ergodicità e Simmetria . . . . .	45
4.3 Sviluppo in Serie ad Alta Temperatura . . . . .	46
4.4 La Formula di Feynman-Kac . . . . .	48
4.5 Annealing Simulato per il Modello di Ising: Complessità . . . . .	54
4.6 Annealing Simulato per il modello di Ising: Risultati . . . . .	55
<b>Conclusioni e Ringraziamenti</b>	<b>60</b>
<b>Appendice</b>	<b>I</b>
<b>Bibliografia e Riferimenti</b>	<b>X</b>

# Introduzione

Nel 1950 il matematico Alan Turing si pone la domanda “*Can machines think?*”. In italiano, “*Le macchine possono pensare?*”. Nel suo articolo *Computing Machinery and Intelligence*, egli propone una sorta di gioco, un vero e proprio test votato all’imitazione di reazioni umane da parte di un elaboratore, per verificarne così l’effettiva intelligenza.

Sempre nell’articolo si legge che tale macchina potrebbe ispirarsi al modello di funzionamento della mente umana e al processo di ragionamento. Tuttavia ci potrebbero essere delle difficoltà:

*“Il problema è principalmente di programmazione. [...] Stime nella capacità di memorizzazione del cervello variano da  $10^{10}$  a  $10^{15}$  cifre binarie. [...] Una capacità di memorizzazione di  $10^7$  sarebbe già fattibile con le tecniche attuali. Non è nemmeno probabilmente necessario aumentare la velocità delle operazioni nelle macchine. Parti di quelle moderne, che possono essere riviste come gli analoghi delle cellule nervose, lavorano circa un migliaio di volte più veloci di queste ultime. Il nostro problema è allora trovare un modo di programmare queste macchine perché possano partecipare al gioco.”*

Turing si interessa allora al possibile sviluppo cui dovrebbe attenersi tale “macchina intelligente”:

*“Invece di provare a produrre un programma per simulare la mente adulta, perché non provare a riprodurre uno che simuli quella di un bambino? [...] Abbiamo così diviso il nostro problema in due parti. Il programma infante ed il processo di educazione. Questi rimangono strettamente connessi. [...] Si deve sperimentare provando l’insegnamento su una di esse [macchine infanti] e vedere quanto e come apprende.”*

Ed infine avvisa anche sull’impossibilità di conoscere approfonditamente le dinamiche del ragionamento di tale apparato:

*“Una caratteristica importante di una macchina in grado di apprendere è che il suo insegnante spesso sarà fondamentalmente ignaro dei processi che si verificano all’interno, nonostante egli sia in grado di prevedere il comportamento del proprio pupillo. Ciò è in chiaro contrasto con le procedure normali quando si usa una macchina per computazioni. [...] Il comportamento intelligente si discosta presumibilmente dal comportamento totalmente disciplinato nelle computazioni, ma solo di poco [...]”*

È interessante notare come l’ispirazione ai modelli biologici, la capacità di apprendimento dall’ambiente e la complessità comportamentale siano tutti elementi di sistemi di notevole interesse scientifico: le *Reti Neurali Artificiali* (in inglese, *Artificial Neural Networks* o *ANN*). L’obiettivo principale di questa tesi è l’identificazione dei concetti fondamentali alla base della teoria delle Reti Neurali e le sue strette correlazioni con i sistemi fisici e la loro modellizzazione.

Nel Capitolo 1 verranno introdotti i concetti di base delle *ANN*, partendo dai fondamenti della teoria e dando una definizione formale di Rete Neurale. Si farà inoltre particolare attenzione all’aspetto computazionale legato agli incarichi di base di una *Artificial Neural Network*.

All’interno del Capitolo 2 si analizzerà invece una tipologia particolare di reti, le cosiddette Reti Ricorrenti. Verrà introdotto il formalismo del modello di Hopfield e se ne verificheranno le analogie con un sistema fisico dotato di punti di equilibrio. Si proporrà anche un’applicazione di Reti Neurali al Problema del Commesso Viaggiatore.

L’argomento principale del Capitolo 3 sarà la Meccanica Statistica impiegata nelle fondamenta teoriche delle Macchine di Boltzmann. Si introdurranno inoltre alcuni concetti relativi all’algoritmo di Simulated Annealing e ai Processi di Markov.

È nel Capitolo 4 che verrà introdotto il modello di Ising, impiegato dai fisici nelle Transizioni di Fase. Si darà luce alla dinamica di una porzione di ferromagnete e se ne studierà l’evoluzione attraverso *cammini aleatori*. Si presenterà infine una simulazione attraverso Reti Neurali Ricorrenti di un ferromagnete, evidenziandone la complessità computazionale e l’esito della metodologia impiegata.

# 1 Introduzione Alle Reti Neurali Artificiali Multistrato

Possiamo scrutare in avanti solo a breve  
distanza, ma possiamo vedere che lì è  
pieno di cose da fare.

---

A.M. TURING

## 1.1 Cenni Introductivi alla Teoria dei Grafi

La Teoria delle Reti affonda una parte delle proprie radici matematiche nella relativamente recente Teoria dei Grafi, il cui fondatore è stato Leonhard Euler, [3].

Celeberrima è la sua risoluzione del problema dei ponti della cittadina di *Königsberg*, così riassunto nel suo lavoro “*Solutio problematis ad geometriam situs pertinentis*” del 1735:

*Nella città di Königsberg, in Prussia, si trova un’isola A, chiamata “Kneiphoff”, con le due anse del fiume (Pregel) tutte attorno. Vi sono sette ponti, a, b, c, d, e, f, e g, che attraversano le due anse. La questione è se una persona possa o meno passeggiare in modo tale che intersechi ogni ponte una sola volta e non di più.*

Eulero cercò di generalizzare la questione per una configurazione qualsiasi dei ponti e si accorse che la distanza fisica non aveva alcuna importanza nella risoluzione del problema, che fu invece affrontata schematizzando il sistema in esame sotto forma di *grafo*.

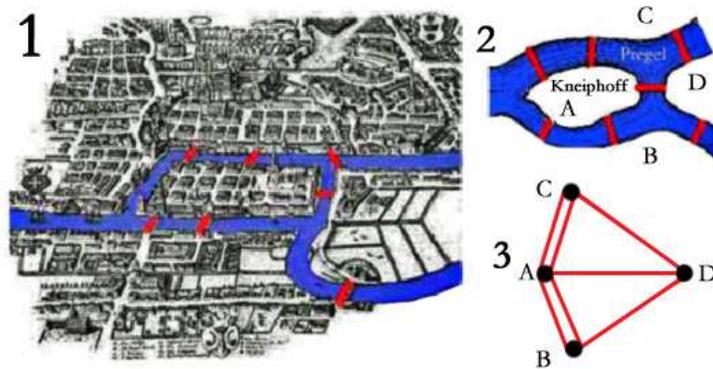


Figura 1.1: (1) Una mappa della città di *Königsberg* dell’epoca, con evidenziati i sette ponti per l’isola di Kneiphoff. (2) Rappresentazione schematica dei sette ponti. Alcuni collegamenti possono essere raggruppati tra di loro, a seconda delle zone di transito. (3) Rappresentazione del problema sotto forma di grafo. Si possono riconoscere le zone collegate come l’isola, la zona meridionale, quella settentrionale e quella orientale come indicate rispettivamente dai nodi A, B, C e D.

Allo scopo di comprendere meglio la questione, diamo una informale definizione di *grafo non orientato*, o semplicemente *grafo*, come un insieme di punti connessi da linee. I primi sono chiamati *nodi* o *vertici*, mentre le seconde vengono chiamate *archi*. Un *cammino* in un grafo è una sequenza di nodi connessi da archi. Un cammino tra due nodi qualsiasi del grafo, tale che *ogni* arco del grafo venga percorso una e una sola volta, viene detto *cammino Euleriano*.

Si definisce inoltre il *grado di un nodo* come il numero di archi che lo connettono.

Eulero divise i nodi del grafo del problema in esame in pari e dispari, a seconda della parità del grado  $k_i$  del nodo  $i$ -esimo. Inoltre, basandosi sul fatto che ogni arco connette due nodi differenti, dimostrò che qualsiasi grafo con  $N$  nodi ed  $A$  archi non orientati soddisfa la relazione:

$$\sum_{i=1}^N k_i = 2A \quad (1.1)$$

Ne sovengono alcune considerazioni:

- La somma dei gradi di un grafo è un numero pari, pertanto<sup>1</sup> ogni grafo possiede un numero pari  $N_d$  di nodi dispari;

<sup>1</sup>Si ricorda che la somma di un numero pari di numeri dispari ha per risultato un numero pari.

- Se  $N_d > 2$ , non esiste alcun cammino Euleriano;
- Se  $N_d = 2$ , esiste un cammino Euleriano avente necessariamente come punto di partenza uno dei due nodi dispari;
- Se  $N_d = 0$ , esiste un cammino Euleriano avente come punto di partenza un nodo arbitrario.

Poiché, allora, tutti e quattro i nodi nel problema dei ponti succitato sono dispari, Eulero dimostrò come non vi fosse alcuna soluzione possibile a quest'ultimo. Secondo alcuni aneddoti anche dopo la diffusione dei risultati di Eulero, molti nobili della cittadina di *Königsberg* continuarono a compiere lunghe passeggiate nel tentativo di trovare il percorso giusto. Inutilmente.

Lo studio di Eulero gettò le basi per quella che diventerà una vera e propria branca della matematica del discreto: la Teoria dei Grafi. Allo scopo di dare una chiara definizione di rete, proponiamo ora alcune definizioni, [4] e [23], che torneranno utili nei prossimi paragrafi.

In un *grafo semplice*  $G$  che contenga i nodi  $i$  e  $j$ , la coppia  $(i, j)$  rappresenta l'*unico* arco che connette i due nodi suddetti. L'ordine di  $i$  e di  $j$  non è determinante in un grafo non orientato, pertanto le coppie  $(i, j)$  e  $(j, i)$  rappresentano il medesimo arco. Un grafo semplice in cui ogni nodo è connesso ai restanti viene detto *completo*.

Se  $V$  è l'insieme di nodi ed  $E$  è l'insieme di archi di un grafo  $G$ , esso si denota con la scrittura  $G = (V, E)$ . Ad esempio il grafo dei sette ponti si può denotare formalmente con la notazione  $(\{A, B, C, D\}, \{(A, B), (A, C), (A, B), (A, C), (A, D), (B, D), (C, D)\})$ .

Frequentemente i grafi vengono utilizzati per rappresentare dati discreti. Talvolta, per convenienza, si danno delle etichette ai nodi o agli archi del grafo, che assume la connotazione di *grafo nominato (labeled)*.

Con il termine di *grafo pesato* si intende invece un grafo ai cui archi vengano associati dei numeri reali, detti *pesi*.

Si dice che il grafo  $G$  è un *sottografo* del grafo  $H$  se i nodi di  $G$  sono un sottoinsieme di quelli di  $H$  e gli archi di  $G$  sono i medesimi di  $H$  sui corrispondenti nodi.

Un *cammino semplice* è un cammino che non ripete alcun nodo. Un grafo è *connesso* se presi due suoi qualsiasi nodi, essi possono essere collegati tramite un cammino. Un cammino è un *ciclo* se esso parte e termina nel medesimo nodo. Un *cammino Hamiltoniano* di un grafo non orientato è un particolare ciclo che visita tutti i nodi del grafo una e una sola volta. Si definisce *planare* un grafo che può essere ridisegnato nel piano in modo che le uniche intersezioni tra archi siano dei nodi.

Se un grafo è connesso tramite frecce e non tramite linee semplici, allora esso assume la connotazione di *grafo orientato*. Il numero di frecce che puntano ad un nodo particolare è detto *grado in ingresso* o *indegree* mentre il numero di frecce uscenti da un particolare nodo è detto *grado in uscita* o *outdegree*. In un grafo orientato si rappresenta un arco da  $i$  a  $j$  come la coppia  $(i, j)$ . La descrizione formale di un grafo orientato  $G$  è la coppia  $(V, E)$  dove  $V$  è l'insieme di nodi ed  $E$  è l'insieme di archi orientati. In analogia al caso precedente, si dice che un cammino in cui tutte le frecce puntano nella medesima direzione lungo i suoi passi è un *cammino orientato*.

## 1.2 Connessioni, Complessità e Adattamento : le Reti Neurali Artificiali

Le Reti Neurali Artificiali, da ora in poi abbreviate in ANN, costituiscono un vero e proprio approccio alternativo alla computabilità.

Seguiamo ora la cosiddetta *tesi di Turing* per dare una definizione formale di “*computazione*”, [4] e [5].

A tale scopo definiamo una *macchina di Turing*  $M$  come un controllo a stati finiti con una testina in grado di leggere o scrivere su un nastro infinito, muovendosi su di esso. Tale nastro è diviso in celle, ognuna delle quali contiene un simbolo di un insieme finito  $\Gamma$ , detto *alfabeto*, che include anche il simbolo vuoto  $b$ . Ogni macchina  $M$  ha uno specifico alfabeto di input  $\Sigma$ , che è un sottoinsieme di  $\Gamma$  che tuttavia non contiene  $b$ . Sia  $\Sigma^*$  un insieme finito di stringhe di elementi di  $\Sigma$ .

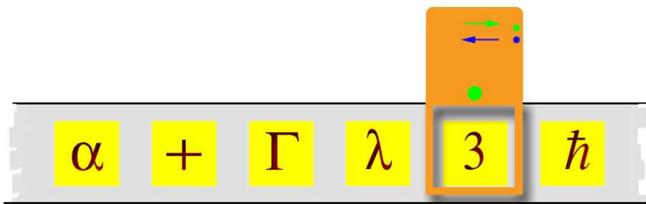


Figura 1.2: Rappresentazione pittorica di una macchina di Turing a nastro infinito.

Ad ogni passo di un incarico computazionale,  $M$  si trova in uno stato  $q$  appartenente ad un insieme finito  $Q$  di stati possibili.

Inizialmente una stringa di input finito su  $\Sigma$  è scritta su alcuni quadrati adiacenti del nastro (tutti gli altri quadrati contengono il carattere  $b$ ), la testina scansiona il simbolo immediatamente a sinistra della stringa di input ed  $M$  si trova nello stato  $q_0$ . Ad ogni passo la

testina scansiona un quadrato contenente un generico simbolo  $s$ , si porta nel generico stato  $q$  ed esegue delle operazioni dettate dalla coppia  $(q,s)$  e specificate dal programma  $\delta$  cui obbedisce  $M$ . L'azione eseguita consiste nella stampa di un simbolo sul quadrato scansionato, muovendosi poi la testina a destra o a sinistra di una cella ed assumendo un nuovo stato.

Formalmente si può schematizzare una macchina di Turing  $M$  come una  $n$ -tuple  $\langle \Sigma, \Gamma, Q, \delta \rangle$  dove  $\Sigma$ ,  $\delta$  e  $Q$  sono insiemi finiti non vuoti con  $\Sigma \subseteq \Gamma$  e  $b \in \Gamma - \Sigma$ . L'insieme degli stati  $Q$  contiene tre stati speciali:  $q_0$ ,  $q_{accept}$  e  $q_{reject}$ .

Si definisce una *configurazione* di  $M$  una stringa  $xqy$ , con  $x, y \in \Gamma^*$ , con  $y \neq b$  e  $q \in Q$ . L'interpretazione della configurazione  $xqy$  è che  $M$  è nello stato  $q$  con  $xy$  sul suo nastro, con la testa che sta scansionando il simbolo più a sinistra di  $y$ . Si dice che una configurazione  $xqy$  è di arresto se  $q \in \{q_{accept}, q_{reject}\}$ .

Una *computazione di  $M$  sull'input  $w \in \Sigma^*$*  è pertanto la sequenza unica  $C_0, C_1, \dots$  di configurazioni tali che  $C_0 = q_0w$  e la generica configurazione  $C_i$  è univocamente determinata dalla precedente attraverso il programma  $\delta$ . Tale sequenza può essere infinita o può terminare con una configurazione di arresto. Se la computazione è finita, allora il numero di passi è pari al numero di configurazioni meno uno, altrimenti il numero di passi è infinito.

A differenza della macchina di Turing, una generica rete neurale non opera sequenzialmente e l'intero gruppo di dati può essere processato simultaneamente. Inoltre, contrariamente ai computer odierni, non è necessaria la creazione di alcun programma per una certa computazione, poiché essa è eseguita dalla rete nella ricerca adattiva di alcuni suoi parametri liberi.

È giunto il momento di introdurre allora una *Rete Neurale Artificiale* come un *grafo pesato orientato* i cui nodi processano l'informazione e prendono il nome di *unità computazionali*

o *agenti* o *neuroni artificiali* e i cui archi vengono chiamati *sinapsi* o *interconnessioni* o *connessioni*, le quali trasmettono dati numerici da nodo a nodo e memorizzano i risultati di una specifica computazione. Si noti che la definizione di grafo caratterizza, in parte, l'*architettura* della rete, poiché ne determina la coppia  $(V, E)$  di nodi e di connessioni.

L'unità fondamentale di una ANN è il *neurone artificiale*, rappresentato nella figura sottostante. Esso può ammettere in ingresso  $n$  canali di input, ciascuno in grado di trasmettere un valore reale generico  $x_i$ . La *funzione primitiva* o *funzione di attivazione*  $f$  computata nel corpo del neurone artificiale può essere selezionata arbitrariamente. Usualmente i canali di input hanno un peso associato, pertanto l'informazione in ingresso  $x_i$  è moltiplicata per il corrispondente peso  $w_i$ . L'informazione così trasmessa è integrata nel neurone, che genericamente effettua una somma dei prodotti  $x_i w_i$  e la funzione primitiva è così valutata.

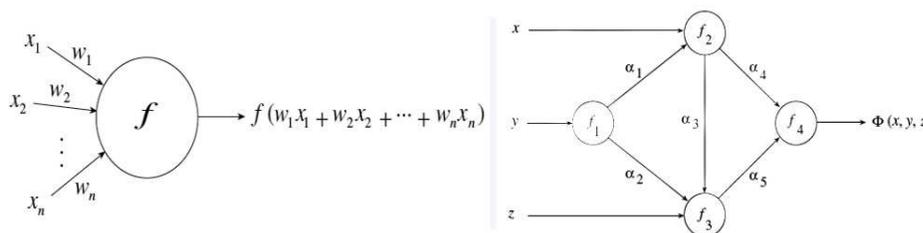


Figura 1.3: A sinistra, la schematizzazione di un neurone artificiale. A destra la schematizzazione di una generica rete neurale artificiale.

Una rete generica, come quella rappresentata nella figura di sopra, può essere pensata come una funzione  $\Phi$  che viene valutata nel punto  $(x, y, z)$ . I nodi implementano le funzioni primitive  $f_1, f_2, f_3, f_4$  che vengono combinate per produrre  $\Phi$ , che verrà chiamata *funzione di rete* o *network function*. Differenti selezioni dei pesi  $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$  producono differenti funzioni di rete, [2]. Essenzialmente quattro sono gli elementi particolarmente importanti in una qualsiasi architettura di ANN: la *struttura* dei nodi, le arbitrarie *funzioni primitive*, la *topologia* della rete, cioè la distribuzione generale delle connessioni, e l'*algoritmo di apprendimento* usato per trovare i pesi della rete.

In ogni Rete Neurale Artificiale è presente, inoltre, una ipotesi basilare: l'intelligenza del sistema globale non implica necessariamente l'intelligenza dei suoi singoli componenti. L'attenzione è quindi spostata dall'aspetto microscopico ai "fenomeni emergenti", cioè quelle evidenze macroscopiche che presenta il sistema nella sua interezza e che non possono essere predette o dedotte dal funzionamento delle singole unità di base. Questo è un elemento caratterizzante dei cosiddetti *sistemi complessi*. Tale *emergenza*, avente ripercussioni importanti anche da un punto di vista squisitamente fisico, verrà analizzato con calma più avanti.

Va inoltre sottolineata la capacità di una ANN di adattamento all'ambiente circostante. Si è già citato il termine "adattativo", indice di una specifica relazione generale tra il sistema stesso e il suo ambiente, dove il primo è in grado di apprendere dal secondo senza programmazione specifica. Per sviluppare una teoria sull'apprendimento, lo studio delle ANN si avvale di strumenti matematici che derivano dalla teoria della probabilità, con i quali la rete viene considerata anche come un *agente statistico*, che possiede una certa conoscenza dell'ambiente circostante e acquisisce nuove informazioni per ridurre il proprio grado di incertezza.

### 1.3 Analogie biologiche e breve storia delle ANN

Il modello costitutivo di una ANN è ispirato ai fondamenti biologici riguardanti i tessuti nervosi del cervello umano, [2] e [6].

Quest'ultimo contiene circa  $10^{11}$  *neuroni* (detti anche *cellule nervose*). Ogni neurone può essere distinto in tre parti: un insieme di fibre in ingresso (dette *dendriti*), un corpo cellulare (detto *soma*) ed una diramazione principale (detta *assone*). A loro volta gli assoni si dividono in differenti terminazioni, ognuna delle quali si trova a contatto con altri neuroni. Ogni cellula nervosa può ricevere sino a 10.000 impulsi in ingresso e le zone di contatto e di trasmissione di tali impulsi vengono dette *sinapsi* (il cervello umano ne contiene mediamente  $3 \cdot 10^{14}$ ). Le controparti artificiali di neuroni e sinapsi sono le unità computazionali e le interconnessioni.

Come già anticipato, i neuroni possono generare impulsi, di tipo elettrico, che vengono trasmessi attraverso l'assone e verso le sinapsi. Ogni impulso elettrico può contribuire all'eccitazione o all'inibizione del neurone ricevente. Inoltre l'efficienza con cui le sinapsi trasmettono i segnali elettrici può essere variata nel tempo. All'interno del cervello, i neuroni sono organizzati in vere e proprie reti, a loro volta incluse in moduli indipendenti gli uni dagli altri. Tale fattore assicura la possibilità di elaborare dati *in parallelo* ed operare più processi simultaneamente. Il modello artificiale di trasmissione delle informazioni tra i vari nodi di una ANN risulta proprio una esemplificazione di tali impulsi elettrici dei sistemi biologici.

Si conosce ancora poco riguardo alcuni meccanismi di funzionamento del cervello umano quali la memoria, la coscienza, il ragionamento e la comprensione. Ciò non deve sbalordire, essendo lo studio della struttura e della fisiologia delle cellule nervose relativamente recente. I primi risultati empirici si ottennero pressapoco attorno al 1850 e solo un secolo dopo si ebbero le prime modellizzazioni matematiche, attraverso equazioni differenziali, per la propagazione degli impulsi elettrici, [Cro87]. Tali modelli rappresentano un punto di contatto tra biologia e fisica nel campo neuro-cognitivo.

In realtà lo studio delle ANN iniziò già nel 1927, con i primi studi portati avanti dal neuroscienziato Warren McCulloch, che nel 1943, con la collaborazione del logico Walter Pitts, propose il primo modello di *neurone artificiale* nell'articolo "*A logical calculus of the ideas immanent in nervous activity*". In tale modello i nodi producono risultati binari e vi possono essere un numero illimitato di input, distinti in *inibitori* o *eccitatori*, a seconda delle connessioni di trasmissione. Se anche solo uno degli input inibitori è 1, l'unità neurale viene inibita e il risultato della computazione è 0. Se ciò non avviene, si computa l'eccitazione totale  $x$ , cioè la semplice somma degli input eccitatori, e la si confronta con un valore di soglia  $\theta$  propria del neurone artificiale. Nell'eventualità che  $x \geq \theta$  l'unità *riporta in uscita* il valore 1, altrimenti il risultato della computazione è ancora 0.

Nonostante l'enorme flessibilità dei neuroni di McCulloch-Pitts, cui si interessarono anche scienziati del calibro di Von Neumann, la prima rete neurale che conobbe un notevole successo commerciale fu il *perceptrone*, proposto da Frank Rosenblatt nel 1958. L'innovazione principale di tale nuovo modello di ANN fu l'introduzione di pesi differenti per differenti connessioni e una speciale struttura di interconnessione. Il perceptrone è oggi impiegato nel riconoscimento delle forme nei problemi di classificazione. Si tratta di una rete formata da un sottoinsieme, detto *strato*, di unità di input e da un altro strato di unità di output. Non vi sono connessioni tra le unità di uno stesso strato e le connessioni tra le unità di input e di output sono unidirezionali dalle prime alle seconde.

Le reti che non contengono connessioni tra le unità di uno stesso strato e in cui la propagazione dei dati è diretta esclusivamente dalle unità di ingresso a quelle di uscita sono dette *reti*

*feedforward*. Qui l'informazione viene collezionata da ciascun nodo tramite  $n$  linee di input e computata attraverso una *funzione di integrazione*  $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ , che nel perceptrone è una semplice somma degli input. L'eccitazione totale delle unità computazionali viene valutata usando una *funzione di attivazione*  $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ , che, sempre nel perceptrone, compara anche la somma suddetta con un valore di soglia. L'addestramento del perceptrone avviene in maniera iterativa, stabilendo dall'esterno se la risposta fornita dalla rete, a fronte di un certo input, sia corretta o errata. In questo secondo caso si alterano i pesi delle interconnessioni in modo che la rete si avvicini alla risposta corretta.

Nel 1969, nel loro libro *Perceptrons*, i matematici Minsky e Papert analizzarono approfonditamente i limiti e le possibilità del perceptrone semplice (con un unico strato di pesi adattabili), mostrando inoltre che la classe di funzioni che esso è in grado di discriminare è limitata alle *forme linearmente separabili*. La necessità di prestazioni più varie portò all'ideazione di *reti multistrato* e all'introduzione di un conseguente nuovo algoritmo di apprendimento, ideato dal matematico Werbos nel 1974. Da allora, sino ai giorni nostri, il settore delle ANN ha goduto di una crescente matematizzazione e modellizzazione, anche ad opera di raffinati strumenti di indagine prettamente fisici, quali ad esempio la Meccanica Statistica.

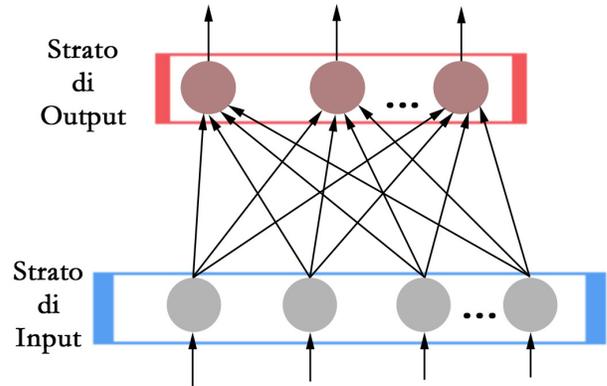


Figura 1.4: Architettura di un Perceptrone.

## 1.4 Funzioni Booleane Linearmente Separabili

Per comprendere le limitazioni insite nel perceptrone e le rispettive implicazioni, si è ora interessati alla identificazione di un particolare tipo di funzioni booleane, [1].

Sia  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  una generica funzione booleana. Il suo dominio rappresenta un ipercubo nello spazio  $n$ -dimensionale, ossia un insieme di vettori di  $\mathbb{R}^n$  a componenti  $\{0, 1\}$ . È noto che il lato  $l$  di un ipercubo è individuato da una coppia di vertici  $\underline{\nu} = (\nu_1, \nu_2, \dots, \nu_n)$  e  $\underline{\nu}' = (\nu'_1, \nu'_2, \dots, \nu'_n)$  che differiscono per una sola componente. Alla luce di ciò si può rivedere una funzione booleana come il sottoinsieme  $A_f$  di vertici di un ipercubo di cui è la funzione caratteristica:  $A_f = \{\underline{\nu} \mid f(\nu_1, \nu_2, \dots, \nu_n) = 1\}$ .

Ogni funzione booleana è completamente determinata da una delle due immagini della sua inversa  $f^{-1}(1) = \{X \in \{0, 1\}^n : f(X) = 1\}$ , che coincide con l'insieme dei vertici appartenenti a  $\{0, 1\}^n$  in cui la funzione booleana  $f$  vale 1, ed  $f^{-1}(0) = \{X \in \{0, 1\}^n : f(X) = 0\}$ , che coincide con l'insieme dei vertici appartenenti a  $\{0, 1\}^n$  in cui la funzione booleana  $f$  vale 0. Per quanto detto sopra,  $f$  è la funzione caratteristica del sottoinsieme  $f^{-1}(1)$ , per tale ragione la si può significativamente identificare con tale sottoinsieme.

In figura vengono riportate tre note funzioni booleane  $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ , rispettivamente AND, OR e XOR.

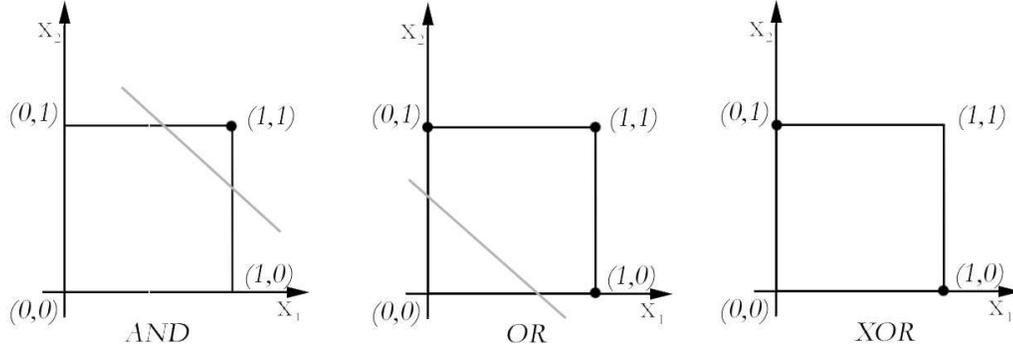


Figura 1.5: Alcuni esempi di funzioni booleane. La palla nera indica i punti in cui la funzione booleana assume il valore 1. Si noti che nel caso delle funzioni AND e OR, con una linea retta è possibile separare il piano in due semiregioni distinte, una delle due contenente solo i vertici del quadrato sui quali la  $f$  assume valore 1. Tale separazione grafica non è evidentemente possibile nel caso della funzione XOR. Ciò rappresenta una prima visualizzazione del concetto di funzione booleana linearmente separabile.

L'interpretazione geometrica di  $\{0, 1\}^n$  gioca un ruolo determinante nello studio di una sottoclasse di funzioni booleane, dette *linearmente separabili*.

Una funzione booleana  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  si dice linearmente separabile o *funzione soglia* se esiste un iperpiano nello spazio euclideo ad  $n$  dimensioni  $\mathbb{R}^n$  che separa l'insieme  $f^{-1}(1)$  dall'insieme  $f^{-1}(0)$ . Il suddetto iperpiano è detto *iperpiano separatore* per la funzione  $f$ . Osservando la figura sopra riportata, appare evidente come la funzione XOR non sia linearmente separabile. Pertanto non tutte le funzioni booleane sono funzioni soglia.

L'iperpiano separatore per una funzione linearmente separabile  $f$  è  $S = w_1x_1 + w_2x_2 + \dots + w_nx_n$ . Inoltre i due semispazi generati da  $S$  sono :

$$S_p^+ = \{X \mid w_1x_1 + w_2x_2 + \dots + w_nx_n \geq S\} \text{ e } S_p^- = \{X \mid w_1x_1 + w_2x_2 + \dots + w_nx_n < S\}$$

Per un punto arbitrario  $X = (X_1, \dots, X_n)$  dell'ipercubo  $\{0, 1\}^n$  valgono le relazioni:

$$w_1x_1 + w_2x_2 + \dots + w_nx_n \geq S \text{ se } f(X) = 1 \tag{1.2}$$

$$w_1x_1 + w_2x_2 + \dots + w_nx_n < S \text{ se } f(X) = 0 \tag{1.3}$$

Portando  $S$  al primo membro in entrambe le disuguaglianze, si ha una espressione che ricorda la funzione di Heaviside, che permette di esprimere pertanto la funzione  $f$  come  $f(x_1, \dots, x_n) = H(w_1x_1 + w_2x_2 + \dots + w_nx_n - S)$ .

Si definisce *sistema separatore* l'insieme dei coefficienti  $(w_1, \dots, w_n, S)$  che individuano l'iperpiano separatore, dove gli  $n$  coefficienti reali  $w_1, w_2, \dots, w_n$  sono denominati *pesi* e il coefficiente  $S$  è chiamato *soglia*. Si noti l'assoluta analogia con quanto descritto in maniera introduttiva sulle unità di McCulloch-Pitts. Si veda anche che moltiplicando per un numero positivo i coefficienti del sistema separatore, l'iperpiano e i due semispazi da esso generati non cambiano, per le proprietà della funzione di Heaviside.

Si prenda ora come modello di computazione per le funzioni booleane AND, OR e XOR una unità computazionale a soglia con due ingressi,  $x_1$  e  $x_2$ . Esprimendo le funzioni booleane nella forma  $H(w_1x_1 + w_2x_2 - S)$ , verificando le possibili combinazioni di  $x_1$  ed  $x_2$  e i rispettivi

valori delle funzioni booleane, si ottengono tre sistemi. Per l'AND e per l'OR si ottengono infinite soluzioni<sup>2</sup> possibili.

Dai calcoli relativi allo XOR si osserva, [1], che se esso ammettesse soluzione allora risulterebbe  $2S \leq w_1 + w_2 < S$ , con  $S > 0$ . Ciò equivale ad una contraddizione e comporta l'impossibilità di esprimere la funzione OR-esclusivo come funzione linearmente separabile. Il fatto che non esistano dei pesi reali per il modello di neurone binario a soglia presupposto, del tutto equivalente ad un perceptrone, esclude il fatto che quest'ultimo possa calcolare la funzione XOR.

## 1.5 Le Reti Multistrato

Le reti multistrato rappresentano una generalizzazione ai principi base del perceptrone.

Si tratta di reti feedforward con strati di unità compresi tra lo strato di input e quello di output e che non interagiscono direttamente con l'ambiente esterno. Per tale motivo questi nodi vengono chiamati *unità nascoste* e sono il punto chiave per il superamento dei limiti computazionali del perceptrone. Infatti, esse consentono di elaborare una rappresentazione interna dei vettori di input, utile per realizzare trasformazioni non linearmente separabili tra le unità di ingresso e di uscita. Minsky e Papert misero in evidenza, nell'edizione del 1987 di *Perceptrons*, che una rete multistrato con un solo strato di unità nascoste è sempre in grado di realizzare la trasformazione richiesta per un qualsiasi tipo di funzione booleana da computare.

L'architettura generale di una rete multistrato, detta anche *perceptrone generalizzato*, può contenere più strati di unità nascoste, dove le interconnessioni sono unidirezionali e vanno dalle unità di un certo strato a quelle di uno strato di livello superiore, cioè più prossimo allo strato di output. Lo stato  $x$  di un neurone appartiene all'intervallo  $(0, 1)$  e la funzione di attivazione è data da

$$x_j = g \left( \sum_k w_{jk} x_k \right) \quad (1.4)$$

dove la funzione  $g$  è differenziabile.

Per ogni configurazione  $\underline{x}$  dello strato di ingresso la rete calcola una configurazione  $\underline{y}$  dello strato di uscita. Si fissa dunque una trasformazione  $f$  tra le configurazioni di ingresso e quelle di uscita. Si fissano inoltre dei pesi iniziali casuali. Fornendo alla rete una sequenza di stimoli  $\{\underline{x}_k\}$ , essa cambia i pesi delle connessioni in modo che, dopo un numero finito  $N$  di passi del cosiddetto *apprendimento*, la funzione di rete, cioè l'uscita  $\underline{y}_k$ , coincida con  $f(\underline{x}_k)$  per ogni  $k > N$ , con la precisione voluta. Appare così evidente che la modifica dei pesi avviene in modo da minimizzare le discrepanze tra la risposta della rete e quella ricercata. Si dimostra comunque che l'approssimazione di una funzione continua attraverso una rete neurale è sempre permessa, [2].

Ci si può chiedere se sia opportuno spingersi ancora oltre l'approssimazione e puntare alla rappresentazione esatta attraverso una rete neurale. Utilizzando le giuste connessioni tra le unità di input ed un insieme sufficientemente ampio di unità nascoste, è sempre possibile trovare una rappresentazione che realizzi la trasformazione  $f$ , assegnata implicitamente attraverso il set di dati di addestramento. A tal proposito è interessante proporre il seguente teorema,

---

<sup>2</sup>Un sistema separatore per l'AND può essere ( $w_1 = 1, w_2 = 1; S = \frac{3}{2}$ ) mentre per l'OR si può avere ( $w_1 = 1, w_2 = 1; S = \frac{1}{2}$ ).

dimostrato nel 1957 dal celebre matematico russo Kolmogorov, [2], sulla rappresentazione di funzioni continue ad  $n$  argomenti:

**Teorema (di Kolmogorov):** *Se  $f : [0,1]^n \rightarrow [0,1]$  è una funzione continua, allora esistono funzioni ad un argomento  $g$  e  $\phi_q$  per  $q = 1, \dots, 2n+1$  e delle costanti  $\lambda_p$ , per  $p = 1, \dots, n$  tali che*

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} g \left( \sum_{p=1}^n \lambda_p \phi_q(x_p) \right).$$

Il teorema di Kolmogorov, la cui dimostrazione si può trovare in [Spr64], asserisce che funzioni continue ad  $n$  argomenti possono sempre essere rappresentate usando una composizione finita di funzioni ad un singolo argomento e l'addizione. Per quanto detto sulle funzioni di rete e sulle funzioni primitive, il teorema sopra citato diviene particolarmente importante nel campo delle reti neurali, poiché comporta che ogni funzione continua possa essere riprodotta esattamente da una opportuna rete finita di unità computazionali, qualora le funzioni primitive per ogni nodo e i pesi per le interconnessioni siano individuabili.

Chiudiamo questa breve introduzione alle reti multistrato illustrando un utile formalismo nella rappresentazione dell'output. Si consideri un perceptrone con tre unità di input, 1-2-3, e due unità di output, 1-2. Sia  $w_{ij}$  il peso relativo all'interconnessione tra il nodo di input  $j$  e l'unità di output  $i$ . Definendo rispettivamente l'input e l'output della rete con  $\underline{x}$  e  $\underline{y}$ , si ha:

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \underline{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad \longrightarrow \quad \underline{y} = \mathbf{W}\underline{x} \quad \text{con} \quad \mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \quad (1.5)$$

## 1.6 L'Algoritmo di Retropropagazione

Si consideri una rete feedforward, avente vari strati di unità nascoste, con funzioni di attivazione non lineari, per non perdere in generalità.

Al  $k$ -esimo passo, l'output totale di uno strato della rete è:

$$z_{kj} = \sum_i w_{ji} o_{ki} \quad (1.6)$$

dove  $o_{ki}$  rappresenta lo stato dell'unità  $u_i$  al passo  $k$ ,  $z_{kj}$  è l'input di rete dell'unità  $u_j$  dello strato successivo, al passo  $k$ , e  $w_{ji}$  è il peso della connessione tra le unità  $u_i$  e  $u_j$ . Nel caso particolare in cui  $u_i$  sia rispettivamente una unità di input o una unità di output, allora  $o_i = x_i$  oppure  $o_i = y_i$ .

La funzione di attivazione qui considerata è semilineare, cioè è tale che  $o_{kj} = f_j(z_{kj})$  ed  $f_j$  è non decrescente e differenziabile nella variabile  $z_{kj}$ . Si osserva ad esempio che la funzione lineare a soglia del perceptrone non soddisfa la richiesta poiché la sua derivata è discontinua.

L'obiettivo ora è minimizzare l'errore quadratico medio sulla risposta della rete ad un input di ingresso  $\underline{x}_k$ , definito come:

$$E_k = \frac{1}{2} \sum_j (t_{kj} - y_{kj})^2 \quad (1.7)$$

dove  $t_{kj}$  è la  $j$ -esima componente del vettore atteso in uscita e  $y_{kj}$  è la  $j$ -esima componente del vettore effettivamente ottenuto in output.

Si definisce *misura globale dell'errore* la quantità  $E = \sum_k E_k$ .

Si può dimostrare, [1] e [2], che l'algoritmo di retropropagazione realizza una discesa lungo la superficie descritta da  $E$  se la modifica al passo  $k$  del peso  $w_{ji}$  è proporzionale all'opposto della derivata parziale della misura dell'errore rispetto al peso, cioè se

$$\Delta_k w_{ji} \propto -\frac{\partial E_k}{\partial w_{ji}} \quad (1.8)$$

Si può dimostrare, come riportato in [1], che il cambiamento dei pesi avviene in maniera proporzionale ad un preciso *segnale d'errore*  $\delta$  (relativo all'unità che riceve l'input lungo quella data connessione) per l'uscita  $o$  (relativa all'unità che invia l'attivazione lungo quella data connessione), secondo l'equazione seguente, che prende il nome di *regola delta generalizzata*:

$$\Delta_k w_{ji} = \mu \delta_{kj} o_{ki} \quad (1.9)$$

dove la grandezza  $\mu$  è un numero reale, detto *tasso di apprendimento*.

Si noti che la regola delta generalizzata è in realtà una estensione a funzioni semilineari della cosiddetta *regola delta standard*, che effettua un'analisi dell'errore quadratico medio per funzioni di attivazione lineari e per reti senza strati di unità nascoste.

Si può riassumere l'algoritmo di retropropagazione in una scaletta esemplificativa delle varie fasi:

1. STIMOLAZIONE : ad ogni unità di ingresso  $u_j$  si assegna lo stato  $x_{kj}$ ;
2. PROPAGAZIONE : per ogni unità nascosta e di uscita si calcola l'uscita  $o_{kj} = f_j \left( \sum_i w_{ji} o_{ki} \right)$  (*dinamica dell'attivazione*);
3. CONFRONTO : per le unità d'uscita si trova il segnale d'errore  $\delta_{kj} = (t_{kj} - o_{kj}) f'_j \left( \sum_i w_{ji} o_{ki} \right)$ ;
4. RETROPROPAGAZIONE : i valori calcolati in uscita diventano input per le unità di uscita della rete, quindi li si trasmette all'indietro per ogni unità nascosta, di strato in strato, sino a giungere alle unità di input; in tale *retropropagazione*, per ogni strato si calcola ricorsivamente il segnale di errore  $\delta_{kj} = f'_j \left( \sum_i w_{ji} o_{ki} \right) \sum_l \delta_{kl} w_{il}^{k-1}$  (*dinamica dei co-stati*), dove l'esponente sui pesi connota il passo;
5. AGGIORNAMENTO DEI PESI : si aggiornano i pesi secondo  $w_{ji}^k = w_{ji}^{k-1} + \mu \delta_{kj} o_{ki}$ .

Nell'implementazione effettiva dell'algoritmo di retropropagazione si deve scegliere una certa funzione di attivazione che sia differenziabile (necessità di calcolarne la derivata prima) e monotona (necessità di implementare una discesa lungo la superficie rappresentata da  $E$ ). Una scelta molto frequente è la funzione logistica

$$z = \frac{1}{1 + e^{-x}} \quad (1.10)$$

la cui derivata prima è esprimibile in funzione dell'output  $z$ , difatti  $z' = \frac{1}{(1+e^{-x})} \frac{1+e^{-x}-1}{1+e^{-x}} = z \cdot (1 - z)$ .

## 1.7 Tasso di Apprendimento e Momento

Ci interroghiamo ora sulla natura del tasso di apprendimento  $\mu$  nella formula (1.9). È noto che si tratta di un numero reale. Ma di quale entità?

La risposta a questa domanda proviene dalla regola delta generalizzata, pensata come una discesa nello spazio dei pesi, alla ricerca del minimo della funzione

$$E(\underline{w}) = \frac{1}{2} \sum_i (y_i^* - \mathbf{W}\underline{x}_i)^2 \quad (1.11)$$

dove  $y_i^*$  è la risposta attesa dalla rete, mentre il relativo minuendo altro non è che la risposta data dalla rete,  $y_i$ , secondo il formalismo introdotto nel paragrafo sulle reti multistrato. Tale discesa non è la più ripida possibile poiché avviene in base all'aggiornamento dei vari pesi della rete ad ogni singola coppia di vettore di input e di output desiderato e non solo dopo aver considerato l'intero insieme di *addestramento*, cioè l'insieme complessivo<sup>3</sup> delle coppie  $(\underline{x}_i, y_i^*)$ .

Invece, un algoritmo che modifica i parametri solo dopo che tutti i dati di addestramento sono stati considerati, al fine di calcolare le varie  $y_i$ , viene definito *batch*. La tecnica di *discesa più ripida del gradiente*<sup>4</sup> o *steepest gradient descent* ne è un esempio.

La regola delta generalizzata non coincide chiaramente con la discesa più ripida del gradiente, poiché non è di tipo batch ma *on-line*. Tuttavia si può approssimare la discesa realizzata dalla regola delta generalizzata a quella più ripida se e solo se ad ogni passo del processo iterativo di apprendimento il tasso di apprendimento  $\mu$  è sufficientemente piccolo da non modificare di quantità troppo grandi i pesi. Di contro, se il valore di  $\mu$  è troppo piccolo il processo di discesa può essere estremamente lento. Ma d'altronde valori di  $\mu$  troppo grandi provocano forti oscillazioni sulla discesa. Tali oscillazioni sono legate alla presenza di valli profonde, con pendenza quasi nulla, sul fondo della superficie della funzione d'errore. In tali zone, il vettore dei pesi, ad ogni aggiornamento, continua a puntare verso il minimo della funzione d'errore ma possiede anche una componente che lo devia da tale direzione e che è la causa di oscillazioni dirette in verso opposto in aggiornamenti consecutivi.

Un buon compromesso è una modifica della regola delta generalizzata tramite l'aggiunta di un termine *momento*, che comprende la direzione del cambiamento precedente:

$$\Delta w_{ji}(n+1) = \mu \delta_{kj} o_{ki} + \alpha \Delta w_{ji}(n) \quad (1.12)$$

dove  $n$  è il numero dell'aggiornamento ed  $\alpha$  è una costante che determina l'effetto dell'ultima modifica del peso  $w_{ji}$  sulla nuova direzione del vettore dei pesi.

L'aggiunta del momento smorza le oscillazioni, in quanto sommando variazioni di peso consecutive si annullano le componenti opposte e si sommano le componenti concordi, dirette verso il minimo. L'apprendimento diventa ancora più veloce quando  $\alpha$  e  $\mu$  assumono entrambi valori più elevati. Inoltre la smorzatura dovuta al momento si concretizza in una ricerca del minimo più rapida e più robusta. Anche il tasso di apprendimento garantisce una certa

<sup>3</sup>Un intero ciclo di apprendimento sull'insieme di addestramento viene detto *epoca*.

<sup>4</sup>Si tratta di un algoritmo batch relativo a problemi di ottimizzazione che raggiunge il minimo della funzione costo  $J$ , nel nostro caso pari alla (1.11), attraverso la modifica dei pesi:

$$\Delta \underline{w} = -\lambda \nabla_{\underline{w}} J = \lambda \sum_i (y_i^* - \underline{w}^T \underline{x}_i) \underline{x}_i$$

robustezza all'algoritmo di apprendimento, poiché lo preserva da eventuale rumore presente nei dati dell'insieme di addestramento. Per determinare un valore opportuno del valore di  $\mu$  si possono utilizzare informazioni sulla dimensione dei vettori di ingresso; per quantificare il momento, invece, è necessario disporre di informazioni sulla forma della superficie della funzione di errore, [1].

## 1.8 Il problema dell'Apprendimento e sua Complessità

Nelle ANN gioca un ruolo fondamentale l'algoritmo di apprendimento, che si occupa di minimizzare l'errore in uscita. La prova del teorema di Kolmogorov, in quanto non costruttiva, non dà alcun indizio sulla possibile scelta delle funzioni primitive per un determinato incarico, né tanto meno sulla scelta dei pesi  $\lambda_i$ .

Il *problema generale dell'apprendimento* per una rete neurale consiste nel trovare gli elementi ignoti di una data architettura, quali ad esempio la funzione di attivazione delle unità computazionali o i pesi delle interconnessioni, partendo da un insieme di coppie di vettori di input e di output desiderato. La dimensione del problema dell'apprendimento dipende chiaramente dal numero di variabili ignote da determinare e dal tipo di algoritmo scelto per la loro individuazione.

È stato provato, si veda [2], che tale problema generale per le reti neurali è *intrattabile*, cioè non efficientemente risolvibile per ogni possibile eventualità. Nella terminologia della teoria della Complessità si dice che il problema generale dell'apprendimento per reti neurali è *NP-completo*. A tal proposito introduciamo alcune utili nozioni di base, [2,4,5,7], riguardanti le classi di complessità computazionale, come quella appena citata della NP-completezza.

Formalmente gli elementi di una classe computazionale sono *linguaggi*, cioè sottoinsiemi finiti di stringhe di un alfabeto  $\Sigma$  con almeno due elementi. Un linguaggio è *accettato* da una macchina di Turing  $M$ , e si indica con  $L(M)$ , se un input  $w$  formato da elementi di  $L$  viene accettato da  $M$ .

Si denota come  $tM(w)$  il numero di passi nella computazione di  $M$  riguardo l'input  $w$ , relativo ad un determinato problema. Se tale computazione non si arresta mai, allora  $tM(w) = \infty$ . Per  $n \in \mathbb{N}$  si indica con  $TM(n)$  la computazione che richiede il maggior numero di passi di  $M$ . Si dice che  $M$  *impiega un tempo polinomiale* se esiste un  $k$  tale che per ogni  $n$  risulti  $TM(n) \leq nk + k$ . Dunque si definisce la classe  $P$  di linguaggi come :

$$P = \{L \mid L = L(M) \text{ per una macchina di Turing } M, \text{ deterministica e a nastro singolo, che termina la computazione in un tempo polinomiale}\}$$

Si può definire la classe di linguaggi<sup>5</sup>  $NP$  :

$$NP = \{L \mid L = L(M) \text{ per una macchina di Turing } M, \text{ non deterministica e a nastro singolo, il cui risultato della computazione è verificabile in tempo polinomiale}\}$$

Si ricorda che la verifica di un risultato è a sua volta un algoritmo, esprimibile in termini di linguaggi. L'uguaglianza  $P = NP$  è uno dei Problemi del Millennio ancora irrisolto:  $P \subset NP$  o  $P = NP$  ? La nozione di computazione in tempo polinomiale fu introdotta negli anni Sessanta da Cobham ed Edmonds, sebbene Von Neumann nel 1953 avesse già distinto tra

<sup>5</sup>NP sta per "nondeterministic polynomial time". Si ricorda che in una macchina non deterministica esistono più configurazioni possibili successive alla configurazione  $i$ -esima, in un incarico computazionale.

algoritmi risolvibili in tempi polinomiali o esponenziali, appartenenti questi ultimi alla classe *EXPTIME*.

Necessaria per la definizione di *NP-completezza* è la nozione di *p-riducibilità* ( $\leq_p$ ).

*Sia  $L_i$  sia un linguaggio su  $\Sigma_i$ ,  $i = 1, 2$ . Allora  $L_1 \leq_p L_2$  ( $L_1$  è *p-riducibile* ad  $L_2$ ) se esiste una funzione computabile in un tempo polinomiale  $f : \Sigma_1^* \rightarrow \Sigma_2^*$  tale che  $x \in L_1 \iff f(x) \in L_2, \forall x$  appartenente a  $\Sigma_1^*$ , dove l'apice \* caratterizza sottoinsiemi finiti di stringhe dell'alfabeto. Un linguaggio  $L$  è NP-completo se  $L$  si trova in NP, ed  $L' \leq_p L$  per ogni linguaggio  $L'$  in NP.*

I problemi NP-completi sono intrattabili, nel senso precedentemente descritto, [12].

Immaginiamo ad esempio di aver determinato i pesi  $\omega_1^*, \omega_2^*, \dots, \omega_n^*$  di una rete neurale, uniche incognite di un problema di apprendimento. Allora si può ottenere dalla rete, con una sola stimolazione, un output opportuno e confrontarlo con il risultato atteso. Il numero dei passi computazionali per la verifica del risultato è direttamente proporzionale al numero  $n$  di variabili e al numero  $m$  di ipotesi disgiuntive ( $w_i = \omega_i^* \vee w_i \neq \omega_i^*$ ), che è legato a sua volta ad una espressione polinomiale in  $n$ , per ipotesi aggiuntive, [2], che qui si tralasciano per brevità. Il tempo richiesto per verificare una istanza è perciò legato ad un polinomio in  $n$ . Dunque il problema, del tutto generale, di tale apprendimento appartiene sicuramente alla classe NP. Il passo finale, cioè la dimostrazione dell'appartenenza di tale problema alla classe di NP-completezza, è legato alla possibilità di ridurre altri problemi NP-completi in termini di apprendimento di una rete, [Jud90] e [2].

Si è comunque portati a pensare che al diminuire del numero di pesi la complessità temporale del problema diminuisca, ma non è così in generale. È stato infatti dimostrato che l'addestramento di una rete con solo tre unità può addirittura essere NP-completo, per alcuni insiemi di addestramento, [Nat91].

Si noti che il problema sin qui analizzato viene riportato in letteratura come *apprendimento supervisionato*. Esistono tuttavia anche il problema dell'*apprendimento con rinforzo*, dove la rete si comporta come un agente in grado di imparare dall'ambiente circostante per mezzo di un processo di prova ed errore, ed il problema dell'*apprendimento non supervisionato*, in cui la rete non riceve alcuna informazione dall'ambiente sui valori desiderati in risposta ma estrapola autonomamente le regolarità statistiche dei dati di input ed acquisisce la possibilità di individuare correlazioni in nuovi set di dati. Tale classificazione è puramente indicativa.

Si parla anche di *complessità spaziale*, secondo il *numero di celle* scandite da M. Si possono definire, analogamente a quanto già mostrato, la classe PSPACE di linguaggi che impiegano spazi polinomiali e la sua controparte non deterministica, NPSPACE. In termini di spazio e tempo impiegati sussistono le seguenti inclusioni:

$$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME \quad (1.13)$$

Si è dimostrato formalmente che  $P \neq EXPTIME$ . Esistono tutt'oggi problemi intrattabili, in linea teorica risolvibili ma troppo dispendiosi per spazio e tempo richiesti, anche per i relativamente potenti elaboratori odierni. Non solo. La tendenza generale è quella di sottovalutare l'aspetto computazionale, relegandolo al semplice calcolo, sebbene esso includa problematiche che potrebbero avere risvolti rivoluzionari per il panorama del formalismo matematico e fisico. Ad esempio se  $P = NP$ , allora *un computer troverebbe una prova formale di un teorema*, [3], avente una dimostrazione di ragionevole lunghezza, essendo tali dimostrazioni verificabili in tempo polinomiale.

## 2 Reti ricorrenti e Reti di Hopfield

Perchè, cosa potrebbe essere più soddisfacente di essere in grado di andare [...] in venti o trenta città differenti e prendere un telefono ed essere ricordato e amato ed aiutato da così tante persone?

---

A. MILLER, MORTE DI UN COMMES-  
SO VIAGGIATORE

## 2.1 Breve Introduzione alla Teoria di Lyapunov

In presenza di ipotesi di base interagenti di alcuni problemi, la struttura unidirezionale intrinseca delle reti feedforward è fortemente limitante, [1]. È possibile tuttavia partire da essa e creare reti in cui i risultati della computazione di una unità influiscano sul processo computazionale delle altre. Appare evidente che gli algoritmi che manovrano le dinamiche di tali reti debbano soddisfare nuovi criteri di convergenza.

La rete neurale verrà ora inquadrata, in maniera del tutto generale, come un *sistema dinamico*, ossia un sistema in grado di evolversi nel tempo, [1]. Sia  $\underline{x}(t) \in \mathbb{R}^n$  il vettore che rappresenta lo stato di un sistema dinamico al tempo  $t$  e sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  la funzione che ne descrive il campo vettoriale. Allora la *dinamica* del sistema è definita dall'equazione differenziale autonoma:

$$\dot{\underline{x}}(t) = \frac{d\underline{x}}{dt} = f(\underline{x}) \quad (2.1)$$

Si definisce *traiettoria* di un sistema dinamico la legge che assegna ad ogni istante uno stato del sistema. L'insieme di tali traiettorie individua il comportamento possibile del sistema in uno spazio  $2n$  dimensionale, detto<sup>1</sup> *spazio delle fasi*. Le varie traiettorie non si incrociano, poiché ad ogni istante di tempo il sistema può trovarsi in uno e un solo stato. Uno stato si dice *stabile* se la dinamica del sistema converge a tale stato, mentre invece si parla di stato *instabile* se la dinamica non converge ad esso. C'è una terza eventualità: si dice che uno stato è un *punto di sella* se il sistema converge ad esso ma può ancora divergere per piccole perturbazioni. Tali punti vengono anche definiti *semistabili*.

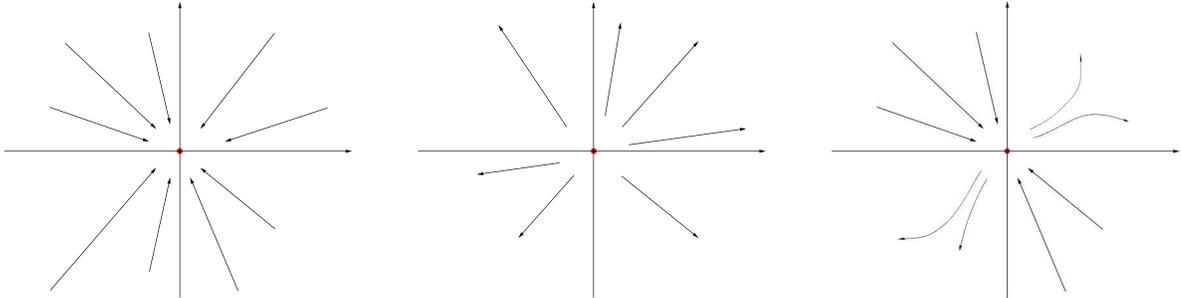


Figura 2.1: Da sinistra verso destra, rispettivamente, una configurazione stabile, instabile e punto di sella.

La proprietà di stabilità implica che  $f(\underline{x}) = \underline{0}$ . Si dice a tal proposito che lo stato di equilibrio  $\underline{x}_0$  è *asintoticamente stabile* se esiste un intorno  $\varepsilon(\underline{x}_0)$  tale che per  $t \rightarrow \infty$  allora  $\underline{x}(t) \rightarrow \underline{x}_0 \forall \underline{x}(0) \in \varepsilon(\underline{x}_0)$ .

Sin qui non è stato detto nulla di nuovo ma ora si può fare un passo aggiuntivo, legando il concetto di stabilità ad un nuovo concetto fondamentale, noto alla teoria della fisica matematica: *le funzioni di Lyapunov*.

Si dice che una funzione  $L$ , continua, insieme alle sue derivate prime, nel suo dominio di definizione  $D$ , è una *funzione di Lyapunov candidata* se valgono le condizioni seguenti:

$$\mathbf{L1)} \ L(\underline{x}) : D \subset \mathbb{R}^n \rightarrow \mathbb{R} \quad \mathbf{L2)} \ L(\underline{0}) = 0 \quad \mathbf{L3)} \ L(\underline{x}) > 0 \ \forall \underline{x} \neq \underline{0}.$$

<sup>1</sup>Nello spazio delle fasi, l'evoluzione del punto rappresentativo del sistema in esame è descritta da  $2n$  variabili, ad esempio  $x$  e  $\dot{x}$ .

Inoltre una funzione  $L(\underline{x})$  è una *funzione di Lyapunov per un sistema dinamico* se essa è una funzione di Lyapunov candidata e  $\dot{L} < 0 \forall \underline{x} \neq \underline{0}$ .

Sulla base di queste definizioni si verifica che un sistema dinamico è *asintoticamente stabile nell'origine*  $\underline{0}$  se il sistema ha una funzione di Lyapunov nel senso<sup>2</sup> di cui sopra.

Si osservi che  $\dot{L}$  è

$$\dot{L} = \frac{dL}{dt} = \left( \frac{dL}{d\underline{x}} \right) \left( \frac{d\underline{x}}{dt} \right) = \nabla_{\underline{x}} L \cdot f(\underline{x}) \quad (2.2)$$

dunque la condizione  $\dot{L} < 0$  spinge l'evoluzione del sistema dinamico verso stati ai quali la funzione di Lyapunov associa valori sempre minori.

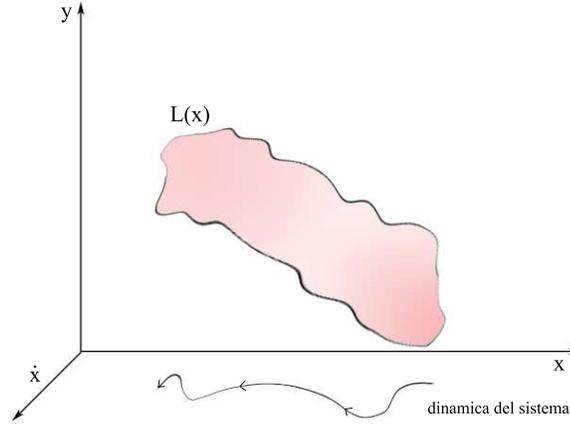


Figura 2.2: Visualizzazione grafica della superficie della funzione di Lyapunov  $L(\underline{x})$ .

Tale risultato è estremamente importante, poiché indica che la stabilità può essere provata senza conoscere effettivamente l'energia fisica del sistema, a condizione che esista una funzione di Lyapunov che soddisfi i vincoli suddetti. In realtà, in generale, per un sistema dinamico esistono infinite funzioni di Lyapunov. Il criterio in sé inoltre è *sufficiente* ma non *necessario*: la funzione di Lyapunov potrebbe non esistere ma un punto potrebbe continuare ad essere un punto stabile.

Più avanti si analizzeranno sistemi a tempo e stati discreti, per i quali la dinamica è definita dall'equazione

$$\underline{x}(t+1) = f(\underline{x}(t)) \quad (2.3)$$

dove  $\underline{x}(t)$  può assumere solo *un insieme finito* di valori. In questo caso  $L : \mathbb{R}^n \rightarrow \mathbb{R}$  è una *funzione di Lyapunov per il sistema* se  $L(\underline{x}(t+1)) < L(\underline{x}(t))$ ,  $\forall \underline{x}(t) \neq 0$ , [1].

## 2.2 Reti di Hopfield

In un suo articolo del 1982, [10], il fisico J. J. Hopfield propose un nuovo modello di rete neurale, ispirato ai principi di funzionamento dei neuroni di McCulloch e interessato ad una proprietà di alcuni sistemi complessi: l'insorgenza di fenomeni collettivi, inspiegabili in termini dei singoli costituenti del sistema fisico. Saranno di seguito presentate due varianti delle

<sup>2</sup>La proposizione appena fornita è l'enunciato del secondo *teorema sulla stabilità di Lyapunov*, [Hir04].

cosiddette *reti di Hopfield*, le cui unità costituenti restano relativamente semplici e la cui dinamica, in quanto sistema fisico, verrà delineata alla luce della teoria di Lyapunov.

### 2.2.1 Reti di Hopfield discrete

L'architettura generale di una *rete di Hopfield discreta* può essere riassunta tramite il seguente grafo:

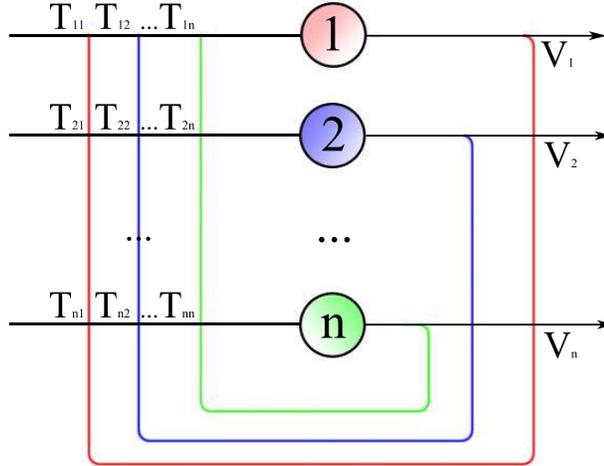


Figura 2.3: Architettura di una rete di Hopfield discreta. Con i numeri si sono indicate le varie unità computazionali. Si noti come ognuna di queste sia collegata alle altre per mezzo di una intricata serie di connessioni.

Si fa l'ipotesi che le connessioni che uniscono i nodi  $i$  e  $j$  abbiano le medesime intensità in entrambe le direzioni. Dunque la matrice dei pesi  $\mathbf{T}$  è a sua volta simmetrica:

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} & \dots & T_{1n} \\ T_{21} & T_{22} & \dots & T_{2n} \\ \dots & \dots & \dots & \dots \\ T_{n1} & T_{n2} & \dots & T_{nn} \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} & \dots & T_{1n} \\ T_{12} & T_{22} & \dots & T_{2n} \\ \dots & \dots & \dots & \dots \\ T_{1n} & T_{2n} & \dots & T_{nn} \end{pmatrix} \quad (2.4)$$

Poiché come si è detto, per determinare l'architettura di una rete neurale non basta solo definirne le connessioni topologiche e i pesi associati, si precisa anche la funzione di attivazione delle unità della rete, che in questo caso è una funzione binaria a soglia. Indicati allora con  $V_i$ ,  $U_i$  e  $I_i$  rispettivamente l'output, la soglia e l'input dell' $i$ -esimo neurone, si ha:

$$V_i = \begin{cases} 0 & \text{se } \sum_{j \neq i}^n T_{ij} V_j + I_i < U_i \\ 1 & \text{altrimenti} \end{cases} \quad (2.5)$$

Se da una parte la funzione di attivazione ricorda il perceptrone, precedentemente analizzato, dall'altra si sottolinea che nelle reti di Hopfield discrete le unità computazionali interagiscono tra di loro in modo *asincrono*: ad ogni istante di tempo  $t$ , la rete seleziona in modo casuale

un'unità, che, secondo la propria soglia, può essere o meno disponibile ad interagire con l'ambiente per modificare il proprio stato in base all'input in ingresso.

Si può introdurre la seguente funzione energia per caratterizzare il tipo di rete in esame,

$$E = -\frac{1}{2} \sum_{i \neq j} T_{ij} V_i V_j - \sum_i V_i I_i + \sum_i V_i U_i \quad (2.6)$$

che si dimostra essere una funzione di Lyapunov della rete neurale, secondo quanto detto per i sistemi discreti alla fine del precedente paragrafo. Infatti  $E$  assume un numero finito di valori, essendo finito il numero dei possibili stati della rete (ecco pertanto spiegata la dicitura "discreta"). Inoltre, vista la finitudine suddetta, essa è evidentemente limitata inferiormente. Il primo termine è l'analogo di un termine di energia di interazione binaria (il fattore  $\frac{1}{2}$  evita di contare due volte gli stessi nodi) tra le unità della rete, mentre il secondo e il terzo termine tengono conto della possibile presenza di influenze esterne al sistema (l'analogia potrebbe essere con l'interazione con un campo esterno).

Si può mostrare che per variazioni unitarie del tempo, il sistema converge ad un minimo, locale o meno, di  $E$ .

Sia ora  $\Delta V = V_i(t) - V_i(t-1)$ , dove  $i$  è l'unità selezionata per l'aggiornamento al tempo  $t$ . I due stati in differenza sono diversi solo se  $V_i(t-1) = 0$  ma  $V_i(t) = 1$ , oppure se  $V_i(t-1) = 1$  e  $V_i(t) = 0$ . Dunque  $\Delta V$  potrà assumere solo i valori -1, 0 o 1. Discriminiamo allora tre casi.

Se  $\Delta V = 0$ , l'unità  $i$ -esima non avrà cambiato il proprio stato, quindi l'energia della rete sarà rimasta immutata.

Se  $\Delta V = 1$ , ad  $E$  si aggiungeranno nuovi termini e la variazione di energia, definita in analogia a quella di  $V$ , sarà  $\Delta E = U_i - I_i - \frac{1}{2} \sum_{j \neq k} T_{kj} V_k V_j$ , dove tuttavia o  $k$  o  $j$  possono assumere il ruolo dell'indice  $i$ . Dalla simmetria della matrice dei pesi  $\mathbf{T}$ , si ricava allora che  $\Delta E = U_i - I_i - \sum_{j \neq i} T_{ij} V_j = - \left( \sum_{j \neq i} T_{ij} V_j + I_i - U_i \right) \Delta V$ . Possiamo esprimerci sul segno di questa espressione, infatti, proprio poiché  $\Delta V = 1$ , il termine tra parentesi dovrà essere positivo, secondo quanto detto sulla funzione di attivazione delle unità computazionali. Dunque la variazione di energia sarà una quantità negativa.

Se  $\Delta V = -1$ , si procede esattamente come nel caso di prima, giungendo a scrivere  $\Delta E = -U_i - I_i + \sum_{j \neq i} T_{ij} V_j = - \left( \sum_{j \neq i} T_{ij} V_j + I_i - U_i \right) \Delta V$ . Tuttavia in questo caso è necessario fare una ipotesi aggiuntiva, visto che le condizioni sulla funzione di attivazione sono più restrittive. Si tratta in particolare della richiesta che il generico termine  $T_{ii} \geq 0$ . Con tale aggiunta  $\left( \sum_{j=1}^n T_{ij} V_j + I_i - U_i \right) < 0 \Rightarrow \left( \sum_{j \neq i} T_{ij} V_j + I_i - U_i \right) < 0$ , e anche qui, in definitiva,  $\Delta E < 0$ .

Dunque si è informalmente dimostrato che  $\Delta E < 0$  per ogni aggiornamento non nullo di una unità computazionale. Ciò comporta che la funzione  $E : \mathbb{R}^n \rightarrow \mathbb{R}$  è una funzione di Lyapunov per una rete di Hopfield a stati e a tempo discreti, avente una matrice dei pesi con elementi diagonali positivi o nulli. Ciò garantisce la convergenza della rete ad uno stato di equilibrio, che corrisponde ad un minimo locale della funzione energia. Si noti che nonostante la semplicità della verifica di questa osservazione non si hanno condizioni aggiuntive sul grado di convergenza, al momento.

### 2.2.2 Reti di Hopfield continue

A differenza di quelle discrete, le *reti di Hopfield continue* lavorano in modo *sincrono*, cioè tutte le unità si attivano contemporaneamente ad ogni istante di tempo, [9] e [13].

La struttura generale della rete è riportata di seguito:

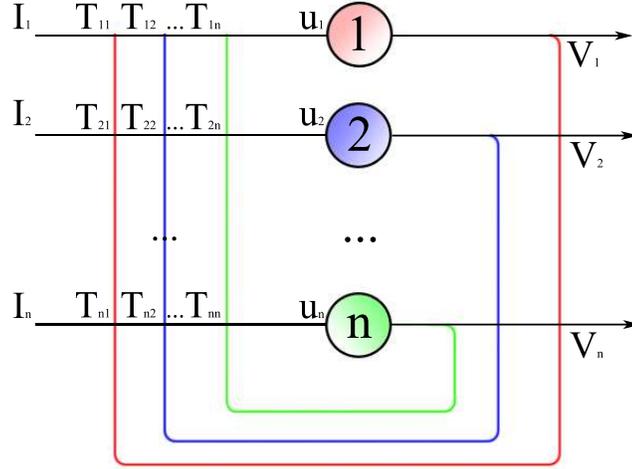


Figura 2.4: Architettura di una rete di Hopfield continua.

Indicate con  $g$  la funzione di attivazione delle unità, cioè la funzione logistica riportata nella (1.10), e con  $u_i$  l'input totale dell'unità  $i$ -esima, allora, [9] e [13], la dinamica del sistema è data da:

$$C_i \frac{\partial u_i}{\partial t} = \sum_j T_{ij} V_j - \frac{u_i}{R_i} + I_i \quad \text{dove} \quad V_i = g(u_i) \quad (2.7)$$

In termini di circuiti,  $T_{ij} V_j$  rappresenta la corrente elettrica in ingresso al nodo  $i$  dovuta al potenziale presente nel nodo  $j$ ;  $C_i$  ed  $R_i$  sono due costanti positive dette *capacità* e *resistenza* del nodo  $i$ , che richiamano la realizzazione elettronica di questa rete attraverso circuiti operazionali;  $u_i$  rappresenta il potenziale medio del neurone  $i$  e  $g$  il termine di amplificazione in tensione, considerato  $V_i$  come il potenziale in uscita del nodo  $i$ . In effetti, tenuto conto che nel membro a sinistra della prima uguaglianza si hanno solo correnti, trascurando i contributi additivi della corrente di ingresso  $I_i$  e della sommatoria, si può rivedere l'equazione come una semplice equazione di un circuito  $RC$  per ogni singolo nodo, infatti  $C_i = \frac{Q_i}{V_i} \rightarrow C_i V_i = \frac{\int V_i dt}{R_i}$ , con i dovuti segni relativi alle correnti di entrata.

Si dimostra, [9], che una funzione di Lyapunov associata a tale sistema dinamico è:

$$E = -\frac{1}{2} \sum_i \sum_j T_{ij} V_i V_j + \sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(\eta) d\eta - \sum_i I_i V_i \quad (2.8)$$

In questo caso  $V_i \in (-1, 1)$ . Inoltre sempre su tale intervallo,  $g^{-1}$  è limitata inferiormente. Ne consegue che la funzione energia è limitata e continua nel suo intervallo di definizione, limitato, dunque ammette minimo. Per verificare che la rete converge sempre ad uno stato di equilibrio, verifichiamo ora che  $\dot{E} < 0$ . In effetti si ha che:

$$\dot{E} = -\frac{1}{2} \sum_i \sum_j T_{ij} V_i \dot{V}_j - \frac{1}{2} \sum_i \sum_j T_{ij} \dot{V}_i V_j + \sum_i \frac{1}{R_i} \left[ g^{-1}(\eta) \right]_0^{V_i} \dot{V}_i - \sum_i I_i \dot{V}_i \quad (2.9)$$

Ma questa espressione può essere scritta in modo più compatto, poiché la matrice  $\mathbf{T}$  è simmetrica e inoltre  $g^{-1}(0) = 0$ , pertanto:

$$\dot{E} = -\sum_i \sum_j T_{ij} V_i \dot{V}_j + \sum_i \frac{1}{R_i} \left( g^{-1}(V_i) \right) \dot{V}_i - \sum_i I_i \dot{V}_i = -\sum_i \dot{V}_i \left[ \sum_j T_{ij} V_j - \frac{1}{R_i} g^{-1}(V_i) + I_i \right] \quad (2.10)$$

Inoltre, ricordando le espressioni che regolano la dinamica del sistema, citate nella (2.7), si ha:

$$\dot{E} = -\sum_i \dot{V}_i [C_i \dot{u}_i] \quad (2.11)$$

Possiamo ricavare ulteriore informazione dalla relazione  $u_i = g^{-1}(V_i)$ , scrivendo:

$$\dot{u}_i = \frac{dg^{-1}}{dt} = \frac{dg^{-1}}{dV_i} \frac{dV_i}{dt} = \frac{dg^{-1}(V_i)}{dV_i} \dot{V}_i \quad (2.12)$$

e giungendo infine alla seguente scrittura di  $\dot{E}$ :

$$\dot{E} = -\sum_i \dot{V}_i^2 C_i \frac{dg^{-1}(V_i)}{dV_i} \quad (2.13)$$

Ogni singolo addendo della sommatoria è una quantità positiva, poiché le capacità sono costanti positive e l'inversa della funzione logistica è una funzione monotona crescente in  $V_i$ . Dunque l'intera sommatoria è una quantità positiva, ne sovviene che  $\dot{E} < 0$ .

È stato pertanto dimostrato, in maniera informale, che  $E$  è una funzione di Lyapunov del sistema dinamico costituito da una rete di Hopfield continua, sistema che converge sempre ad uno stato di equilibrio. Si noti che anche qui non sono date informazioni utili sul grado di convergenza e sull'individuazione del minimo, locale o meno, al quale tende la funzione energia del sistema, [2].

## 2.3 Reti di Hopfield e Sistemi Fisici: l'Emergenza di Proprietà Collettive

Da quanto detto sinora è evidente una particolarità delle reti di Hopfield: nonostante la relativa semplicità della struttura di una singola unità computazionale, il cui stato può essere al più 0 o 1 o -1 secondo poche e prestabilite condizioni, la rete costituita dall'insieme di tali unità dà origine a fenomeni del tutto nuovi, quali il decremento della funzione energia del sistema nel tempo. Il medesimo fenomeno si ritrova in molti sistemi fisici, [10]. L'emergenza di nuove proprietà collettive e macroscopiche dalle interazioni di moltissimi elementi microscopici e piuttosto semplici è una caratteristica dei cosiddetti *sistemi complessi*, tra i quali ad esempio i domini magnetici nei materiali ferromagnetici o i vortici di alcuni tipi di fluidi.

Consideriamo ora un sistema fisico descritto dal vettore  $\underline{x} = (x_1, x_2, \dots, x_n)$  dotato di punti asintoticamente stabili  $\underline{x}_a, \underline{x}_b, \dots$ . Gli stati cui le eventuali dinamiche del sistema convergono,

nello spazio delle fasi, vengono detti *attrattori*. A tali attrattori si può associare una dimensione  $d_A$ ; cosicché ad esempio un attrattore formato da un punto abbia  $d_A = 0$  mentre un attrattore costituito da una traiettoria avrà  $d_A = 1$ , [11]. In caso di più attrattori, il sistema può convergere ad uno di essi in maniera più o meno spiccatamente dipendente dalle condizioni iniziali: se ad esempio il sistema fisico parte dal punto  $x = x_a + \Delta$ , con  $\Delta$  sufficientemente “piccolo”, dopo un certo periodo di tempo esso si sarà portato ad uno stato prossimo a quello di stabilità  $x' \approx x_a$ . In termini più rigorosi si può dire che la configurazione iniziale del sistema viene trasformata dalla dinamica e portata in uno stato di equilibrio. L'intero spazio delle configurazioni può allora essere pensato come diviso in *bacini di attrazione*, cioè insiemi di stati iniziali accomunati dallo stesso attrattore. Si noti comunque che questa suddivisione è prettamente schematica, poiché la divisioni tra bacini di attrazione differenti non sono ben distinte. Difatti tale flusso nello spazio delle fasi non è prettamente deterministico ed il sistema può rispondere a condizioni iniziali ambigue con una scelta statistica del punto di equilibrio cui convergere. Ciò è alla base della non linearità effettiva della rete.

Si noti ora che tutte queste conseguenze, relative alla dinamica del sistema fisico in esame, sono esattamente applicabili ad una rete neurale di Hopfield. Tale parallelismo è veramente impressionante. In effetti è intuitivo riconoscere che gli stati di un rete di Hopfield discreta e formata da  $n$  nodi possono essere rappresentati con il vettore binario  $\underline{x}$ , la cui  $i$ -esima componente può essere al più 0 o 1, a seconda dello stato delle unità computazionali. C'è inoltre una corrispondenza biunivoca tra tale rappresentazione e i vertici di un ipercubo di lato unitario, per quanto detto sulle le funzioni booleane del capitolo precedente. In analogia a ciò gli stati di una rete continua di Hopfield con  $m$  nodi possono essere rappresentati come i punti interni ( $V_i$  in questo caso varia con continuità da -1 a 1) ad un ipercubo di dimensione  $m$ .

In base a tali rappresentazioni si può dire che gli stati stabili di una rete di Hopfield discreta e, in linea di massima, quelli di una rete di Hopfield continua corrispondono ai particolari vertici di un ipercubo che minimizzano la funzione energia della rete. Nella seconda tipologia di reti, tuttavia, nel caso in cui  $\mathbf{T}$  è definita positiva o negativa, alcuni punti stabili possono trovarsi all'interno dell'ipercubo stesso. Nel caso particolare in cui la somma dei termini integrali della funzione energia continua sia trascurabile, i punti stabili di una rete di Hopfield continua possono essere messi in relazione biunivoca con quelli dell'analogo discreto, [9,10,13]. Si noti un particolare strabiliante: il passaggio da continuo a discreto non altera la presenza di stati stabili e dunque il comportamento emergente delle reti, nonostante la loro modellizzazione sia differente.

È evidente una forte ispirazione rispetto ai concetti fisici sulla dinamica di un sistema e sulla fisica degli attrattori, che permette di costruire, attraverso una rete di Hopfield, una *memoria associativa*. Cosa accadrebbe se il vettore iniziale dell'esempio precedente,  $x = x_a + \Delta$ , rappresentasse una conoscenza parziale e affetta da errore dello stato  $x_a$ ? A patto di avere un  $\Delta$  opportuno, la rete di Hopfield o equivalentemente il sistema fisico, verrebbe guidato proprio nell'attrattore  $x_a$ . Una memoria associativa non è altro che una corrispondenza biunivoca tra vettori di informazione, detti genericamente *pattern*, e stati stabili di un sistema, di modo che portando tale sistema in una configurazione iniziale opportuna esso evolva nel tempo verso l'attrattore più vicino, il cui stato venga poi prontamente decodificato. Tale memoria si basa esclusivamente sul tipo di informazione adoperata e non sulla sua ubicazione fisica, identificata magari da un indirizzo di memoria.

Inserendo in una memoria associativa l'input “Fiorinni”, nonostante l'evidente errore di battitura, si potrebbero richiamare informazioni relative alla Facoltà di Fisica di Lecce.

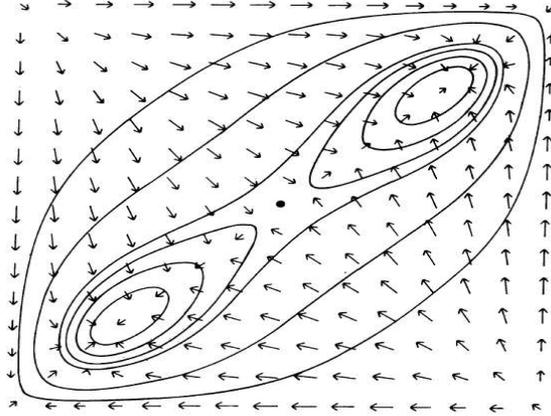


Figura 2.5: Questa immagine è stata tratta da [9]. Si tratta della visualizzazione dell'energia di una rete continua a due stati stabili, formata da due soli neuroni. Sui due assi si trovano gli output dei neuroni. Gli stati stabili sono indicati dalle frecce della dinamica del sistema. Nella zona centrale si osserva un punto di sella.

Le memorie associative rappresentano una applicazione interessante delle reti di Hopfield, queste ultime presentate alla comunità scientifica da John J. Hopfield, medaglia Dirac, in una sua pubblicazione del 1982, [10]. Nel medesimo articolo furono rimosse alcune restrizioni fatte sin qui: una rete con una matrice dei pesi  $\mathbf{T}$  non simmetrica, ma con il peso  $T_{ij}$  scelto stocasticamente con media 0 tra i valori limite -1 e 1, presenta ancora stati stabili ed è adatta alla modellizzazione per le memorie associative. Sotto tali ipotesi, tuttavia, non è esclusa la presenza anche di punti metastabili.

Attraverso alcune simulazioni di Montecarlo per  $N=30$  e  $N=100$ , [10], si è visto per estrapolazione che per una rete di  $N$  unità, possono essere memorizzati all'incirca  $\frac{3}{20}N$  stati. L'aggiunta di nuovi stati di memoria sovraccarica il sistema e rende impossibile il recupero di informazioni. Inoltre lo stato corrispondente al vettore nullo è l'unico sempre stabile, indifferentemente da  $N$ . Per  $N=30$  e 5 stati da memorizzare, l'85% delle prove fornisce il risultato corretto, il 10% dà in uscita un errore ed il 5% fornisce un risultato ambiguo, derivante dalla scelta statistica tra due attrattori.

## 2.4 La regola di Hebb e sua generalizzazione

Come è possibile suddividere l'intero spazio delle fasi in diversi bacini di attrazione? Una strada è quella di introdurre una certa non linearità nella attivazione dei neuroni della rete, [1] e [2], di modo che l'intero sistema dinamico diventi non lineare. Si utilizza per tale scopo la *funzione segno*  $\text{sgn}(x)$ , con  $x \in \mathbb{R}$ . Consideriamo una rete monostrato di  $k$  unità con tale funzione di attivazione bipolare. L'obiettivo è trovare i pesi appropriati per mappare il vettore di input  $\underline{x}$   $n$ -dimensionale nel vettore di output  $k$ -dimensionale  $\underline{y}$  e immagazzinare così l'informazione necessaria. L'aggiornamento più semplice e sufficientemente efficace in questo caso è una modellizzazione della *regola di Hebb*, [Heb49]: *due neuroni che sono simultaneamente attivi sviluppano un grado di interazione superiore a quella, trascurabile, dei neuroni le cui attività sono scorrelate nel tempo*. Nel caso di una rete ricorrente e votata all'associazione si parla di *apprendimento di Hebb* per indicare l'aggiornamento dei pesi di un incremento  $\Delta w_{ij}$

che sia proporzionale alle correlazioni tra gli stati  $x_i$  e  $x_j$  dei neuroni<sup>3</sup>  $i$  e  $j$  in interazione

$$\Delta w = \gamma x_i x_j \quad (2.14)$$

secondo una costante di apprendimento  $\gamma$ .

La matrice dei pesi  $\mathbf{T}$  della rete viene inizializzata a zero prima dell'apprendimento di Hebb, invece la matrice dei pesi aggiornata, indicata con  $\mathbf{W}$ , è la matrice di correlazione, di dimensione  $n \times k$ , tra il vettore di input  $\underline{x}^1$  e quello di output  $\underline{y}^1$  ed ha la forma

$$\mathbf{W} = [w_{ij}]_{n \times k} = [x_i^1 y_j^1]_{n \times k} \quad (2.15)$$

$\mathbf{W}$  stabilisce una relazione biunivoca o *mappa* il vettore riga non nullo di input  $\underline{x}^1$  esattamente nel vettore riga<sup>4</sup> di output  $\underline{y}^1$ . Ciò si può vedere dall'eccitazione delle  $k$  unità di output della rete, argomento della funzione di attivazione, che è data da

$$\underline{x}^1 \mathbf{W} = \left( y_1^1 \sum_{i=1}^n x_i^1 x_i^1, y_2^1 \sum_{i=1}^n x_i^1 x_i^1, \dots, y_k^1 \sum_{i=1}^n x_i^1 x_i^1 \right) = \underline{y}^1 (\underline{x}^1 \cdot \underline{x}^1) \quad (2.16)$$

Per  $\underline{x}^1 \neq \underline{0}$  si ha che la parte tra parentesi è strettamente maggiore di zero e l'output della rete diventa

$$\text{sgn}(\underline{x}^1 \mathbf{W}) = (y_1^1, y_2^1, \dots, y_k^1) = \underline{y}^1 \rightarrow \underline{y}^1 = \text{sgn}(\underline{x}^1 \mathbf{W}) \quad (2.17)$$

La condizione a destra della freccia nella formula di cui sopra viene chiamata *condizione di stabilità* del pattern  $\underline{y}^1$ : se si inizializza l'intero sistema fornendo in input un pattern stabile, esso non cambia stato. Si può generalizzare la regola di Hebb. Infatti, per associare  $m$  vettori  $n$ -dimensionali non nulli,  $\underline{x}^1, \underline{x}^2, \dots, \underline{x}^m$  ad  $m$  vettori  $k$ -dimensionali  $\underline{y}^1, \underline{y}^2, \dots, \underline{y}^m$ , si deve applicare l'apprendimento di Hebb ad ogni coppia ingresso-uscita e la matrice dei pesi risultante è la somma delle matrici di correlazione  $n \times k$  dei vettori  $\underline{x}^h$  e  $\underline{y}^h$ . Se l'input della rete è il vettore  $\underline{x}^p$ , con  $1 < p < m$ , l'eccitazione finale è

$$\underline{x}^p \mathbf{W} = \underline{x}^p (\mathbf{W}^1 + \mathbf{W}^2 + \dots + \mathbf{W}^m) = \underline{x}^p \mathbf{W}^p + \sum_{h \neq p}^m \underline{x}^p \mathbf{W}^h = \underline{y}^p (\underline{x}^p \cdot \underline{x}^p) + \sum_{h \neq p}^m \underline{y}^h (\underline{x}^h \cdot \underline{x}^p) \quad (2.18)$$

Nell'espressione dell'eccitazione ora compare un termine addizionale,  $\sum_{h \neq p}^m \underline{y}^h (\underline{x}^h \cdot \underline{x}^p)$ , detto *diafonia* o *crosstalk*. Si tratta di un contributo perturbativo che è evidentemente nullo se i vettori di input  $\underline{x}^1, \underline{x}^2, \dots, \underline{x}^m$  sono a due a due ortogonali. In tale evenienza la risposta della rete è proprio il vettore ricercato  $\underline{y}^p$ . È ora evidente l'eventualità di un fallimento dell'associazione tra pattern in base allo stato iniziale scelto. In effetti però tale associazione funziona anche nella condizione meno restrittiva che la diafonia sia molto minore della quantità  $\underline{y}^p (\underline{x}^p \cdot \underline{x}^p)$ . L'output della rete in questo caso deve essere:

<sup>3</sup>Vista la funzione di attivazione scelta, cioè la funzione segno, la logica della intera rete sarà bipolare e dunque gli stati possibili di ogni singolo neurone saranno -1 o 1.

<sup>4</sup>L'apice 1 servirà a differenziare questi vettori da altri che verranno utilizzati in seguito.

$$\text{sgn}(\underline{x}^p W) = \text{sgn}\left(\underline{y}^p (\underline{x}^p \cdot \underline{x}^p) + \sum_{h \neq p}^m y^h (\underline{x}^h \cdot \underline{x}^p)\right) = \text{sgn}\left(\underline{y}^p + \sum_{h \neq p}^m y^h \frac{(\underline{x}^h \cdot \underline{x}^p)}{(\underline{x}^p \cdot \underline{x}^p)}\right) = \underline{y}^p \quad (2.19)$$

dove si è tenuto conto della positività della norma al quadrato di  $\underline{x}^p$ . Perché il contributo della diafonia sia piccolo, deve accadere che il valore assoluto di tutte le componenti della somma sia almeno più piccolo di 1. In tal caso l'apprendimento Hebbiano risolverà il problema generale dell'apprendimento per la rete e si avrà stabilità.

Se i bit errati nel pattern di partenza  $\underline{x}$  sono meno della metà, la loro influenza può essere mascherata dalla maggioranza dei bit corretti nel calcolo dell'eccitazione della rete. Il numero di componenti differenti tra due vettori è formalizzato dalla *distanza di Hamming*, [2] e [4], che tra due vettori bipolari  $\underline{x}$  e  $\underline{y}$   $n$ -dimensionali è

$$h = \sum_{i=1}^n \frac{|x_i - y_i|}{2} \quad (2.20)$$

Si dice che una configurazione iniziale vicina allo stato stabile  $\underline{x}_a$ , in termini di  $h$ , converge proprio a tale stato, che è quindi un attrattore.

Si noti infine che in realtà per ogni attrattore  $\underline{x}_a$   $n$ -dimensionale della rete, anche  $-\underline{x}_a$  è un punto stabile di attrazione spurio, detto *stato opposto*, [1,2,10,13], al quale possono convergere tutti i pattern iniziali con una distanza di Hamming da  $\underline{x}_a$  maggiore di  $n/2$ . Dunque, effettivamente, lo spazio delle configurazioni risulta diviso in più bacini di attrazione di quanto non ci si aspetterebbe ed ecco allora che i risultati delle simulazioni numeriche del paragrafo precedente diventano più chiare.

## 2.5 Reti di Hopfield ed il Problema del Commesso Viaggiatore

Dopo aver osservato i principi delle reti di Hopfield è giunto il momento di interrompere l'esposizione di modelli teorici e analizzare una loro interessante applicazione.

Data una particella neutra in un reticolo cristallino, quale è il minimo cammino che essa può percorrere urtando una e una sola volta ogni atomo del cristallo? Si tratta di un problema di ottimizzazione combinatoria, noto come *Traveling Salesman Problem (TSP)* o il *Problema del Commesso Viaggiatore*.

Un problema di ottimizzazione combinatoria non è altro che la ricerca della soluzione *migliore*, appartenente ad un insieme al più numerabilmente infinito di soluzioni, indicato con  $\Omega$  o  $S$ , ad un problema esprimibile sotto forma di una funzione a valori reali, detta *funzione costo*. La notazione classica del TSP è la seguente:

“Un insieme di  $n$  città  $a, b, c, \dots$  ha distanze di separazione  $d_{ab}, d_{ac}, \dots, d_{bc}, \dots$ . Trovare il tour che visiti ogni città una volta e ritorni nella città iniziale, percorrendo la distanza minima totale.” **ossia**

“Dato un grafo pesato completo (dove i vertici rappresentano le città, gli archi le strade e i pesi le distanze tra città) trovare un cammino Hamiltoniano con il peso complessivo minore.”

Si riporta ora il seguente teorema, dimostrato da R. Karp nel 1972:

**Teorema (di NP-Completezza):** Sia  $G=(V,E)$  un generico grafo. Il problema della ricerca di un cammino Hamiltoniano in  $G$  è NP-completo.

Il TSP è quindi un problema NP-completo e computazionalmente estremamente complicato di ottimizzazione combinatoria, in cui la funzione costo è la distanza percorsa e la soluzione migliore è quella con distanza minore. Hopfield e Tank proposero una risoluzione, [13], di questo problema utilizzando una rete neurale continua di Hopfield con  $V_i \in [0, 1]$ .

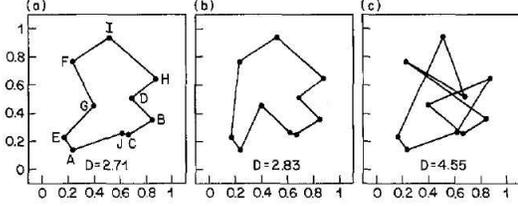


Figura 2.6: L'immagine viene da [13]. (a) Cammino ottimale per un problema TSP con 10 città. (b) Cammino sub-ottimale, sempre con rete di Hopfield continua. (c) Cammino individuato da una rete di Hopfield discreta.

Per la rappresentazione di un tour completo con  $n$  città è necessaria una matrice  $n \times n$  in cui ogni riga rappresenti una città e ogni colonna una fermata del viaggio. Le combinazioni possibili di cammini sono  $n!$ , tuttavia considerando anche l'ordine in cui le città vengono visitate si devono evitare i cammini ripetuti con la medesima distanza percorsa, che ammontano a  $2n$  (a seconda della città di partenza si hanno  $n$  cammini di uguale lunghezza che possono essere percorsi in 2 versi diversi). Pertanto si hanno  $n!/2n$  cammini chiusi di lunghezza distinta.

Introduciamo la notazione seguente per rappresentare le soluzioni del problema:

$$V_{ia} = \begin{cases} 1 & \text{se la città } i \text{ è la } a - \text{esima fermata del tour} \\ 0 & \text{altrimenti} \end{cases} \quad (2.21)$$

Ad ogni fermata, ovviamente, solo una città può essere visitata e dunque le colonne della matrice di cui sopra saranno composte da  $n - 1$  zeri e solo 1 uno. Allo stesso modo, ogni città deve essere visitata solamente una volta, dunque ogni riga della matrice dovrà essere composta da 1 solo uno ed  $n - 1$  zeri.

Quindi sono presenti i seguenti vincoli sul problema:

$$\sum_a V_{ia} = 1 \quad \forall \text{ città } i \quad e \quad \sum_i V_{ia} = 1 \quad \forall \text{ fermata } a \quad (2.22)$$

Perché una eventuale rete computi una soluzione del problema è possibile descrivere questo in termini di una funzione di energia  $E$ , i cui minimi corrispondano alla migliore soluzione.  $E$  deve essere tale da favorire attrattori in cui lo stato continuo del neurone assuma solo il valore 1 o 0, cosicché lo stato della rete coincida *esattamente* con la rappresentazione matriciale delle soluzioni possibili, al termine della minimizzazione della funzione energia. Una parte di  $E$  è:

$$E' = \frac{A}{2} \sum_i \sum_b \sum_{a \neq b} V_{ib} V_{ia} + \frac{B}{2} \sum_b \sum_i \sum_{i \neq h} V_{ib} V_{hb} + \frac{C}{2} \left( \sum_i \sum_b V_{ib} - n \right)^2 \quad (2.23)$$

dove  $A$ ,  $B$  e  $C$  sono quantità positive che determinano il peso dei singoli termini, gli indici  $i$  e  $h$  si riferiscono alle città mentre  $a$  e  $b$  indicano le fermate. La prima somma tripla è nulla se e solo se ogni riga-città soddisfa il vincolo di cui sopra. Stessa cosa accade per la seconda tripla somma, riferendosi alle colonne-fermate. Il terzo termine è zero se e solo se vi sono  $n$  entrate pari ad 1 nell'intera matrice. In questo modo la funzione energia ha valore minimo assoluto  $E = 0$  per tutte le matrici con un solo 1 in ogni riga e colonna, inoltre, una minimizzazione

## 2 Reti ricorrenti e Reti di Hopfield

su tutti e tre i termini tiene conto delle richieste di cui sopra, ossia vincoli e condizione di discretizzazione. Si può inserire in  $E$  un termine che corrisponda alla lunghezza del cammino di un dato tour

$$E'' = \frac{1}{2} D \sum_i \sum_{h \neq i} \sum_b d_{ih} V_{ib} (V_{h,b+1} + V_{h,b-1}) \quad (2.24)$$

dove ancora gli indici  $i$  e  $h$  si riferiscono alle città mentre  $a$  e  $b$  indicano le fermate, inoltre  $D$  è una quantità positiva per determinare il peso del termine di lunghezza e  $d_{ih}$  si riferisce alla distanza tra città, distanza che viene annullata o meno a seconda della città di arrivo  $i$  alla fermata  $b$  e da quella alla fermata precedente o successiva  $h$ .

Una funzione energia appropriata per la rete ricorrente per il TSP consiste nella somma di  $E''$  ed  $E'$ . Se  $A$ ,  $B$  e  $C$  sono sufficientemente grandi, tutti gli stati a bassissima energia della rete descritta da tale funzione avranno la forma di un tour valido e l'energia totale dello stato sarà relativa alla lunghezza del viaggio. Tra questi, gli stati con il percorso più breve avranno anche energia minima. La matrice dei pesi dovrà tenere inoltre conto dell'inibizione nelle connessioni tra ogni riga e in quelle tra ogni colonna e della disposizione spaziale delle città, che determina quale problema TSP risolvere e quale miglior cammino scegliere. La figura esplicativa descrive la struttura topologica della rete TSP fin qui delineata, ad esempio per 4 città:

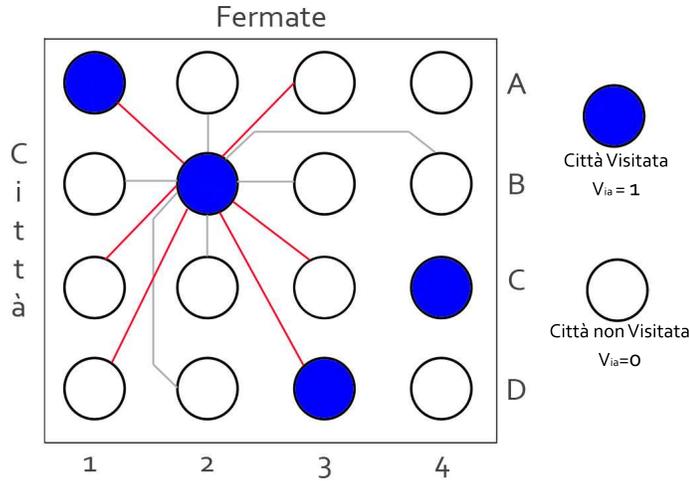


Figura 2.7: Rete per un Problema del Commesso Viaggiatore con  $n=4$  città. La rete ha individuato il percorso A1-B2-D3-C4. Sono mostrate le connessioni inibitorie tra elementi della rete della stessa riga e colonna (in grigio) e alcune delle connessioni inibitorie globali (in rosso). Non sono state riportate tutte le connessioni per evitare confusione.

Detto  $u_{ia}$  il potenziale medio in ingresso all'unità della rete, unità che viene indicata proprio con la notazione  $ia$  della matrice  $n \times n$ , il potenziale in uscita al neurone artificiale  $V_{ia}$  dipenderà anche dalla scelta della funzione di attivazione  $g$ , in cui comparirà anche la costante  $u_0$ , ricavabile, [13], dai vincoli in (2.22):

$$V_{ia} = g(u_{ia}) = \frac{1}{2} \left( 1 + \tanh \left( \frac{u_{ia}}{u_0} \right) \right) \quad \forall a, i. \quad (2.25)$$

La scelta della funzione tangente iperbolica risiede nella volontà di imitare le relazioni tra

ingresso e uscita di un amplificatore reale. Con riferimento alle osservazioni di questo e del precedente paragrafo, la dinamica utilizzata per l'evoluzione del nodo  $ia$  è la seguente:

$$\frac{du_{ia}}{dt} = -u_{ia} - A \sum_{b \neq a} V_{ib} - B \sum_{h \neq i} V_{ha} - C \left( \sum_i \sum_b V_{ib} - n \right) - D \sum_i d_{ih} (V_{i,a+1} + V_{i,a-1}) \quad (2.26)$$

Nel loro articolo, Tank e Hopfield propongono una simulazione di una rete TSP per 10 città, con i parametri fissati  $A = B = D = 500$ ,  $C = 200$ ,  $u_0 = 0.02$  ed  $n = 15$ . Si verifica in effetti che partendo, senza alcuna conoscenza *a priori* della migliore soluzione possibile, ogni cammino ha  $2n$  tour equivalenti che lo descrivono ed il sistema evolve con ambiguità nel tempo, senza convergere ad un particolare tour. L'aggiunta di rumore ai voltaggi di input  $u_{ia}$  della rete *rompe* in qualche modo questa "simmetria" e permette al sistema di convergere verso un viaggio. Pertanto si inserisce una piccola deviazione casuale nella scelta del cammino, in  $u_{ia} = u_{00} + \Delta u_{ia}$ , dove  $u_{00}$  è scelta in modo che  $\sum_{i,a} V_{ia} = n$  all'istante iniziale. La scelta di

opportuni parametri per la rete, come quelli sopra fissati è un compromesso tra l'ottenimento di tour legittimi (D piccolo) e la maggiore esaltazione dell'importanza data alla minimizzazione della distanza percorsa (D grande). Si nota inoltre che per  $u_0$  troppo grandi, i risultati degli stati finali dei  $V_{ia}$  non sono né 1, né 0. Tali stati non sono legittime rappresentazioni di un viaggio (visiteremmo ad esempio contemporaneamente un po' di Barcellona e un po' di Roma alla stessa fermata!). Di contro valori di  $u_0$  troppo piccoli costituiscono una selezione esigua di cammini candidati. Per 10 città, [13], si calcola che vi sono oltre  $\frac{10!}{20} = 181.400$  cammini distinti, la cui distribuzione di lunghezza può essere computata. All'incirca il 50% delle prove, contando convergenze da stati random a stati finali in 20 passi, sono giunti alla *soluzione ottimale*, ossia il tour più breve. È stata studiata una seconda istanza del problema con 30 città ed un insieme di 900 neuroni, cioè all'incirca  $\frac{30!}{60} = 4.4 \times 10^{30}$  cammini. In una sola convergenza, la rete è riuscita ad escludere oltre  $10^{23}$  cammini nella minimizzazione della distanza percorsa.

Il problema principale di questo approccio è che richiede sforzi a livello hardware onerosi: solamente il numero di connessioni necessarie, per un problema con  $n$  città, cresce almeno come  $n^3$ . Inoltre la funzione energia che viene computata può presentare così tanti minimi locali che non sia possibile garantire la convergenza dell'algoritmo di minimizzazione al 100% verso il tour migliore, almeno in tempi polinomiali.

Recentemente è stato risolto un Problema di Commesso Viaggiatore con oltre 1000 città, tramite rete TSP [Tal02]. Degne di nota sono anche altre applicazioni, [26], che impiegano il *simulated annealing*, presentato nel prossimo capitolo. Inoltre, ANN combinate con l'algoritmo euristico *Lin-Kernighan* hanno risolto efficientemente e con una precisione dell'1% istanze del TSP con oltre 1.000.000 di città [Mul03]. Come curiosità si cita la *risoluzione esatta* di un TSP con 85.900 città tramite un *codice di oltre 130.000* righe in linguaggio ANSI C, attraverso la generalizzazione di alcuni algoritmi *esatti* (tra i quali il *cutting-plane*), [App06].

### 3 Simulated Annealing e Macchine di Boltzmann

Ludwig Boltzmann, che passò la maggior parte della propria vita studiando Meccanica Statistica, morì nel 1906, per mano propria. Paul Ehrenfest, portando avanti tali studi, morì in circostanze simili nel 1933. Ora è giunto il nostro momento per studiare Meccanica Statistica. Forse sarebbe opportuno avvicinarsi alla materia con una certa cautela.

---

DAVID L. GOODSTEIN, STATES OF  
MATTER.

### 3.1 Annealing Simulato ed Energia Libera di Helmholtz

L'algoritmo di *simulated annealing* (o *tempra simulata* o *annealing simulato*) si ispira ai processi termici metallurgici in cui dapprima si riscalda un solido sino alla temperatura di fusione, per poi raffreddarlo gradualmente fino a che i suoi atomi non si dispongano in una struttura cristallina, [1,2,26]. Allo stato cristallino, tali atomi sono organizzati nello spazio secondo una disposizione ordinata, detta *reticolo cristallino*, alla quale corrisponde un minimo dell'energia configurazionale del sistema. Si vedrà in seguito che è possibile simulare tale processo di raffreddamento seguendo delle analogie tra un sistema fisico e un problema di ottimizzazione combinatoria, [14,15,16,26], dove agli stati del primo corrispondono le soluzioni del secondo e dove il costo di ogni soluzione equivale all'energia di uno stato.

Ricordiamo il concetto di *energia libera di Helmholtz* per un sistema fisico

$$F = E - TS \quad (3.1)$$

dove  $E$  ne è l'energia in senso termodinamico,  $T$  ne è la temperatura ed  $S$  ne è l'entropia, definita, per un sistema a stati discreti, dalla *formula di Boltzmann*

$$S = -\sum_i p_i \ln(p_i) \quad (3.2)$$

con  $p_i$  intesa come la *probabilità* che il sistema si trovi nello stato  $i$  con energia  $E_i$ . È possibile utilizzare la minimizzazione di  $F$  per formalizzare la dinamica di un sistema fisico. In particolare, per ridurre al minimo tale grandezza occorre minimizzare l'energia media e simultaneamente massimizzare l'entropia del sistema, come risulta evidente dalla formula di cui sopra. Queste due evenienze accadono in condizioni differenti: infatti l'entropia  $S$  è massima quando tutti gli stati del sistema sono equiprobabili, mentre l'energia  $E$  assume valore minimo quando la sola probabilità associata allo stato avente energia minima è massima. È allora necessario che la minimizzazione dell'energia libera risulti un compromesso tra la minimizzazione dell'energia media e la massimizzazione dell'entropia. In tale discorso la temperatura  $T$  costituisce un parametro che può essere utilizzato proprio per modulare la minimizzazione di  $F$ . Per temperature elevate prevale il termine negativo della formula (3.1), ciò significa che una eventuale minimizzazione di  $F$  renderebbe massima l'entropia e allora la distribuzione di probabilità degli stati del sistema diventerebbe uniforme. Per temperature via via più basse, invece, una minimizzazione di  $F$  comporterebbe una minimizzazione anche dell'energia, determinando una distribuzione di probabilità con un picco in corrispondenza dello stato avente energia minima.

È possibile determinare la distribuzione di probabilità che minimizza  $F$  attraverso il metodo dei moltiplicatori di Lagrange, imponendo il vincolo che la somma di probabilità degli stati del sistema valga 1 e ricordando che l'energia libera è una funzione delle probabilità degli  $M$  stati del sistema. Dunque si ottiene

$$\tilde{F} = F + \lambda \left(1 - \sum_i p_i\right) = E - TS + \lambda \left(1 - \sum_i p_i\right) = \sum_i p_i E_i + T \sum_i p_i \ln(p_i) + \lambda \left(1 - \sum_i p_i\right) \quad (3.3)$$

$$\frac{\partial \tilde{F}}{\partial p_i} = 0 \longrightarrow E_i + T \ln(p_i) + T - \lambda = 0 \longrightarrow p_i = e^{-\frac{E_i}{T} + \frac{\lambda}{T} - 1} \quad (3.4)$$

Sommando su tutti gli stati  $i$  ed utilizzando il vincolo imposto, si può ricavare:

$$1 = e^{\frac{\lambda}{T} - 1} \sum_i e^{-\frac{E_i}{T}} \longrightarrow e^{\frac{\lambda}{T} - 1} = \frac{1}{\sum_i e^{-\frac{E_i}{T}}} \quad (3.5)$$

Sostituendo nell'equazione ricavata al passo precedente, si ricava il seguente risultato:

$$p_i = \frac{e^{-\frac{E_i}{T}}}{\sum_j e^{-\frac{E_j}{T}}} = \frac{e^{-\frac{E_i}{T}}}{Z} = \frac{1}{\sum_j e^{-\frac{E_j - E_i}{T}}} = \frac{1}{1 + \sum_{j \neq i} e^{-\frac{E_j - E_i}{T}}} \quad (3.6)$$

La distribuzione ottenuta è quella che minimizza  $F$ . Essa è dipendente da  $T$  ed è detta *distribuzione di Gibbs*. Inoltre, la funzione  $Z$  è detta *funzione di partizione*.

È immediato ora verificare le affermazioni di cui sopra:

$$\lim_{T \rightarrow \infty} p_i = \lim_{T \rightarrow \infty} \frac{e^{-\frac{E_i}{T}}}{\sum_j e^{-\frac{E_j}{T}}} = \frac{1}{N} \quad \forall i \quad (3.7)$$

$$\lim_{T \rightarrow 0} p_i = \lim_{T \rightarrow 0} \frac{1}{1 + \sum_{j \neq i} e^{-\frac{E_j - E_i}{T}}} = 1 \Leftrightarrow E_i < E_j \quad \forall j \neq i \quad (3.8)$$

La distribuzione di Gibbs, indicata talvolta anche con il nome di *distribuzione di Boltzmann*, regola la densità di probabilità di un *ensemble canonico* (o *insieme statistico canonico*<sup>1</sup>) e permette di trattare le proprietà statistiche di un sistema *in contatto termico* con uno molto più grande (detto, quest'ultimo, *bagno termico* o *sistema ambiente*), dove però i numeri di particelle di entrambi i sistemi si mantengono costanti nel tempo, [2,24,25].

Poiché è noto che ogni distribuzione di probabilità definisce una misura di probabilità<sup>2</sup> è possibile definire la formula (3.6) come un particolare tipo di misura, detta *misura di Gibbs*. È possibile estendere il concetto di misura di Gibbs ad un sistema discreto, [18] e [Geo88], quale un reticolo formato dalle  $N$  unità computazionali di una rete neurale di Hopfield discreta ed associare pertanto ad ogni configurazione del sistema (nel caso di una rete di Hopfield discreta lo spazio degli stati conterrà  $2^N$  configurazioni disposte sui vertici di un ipercubo  $N$ -dimensionale) una probabilità  $\propto e^{-\frac{E_a}{T}}$ , con

$$E_a = -\frac{1}{2} \sum_{ij} w_{ij} S_i S_j - \sum_i I_i S_i \quad (3.9)$$

<sup>1</sup>Un insieme statistico è una costruzione formale che associa ad un dato sistema la collezione di un numero di sue copie, ognuna relativa ad un diverso stato microscopico compatibile con le condizioni macroscopiche prefissate, [24] e [25].

<sup>2</sup>Con misura di probabilità si intende una funzione che assegni agli esiti di un determinato evento la probabilità che essi si realizzino. Una definizione formale di tale misura trova le sue fondamenta teorica nella definizione assiomatica della probabilità. Si rimanda a [18] per una formulazione più esaustiva.

dove  $S_i$  e  $I_i$  sono pari rispettivamente al potenziale di output e di input del nodo  $i$  mentre  $w_{ij}$  è il peso della connessione tra l'unità  $i$  e quella  $j$ .

Da questa generalizzazione si ricava, [18], immediatamente *la misura di Gibbs* di cui sopra, con gli opportuni indici. Non si deriverà formalmente la misura di Gibbs, o distribuzione di Boltzmann, per una ANN ma ne verificheremo l'effettiva stabilità (minimizzazione di  $F$ ) in seguito.

### 3.2 L'Algoritmo di Metropolis, la Macchina di Boltzmann e la Matrice di Markov

La distribuzione di Boltzmann ricavata viene impiegata nell'*algoritmo di Metropolis*, [1,2,7,15,27,29], che iterato e valutato per valori decrescenti del parametro di controllo  $T$  permette di realizzare il *simulated annealing*. Si consideri un sistema i cui stati possibili siano solamente  $s_1$  e  $s_2$  e siano  $E_1$  ed  $E_2$  le rispettive energie. La probabilità che il sistema si trovi in uno dei due stati è data da:

$$P(s_1) = \frac{e^{-\frac{E_1}{T}}}{e^{-\frac{E_1}{T}} + e^{-\frac{E_2}{T}}} = \frac{1}{1 + e^{-\frac{(E_2-E_1)}{T}}} \quad P(s_2) = 1 - P(s_1) = \frac{1}{1 + e^{-\frac{(E_1-E_2)}{T}}} \quad (3.10)$$

Se accade che  $E_1 < E_2$ , allora  $P(s_2) < P(s_1)$ .

L'algoritmo di Metropolis realizza una discesa stocastica del gradiente. Esso fa evolvere il sistema sia accettando con una probabilità 1 le transizioni che diminuiscono il valore dell'energia associata alla rete, sia accettando con una probabilità data dalla distribuzione di Boltzmann quelle che invece ne aumentano l'energia. La condizione di partenza dell'algoritmo vede temperature relativamente alte, che diminuiscono le differenze tra stati in termini di probabilità, come si è precedentemente visto. Dunque, a temperature elevate l'algoritmo può accettare di frequente delle transizioni che aumentano l'energia. Al decrescere di  $T$  tale frequenza diminuisce, in quanto la distribuzione di probabilità diventa più piccata. Quando  $T$  è pari a zero, nessun aumento di energia può essere accettato, visto che allo stato con energia minima è associata la probabilità massima. Con queste caratteristiche l'algoritmo di annealing simulato può *uscire dai punti di minimo locale*.

Si presenta ora il formalismo di una *macchina di Boltzmann*, [14], cioè una rete ricorrente ad  $N$  neuroni avente architettura affine a quella di una rete di Hopfield con il medesimo numero di nodi ed energia associata pari a:

$$E = -\frac{1}{2} \sum_{i \neq j} T_{ij} V_i V_j - \sum_i I_i V_i \quad (3.11)$$

dove  $V_i$  ed  $I_i$  sono rispettivamente il potenziale di output e di input del nodo  $i$  mentre  $T_{ij}$  è il peso della connessione tra l'unità  $i$  e quella  $j$ .

Ad ogni istante di tempo lo *stato corrente*  $s = (V_1, V_2, V_3, \dots, V_N)$  differisce dallo *stato candidato*  $s'$  per il valore di un solo potenziale, scelto a caso da una distribuzione uniforme. Essendo l'attivazione asincrona, se in  $s'$  il valore  $V_i$  dell'unità  $i$  predestinata casualmente all'aggiornamento è pari ad 1, allora si avrà il cambio  $V_i = 1 \rightarrow V_i = 0$ , in caso contrario si avrà invece il cambio  $V_i = 0 \rightarrow V_i = 1$ . Considerando il primo caso, si vuole determinare la probabilità che la rete passi nello stato  $s'$ , in cui  $V_i = 1$ , attraverso la formula (3.10). Da

quanto detto riguardo le differenze tra gli stati, si può calcolare:

$$E' - E = \sum_j T_{ij} V_j + I_i \quad \rightarrow \quad P(s') = \frac{1}{1 + e^{-\frac{\left(\sum_j T_{ij} V_j + I_i\right)}{T}}} \quad (3.12)$$

Dalla definizione di probabilità, visto che gli esiti possibili sono due, cioè  $V_i = 0$  e  $V_i = 1$ , allora  $P(s) = 1 - P(s')$ . Partendo dallo stato corrente si decide casualmente il potenziale di quale unità computazionale cambiare, ciò equivale a fissare lo stato candidato ed anche il salto di energia e la probabilità di effettuare la transizione tra i due stati, secondo la distribuzione di Boltzmann.

Poiché l'aggiornamento di un  $V_i$  casuale definisce una transizione tra gli stati  $s \rightarrow s'$ , si può definire una *matrice di transizione*  $\mathbf{P}_T$  di una macchina di Boltzmann con  $N$  unità come una matrice  $2^N \times 2^N$ , i cui elementi  $P_{ij}$  rappresentano le probabilità di transizione dal generico stato  $s_i$  al generico stato  $s_j$  con un unico aggiornamento e alla temperatura  $T$ , [2,15,16,17]. Gli elementi diagonali di  $\mathbf{P}_T$  rappresentano le probabilità che lo stato iniziale non cambi. Ovviamente vale  $0 \leq P_{ij} \leq 1$ . Sino a quando  $T > 0$ , la rete non si trova in alcun stato stabile e  $P_{ij} \geq 0$ . Inoltre sulla matrice  $\mathbf{P}_T$  è valido il seguente vincolo:

$$\sum_{j=1}^N P_{ij} = 1 \quad (3.13)$$

Una matrice i cui elementi  $P_{ij} \geq 0$  e le cui righe diano come somma 1 è detta *matrice stocastica*.

Si definiscono le macchine di Boltzmann come *processi di Markov del primo ordine*, poiché le probabilità di transizione suddette dipendono solamente dallo stato corrente e non dai precedenti.  $\mathbf{P}_T$  è anche detta *matrice di Markov*. Lo sviluppo temporale della distribuzione di probabilità  $\nu_t = (p_{1t}, p_{2t}, \dots, p_{2^N t})$  degli stati della rete ad ogni transizione può essere calcolato in generale applicando alla distribuzione di probabilità dell'aggiornamento precedente,  $\nu_{t-1}$ , la matrice  $\mathbf{P}_T$ , in modo che:

$$\nu_t = \nu_{t-1} \mathbf{P}_T \quad (3.14)$$

Inoltre un sistema dinamico, quale abbiamo visto essere una rete di Hopfield e dunque anche una macchina di Boltzmann, che obbedisca ad una relazione quale la (3.14), viene detto *catena di Markov*. La distribuzione finale di probabilità è il *vettore autoconsistente*  $\nu$  tale che  $\nu = \nu \mathbf{P}_T$ . In altre parole, si può dire che il vettore  $\nu$  della distribuzione finale di probabilità è l'autovettore di  $\mathbf{P}_T$  con autovalore 1. Dalla teoria delle catene di Markov è noto che tale *distribuzione stabile*  $\nu$  esiste sempre se tutti gli stati possono essere raggiunti da qualsiasi altro stato in una o più transizioni con probabilità non nulle, [2,15,16].

Poiché qualsiasi rete di Boltzmann con  $T > 0$  soddisfa queste condizioni è assicurata l'esistenza di tale distribuzione, che rappresenta proprio la misura di Gibbs introdotta nel paragrafo precedente, valida nel caso in cui la rete si trovi in *equilibrio termico*. Tale *distribuzione di equilibrio* è indipendente dal cammino percorso nel raggiungimento del minimo. In analogia a quanto calcolato ad inizio paragrafo, per un sistema ad  $M$  stati con energie associate  $E_1, \dots, E_M$  la matrice di transizione  $\mathbf{P}_T$  sarà:

$$\mathbf{P}_T = \begin{pmatrix} 1 - \sum_{i=2}^M \frac{1}{1+e^{(E_i-E_1)/T}} & \cdots & \frac{1}{1+e^{(E_M-E_1)/T}} \\ \frac{1}{1+e^{(E_1-E_2)/T}} & \ddots & \vdots \\ \vdots & \vdots & \vdots \\ \frac{1}{1+e^{(E_1-E_M)/T}} & \cdots & 1 - \sum_{i=1}^{M-1} \frac{1}{1+e^{(E_i-E_M)/T}} \end{pmatrix} \quad (3.15)$$

Si verifica, con un prodotto righe per colonne, che il vettore  $\nu = \frac{1}{Z} \left( e^{-\frac{E_1}{T}}, e^{-\frac{E_2}{T}}, \dots, e^{-\frac{E_M}{T}} \right)$ , rappresentante la misura di Gibbs, moltiplicato per  $\mathbf{P}_T$  dà nuovamente  $\nu$ . Ciò significa che la distribuzione di Boltzmann è *stabile* rispetto alla  $\mathbf{P}_T$  introdotta e rappresenta effettivamente la *distribuzione di equilibrio* ricercata nel paragrafo precedente, [2].

È bene porre l'attenzione sul fatto che l'intero discorso fatto sin qui è basato sull'ipotesi di poter esprimere l'energia libera anche per un sistema discreto quale una rete neurale. Infatti da  $F$  si è ricavata la distribuzione di Boltzmann. Tale ipotesi andrebbe corroborata alla luce di un più attento studio dell'ipotesi ergodica per le catene di Markov, [Geo88] o [16]. Più avanti svilupperemo ulteriormente i risultati qui accennati.

Il tempo richiesto per il raggiungimento dell'equilibrio è legato alla temperatura: per  $T$  basse il processo può essere lungo, mentre per  $T$  più alte ciò richiede tempi minori. Un buon compromesso è partire da una alta temperatura e poi ridurla gradualmente, proprio come nella tempra di un sistema fisico [Kir83]. Ad alte  $T$  la rete ignorerà piccole differenze di energia e raggiungerà rapidamente l'equilibrio, effettuando una ricerca della complicata struttura nello spazio degli stati e ivi troverà un buon minimo. Quando la temperatura verrà abbassata, la rete inizierà a rispondere a differenze di energia via via minori e troverà allora un nuovo minimo in un intorno del precedente. Si noti che per  $T \rightarrow 0$  le transizioni diventano quelle delle reti di Hopfield. Approfondiremo ora tale algoritmo di ricerca.

### 3.3 Convergenza asintotica del Simulated Annealing

Per una  $T$  fissata si può descrivere una macchina di Boltzmann come catena di Markov. Per una  $T$  variabile, si deve parlare in termini di algoritmo di tempra simulata. Esso può essere pensato come una *ricerca locale* nello spazio  $S$  delle soluzioni di un problema di ottimizzazione combinatoria, espresso dalla funzione costo  $f$ . Tali algoritmi si basano sul miglioramento di una soluzione passo dopo passo, esplorando quelle ad essa vicine e determinando quali fra queste ottimizzano  $f$ .

Per comprendere il termine “vicino”, si daranno ora alcune definizioni, [16].

Si definisce *istanza* di un problema di ottimizzazione combinatoria la coppia  $(S, f)$ , dove la *funzione costo* è detta anche *funzione obiettivo* ed è definita sullo spazio delle soluzioni  $S$  e a valori reali. Come si è già detto, il problema si riduce all'individuazione di una soluzione *globalmente ottima*  $i_{opt}$ , tale che  $f(i_{opt}) \geq f(i)$  o  $f(i_{opt}) \leq f(i)$ ,  $\forall i \in S$ .

Una *struttura locale* di una istanza  $(S, f)$  è una funzione  $\zeta : S \rightarrow 2^S$  che per ogni soluzione  $i$  definisce un insieme  $S_i \in S$  di soluzioni *vicine* ad  $i$  secondo criteri specifici. L'insieme  $S_i$  prende allora il nome di *soluzioni vicine ad  $i$* . Estremamente interessante è il seguente, [15,16]:

**Teorema (di Convergenza rispetto alla Temperatura):** *Siano date l'istanza  $(S, f)$  di un problema di ottimizzazione statistica ed una struttura locale opportuna  $\zeta$ . Si consideri la distribuzione di probabilità:*

$$P_T(X = i) \stackrel{def}{=} \nu_i(T) = \frac{e^{-\frac{f(i)}{T}}}{\sum_{j \in S} e^{-\frac{f(j)}{T}}} \quad (3.16)$$

con  $X$  variabile aleatoria che identifica la soluzione corrente ottenuta da un algoritmo di ricerca locale ed  $f$  funzione costo.

Si dimostra che:

$$\lim_{T \rightarrow 0} \nu_i(T) \stackrel{def}{=} \nu_i^* = \frac{1}{|S_{opt}|} \Lambda_{(S_{opt})}(i) \quad (3.17)$$

dove  $S_{opt}$  rappresenta l'insieme delle soluzioni globalmente ottime, di dimensione  $|S_{opt}|$ , mentre  $\Lambda$  è la funzione<sup>3</sup> caratteristica.

Tale teorema, [16], garantisce l'asintotica convergenza degli algoritmi di ricerca locale, come il *simulated annealing*, all'insieme delle *soluzioni globalmente ottime*.

Si può modellizzare anche l'algoritmo di tempra simulata tramite catene di Markov. Sia  $O$  l'insieme dei possibili esiti in un processo di estrazione. Formalmente una *catena di Markov* è una sequenza di estrazioni, dove la probabilità dell'esito di una di queste dipende solamente da quello della precedente. Se  $x(k)$  è una variabile casuale che rappresenta l'esito della  $k$ -esima estrazione, per ogni coppia di valori  $i, j \in O$ , si può definire la *probabilità di transizione* dall'esito  $i$  all'esito  $j$  nell'estrazione  $k$  come la probabilità condizionata:

$$P_{ij}(k) = P(x(k) = j | x(k-1) = i) \quad (3.18)$$

Le varie probabilità di transizione formano la matrice di transizione  $\mathbf{P}$ , in analogia al caso precedente. Una catena di Markov è detta *finita* se è definita su un insieme di esiti  $O$  finito. Inoltre, una catena di Markov è detta *disomogenea* se le probabilità di transizione ad essa associate dipendono dal numero d'estrazione  $k$ , altrimenti essa è detta *omogenea*.

Una catena di Markov si dice poi *irriducibile* se, per ogni coppia di soluzioni  $i, j \in S$ , esiste una probabilità positiva di raggiungere  $j$  da  $i$  in un numero finito di transizioni, in simboli  $\forall i, j \in S, \exists n \geq 1 : (P^n)_{ij} > 0$ .

Si definisce invece *aperiodica* una catena di Markov tale che, per ogni soluzione  $i \in S$ , il massimo comun divisore  $MCD(D_i) = 1$ , essendo  $D_i$  l'insieme di tutti gli interi  $n > 0$  tali che  $(P^n)_{ii} > 0$ . Il massimo comun divisore viene detto *periodo della soluzione  $i$* . Una catena di Markov è aperiodica anche se  $\exists j \in S : P_{jj} > 0$ .

Si dimostra, [15] e [16], che all'algoritmo di *annealing simulato* si può associare una *catena di Markov finita, omogenea, irriducibile ed aperiodica*, le cui probabilità di transizione siano:

$$\forall i, j \in S : P_{ij}(k) = P_{ij}(T_k) = \begin{cases} \left( \frac{1}{|S_i|} \Lambda_{(S_i)}(j) \right) e^{-\frac{a^+}{T_k}}, & i \neq j \\ 1 - \sum_{l \in S, l \neq i} P_{il}(T_k), & i = j \end{cases}$$

dove  $a^+ = f(j) - f(i)$ , se  $f(j) > f(i)$  o  $a^+ = 0$  altrimenti (questo nel caso in cui l'ottimizzazione sia una minimizzazione della funzione costo).

Ad una catena di Markov così definita si trova corrispondere una distribuzione stazionaria

---

<sup>3</sup>Abbiamo incontrato già il termine "funzione caratteristica", pur con soli esempi espliciti. Dati due insiemi  $A$  e  $A_0 \subset A$ , la funzione caratteristica  $\Lambda : A \rightarrow 0, 1$  dell'insieme  $A_0$  è definita come  $\Lambda_{(A_0)}(a) = 1$ , se  $a \in A_0$  e  $\Lambda_{(A_0)}(a) = 0$ , altrimenti.

coincidente proprio con la distribuzione di Boltzmann, [15], pari cioè al vettore  $\nu$ , le cui componenti siano

$$\nu_i = \lim_{k \rightarrow \infty} P(x(k) = i | x(0) = j), \forall j \quad (3.19)$$

e tali da soddisfare l'equazione del *bilancio dettagliato*  $\nu_i P_{ij} = \nu_j P_{ji}$  ( $\forall i, j \in S$ ) e l'equazione del *bilancio globale*  $\sum_j \nu_j P_{ji} = q_i$  ( $\forall i$ ). Se esiste tale distribuzione allora

$$\lim_{k \rightarrow \infty} P(x(k) = i) = \lim_{k \rightarrow \infty} \sum_j P(x(k) = i | x(0) = j) P(x(0) = j) = \nu_i \sum_j P(x(0) = j) = \nu_i \quad (3.20)$$

Dunque la distribuzione stazionaria non è altro che la distribuzione di probabilità degli esiti (o soluzioni) dopo un numero infinito di transizioni. Nel caso del simulated annealing, per l'appunto, si può verificare tramite la relazione (3.14) che

$$\nu_i(T) = \frac{e^{-\frac{f(i)}{T}}}{\sum_{j \in S} e^{-\frac{f(j)}{T}}} \quad \forall i \in S \quad (3.21)$$

Mettendo insieme i risultati sulla distribuzione stazionaria della catena di Markov per la tempra simulata e del teorema di convergenza rispetto alla temperatura si ha:

$$\lim_{T \rightarrow 0} \nu_i(T) = \frac{1}{|S_{opt}|} \Lambda_{(S_{opt})}(i) = \lim_{T \rightarrow 0} \lim_{k \rightarrow \infty} P_T(x(k) = i) \quad (3.22)$$

da cui, ricordando la definizione della funzione caratteristica e tenendo conto della seconda uguaglianza, si può scrivere

$$\lim_{T \rightarrow 0} \lim_{k \rightarrow \infty} P_T(x(k) \in S_{opt}) = 1 \quad (3.23)$$

Tale risultato, [15,16], è estremamente più specifico dei precedenti e riflette la proprietà principale dell'algorithm di annealing simulato: *l'asintotica convergenza verso la soluzione ottima*. Tale caratteristica evidenzia l'aspetto *metaeuristico*, [Glo86], dell'algorithm di tempra simulata. È metaeuristico il generico algorithm non deterministico che sfrutta *processi euristici legati da una strategia "intelligente" più generale*, volta alla risoluzione di un problema combinatorio. Tale strategia è ad un *livello superiore rispetto alle fondamenta euristiche* e deve inoltre soddisfare i criteri di *diversificazione* (esplorazione dello spazio delle soluzioni) ed *intensificazione* (sfruttamento della conoscenza pregressa per evitare intrappolamenti e garantire la convergenza ad una soluzione sub-ottimale, approssimata, o ottimale, esatta). L'interesse per la metaeuristica è legato alla crescente individuazione di problemi *intrattabili*, per cui non esistono algoritmi di risoluzione esatta in tempi polinomiali, [4,27,30].

Nel simulated annealing si può fare a meno dell'ipotesi di effettuare infinite estrazioni ( $k \rightarrow \infty$ ) e garantire comunque la convergenza all'insieme delle soluzioni ottime tramite una "lenta" diminuzione del parametro di controllo  $T$ , [14,15]. Si dice che il simulated annealing è *metaeuristico esatto*, poiché esso giunge alla soluzione ottimale esatta in un numero finito di passi [27].

### 3.4 L'Algoritmo di Apprendimento delle Macchine di Boltzmann

Le unità computazionali di una macchina di Boltzmann si dividono in due gruppi: le unità *visibili*, che rappresentano l'interfaccia tra la rete e il mondo esterno, e quelle *nascoste*, se presenti, in grado di modellare la distribuzione di probabilità dei dati in ingresso. Consideriamo ora l'eventualità di fornire alla rete dei dati con una determinata distribuzione e di voler stabilire una distribuzione di rete che meglio la approssimi. A tal proposito definiamo una funzione costo  $J$ , detta *entropia relativa*, che è una misura della divergenza<sup>4</sup> tra le due distribuzioni  $R_a$  (distribuzione obiettivo) e  $P_a$  (distribuzione della rete), dove l'indice  $a$  scorre sugli stati delle sole unità visibili, [1]

$$J = \sum_a R_a \ln \frac{R_a}{P_a} \quad (3.24)$$

La probabilità  $P_a$  è data dalla misura di Gibbs:

$$P_a = \frac{\sum_g e^{-\frac{E_{(a,g)}}{T}}}{\sum_{(a,g)} e^{-\frac{E_{(a,g)}}{T}}} \quad (3.25)$$

L'indice  $g$  scorre sugli stati delle sole unità nascoste, mentre  $E_{(a,g)}$  rappresenta l'energia associata alla configurazione in cui le unità visibili siano nello stato  $a$  e quelle nascoste nello stato  $g$ . L'espressione di tale energia è:

$$E_{ag} = -\frac{1}{2} \sum_{ij} w_{ij} S_i^{(a,g)} S_j^{(a,g)} \quad (3.26)$$

dove  $S_i^{(a,g)}$  è lo stato dell'unità  $i$ -esima quando la rete si trova nella configurazione  $(a, g)$  e analogamente per  $S_j^{(a,g)}$ . Si calcola che, ricordando la definizione di valore medio:

$$\frac{\partial P_a}{\partial w_{ij}} = \frac{\left( \sum_g e^{-\frac{E_{(a,g)}}{T}} S_i^{(a,g)} S_j^{(a,g)} \right)}{T \left( \sum_{ag} e^{-\frac{E_{ag}}{T}} \right)} - \frac{\left( \sum_g e^{-\frac{E_{(a,g)}}{T}} \sum_{lh} e^{-\frac{E_{(l,h)}}{T}} S_i^{(l,h)} S_j^{(l,h)} \right)}{T \left( \sum_{ag} e^{-\frac{E_{(a,g)}}{T}} \right)^2} = \frac{\sum_g P_{ag} S_i^{(a,g)} S_j^{(a,g)} - P_a \langle S_i S_j \rangle}{T} \quad (3.27)$$

Al fine di ricavare la regola di apprendimento dei pesi, si può ora determinare la quantità:

$$\frac{\partial J}{\partial w_{ij}} = -\sum_a R_a \frac{1}{P_a} \frac{\partial P_a}{\partial w_{ij}} = \frac{1}{T} \left( -\sum_{ag} R_a \frac{P_{(a,g)}}{P_a} S_i^{(a,g)} S_j^{(a,g)} + \langle S_i S_j \rangle \right) \quad (3.28)$$

ricordando che  $\sum_a R_a = 1$ . Poiché la regola di aggiornamento dei pesi è direttamente proporzionale al gradiente della funzione costo, attraverso la costante  $\eta$ , allora:

<sup>4</sup>In letteratura l'*entropia relativa* è nota anche con il nome di *divergenza di Kullback-Leibler*, [Kul51], ma non è una *metrica* (non soddisfa la condizione di simmetria e la proprietà triangolare) bensì una *pseudometrica*. Ciò giustifica l'uso della parola "divergenza" rispetto a "distanza". Sebbene ad un livello puramente intuitivo esse sembrano coincidere, tale precisazione è formalmente doverosa.

$$\Delta w_{ij} = \frac{\eta}{T} \left( \sum_{(a,g)} R_a \frac{P_{(a,g)}}{P_a} S_i^{(a,g)} S_j^{(a,g)} - \langle S_i S_j \rangle \right) \quad (3.29)$$

Tale equazione fornisce la regola fondamentale dell'apprendimento nelle macchine di Boltzmann. Il primo termine corrisponde ad un termine di Hebb, con le unità visibili fissate nello stato  $a$ , mentre il secondo realizza la libera evoluzione del sistema verso uno stato qualsiasi. Il processo converge quando le correlazioni  $\langle S_i S_j \rangle$  tra le unità, relative al sistema lasciato libero di evolvere, risultano uguali alle correlazioni relative al sistema bloccato. Un aspetto sorprendente di questa regola è che essa sfrutta solo informazioni di tipo locale: il cambio di un peso dipende solo dal comportamento delle due unità connesse, sebbene poi questo finisca per minimizzare la distanza globale delle distribuzioni. Se non vi fossero unità nascoste, si può verificare, [1], che lo spazio in cui opera  $J$  presenterebbe un unico minimo globale e in questa evenienza si potrebbe utilizzare anche l'algoritmo di retropropagazione per l'apprendimento, sebbene con risultati peggiori.

Al fine di ottenere tutte le informazioni necessarie si fanno le *medie temporali*<sup>5</sup> della quantità  $S_i S_j$  per ciascuno dei possibili stati  $a$  tali che  $R_a > 0$  e, per una volta, anche con le unità visibili non bloccate. In ogni media l'intervallo temporale è quello tra lo stato di partenza e lo stato di equilibrio. Si noti comunque che anche ricorrendo all'annealing simulato, l'apprendimento con una macchina di Boltzmann richiede una computazione intensiva, poiché in ordine:

- è necessario aggiornare ciascun peso ripetute volte per ottenere la convergenza;
- per ogni aggiornamento si deve calcolare  $\langle S_i S_j \rangle$  in uno stato non bloccato e con le unità visibili bloccate su ciascuno dei pattern forniti in ingresso;
- per calcolare ciascuna di queste medie si deve portare il sistema all'equilibrio, utilizzando un programma di annealing (*annealing schedule*) per far diminuire la temperatura nel tempo;
- in corrispondenza delle temperature di equilibrio si devono campionare le diverse unità ed aggiornarle in accordo con la probabilità di transizione definita dalla misura di Gibbs.

Utilizzare brevi intervalli per calcolare le medie può ridurre i tempi degli aggiornamenti ma anche diminuirne l'accuratezza e in definitiva richiedere un numero maggiore di aggiornamenti. Si può dimostrare, [15,17], che affinché la rete trovi efficacemente il minimo globale, una condizione sufficiente è che la temperatura venga diminuita secondo un andamento del tipo:

$$T(t_n) = \frac{T_0}{1 + \log(t_n)} \quad (3.30)$$

dove  $T_0$  è la temperatura iniziale e la variabile discreta di tempo  $t_n$  rappresenta l' $n$ -esimo ciclo di processamento. A seconda dell'architettura della rete e della struttura dei pattern di ingresso, si possono trovare *programmi di annealing* ben più rapidi. Nonostante la loro estrema lentezza, le macchine di Boltzmann rappresentano un esempio interessante di modello stocastico.

---

<sup>5</sup>Verificheremo la correttezza di quanto fatto quando accenneremo al *teorema ergodico* nel capitolo 4.

### 3.5 Dall'Energia Libera all'Entropia

Concludiamo questo capitolo evidenziando la profonda importanza dell'energia libera nella dinamica delle macchine di Boltzmann, [1]. Si consideri un sistema fisico con un insieme di stati possibili  $a$ , ognuno con energia  $E_a$ . Ad una data  $T$ , il suo stato può variare e così anche le quantità che dipendono da esso. Dopo un certo periodo di tempo, il sistema, isolato da stimoli esterni, raggiungerà l'*equilibrio termico*, dove ciascuno dei possibili stati  $a$  si verificherà con probabilità

$$P_a = \frac{1}{Z} e^{-\frac{E_a}{T}} \quad (3.31)$$

dove la *funzione di partizione*  $Z$  può essere piuttosto complicata da calcolare. Questo è il principio di base dell'algoritmo di Metropolis, [26].

Calcoliamo la  $Z$  di una rete di Hopfield *bipolare* (i cui potenziali  $S_i$  delle unità computazionali possono essere 0 o -1) con  $N$  neuroni e con funzione energia data dall'espressione:

$$E = -\frac{1}{2} \sum_{ij} w_{ij} S_i S_j - \sum_i I_i S_i \quad (3.32)$$

dove  $w_{ij}$  è il peso della connessione tra le unità  $i$  e  $j$  mentre  $I_i$  è il potenziale in input all'unità  $i$ .

$Z$  è una somma su tutti i possibili stati (pari alle  $2^N$  combinazioni di  $S_i$ ), ossia una *traccia*:

$$Z = \sum_{S_1=\pm 1} \dots \sum_{S_N=\pm 1} e^{\left(\frac{1}{2T} \sum_{ij} w_{ij} S_i S_j + \frac{1}{T} \sum_i I_i S_i\right)} = Tr_s e^{\left(\frac{1}{2T} \sum_{ij} w_{ij} S_i S_j + \frac{1}{T} \sum_i I_i S_i\right)} \quad (3.33)$$

L'attivazione media  $\langle S_i \rangle$  dell'unità  $i$  è data dalla definizione di valore medio:

$$\langle S_i \rangle = \sum_a (S_i)_a P_a = \frac{1}{Z} Tr_s \left( S_i e^{\left(\frac{1}{2T} \sum_{ij} w_{ij} S_i S_j + \frac{1}{T} \sum_i I_i S_i\right)} \right) \quad (3.34)$$

Si può riscrivere l'attivazione media, vedendo che

$$\frac{\partial Z}{\partial I_i} = \frac{1}{T} Tr_s \left( S_i e^{\left(\frac{1}{2T} \sum_{ij} w_{ij} S_i S_j + \frac{1}{T} \sum_i I_i S_i\right)} \right) \longrightarrow \langle S_i \rangle = \frac{T}{Z} \frac{\partial Z}{\partial I_i} = T \frac{\partial}{\partial I_i} \ln Z \quad (3.35)$$

Riscrivendo l'energia libera come  $F = -T \ln Z$ , le espressioni per l'attivazione media e la funzione di correlazione si semplificano:

$$\langle S_i \rangle = -\frac{\partial F}{\partial I_i} \quad \langle S_i S_j \rangle = -\frac{\partial F}{\partial w_{ij}} \quad (3.36)$$

Si può verificare che l'espressione  $F = -T \ln Z$  è coerente con la definizione data ad inizio capitolo,  $F = E - TS$ , infatti

$$E - F = \sum_a P_a E_a + 1 \cdot T \ln Z = T \sum_a P_a \left( \frac{E_a}{T} + \ln Z \right) = -T \sum_a P_a \ln \left( \frac{e^{-\frac{E_a}{T}}}{Z} \right) = TS \quad (3.37)$$

Da tali espressioni scaturisce il legame tra la regola di apprendimento ed  $F$ , poiché:

$$e^{-\frac{F}{T}} = e^{\ln Z} = Z = \sum_a e^{-\frac{E_a}{T}} \longrightarrow \frac{e^{-\frac{F}{T}}}{Z} = \sum_a \frac{e^{-\frac{E_a}{T}}}{Z} = \sum_a P_a = 1 \quad (3.38)$$

e da qui si può calcolare la distribuzione di probabilità della rete quando le unità visibili sono fissate sugli input e sostituirla nell'entropia relativa:

$$P_a^{rete} = \frac{e^{-\frac{F_a^{fissata}}{T}}}{e^{-\frac{F}{T}}} \longrightarrow J = \sum_a R_a \ln R_a - \sum_a R_a \ln P_a^{rete} = \sum_a R_a \ln R_a - \sum_a R_a \left( -\frac{F_a^{fissata} - F}{T} \right) \quad (3.39)$$

Durante l'apprendimento, la quantità che viene minimizzata è la differenza tra l'energia libera della rete quando le unità visibili sono fissate sul vettore di input e l'energia libera del sistema in evoluzione. Ecco dunque introdotta una *nuova dinamica* della rete *in termini* non di funzione energia ma *di energia libera*.

A proposito dell'entropia  $S$ , è interessante vedere che essa può avere differenti interpretazioni. L'entropia esprime l'ampiezza della distribuzione di probabilità  $P_a$ , poiché maggiore è il numero di stati  $a$  che hanno una probabilità simile, maggiore è  $S$ . Un'altra interpretazione dell'entropia deriva dalla *teoria dell'informazione*, [1] e [17], in cui il termine bit assume un significato più ampio di quello usuale di *binary digit*. Supponiamo che un evento  $e$  avvenga con una probabilità  $P(e)$ . Se si è osservato tale evento si dice che *si sono ricevuti*  $I(e) = \log_2 \frac{1}{P(e)}$  *bit di informazione*. Se  $P(e) = 0.5$  allora  $I(e) = 1$  *bit*. Si può pertanto ridefinire un *bit* come la *quantità di informazione* ricevuta quando uno di due esiti equiprobabili è specificato. In questa definizione, ovviamente, il bit può non essere un numero intero. Nel caso di una macchina di Boltzmann si può esprimere l'*informazione* media ricevuta necessaria per specificare uno degli stati della rete come la quantità

$$\langle I \rangle = \sum_a P_a I(a) = -\sum_a P_a \log_2 P_a \quad (3.40)$$

dove  $a$  scorre su tutti i possibili eventi cui è associata una probabilità  $P_a$ . È evidente il legame<sup>6</sup> tra l'entropia e l'informazione addizionale. Se gli stati sono equiprobabili, è massima la difficoltà di riconoscerne uno e allo stesso modo massima è la quantità di informazione addizionale da avere, così come massima è l'entropia.

La teoria dell'informazione nacque proprio dallo studio dell'entropia, ad opera del matematico e ingegnere Claude Shannon: "*Pensavo di chiamarla informazione, ma la parola era fin troppo usata [...] Quando discussi della cosa con John Von Neumann [...] mi disse che avrei dovuto chiamarla entropia, per due motivi «Innanzitutto, la tua funzione d'incertezza è già nota nella meccanica statistica con quel nome. In secondo luogo, e più significativamente, nessuno sa cosa sia con certezza l'entropia, così in una discussione sarai sempre in vantaggio»*".

Queste righe sottolineano i profondi risultati, quasi 'precursori', della Meccanica Statistica.

<sup>6</sup>Secondo la convenzione usata  $\ln(x) = \log(x)$ . Inoltre è noto che  $\log_2(x) = \frac{\log(x)}{\log(2)}$ .

## 4 Il Modello di Ising

Se in un cataclisma andasse distrutta tutta la conoscenza scientifica e soltanto una frase potesse essere trasmessa alle generazioni successive, quale affermazione conterrebbe la massima quantità di informazioni nel numero minimo di parole? Io credo che sarebbe l'ipotesi atomica [...] In questa singola frase c'è un'enorme quantità di informazione sul mondo che ci circonda, se soltanto ci si riflette sopra con un po' di immaginazione.

---

R. FEYNMAN. SEI PEZZI FACILI.

## 4.1 Modello di Ising: isomorfismo con il modello di Hopfield

Scopo di quest'ultimo capitolo è presentare una caratterizzazione formale di un modello fisico vicinissimo al formalismo delle ANN ed ampiamente utilizzato nella trattazione delle Transizioni di Fase: il modello di Ising.

Quest'ultimo fu originariamente introdotto per trattare analiticamente la dinamica di un ferromagnete, [21] e [24], cioè un materiale in cui i momenti di dipolo magnetico  $\vec{S}_i$  degli atomi (dovuti sia al momento intrinseco che al momento orbitalico degli elettroni e anche ad un eventuale momento intrinseco del nucleo) siano in larga parte allineati nella medesima direzione, dando luogo ad una magnetizzazione spontanea macroscopica  $\vec{M} = \sum_i \vec{S}_i$  non nulla.

In tali sostanze ferromagnetiche, quali ad esempio *Fe*, *Ni* o *Co*, l'allineamento si riscontra solo per temperature  $T$  minori di una certa temperatura critica  $T_c$  detta *temperatura di Curie*. Quando  $T = T_c$ , il ferromagnete subisce una *transizione di fase del secondo ordine*<sup>1</sup> dalla fase ferromagnetica a quella paramagnetica ed  $\vec{M}$  cambia. In particolare, per  $T > T_c$  si registra  $\vec{M} = \vec{0}$ , in assenza di campi esterni.

Il modello di Ising effettua comunque delle esemplificazioni, trattando i dipoli magnetici o *spin* come vere e proprie quantità scalari, indicate d'ora in poi con  $S_i$ , poste nei siti di un reticolo ordinato d-dimensionale. Di ogni spin se ne trascurano inizialmente le origini quantistiche e si impone che  $S_i$  possa assumere solo valori  $+1$  o  $-1$ , a seconda che questo sia allineato o anti-allineato lungo una direzione di riferimento. Qualora il ferromagnete fosse immerso in un campo magnetico  $\vec{B}$  esterno, tale direzione di riferimento sarebbe coincidente con quella del campo.

La dinamica dell'intero sistema vedrà una minimizzazione dell'energia libera  $F = E - TS$ , in analogia a quanto già detto. Nella ricerca della configurazione di equilibrio, le interazioni tra spin potranno modificare il loro allineamento (ogni spin contribuisce a quello che viene chiamato *campo magnetico interno*). La frequenza di tali modifiche dipenderà da  $T$ , che regola anche le due eventualità concorrenti di *minimizzazione dell'energia*  $E$  (che porterà gli spin ad allinearsi ordinatamente) e *massimizzazione dell'entropia*  $S$  (che contribuirà al disordine del sistema). La Hamiltoniana  $\mathcal{H}$  del modello di Ising contiene i contributi dell'interazione binaria tra spin (evitando di contare due volte la medesima coppia di spin interagenti) e dell'interazione tra spin e campo magnetico esterno, avente intensità localmente variabile, pertanto

$$\mathcal{H} = -\frac{1}{2} \sum_{i,j} J_{ij} S_i S_j - \sum_i B_i S_i \quad (4.1)$$

dove si è sfruttato il fatto che gli spin sono orientati lungo la direzione del campo  $\vec{B}$  e dove  $J_{ij}$  rappresentano le costanti di accoppiamento tra gli spin degli atomi nei siti  $i$  e  $j$ , [2]. Per materiali ferromagnetici, le costanti di accoppiamento sono tutte strettamente positive.

A meno della notazione specifica, si verifica immediatamente che la formula (4.1) *ha la medesima struttura* della funzione energia di una rete di Hopfield bipolare con input ed output

<sup>1</sup>Nelle transizioni di fase del secondo ordine si ha un cambio continuo delle proprietà del sistema, a differenza di quelle del primo ordine, in cui invece si registrano cambiamenti più bruschi.

dell'unità  $i$ -esima  $I_i$  e  $V_i$ , essendo  $T_{ij}$  il peso della connessione tra il neurone  $i$  e quello  $j$ :

$$E = -\frac{1}{2} \sum_{ij} T_{ij} V_i V_j - \sum_i I_i V_i \quad (4.2)$$

Si può dire che il modello di Ising e quello di una rete di Hopfield bipolare sono *isomorfi*, [2,6].

Ne possiamo dare una prova grafica ricorrendo alla Teoria dei Grafi. Siano  $V(G)$  l'insieme dei nodi di un grafo semplice ed  $E(G)$  l'insieme dei suoi archi. Si definisce *isomorfismo* tra un grafo semplice  $G$  ed un grafo semplice  $H$  la funzione bigettiva  $f : V(G) \rightarrow V(H)$  che preserva la *struttura relazionale* dei due grafi, ossia tale che l'arco  $(i, j) \in E(G)$  se e solo se  $(f(i), f(j)) \in E(H)$ . Pertanto si può dire che  $f$  costituisce una relazione bigettiva tra tutti i nodi e i rispettivi archi di  $G$  e tutti i nodi e i rispettivi archi di  $H$ . Se tale  $f$  esiste, si dice che  $G$  ed  $H$  sono *isomorfi*, [4,23].

Assumiamo per pura semplicità di esposizione che il campo magnetico  $\vec{B}$  sia uniforme e che gli input della rete di Hopfield siano i medesimi. L'architettura di una rete di Hopfield con  $N$  neuroni ed un unico input  $I$  può essere espressa tramite un grafo semplice in cui ogni unità computazionale sia collegata alle altre, essendo per ipotesi esterna  $T_{ij} = T_{ji}$ . Si fa l'ipotesi aggiuntiva, necessaria per garantire la convergenza della dinamica della rete verso un minimo, che  $T_{ii} = 0$ . A tali  $N-1$  connessioni per ogni nodo della rete se ne aggiunge un'altra, cioè la connessione tra l'unità  $i$  e l' $N + 1$ -esimo nodo, rappresentante il dato di input  $I$ . Sia  $T_{iI}$  il peso di questa connessione. Quest'ultima dicitura ha un puro valore di etichetta.

Le interazioni del modello di Ising ad  $N$  spin possono essere riassunte sotto forma di un grafo semplice in cui  $N$  nodi rappresentano ciascuno un atomo dotato di momento magnetico, collegato agli altri  $N-1$  nodi, dove il peso della connessione  $J_{ij}$  è rappresentato dalla costante di accoppiamento di interazione tra  $S_i$  e  $S_j$ . Nessuno spin interagisce con se stesso, per cui  $J_{ii} = 0$ . Inoltre ogni nodo di tale grafo è connesso ad un  $N + 1$ -esimo nodo, rappresentante il campo magnetico esterno  $B$ . Sia  $J_{iB}$  il peso di questa connessione. Quest'ultima dicitura ha un puro valore di etichetta.

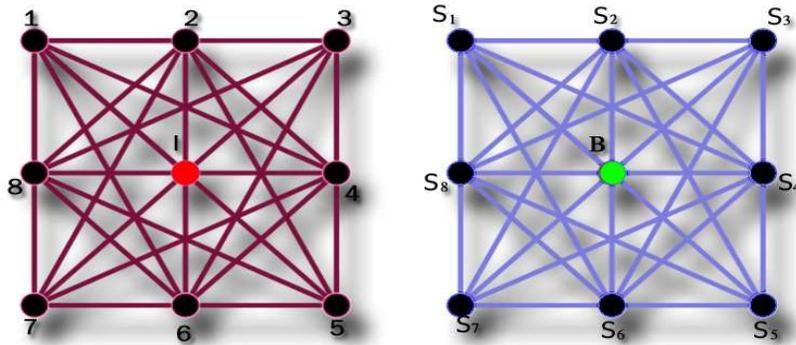


Figura 4.1: I grafi, che risultano essere completi, dei modelli di Hopfield (a sinistra) e di Ising (a sinistra).

In generale dimostrare che due grafi siano isomorfi è un problema di tipo *NP*. Immaginiamo ora di etichettare le unità computazionali come fatto nel capitolo 2. Si deve identificare una possibile relazione che li metta in corrispondenza bigettiva. Possiamo ad esempio riempire una tabella a due colonne come la 4.1.

Sotto ogni casella della tabella inseriamo poi l'elenco ordinato in modo crescente delle connessioni cui partecipa ogni spin o unità computazionale. Indichiamole con la dicitura del rispettivo peso. Quel che si ottiene è una corrispondenza bigettiva tra le righe della nostra tabella, infatti ogni nodo dei due grafi ha ben  $N$  connessioni, come si vede dalla 4.2.

Si noti che  $J_{Bi} = J_{iB}$  e  $T_{Ii} = T_{iI}$ .

Abbiamo dunque verificato graficamente che esiste una relazione bigettiva tra tutti i nodi e i rispettivi archi del grafo rappresentante il modello di Hopfield e tutti i nodi e i rispettivi archi del grafo rappresentante il modello di Ising.

Ciò comprova e dimostra che esiste un *isomorfismo* tra i due modelli e questo significa che essi, per via delle ipotesi fatte lungo il corso di questo e dei precedenti paragrafi, hanno una struttura fondamentalmente<sup>2</sup> *identica*. Il risultato trovato è di assoluta importanza, poiché ciò significa che è possibile controllare il modello di Ising attraverso il formalismo di una rete di Hopfield e, in chiave definitiva, trattare attraverso una rete neurale fenomeni quale l'evoluzione dinamica di un pezzo di ferromagnete alla temperatura  $T = 0$ . Con la generalizzazione stocastica delle reti di Hopfield, rappresentata dalle macchine di Boltzmann, è possibile estendere, grazie a preziosi risultati della Meccanica Statistica, il discorso fatto a temperature qualsiasi. Non solo. Oltre a poter applicare risultati matematici propri delle ANN a fenomeni di natura squisitamente fisica, è anche vero il contrario: si possono infatti applicare molti risultati generali della fisica e della meccanica statistica alla dinamica di una rete di Hopfield. L'esempio più lampante è la *funzione energia*, introdotta nel secondo capitolo. Infatti la funzione energia di una rete di Hopfield è strutturalmente identica ad una opportuna Hamiltoniana, il cui sinonimo inglese è proprio *energy function*. Da qui la nomenclatura utilizzata sinora.

Sulla scorta di questo risultato cercheremo, in seguito, di modellizzare e simulare una porzione di ferromagnete attraverso il metodo metaeuristico esatto del simulated annealing, pre-

Spin	Unità Comp.
$S_1$	1
$S_2$	2
...	...
$S_N$	N
$B$	$I$

Tabella 4.1: Corrispondenza tra gli spin e le unità computazionali per i due grafi.

Spin	Unità Comp.
$S_1$	1
$J_{1B}$	$T_{1I}$
$J_{12}$	$T_{12}$
...	...
$J_{1N}$	$T_{1N}$
$S_2$	2
$J_{2B}$	$T_{2I}$
$J_{21}$	$T_{21}$
...	...
$S_N$	N
$J_{NB}$	$T_{NI}$
$J_{N1}$	$T_{N1}$
...	...
$J_{NN-1}$	$T_{NN-1}$
$B$	$I$
$J_{B1}$	$T_{I1}$
...	...
$J_{BN}$	$T_{IN}$

Tabella 4.2: Corrispondenza tra le connessioni per gli  $N+1$  nodi di ciascuno dei due grafi.

<sup>2</sup>I due modelli sono strutturalmente identici a meno di fenomeni di cosiddetta *grana-fine* che potrebbero differire per via delle definizioni originariamente date o della natura delle approssimazioni fatte. Una dimostrazione più formale dell'isomorfismo presente tra i due modelli richiederebbe conoscenze di algebra che qui non si possono fornire, per ragioni di spazio.

via quindi macchine di Boltzmann. Studieremo inoltre anche la complessità dell'algoritmo impiegato e verificheremo la bontà dei risultati ottenuti.

Tuttavia, presenteremo prima un *metodo alternativo di risoluzione esatta* del modello di Ising bidimensionale, insieme ad una serie di concetti essenziali sulle transizioni di fase.

Sebbene a prima vista il modello di Ising appaia una approssimazione eccessivamente semplicistica della realtà, esso ha il vantaggio di fornire una certa semplicità di calcolo in una e due dimensioni e dà dunque la possibilità di rendersi effettivamente conto delle modalità di avvenimento di una transizione di fase. Il primo ad effettuare calcoli e a risolverlo in una dimensione fu proprio Ernst Ising, nel suo lavoro di tesi sotto la guida di Wilhelm Lenz, vero ideatore del modello, del 1925 [21]. Ising giunse ad una risoluzione esatta del modello nella dimensione  $d = 1$ , mostrando l'assenza di una transizione di fase per  $T > 0$  e generalizzando questo risultato anche per qualsiasi  $d > 1$ . Nel modello originariamente proposto da Lenz, tuttavia, i singoli spin potevano interagire con un numero limitato di altri spin, ossia i quattro più vicini. L'assunzione generale di Ising fu smentita dai risultati di anni successivi. Inoltre anche il modello unidimensionale, pur tuttavia con determinati limiti sulle interazioni, può presentare transizioni di fase, come vedremo più avanti.

Inteso come un trampolino di lancio per lo studio di eleganti proprietà matematiche ed universali leggi della fisica, il modello di Ising ha appassionato la comunità scientifica, contribuendo alla pubblicazione di oltre 12.000 articoli tra il 1969 e il 2002, [24].

## 4.2 Ergodicità e Simmetria

La Hamiltoniana del modello di Ising in assenza di campo magnetico esterno, analoga alla funzione energia di una rete di Hopfield con input nulli, è la seguente:

$$\mathcal{H} = -\frac{1}{2} \sum_{i,j} J_{ij} S_i S_j \quad (4.3)$$

Si nota che  $\mathcal{H}$  non contiene termini direzionali, poiché l'interazione tra spin tiene conto solo del posizionamento reciproco di coppie di momenti di dipolo. Dunque una qualsiasi rotazione dell'intero sistema la lascerebbe inalterata. Si dice pertanto che tale Hamiltoniana è *invariante sotto rotazioni arbitrarie*, [24]. Si è già accennato che per  $T > T_c$  la magnetizzazione totale del sistema  $\vec{M} = \sum_i \vec{S}_i$  è nulla, ciò significa che il materiale ferromagnetico, in tale regione di *alte temperature*, non presenta alcuna direzione privilegiata ed è anche esso, come la rispettiva Hamiltoniana, simmetrico per rotazioni lungo tutti e tre gli assi di un ipotetico sistema di riferimento. Viceversa invece, nella regione di *basse temperature*  $T < T_c$ , il sistema acquista una magnetizzazione non nulla e dunque una direzione privilegiata, identificata dal vettore  $\vec{S}_m = \langle \vec{S}_i \rangle$  lungo cui puntano in media tutti gli spin. Pertanto il sistema in tale regione è invariante sotto rotazioni rispetto ad un solo asse, identificato dalla direzione di  $\vec{S}_m$ . Si dice allora che il sistema ha simmetria minore della sua Hamiltoniana  $\mathcal{H}$ . Una quantità come  $\langle \vec{S}_i \rangle$ , nulla in una regione termica e non nulla in un'altra, viene detta *parametro d'ordine* e può risultare utile nello studio delle transizioni di fase. Invece il fenomeno di modifica della simmetria del sistema dalle alte alle basse temperature prende il nome di *rottura spontanea della simmetria*, [24] e [19].

In effetti nella transizione dalla fase paramagnetica a ferromagnetica si osserva un secondo fenomeno di rottura, relativo all'*ergodicità*.

Sfruttando ulteriormente l'isomorfismo del paragrafo precedente, si può associare anche al modello finito di Ising una matrice di Markov  $\mathbf{P}$  e portare avanti un discorso analogo al simulated annealing, (ora una vera e propria dinamica *aggiuntiva* del modello) che ci permette di individuare la convergenza del sistema ad una determinata ed unica distribuzione di probabilità  $\nu$ , per una configurazione iniziale qualsiasi e a patto di lasciar evolvere il sistema per tempi sufficientemente lunghi. Nel seguito intenderemo per *ergodicità* proprio tale proprietà di convergenza asintotica ad un'unica distribuzione<sup>3</sup> stazionaria. Tale proprietà è stata già utilizzata implicitamente, infatti un teorema fondamentale, attribuito a Gibbs e detto *teorema ergodico*, ci ha permesso di calcolare le medie statistiche  $\langle S_i S_j \rangle$  come medie temporali, nell'apprendimento di una macchina di Boltzmann, vista l'unicità della distribuzione obiettivo, [19] e [20].

Nel limite termodinamico del modello di Ising, cioè per il numero di spin  $N \rightarrow \infty$ , lo spazio degli stati passa da finito ad infinito e la proprietà di irriducibilità della catena di Markov associata alla dinamica può venire meno (non è più scontato che il sistema possa passare dallo stato  $i$  a quello  $j$  in un numero finito di passi). Con essa viene meno anche la garanzia dell'esistenza di un'unica distribuzione stazionaria. Lo stesso spazio delle fasi del sistema sarebbe così costituito da più bacini di attrazione. Questo fenomeno di *non-ergodicità* si riscontra per il modello di Ising 1d completamente connesso con opportune costanti  $J_{ij}$ , [Cas09], mentre il modello di Ising 1d in cui si tengano in conto solo le interazioni tra spin adiacenti (*interazioni nearest-neighbor* o *tra primi vicini*) si dimostra essere ergodico, [19].

Evidentemente le connessioni  $J_{ij}$  costituiscono dei vincoli essenziali sulla dinamica del sistema, ma è necessario considerarne l'intensità. Ad esempio un modello di Ising infinito, bidimensionale, completamente connesso e con opportune  $J_{ij}$  è ergodico solamente per  $T > T_c$  e quindi converge all'unica distribuzione fondamentale caratterizzata da  $M = 0$ . Per  $T < T_c$ , l'assenza di simmetria rende differente lo stato in cui si registra una magnetizzazione complessiva  $+M$  da quello in cui invece si ha  $-M$ , pur avendo entrambi la medesima energia libera minima e quindi la stessa distribuzione di Gibbs. Uno stato iniziale (cioè immediatamente antecedente alla transizione di fase) con magnetizzazione totale positiva tenderà dinamicamente al minimo di  $F$  relativo al valore  $+M$  dell'osservabile magnetizzazione. Viceversa si avrà nel caso contrario. Se poi inizialmente la magnetizzazione complessiva è nulla, il rumore termico contribuirà a far evolvere il sistema verso uno dei due bacini<sup>4</sup> di attrazione. Da ciò si registra una dipendenza dalla condizione iniziale e dunque una *non-ergodicità*. Poiché inoltre lo stesso sistema presenta o meno la proprietà di ergodicità a temperature differenti, si parla di *rottura dell'ergodicità*. La medesima rottura si ripresenta nel modello 2d con interazioni tra *primi vicini*, [19] e [28].

La presenza di ergodicità è dunque da ricercare non solo nella topologia del grafo associato al modello ma anche nella intensità delle connessioni. Per una giustificazione teorica, basata sulla Teoria dei Grafi e sulla Percolazione, che qui non è possibile proporre per problemi di spazio, si rimanda a [Ber93].

### 4.3 Sviluppo in Serie ad Alta Temperatura

Prima di procedere nella trattazione analitica esatta, faremo alcune approssimazioni. Tratteremo infatti il modello di Ising bidimensionale con interazioni a corto raggio ed omogenee,

<sup>3</sup>Si noti che la mancanza di ergodicità va ad inficiare la definizione di probabilità frequentista.

<sup>4</sup>Una analogia *degenerazione* rispetto all'energia si era osservata anche nel problema del Commesso Viaggiatore, dove il rumore aveva l'esatto compito di garantire la convergenza ad un minimo.

[22] e [24]. Il primo a dimostrare rigorosamente la presenza delle due regioni di alta e di bassa temperatura in un modello siffatto fu Peierls, in un articolo del 1936. Il suo *argomento* imponeva come condizioni al contorno un bordo del reticolo 2D con gli spin tutti +1 e cercava di calcolare la probabilità che lo spin del sito interno  $i$  fosse  $S_i = -1$ . In assenza di campo magnetico tale probabilità sarebbe  $\frac{1}{2}$ , tuttavia per le interazioni con gli spin al bordo è possibile che essa sia inferiore. Peierls in particolare dimostrò che per temperature  $T < T_c$  ma non nulle, la probabilità in questione è minore di 0.5 ed il campione presenta pertanto magnetizzazione spontanea, [24].

Studiamo ora analiticamente il modello di Ising.

Consideriamo un reticolo quadrato  $\Lambda$  con  $M$  segmenti orizzontali ed  $M$  segmenti verticali che colleghino i vari siti. Consideriamo due tipi di costanti di accoppiamento: sia  $J$  quella per interazioni lungo la direzione orizzontale e  $J'$  quella per interazioni in verticale. Poiché si considerano interazioni di un sito con i soli vicini in direzione nord-sud-est-ovest si parla di *intorno di Von Neumann*. In assenza di campo magnetico, ricordando l'isomorfismo e la formula (3.33) a pag. 39, la funzione di ripartizione sarà

$$Z = \sum_{\{S\}} \exp \left[ K \sum_{(i,j)} S_i S_j + L \sum_{(i,k)} S_i S_k \right] \quad (4.4)$$

dove la prima somma tra parentesi quadre è sugli spin vicini lungo i segmenti orizzontali e la seconda su quelli vicini lungo i segmenti verticali, dove  $K = \frac{J}{T}$  e  $L = \frac{J'}{T}$ . Utilizzando l'identità<sup>5</sup>  $\exp(x S_i S_l) = \cosh x (1 + S_i S_l \tanh x)$ , possiamo riscrivere la funzione di ripartizione, sostituendo alle due sommatorie sulle  $M$  coppie due produttorie di  $M$  termini:

$$Z = (\cosh K \cosh L)^M \sum_{\{S\}} \prod_{(i,j)} (1 + S_i S_j \tanh K) \prod_{(i,k)} (1 + S_i S_k \tanh L) \quad (4.5)$$

I parametri  $v = \tanh K$  e  $w = \tanh L$  sono dipendenti dalla temperatura e sono piccoli per alte  $T$ , mentre sono entrambi pari ad 1 per  $T = 0$ . Possiamo sviluppare  $Z$  rispetto a questi ultimi, in un intorno del punto  $T = \infty$ , ottenendo  $2^{2M}$  termini, visto che vi è un fattore per ogni segmento orizzontale e verticale (in tutto sono  $2M$ ) e ognuno di essi è degenerato due volte (a seconda che si vada da  $i$  a  $j$  o da  $j$  a  $i$ ). Ognuno di questi termini può essere rappresentato graficamente: se in esso compare  $v S_i S_j$ , si traccia l'arco orizzontale  $(i, j)$ , se invece compare  $w S_i S_k$ , allora si traccia l'arco verticale  $(i, k)$ , se infine compare il fattore 1 non si traccia alcun segmento. Ogni termine sarà nella forma

$$v^r w^p S_1^{n_1} S_2^{n_2} S_3^{n_3} \dots \quad (4.6)$$

dove  $r$  è il numero totale di archi orizzontali tracciati,  $p$  quello di archi verticali ed  $n_i$  è il grado in ingresso del sito  $i$ . Effettuando la somma su tutti gli stati del reticolo, poiché ogni spin può assumere i valori  $\pm 1$ , per evitare di avere un contributo complessivo nullo, si può imporre che tutti gli esponenti  $n_1, n_2, \dots, n_N$  siano pari, dunque  $S_i^{n_i} = 1$  per ogni  $i$  e la funzione di ripartizione sarà:

$$Z = (\cosh K \cosh L)^M \sum_{\{S\}} 2^N v^r w^p = 2^N (\cosh K \cosh L)^M \sum_P v^r w^p \quad (4.7)$$

<sup>5</sup>Tale identità si può verificare sviluppando ambo i membri in serie di potenze di  $x$  e ricordando che  $S_i^2 = 1$ .

Adesso la somma è estesa a tutte le configurazioni  $P$  poligonali chiuse del reticolo  $\Lambda$ , conseguenza del fatto che si sono scelti solo *gradi dei nodi*<sup>6</sup> pari. Sono ammissibili anche autointersezioni dei nodi.

Il primo termine della quantità geometrica  $\sum_P v^r w^p$  è pari ad 1, corrispondente, come abbiamo detto, al caso in cui nessun poligono sia stato tracciato sulla griglia  $\Lambda$ . Il secondo termine corrisponde al quadrato di lato unitario ( $r = 2, p = 2$ ), cioè al poligono chiuso più piccolo e non nullo. Vi è un termine di degenerazione, pari ad  $N$ , a seconda del sito da cui si inizi tracciare il poligono. Il secondo termine è  $N(vw)^2$ . Il poligono chiuso immediatamente successivo al quadrato è un rettangolo, composto da sei archi e orientato verticalmente ( $r = 2, p = 4$ ) o orizzontalmente ( $r = 4, p = 2$ ). Pertanto ve ne saranno  $N$  del tipo  $v^2w^4$  ed  $N$  del tipo  $v^4w^2$ . In conclusione, si avrà

$$\Phi(v, w) = \sum_P v^r w^p = 1 + N(vw)^2 + Nv^2w^4 + Nv^4w^2 + \dots \quad (4.8)$$

Una prima determinazione della temperatura critica fu effettuata da Kramers e Wannier effettuando delle considerazioni tra questo e lo sviluppo in serie di basse temperature, portato avanti sul *reticolo duale*<sup>7</sup>  $\Lambda_d$  di  $\Lambda$ . Si verifica che per  $T < T_c$  l'introduzione del reticolo duale permette di isolare i domini magnetici di un esiguo numero di spin antiallineati rispetto alla maggioranza ed una relazione di dualità permette di giungere alla temperatura critica. Tale dualità contiene informazioni importanti sulla simmetria degli sviluppi a basse e ad alte  $T$ , [Kra41].

## 4.4 La Formula di Feynman-Kac

Proseguiremo ora con l'illustrazione di una soluzione combinatoria esatta del modello di Ising 2d, proposta da M. Kac e J. C. Ward e alla quale lavorarono anche R. Feynman ed N. V. Vdovichenko.

Sul discorso più generale fatto sullo sviluppo ad alte temperature si fa immediatamente l'approssimazione che  $J = J'$ , di modo che la funzione di partizione sia dipendente dal solo parametro  $v = \tanh \frac{J}{T}$ . Inoltre si suppone che il numero  $N$  di spin sia macroscopicamente grande, in modo da poter fare l'approssimazione<sup>8</sup>  $M \simeq N$  ma in modo anche che la magnetizzazione del sistema e le costanti di accoppiamento risultino sempre finite. Ricordando l'identità  $\cosh^2 x - \sinh^2 x = 1$ , sulla scorta delle assunzioni fatte si può riscrivere la funzione di ripartizione:

$$Z = 2^N (1 - v^2)^{-N} \sum_P v^r = 2^N (1 - v^2)^{-N} \sum_r g_r v^r \quad (4.9)$$

dove il fattore  $g_r$  è il termine di degenerazione relativo al numero di grafi chiusi e costituiti da uno stesso numero pari  $r$  di segmenti. Non è detto che i grafi siano connessi, in tal caso la chiusura, derivante dalla parità dei gradi dei nodi, si applica ai rispettivi sottografi.

Si procederà ora alla *riorganizzazione della somma sui poligoni in termini delle sotto-*

<sup>6</sup>Intenderemo i poligoni tracciati sul reticolo come grafi chiusi semplici.

<sup>7</sup>Ogni reticolo planare  $\Lambda$  quadrato ammette un reticolo duale  $\Lambda_d$  anche esso quadrato, avente per nodi i centri geometrici delle celle della griglia  $\Lambda$  e per archi le connessioni orizzontali e verticali tra centri geometrici adiacenti.

<sup>8</sup>Si ricorda che  $M = (n^\circ \text{ di segmenti lungo un lato del reticolo quadrato}) \cdot (n^\circ \text{ di nodi lungo un lato del reticolo quadrato})$ .

*parti chiuse e connesse*, chiamate d'ora in poi *maglie*. Per grafi con autointersezioni, la decomposizione in maglie può risultare ambigua e dipendere dalla loro costruzione.

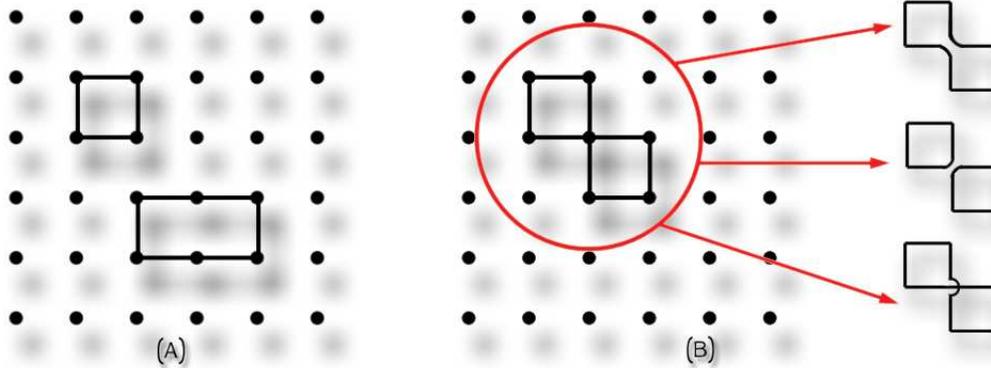


Figura 4.2: Esempio di grafo non connesso (A). Esempio di grafo connesso con autointersezione e tre diverse decomposizioni in maglie (B).

Si deve pertanto tenere conto di tale fatto, poiché nella decomposizione in maglie si potrebbe contare erroneamente più volte una medesima configurazione originale. L'idea è quella di pesare ciascun grafo con il fattore  $(-1)^n$ , dove  $n$  rappresenta il numero totale di autointersezioni delle maglie del grafo stesso. Poiché infatti si verifica che si hanno tre decomposizioni simili a quelle della figura di cui sopra per tutte le autointersezioni di grafi, si vede che le prime due decomposizioni avrebbero segno  $+$  mentre la terza  $-$ , pertanto nella somma due di queste si eliderebbero, permettendo eventualmente di tenere conto di un unico termine. Tuttavia nella somma delle decomposizioni in maglie potrebbero ora rientrare anche grafi con legami ripetuti contenenti però dei nodi dispari e quindi formalmente non inclusi nello sviluppo in serie ad alte temperature di partenza. Tuttavia la riproposizione della somma attraverso il peso dei termini li fa elidere vicendevolmente, come si vede facilmente dal numero di intersezioni delle decomposizioni.

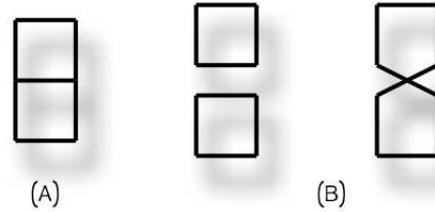


Figura 4.3: Grafo con legame ripetuto (A) e sue possibili decomposizioni (B).

Formalmente però la somma sulle diverse maglie ha un inconveniente, poiché il peso  $(-1)^n$  dipende da una proprietà globale quale il numero totale di intersezioni della decomposizione, mentre sarebbe opportuno ricorrere a proprietà locali. Si ricordi che l'angolo totale di rotazione della tangente durante l'aggiramento di una maglia piana è uguale a  $2\pi(l + 1)$  con  $l \in \mathbb{Z}$ , dove la parità di  $l$  coincide con quella del numero  $c$  di autointersezioni della maglia. Quindi, a ciascun nodo della maglia, in cui si può avere una rotazione pari ad  $\alpha = 0, \pm\frac{\pi}{2}$ , si fa corrispondere un fattore di fase  $e^{i\frac{\alpha}{2}}$ . Alla chiusura della maglia il prodotto di tali fattori<sup>9</sup> darà  $(-1)^{c+1}$ , per  $s$  maglie non connesse si avrà l'analogo risultato  $(-1)^{C+s}$ , con  $C = \sum c$ . Poiché appare evidente che  $C = n$ , allora si

<sup>9</sup>Ad esempio nel caso di un quadrato di lato unitario, ogni nodo avrà un fattore di fase  $e^{-i\frac{\pi}{4}}$  ed il loro prodotto sarà  $e^{-i\pi} = -1$ .

può passare da una proprietà globale ad una locale tenendo conto del fattore di fase  $e^{i\frac{\alpha}{2}}$  per ogni nodo e includendo nel peso anche il fattore  $(-1)^s$ , con  $s$  pari al numero di maglie non connesse, in modo da ottenere la quantità  $(-1)^{n+s}$ .

Pertanto possiamo riscrivere la funzione  $\Phi(v)$  non come somma di grafi ma come somma delle rispettive decomposizioni in  $s$  maglie isolate:

$$\Phi(v) = \sum_{s=0}^{\infty} (-1)^s \frac{1}{s!} \sum_{r_1, r_2, \dots = 1}^{\infty} v^{r_1+r_2+\dots+r_s} f_{r_1} \dots f_{r_s} \quad (4.10)$$

dove il termine  $\frac{1}{s!}$  tiene conto del fatto che permutazioni delle  $s$  etichette delle maglie danno luogo a decomposizioni del medesimo grafo; l'indice  $r_i$  denota il numero di segmenti della  $i$ -esima maglia solitaria e assume tutti i valori possibili<sup>10</sup> da 1 ad  $\infty$ ;  $(-1)^s$  è solo una parte del peso relativo alle maglie, essendo ogni nodo implicitamente corredato di un fattore di fase, incluso in  $f_{r_i}$ ; la somma  $r = r_1 + \dots + r_s$  rappresenta il numero complessivo variabile di segmenti della decomposizione e la quantità  $f_{r_i}$  tiene conto della degenerazione  $g_r$  e generalizza la somma a *tutte le possibili* maglie solitarie composte da  $r_i$  segmenti.

Poiché gli indici  $r_1, r_2, \dots, r_s$  assumono tutti i valori interi tra 1 ed infinito in modo indipendente, si può scrivere

$$\sum_{r_1, r_2, \dots = 1}^{\infty} v^{r_1+r_2+\dots+r_s} f_{r_1} \dots f_{r_s} = \left( \sum_{r=1}^{\infty} v^r f_r \right)^s \rightarrow \Phi(v) = \exp \left[ - \sum_{r=1}^{\infty} v^r f_r \right] \quad (4.11)$$

Siamo giunti ad un risultato importante, poiché abbiamo *ridotto la somma su poligoni* ad una *somma su cammini chiusi senza intersezioni* e quindi *generalizzato* la somma stessa a *tutti i possibili cammini Euleriani chiusi*.

Dobbiamo tuttavia valutare la quantità  $f_r$ . Numeriamo le quattro diverse direzioni di spostamento, verso est, nord, ovest e sud, rispettivamente con l'indice  $\mu = 1, 2, 3, 4$ .

Introduciamo anche la funzione  $W_r(i, j, \mu)$ , definendola come la somma su tutte le transizioni di lunghezza  $r$  che congiungono il nodo iniziale individuato da  $(i_0, j_0)$ , con una direzione di partenza  $\mu_0$ , ed il nodo  $(i, j)$ , in direzione di arrivo  $\mu$ . Anche i cammini relativi alle transizioni di  $W_r(i, j, \mu)$  sono pesati con i fattori di fase  $e^{i\frac{\alpha}{2}}$  per ciascun legame. Tale funzione soddisfa alcune equazioni ricorsive, una per ogni direzione del reticolo. Si può ad esempio arrivare al punto  $(i, j)$  in direzione 1 eseguendo l'ultimo  $r + 1$ -esimo passo da sinistra, dal basso o dall'alto, ma non da destra, poiché si ripercorrerebbe due volte in senso opposto il medesimo arco orizzontale a destra di  $(i, j)$ . È allora possibile scrivere

$$W_{r+1}(i, j, 1) = W_r(i-1, j, 1) + e^{-i\frac{\pi}{4}} W_r(i, j-1, 2) + 0 + e^{i\frac{\pi}{4}} W_r(i, j+1, 4) \quad (4.12)$$

dove chiaramente i fattori di fase tengono conto dei relativi cambiamenti di direzione.

Fissiamo l'attenzione su  $W_r(i_0, j_0, \mu)$ : essa è la somma su tutte le maglie uscenti rientranti nel nodo  $(i_0, j_0)$ . Si noti che in tali cammini chiusi non si può percorrere due volte lo stesso segmento in senso contrario. Pertanto si stabilisce la seguente uguaglianza tra la somma a

<sup>10</sup>Si noti che le somme non sono limitate superiormente, pur essendo il numero possibile di nodi pari ad  $N$ . Ciò è possibile poiché le maglie con più di  $N$  nodi danno contributo nullo alla somma, in quanto sicuramente conterranno legami ripetuti e dunque si elideranno vicendevolmente.

tutte le possibili maglie solitarie di lunghezza  $r$  e le transizioni della medesima lunghezza:

$$f_r = \frac{1}{2r} \sum_{i_0, j_0, \mu} W_r(i_0, j_0, \mu) \quad (4.13)$$

Il fattore  $\frac{1}{2r}$  tiene conto che ogni maglia chiusa, in esatta analogia a quanto detto nel considerare i tour del Commesso Viaggiatore, può essere percorsa in due direzioni opposte e può avere uno qualsiasi dei suoi  $r$  nodi come punto iniziale. Possiamo sviluppare ulteriormente questa uguaglianza, ricordando le equazioni ricorsive, come la (4.12), cui soddisfa la funzione  $W_r(i, j, \mu)$  e che possiamo sinteticamente riassumere come:

$$W_{r+1}(i, j, \mu) = \sum_{i', j', \mu'} \mathbf{P}(ij\mu|i'j'\mu') W_r(i', j', \mu') \quad (4.14)$$

Gli elementi della matrice  $\mathbf{P}$  tengono conto dei fattori di fase corrispondenti ai cambi di direzione di un punto “errante” passo passo nel reticolo e rappresentano le probabilità di transizione da un nodo all’altro. L’equazione di cui sopra rappresenta pertanto un *processo di Markov* relativo ad un *moto aleatorio* sul reticolo quadrato. Poiché le direzioni possibili sono quattro e visto che  $\mathbf{P}$  è relativa ai cambi di direzione ( $\leftarrow, \downarrow, \rightarrow, \uparrow$ ), allora essa è una matrice  $4 \times 4$ , rappresentabile graficamente grazie ai cambi di direzione:

$$\mathbf{P} = \begin{bmatrix} \rightarrow\rightarrow & \uparrow\rightarrow & \leftarrow\rightarrow & \downarrow\rightarrow \\ \rightarrow\uparrow & \uparrow\uparrow & \uparrow\leftarrow & \uparrow\downarrow \\ \leftarrow\rightarrow & \leftarrow\uparrow & \leftarrow\leftarrow & \leftarrow\downarrow \\ \rightarrow\downarrow & \uparrow\downarrow & \downarrow\leftarrow & \downarrow\downarrow \end{bmatrix} \quad (4.15)$$

La probabilità di transizione relativa ad un cammino di lunghezza totale  $r$  sarà data da  $\mathbf{P}^r$ , la cui somma degli elementi diagonali esprimerà la probabilità di tornare al punto iniziale dopo aver fatto un cammino chiuso di lunghezza  $r$ . Pertanto si avrà la seguente uguaglianza:

$$Tr \mathbf{P}^r = \sum_{i_0, j_0, \mu} W_r(i_0, j_0, \mu) \quad (4.16)$$

dove la funzione  $W_r$  è intesa ora in senso probabilistico. Da tale risultato possiamo scrivere che la quantità  $f_r$  è data da:

$$f_r = \frac{1}{2r} \sum_{i_0, j_0, \mu} W_r(i_0, j_0, \mu) = \frac{1}{2r} Tr \mathbf{P}^r = \frac{1}{2r} \sum_a \lambda_a^r \quad (4.17)$$

dove  $\lambda_a$  indica il generico autovalore della matrice  $\mathbf{P}$ . Ora che siamo riusciti a quantificare gli  $f_r$ , mettendo in evidenza che in essi sono presenti i termini di fase, possiamo riscrivere ancora la funzione  $\Phi(v)$ , ricordando che per alte temperature  $v \ll 1$  e dunque applicando risultati noti degli sviluppi in serie:

$$\Phi(v) = \exp \left[ - \sum_{r=1}^{\infty} v^r f_r \right] = \exp \left[ - \frac{1}{2} \sum_a \sum_{r=1}^{\infty} \frac{1}{r} v^r \lambda_a^r \right] = \exp \left[ \frac{1}{2} \sum_a \log(1 - v \lambda_a) \right] = \prod_a (1 - v \lambda_a)^{\frac{1}{2}} \quad (4.18)$$

La determinazione della funzione  $\Phi(v)$  si è ora trasformata in un problema di ricerca degli autovalori della matrice di Markov  $\mathbf{P}$ . È possibile diagonalizzare tale matrice rispetto alle coordinate del reticolo  $(i, j)$  mediante il passaggio ad un’altra rappresentazione tramite la

trasformazione di Fourier:

$$W_r(p, q, \mu) = \sum_{i,j=0}^L e^{-\frac{i2\pi}{L}(pi+qj)} W_r(i, j, \mu) \quad (4.19)$$

dove si ricorda che  $L$  è dato da  $N = L^2$ . Prendendo la trasformata di Fourier delle quattro equazioni ricorsive cui soddisfa  $W_r(i, j, \mu)$ , di cui una è la (4.12) e le altre tre sono costruite in modo analogo, si ricava, posti  $\varepsilon = e^{\frac{2\pi i}{L}}$  e  $\kappa = e^{\frac{i\pi}{4}}$ :

$$\begin{aligned} W_{r+1}(p, q, 1) &= \varepsilon^{-p} W_r(p, q, 1) + \varepsilon^{-q} \kappa^{-1} W_r(p, q, 2) + 0 + \varepsilon^q \kappa W_r(p, q, 4) \\ W_{r+1}(p, q, 2) &= \varepsilon^{-p} W_r(p, q, 1) + \varepsilon^{-q} W_r(p, q, 2) + \varepsilon^p \kappa^{-1} W_r(p, q, 3) + 0 \\ W_{r+1}(p, q, 3) &= 0 + \varepsilon^{-q} \kappa W_r(p, q, 2) + \varepsilon^p W_r(p, q, 3) + \varepsilon^q \kappa^{-1} W_r(p, q, 4) \\ W_{r+1}(p, q, 4) &= \varepsilon^{-p} \kappa^{-1} W_r(p, q, 1) + 0 + \varepsilon^p \kappa W_r(p, q, 3) + \varepsilon^q W_r(p, q, 4) \end{aligned}$$

Si nota immediatamente che, a differenza di quanto accade nelle equazioni ricorsive di partenza, come ad esempio la (4.12), in questo nuovo sistema di equazioni la funzione  $W_r(p, q, \mu)$  compare a destra e a sinistra di ogni uguaglianza con i medesimi indici spaziali  $p$  e  $q$ . Ciò è indice del fatto che la matrice  $\mathbf{P}$  è diagonale rispetto a tali indici e si ha:

$$\mathbf{P}(p, q, \mu | p, q, \mu') = \begin{pmatrix} \varepsilon^{-p} & \varepsilon^{-q} \kappa^{-1} & 0 & \varepsilon^q \kappa \\ \varepsilon^{-p} & \varepsilon^{-q} & \varepsilon^p \kappa^{-1} & 0 \\ 0 & \varepsilon^{-q} \kappa & \varepsilon^p & \varepsilon^q \kappa^{-1} \\ \varepsilon^{-p} \kappa^{-1} & 0 & \varepsilon^p \kappa & \varepsilon^q \end{pmatrix}$$

In questa rappresentazione il calcolo degli autovalori è più agevole e permette di riscrivere la funzione di  $\Phi(v)$  e dunque l'intera funzione di ripartizione del sistema come:

$$Z = 2^N (1 - v^2)^{-N} \Phi(v) = 2^N (1 - v^2)^{-N} \prod_{p,q=0}^L \left[ (1 + v^2) - 2v(1 - v^2) \left( \cos \frac{2\pi p}{L} + \cos \frac{2\pi q}{L} \right) \right]^{\frac{1}{2}} \quad (4.20)$$

Siamo giunti ad un risultato fondamentale, che ci permette ora di ricavare grandezze termodinamiche come l'energia libera per il modello di Ising presupposto. Infatti al limite per  $L \rightarrow \infty$  si può sostituire la somma con un integrale e scrivere  $F$  come:

$$F = -T \ln Z = -NT \ln 2 + NT \ln (1 - v^2) + \quad (4.21)$$

$$- \frac{NT}{2(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} \ln \left[ (1 + v^2)^2 - 2v(1 - v^2) (\cos w_1 + \cos w_2) \right] dw_1 dw_2$$

dove  $w_1 = \frac{2\pi p}{L}$  e  $w_2 = \frac{2\pi q}{L}$ . Si nota che l'energia libera è proporzionale al numero di siti  $N$  (cioè  $F$  scala con  $N$ ) ed inoltre essa ha un punto singolare per un valore finito e non nullo di  $T$ , quando l'argomento del logaritmo si annulla. Se si prende tale logaritmo come funzione di  $w_1$  e  $w_2$ , si verifica che esso è minimo quando  $\cos w_1 = \cos w_2 = 1$ , assumendo il valore  $(1 + v^2)^2 - 4v(1 - v^2) = (v^2 + 2v - 1)^2$ . Tale argomento minimo si annulla per il

valore positivo  $v = v_c = \sqrt{2} - 1$ . Ricordando che  $v = \tanh \frac{J}{T}$ , si ricava che:

$$\tanh \frac{J}{T_c} = v_c \longrightarrow T_c \simeq 2.26922 J \quad (4.22)$$

Tale temperatura è indicata con  $T_c$  poiché essa *individua la transizione di fase del reticolo quadrato*. In particolare, sviluppando l'energia libera  $F(T)$  in serie di potenze in  $t = \frac{T-T_c}{J}$  nell'intorno di  $T = T_c$ , si possono ricavare molte informazioni utili. Si vede ad esempio che vicino alla transizione di fase, definite le costanti A e B, con  $B > 0$ , l'andamento di  $F$  è

$$F(T) \simeq A - \frac{B}{2} (T - T_c)^2 \ln |T - T_c| \quad (4.23)$$

Pertanto l'energia libera è continua nel punto di transizione, mentre il calore specifico ivi ha una singolarità logaritmica

$$C \simeq -BT_c \ln |T - T_c| \quad (4.24)$$

Si noti che l'andamento ad infinito è simmetrico da ambo i lati di tale punto.

I risultati precedenti, formalizzati da N.V. Vdovichenko, sono in pieno accordo con i calcoli, in generale più complicati, portati avanti da L. Onsager, il primo a risolvere esattamente il modello di Ising bidimensionale, nel 1947. Negli anni strade di calcolo più semplici sono state aperte dal Gruppo di Rinormalizzazione e dalla Teoria dei Campi.

Sulla scorta di risultati analoghi a quelli presentati per la funzione di ripartizione Z, è possibile anche calcolare la funzione di correlazione  $\langle S_i S_j \rangle$ , che esprime la probabilità condizionata che i due spin  $S_i$  ed  $S_j$  abbiano lo stesso valore. Tale interazione tra due spin  $S_i$  ed  $S_j$  *primi vicini* è regolata non solo dalla loro eventuale interazione diretta ma anche da tutte le interazioni indirette con gli spin situati nelle vicinanze. Per  $T \neq T_c$

$$\langle S_i S_j \rangle \simeq M^2 + A \exp \left[ -\frac{|i-j|}{\xi} \right] \quad T \neq T_c \quad (4.25)$$

dove  $M$  è la magnetizzazione spontanea del reticolo,  $A$  è una costante positiva, la grandezza  $|i-j|$  indica la distanza tra il nodo dello spin  $S_i$  e quello dello spin  $S_j$ , mentre  $\xi$  è una quantità detta *lunghezza di correlazione* ed indica la dimensione delle regioni in cui gli spin assumono lo stesso valore. Si noti che per  $T \neq T_c$  le interazioni decadono all'infinito in modo esponenziale. Tuttavia, in vicinanza della temperatura critica, la lunghezza di correlazione diverge secondo una legge di potenza

$$\xi \simeq |T - T_c|^{-1} \quad T \simeq T_c \quad (4.26)$$

Poiché allora  $\xi \rightarrow \infty$  per  $T \rightarrow T_c$ , si può concludere che nella transizione di fase si verifica un comportamento *collettivo* degli spin. Proprio per  $T = T_c$ , la funzione di correlazione decade non più esponenzialmente rispetto alla distanza ma come:

$$\langle S_i S_j \rangle \simeq \frac{1}{|i-j|^{\frac{1}{4}}} \quad T = T_c \quad (4.27)$$

Dunque alla temperatura critica le *interazioni medie tra spin diventano a lungo raggio* e ciò giustifica la collettività del fenomeno di transizione di fase. Si chiude questo paragrafo riportando i risultati analitici, ottenuti da calcoli complessi, [24], per la magnetizzazione

spontanea:

$$M^2 = \begin{cases} \left[1 - \left(\frac{1-v^2}{2v}\right)^4\right]^{\frac{1}{4}} & T < T_c \\ 0 & T > T_c \end{cases} \quad (4.28)$$

## 4.5 Annealing Simulato per il Modello di Ising: Complessità

Consideriamo un modello di Ising con  $N$  spin, definito sul grafo  $G = (V, E)$ , con Hamiltoniana  $\mathcal{H}' = -\frac{1}{2} \sum_{i,j} J_{ij} S_i S_j$  ed avente uno spazio degli stati  $\Omega = \{-1, +1\}^N$ . L'energia di ogni stato  $\omega = (S_1, S_2, \dots, S_N)$  sarà dato da  $\mathcal{H}'(\omega)$ , che può essere definita in termini di *tagli* di  $G$ , [30]. Denotiamo ora con  $C^+ = \{v_i | S_i = +1\}$  l'insieme dei vertici i cui corrispondenti spin siano orientati verso l'alto. Analogamente consideriamo anche  $C^- = \{v_i | S_i = -1\}$ . Si definisce *taglio di  $G$*  l'insieme  $C = \{C^+, C^-\}$ . Consideriamo inoltre l'insieme di *archi* con entrambi gli estremi appartenenti a  $C^+$  ed indichiamolo con  $E^+$ . Analogamente consideriamo anche  $E^-$ . La parola *taglio* si riferisce a quell'insieme di archi di  $G$  che non appartengono né ad  $E^+$  e neppure ad  $E^-$ , poiché hanno un estremo in  $C^+$  ed uno in  $C^-$ . Sia  $E^{+-}$  l'insieme di archi che costituiscono tale taglio e sia  $W(C)$  il suo peso, corrispondente pertanto alla somma dei singoli  $J_{ij}$  degli archi costituenti  $E^{+-}$ .

Al variare dello stato  $\omega$  varia anche il corrispondente taglio  $C$ . Poiché, inoltre, per ogni configurazione esiste uno ed un solo taglio, si può riscrivere l'Hamiltoniana come:

$$\mathcal{H}'(C) = - \sum_{(i,j) \in E^+} J_{ij} - \sum_{(i,j) \in E^-} J_{ij} + \sum_{(i,j) \in E^{+-}} J_{ij} = - \sum_{(i,j) \in E} J_{ij} + 2 \sum_{(i,j) \in E^{+-}} J_{ij} \quad (4.29)$$

Si è ottenuto pertanto che l'Hamiltoniana del modello è ora la somma di un termine costante e derivante dal grafo  $G$  in questione più un termine dipendente dal doppio del peso del taglio  $C$ . La minimizzazione di  $\mathcal{H}'(C)$  corrisponde pertanto al problema di trovare il *taglio con il peso minore*, che a sua volta si dimostra essere un problema appartenente a P per *grafi planari* mentre NP-Completo per grafi arbitrari, [12] e [30]. La complessità della computazione dello stato fondamentale si riflette nella difficoltà estrema nel *computare* ossia *calcolare simbolicamente* la funzione di ripartizione del sistema, [21] e [30].

Il lattice quadrato con interazioni di vicinato, in assenza di campo esterno, è un grafo planare e infatti per il modello di Ising su tale grafo esistono soluzioni analitiche esatte. Il grafo bidimensionale con campo esterno e il lattice tridimensionale si dimostrano non essere planari. Da qui la loro NP-completezza e l'estrema difficoltà della rispettiva risoluzione analitica. Ancora oggi non sono state annunciate risoluzioni analitiche dei due modelli NP-completi ma una loro eventuale risoluzione esatta potrebbe chiarificare la questione sulle classi P ed NP accennata nel primo capitolo.

È ben evidente il *forte parallelismo* matematico tra Complessità Computazionale e risoluzione analitica esatta di un problema.

A prima vista sembrerebbe che l'algoritmo di tempra simulata sia la chiave della questione, poiché esso è in grado di individuare lo stato fondamentale ricercato in tempi polinomiali, come verificheremo più avanti. In realtà l'algoritmo di Metropolis non esplora l'intero spazio degli stati  $\Omega = (w_1, w_2, \dots, w_{2N})$ , che può anche essere numerabilmente infinito, ma solamente un suo sottoinsieme finito. Questa differenza *determina una precisa dinamica* del sistema in esame e, a sua volta, tale imposizione si traduce nella possibilità di non trovare effettivamente

un minimo globale della funzione costo ma piuttosto *un minimo locale* o *uno stato metastabile*, [26] e [28].

Ciò non costituisce un problema nella simulazioni dei modelli unidimensionali e bidimensionali, in assenza di campo esterno. Non è così nelle eventualità NP-Complete del problema. Ad esempio l'applicazione di un campo magnetico esterno  $B$  positivo ad un sistema completamente connesso con  $J = 1/N$  toglie la degenerazione dei due minimi di  $F$ , distinti da valori diversi dell'osservabile  $M$  e di cui solo uno rappresenterà effettivamente lo stato stazionario  $w_{ming}$  ricercato. Lo scenario che si presenta è quello di un minimo locale  $F(w_{minl})$  (cui compete  $M = -N$ ), separato da una barriera alta  $F(w_0)$  dal secondo minimo, quello globale  $F(w_{ming})$  (con magnetizzazione  $M = +N$ ). Se la configurazione iniziale  $w_i$  ha magnetizzazione minore di quella dello stato  $w_0$ , in cui  $F$  ha un massimo locale, il sistema può comunque raggiungere in tempi finiti la configurazione del minimo locale ed impiegare tempi esponenzialmente grandi nel risalire e proseguire verso l'effettivo minimo globale dell'energia libera. Tale fenomeno è noto come *intrappolamento nello stato metastabile*, ossia in una regione dello spazio delle configurazioni in cui il sistema transita per tempi esponenzialmente lunghi ma comunque finiti, non competendo a tale regione un punto di stabilità definito nel capitolo 2. Si dimostra, [28], che tale tempo di intrappolamento è dato dalla relazione:

$$t_{intr} = \exp \left[ \frac{N}{T} (F(w_0) - F(w_{minl})) \right] = \exp(N\alpha) \quad (4.30)$$

La conseguenza di questo discorso è che il tempo finito necessario per la convergenza del simulated annealing non è più un tempo polinomiale ma esponenziale. Un incauto fisico computazionale che effettuasse ad occhi chiusi una computazione per tempi minori<sup>11</sup> di  $t_{intr}$  ricaverebbe risultati del tutto erronei, poiché lo stato stazionario sarebbe quello con magnetizzazione negativa, chiara contraddizione con l'applicazione del campo magnetico positivo! La presenza di stati metastabili risiede non solo nel numero di connessioni del modello di Ising in esame, legato a sua volta alla dimensionalità del sistema, ma anche nella loro intensità, [28].

## 4.6 Annealing Simulato per il modello di Ising: Risultati

Applichiamo ora la dinamica di una macchina di Boltzmann e della tempra simulata per la simulazione di un modello di Ising. Si rimanda al codice scritto in una combinazione di C e ROOT/CINT. Per l'immagazzinamento dati si è ricorsi a delle *NTuple*, ossia sequenze ordinate di elementi. Effettuando l'*analisi asintotica dell'algoritmo* implementato, ossia computando il numero di passi effettuati per completare l'incarico, possiamo dire che per  $N$  unità computazionali esso impiega  $O(N^{\frac{3}{2}})$  passi in fase di puro calcolo ed  $O(N^2)$  passi in fase di calcolo e memorizzazione dello stato. Dunque l'algoritmo di simulated annealing qui implementato termina la computazione in tempi polinomiali. Tra le imposizioni esterne la più importante è la scelta di un programma di raffreddamento più rapido di  $T(t_n) = \frac{T_0}{1+\log(t_n)}$ , dove  $T_0$  è la temperatura iniziale e  $t_n$  il passo  $n$ -esimo dell'iterazione. Si è infatti cercato un compromesso tra un buon grado di convergenza e buone prestazioni a livello di tempo e di spazio. La soluzione è stata quella di un raffreddamento costante  $T_k = T_{k-1} - \Delta T$ , dove  $k$

<sup>11</sup>Per  $T = 0.7$ ,  $J = 1/N$  e  $h = 0.03$  si calcola  $\alpha \simeq 0.066$ . Per  $N = 1000$  spin, un ferromagnete simulabile anche con semplici PC, il tempo di intrappolamento sarebbe di circa  $10^{28}$  iterazioni. Alla frequenza degli odierni elaboratori di una iterazione ogni  $10^{-9}$ s, tale  $t_{intr}$  sarebbe pari a cento volte l'età dell'universo attualmente stimata.

indica un intero ciclo di rilassamento termico e  $\Delta T$  il salto termico. Questo *annealing schedule* ha garantito la convergenza per stati iniziali non troppo rumorosi. Il problema è che quando il sistema si porta alla  $T_c$  si iniziano a formare dei conglomerati di spin rivolti nella medesima direzione: si tratta, su piccola scala, di *domini magnetici*, [7,25,29]. A temperature ancora più basse queste strutture, senza un raffreddamento sufficientemente lento, possono diventare permanenti, poiché un singolo capovolgimento di uno spin viene accettato con probabilità via via minore. Per raggiungere la configurazione di equilibrio ci vorranno allora molte mosse, in modo da avere quelle dei nodi opportuni: si parla di *raffreddamento critico*, [7], fenomeno dovuto all'autocorrelazione dell'energia totale del sistema. Essa è altamente correlata per  $T \leq T_c$ , poiché la gran parte degli aggiornamenti delle unità computazionali possono essere rifiutati. Il tempo di rilassamento  $t_R$  necessario perché tale autocorrelazione vada a zero, grazie al rumore termico, è dato da  $t_R = cost \cdot \xi^z$ , dove  $\xi$  è la lunghezza di correlazione e  $z$  è detto *esponente critico dinamico*. Per il modello di Ising con dinamica di Metropolis  $z \simeq 2$ . Quando il sistema giunge e supera la  $T_c$ ,  $\xi$  cresce sino ad  $L$ , quindi si mantiene sui valori relativi alla dimensione dei domini formati. Ciò si traduce in un evidente incremento del tempo di rilassamento. Per evitare il raffreddamento critico è possibile modificare la dinamica degli aggiornamenti, [Wol89], in modo da capovolgere interi blocchi di spin. Pur tuttavia tali modifiche sono realmente efficaci solo per modelli di Ising di dimensione pari o superiore a 3. La strategia utilizzata qui è stata invece quella di costruire ogni ciclo in modo da avere un tempo di rilassamento nella forma  $t_R = a \cdot L^2$ , dove tuttavia  $a$  cresce con la temperatura. Si noti che in questa trattazione sul raffreddamento critico non sono date indicazioni sul programma di raffreddamento e dunque non è garantita la convergenza allo stato fondamentale per qualsiasi condizione iniziale.

Le simulazioni sono state effettuate su macchina virtuale Ubuntu 9.04 con 400 MB di RAM dedicati (la macchina reale è un Vaio VGN-AR11M con processore Intel Core Duo T2400, 1.83 GHz). Onde evitare eccessivi errori di arrotondamento si è evitato di normalizzare le quantità termodinamiche, conservandone l'esattezza in assenza di campo magnetico e per interazioni tra primi vicini.

### Modello di Ising 1d

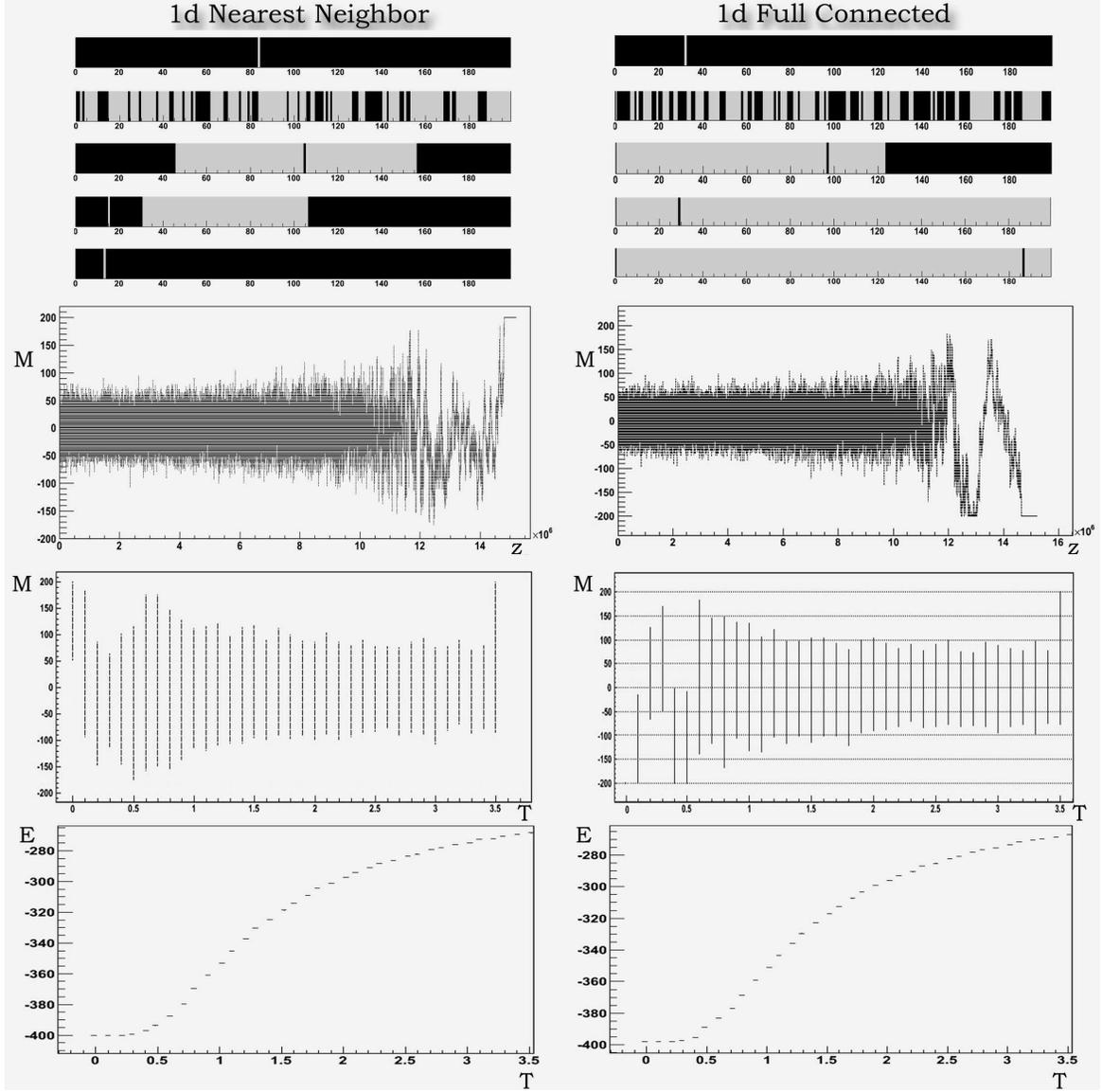
Supponiamo che il sistema si trovi in uno stato completamente ordinato, [24] e [Hua87], in cui magari tutti gli spin siano allineati all'insù, cioè  $S_i = 1$ . Per una siffatta catena, in assenza di campo magnetico esterno e con condizioni al contorno periodiche, l'energia dello stato con spin allineati è  $E_0 = -NJ$ , mentre l'entropia vale  $S_0 = \ln 1 = 0$ . Le configurazioni con energia immediatamente superiore sono tutte quelle con un solo spin capovolto. Vista l'arbitrarietà di tale spin, vi sono  $N$  configurazioni di questo tipo, con energia  $E_1 = E_0 + 2J$  ed entropia  $S_1 = \ln N$ . La variazione di energia libera dovuta a tali capovolgimenti è pertanto:

$$\Delta F = \Delta E - T\Delta S = 2J - T \ln(N) < 0 \quad \text{per } N \rightarrow \infty \quad (4.31)$$

Lo stato di spin allineati non corrisponde ad un punto di minimo di  $F$ , poiché il disordine dovuto al capovolgimento di uno spin permette al sistema di accedere a stati con energia libera minori. Se ne deduce che, per  $T \neq 0$ , la catena non raggiungerà mai stabilmente una fase ordinata, corrispondente alla fenomenologia ferromagnetica. Tuttavia essa presenta una transizione di fase per  $T = 0$ , come comprovato da risultati sperimentali, [Bir78]. Se poi le interazioni sono a lungo raggio e sufficientemente intense, vi è una transizione di fase anche

## 4 Il Modello di Ising

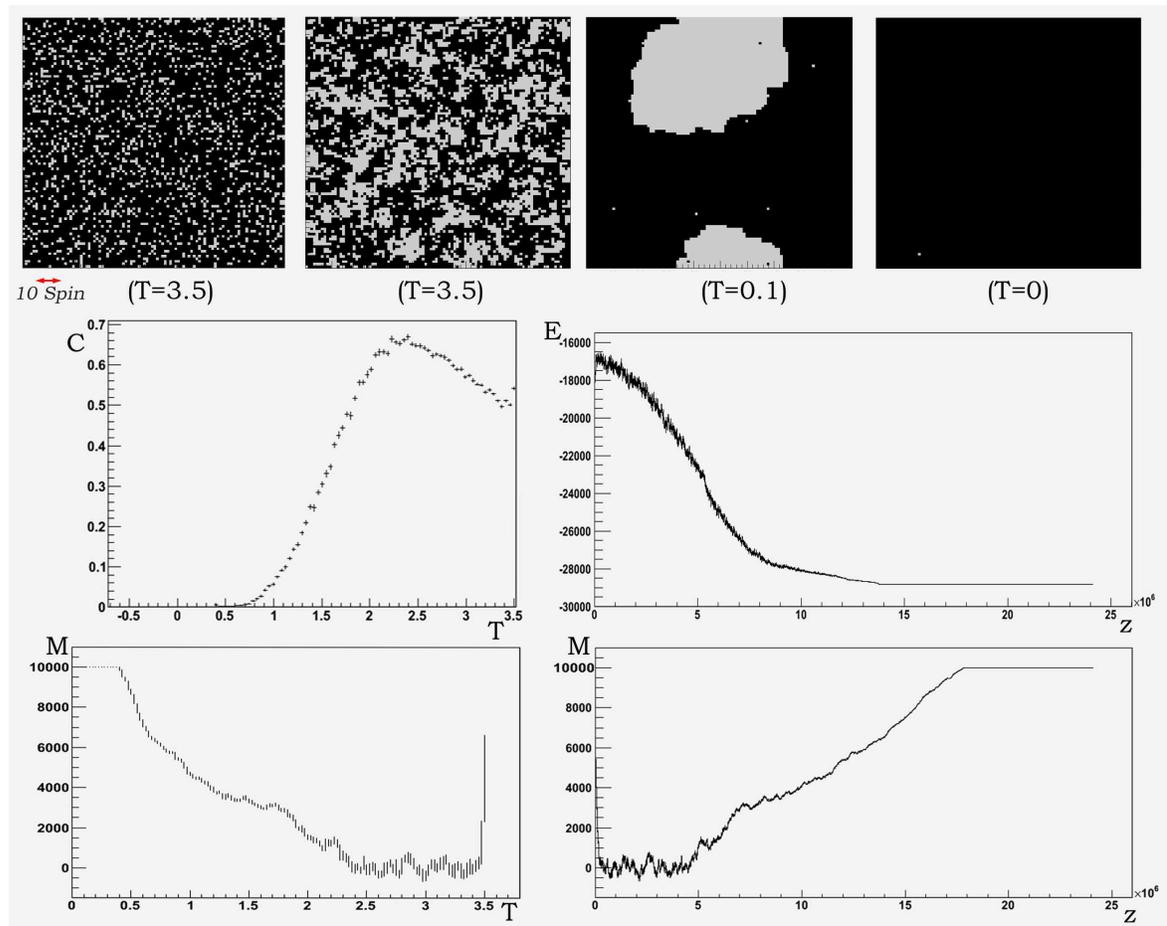
per  $T \neq 0$ , [Dys69]. Qui riportiamo due simulazioni per il modello di Ising 1d con interazioni a lungo raggio e a soli primi vicini.



Entrambi i risultati sono relativi ad una catena di  $N = 200$  spin, con interazioni a lungo raggio  $J_{ij} = \frac{1}{(i-j)^2}$  nel modello *Full Connected* o *FC* e di solo vicinato  $J_{ij} = J = 1$  per il *Nearest Neighbor* o *NN* e condizioni al contorno periodiche. Entrambe le catene partono con una configurazione completamente ordinata, ogni striscia nera indica un  $S_i = +1$ , mentre una grigia  $S_i = -1$ . In ogni istogramma di stato è rappresentato lo spin capovolto prima della fase di accettazione dell'aggiornamento. Quindi il rumore termico provvede al capovolgimento casuale degli spin. Al termine del primo ciclo, sempre per  $T = 3.5$ , le due catene hanno due configurazioni frastagliate e differenti. Per  $T = 0.1$  si nota la formazione di domini magnetici. Tuttavia alla prima iterazione per l'ultimo ciclo con  $T = 0$ , si nota che il *FC* ha già subito transizione di fase e si è portato in uno stato ordinato, mentre nel *NN* persiste una situazione di parziale disordine. Anche in tale modello, pur tuttavia, avviene una transizione di fase per

$T = 0$  e la catena raggiunge una configurazione ordinata. La rottura di ergodicità è evidente dall'analisi della magnetizzazione  $M$ , registrata per ogni iterazione  $z$ . Per  $T > T_c$ , cioè nella fase paramagnetica, il sistema tende all'unica misura di Gibbs stazionaria, cui corrisponde una  $M$  mediamente nulla. Appena si giunge e si supera  $T_c$  la magnetizzazione del sistema tende ad uno dei due estremi  $M = \pm N$ . Nella penultima coppia di istogrammi viene discretizzato questo andamento in base alla temperatura, l'ampiezza delle bande è proporzionale alla variazione complessiva di  $M$  in tutto il ciclo di rilassamento. Nel grafico  $E vs T$  appare evidente la minimizzazione dell'energia conseguente al raffreddamento. Le computazioni hanno richiesto un tempo di processamento in CPU di circa 23 s per la  $NN$  e 47 s per la  $FC$ . Si noti che risultati del  $FC$  sono in accordo con la teoria, [Frö82].

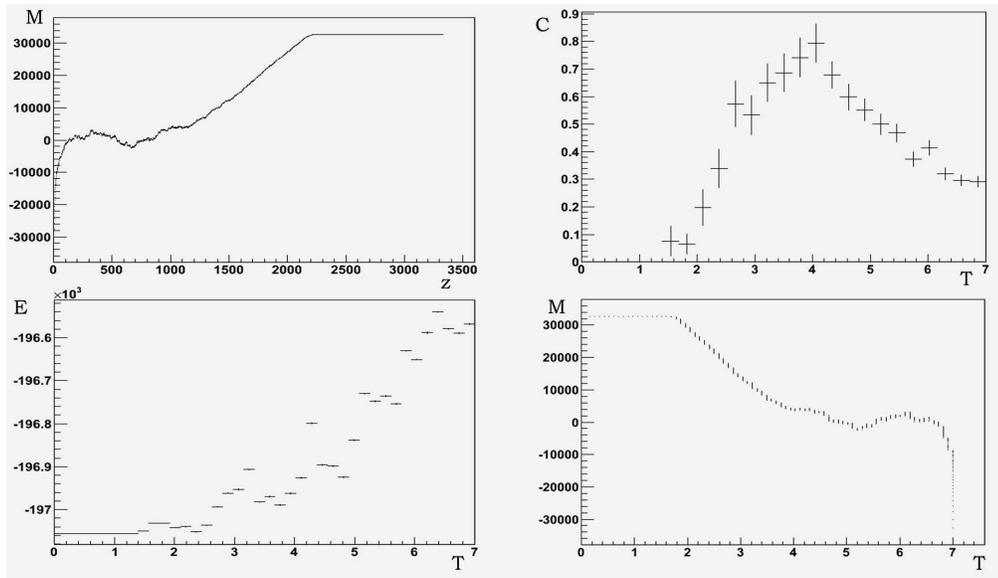
### Modello di Ising 2d con interazioni ai primi vicini



Qui si è simulato un reticolo quadrato composto da  $N = 10.000$  spin con interazioni  $NN$ . Questa volta la condizione iniziale ( $T = 3.5$ ) è rumorosa. Il rumore termico, già alla fine del primo ciclo, sempre per  $T = 3.5$ , ha modificato il numero di spin  $+1$  (neri) e  $-1$  (grigi). Già qui si osserva la formazione di domini. Il picco del calore specifico  $C$ , che non è simmetrico per via della finitudine del sistema e degli errori statistici, [29], e l'andamento della magnetizzazione complessiva (che vira dal valore medio nullo) individuano una temperatura di transizione di fase compresa tra 2.1 e 2.4. Eseguendo un fit con una gaussiana sulla parte sinistra dei

risultati relativi a  $C$  si individua un massimo per  $T'_c = (2.236 \pm 0.009)$ . Tali numeri sono in buon accordo con i risultati ottenuti per il limite termodinamico  $N \rightarrow \infty$ . Per  $T = 0.1$  si osserva un unico, viste le condizioni al contorno periodiche, dominio magnetico. Per evitare fenomeni di raffreddamento critico si è variata considerevolmente la costante  $a$  per i cicli a temperature più basse, si veda l'Appendice per ulteriori dettagli. Per  $T = 0$  il sistema raggiunge lo stato di equilibrio, determinato dalla magnetizzazione complessiva che aveva alla fine del ciclo più vicino alla temperatura critica. Le barre verticali nell'istogramma in basso a sinistra indicano la variazione di  $M$  per una data  $T$ . Un sistema con 10.000 spin ha  $2^{10.000} \simeq 10^{3010}$  stati possibili, dagli istogrammi su  $z$  emerge che la tempra simulata ne ha generati poco meno di  $2.5 \cdot 10^7$  prima di computare lo stato fondamentale, impiegando un tempo di 52 s. Non si è riportato l'andamento dell'energia poiché esso è analogo al caso  $d = 1$ .

### Modello di Ising 3d con interazioni ai primi vicini



Si è simulato un reticolo cubico contenente  $32^3 = 32768$  spin, con interazioni  $NN$ . La configurazione iniziale vede  $M = -32^3$ . Il sistema evolve rapidamente verso la situazione di magnetizzazione media nulla. Dall'analisi degli istogrammi, l'innalzamento di  $M$  verso il valore  $+32^3$  e il corrispondente picco nel calore specifico  $C$  individuano una transizione di fase nell'intervallo  $[4.2, 4.6]$ . Simulazioni Monte Carlo più raffinate, con una autocorrelazione minore, generatori di numeri pseudocasuali con periodi più lunghi, programmi di raffreddamento più lenti ed una dinamica di aggiornamento a *cluster*, individuano  $T_c = (4.52 \pm 0.03)$ , [Wan95]. Per evitare il raffreddamento critico si sono dovuti aumentare i tempi di rilassamento, solo qui proporzionali a  $L^3$ . Ciò potrebbe essere legato all'esponente critico  $z$  a sua volta dipendente dalla dinamica della simulazione. Questa simulazione ha richiesto un tempo di processamento di oltre 464 s. Per aumentarne la velocità si sono limitati gli accessi di scrittura dei risultati, immagazzinando solo 1 risultato ogni 10.000. È evidente la convergenza del ferromagnete allo stato stazionario determinato dalla  $M$  che il sistema aveva vicino alla temperatura critica, cioè nello stato iniziale della transizione di fase.

Risultati quantitativi rigorosi possono essere ricavati da simulazioni MC più raffinate di quelle riportate, applicando la teoria del *finite size scaling*, [Bin01].

# Conclusioni e Ringraziamenti

Le metodologie impiegate in questa tesi si sono dimostrate di uno squisito interesse fisico e matematico. Il punto di partenza è stata la *Teoria dei Grafi*, necessaria per la definizione formale di *Rete Neurale Artificiale* come entità in grado di rappresentare esattamente o approssimare *funzioni di rete* e costituita fondamentalmente da *unità computazionali interconnesse* in grado di svolgere *computazioni* elementari. Si sono quindi analizzate differenti strutture topologiche di ANN quali il *Neurone* di McCulloch-Pitts, il *Percettrone* di Rosenblatt, le *Reti Multistrato* ed infine le *Reti Ricorrenti*. Collateralmente si è accennato alla *Teoria della Complessità Computazionale*, necessaria per la classificazione degli algoritmi di risoluzione di determinati problemi. Tra le Reti Ricorrenti si sono analizzate le *Reti di Hopfield*, sfruttando il concetto di *dinamica*, utile per la comprensione della loro evoluzione temporale. Sebbene tali reti siano composte da neuroni strutturalmente *semplici*, esse presentano l'*emergenza* di proprietà collettive, su scala macroscopica, quali la presenza di *attrattori stabili* o la minimizzazione di quantità quali la *funzione energia*. Si è evidenziato più volte il *parallelismo fisico* tra il modello computazionale di tali ANN e la dinamica di un sistema discreto, ricorrendo anche ad esempi pratici quali il *Problema combinatorio del Commesso Viaggiatore*, in cui sono tornate prepotentemente la Teoria dei Grafi e la Teoria della Complessità Computazionale.

A completamento del suddetto parallelismo si è scavato nelle radici probabilistiche delle Reti di Hopfield, trattandone la generalizzazione stocastica nota in letteratura come *Macchina di Boltzmann*, a sua volta ottimo trampolino di lancio per lo studio degli *Insiemi Statistici*, della *Teoria Ergodica* e del formalismo relativo ai *Processi di Markov* e alla struttura generale degli algoritmi *Metaeuristici*, superiori per generalità alle controparti *Euristiche* ed in grado inoltre, come verificato, di giungere a *soluzioni esatte* di determinati problemi combinatori. Ed è ancora la *Combinatoria*, insieme ai grafi e alla *Meccanica Statistica*, la chiave di volta della *risoluzione esatta del modello di Ising*, introdotto nel 1920 da Lenz e cardine portante della teoria delle *Transizioni di Fase*. Attraverso gli strumenti matematici sviluppati precedentemente ed introdotti sempre in maniera informale ma esaustiva, si è verificato l'*isomorfismo presente tra il Modello di Ising ed il Modello di Hopfield*, generalizzabile sotto imposizioni più restrittive anche alle macchine di Boltzmann. Si è quindi portato a conclusione la tesi con la realizzazione originale di algoritmi di simulazione in una, due e tre dimensioni del modello di Ising attraverso il formalismo delle Reti Neurali Artificiali, con un occhio particolare rivolto all'*ergodicità* ed alla *complessità computazionale*.

L'intero lavoro vuole sottolineare la presenza di strumenti matematico-computazionali spesso messi da parte e che invece si dimostrano estremamente efficienti, in grado di giungere a soluzioni esatte o a risultati fondamentali e legati più che saldamente all'intuizione *Fisica*.

---

*Ringrazio il dott. Martello per il supporto costante e la libertà creativa che mi ha concesso in questi mesi, quindi un grazie va anche alla dott.ssa Cataldi per la sua disponibilità ed anche ai prof.ri Mario e Rosario Leo, che mi hanno celermente consigliato un testo completo sul modello di Ising.*

*Il grazie finale va alla mia famiglia, per avermi sempre supportato in tutto e per tutto.*

# Appendice

Qui si riportano i programmi relativi ai modelli di Ising 1d, 2d e 3d. Tutti e quattro gli algoritmi contemplano la possibilità di un campo magnetico esterno, annullato nelle simulazioni per motivi attinenti ai tempi computazionali a disposizione. Si noti che nei codici di queste simulazioni non si è implementato il valore medio di  $C$ , che viene calcolato in interattivo da ROOT.

ISING 1D-NEARESTNEIGHBOR:

```
#include <stdio.h>
#include <stdlib.h>
//Ecco gli include da ROOT
#include "TH2.h"
#include "TFile.h"
#include "TMath.h"
#include "Riostream.h"
#include "TRandom2.h"
#include "TNtupleD.h"
#include "TStopwatch.h"
#define L 200

int S[L+2]; //Vettore degli spin del reticolo
float B=0.; //Campo magnetico, qui è omogeneo
float T=3.5; //Temperatura
float DT=0.1; //Intervallo di Diminuzione della Temperatura
float E=0.; //Differenza di Energia delle configurazioni
int i,j,a,b,z; //Coordinate per gli spin e counter
float P=0.; //Misura di Gibbs associata
float choic=0.; //Numero casuale per l'accettazione in caso di E>0
int t=0; //Segnalino di rilassamento, t è locale ad una T, r è globale
int M=0; //Magnetizzazione della configurazione
float Et=0.; //Energia totale della configurazione
float C=0.; //Calore specifico del sistema
int md=25000; //Modulatore, regola la frequenza di stampa degli stati
int bag=10; //Bagno, determina il tempo di rilassamento
char name[40], title[40]; //Etichette per gli istogrammi in uscita

main() {
TRandom2 gen(0); //Inizializzazione del seme
TH2C* h; //Puntatore ad un istogramma 2D di short – Istogramma Stati TNtupleD
nt("nt","NTupla di Riferimento","z:T:t:Et:M:C");
//Una Ntuple è una sequenza ordinata di N insiemi
double x[6]; //Vettore dimensionale del numero di insiemi della Ntuple nt
TStopwatch timer; TFile f("IsingH.root","recreate"); //Repository Dati
z=1; //z ha il ruolo di t ma coinvolge tutte le temperature
h = new TH2C("0 StatoIniziale","Stato Iniziale",L,0,L-1,1,0,1);

timer.Start();
for(a=1;a<L+1;a++) { //Inizializzazione degli spin (stato iniziale "caldo")
```

## Appendice

```

    if (gen.Uniform(0,3*L) <= -1) S[a]=-1;
    else S[a]=1;
    h->SetBinContent(a,1,S[a]);      }
    h->Write(); //Scriviamo il primo istogramma di stato
//Condizioni al contorno di tipo periodico
S[0]=S[L]; S[L+1]=S[1];
//Si copia la cella L prima della 1 e la 1 dopo la L. Il sistema è un anello
for(a=1;a<L+1;a++) { //Calcoliamo M e Et
    M=M+S[a];
    Et=Et-S[a]*(S[a+1]+S[a-1]+B); }
//Conclusa questa FASE INIZIALE passiamo al ciclo principale del programma
while(T > DT) {
    for (t=0;t<bag*L*L;t++) {
//Condizioni al contorno periodico
        S[0]=S[L]; S[L+1]=S[1];
//Random Flip di uno spin
        i=gen.Uniform(0,L)+1;
        if (S[i]==1) { S[i]=-1; M=M-2; }
        else { S[i]=1; M=M+2; }
//Corpo Opzionale di Stampa dello Stato
        if (t%and==0) {
            sprintf(name,"%f-%d",T,t);
            sprintf(title,"Temp:%f-Passo:%d-Magn:%d",T,t,M);
            h = new TH2C(name,title,L,0,L-1,1,0,1);
            for(a=1;a<L+1;a++) {
                h->SetBinContent(a,1,S[a]); }
            h->Write(); }
//La variazione E sarà dovuta all'interazione dello spin capovolto con i circostanti
            E=-2*S[i]*(S[i+1]+S[i-1])+B;
            if (E>0) {
                P=1/(1+TMath::Exp(E/T));
                choic = gen.Uniform(0.,1000.)/999.;
                if (P <= choic) {
                    if (S[i]==1) { S[i]=-1; M=M-2; E=0.;}
                    else { S[i]=1; M=M+2; E=0.; } }// M ed Et tra stati sono legate
                } Et=Et+E;
//Punto di Calcolo del Calore Specifico - C=dU/dT (U=Et-B) ->
                C^2=E*E/T*T
                if (E==0) C=0;
                else C=((E-B)*(E-B))/(T*T); //Passo dall'energia interna a quella totale
//Chiudo il Punto di Calcolo di C con una inizializzazione
//Salvo M, Et, t e z nella NTupla
                x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M; x[5]=C;
                nt.Fill(x);
                z=z+1;
//Inizializzo la differenza di energia energia
                E=0.;
//La macchina di Boltzmann deve ripetere l'iterazione per trovare il minimo
                } //Raffreddiamo il sistema
                T=T-DT;
                if (T <= 1.3 && T > 0.6) bag=11;
                if (T <= 0.6 && T > 0.2) bag=12;
                }
T=0.;
bag=13;
if(T <= 0.) { //La macchina di Boltzmann degenera in rete di Hopfield
//Adesso infatti T=0.
    for (t=0;t<bag*L*L;t++) { //Per T basse il tempo di rilassamento è maggiore

```

## Appendice

```

S[0]=S[L]; S[L+1]=S[1]; //Random Flip di uno spin
i=gen.Uniform(0,L)+1;
if (S[i]==1) { S[i]=-1; M=M-2; }
else { S[i]=1; M=M+2; } //Corpo Opzionale di Stampa dello Stato
if (t%nd==0) {
    sprintf(name,"H%f-%d",T,t);
    sprintf(title,"Temp:%f-Passo:%d-Magn:%d",T,t,M);
    h = new TH2C(name,title,L,0,L-1,1,0,1);
    for(a=1;a<L+1;a++) { h->SetBinContent(a,1,S[a]); }
    h->Write(); } //Calcoliamo la variazione di energia
E=-2*S[i]*(S[i+1]+S[i-1])+B;
//Se E<0 accettiamo lo stato, altrimenti lo rifiutiamo
if (E>0.) {
    if (S[i]==1) { S[i]=-1; M=M-2; E=0.; }
    else { S[i]=1; M=M+2; E=0.; }
    Et=Et+E;
    x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M;
    nt.Fill(x);
    z=z+1;
    E=0.; //Inizializzo la differenza di energia }
}
nt.Write();
f.Close();
timer.Stop();
printf("Computazione terminata. TR: %f / Cpu: %f \n",timer.RealTime(),timer.CpuTime());
}

```

### ISING 1D-FULLCONNECTED:

```

#include <stdio.h>
#include <stdlib.h>
//Ecco gli include da ROOT
#include "TH2.h"
#include "TFile.h"
#include "TMath.h"
#include "Riostream.h"
#include "TRandom2.h"
#include "TNTupleD.h"
#include "TStopwatch.h"
#define L 200

int S[L+2]; //Vettore degli spin del reticolo
float J[L+2][L+2]; //Matrice delle interazioni
float B=0.; //Campo magnetico, qui è omogeneo
float T=3.5; //Temperatura
float DT=0.1; //Intervallo di Diminuzione della Temperatura
float E=0.; //Differenza di Energia delle configurazioni
int i,j,a,b,z; //Coordinate per gli spin e counter
float P=0.; //Misura di Gibbs associata
float choic=0.; //Numero casuale per l'accettazione in caso di E>0
int t=0; //Segnalino di rilassamento, t è locale ad una T, r è globale
int M=0; //Magnetizzazione della configurazione
float Et=0.; //Energia totale della configurazione
float C=0.; //Calore specifico del sistema
int md=25000; //Modulatore, regola la frequenza di stampa degli stati
int bag=10; //Bagno, determina il tempo di rilassamento
char name[40], title[40]; //Etichette per gli istogrammi in uscita

```

## Appendice

```

main() {
TRandom2 gen(0); //Inizializzazione del seme
TH2C* h; //Puntatore ad un istogramma 2D di short - Istogramma Stati TNtupleD
nt("nt", "NTupla di Riferimento", "z:T:t:Et:M:C");
//Una Ntuple è una sequenza ordinata di N insiemi
double x[6]; //Vettore dimensionale del numero di insiemi della Ntuple nt
TStopwatch timer; TFile f("IsingH.root", "recreate"); //Repository Dati
z=1; //z ha il ruolo di t ma coinvolge tutte le temperature
h = new TH2C("0 StatoIniziale", "Stato Iniziale", L, 0, L-1, 1, 0, 1);

timer.Start();
for(a=1; a<L+1; a++) { //Inizializzazione degli spin (stato iniziale "caldo")
    if (gen.Uniform(0, 3*L) <= -1) S[a]=-1;
    else S[a]=1;
    h->SetBinContent(a, 1, S[a]); }
h->Write(); //Scriviamo il primo istogramma di stato
//Condizioni al contorno di tipo periodico
S[0]=S[L]; S[L+1]=S[1];
//Si copia la cella L prima della 1 e la 1 dopo la L. Il sistema è un anello
for(a=1; a<L+1; a++) { //Calcoliamo M e Et
    for(b=1; b<L+1; b++) {
        if(a==b) J[a][b]=0;
        else J[a][b]=1/((a-b)*(a-b));
        Et=Et-S[a]*(J[a][b]*S[b]+B); }
    M=M+S[a]; }
//Conclusa questa FASE INIZIALE passiamo al ciclo principale del programma
while(T > DT) {
    for (t=0; t<bag*L*L; t++) {
//Condizioni al contorno periodico
        S[0]=S[L]; S[L+1]=S[1];
//Random Flip di uno spin
        i=gen.Uniform(0, L)+1;
        if (S[i]==1) { S[i]=-1; M=M-2; }
        else { S[i]=1; M=M+2; }
//Corpo Opzionale di Stampa dello Stato
        if (t%nd==0) {
            sprintf(name, "H%f-%d", T, t);
            sprintf(title, "Temp:%f-Passo:%d-Magn:%d", T, t, M);
            h = new TH2C(name, title, L, 0, L-1, 1, 0, 1);
            for(a=1; a<L+1; a++) {
                h->SetBinContent(a, 1, S[a]); }
            h->Write(); }
//La variazione E sarà dovuta all'interazione dello spin capovolto con gli altri N-1
            for(b=1; b<L+1; b++) {
                E=E-2*S[i]*(J[i][b]*S[b]); }
            E=E+B;
            if (E>0) {
                P=1/(1+TMath::Exp(E/T));
                choic = gen.Uniform(0., 1000.)/999.;
                if (P <= choic) {
                    if (S[i]==1) { S[i]=-1; M=M-2; E=0.; }
                    else { S[i]=1; M=M+2; E=0.; } } //M ed Et tra stati sono legate
                } Et=Et+E;
//Punto di Calcolo del Calore Specifico - C=dU/dT (U=Et-B) ->
                C^2=E*E/T*T
                if (E==0) C=0;
                else C=((E-B)*(E-B))/(T*T); //Passo dall'energia interna a quella totale

```

## Appendice

```

//Chiudo il Punto di Calcolo di C con una inizializzazione
//Salvo M, Et, t e z nella NTupla
x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M; x[5]=C;
nt.Fill(x);
z=z+1;
//Inizializzo la differenza di energia energia
E=0.;
//La macchina di Boltzmann deve ripetere l'iterazione per trovare il minimo
} //Raffreddiamo il sistema
T=T-DT;
if (T <= 1.3 && T > 0.6) bag=11;
if (T <= 0.6 && T > 0.2) bag=12;
}
T=0.;
bag=13;
if(T <= 0.) { //La macchina di Boltzmann degenera in rete di Hopfield
//Adesso infatti T=0.
for (t=0;t<bag*L*L;t++) { //Per T basse il tempo di rilassamento è maggiore
S[0]=S[L]; S[L+1]=S[1]; //Random Flip di uno spin
i=gen.Uniform(0,L)+1;
if (S[i]==1) { S[i]=-1; M=M-2; }
else { S[i]=1; M=M+2; } //Corpo Opzionale di Stampa dello Stato
if (t%nd==0) {
sprintf(name,"H%f-%d",T,t);
sprintf(title,"Temp:%f-Passo:%d-Magn:%d",T,t,M);
h = new TH2C(name, title ,L,0,L-1,1,0,1);
for(a=1;a<L+1;a++) { h->SetBinContent(a,1,S[a]); }
h->Write(); }
//Calcoliamo la variazione di energia
for(b=1;b<L+1;b++) {
E=E-2*S[i]*(J[i][b]*S[b]); }
E=E+B;
//Se E<0 accettiamo lo stato, altrimenti lo rifiutiamo
if (E>0.) {
if (S[i]==1) { S[i]=-1; M=M-2; E=0.; }
else { S[i]=1; M=M+2; E=0.; } }
Et=Et+E;
x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M;
nt.Fill(x);
z=z+1;
E=0.; //Inizializzo la differenza di energia }
}
nt.Write();
f.Close();
timer.Stop();
printf("Computazione terminata. TR: %f / Cpu: %f \n",timer.RealTime(),timer.CpuTime());
}

```

ISING2D-NEARESTNEIGHBOR:

```

#include <stdio.h>
#include <stdlib.h>
//Ecco gli include da ROOT
#include "TH2.h"
#include "TFile.h"
#include "TMath.h"
#include "Riostream.h"
#include "TRandom2.h"

```

## Appendice

```

#include "TNtupleD.h"
#include "TStopwatch.h"
#define L 100

int S[L+2][L+2];           //Matrice degli spin del reticolo
float B=0.;                //Campo magnetico, qui è omogeneo
float T=3.5;               //Temperatura
float DT=0.025;           //Intervallo di Diminuzione della Temperatura
float E=0.;                //Differenza di Energia delle configurazioni
int i,j,a,b,z;            //Coordinate per gli spin e counter
float P=0.;                //Misura di Gibbs associata
float choic=0.;           //Numero casuale per l'accettazione in caso di E>0
int t=0;                  //Segnalino di rilassamento, t è locale ad una T, r è globale
int M=0;                  //Magnetizzazione della configurazione
float Et=0.;              //Energia totale della configurazione
float C=0.;                //Calore specifico del sistema
int md=30000;             //Modulatore, regola la frequenza di stampa degli stati
int bag=10;                //Bagno, determina il tempo di rilassamento
char name[40], title[40]; //Etichette per gli istogrammi in uscita

main() {
TRandom2 gen(0); //Inizializzazione del seme
TH2C* h;         //Puntatore ad un istogramma 2D di short – Istogramma Stati TNtupleD
nt("nt", "NTupla di Riferimento", "z:T:t:Et:M:C");
//Una Ntuple è una sequenza ordinata di N insiemi
double x[6]; //Vettore dimensionale del numero di insiemi della Ntuple nt
TStopwatch timer; TFile f("IsingH.root", "recreate"); //Repository Dati
z=1; //z ha il ruolo di t ma coinvolge tutte le temperature
h = new TH2C("0 StatoIniziale", "Stato Iniziale", L,0,L-1,L,0,L-1);

timer.Start();
for(a=1;a<L+1;a++) { for(b=1;b<L+1;b++) {
//Inizializzazione degli spin (stato "caldo")
if (gen.Uniform(0,3*L) <= 4*L) S[a][b]=-1;
else S[a][b]=1;
h->SetBinContent(a,b,S[a][b]); } }
h->Write(); //Scriviamo il primo istogramma di stato
//Condizioni al contorno di tipo periodico
for(a=1;a<L+1;a++) {
S[0][a]=S[L][a]; S[L+1][a]=S[1][a]; S[a][0]=S[a][L]; S[a][L+1]=S[a][1];
//Si copiano i piani uno adiacente all'altro. Il sistema è richiuso su sè stesso.
//Calcoliamo M e Et
for(b=1;b<L+1;b++) {
M=M+S[a][b];
Et=Et-S[a][b]*(S[a+1][b]+S[a-1][b]+S[a][b-1]+S[a][b+1]+B); } }
//Conclusa questa FASE INIZIALE passiamo al ciclo principale del programma
while(T > DT) {
for (t=0;t<bag*L*L;t++) {
//Condizioni al contorno di tipo periodico
for(a=0;a<L+1;a++) {
S[0][a]=S[L][a]; S[L+1][a]=S[1][a]; S[a][0]=S[a][L]; S[a][L+1]=S[a][1];
}
i=gen.Uniform(0,L)+1;
j=gen.Uniform(0,L)+1;
if (S[i][j]==1) { S[i][j]=-1; M=M-2; }
else { S[i][j]=1; M=M+2; }
//Corpo Opzionale di Stampa dello Stato

```

## Appendice

```

if (t%nd==0) {
    sprintf(name, "H%f-%d", T, t);
    sprintf(title, "Temp:%f-Passo:%d-Magn:%d", T, t, M);
    h = new TH2C(name, title, L, 0, L-1, 1, 0, 1);
    for (a=1; a<L+1; a++) {
        for (b=1; b<L+1; b++) {
            h->SetBinContent(a, b, S[a][b]); } }
    h->Write(); }
//La variazione E sarà dovuta all'interazione dello spin capovolto con i circostanti
E=-2*S[i][j]*(S[i+1][j]+S[i-1][j]+S[i][j-1]+S[i][j+1])+B;
if (E>0) {
    P=1/(1+TMath::Exp(E/T));
    choic = gen.Uniform(0., 1000.)/999.;
    if (P <= choic) {
        if (S[i][j]==1) { S[i][j]=-1; M=M-2; E=0.;}
        else { S[i][j]=1; M=M+2; E=0.; } //M ed Et tra stati sono legate
    } Et=Et+E;
//Punto di Calcolo del Calore Specifico - C=dU/dT (U=Et-B) ->
C^2=E*E/T*T
if (E==0) C=0;
else C=((E-B)*(E-B))/(T*T); //Passo dall'energia interna a quella totale
//Chiudo il Punto di Calcolo di C con una inizializzazione
//Salvo M, Et, t e z nella NTupla
x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M; x[5]=C;
nt.Fill(x);
z=z+1;
//Inizializzo la differenza di energia energia
E=0.;
//La macchina di Boltzmann deve ripetere l'iterazione per trovare il minimo
} //Raffreddiamo il sistema
T=T-DT;
if (T <= 1.3 && T > 0.6) bag=11;
if (T <= 0.6 && T > 0.2) bag=12;
}
T=0.;
bag=13;
if(T <= 0.) { //La macchina di Boltzmann degenera in rete di Hopfield
//Adesso infatti T=0.
for (t=0; t<bag*L*L; t++) { //Per T basse il tempo di rilassamento è maggiore
//Condizioni al contorno di tipo periodico
for (a=0; a<L+1; a++) {
    S[0][a]=S[L][a]; S[L+1][a]=S[1][a]; S[a][0]=S[a][L]; S[a][L+1]=S[a][1];
}
i=gen.Uniform(0, L)+1;
j=gen.Uniform(0, L)+1;
if (S[i][j]==1) { S[i][j]=-1; M=M-2; }
else { S[i][j]=1; M=M+2; }
//Corpo Opzionale di Stampa dello Stato
if (t%nd==0) {
    sprintf(name, "H%f-%d", T, t);
    sprintf(title, "Temp:%f-Passo:%d-Magn:%d", T, t, M);
    h = new TH2C(name, title, L, 0, L-1, 1, 0, 1);
    for (a=1; a<L+1; a++) {
        for (b=1; b<L+1; b++) {
            h->SetBinContent(a, b, S[a][b]); } }
    h->Write(); } //Calcoliamo la variazione di energia
E=-2*S[i][j]*(S[i+1][j]+S[i-1][j]+S[i][j-1]+S[i][j+1])+B;
//Se E<0 accettiamo lo stato, altrimenti lo rifiutiamo

```

## Appendice

```
    if (E>0.) {
        if (S[i][j]==1) { S[i][j]=-1; M=M-2; E=0.;}
        else { S[i][j]=1; M=M+2; E=0.; } }
    Et=Et+E;
    x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M;
    nt.Fill(x);
    z=z+1;
    E=0.; //Inizializzo la differenza di energia }
}
nt.Write();
f.Close();
timer.Stop();
printf("Computazione terminata. TR: %f / Cpu: %f \n",timer.RealTime(),timer.CpuTime());
}
```

### ISING3D-NEARESTNEIGHBOR:

```
#include <stdio.h>
#include <stdlib.h>
//Ecco gli include da ROOT
#include "TH2.h"
#include "TFile.h"
#include "TMath.h"
#include "Riostream.h"
#include "TRandom2.h"
#include "TNtupleD.h"
#include "TStopwatch.h"
#define L 32

int S[L+2][L+2][L+2]; //Tensore degli spin del reticolo
float B=0.; //Campo magnetico, qui è omogeneo
float T=7.; //Temperatura
float DT=0.07; //Intervallo di Diminuzione della Temperatura
float E=0.; //Differenza di Energia delle configurazioni
int i,j,k,a,b,c,z; //Coordinate per gli spin e counter
float P=0.; //Misura di Gibbs associata
float choic=0.; //Numero casuale per l'accettazione in caso di E>0
int t=0; //Segnalino di rilassamento, t è locale ad una T, r è globale
int M=0; //Magnetizzazione della configurazione
float Et=0.; //Energia totale della configurazione
float C=0.; //Calore specifico del sistema
int md=10000; //Modulatore, regola la frequenza di stampa degli stati
int mdd=10000; //Modulatore, regola la frequenza di stampa sulla Ntupla
int bag=10; //Bagno, determina il tempo di rilassamento
char name[40], title[40]; //Etichette per gli istogrammi in uscita

main() {
TRandom2 gen(0); //Inizializzazione del seme
TH3C* h; //Puntatore ad un istogramma 2D di short - Istogramma Stati TNtupleD
nt("nt","NTupla di Riferimento","z:T:t:Et:M:C");
//Una Ntuple è una sequenza ordinata di N insiemi
double x[6]; //Vettore dimensionale del numero di insiemi della Ntupla nt
TStopwatch timer; TFile f("IsingH.root","recreate"); //Repository Dati
z=1; //z ha il ruolo di t ma coinvolge tutte le temperature
h = new TH3C("0 StatoIniziale","Stato Iniziale",L,0,L-1,L,0,L-1,L,0,L-1);

timer.Start();
```

## Appendice

```

for (a=1;a<L+1;a++) { for (b=1;b<L+1;b++) { for (c=1;c<L+1;c++) {
//Inizializzazione degli spin (stato "caldo")
  if (gen.Uniform(0,3*L) <= 4*L) S[a][b][c]=-1;
  else S[a][b][c]=1;
  h->SetBinContent(a,b,c,S[a][b][c]);    } } }
h->Write(); //Scriviamo il primo istogramma di stato
//Condizioni al contorno di tipo periodico
for (a=1;a<L+1;a++) { for (b=1;b<L+1;b++) {
  S[0][a][b]=S[L][a][b]; S[L+1][a][b]=S[1][a][b];
  S[a][0][b]=S[a][L][b]; S[a][L+1][b]=S[a][1][b];
  S[a][b][0]=S[a][b][L]; S[a][b][L+1]=S[a][b][1];
//Si copiano i piani uno adiacente all'altro. Il sistema è richiuso su sè stesso.
  for (c=1;c<L+1;c++) {
//Calcoliamo M e Et
    M=M+S[a][b][c];
    Et=Et-S[a][b][c]*(S[a+1][b][c]+S[a-1][b][c]+S[a][b-1][c]+
//Interruzione per l'impaginazione
S[a][b+1][c]+S[a][b][c+1]+S[a][b][c-1]+B);
  } } }
//Conclusa questa FASE INIZIALE passiamo al ciclo principale del programma
while(T > DT) {
  for (t=0;t<bag*L*L*L;t++) {
//Condizioni al contorno di tipo periodico
    for (a=1;a<L+1;a++) { for (b=1;b<L+1;b++) {
      S[0][a][b]=S[L][a][b]; S[L+1][a][b]=S[1][a][b];
      S[a][0][b]=S[a][L][b]; S[a][L+1][b]=S[a][1][b];
      S[a][b][0]=S[a][b][L]; S[a][b][L+1]=S[a][b][1];    } }
    i=int (gen.Uniform(0,L)+1);
    j=int (gen.Uniform(0,L)+1);
    k=int (gen.Uniform(0,L)+1);
    if (S[i][j][k]==1) { S[i][j][k]=-1; M=M-2; }
    else { S[i][j][k]=1; M=M+2; }
//Corpo Opzionale di Stampa dello Stato
    if (t%nd==0) {
      sprintf(name,"H%f-%d",T,t);
      sprintf(title,"Temp:%f-Passo:%d-Magn:%d",T,t,M);
      h = new TH3C(name,title,L,0,L-1,L,0,L-1,L,0,L-1);
      for (a=1;a<L+1;a++) { for (b=1;b<L+1;b++) { for (c=1;c<L+1;c++) {
        h->SetBinContent(a,b,c,S[a][b][c]);    } } }
      h->Write(); }
//La variazione E sarà dovuta all'interazione dello spin capovolto con i circostanti
    E=-2*S[i][j][k]*(S[i+1][j][k]+S[i-1][j][k]+S[i][j-1][k]+
//Interruzione per l'impaginazione
S[i][j+1][k]+S[i][j][k+1]+S[i][j][k-1]+B);
    if (E>0) {
      P=1/(1+TMath::Exp(E/T));
      choic = gen.Uniform(0.,1000.)/999.;
      if (P <= choic) {
        if (S[i][j][k]==1) { S[i][j][k]=-1; M=M-2; E=0.; }
        else { S[i][j][k]=1; M=M+2; E=0.; } } //M ed Et tra stati sono legate
      } Et=Et+E;
//Punto di Calcolo del Calore Specifico - C=dU/dT (U=Et-B) ->
    if (t%ndd==0) {
      C^2=E*T/T;
      if (E==0) C=0;
      else C=((E-B)*(E-B))/(T*T); //Passo dall'energia interna a quella totale
//Chiudo il Punto di Calcolo di C con una inizializzazione
//Salvo M, Et, t e z nella NTupla

```

## Appendice

```

    x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M; x[5]=C;
    nt.Fill(x);
    z=z+1;
}
//Inizializzo la differenza di energia energia
E=0.;
//La macchina di Boltzmann deve ripetere l'iterazione per trovare il minimo
} //Raffreddiamo il sistema
T=T-DT;
if (T <= 3.5 && T > 1.6) bag=14;
if (T <= 1.6 && T > 0.2) bag=18;
}
T=0.;
bag=23;
if(T <= 0.) { //La macchina di Boltzmann degenera in rete di Hopfield
//Adesso infatti T=0.
for (t=0;t<bag*L*L*L;t++) { //Per T basse il tempo di rilassamento è maggiore
//Condizioni al contorno di tipo periodico
for(a=1;a<L+1;a++) { for(b=1;b<L+1;b++) {
S[0][a][b]=S[L][a][b]; S[L+1][a][b]=S[1][a][b];
S[a][0][b]=S[a][L][b]; S[a][L+1][b]=S[a][1][b];
S[a][b][0]=S[a][b][L]; S[a][b][L+1]=S[a][b][1]; } }
i=int(gen.Uniform(0,L)+1);
j=int(gen.Uniform(0,L)+1);
k=int(gen.Uniform(0,L)+1);
if (S[i][j][k]==1) { S[i][j][k]=-1; M=M-2; }
else { S[i][j][k]=1; M=M+2; }
//Corpo Opzionale di Stampa dello Stato
if (t%nd==0) {
sprintf(name,"H%f-%d",T,t);
sprintf(title,"Temp:%f-Passo:%d-Magn:%d",T,t,M);
h = new TH3C(name,title,L,0,L-1,L,0,L-1,L,0,L-1);
for(a=1;a<L+1;a++) { for(b=1;b<L+1;b++) { for(c=1;c<L+1;c++) {
h->SetBinContent(a,b,c,S[a][b][c]); } } }
h->Write(); } //Calcoliamo la variazione di energia
E=-2*S[i][j][k]*(S[i+1][j][k]+S[i-1][j][k]+S[i][j-1][k]+
//Interruzione per l'impaginazione
S[i][j+1][k]+S[i][j][k+1]+S[i][j][k-1])+B;
//Se E<0 accettiamo lo stato, altrimenti lo rifiutiamo
if (E>0.) {
if (S[i][j][k]==1) { S[i][j][k]=-1; M=M-2; E=0.; }
else { S[i][j][k]=1; M=M+2; E=0.; } }
Et=Et+E;
x[0]=z; x[1]=T; x[2]=t; x[3]=Et; x[4]=M;
nt.Fill(x);
z=z+1;
E=0.; //Inizializzo la differenza di energia }
}
nt.Write();
f.Close();
timer.Stop();
printf("Computazione terminata. TR: %f / Cpu: %f \n",timer.RealTime(),timer.CpuTime());
}

```

# Bibliografia e Riferimenti

- [1] C. Domeniconi, M. Jordan. *Discorsi sulle reti neurali e l'apprendimento*. FRANCOANGELI (2001).
- [2] R. Rojas. *Neural Networks: A Systematic Introduction*. SPRINGER-VERLAG (1996).
- [3] L.A.N. Amaral, J.M. Ottimo. *Complex Networks*. EUROPEAN PHYSICAL JOURNAL B-38, 147-162 (2004).
- [4] M. Sipser. *Introduction to the Theory of Computation, Second Edition*. THOMSON COURSE TECHNOLOGY (2006).
- [5] S. Cook. *The P versus NP Problem* (article). J. Carlson et al. *The Millennium Prize Problems*. CLAY MATHEMATICS INSTITUTE, AMERICAN MATHEMATICAL SOCIETY (2000).
- [6] H. G. Schaap. *Ising models and Neural Networks*. PH.D. THESIS. CENTRE FOR THEORETICAL PHYSICS IN GRONINGEN (2005).
- [7] T. Pang. *An Introduction to Computational Physics*. CAMBRIDGE UNIVERSITY PRESS (2006).
- [8] A. M. Turing. *Computing machinery and intelligence*. MIND, 59, 433-460 (1950).
- [9] J. J. Hopfield. *Neurons with graded response have collective computational properties like those of two-state neurons*. PROC. NATL. ACAD. SCI. USA VOL. 81, PP. 3088-3092 (1984).
- [10] J. J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities*. PROC. NATL. ACAD. SCI. USA VOL. 79, PP. 2554-2558 (1982).
- [11] H. Goldstein, C. Poole, J. Safko. *Meccanica Classica*. ZANICHELLI (2005).
- [12] R. Karp. *Reducibility among combinatorial problems*. UNIVERSITY OF CALIFORNIA (1972).
- [13] J. J. Hopfield, D. W. Tank. *"Neural" Computation of Decisions in Optimization Problems*. BIOL. CYBERN: 52, 141-152 (1985).
- [14] D. H. Ackley, G. E. Hinton, T. J. Sejnowski. *A Learning Algorithm for Boltzmann Machines*. COGNITIVE SCIENCE 9, 147-169 (1985).
- [15] E. Aarts, J. Korst. *Simulated Annealing and Boltzmann Machines: a stochastic approach to combinatorial optimization and neural computing*. WILEY (1989).
- [16] E. Aarts, J. K. Lenstra. *Local search in combinatorial optimization*. WILEY (1997).
- [17] J. A. Freeman, D. M. Skapura. *Neural networks: algorithms, applications, and programming techniques*. ADDISON-WESLEY (1991).
- [18] R. Kinderman, J. L. Snell. *Markov Random Fields and their Applications*. AMS, PROVIDENCE (1980).
- [19] D. J. Amit. *Modeling Brain Function The world of attractor neural networks*. CAMBRIDGE UNIVERSITY PRESS (1989).

## Bibliografia e Riferimenti

- [20] D. J. Amit, H. Gutfreund, H. Sompolinsky. *Spin glass models of neural networks*. PHYSICAL REVIEW (1985).
- [21] M. Niss. *History of the Lenz-Ising Model 1920–1950: From Ferromagnetic to Cooperative Phenomena*. Arch. Communicated by R. H. Stuewer. HIST. EXACT SCI. 59, 267–318 (2005).
- [22] L. D. Landau, E. M. Lifšits, L. P. Pitaevskij. *Fisica Teorica 5 - Fisica Statistica Parte Prima*. EDITORI RIUNITI (1986).
- [23] Lih-Hsing Hsu, Cheng-Kuan Lin. *Graph Theory and Interconnection Networks*. CRC PRESS, TAYLOR & FRANCIS GROUP (2008).
- [24] G. Mussardo. *Il modello di Ising. Introduzione alla teoria dei campi e delle transizioni di fase*. BOLLATI BORINGHIERI (2007).
- [25] R. H. Landau, M. J. Pàez, C. Bordeianu. *Computational Physics: Problem Solving with Computers*. WILEY-VCH (2007).
- [26] V. Černỳ. *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm*. Communicated by S. E. Dreyfus. JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS (1985).
- [27] C. Blum, A. Roli. *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*. ACM COMPUTING SURVEYS, VOL. 35, No. 3 (2003).
- [28] R. Monasson. *Phase Fransitions in Complex Problems, from Statistical Physics to Optimisation and Algorithms*. OXFORD UNIVERSITY PRESS (2005).
- [29] W. Kinzel, G. Raentz. *Physics by Computer*. SPRINGER (1998).
- [30] S. Istrail. *Universality of Intractability for the Partition Function of the Ising Model Across Non-Planar Lattices*. ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING (2000).
- [Cro87] J. Cronin. *Mathematical Aspects of Hodgkin-Huxley Theory*. CAMBRIDGE UNIVERSITY PRESS (1987).
- [Spr64] D. Sprecher. *On the Structure of Continuous Functions of Several Variables*. TRANSACTIONS OF THE AMERICAN MATHEMATICAL SOCIETY, VOL. 115, PP. 340–355 (1964).
- [Jud90] J. S. Judd. *Neural Network Design and the Complexity of Learning*. MIT PRESS (1990).
- [Nat91] B. Natarajan. *Machine Learning – A Theoretical Approach*. MORGAN KAUFMANN (1991).
- [Hir04] M. W. Hirsch, S. Smale, R. L. Devaney. *Differential Equations, Dynamical Systems and an Introduction to Chaos*. ACADEMIC PRESS/ELSEVIER (2004).
- [Heb49] D. O. Hebb. *The organization of behavior*. NEW YORK, WILEY (1949).
- [Tal02] P. M. Talaván, J. Yáñez. *Parameter setting of the Hopfield network applied to TSP*. NEURAL NETWORKS, VOLUME 15, NUMBER 3, PP. 363-373 (2002).
- [Mul03] S.A. Mulder, D.C.Wunsch. *Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks*. NEURAL NETWORKS, VOLUME 16 , SPECIAL ISSUE: ADVANCES IN NEURAL NETWORKS RESEARCH (2003).
- [App06] D. L. Applegate, R. E. Bixby. *The Traveling Salesman Problem: A Computational Study*. PRINCETON UNIVERSITY PRESS (2006).
- [Geo88] H. O. Georgii. *Gibbs Measures and Phase Transitions*. WALTER DE GRUYTER (1988).

## Bibliografia e Riferimenti

- [Kir83] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. *Optimization by Simulated Annealing*. SCIENCE 220(4598), PP. 671-680 (1983).
- [Glo86] F. Glover. *Future paths for integer programming and links to artificial intelligence*. COMPUT. OPER. RES. 13, 533-549 (1986).
- [Kul51] S. Kullback, R. A. Leibler. *On Information and Sufficiency*. THE ANNALS OF MATHEMATICAL STATISTICS 22 (1): 79-86 (1951).
- [Cas09] M. Cassandro, E. Orlandi, P. Picco. *Phase Transition in the 1d Random Field Ising Model with long range interaction*. Communicated by H. Spohn. COMMUN. MATH. PHYS. 288, 731-744 (2009).
- [Ber93] J. van den Berg. *A Uniqueness Condition for Gibbs Measures, with Application to the 2-Dimensional Ising Antiferromagnet*. COMMUN. MATH. PHYS. 152, 161-166 (1993).
- [Kra41] H. A. Kramers and G. H. Wannier. *Statistics of the two-dimensional ferromagnet*. PHYSICAL REVIEW 60: 252-262 (1941).
- [Wol89] U. Wolff. *Collective Monte Carlo updating for spin systems*. PHYSICAL REVIEW LETTERS 62, 361-4 (1989).
- [Hua87] K. Huang. *Statistical mechanics. Second Edition*. WILEY (1987).
- [Bir78] R. Birgeneau, G. Shirane. *Magnetism in One Dimension*. PHYS. TODAY 31, 32-43 (1978).
- [Dys69] F. J. Dyson. *Existence of a phase transition in a one-dimensional Ising ferromagnet*. COMM. MATH. PHYS, 12, 91-107 (1969).
- [Frö82] J. Fröhlich, T. Spencer. *The phase transition in the one-dimensional Ising model with  $1/r^2$  interaction energy*. COMM. MATH. PHYS. 84, 87-101 (1982).
- [Wan95] F. G. Wang, M. Suzuki, N. Hatano. *Study on Dynamical Critical Exponents of the Ising Model using the Damage Spreading Method*. JOURNAL OF PHYSICS. A, MATHEMATICAL AND GENERAL, VOL.28, NO.16, (1995).
- [Bin01] K. Binder, E. Luijten, M. Müller, N. B. Wilding, H. Blöte. *Monte Carlo investigations of phase transitions: status and perspectives*. PHYSICA A, VOL. 281, 112-128 (2001).