

# Apprendimento di transizioni di fase

Martina CRIPPA, Pietro Francesco FONTANA

18 ottobre 2019

## 1 Introduzione

Nel campo della meccanica statistica un aspetto fondamentale è lo studio delle transizioni di fase, ovvero la trasformazione dello stato di un sistema al mutare di determinate variabili termodinamiche. I sistemi considerati in questo lavoro, descritti dal modello di Ising e dal modello XY, transiscono al variare della temperatura e nel punto di passaggio questa viene detta temperatura di transizione. In generale la transizione di fase è descritta da una funzione caratteristica del sistema, detta parametro d'ordine, ad esempio la magnetizzazione nel modello di Ising. L'obiettivo del progetto è implementare e studiare una rete neurale in grado di apprendere il parametro d'ordine per il modello di Ising e quindi classificare correttamente la fase di diverse configurazioni del sistema, individuando infine la temperatura di transizione. Successivamente si è provato ad eseguire una procedura analoga nel caso del modello XY, un sistema che non presenta una transizione di fase descrivibile attraverso un parametro d'ordine, quindi dove la rete neurale apprende altre caratteristiche del sistema per classificarne la fase. Il lavoro prende spunto da diversi articoli pubblicati negli ultimi due anni che affrontano la stessa tematica [1, 2, 3].

## 2 Sistemi fisici e simulazioni

In questo lavoro sono stati simulati e studiati due sistemi classici della meccanica statistica su reticolo bidimensionale: il modello di Ising e il modello XY.

### 2.1 Modello di Ising

Il modello di Ising descrive il comportamento di spin su reticolo che assumono valori  $\{+1; -1\}$  e interagiscono attraverso un'hamiltoniana la cui forma più semplice, utilizzata in questo lavoro, è

$$H = - \sum_{\langle i \ j \rangle} \sigma_i \sigma_j \quad (1)$$

dove le parentesi angolari indicano l'interazione a primi vicini fra gli spin  $\sigma$ . Quindi viene assunto nullo il campo magnetico esterno e la costante di accoppiamento (solitamente indicata con  $J$ ) assume lo stesso valore fra tutti gli spin, pari a 1. Ad alte temperature il contributo entropico fa sì che il sistema si trovi in uno stato di disordine, ovvero nella fase paramagnetica, illustrata in figura 1a. Abbassando la temperatura prevale l'interazione tra gli spin e al di sotto della temperatura di transizione questi tendono ad allinearsi, quindi il sistema transisce alla fase ferromagnetica, rappresentata in figura 1b. Tale transizione di fase è detta del *secondo ordine* ed è caratterizzata da un comportamento critico, per questo la temperatura di transizione viene anche detta temperatura critica.

Il parametro d'ordine che evidenzia la transizione di fase è la magnetizzazione media per spin del sistema

$$m = \frac{1}{N} \sum_i \sigma_i \quad (2)$$

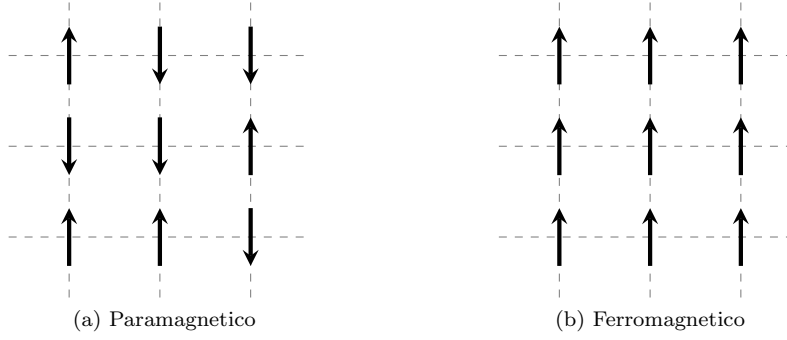


Figura 1: Ising 2D

che risulta vicina a 0 se valutata ad alte temperature, mentre a basse temperature tende a  $+1$  o a  $-1$ . Sono stati studiati sistemi con geometrie di reticolo differenti e conseguentemente aventi un numero di primi vicini (nn) e temperatura critica diversa, i valori sono riportati in tabella 1.

Reticolo	nn	$T_c [J/k_B]$	$T_c \text{ approx } [J/k_B]$	$T_{init} [J/k_B]$
Honeycomb	3	—	1.5187	0.0
Quadrato	4	$2/\ln(1 + \sqrt{2})$	2.2692	1.0
Triangolare	6	$4/\ln 3$	3.6410	2.0

Tabella 1: Temperatura critica per i reticoli studiati. Con  $k_B$  si indica la costante di Boltzmann.

La simulazione dei vari sistemi è stata effettuata con metodi Monte Carlo per 40 diverse temperature distribuite simmetricamente attorno alla temperatura critica, specificando la temperatura iniziale  $T_{init}$  riportata in tabella 1. Il sistema è stato inizializzato a  $T_{init}$ , lasciandolo raggiungere l'equilibrio a temperatura fissata per poi passare alla temperatura successiva senza inizializzare nuovamente il sistema. I modelli di Ising su reticolo quadrato e honeycomb sono stati simulati tramite l'algoritmo di Wolff [4], quindi eseguendo mosse a cluster, mentre il modello su reticolo triangolare è stato simulato con l'algoritmo di Metropolis [5], quindi ad ogni mossa viene provata l'inversione di un singolo spin. Nelle simulazioni con l'algoritmo di Wolff il sistema è inizializzato random, mentre nella simulazione con l'algoritmo di Metropolis il sistema è inizializzato ordinato per facilitare il raggiungimento dell'equilibrio a bassa temperatura. In tutti i sistemi studiati vengono applicate condizioni al contorno periodiche. Le simulazioni sono state scritte in linguaggio C++, il generatore di numeri casuali utilizzato è il Mersenne Twister 19937 [6].

Nel caso del reticolo honeycomb le dimensioni si riferiscono ad un reticolo quadrato su cui si intrecciano due reticoli honeycomb, le interazioni sono progettate in modo che i due reticoli intrecciati si connettano lungo una direzione tramite condizioni al contorno periodiche, riproducendo il comportamento di un reticolo honeycomb unico con una dimensione doppia. L'accuratezza del procedimento viene confermata dai risultati stessi del lavoro.

## 2.2 Modello XY

Il modello XY descrive il comportamento di spin su reticolo in grado di ruotare assumendo valori nell'intervallo  $[0, 2\pi)$ ; gli spin interagiscono a primi vicini tramite l'hamiltoniana

$$H = - \sum_{\langle i, j \rangle} \cos(\theta_i - \theta_j) \quad (3)$$

dove la costante di accoppiamento (solitamente indicata con  $J$ ) è stata fissata pari ad 1 ed è stato posto nullo il campo magnetico esterno. Nel modello XY, secondo il teorema di Mermin-Wagner [7],

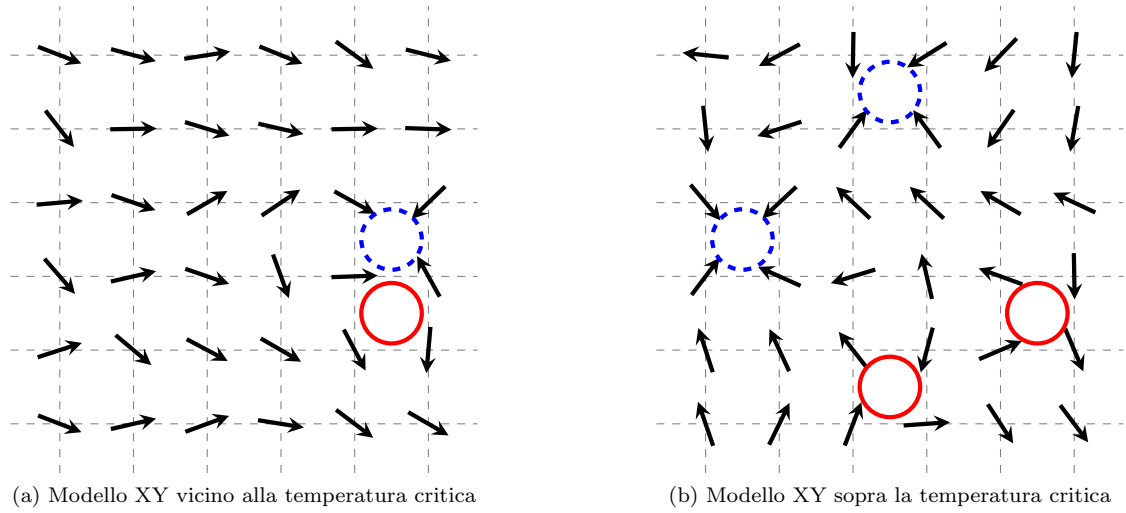


Figura 2: Modello XY a due diverse temperature, i vortici sono indicati in blu, gli antivortici in rosso. Si può notare una chiara coppia vortice-antivortice nella figura di sinistra, mentre nella figura di destra non è presente alcuna coppia.

non è prevista alcuna transizione di fase del secondo ordine. Tuttavia presenta una transizione di fase *topologica*, ovvero la transizione BKT [8], che coinvolge la creazione di vortici e antivortici, configurazioni topologicamente stabili. Appena sotto la temperatura di transizione si osserva la formazione di coppie vortice-antivortice mentre non è energeticamente conveniente la creazione di vortici e antivortici singoli, abbassando la temperatura la maggior parte degli spin tende ad orientarsi in modo parallelo. Al di sopra della temperatura critica vince il contributo entropico e compaiono vortici e antivortici non accoppiati, gli spin in generale non sono più allineati tra loro. Due configurazioni esemplari a temperature diverse sono riportate in Figura 2.

Anche questo sistema è stato simulato tramite metodi Monte Carlo, utilizzando l'algoritmo di Wolff e seguendo una procedura analoga a quella utilizzata per il modello di Ising, i parametri sono riportati in tabella 2. L'inizializzazione degli spin sul reticolo è casuale e non viene ripetuta per ogni passo di temperatura ma solo all'inizio della simulazione, a bassa temperatura.

Reticolo	nn	$T_t$	$T_t$ approx $[J/k_B]$	$T_{init}$ $[J/k_B]$
Quadrato	4	—	0.893	0.01

Tabella 2: Temperatura critica per il reticolo studiato.

### 3 Visualizzazione dei dati

Prima di procedere con l'apprendimento è possibile visualizzare i dati tramite una riduzione bidimensionale delle configurazioni, alcuni esempi per il modello di Ising sono riportati in figura 3. La tecnica utilizzata per ridurre la dimensionalità dei dati è t-SNE [9]. In tutte le figure è possibile osservare la creazione di cluster per i due possibili stati del sistema, al di sopra della temperatura di transizione e al di sotto di essa.

Sono stati generati due grafici per il reticolo triangolare; il primo basato sui dati di una simulazione in cui gli spin vengono inizializzati tutti a 1 quindi in uno stato naturale a bassa temperatura; il secondo proviene da una simulazione con inizializzazione casuale degli spin quindi in uno stato naturale ad alta temperatura. La differenza tra le due figure di sistemi su reticolo triangolare è

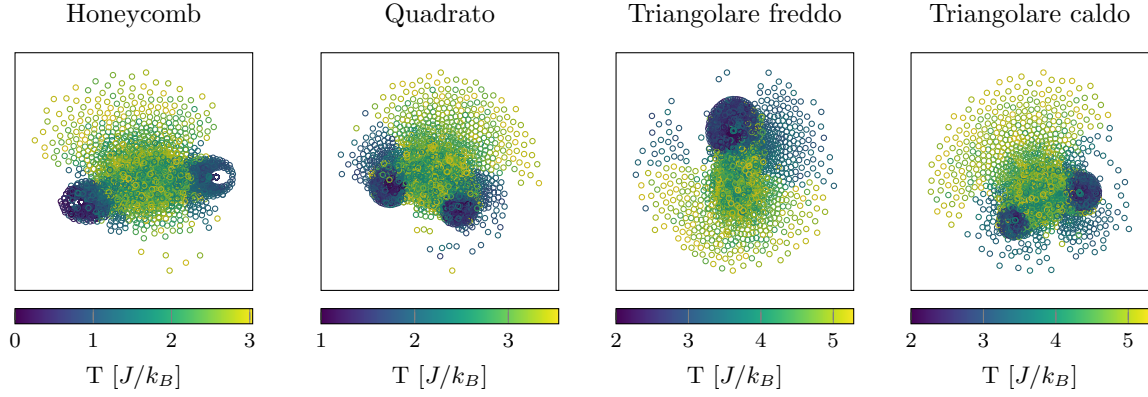


Figura 3: Riduzioni bidimensionali di 5000 configurazioni di modelli di Ising su reticoli di lato 50. Il colore indica la temperatura a cui la configurazione è stata simulata, la temperatura di transizione è sempre al centro della scala.

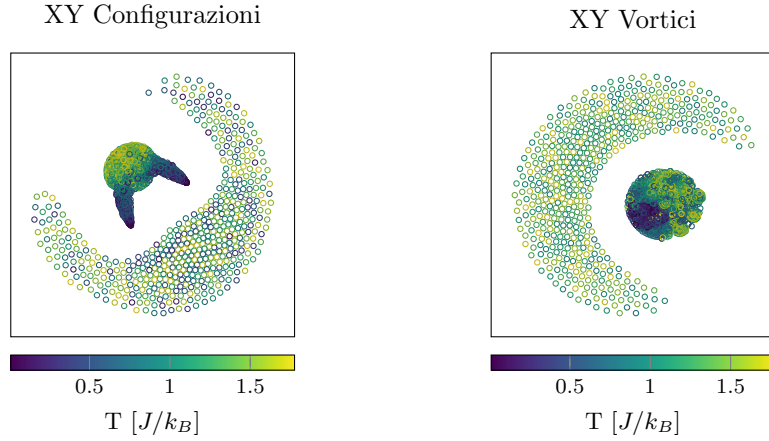


Figura 4: Riduzione bidimensionale di 5000 configurazioni di un modello XY su reticolo di lato 32. Il colore indica la temperatura a cui la configurazione è stata simulata, la temperatura di transizione è sempre al centro della scala.

dovuta all'evoluzione del sistema a basse temperature, quando gli spin vengono inizializzati tutti a 1 è molto improbabile che la magnetizzazione si inverta completamente, mentre partendo da un'inizializzazione casuale il sistema può trovare l'equilibrio con eguale probabilità in una delle due polarizzazioni opposte. Come indicato in sezione 2.1 la simulazione su reticolo triangolare inizia da un sistema completamente ordinato per tutti i dati utilizzati nel resto del lavoro.

La riduzione di dimensionalità è stata applicata anche alle configurazioni di un modello XY su reticolo quadrato, il risultato è rappresentato in figura 4. Fornendo le semplici configurazioni il risultato è poco chiaro, ma fornendo le configurazioni espresse in vortici (vd. sezione 4.2) peggiora ulteriormente e compare un unico grosso cluster; in entrambi i casi molti punti fuoriescono dal cluster principale, mostrando come la classificazione non sia affatto completa. Nel caso della figura a sinistra si possono notare due zone in cui sono concentrate le configurazioni a bassa temperatura, come nel caso di Ising, ma la simulazione non prevede magnetizzazione discreta nel caso di XY e l'inizializzazione del reticolo avviene in modo casuale, le configurazioni a bassa temperatura appaiono quindi come spin equi-orientati ma in direzione diversa per ogni singola simulazione.

In conclusione questa tecnica può fornire un'utile anteprima della struttura dei dati nel caso dei sistemi di Ising, al contrario nel caso dei sistemi XY non permette di trarre conclusioni, confermando

la difficoltà presenti nella classificazione delle configurazioni di questo modello.

## 4 Metodi di apprendimento

A partire dai sistemi fisici simulati, descritti nella sezione precedente, sono stati sviluppati diversi metodi di apprendimento con l'obiettivo di classificare automaticamente la fase in cui si trova il sistema. I metodi utilizzati per l'apprendimento differiscono per i due modelli fisici analizzati, il modello di Ising e il modello XY, seguendo la stessa strada di altri lavori [3].

Le reti sono state implementate tramite Keras [10], utilizzando Tensorflow [11] come backend.

### 4.1 Modello di Ising

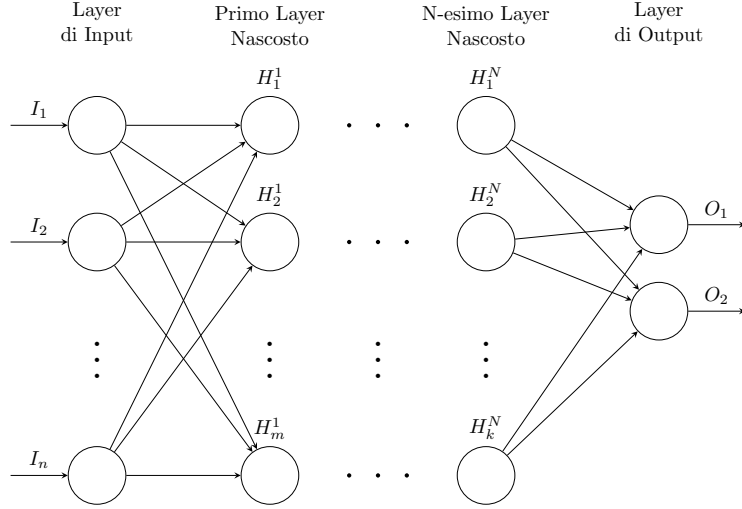


Figura 5: Schema di una rete neurale feed forward a più layer nascosti fully connected.

Per classificare le configurazioni di un modello di Ising è stata utilizzata una rete neurale *feed forward* a uno o più layer nascosti *fully connected*, uno schema generico della rete è rappresentato in Figura 5. Al layer di input vengono fornite le configurazioni degli spin del sistema fisico nella forma di un vettore di elementi  $\{+1; -1\}$ , la dimensione di questo vettore dipende ovviamente dalla dimensione del reticolo di Ising utilizzato, nel caso del presente lavoro spazia da un minimo di 400 elementi (reticolo  $20 \times 20$ ) ad un massimo di 8100 elementi (reticolo  $90 \times 90$ ). Le etichette con cui vengono classificate le configurazioni, e che vengono fornite per l'apprendimento, sono variabili binarie che indicano se il sistema si trova sopra o sotto la temperatura di transizione. In particolare vengono fornite coppie di valori ( $[1,0]$  o  $[0,1]$ ) come etichette, per ottenere due valori in output. Essendo un problema di classificazione binaria è possibile utilizzare uno o due nodi nel layer di output, per una questione di comodità nell'analisi dei risultati fisici è stato utilizzato un layer di output a due nodi. I nodi di output restituiscono la probabilità che la configurazione fisica sia in una fase ferromagnetica o in quella paramagnetica, le due probabilità sommano quindi a 1.

La rete viene sempre allenata su configurazioni del modello di Ising su reticolo quadrato classico, ossia con 4 primi vicini; se riesce ad apprendere correttamente la transizione di fase dovrebbe essere in grado di classificare adeguatamente anche le configurazioni dei modelli a lei sconosciuti, come Ising su reticolo triangolare e honeycomb.

### 4.2 Modello XY

Il metodo migliore per classificare un modello XY risulta essere una rete convoluzionale [2], uno schema di una rete di questo tipo effettivamente utilizzata in questo lavoro è rappresentato in

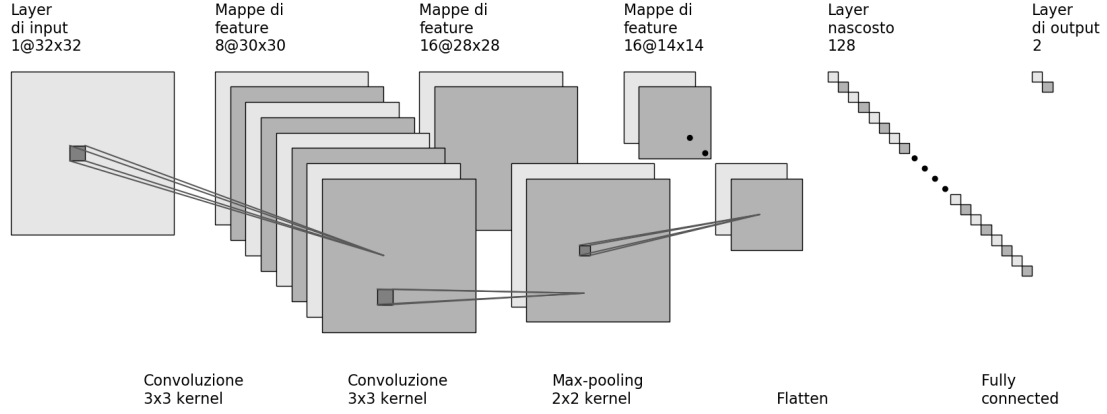


Figura 6: Schema di una rete convoluzionale effettivamente utilizzata per l'analisi di un sistema XY su reticolo  $32 \times 32$ , sono stati tralasciati i layer *dropout*. Immagine generata adattando il codice disponibile presso [https://github.com/gwding/draw\\_convnet](https://github.com/gwding/draw_convnet).

Figura 6. I primi due layer, detti convoluzionali, sono fondamentali per estrarre le caratteristiche (*feature*) che permettono la classificazione delle configurazioni. La struttura della rete è quella tipica utilizzata per la classificazione di immagini, per esempio il riconoscimento di caratteri scritti a mano del database MNIST.

Il layer di input può ricevere come argomento la semplice configurazione degli spin, quindi una matrice di valori reali nell'intervallo  $[0, 2\pi)$ , oppure la configurazione espressa in vortici e antivortici, che appare come una matrice con valori  $\{-1; 0; +1\}$  che rappresentano rispettivamente la presenza di un antivortice, l'assenza di strutture e la presenza di un vortice. Come nel caso di Ising le etichette indicano se la temperatura del sistema si trova sopra o sotto la temperatura di transizione e il layer di output ha due nodi, che rappresentano le probabilità complementari di essere in una delle due fasi.

## 5 Impostazione dell'apprendimento

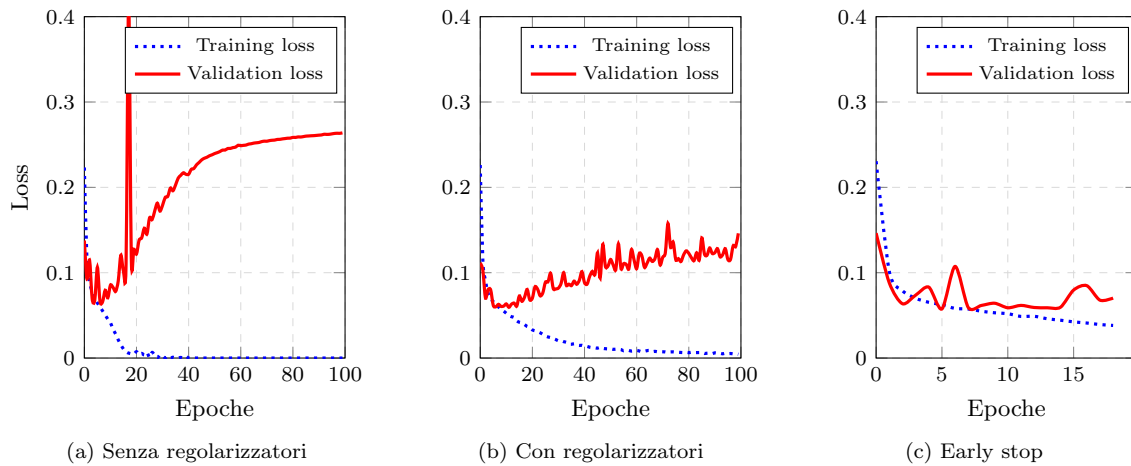


Figura 7: Risultati dell'apprendimento di una rete convoluzionale, del tipo rappresentato in figura 6. I dati utilizzati sono relativi ad un modello XY su reticolo quadrato di lato 32.

Il primo passo nel lavoro di creazione della rete è stato valutare se questa potesse apprendere efficacemente da un insieme di dati sufficientemente vasto. Dopo aver generato circa 100 000 configurazioni per l'apprendimento queste sono state divise in *training set* e *validation set* in una percentuale rispettivamente dell'80% e del 20%. Attraverso il *validation set* è possibile stimare le performance della rete con l'avanzare del processo di apprendimento e capire se sta avvenendo un *overfit*.

In figura 7 è possibile confrontare tre processi di apprendimento della rete convoluzionale utilizzata per la classificazione dei sistemi XY, vengono rappresentate la *training loss* e la *validation loss*, quando la seconda inizia a crescere si sta verificando un *overfit*, la rete sta imparando caratteristiche specifiche del *training set* e non generalizzabili ai dati sconosciuti del *validation set*. Nel caso della figura a sinistra non è stato utilizzato alcuno strumento di regolarizzazione; si vede immediatamente come la rete inizia un *overfit* già dopo la decima epoca. Per la figura al centro sono stati aggiunti due layer *dropout*, uno prima ed uno dopo il layer nascosto da 128 neuroni, inoltre viene imposto un valore massimo pari a 5 alla norma dei pesi di questo layer nascosto; in questo caso la rete ha maggiori difficoltà a specializzarsi sui dati del *training set*. Infine nella figura di destra viene applicato un *early stop*, ossia l'apprendimento viene bloccato in automatico quando una certa quantità (in questo caso la *validation accuracy*) non cresce significativamente per diverse epoche; si osserva nell'immagine che il processo viene efficacemente interrotto quando la *validation loss* non ha ancora iniziato a crescere. Nell'apprendimento di sistemi di Ising il fenomeno di *overfit* è più contenuto, ma ai layer nascosti è stato ugualmente aggiunto un regolarizzatore L2, che limita la libertà di evoluzione dei pesi della rete impedendo la specializzazione sul *training set*.

## 5.1 Statistica sui dati

Per migliorare l'affidabilità delle variabili misurate, come la test accuracy, sono stati generati test set da 100 000 elementi che vengono divisi equamente in 10 insiemi su cui ogni modello viene valutato. Allo stesso modo, per tenere conto della componente stocastica presente nel processo di apprendimento di una rete, sono stati generati 10 modelli per ciascuna rete sullo stesso training set. Questo procedimento genera quindi 10 misure per ciascuno dei 10 modelli, per un totale di 100 misure, da cui vengono estratti la media e l'errore, come deviazione standard della media. Questo sarà lo standard utilizzato in tutte le analisi successive.

Si osserva che le misure effettuate non sono indipendenti tra loro, l'intero insieme di 100 misure condivide il medesimo training set e a gruppi di 10 condividono lo stesso modello, quindi non è possibile applicare né i teoremi del limite centrale né la legge dei grandi numeri. Le medie e le deviazioni standard delle medie vengono ugualmente utilizzate nel lavoro perché sono lo strumento più semplice a disposizione. Si potrebbero ottenere misure indipendenti partendo da un training set diverso per ogni misura, ma questa procedura richiede un impegno computazionale non praticabile.

## 6 Modello di Ising: risultati e conclusioni

### 6.1 Valutazione della geometria della rete

Per meglio valutare il contributo dei diversi parametri della rete sono stati generati più modelli, eseguendo la fase di apprendimento sullo stesso training set e variando di volta in volta alcune caratteristiche quali il numero di neuroni in un layer o il numero di layer. In questa sezione le principali caratteristiche della rete che non vengono variate sono riportate in tabella 3.

#### 6.1.1 Diverso numero di neuroni

La prima analisi è stata effettuata allenando diverse reti a singolo layer nascosto con numero di neuroni variabile, i valori esplorati sono: 3, 10, 50, 100, 200 e 1000. L'analisi è stata limitata a modelli di Ising su reticoli di lato 30, 50 e 80. Nel grafico in figura 8 sono riportati i valori di test accuracy al variare del numero di neuroni, rappresentati in scala logaritmica.

Si osserva che per il reticolo di lato 30 una rete a 10 neuroni è sufficiente per classificare in maniera adeguata la fase del sistema, raggiungendo una test accuracy circa del 98%. Per reticoli di maggiori

Batch size:	100
Pesi $w$ e $b$ iniziali:	Gauss( $\nu = 0, \sigma = 1$ )
Fz. attivazione:	Sigmoid
Earlystop:	validation accuracy
Regolarizzazione:	L2 0.01
Ottimizzazione:	Adam 0.0001

Tabella 3: Caratteristiche della rete fissate.

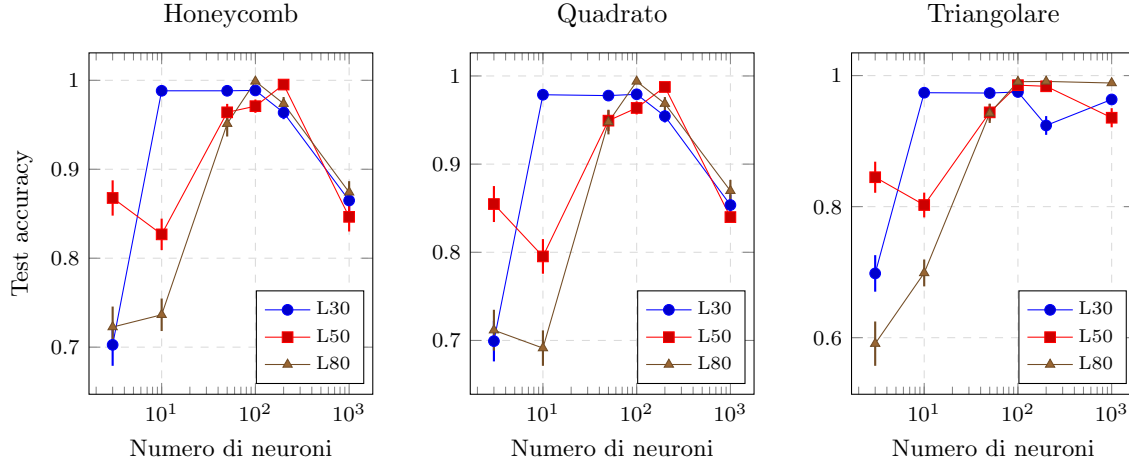


Figura 8: Performance della rete feed forward a singolo layer fully connected sul modello di Ising.

dimensioni sono necessari almeno 50 neuroni per raggiungere livelli di test accuracy superiori al 90%. Per tutte le dimensioni considerate aumentando il numero di neuroni a 1000 si osserva un crollo nelle performance per i reticoli honeycomb e quadrato, mentre per il reticolo triangolare la rete continua a classificare in modo soddisfacente. Per limiti computazionali non sono state realizzate reti con un numero di neuroni superiore, quindi non è stato possibile chiarire l'andamento complessivo per il reticolo triangolare. Si osserva che una differenza tra la configurazione su reticolo triangolare e gli altri due sistemi, oltre alla geometria del reticolo, è l'algoritmo utilizzato per la simulazione del sistema, come riportato in sezione 2.1.

### 6.1.2 Diverso numero di layer

Il primo grafico riportato in figura 9 presenta l'andamento del numero di parametri normalizzato sul numero di training data rispetto al numero di neuroni per reti a un layer nascosto. Come si può notare il numero di parametri segue un andamento lineare e cresce velocemente. Diversamente, nel secondo grafico in figura 9 si osserva che aumentando il numero di layer il numero di parametri cresce sempre linearmente ma in maniera molto meno marcata, la crescita è rappresentata rispetto al numero di layer nascosti che è proporzionale al numero totale di neuroni. Le caratteristiche delle reti a più layer sono specificate successivamente.

Nell'ambito delle reti neurali si cerca di avere sempre un equilibrio tra il numero di parametri da apprendere ed il numero di dati utilizzati per l'apprendimento, preferendo se possibile una prevalenza della seconda quantità. Si può trovare un parallelo con il numero di vincoli necessari per ottenere una soluzione unica in un sistema a molte variabili. Un metodo per diminuire il numero di parametri totali è aumentare il numero di layer nascosti diminuendo i neuroni presenti su ciascun layer.

È stata effettuata quindi un'analisi allenando reti a più layer nascosti, pari a 1, 2, 3, 5 e 10, con ciascun layer popolato da 10 neuroni. I risultati, riportati nei grafici di 10, mostrano che si ottiene



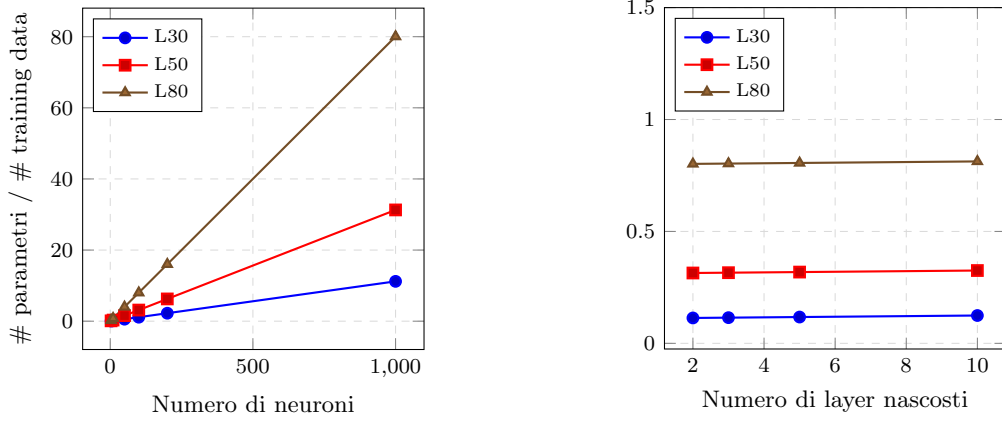


Figura 9: Rapporto tra il numero di parametri della rete e il numero di elementi del training set utilizzato per l'apprendimento.

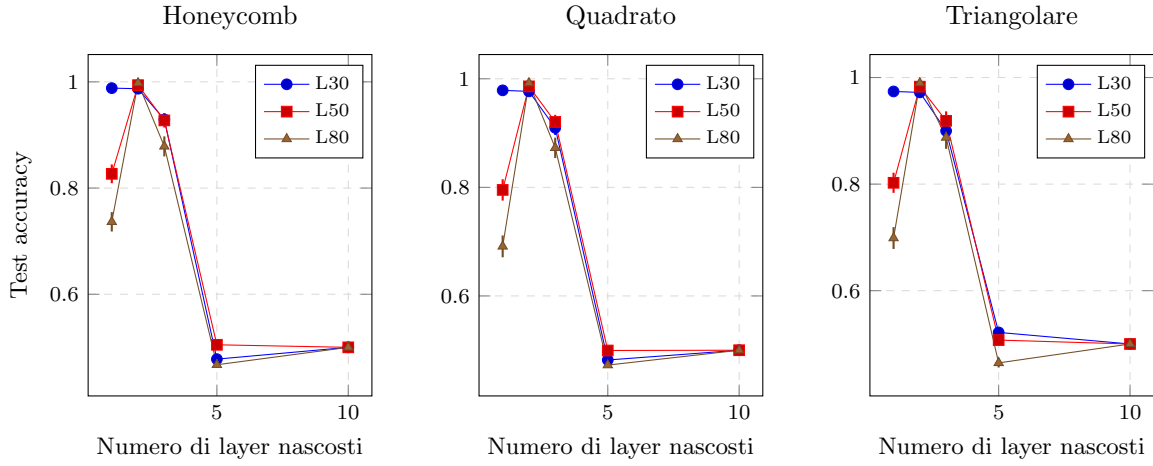


Figura 10: Performance della rete feed forward a layer multipli fully connected da dieci neuroni ciascuno sul modello di Ising.

un valore ottimale di test accuracy per tutte le tipologie di reticolo classificando con reti a 2 layer. Aggiungendo un layer le performance si abbassano notevolmente, mentre per i reticoli di lato 30 si ottengono buoni risultati anche per reti aventi un singolo layer. Inoltre dai risultati di questa analisi si può osservare come le performance sul modello di Ising con reticolo triangolare siano consistenti con quelle degli altri modelli, al contrario di quanto riportato nella sezione precedente. In definitiva, confrontando i valori in tabella 4, si può notare come l'apprendimento sul modello di Ising non sembri trarre vantaggio dall'utilizzo di reti a più layer. Nel presente lavoro la sovrabbondanza di parametri nella rete rispetto al numero di dati non sembra penalizzare l'apprendimento della rete stessa.

### 6.1.3 Due layer e diverso numero di neuroni

I risultati delle analisi descritte nei capitoli precedenti, riportati in tabella 4, mostrano che per tutti modelli considerati si ottengono valori di test accuracy massimi per reti a singolo layer con 100 neuroni e per reti a 2 layer con 10 neuroni ciascuno. Prendendo spunto da questi risultati sono stati generati modelli di rete a 2 layer con un numero totale di neuroni pari a 100, variando il numero di neuroni dei singoli layer. È stata allenata una rete a 30 e 70 neuroni, quindi fornendo ad entrambi i layer un numero consistente di neuroni. Come

Rete	L30 TA	L50 TA	L80 TA
1x100	0.981(4)	0.973(6)	0.994(2)
2x10	0.979(4)	0.988(3)	0.993(2)

Tabella 4: Test accuracy valutata su una rete a singolo layer da 100 neuroni e su una rete a 2 layer da 10 neuroni ciascuno. I valori corrispondono alla media sulle tre diverse tipologie di reticolo per ciascuna dimensione.

si può osservare dai risultati riportati in tabella 5, questi sono compatibili entro l'errore con quelli in tabella 4, a parte la test accuracy per la rete 1x100 sul reticolo L50 che si discosta di circa  $2\sigma$  dagli altri valori. Anche invertendo i layer da 30 e 70 neuroni si ottengono valori di test accuracy equivalenti, considerando l'errore.

Rete	L30 TA	L50 TA	L80 TA
1x70+1x30	0.979(4)	0.988(3)	0.993(2)
1x30+1x70	0.978(5)	0.988(3)	0.993(2)

Tabella 5: Test accuracy (TA) valutata su una rete a 2 layer da 70 e 30 neuroni e su una rete a 2 layer da 30 e 70 neuroni. I valori corrispondono alla media sulle tre diverse tipologie di reticolo per ciascuna dimensione.

Quindi, nel caso di una rete a due layer entrambi con un numero consistente di neuroni, l'inversione dei layer non sembra aver modificato le sue capacità di apprendimento. Si prosegue allenando una rete con un numero di neuroni per layer meno equilibrato: 95 e 5.

Rete	L30 TA	L50 TA	L80 TA
1x95+1x5	0.979(4)	0.988(3)	0.994(2)
1x5+1x95	0.978(4)	0.987(3)	0.992(3)

Tabella 6: Test accuracy (TA) valutata su una rete a 2 layer da 95 e 5 neuroni e su una rete a 2 layer da 5 e 95 neuroni. I valori corrispondono alla media sulle tre diverse tipologie di reticolo per ciascuna dimensione.

Anche in questo caso i risultati riportati in tabella 6 sono assolutamente compatibili con i risultati in tabella 5, quindi anche una rete a due layer con un numero di neuroni sbilanciato non risulta essere sensibile all'inversione di layer.

Provando a sbilanciare ulteriormente il numero di neuroni dei due layer, ovvero allenando una rete con un layer da 98 neuroni e un layer da 2 neuroni, si è ottenuto un risultato interessante: la rete con il primo layer da 98 neuroni e il secondo da 2 classifica correttamente la fase del sistema con una test accuracy alta, mentre invertendo i layer si ottengono risultati di test accuracy molto bassi. Questo è un segnale del fatto che 2 neuroni non sempre sono sufficienti per passare le informazioni necessarie per la classificazione al layer successivo, viene persa parte dell'informazione. Si è osservato che 5 neuroni nel primo layer sono sufficienti per classificare correttamente, per trovare il numero minimo di neuroni per una buona classificazione si è provato ad allenare reti 97-3, 96-4 e le relative reti inverse ottenendo i risultati riportati in tabella 7.

In figura 11 sono riportati i grafici relativi a reti a due layer con il primo layer di piccole dimensioni. Si osserva che, diversamente dalle reti a singolo layer trattate in sezione 6.1.1, sono sufficienti 3 neuroni per ottenere una test accuracy superiore al 90% con reticoli di lato 50 e 80, mentre per i reticoli di lato 30 si raggiunge una test accuracy (in media sulla tipologia di reticolo, vd tabella

Rete	L30 TA	L50 TA	L80 TA
1x98+1x2	0.977(5)	0.940(3)	0.993(2)
1x2+1x98	0.72(2)	0.67(2)	0.68(2)
1x97+1x3	0.979(4)	0.989(3)	0.994(3)
1x3+1x97	0.852(6)	0.920(4)	0.906(1)
1x96+1x4	0.979(4)	0.989(3)	0.994(2)
1x4+1x96	0.946(4)	0.970(3)	0.992(3)

Tabella 7: Test accuracy (TA) per diverse reti. I valori sono le medie dei risultati ottenuti per il reticolo honeycomb, quadrato e triangolare.

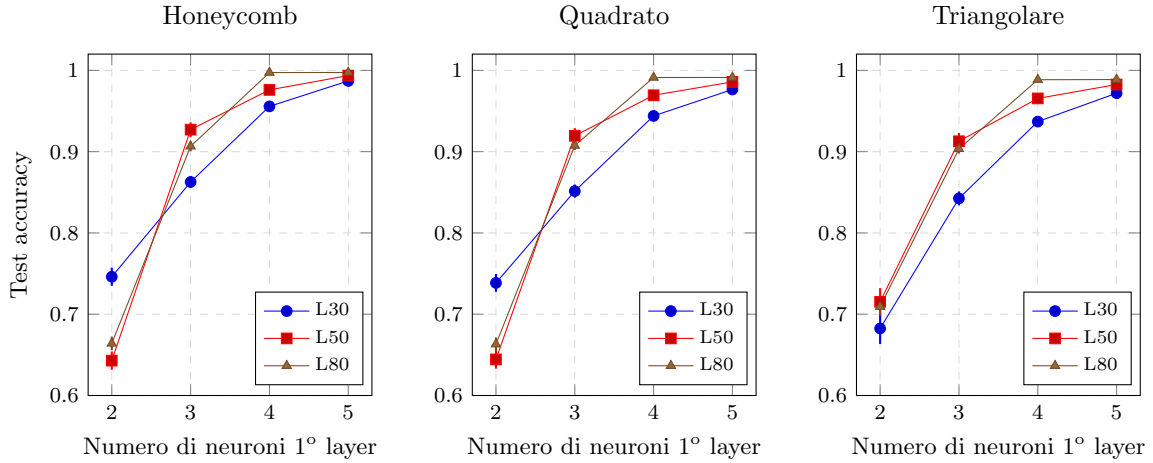


Figura 11: Performance della rete a 2 layer nascosti con numero di neuroni totali pari a 100.

7) superiore all'85%. Questo sottolinea come il secondo layer abbia un ruolo non indifferente nella capacità di apprendimento della rete.

## 6.2 Valutazione dei parametri della rete

Dopo aver esplorato la geometria della rete, si è provato ad allenare reti variando altri tre parametri: la funzione di attivazione, l'ottimizzatore e la batch size.

### 6.2.1 Funzione di attivazione: Sigmoid vs ReLu vs Tanh

Le tre funzioni riportate in figura 12 sono le più comuni funzioni di attivazione utilizzate nelle reti neurali: le reti analizzate in precedenza sono state implementate con la funzione di attivazione Sigmoid, utilizzando la medesima funzione di attivazione per tutti i layer nascosti.

In questa sezione sono state testate reti a singolo layer con 100 neuroni utilizzando queste tre funzioni di attivazione. Si riportano i risultati per la rete 1x100 in tabella 8, mediando sulle tipologie di reticolo. Come si può osservare, per le reti a singolo layer i valori di test accuracy per Sigmoid e Tanh distano al più  $1.5\sigma$ , mentre la rete implementata con ReLu raggiunge prestazioni nettamente inferiori.

Si è quindi variata la geometria della rete, studiando una rete profonda a 5 layer da 10 neuroni per layer, implementando sia la funzione di attivazione Sigmoid che Relu. I risultati sono riportati in tabella 9. Utilizzando la funzione di attivazione ReLu si osserva un notevole risultato, le sue prestazioni aumentano rispetto alla rete implementata con Sigmoid. Il risultato giustifica, per questo modello, le scarse prestazioni ottenute attraverso le reti implementate nella sezione 6.1.2. La funzione

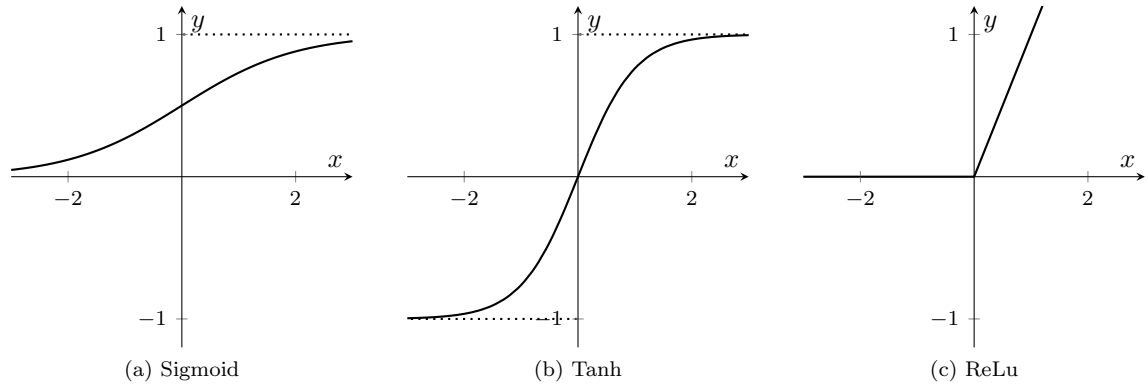


Figura 12: Funzioni di attivazione studiate.

Rete	L30 TA	L50 TA	L80 TA
Sigmoid	0.981(4)	0.973(6)	0.994(2)
Tanh	0.981(4)	0.990(3)	0.994(2)
ReLu	0.833(4)	0.91(1)	0.84(3)

Tabella 8: Test accuracy valutata su reti di lato 30, 50, 80, per rete a singolo layer con 100 neuroni. Ogni valore si riferisce alla media tra i reticoli honeycomb, quadrato e triangolare.

Rete	L30 TA	L50 TA	L80 TA
Sigmoid	0.49(1)	0.504(2)	0.468(2)
ReLu	0.964(6)	0.982(2)	0.877(7)

Tabella 9: Test accuracy valutata su reti di lato 30, 50, 80, per rete da 5 layer a 10 neuroni per layer. Ogni valore si riferisce alla media tra i reticoli honeycomb, quadrato e triangolare.

di attivazione ReLu sembra garantire un'elevata capacità di classificazione per reti a più layer, al contrario della funzione di attivazione Sigmoid che pare diminuire notevolmente le performance.

### 6.2.2 Ottimizzazione: Adam vs SGD

Rete L30	lr	hc TA	sq TA	tr TA
Adam	0.0001	0.9886(1)	0.9792(1)	0.9750(2)
SGD	0.001	0.9875(1)	0.9776(1)	0.9728(2)

Tabella 10: Test accuracy valutata su reti implementate sia con ottimizzazione Adam che SGD; per reticolo di lato 30, honeycomb, quadrato e triangolare.

Tutte le reti analizzate in precedenza sono state implementate attraverso l'ottimizzatore Adam, con un *learning rate* di 0.0001. Si è provato a variare l'ottimizzatore impostando SGD, ovvero *stochastic gradient descend*, con il medesimo learning rate per una rete a singolo layer con 100 neuroni, implementata con l'attivatore Sigmoid e su reticolo di lato 30.

Tuttavia il sistema per convergere deve effettuare più di 500 epoche, risultando impossibile allenare 10 modelli differenti entro limiti di tempo convenienti. Quindi si è provato ad alzare il learning rate

a 0.001, ottenendo risultati comunque soddisfacenti, riportati in tabella 10. Come si può osservare tutti i valori relativi alla rete ottimizzata con Adam distano più di  $5\sigma$  rispetto ai valori ottenuti con SGD, risulta quindi sconsigliato utilizzare quest'ultimo.

### 6.2.3 Diversa batch size

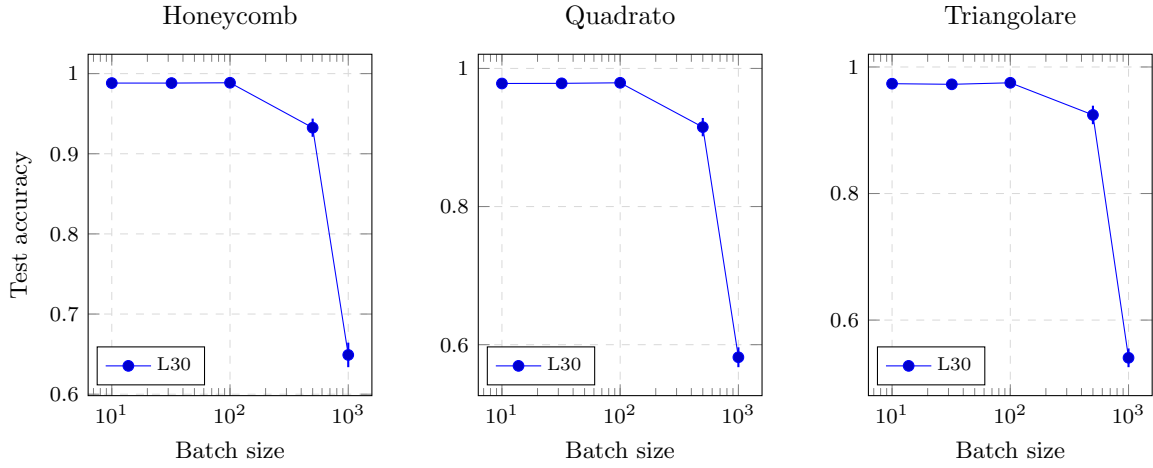


Figura 13: Andamento della test accuracy per diversa batch size. I dati provengono da un sistema di Ising di lato 30.

Infine si è provato a variare la dimensione dei batch, ossia il numero di dati utilizzati per un singolo aggiornamento del gradiente. Come riportato in figura 13 per batch tra i 10 e i 100 elementi si ottiene una buona test accuracy, aumentandone ulteriormente la dimensione la test accuracy diminuisce. Dimensioni minori di 10 elementi non sono convenienti da un punto di vista computazionale. L'analisi è stata effettuata su reticolo di lato 30, allenando una rete a singolo layer con 100 neuroni, funzione di attivazione Sigmoid e ottimizzatore Adam.

## 6.3 Apprendimento e magnetizzazione

Come è stato mostrato nella sezione 6.1.3 esiste una soglia nel numero di neuroni del primo layer al di sotto della quale la rete fa fatica ad apprendere correttamente la fase del sistema con una buona test accuracy. In quella sezione le reti sono state allenate senza scarti, tenendo conto della media delle performance, mentre in questa sezione le reti sono state allenate fino ad ottenere un unico buon risultato.

Seguendo il procedimento in [1] è stato rappresentato l'apprendimento dei pesi della rete in figura 14, si può notare come la funzione ottenuta dall'applicazione dei pesi sia lineare nella magnetizzazione, che è il parametro d'ordine per la transizione di fase del modello di Ising. La rete sembra quindi aver appreso il parametro d'ordine di questa transizione di fase.

I grafici in figura 14 si possono facilmente interpretare, ad esempio nel caso della rete a 2 neuroni si può osservare come uno dei due riceva un input maggiore di zero per magnetizzazione positiva, l'altro per magnetizzazione opposta. I neuroni hanno Sigmoid come funzione di attivazione (vd. figura 12). Nel caso in cui il sistema di spin sia ordinato, quindi con magnetizzazione non nulla, uno dei due neuroni avrà output vicino a 1 e l'altro avrà output vicino a 0; nel caso in cui la configurazione degli spin sia disordinata, quindi con magnetizzazione nulla, entrambi i neuroni avranno output vicino a 0. La rete può riconoscere facilmente in quale delle due fasi si trova il sistema.

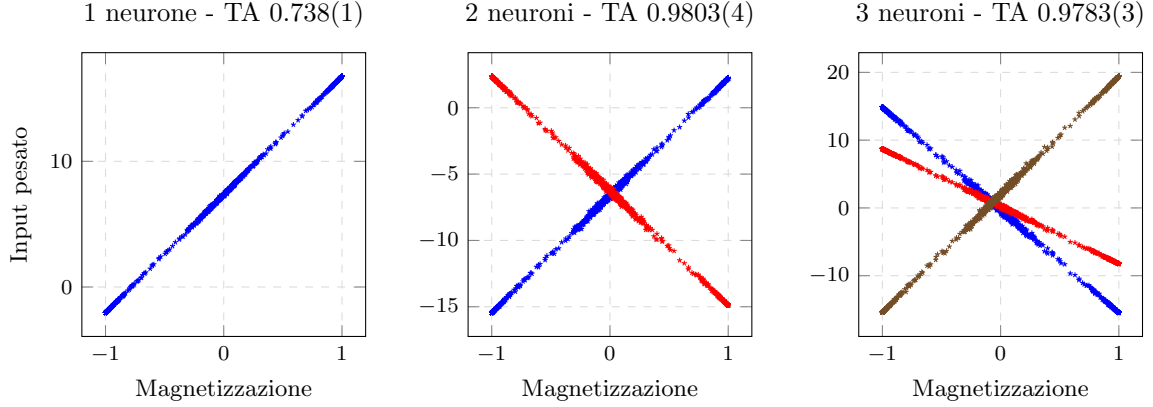


Figura 14: Valutazione degli input pesati in ingresso all'unico layer nascosto per diverse reti, sono rappresentati solo 1000 elementi del test set, in alto viene indicato il numero di neuroni e la test accuracy. L'apprendimento è stato eseguito su sistemi di Ising di lato 30, i modelli sono stati valutati su sistemi a reticolo quadrato. Sull'asse delle ascisse è posta la magnetizzazione media per spin, valutata tramite la formula 2. Sull'asse delle ordinate è posto l'input pesato tramite la formula  $I_w = Wx + b$ , dove  $W$  è il vettore dei pesi,  $b$  è il bias applicato e  $x$  è il vettore di input ossia una configurazione del sistema di spin.

## 6.4 Stima della temperatura critica

Attraverso l'utilizzo delle reti neurali è stato possibile stimare la temperatura critica  $T_c$  del modello di Ising. Il valore teorico, riportato in tabella 1, è esatto per sistemi su reticolo di lato infinito; per sistemi di lato finito la temperatura critica presenta una correzione che aumenta al diminuire della dimensione del reticolo [12]. Per stimare il valore esatto per dimensione infinità è stato effettuato il calcolo della temperatura critica su diversi reticoli al variare del lato da 20 a 90, con passo 10, andando poi ad interpolare i risultati e ottenendo il valore per  $L \rightarrow \infty$  come ordinata all'origine della retta che interpola i dati. Un esempio di misura della temperatura critica è presentato in figura 15, il punto in cui le due curve si incontrano è preso come stima della temperatura critica per quel set di dati.

Layer:	1
Neuroni:	100
Batch size:	100
Pesi $w$ e $b$ iniziali:	Gauss( $\nu = 0$ , $\sigma = 1$ )
Fz. attivazione:	Sigmoid
Earlystop:	validation accuracy
Regolarizzazione:	L2 0.01
Ottimizzazione:	Adam 0.0001

Tabella 11: Parametri

La rete è stata allenata su set di almeno 100 000 configurazioni (prendendo un sottoinsieme di configurazioni pari al 20% del training set per la validazione) a reticolo quadrato, e valutata su set di 100 000 configurazioni per tutte e tre le geometrie di reticolo. Ciascun valore di temperatura riportato è dato dalla media di 100 temperature, i cui valori sono classificati da 10 diverse reti su 10 diversi test set, come spiegato nella sezione 5.1. Le reti utilizzate per questo scopo sono state scelte singolarmente tra quelle con test accuracy alta e test loss bassa. I parametri comuni alle reti sono riportati in tabella 11.

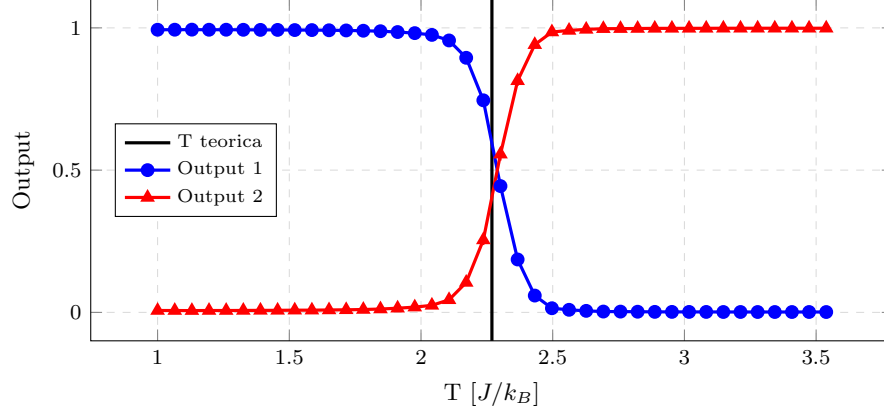


Figura 15: Output dei due neuroni presenti nel layer di output, la rete è valutata su configurazioni di un sistema di Ising di lato 30. La temperatura critica per il modello di Ising si trova in tabella 1. I risultati sono ottenuti dalla media su 250 configurazioni per ciascuna temperatura, per un totale di 10 000 configurazioni.

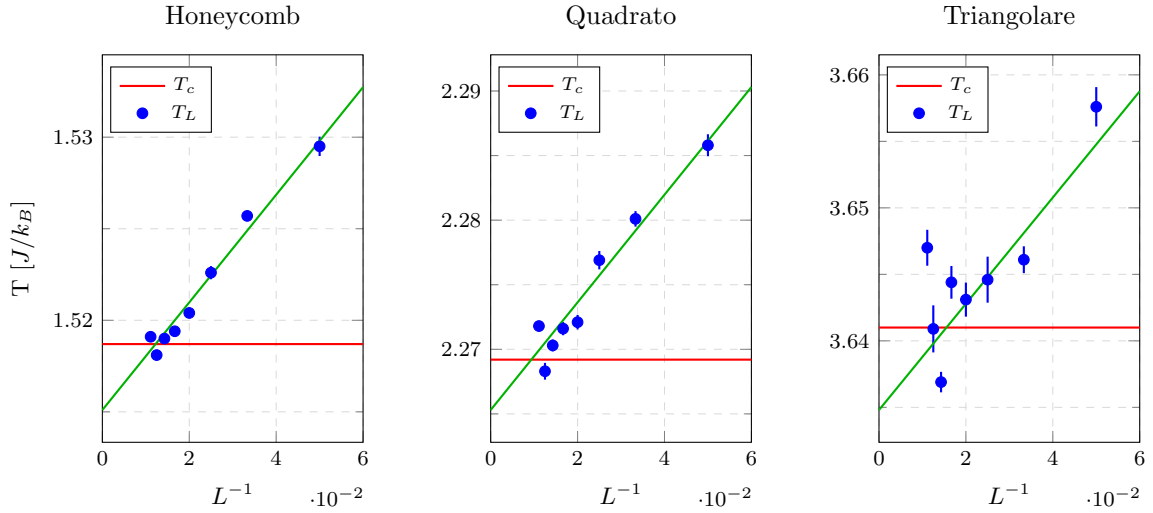


Figura 16: Interpolazione temperatura

Reticolo	T estratta [ $J/k_B$ ]	T teorica [ $J/k_B$ ]
Honeycomb	1.52(4)	1.519
Quadrato	2.27(7)	2.269
Triangolare	3.63(16)	3.641

Tabella 12: Valori ottenuti tramite il fit per la temperatura di transizione. I valori teorici sono riportati con maggiore precisione in tabella 1.

Nei grafici in figura 16 sono riportate le medie delle temperature  $T_L$  al variare del lato del reticolo, con le relative barre d'errore date dalla deviazione standard della media; inoltre è riportato il valore teorico della temperatura di transizione  $T_c$  e la retta che interpola le temperature estratte, il fit è stato eseguito con il metodo dei minimi quadrati. In tabella 12 sono riportati i valori finali ottenuti per la temperatura di transizione, come si può osservare l'errore relativo varia da un minimo del 2.5% per il reticolo honeycomb ad un massimo del 4.5% per il reticolo triangolare.

## 7 Modello XY: risultati e conclusioni

Contrariamente al modello di Ising, per sistemi XY non è noto un parametro d'ordine che manifesti la transizione di fase, quindi non si può stabilire con certezza cosa apprenda la rete. La letteratura [2] suggerisce che per sistemi finito dimensionali, come quelli studiati in questo lavoro, il sistema sia parzialmente magnetizzato e quindi si suppone che la rete allenata sulle configurazioni di spin possa apprendere questa leggera magnetizzazione dovuta alla non idealità del sistema. Perciò, come riportato nell'introduzione, si è provato ad allenare la rete anche sulle configurazioni di vortici ottenendo risultati simili a quelli relativi alle configurazioni. Tuttavia, sempre come osservato nell'articolo [2], non è chiaro cosa la rete riesca ad apprendere e soprattutto se l'apprendimento sia reso possibile unicamente dalla non idealità del sistema.

In questo lavoro viene effettuata solo una breve analisi sul modello XY, giustificata principalmente dall'interesse fisico, approfondimenti significativi sulla parte di apprendimento richiederebbero risorse computazionali di gran lunga maggiori. La struttura di rete da cui si è partiti corrisponde a quella rappresentata in figura 6.

### 7.1 Breve valutazione dei parametri della rete

Analogamente a quanto svolto per il modello di Ising si è provato a variare alcuni parametri significativi della rete convoluzionale utilizzata per il modello XY, testandola sia su configurazioni di dati non elaborati (quindi su valori in  $[0, 2\pi)$ ) che sulle configurazioni di vortici (quindi su valori in  $\{1; 0; -1\}$ ), sempre per un reticolo a quattro primi vicini di lato 32. Innanzi tutto si è provato a variare il kernel del primo layer convoluzionale, allenando e testando la rete per un kernel  $2 \times 2$  e  $3 \times 3$ .

Kernel 1° layer	Configurazioni	Vortici
$2 \times 2$	0.9738(2)	0.9743(2)
$3 \times 3$	0.9729(2)	0.9743(2)

Tabella 13: Confronto tra reti con diversa dimensione del kernel nel primo layer convoluzionale, per il resto la struttura delle reti è identica a quella in figura 6.

Come si può osservare dai risultati riportati in tabella 13, la test accuracy per le configurazioni classificate con kernel  $2 \times 2$  risulta essere superiore di  $3\sigma$  rispetto alle configurazioni classificate con la rete avente kernel  $3 \times 3$ . Al contrario per i vortici si ottiene esattamente lo stesso valore. La differenza di performance nel caso delle semplici configurazioni di spin si può imputare alla struttura delle caratteristiche che la rete deve apprendere, i vortici e gli antivortici possono infatti essere identificati su sottomatrici di lato 2, come mostrato in figura 2.

Successivamente si è variata la geometria del layer nascosto a 128 neuroni, allenando una rete a due layer da 16 neuroni ciascuno. L'analisi è stata effettuata sia sul set di configurazioni che sul set di vortici, per una rete con un kernel  $2 \times 2$  per il primo layer convoluzionale. In tabella 14 si confrontano i risultati dell'analisi.

NN Layer Dense	Configurazioni	Vortici
128	0.9738(2)	0.9743(2)
16+16	0.9722(3)	0.9740(2)

Tabella 14: Confronto fra le test accuracy di una rete convoluzionale con un singolo layer nascosto da 128 neuroni e una con due layer da 16 neuroni ciascuno.

In entrambi i casi risultano superiori le prestazioni per la rete con singolo layer da 128 neuroni, sebbene nel caso dei vortici la test accuracy sia superiore di appena  $1\sigma$ .



## 7.2 CNN vs FFNN

Per verificare gli effettivi vantaggi di una rete convoluzionale nel caso in cui non si forniscano alla rete le caratteristiche da apprendere, si è provato a confrontare le prestazioni in classificazione fra una rete convoluzionale e una rete feed forward a singolo layer. L'analisi è stata effettuata sempre per il reticolo di lato 32, sia su configurazioni di vortici che di spin. La rete utilizzata è a singolo layer da 128 neuroni, con attivatore Sigmoid, ottimizzatore Adam e i vari metodi descritti in precedenza per evitare l'overfitting.

Rete	Configurazioni	Vortici
FFNN	0.9116(4)	0.9610(2)
CNN	0.9738(2)	0.9743(2)

Tabella 15: Confronto tra una rete feed forward a singolo layer nascosto da 128 neuroni e la rete convoluzionale più performante (vd. tabella 13).

Come si può osservare dai risultati riportati in tabella 15, la rete convoluzionale risulta essere nettamente più performante, giustificando le scelte effettuate in precedenza. Infatti una rete priva dei layer convoluzionali fatica ad apprendere le caratteristiche del sistema, in particolare nel caso in cui vengono fornite in input le sole configurazioni degli spin.

## 7.3 Stima della temperatura di transizione

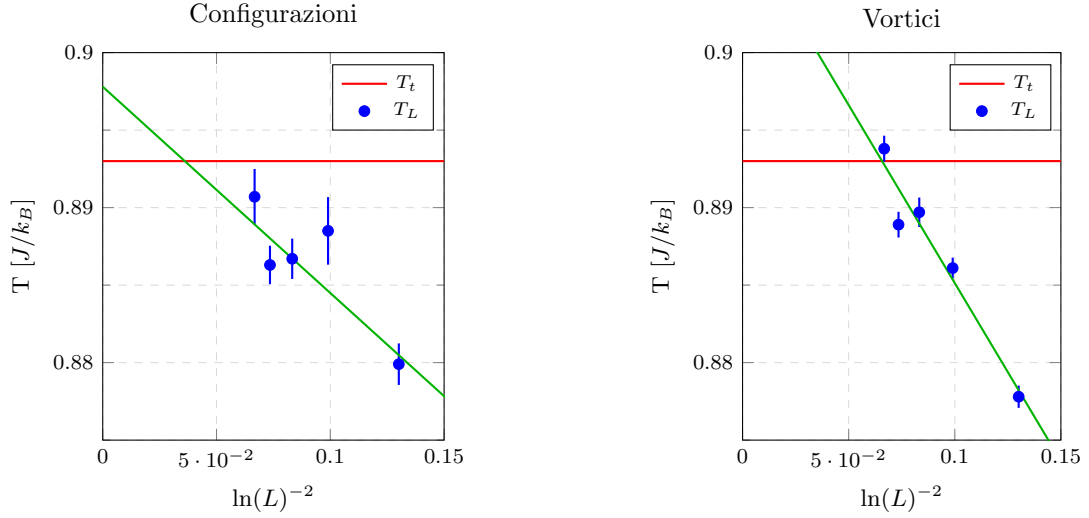


Figura 17: Interpolazione temperatura

Dati	T estratta $[J/k_B]$	T teorica $[J/k_B]$
Configurazioni	0.90(7)	0.893
Vortici	0.91(5)	0.893

Tabella 16: Valori ottenuti tramite il fit per la temperatura di transizione.

Anche per il modello XY si è provato a stimare la temperatura a cui avviene la transizione di fase, sia utilizzando le semplici configurazioni sia utilizzando i vortici. Basandosi sui risultati riportati in

tabella 13, sono state allenate due reti convoluzionali differenti per le due tipologie di dati in input. Per la classificazione basata sulle configurazioni di spin è stata allenata una rete avete il primo layer convoluzionale con kernel  $2 \times 2$ , il secondo  $3 \times 3$  e un layer fully connected a 128 neuroni. Invece per la rete che classifica basandosi sui vortici è stato impostato il primo layer convoluzionale con kernel  $3 \times 3$ , mantenendo le altre caratteristiche invariate.

Nei grafici di figura 17 sono riportati le temperature e gli errori relativi ai reticoli di lato 16, 24, 32, 40 e 48; in tabella 16 sono indicati i valori estratti tramite i due procedimenti.

Entrambi i grafici presentano un andamento anomalo rispetto a quanto ci si aspetterebbe infatti, come riportato in [2], la temperatura di transizione teorica per reticoli di lato finito  $T_L = T(L)$  è legata alla temperatura di transizione del sistema ideale nel limite termodinamico  $T_t$  dalla formula

$$T(L) \approx T_t + \frac{\pi^2}{4c \ln L^2} \quad (4)$$

quindi ci si aspetterebbe una convergenza a  $T_t$  dall'alto.

Come si può osservare dal primo grafico riportato in figura 17, ovvero quello relativo alle configurazioni non elaborate in vortici, i valori per le diverse dimensioni di reticolo presentano barre d'errore estese, mentre le temperature estrapolate dalla classificazione di sistemi di vortici presentano barre d'errore più contenute. In entrambi i casi i dati non sono sufficienti per avere una solida interpolazione lineare, infatti l'errore relativo supera il 5%. Si ottiene però una stima ragionevole della temperatura di transizione, assolutamente compatibile con il valore teorico.

## Riferimenti bibliografici

- [1] J. Carrasquilla and R. G. Melko, “Machine learning phases of matter,” *Nature Physics*, vol. 13, Feb 2017.
- [2] M. J. S. Beach, A. Golubeva, and R. G. Melko, “Machine learning vortices at the kosterlitz-thouless transition,” *Phys. Rev. B*, vol. 97, p. 045207, Jan 2018.
- [3] P. Suchsland and S. Wessel, “Parameter diagnostics of phases and phase transition learning by neural networks,” *Phys. Rev. B*, vol. 97, p. 174435, May 2018.
- [4] U. Wolff, “Collective monte carlo updating for spin systems,” *Phys. Rev. Lett.*, vol. 62, pp. 361–364, Jan 1989.
- [5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [6] M. Matsumoto and T. Nishimura, “Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator,” *ACM Trans. Model. Comput. Simul.*, vol. 8, pp. 3–30, Jan. 1998.
- [7] N. D. Mermin and H. Wagner, “Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic heisenberg models,” *Phys. Rev. Lett.*, vol. 17, pp. 1133–1136, Nov 1966.
- [8] J. Kosterlitz and D. J. Thouless, “Ordering, metastability and phase transitions in two-dimensional systems,” *J. Phys. C: Solid State Phys.*, vol. 6, p. 1181, 04 1973.
- [9] L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-sne,” *Journal of Machine Learning Research*, pp. 2579–2605, Nov 2008.
- [10] F. Chollet *et al.*, “Keras,” 2015. Software available from keras.io.

- [11] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.
- [12] A. E. Ferdinand and M. E. Fisher, “Bounded and inhomogeneous ising models. i. specific-heat anomaly of a finite lattice,” *Phys. Rev.*, vol. 185, pp. 832–846, Sep 1969.