



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA
CURSO DE GRADUAÇÃO EM FÍSICA

DAVI FONSECA PASSOS

**APLICAÇÃO DE REDES DE BOLTZMANN AO APRENDIZADO NÃO
SUPERVISIONADO DO MODELO DE ISING 2D**

FORTALEZA

2025

DAVI FONSECA PASSOS

APLICAÇÃO DE REDES DE BOLTZMANN AO APRENDIZADO NÃO SUPERVISIONADO
DO MODELO DE ISING 2D

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Física do Centro
de Ciências da Universidade Federal do Ceará,
como requisito parcial à obtenção do grau de
bacharel em Física.

Orientador: Prof. Dr. Saulo Davi Soares
e Reis

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- P32a Passos, Davi Fonseca.
 Aplicação de Redes de Boltzmann ao Aprendizado não Supervisionado do Modelo de Ising 2D / Davi
Fonseca Passos. – 2025.
 44 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências,
Curso de Física, Fortaleza, 2025.
 Orientação: Prof. Dr. Saulo Davi Soares e Reis.
1. Modelo de Ising. 2. Máquinas de Boltzmann. 3. Física Estatística. 4. Transições de fase. 5.
Aprendizado não supervisionado. I. Título.

CDD 530

DAVI FONSECA PASSOS

APLICAÇÃO DE REDES DE BOLTZMANN AO APRENDIZADO NÃO SUPERVISIONADO
DO MODELO DE ISING 2D

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Física do Centro
de Ciências da Universidade Federal do Ceará,
como requisito parcial à obtenção do grau de
bacharel em Física.

Aprovada em: 22 de julho de 2025

BANCA EXAMINADORA

Prof. Dr. Saulo Davi Soares e Reis (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. André Auto Moreira
Universidade Federal do Ceará (UFC)

Me. Laísa Viana Quariguasi
Universidade Federal do Ceará (UFC)

A Jesus Cristo, meu Senhor e Salvador, cuja
presença constante sustentou-me em cada etapa
desta caminhada. A Ele toda a honra, pois sem
Sua graça nada disso seria possível.

AGRADECIMENTOS

Aos meus pais, meus familiares do Piauí, meus amigos e meus queridos irmãos da Primeira Igreja Batista de Fortaleza, que, nos momentos de minha ausência dedicados ao estudo, sempre me fizeram entender que o futuro é feito a partir da constante dedicação e que, com fé em Deus, todas as coisas são possíveis.

Ao Professor Dr. Saulo Davi Soares e Reis, por me orientar em todo o processo deste trabalho acadêmico e por ter me ensinado que na vida há processos que por mais que não sejam observáveis, eles fazem parte do nosso crescimento.

Ao aluno de doutorado e amigo, Matheus Santos do curso de ciência da computação da Universidade Federal do Ceará, que com seus conselhos me ajudou bastante na tomada deste tema.

Agradeço a todos os professores do departamento por me proporcionarem o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional e por terem se dedicado tanto a mim durante esta jornada.

“O temor ao Senhor é o princípio do conhecimento, mas os insensatos desprezam a sabedoria e a disciplina.”

(Provérbios 1:7)

RESUMO

A perspectiva deste trabalho é, por meio de métodos de Aprendizado de Máquina, estudar os resultados práticos obtidos por meio da programação aplicados a modelos da física estatística, neste caso, o modelo de Ising bidimensional. Geramos amostras representativas do sistema utilizando um simulador baseado no algoritmo de Wolff, abrangendo diferentes regimes térmicos, com ênfase principal na região crítica, próxima à transição de fase, a qual consideramos $T_c \approx 2.3K$. Essas amostras foram processadas e utilizadas como dados de entrada para o treinamento de modelos generativos estocásticos, especificamente, as Restricted Boltzmann Machines (RBMs) e Deep Boltzmann Machines (DBMs). O treinamento foi realizado de forma não supervisionada, com o uso do algoritmo de *contrastive divergence* e amostragem de Gibbs. As DBMs, formadas por empilhamento de duas RBMs, foram empregadas para reconstruir amostras do sistema e gerar novas configurações sintéticas denominadas *fantasy particles*. Os resultados obtidos mostram que os modelos foram capazes de aprender e replicar características estatísticas do sistema de Ising, como a formação de domínios e padrões espaciais, sobretudo na região crítica. A aplicação de modelos de aprendizado de máquina mostrou-se eficaz na representação e geração de dados de sistemas físicos, sugerindo um caminho promissor para estudos interdisciplinares entre física computacional e inteligência artificial.

Palavras-chave: modelo de Ising; máquinas de Boltzmann; física estatística; transições de fase; aprendizado não supervisionado.

ABSTRACT

The purpose of this work is to study, through Machine Learning methods, the practical results obtained via programming applied to statistical physics models, in this case, the two-dimensional Ising model. Representative samples of the system were generated using a simulator based on the Wolff algorithm, covering different thermal regimes, with a primary focus on the critical region near the phase transition, considered here as $T_c \approx 2.3K$. These samples were processed and used as input data for training stochastic generative models, specifically Restricted Boltzmann Machines (RBMs) and Deep Boltzmann Machines (DBMs). The training was carried out in an unsupervised manner, employing the *contrastive divergence* algorithm and Gibbs sampling. The DBMs, formed by stacking two RBMs, were used to reconstruct system samples and generate new synthetic configurations, known as *fantasy particles*. The results show that the models successfully learned and replicated statistical characteristics of the Ising system, such as domain formation and spatial patterns, particularly in the critical region. The application of machine learning models proved to be effective in representing and generating data from physical systems, suggesting a promising path for interdisciplinary studies bridging computational physics and artificial intelligence.

Keywords: Ising model; Boltzmann machines; statistical physics; phase transitions; unsupervised learning.

LISTA DE FIGURAS

Figura 1 – Amostragem direta	14
Figura 2 – Configurações do modelo de Ising conectadas por um flip de cluster	27
Figura 3 – Visualização comparativa dos dados reais (linha: data), reconstruções pela DBM (linha reconst) e amostras geradas (linha fantasy), todas na fase ordenada $T = 2.0K$	37
Figura 4 – Visualização comparativa dos dados reais (linha: data), reconstruções pela DBM (linha reconst) e amostras geradas (linha fantasy), todas na fase crítica $T = 2.3K$	38
Figura 5 – Visualização comparativa dos dados reais (linha: data), reconstruções pela DBM (linha reconst) e amostras geradas (linha fantasy), todas na fase desordenada $T = 3.0K$	38

LISTA DE SÍMBOLOS

β	Inverso da temperatura, definido como $\beta = 1/(k_B T)$
E_i	Energia do estado i no sistema
$E(\mathbf{v})$	Energia associada a uma configuração \mathbf{v} em uma Máquina de Boltzmann
f	Energia livre por sítio
$G(r)$	Função de correlação de dois spins separados por uma distância r
$H(\sigma)$	Hamiltoniano do sistema para uma configuração de spins σ
h	Campo magnético externo aplicado ao sistema
J	Constante de acoplamento entre spins vizinhos (energia de interação)
k_B	Constante de Boltzmann (1.38×10^{-23} J/K)
λ_{\pm}	Autovalores principais da matriz de transferência T
m	Magnetização média por spin
μ_l	Estado intermediário usado no método da matriz de transferência de Onsager
$P(\mathbf{v})$	Probabilidade de uma configuração \mathbf{v} em uma Máquina de Boltzmann
$\sigma_i, \sigma_j, \sigma_n$	Spin individual em um sítio i, j, n (valores $+1$ ou -1)
$\vec{\sigma}_i$	Vetor spin contínuo no sítio i (no modelo de Heisenberg)
T	Temperatura absoluta em Kelvin (K)
T_c	Temperatura crítica de transição de fase em Kelvin (K)
$T_{\mu, \mu'}$	Elemento da matriz de transferência entre estados μ e μ'
τ	Tempo de relaxação ou autocorrelação próximo à temperatura crítica
θ_i, ϕ_j	Ângulos introduzidos na diagonalização da matriz de Onsager (transformada de Fourier)
ξ	Comprimento de correlação do sistema (escala espacial)
W_{ij}	Peso sináptico (conexão) entre os neurônios i e j em uma Máquina de Boltzmann
v_i	Estado (0 ou 1) do neurônio visível i em uma Máquina de Boltzmann
Z	Função de partição do sistema (soma sobre todos os estados energéticos)

SUMÁRIO

1	MÉTODOS DE MONTE CARLO	13
1.1	Por que Monte Carlo?	13
1.2	Amostragem direta	13
1.3	Algoritmo de Metrópolis	14
1.3.1	<i>Uso do algoritmo de Metropolis</i>	15
2	O MODELO DE ISING	16
2.1	O que é o modelo de Ising?	16
2.2	O problema do modelo de Heisenberg	16
2.3	Solução de Ising	17
2.4	Solução de Lars Onsager	20
2.4.1	<i>Identificação da Temperatura Crítica</i>	21
2.4.2	<i>Sobre a magnetização</i>	22
2.5	O problema do algoritmo de metrópolis	23
2.5.1	<i>Limitações do Algoritmo de Metrópolis perto da Temperatura Crítica</i>	23
2.5.2	<i>Comprimento de correlação $\xi(T)$</i>	24
2.5.3	<i>Tempo de autocorrelação τ</i>	24
2.6	O algoritmo de Wolff	25
2.6.1	<i>Construção do cluster</i>	25
2.6.2	<i>Energia de borda e pesos estatísticos</i>	26
2.6.3	<i>Exemplo de cálculo da probabilidade: $T = T_c$</i>	27
2.7	As máquinas de Boltzmann	27
2.7.1	<i>Definição Geral</i>	27
2.7.2	<i>Relação com a Física Estatística</i>	28
2.7.3	<i>Aprendizado e Dinâmica</i>	28
2.7.4	<i>Limitações e Extensões</i>	29
3	APLICAÇÕES DE APRENDIZADO DE MÁQUINA AO MODELO DE ISING	30
3.1	Metodologia	30
3.1.1	<i>Simulação do Modelo de Ising 2D</i>	30
3.2	Simulação do Modelo de Ising 2D com Algoritmo de Wolff	31

3.2.1	<i>Parâmetros e Inicialização</i>	31
3.2.2	<i>Seleção de Vizinhos com Contorno Periódico</i>	31
3.2.3	<i>Algoritmo de Wolff</i>	31
3.2.4	<i>Geração das Amostras e Visualização</i>	32
3.2.5	<i>Saídas do Simulador</i>	32
3.3	Pré-processamento dos Dados para a DBM	33
3.3.1	<i>Arquitetura dos Modelos</i>	33
3.3.2	<i>Treinamento via Contraste de Divergência</i>	33
3.3.3	<i>Amostragem de Gibbs</i>	33
3.3.4	<i>Treinamento Camada por Camada</i>	34
3.3.5	<i>Gibbs Sampling e Reconstruções</i>	34
3.3.6	<i>Fantasy Particles</i>	35
3.3.7	<i>Fluxo Geral do Código</i>	35
3.3.8	<i>Visualização</i>	35
3.3.9	<i>Saídas do Treinamento</i>	36
4	RESULTADOS	37
4.1	Resultados das DBMs	37
4.1.1	<i>Comparação Visual das Configurações</i>	37
4.2	Análise Qualitativa	38
5	CONCLUSÃO	40
	REFERÊNCIAS	41
	APÊNDICES	43
	APÊNDICE A – BIBLIOTECAS PYTHON	43
	APÊNDICE B – CONFIGURAÇÕES DO MODELO	44

1 MÉTODOS DE MONTE CARLO

1.1 Por que Monte Carlo?

Os métodos de Monte Carlo foram desenvolvidos durante a Segunda Guerra Mundial, nos laboratórios de Los Alamos, por físicos como Stanislaw Ulam e John von Neumann, enquanto trabalhavam no Projeto Manhattan. O nome "Monte Carlo" foi dado em referência ao Cassino de Monte Carlo, localizado na cidade de Mônaco, pois os métodos envolvem o uso de números aleatórios. Os métodos de Monte Carlo são uma classe de técnicas de métodos estatísticos que se baseiam em amostragens aleatórias para resolver problemas determinísticos (Tóth, 2009). Na física estatística, eles são usados, por exemplo, para estimar médias termodinâmicas de sistemas complexos, simulando apenas uma fração representativa dos microestados presentes. Em vez de calcular todas as configurações possíveis, as amostras são geradas com base em regras estocásticas que seguem a distribuição de equilíbrio. Em física computacional, os métodos de Monte Carlo são bastante utilizados quando os combinamos com algoritmos de atualização inteligente, como o algoritmo de Metropolis. Porém, se quisermos entender isso melhor, primeiro devemos discutir um pouco acerca do conceito amostragem direta.

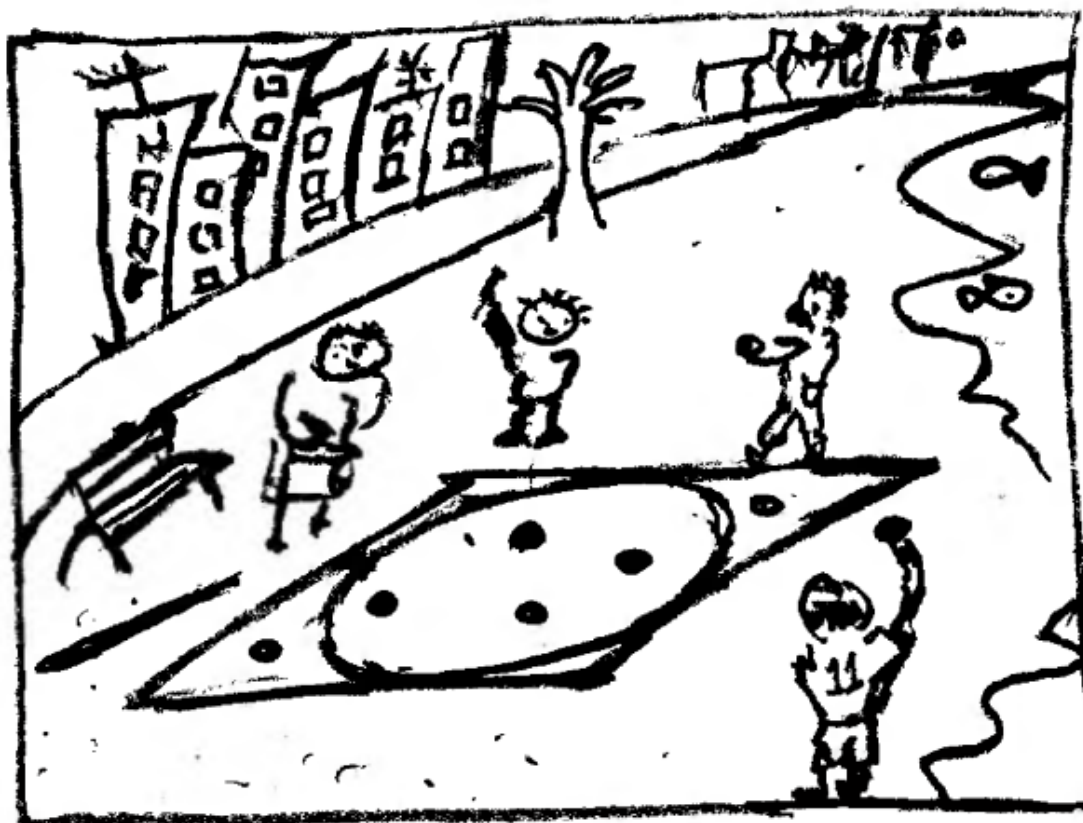
1.2 Amostragem direta

A amostragem direta é um método estatístico onde se geram amostras diretamente vindas de uma distribuição de probabilidade previamente conhecida, sem depender de um processo iterativo. Um bom exemplo que podemos usar para exemplificar esse método é através de um problema clássico: estimar o valor da razão entre a área de um círculo e a de um quadrado que o envolve, sob o raciocínio de um "atirar pedrinhas". O problema consiste em crianças atirando pedras dentro de uma região desenhada, sem que haja um limite no número de pedras. Também é admitido que todas as pedras caem dentro da área do, isto é, todas as amostras são válidas. Após elas arremessarem uma quantidade suficiente de pedrinhas, dentro do círculo central, e, ao final das contas, nós dividimos os dois valores e esperamos que o resultado seja aproximado a $\frac{\pi}{4}$. Ou seja:

$$\frac{N_{circ}}{N_{quad}} \approx \frac{\pi}{4} \quad (1.1)$$

caption

Figura 1 – Amostragem direta



Fonte: Krauth, 2006.

1.3 Algoritmo de Metr polis

O algoritmo de Metropolis foi introduzido em 1953, por Nicholas Metropolis, juntos aos seus colegas de trabalho do Los Alamos National Laboratory (Metropolis *et al.*, 1953). Artigo seminal (Ising, 1925), uma forma eficiente de amostrar distribui  es de probabilidade usando computadores. A ideia deles era gerar seq ncias de estados em que, com o tempo, fosse reproduzido a distribui   de Boltzmann. Esse algoritmo marcou o in cio da simula  o computacional moderna na f sica estat stica e inspirou v rias extens  es, como os algoritmos de Metropolis e de Monte Carlo via Cadeias de Markov (MCMC). A ess ncia do algoritmo est  na decis o de aceitar ou rejeitar uma nova configura  o candidata com base em uma probabilidade de aceita  o que depende da varia  o de energia do sistema.   um algoritmo de amostragem estoc stica usado para gerar configura  es de um sistema f sico (ou qualquer espa o de estados) de acordo com uma distribui  o de probabilidade desejada.

1.3.1 Uso do algoritmo de Metropolis

O método de Monte Carlo pode ser aprimorado utilizando o algoritmo de Metropolis, que evita a geração de pontos completamente aleatórios e, em vez disso, executa uma caminhada estocástica (cadeia de Markov) no domínio de interesse.

1. Inicializa-se em um ponto $(x_0, y_0) \in [0, 1]^2$.
2. Propõe-se um novo ponto $(x', y') = (x + \delta_x, y + \delta_y)$, onde δ_x e δ_y são deslocamentos aleatórios em um intervalo $[-\varepsilon, \varepsilon]$.
3. O ponto proposto é aceito se pertencer ao domínio permitido (o quadrado), isto é:

$$0 \leq x' \leq 1, \quad 0 \leq y' \leq 1$$

Caso contrário, ele é rejeitado e o ponto atual é mantido.

4. Se o ponto proposto for aceito, verifica-se se ele pertence ao quarto de círculo, ou seja, se:

$$x'^2 + y'^2 \leq 1$$

5. Atualiza-se os contadores:

- $N_{\text{total}} \leftarrow N_{\text{total}} + 1$
- Se o ponto estiver dentro do círculo: $N_{\text{dentro}} \leftarrow N_{\text{dentro}} + 1$

6. Repete-se o processo por um número suficientemente grande de iterações.

O algoritmo de Metropolis é projetado para produzir uma sequência de pontos (x_i, y_i) que amostram uniformemente o espaço $[0, 1]^2$, desde que os deslocamentos sejam pequenos e simétricos. Assim, mesmo sem amostragem independente, a proporção de pontos que satisfazem $x^2 + y^2 \leq 1$ se aproxima da razão geométrica entre as áreas:

$$\frac{\pi}{4} \approx \frac{N_{\text{dentro}}}{N_{\text{total}}} \Rightarrow \boxed{\pi \approx 4 \cdot \frac{N_{\text{dentro}}}{N_{\text{total}}}}$$

O uso do algoritmo de Metropolis para estimar π ilustra o poder de métodos estocásticos em problemas de integração e estimação. Ainda que o algoritmo seja mais conhecido por sua aplicação em sistemas físicos (como o modelo de Ising), sua flexibilidade permite aplicá-lo eficientemente em estimativas geométricas como esta. Por isso, tentaremos utilizá-lo no tema proposto por este trabalho.

2 O MODELO DE ISING

2.1 O que é o modelo de Ising?

Em 1924, Wilhelm Lenz, como uma forma de estudar a atividade ferromagnética de alguns materiais, pediu ao seu aluno Ernst Ising que analisasse o modelo em uma dimensão (1D) em sua tese de doutorado (Ising, 1925),(Brush, 1967). Ising resolveu o modelo em 1D e concluiu — corretamente para essa dimensão, mas erroneamente generalizando — que ele não poderia exibir magnetização espontânea, pois o sistema não apresentava transição de fase, o que levou à falsa impressão de que o modelo não seria útil para descrever o magnetismo real. Apenas em 1944 Lars Onsager solucionou o modelo em duas dimensões (2D), na ausência de campo magnético, mostrando que o sistema de fato exibe uma transição de fase (Onsager, 1944). Desde então, o modelo de Ising tornou-se um protótipo para o estudo de ordem e desordem, transições de fase, fenômenos críticos e sistemas complexos.

O motivo de ele ser tão utilizado está justamente em sua simplicidade: (i) é definido por variáveis binárias (spins ± 1) com interações locais, o que o torna matematicamente manejável e computacionalmente eficiente; (ii) em 2D possui solução exata e, ao mesmo tempo, apresenta comportamento crítico não trivial; (iii) ele pode ser mapeado ou adaptado para problemas fora do magnetismo — de redes neurais e aprendizagem, biologia estatística e adsorção em superfícies, o que amplia seu alcance interdisciplinar. Em suma, o Ising é minimalista o suficiente para ser resolvido e, ainda assim, complexo o bastante para capturar a essência de transições de fase e comportamento coletivo.

2.2 O problema do modelo de Heisenberg

Considerando o modelo e a energia contida em seus spins, onde J_{ij} , nossa **constante de acoplamento**, é positiva, e representa a interação entre os spins (σ_n) vizinhos (caso sejam); se eles não forem, ele é considerado nulo. Seja E_{ij} o termo de interação de troca entre dois spins σ_i e σ_j .

$$E_{ij} = -J_{ij}(\vec{\sigma}_i \cdot \vec{\sigma}_j) \quad (2.1)$$

Esse sistema tende a permanecer no seu estado fundamental, o que promove um alinhamento dos seus spins, um comportamento característico de atividade ferromagnética. Além disso, mais tarde, em 1928, Heisenberg considerava um modelo contínuo 3D (Nishikawa *et al.*, 2015),

(Holm; Janke, 1993).

$$H(\sigma) = -J \sum_{\langle i,j \rangle} \vec{\sigma}_i \cdot \vec{\sigma}_j - h \sum_i \sigma_i \quad (2.2)$$

No modelo de Heisenberg, o Hamiltoniano é tratado como o somatório de todos os primeiros vizinhos e seus respectivos spins, claro, acompanhados da constante J. Além disso, a função de partição é definida à partir do somatório de todos os estados de energia do sistema, o que claramente é problemático analítica e computacionalmente falando, e Ising percebeu isso. A conclusão de Ernest Ising foi que as configurações eram infinitas para o sistema, pois é possível observar qualquer direção entre os vetores momento magnético de forma contínua, optando por um modelo 1D.

$$Z = \sum_{E_i} e^{E_i/k_B T} \quad (2.3)$$

Em resumo, acerca do modelo de Heisenberg:

1. É um modelo contínuo (os spins podem apontar em qualquer direção do espaço 3D).
2. Permite descrever interações isotrópicas em materiais reais com magnetismo vetorial.
3. Muito mais difícil de resolver analiticamente devido à complexidade da orientação contínua dos spins.
4. Não possui solução exata geral nem mesmo em 1D.

2.3 Solução de Ising

Ising resolveu o problema das infinitas configurações utilizando uma única coordenada e levando em consideração a projeção dos spins da rede linear. Isso reduziu o sistema para apenas uma dimensão e duas configurações possíveis, spin para cima (sentido positivo) ou spin para abaixo (sentido negativo). Ou seja, avaliando uma rede linear, o uso do produto escalar para os vetores de momento magnético não é mais necessário (Kaya, 2018), (Wei, 2018), (Prados; Brey, 2001) e (Shadrin; Panov, 2023). Em seguida, ele utilizou uma matriz de transferência para os autovalores da função de partição, e, por sua vez, a magnetização espontânea. Para N spins $\sigma_i \in [-1, +1]$ o Hamiltoniano com interações entre vizinhos e sem campo externo é:

$$H(\sigma) = -J \sum_i^N \sigma_i \sigma_{i+1} - h \sum_i^N \sigma_i \quad (2.4)$$

E tendo as condições de contorno periódicas; $\sigma_1 = \sigma_{N+1}$. A nossa função de partição para a solução 1D, onde $\beta = k_B T$ fica:

$$Z = \sum_{\sigma} e^{-\beta H(\sigma)} = \sum_{\sigma} \exp \left[\beta J \sum_{i=1}^N \sigma_i \sigma_{i+1} + \beta h \sum_{i=1}^N \sigma_i \right] \quad (2.5)$$

Cada par $(\sigma_i \sigma_{i+1})$ contribui com:

$$\exp \left[\beta J \sigma_i \sigma_{i+1} + \frac{\beta h}{2} \sigma_i \sigma_{i+1} \right] \quad (2.6)$$

Sendo assim, podemos usar o método da Matriz de transferência. Seja a nossa Matriz de transferência T , tal que, $T_{\sigma, \sigma'}$, simboliza a contribuição entre dois spins consecutivos $\sigma, \sigma' \in [-1, +1]$:

$$T_{\sigma, \sigma'} = e^{\beta J \sigma \sigma'} \implies T = \begin{bmatrix} T_{+,+} & T_{-,+} \\ T_{+,-} & T_{-,-} \end{bmatrix} = \begin{bmatrix} e^{\beta J + \beta h} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J - \beta h} \end{bmatrix} \quad (2.7)$$

Seja $e^{\beta J + \beta h} = a$; $e^{-\beta J} = b$ e $e^{\beta J - \beta h} = c$. Agora, aplicamos a Equação de Autovalores para determinarmos os autovalores de T :

$$\det(T - \lambda I) = \begin{vmatrix} a - \lambda & b \\ b & c - \lambda \end{vmatrix} = 0 \quad (2.8)$$

Expandindo, nós temos:

$$\lambda^2 - (a + c)\lambda + (ac - b^2) = 0 \quad (2.9)$$

Portanto, as nossas raízes para solução de lambda são:

$$\lambda_{\pm} = \frac{(a + c)}{2} \pm \sqrt{\left(\frac{a - c}{2}\right)^2 + b^2} \quad (2.10)$$

$$\lambda_{\pm} = \frac{e^{\beta J + \beta h} + e^{\beta J - \beta h}}{2} \pm \sqrt{\left(\frac{e^{\beta J + \beta h} - e^{\beta J - \beta h}}{2}\right)^2 + e^{-2\beta J}} \quad (2.11)$$

$$\lambda_{\pm} = e^{\beta J} \cosh(\beta h) \pm \sqrt{e^{2\beta J} \sinh^2(\beta J) + e^{-2\beta J}} \quad (2.12)$$

Agora podemos resolver a função de partição.

$$Z = \exp \left[\beta J \sigma_i \sigma_{i+1} + \frac{\beta h}{2} \sigma_i \sigma_{i+1} \right] = \sum_{\sigma_1, \dots, \sigma_N} T_{\sigma_1, \sigma_2} T_{\sigma_2, \sigma_3} \dots T_{\sigma_N, \sigma_1} = \text{Tr}(T^N) \quad (2.13)$$

O traço é uma soma sobre os elementos diagonais da matriz T^N , ou seja, representa a soma sobre todos os caminhos possíveis que retornam ao mesmo estado após N passos. Lembrando que: $T = PDP^{-1}$ e $T^N = PD^N P^{-1}$; onde D é a matriz dos autovalores de T , os quais já encontramos (Baxter, 2011). Portanto:

$$Z = Tr(T^N) = \lambda_+^N + \lambda_-^N \quad (2.14)$$

Sabendo que $\lambda_+ > \lambda_-$, quando $N \rightarrow \infty$:

$$\left(\frac{\lambda_-}{\lambda_+}\right)^N \ll 1 \rightarrow 0 \implies \lambda_-^N = \lambda_+^N \left(\frac{\lambda_-}{\lambda_+}\right)^N \rightarrow 0 \quad (2.15)$$

Então podemos afirmar que:

$$Z \approx \lambda_+^N = \left(\lambda_+ = e^{\beta J} \cosh(\beta h) + \sqrt{e^{2\beta J} \sinh^2(\beta J) + e^{-2\beta J}} \right)^N \quad (2.16)$$

Onde chamaremos $A = e^{\beta J} \cosh(\beta h)$ e $B = \sqrt{e^{2\beta J} \sinh^2(\beta J) + e^{-2\beta J}}$; $\lambda_+ = A + B$ e a magnetização por spin sendo:

$$m = \frac{1}{N} \sum_{i=1}^N \langle \sigma_i \rangle = \frac{1}{N} \frac{\partial \ln Z}{\partial(\beta h)} = \frac{1}{N} \frac{\partial \ln \lambda_+^N}{\partial(\beta h)} = \frac{\partial \ln \lambda_+}{\partial(\beta h)} \quad (2.17)$$

Derivando λ_+ :

$$\frac{d}{d(\beta h)} \ln(\lambda_+) = \frac{1}{\lambda_+} \cdot \frac{d\lambda_+}{d(\beta h)} = \frac{1}{A+B} \cdot \left(\frac{dA}{d(\beta h)} + \frac{dB}{d(\beta h)} \right) \quad (2.18)$$

Encontrando a derivada de $A = e^{\beta J} \cosh(\beta h)$:

$$\frac{dA}{d(\beta h)} = e^{\beta J} \sinh(\beta h) \quad (2.19)$$

Encontrando a derivada de $\sqrt{e^{2\beta J} \sinh^2(\beta h) + e^{-2\beta J}}$:

$$\frac{dB}{d(\beta h)} = \frac{1}{2B} \cdot \frac{d}{d(\beta h)} (e^{2\beta J} \sinh^2(\beta h) + e^{-2\beta J}) = \frac{1}{2B} \cdot 2e^{2\beta J} \sinh^2(\beta h) \cosh(\beta h) \quad (2.20)$$

$$\frac{dB}{d(\beta h)} = \frac{e^{2\beta J} \sinh(\beta h) \cosh(\beta h)}{B} \quad (2.21)$$

Somando tudo, nós temos que:

$$\frac{d\lambda_+}{d(\beta h)} = e^{\beta J} \sinh(\beta h) + \frac{e^{2\beta J} \sinh(\beta h) \cosh(\beta h)}{B} \quad (2.22)$$

Logo:

$$m = \frac{1}{A+B} \cdot \left(\frac{dA}{d(\beta h)} + \frac{dB}{d(\beta h)} \right) = \frac{\sinh(\beta h)}{A+B} \left[e^{\beta J} + \frac{e^{2\beta J} \cosh(\beta h)}{B} \right] \quad (2.23)$$

Deixando de modo mais compacto:

$$m = \frac{\sinh(\beta h)}{\sqrt{\sinh^2(\beta h) + e^{-4\beta J}}} \quad (2.24)$$

Se campo externo $h \rightarrow 0$, não há magnetização espontânea entre os spins.

Apesar de apresentar melhora analítica em relação ao modelo de Heisenberg, o modelo de Ising era muito limitado por sua característica 1D, a qual não conseguia descrever um comportamento ferromagnético real e não apresenta transição de fase a temperatura finita.

2.4 Solução de Lars Onsager

Quando Lars analisou os resultados de Ising [6], dessa vez aplicando suas ideias em um modelo bidimensional, nós consideramos uma rede de spins $\sigma_i \sigma_j$ que também podem assumir os valores $\sigma_i, \sigma_j \in [-1, +1]$. Os spins interagem com seus vizinhos mais próximos e, eventualmente, com um campo externo h , que pode ser nulo em alguns casos, como foi na solução de Onsager. Portanto, o Hamiltoniano do sistema é dada por:

$$H(\sigma) = -J \sum_i^N \sum_j^N (\sigma_{i,j} \sigma_{i,j+1} + \sigma_{i+1,j} \sigma_{i,j}) \quad (2.25)$$

Para o modelo de Ising 2D com interações apenas entre vizinhos mais próximos, interação de acoplamento J (isotrópico), e sem campo externo, a função de partição por sítio no limite termodinâmico é dada por:

$$Z = \sum_{\sigma} \exp [\beta J \sigma_{i+1,j} \sigma_{i,j} + \sigma_{i,j} \sigma_{i,j+1}] \quad (2.26)$$

Cada linha $\mu = (\sigma_i, \sigma_j)$ é um estado possível.

$$T_{\mu, \mu'} = \exp \left[\beta J \sum_l^N \mu_l \mu_{l+1} + \beta J \sum_l^N \mu_l \mu'_l \right] \quad (2.27)$$

Porém, se isso proceguisse, acabaríamos nos deparando com uma matriz 4x4, a qual seria muito trabalhosa resolver manualmente. Então, ao invés de apelarmos para um processo tão mecânica, optaremos por usar transformadas de Fourier para diagonalizar a matriz, assim, podendo expressar Z em termos dos autovalores de T :

$$Z = \prod_{i=1}^N \prod_{j=1}^N \lambda_{i,j}^N \quad (2.28)$$

O que implica que, pela regra do logaritmo:

$$\ln Z = \ln \left(\prod_{i=1}^N \prod_{j=1}^N \lambda_{i,j}^{1/2} \right) = \sum_i \sum_j \ln(\lambda_{i,j}^{1/2}) = \frac{1}{2} \sum_i \sum_j \ln(\lambda_{i,j}) = \frac{1}{2} \sum_{i,j} \ln(\lambda_{i,j}) \quad (2.29)$$

A matriz T tem propriedade que podemos explorar: Primeiro: Ela é translacionalmente invariante, ou seja, de rede regular; Segunda: Os spins nela só irão interagir com seus vizinhos imediatos (cima, baixo, esquerda e direita), onde todos os spins de borda da matriz são periódicos.

Com isso, podemos fazer a seguinte associação:

- $\theta_i = \frac{(2i-1)\pi}{2N}$
- $\phi_j = \frac{(2j-1)\pi}{2N}$

Substituindo em λ , conforme a diagonalização, que leva essa transformação na forma do autovalores da matriz:

$$\lambda_{i,j} = \cosh^2(2\beta J) - \sinh(2\beta J) \cdot [(\cos \theta_i + \cos \phi_j)] \quad (2.30)$$

Esses são os modos próprios (autovalores) da matriz de transferência, expressos agora em termos dos ângulos: θ_i e ϕ_j , que vem da transformada de Fourier (Bhattacharjee; Khare, 1995).

Substituindo isso dentro do somatório:

$$\ln Z = \frac{1}{2} \sum_i \sum_j \ln [\cosh^2(2\beta J) - \sinh(2\beta J)(\cos \theta_i + \cos \phi_j)] + C \quad (2.31)$$

Após isso, analisamos que, no limite de $N \rightarrow \infty$.

$$\ln Z \approx \frac{N^2}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} \ln [\lambda(\theta, \phi)] d\theta d\phi \quad (2.32)$$

Substituindo os valores da equação (2.33).

$$\ln Z \approx \frac{N^2}{(2\pi)^2} \left\{ \int_0^{2\pi} \int_0^{2\pi} \ln [\cosh^2(2\beta J) - \sinh(2\beta J)(\cos \theta + \cos \phi)] d\theta d\phi \right\} \quad (2.33)$$

Tomando $\kappa = \frac{2\sinh(2\beta J)}{\cosh^2(2\beta J)}$ e resolvendo a integral em $d\phi$, obtemos:

$$\ln Z \approx N^2 \left\{ \ln[2\cosh(2\beta J)] + \frac{1}{2\pi} \int_0^\pi \ln \left[\frac{1}{2} \left(1 + \sqrt{1 - \kappa^2 \sin^2 \theta} \right) \right] d\theta \right\} \quad (2.34)$$

2.4.1 Identificação da Temperatura Crítica

A transição de fase ocorre quando a função dentro do logaritmo se torna não analítica, o que acontece quando o argumento da raiz quadrada se anula em algum ponto da integral, ou seja, quando:

$$1 - \kappa^2 \sin^2(\theta) = 0 \quad (2.35)$$

Para $\sin^2(\theta) = 1 \implies \kappa = \pm 1$, mas vamos assumir o resultado positivo, tal que:

$$\frac{2 \sinh(2\beta J)}{\cosh^2(2\beta J)} = 1 \implies \sinh^2(2\beta J) - 2 \sinh(2\beta J) + 1 = 0 \quad (2.36)$$

Ou seja:

$$(\sinh(2\beta J) - 1)^2 = 0 \implies \sinh(2\beta J) = 1 \implies 2\beta J = \sinh^{-1}(1) = \ln(1 + \sqrt{2}) \quad (2.37)$$

Conluímos então que:

$$2\beta J = \frac{2J}{k_B T_c} = \ln(1 + \sqrt{2}) \implies T_c = \frac{2J}{k_B \ln(1 + \sqrt{2})} \quad (2.38)$$

Portanto, para $J = 1$ e $k_B = 1$ (o que é majoritariamente usado computacionalmente), nós temos $T_c = \ln(1 + \sqrt{2}) \approx 2.3K$

2.4.2 Sobre a magnetização

Onsager não calculou diretamente a magnetização, mas ele chegou a forte evidência de sua existência por meio de uma associação simples usando a equação do calor específico C , relacionado-o a energia livre por sítio f . Seja a energia livre:

$$f = -\frac{1}{N^2} k_B T \ln Z \quad (2.39)$$

$$f = -k_B T \left\{ \ln[2 \cosh(2\beta J)] + \frac{1}{2\pi} \int_0^\pi \ln \left[\frac{1}{2} \left(1 + \sqrt{1 - \kappa^2 \sin^2 \theta} \right) \right] d\theta \right\} \quad (2.40)$$

Sendo a equação do calor específico:

$$C = -T \frac{d^2 f}{dT^2} \quad (2.41)$$

Porém, temos em $f(T)$ a integral $\sqrt{1 - \kappa^2 \sin^2 \theta}$, e como $\kappa \rightarrow 1$ e o valor máximo de $\sin^2(\theta)$ é 1, quando $T \rightarrow T_c$, significa que o denominador da raiz se aproxima de 0, gerando uma divergência logarítmica. Chamaremos então:

- $f1 = -k_B T \ln[2 \cosh(2\beta J)]$
- $f2 = \frac{-k_B T}{2\pi} \int_0^\pi \ln \left[\frac{1}{2} \left(1 + \sqrt{1 - \kappa^2 \sin^2 \theta} \right) \right] d\theta$; $I = \int_0^\pi \ln \left[\frac{1}{2} \left(1 + \sqrt{1 - \kappa^2 \sin^2 \theta} \right) \right] d\theta$

Resolvendo $f1$:

$$\frac{d}{dt} f1(T) = -k_B \ln[2 \cosh(2\beta J)] + k_B T \cdot \frac{d}{dT} \ln[2 \cosh(2\beta J)] \quad (2.42)$$

Como $\beta = \frac{1}{k_B T}$, temos:

$$\frac{df_1}{dT} = -k_B \ln[2 \cosh(2\beta J)] - \frac{2J}{k_B T^2} [\sinh(2\beta J)] \quad (2.43)$$

Que é um resultado meramente analítico e suave, ou seja, f_1 não contribui significativamente com a singularidade do ponto crítico.

$$\frac{df_2}{dT} = \frac{-k_B I}{2\pi} + \frac{k_B T}{2\pi} \int_0^\pi \left[\left(\frac{1}{1 + \sqrt{1 - \kappa^2 \sin^2 \theta}} \right) \cdot \left(\frac{1}{2\sqrt{1 - \kappa^2 \sin^2 \theta}} \frac{d}{dT} (1 - \kappa^2 \sin^2 \theta) \right) \right] d\theta \quad (2.44)$$

É evidente a dependencia do termo κ^2 relacionado a T , o qual já observamos que explode quando $\kappa \rightarrow 1$, pois: $\frac{1}{\sqrt{1 - \kappa^2 \sin^2 \theta}}$. O qual mostra ser divergente em $\theta = \frac{\pi}{2}$ (onde, novamente, $\kappa = 1$). Então, no final das contas, podemos afirmar com segurança que o calor específico $C(T) \sim \ln|T - T_c|$. Este é o resultado encontrado por Onsager, isto é: o calor específico diverge logaritmicamente na transição de fase [14].

2.5 O problema do algoritmo de metrópolis

O algoritmo de Metropolis revolucionou a simulação computacional de sistemas estatísticos. A ideia do algoritmo é simular o comportamento térmico de sistemas complexos gerando amostras da distribuição de Boltzmann $P(\sigma) = \frac{1}{Z} e^{-\beta E(\sigma)}$. Tal que:

$$P_{\text{aceite}} = \begin{cases} 1, & \text{caso } \Delta E \leq 0, \\ e^{-\beta \Delta E}, & \text{caso } \Delta E > 0. \end{cases}$$

O problema é que, para um sistema físico como o Modelo de Ising em duas dimensões, principalmente quando perto da zona crítica, que é uma lentidão exponencial da execução do código (Ivaneyko *et al.*, 2006). A razão disso é que o algoritmo de Metropolis modifica a configuração de spins localmente (um spin por vez), o que torna difícil que grandes domínios de spins sejam invertidos e reorganizados.

2.5.1 Limitações do Algoritmo de Metrópolis perto da Temperatura Crítica

O algoritmo de Metrópolis, apesar de amplamente utilizado para simulações de modelos de spins como o de Ising, apresenta sérias limitações nas proximidades da **temperatura crítica** T_c . Nessas condições, surge o fenômeno conhecido como *critical slowing down*, em que

a dinâmica do algoritmo torna-se extremamente lenta. Para compreender essas limitações, é necessário introduzir os conceitos de **comprimento de correlação** e **tempo de autocorrelação**.

2.5.2 Comprimento de correlação $\xi(T)$

O **comprimento de correlação** $\xi(T)$ mede o tamanho típico dos domínios de spins que permanecem correlacionados entre si. A correlação entre dois spins separados por uma distância r é dada pela **função de correlação**:

$$G(r) = \langle \sigma_0 \sigma_r \rangle - \langle \sigma_0 \rangle^2 \sim e^{-r/\xi(T)}. \quad (2.45)$$

Essa expressão mostra que a correlação entre os spins decai **exponencialmente** com a distância r , e o parâmetro $\xi(T)$ controla a taxa de decaimento. Quanto maior $\xi(T)$, mais spins permanecem alinhados, mesmo a distâncias maiores.

Próximo à temperatura crítica T_c , o sistema apresenta **flutuações de longo alcance** e o comprimento de correlação $\xi(T)$ **diverge** de acordo com uma lei de potência:

$$\xi(T) \sim |T - T_c|^{-\nu} \quad \Rightarrow \quad \xi(T) \rightarrow \infty \text{ quando } T \rightarrow T_c, \quad (2.46)$$

onde ν é o **expoente crítico do comprimento de correlação**. Essa divergência significa que, em T_c , **todo o sistema torna-se correlacionado**, dificultando a atualização local de spins.

2.5.3 Tempo de autocorrelação τ

Além do comprimento de correlação, outro conceito importante é o **tempo de autocorrelação** τ , que mede quantos passos de Monte Carlo são necessários para que duas configurações sucessivas sejam **estatisticamente independentes**.

Esse tempo escala com o comprimento de correlação como:

$$\tau \sim \xi^z, \quad (2.47)$$

onde z é o **expoente dinâmico**, característico da dinâmica do algoritmo.

Assim, como $\xi(T)$ diverge em T_c , o tempo de autocorrelação também diverge:

$$\tau \sim |T - T_c|^{-\nu z} \quad \Rightarrow \quad \tau \rightarrow \infty \text{ quando } T \rightarrow T_c. \quad (2.48)$$

Essa divergência tem implicações diretas para as simulações:

- Grandes domínios correlacionados dificultam a exploração eficiente do espaço de estados.

- Flips locais de um único spin tornam-se ineficazes para modificar significativamente a configuração global.
- Simulações próximas de T_c tornam-se computacionalmente caras, pois são necessários muitos mais passos para gerar amostras independentes.

2.6 O algoritmo de Wolff

A simulação de modelos estatísticos de spins, como o modelo de Ising, tornou-se uma ferramenta central no estudo de transições de fase e comportamento crítico. Para esse fim, os métodos de Monte Carlo baseados em cadeias de Markov são amplamente utilizados. No entanto, algumas limitações práticas afetam o seu funcionamento efetivo, de fato, especialmente nas proximidades da temperatura crítica T_c . O algoritmo de Wolff, proposto em 1989 por Ulli Wolff (Wolff, 1989), surgiu como uma alternativa eficiente ao algoritmo de Metropolis, sendo ampla e particularmente usado no tratamento de fenômenos críticos e na redução do tempo de correlação entre amostras sucessivas. Ou seja, ao invés de criar um tratamento individual para cada spin da rede, ele cria um *cluster*, ou um conjunto, o qual os spins entram conforme a probabilidade p para os vizinhos próximos. Próximo à temperatura crítica T_c , algoritmos locais como o de Metropolis tornam-se ineficientes (Zhong *et al.*, 2020). Isso se deve ao fato de que os flips locais de um único spin afetam pouco a configuração global (Mehta *et al.*, 2019), especialmente quando domínios grandes e correlacionados se formam. Desse modo, o algoritmo de Wolff mostra-se uma excelente solução.

2.6.1 Construção do cluster

Começamos com uma configuração a , e construímos um cluster de spins alinhados da seguinte forma:

- Seleciona-se aleatoriamente um sítio j , com spin σ_j ;
- Inicializa-se o cluster $C = \{j\}$ e o *pocket* $\mathcal{P} = \{j\}$;
- Enquanto $\mathcal{P} \neq \emptyset$, repete-se:
 - Escolher $k \in \mathcal{P}$, remover de \mathcal{P} ;
 - Para cada vizinho i de k , se $\sigma_i = \sigma_k$, adicioná-lo a C com probabilidade p ;
 - Os novos adicionados também entram no *pocket* \mathcal{P} .
- Após a construção, todos os spins $\sigma_i \in C$ são invertidos: $\sigma_i \rightarrow -\sigma_i$.

2.6.2 Energia de borda e pesos estatísticos

A energia da configuração está localizada na borda ∂C do cluster, isto é, nas conexões entre spins dentro e fora do cluster. Definimos:

- n_1 : número de arestas do tipo $(+, -)$ ou $(-, +)$ na borda ∂C ;
- n_2 : número de arestas do tipo $(+, +)$ ou $(-, -)$ que poderiam ter sido adicionadas ao cluster.

A energia associada à borda é dada por:

$$E|_{\partial C} = n_1 - n_2, \quad (2.49)$$

O peso de Boltzmann da configuração a é:

$$\pi(a) = \pi_{\text{in}} \cdot \pi_{\text{out}} \cdot e^{-\beta(n_1 - n_2)} \quad (2.50)$$

A probabilidade a priori de construir o cluster e parar é:

$$\mathcal{A}(a \rightarrow b) = \mathcal{A}_{\text{in}} \cdot (1 - p)^{n_2} \quad (2.51)$$

Para preservar a medida de Boltzmann, exigimos equilíbrio detalhado:

$$\pi(a) \mathcal{A}(a \rightarrow b) \mathcal{P}(a \rightarrow b) = \pi(b) \mathcal{A}(b \rightarrow a) \mathcal{P}(b \rightarrow a) \quad (2.52)$$

Onde $\pi(b)$, similarmente a $\pi(a)$, porém seguindo caminho $b \rightarrow a$:

$$\pi(b) = \pi_{\text{in}} \cdot \pi_{\text{out}} \cdot e^{-\beta(n_2 - n_1)} \quad (2.53)$$

Substituindo os termos:

$$e^{-\beta(n_1 - n_2)} (1 - p)^{n_2} \mathcal{P}(a \rightarrow b) = e^{-\beta(n_2 - n_1)} (1 - p)^{n_1} \mathcal{P}(b \rightarrow a) \quad (2.54)$$

Assumindo reversibilidade ($\mathcal{P}(b \rightarrow a) = 1$), obtemos a probabilidade de aceitação:

$$\mathcal{P}(a \rightarrow b) = \min \left[1, \frac{e^{-2\beta(n_2 - n_1)} (1 - p)^{n_1}}{(1 - p)^{n_2}} \right] \quad (2.55)$$

A equação acima se simplifica e resulta em $\mathcal{P} = 1$ (aceitação sempre) quando:

$$p = 1 - e^{-2\beta J} \quad (2.56)$$

Essa é a escolha padrão no algoritmo de Wolff, garantindo transições sempre aceitas, balanceadas estatisticamente com a distribuição de Boltzmann. O algoritmo de Wolff é bastante eficiente próximo à temperatura crítica, onde o algoritmo de Metropolis sofre de *critical slowing down*. Ele permite atualizações de grande escala, reduz a correlação entre amostras consecutivas e é universalmente aceito com a escolha ideal de p .

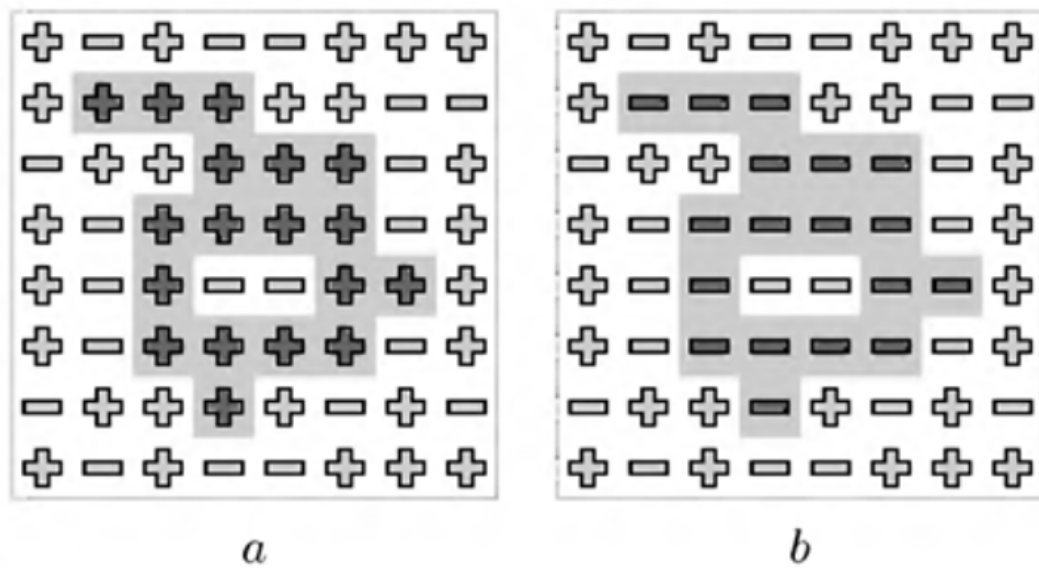
2.6.3 Exemplo de cálculo da probabilidade: $T = T_c$

Para $J = k_B = 1$, calcularemos a probabilidade de um spin a temperatura crítica $T \approx 2.3K$ entrar no *cluster*, temos:

$$\beta = \frac{1}{k_B T} \approx \frac{1}{2.3}, \quad p_{\text{add}} = 1 - e^{-2\beta} \approx 1 - e^{-0.88} \approx 0.58$$

Assim, cada vizinho com o mesmo spin é adicionado ao cluster com cerca de 58% de chance.

Figura 2 – Configurações do modelo de Ising conectadas por um flip de cluster



Fonte: Krauth (2006). Oxford University Press, 2006.

2.7 As máquinas de Boltzmann

As **Máquinas de Boltzmann** (Boltzmann Machines –) são modelos **generativos baseados em energia**, introduzidos por Hinton e Sejnowski na década de 1980 (Hinton; Sejnowski, 1985), cujo objetivo é representar distribuições de probabilidade complexas por meio de uma função de energia associada a um grafo de neurônios binários estocásticos.

2.7.1 Definição Geral

Uma BM é composta por um conjunto de **neurônios binários** $\{v_i\}$ que assumem valores em $\{0, 1\}$ (ou $\{-1, +1\}$, por analogia ao modelo de Ising). A configuração do sistema $\mathbf{v} = (v_1, v_2, \dots, v_N)$ é descrita por uma **função de energia**:

$$E(\mathbf{v}) = - \sum_{i < j} W_{ij} v_i v_j - \sum_i b_i v_i, \quad (2.57)$$

onde:

- W_{ij} são os **pesos sinápticos** (interações entre neurônios),
- b_i é um **viés local** para cada neurônio.

A probabilidade de uma configuração \mathbf{v} segue a **distribuição de Boltzmann**:

$$P(\mathbf{v}) = \frac{1}{Z} e^{-E(\mathbf{v})}, \quad Z = \sum_{\mathbf{v}} e^{-E(\mathbf{v})}, \quad (2.58)$$

onde Z é a **função de partição** que normaliza a distribuição.

2.7.2 Relação com a Física Estatística

As BMs possuem uma analogia direta com sistemas físicos como o **modelo de Ising**:

- Os neurônios equivalem aos **spins** do sistema,
- Os pesos W_{ij} representam as **interações de acoplamento**,
- A energia $E(\mathbf{v})$ é análoga ao **Hamiltoniano** do sistema,
- A probabilidade $P(\mathbf{v})$ segue a mesma forma da **distribuição de Boltzmann** de um sistema em equilíbrio térmico.

Assim, BMs podem ser vistas como um sistema artificial que aprende distribuições da mesma forma que um sistema físico atinge equilíbrio térmico.

2.7.3 Aprendizado e Dinâmica

O treinamento de uma BM busca ajustar os pesos W_{ij} para aproximar uma **distribuição alvo** P_{data} . Isso é feito minimizando a **divergência de Kullback-Leibler** entre P_{data} e $P(\mathbf{v})$.

Para amostrar da distribuição da BM, utiliza-se um processo estocástico de Monte Carlo, geralmente **Gibbs Sampling**, que alterna entre atualizar cada neurônio de forma condicional:

$$P(v_i = 1 | \mathbf{v}_{-i}) = \sigma \left(\sum_j W_{ij} v_j + b_i \right), \quad (2.59)$$

onde $\sigma(x)$ é a função sigmoide logística.

2.7.4 Limitações e Extensões

Apesar de poderosas, as BMs tradicionais são **difíceis de treinar** pois:

- O cálculo exato da função de partição Z é inviável em sistemas grandes,
- A convergência do Gibbs Sampling pode ser lenta, especialmente em regiões críticas.

Para superar essas limitações, surgiram extensões como:

- **Restricted Boltzmann Machines ()** – restringem conexões dentro de uma camada, simplificando o treinamento;
- **Deep Boltzmann Machines ()** – empilham várias RBMs, permitindo capturar **correlações hierárquicas** e distribuições mais complexas.

Resumo: As Máquinas de Boltzmann são **modelos probabilísticos inspirados em sistemas físicos**, capazes de representar distribuições complexas via uma função de energia. Essa analogia natural com a física estatística as torna especialmente adequadas para modelar **sistemas como o modelo de Ising**, que compartilha a mesma estrutura energética (Ghojogh *et al.*, 2021).

3 APLICAÇÕES DE APRENDIZADO DE MÁQUINA AO MODELO DE ISING

Neste capítulo, exploramos a intersecção entre aprendizado de máquina (ML) e física estatística, com foco na identificação de transições de fase no modelo de Ising utilizando classificadores supervisionados. A metodologia principal discutida se baseia em amostras geradas por Monte Carlo para o modelo de Ising bidimensional em uma rede 128×128 , onde as configurações de spin são rotuladas como: ordenada ($T < T_c$), crítica ($T \approx T_c$) e desordenada ($T > T_c$). A abordagem adotada é a regressão logística, uma técnica de classificação binária que utiliza uma função de custo baseada na entropia cruzada. As configurações do modelo são achatadas em vetores 1D e usadas como entrada no modelo. O desempenho do classificador é avaliado em dados de teste e também em amostras próximas da região crítica, destacando a dificuldade adicional de classificar estados nessa faixa devido ao critical slowing down do algoritmo de Metrópolis. Vale ressaltar que utilizaremos o Python como a linguagem a ser responsável por fazer tudo isso acontecer.

3.1 Metodologia

Este trabalho visa explorar o uso de modelos generativos baseados em energia, como as Restricted Boltzmann Machines (RBMs) e Deep Boltzmann Machines (DBMs), na modelagem do modelo de Ising bidimensional. A seguir, detalha-se a metodologia empregada, desde a simulação até a análise dos resultados.

3.1.1 Simulação do Modelo de Ising 2D

Foi desenvolvido um simulador do modelo de Ising bidimensional com condições de contorno periódicas, utilizando o algoritmo de Wolff. Foram geradas configurações de spins para diferentes temperaturas em uma rede de tamanho 128×128 . As amostras foram coletadas em três regimes térmicos distintos:

- Fase ordenada ($T_c = 2.0K$);
- Ponto crítico ($T_c \approx 2.3K$);
- Fase desordenada ($T_c = 3.0K$).

As configurações foram armazenadas em formato NumPy (.npy), representando matrizes com valores $+1$ e -1 .

3.2 Simulação do Modelo de Ising 2D com Algoritmo de Wolff

Para gerar as configurações do modelo de Ising bidimensional, foi implementado um simulador baseado no **algoritmo de Wolff**, que é um método eficiente para atualizar sistemas próximos ao ponto crítico, reduzindo significativamente o problema de *critical slowing down* em comparação ao algoritmo de Metropolis tradicional.

3.2.1 Parâmetros e Inicialização

O sistema é representado por uma grade quadrada de tamanho $L \times L$, com $L = 128$. Cada sítio da rede possui um *spin* $\sigma_{i,j} \in \{-1, +1\}$, inicializado aleatoriamente. A temperatura é definida como $T = 2.3$, próxima à região crítica do modelo, e o parâmetro de interação é $J = 1$.

A probabilidade de adição de um *spin* vizinho ao *cluster* é dada por:

$$p = 1 - \exp(-2\beta J),$$

onde $\beta = \frac{1}{T}$ (considerando $k_B = 1$). Essa probabilidade garante que a evolução do sistema satisfaça o balanço detalhado.

3.2.2 Seleção de Vizinhos com Contorno Periódico

Para cada *spin* selecionado, consideram-se apenas seus quatro vizinhos mais próximos (geometria de cruz ou *von Neumann neighborhood*) com condições de contorno periódicas. Assim, o sistema não apresenta bordas, mantendo a simetria translacional:

$$\text{vizinhos}(i, j) = \{(i \pm 1 \bmod L, j), (i, j \pm 1 \bmod L)\}.$$

3.2.3 Algoritmo de Wolff

O algoritmo de Wolff atualiza os *clusters* inteiros de spins correlacionados:

1. Escolher aleatoriamente um sítio (i, j) da rede como *sítio-semente*.
2. Criar um *cluster* inicial contendo apenas esse sítio.
3. Para cada *spin* na fronteira do cluster, verificar seus vizinhos:
 - Se o vizinho tem o mesmo valor de spin do sítio-semente;

- E se um número aleatório $r \in [0, 1)$ satisfaz $r < p$, onde $p = e^{-2\beta J}$ então esse vizinho é adicionado ao cluster.
4. Após a expansão do cluster, todos os spins pertencentes a ele são **invertidos simultaneamente** $\sigma \rightarrow -\sigma$.

Esse processo respeita o balanço detalhado e garante uma evolução correta da distribuição de Boltzmann. Além disso, a atualização de clusters reduz a autocorrelação das amostras, tornando o método mais eficiente próximo ao ponto crítico.

3.2.4 Geração das Amostras e Visualização

A simulação é executada por um total de $n_{\text{passos}} = 4000$ passos de Wolff. Para fins de visualização:

- A cada 30 passos, a configuração do sistema é armazenada para gerar uma animação da evolução dos *spins*.
- Após o término da simulação, as configurações selecionadas são exibidas como um *heatmap* e salvas como um GIF, ilustrando a evolução temporal dos domínios magnéticos.

A animação mostra claramente o processo de formação e crescimento de domínios de spins, característico do modelo de Ising próximo à região crítica.

3.2.5 Saídas do Simulador

Ao final da simulação, obtêm-se:

- Um conjunto de configurações independentes do modelo de Ising para serem utilizadas como dados de entrada para as RBMs e DBMs;
- Um arquivo *.gif* com a evolução temporal dos spins, útil para visualização qualitativa do processo dinâmico;
- Configurações finais armazenadas em formato *.npy*, facilitando o carregamento e posterior análise.

Dessa forma, o simulador implementado não apenas gera os dados necessários para o treinamento das redes de Boltzmann, mas também permite uma análise visual direta das dinâmicas do modelo de Ising.

3.3 Pré-processamento dos Dados para a DBM

As matrizes de spins foram binarizadas para adequação aos modelos RBM/DBM: valores $+1$ foram mapeados para 1, e -1 para 0. Cada configuração 128×128 foi transformada em um vetor unidimensional de dimensão 16384 (flattening), compondo o conjunto de dados de entrada para treinamento.

3.3.1 Arquitetura dos Modelos

Duas arquiteturas principais foram exploradas:

- **RBM**: composta por uma camada visível com 16384 neurônios e uma camada oculta com 1024 neurônios, para manter que sejam proporcionais ao tamanho da rede;
- **DBM**: composta por duas RBMs empilhadas, com camadas ocultas de 1024 e 64 neurônios. Novamente, mantendo uma correlação de proporcionalidade entre cada camada.

Todos os neurônios foram considerados binários estocásticos, ativados via função sigmoide seguida de amostragem Bernoulli.

3.3.2 Treinamento via Contraste de Divergência

O treinamento foi dividido em duas fases:

1. **Pré-treinamento camada por camada**, usando o algoritmo de Contrastive Divergence com $k = 5$ (passos Gibbs);
2. **Ajuste por amostragem**, utilizando cadeias de Gibbs para refinamento das distribuições.

Cada RBM foi treinada separadamente com otimizador Adam. As amostras reconstruídas foram obtidas via propagação $v \rightarrow h_1 \rightarrow h_2 \rightarrow h'_1 \rightarrow v'$.

3.3.3 Amostragem de Gibbs

A **amostragem de Gibbs** (Gibbs sampling) é um método de Monte Carlo via Cadeias de Markov (MCMC) utilizado para gerar amostras de uma distribuição de probabilidade multidimensional. As DBMs permite simular amostras da distribuição $P(v, h)$ por meio de uma cadeia iterativa que alterna atualizações entre camadas visíveis e ocultas.

Em cada passo:

- Atualiza-se **h** com base em **v**: $P(\mathbf{h} \mid \mathbf{v})$;
- Atualiza-se **v** com base em **h**: $P(\mathbf{v} \mid \mathbf{h})$.

Após várias iterações, as amostras convergem para a distribuição estacionária do modelo, sendo chamadas de *fantasy particles*. Essas amostras são utilizadas tanto para visualização quanto para validação qualitativa da distribuição aprendida.

Cada camada contém neurônios estocásticos binários, ativados pela função sigmoide:

$$P(h_j = 1 \mid \mathbf{v}) = \sigma \left(b_j + \sum_i v_i W_{ij} \right), \quad P(v_i = 1 \mid \mathbf{h}) = \sigma \left(a_i + \sum_j h_j W_{ij} \right),$$

onde W_{ij} são os pesos sinápticos entre as camadas, e a_i, b_j são os vieses das unidades visíveis e ocultas, respectivamente.

3.3.4 Treinamento Camada por Camada

Para contornar a dificuldade de otimização direta de uma DBM completa, foi aplicado um **pré-treinamento camada por camada**:

1. Treinamento da primeira RBM:

- Entrada: configurações reais do modelo de Ising;
- Objetivo: aprender a reconstruir os dados originais após $k = 5$ passos de Gibbs Sampling.

2. Treinamento da segunda RBM:

- Entrada: ativações ocultas \mathbf{h}' obtidas pela primeira RBM;
- Objetivo: aprender representações mais abstratas das correlações dos spins.

O processo de aprendizado usa o **Contrastive Divergence (CD- k)** para aproximar o gradiente da log-verossimilhança dos dados. O gradiente é estimado pela diferença entre as expectativas nos dados e no modelo:

$$\Delta W_{ij} \propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}.$$

A otimização dos pesos é realizada com o algoritmo **Adam**, que ajusta a taxa de aprendizado dinamicamente.

3.3.5 Gibbs Sampling e Reconstruções

O aprendizado é baseado em **Gibbs Sampling**, alternando amostragens entre camadas visíveis e ocultas. Para reconstruir uma configuração \mathbf{v} , o fluxo de propagação é:

$$v \rightarrow h_1 \rightarrow h_2 \rightarrow h'_1 \rightarrow v',$$

onde v' é a reconstrução do dado original após passar por toda a DBM.

3.3.6 *Fantasy Particles*

Após o treinamento, a DBM é capaz de gerar novas amostras sintéticas chamadas **fantasy particles**. O processo é:

1. Inicializar uma configuração aleatória de spins;
2. Executar **múltiplos passos de Gibbs Sampling** (~ 250 passos) alternando entre as camadas da DBM;
3. A configuração final obtida é uma amostra gerada pela distribuição aprendida pelo modelo.

Essas amostras podem ser comparadas com os dados reais para avaliar qualitativamente a capacidade do modelo de reproduzir padrões da física do sistema original.

3.3.7 *Fluxo Geral do Código*

O código segue a seguinte lógica:

1. Carrega os dados simulados do modelo de Ising em (por exemplo: $T = 2.3$);
2. Pré-processa os dados, achatando as configurações 128×128 em vetores binários de tamanho 16384;
3. Treina a primeira RBM por 500 épocas com Contrastive Divergence CD-5;
4. Extrai as ativações ocultas \mathbf{h}' e as utiliza para treinar a segunda RBM;
5. Após o treinamento, gera:
 - **Reconstruções**: propagação ida-e-volta dos dados reais;
 - **Fantasy Particles**: amostras puramente geradas pela DBM a partir de ruído;
6. Salva todos os modelos e resultados para análise posterior.

3.3.8 *Visualização*

Os resultados são apresentados como matrizes 128×128 :

- Linha 1: dados reais do modelo de Ising;
- Linha 2: reconstruções da DBM (após ida-e-volta);
- Linha 3: fantasy particles geradas após Gibbs Sampling.

Essa visualização permite comparar visualmente se as amostras geradas apresentam características estatísticas semelhantes às dos dados originais.

3.3.9 Saídas do Treinamento

Ao final do processo, obtêm-se:

- Pesos treinados das RBMs que compõem a DBM;
- Reconstruções dos dados originais;
- Fantasy particles representando a distribuição aprendida;
- Arquivos .npy com todos os resultados para futuras análises.

Dessa forma, a DBM atua como um modelo generativo probabilístico capaz de aprender e reproduzir a física estatística do modelo de Ising na região crítica.

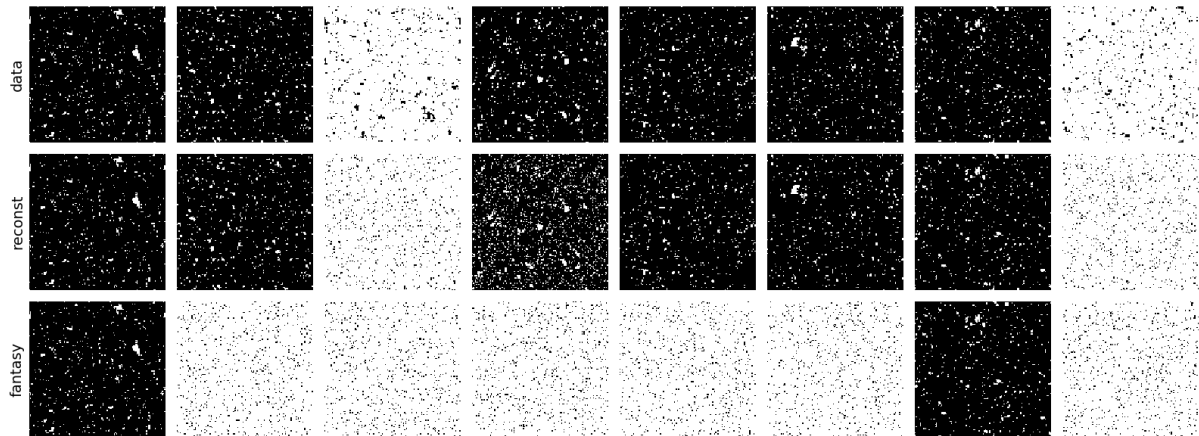
4 RESULTADOS

4.1 Resultados das DBMs

Nesta seção, são apresentados os resultados obtidos com o treinamento das Deep Boltzmann Machines (DBMs) sobre as amostras do modelo de Ising bidimensional nas regiões: Ordenada ($T = 2.0K$); Crítica ($T = 2.3K$); Desordenada ($T = 3.0K$). A análise foi realizada por meio da comparação visual entre os dados reais, gerados pelo simulador do modelo baseado no algoritmo de Wolff, as reconstruções obtidas após propagação e as amostras geradas pela DBM (denominadas *fantasy particles*) que geraram os dados sintéticos na linha inferior.

4.1.1 Comparação Visual das Configurações

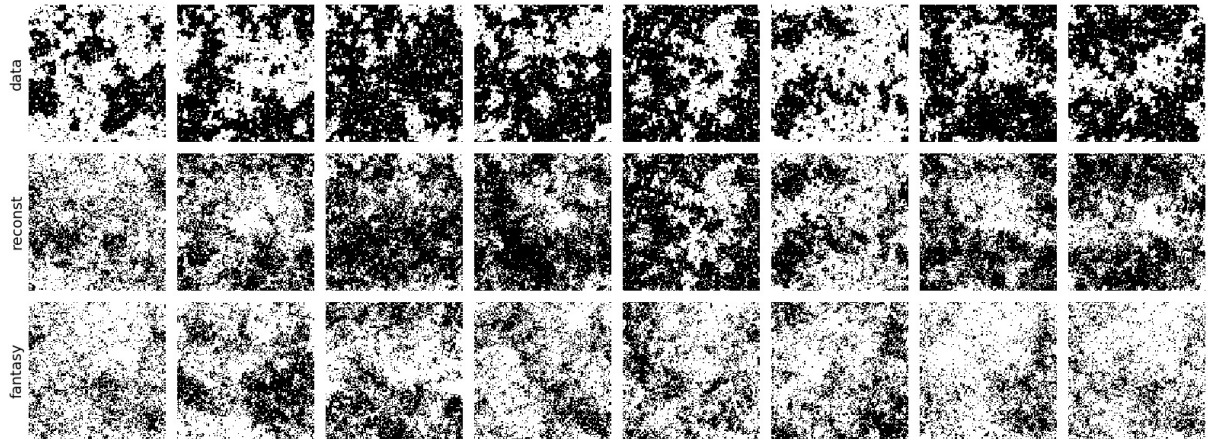
Figura 3 – Visualização comparativa dos dados reais (linha: **data**), reconstruções pela DBM (linha **reconst**) e amostras geradas (linha **fantasy**), todas na fase ordenada $T = 2.0K$



Fonte: Elaboração própria.

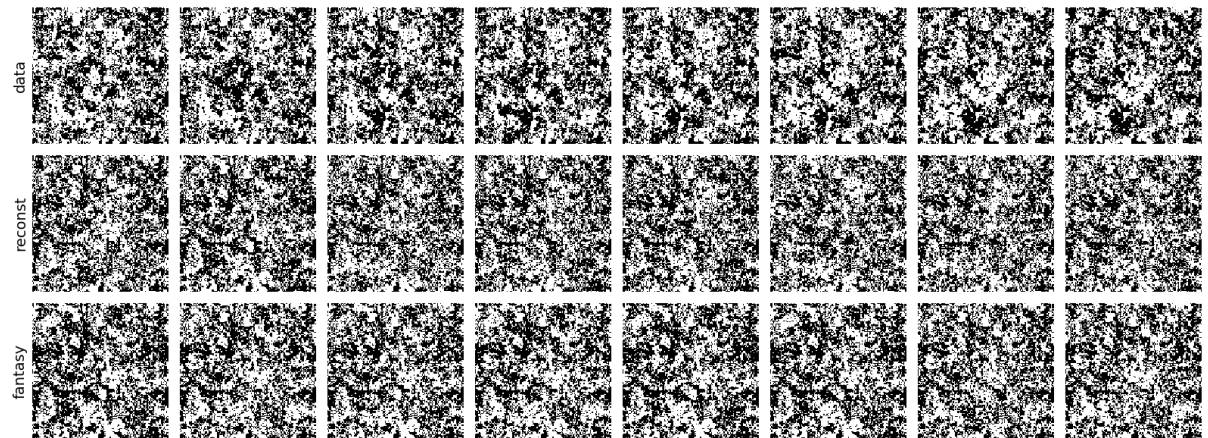
Na figura 4, observamos que: As **configurações reais** (linha superior) apresentam padrões típicos de flutuação crítica, com formação de domínios de spins positivos e negativos. As **reconstruções** (linha intermediária), obtidas após múltiplos passos de Gibbs sampling pela DBM, mantêm boa parte da estrutura original, embora com certo grau de suavização. As **partículas fantasia** (linha inferior), geradas a partir de ruído, exibem características semelhantes às amostras reais, sugerindo que a DBM conseguiu capturar a distribuição de probabilidade subjacente ao modelo de Ising crítico.

Figura 4 – Visualização comparativa dos dados reais (linha: **data**), reconstruções pela DBM (linha **reconst**) e amostras geradas (linha **fantasy**), todas na fase crítica $T = 2.3K$



Fonte: Elaboração própria.

Figura 5 – Visualização comparativa dos dados reais (linha: **data**), reconstruções pela DBM (linha **reconst**) e amostras geradas (linha **fantasy**), todas na fase desordenada $T = 3.0K$



Fonte: Elaboração própria.

4.2 Análise Qualitativa

Notavelmente, as Deep Boltzmann Machines (DBMs) capturaram, com diferentes graus de fidelidade, as características estatísticas das fases do modelo de Ising bidimensional. Tanto as reconstruções quanto as chamadas *fantasy particles* — amostras sintéticas geradas após múltiplos passos de Gibbs Sampling, as quais, apresentaram padrões compatíveis com a distribuição de Boltzmann esperada em cada regime térmico. Essa similaridade qualitativa sugere que a DBM internalizou aspectos relevantes da distribuição de probabilidade do sistema, mesmo em um espaço de alta dimensionalidade.

Na **fase ordenada**, observamos que as DBMs obtiveram grande êxito em replicar os padrões de cluster, cuja predominância era esperada pela baixa entropia, facilitando o aprendizado

em cada uma das imagens geradas pelas *fantasy particles*.

Na vizinhança da fase crítica observamos que o sistema apresenta flutuações de longo alcance e ausência de uma escala característica, a tarefa de modelagem torna-se particularmente desafiadora, pois a captura de padrões de clusters é acometida pela transição de fase. Ainda assim, a DBM mostrou-se capaz de gerar amostras que reproduzem, de forma qualitativa, domínios de spins e padrões correlacionados. Esse comportamento é coerente com observações feitas por Mehta et al., que relacionam a aprendizagem hierárquica de redes generativas profundas com processos de renormalização em sistemas físicos.

Na fase desordenada, o comportamento característico das DBMs foi similar ao da fase ordenada, isto é, replicação de um padrão de fase cujo estado físico a torna indistinguível, mesmo se compararmos todas as imagens umas com as outras, o que denota também a entropia em que o sistema está.

Outro aspecto relevante é a variabilidade do desempenho da DBM entre as diferentes fases do modelo. Em temperaturas muito baixas (fase ordenada) ou muito altas (fase desordenada), as amostras tendem a convergir para configurações homogêneas. Essa limitação pode refletir dificuldades na representação de distribuições com baixa entropia (altamente ordenadas) ou com ruído extremo (altamente aleatórias). Esses resultados indicam que, apesar do poder expressivo da DBM, sua arquitetura pode necessitar de ajustes específicos — como maior profundidade ou mais passos de amostragem — para lidar adequadamente com os extremos do espaço de fases.

Por fim, os resultados obtidos neste trabalho alinham-se com achados recentes da literatura, reforçando a viabilidade do uso de modelos generativos como ferramentas para o estudo de sistemas físicos complexos. Este estudo evidencia que o aprendizado de máquina, além de servir como uma técnica estatística poderosa, pode funcionar como uma ponte interpretativa entre a física computacional e os modelos probabilísticos hierárquicos.

5 CONCLUSÃO

Neste trabalho, exploramos a aplicação de modelos generativos baseados em energia, como as Máquinas de Boltzmann Restritas (RBMs) e Máquinas de Boltzmann Profundas (DBMs), para a modelagem de configurações do modelo de Ising bidimensional. Através das simulações computacionais conduzidas com o algoritmo de Wolff, geramos amostras representativas das diferentes fases do sistema; sendo elas: Ordenada, crítica e desordenada, a quais serviram como base para o treinamento dos modelos.

Os resultados demonstraram que tanto as RBMs quanto as DBMs são capazes de capturar padrões estruturais significativos presentes nas configurações de spins, especialmente na região crítica. As reconstruções e as *fantasy particles* geradas mostraram similaridade visual e estatística com as amostras reais, validando a capacidade dos modelos em aprender distribuições complexas oriundas de sistemas físicos.

Além disso, evidenciamos a utilidade do aprendizado não supervisionado e do algoritmo de *Gibbs Sampling* como ferramentas eficazes para a representação de distribuições de Boltzmann em redes neurais. A arquitetura profunda das DBMs mostrou-se particularmente eficaz na captura de correlações de longo alcance, o que pode ser vantajoso em estudos futuros sobre transições de fase e aprendizado de representações hierárquicas em física estatística.

Em resumo, o trabalho contribui para a interface entre física e aprendizado de máquina, demonstrando como métodos computacionais modernos podem ser aplicados com sucesso à modelagem de sistemas clássicos da mecânica estatística. Investigações futuras podem considerar a expansão para modelos quânticos, otimizações baseadas em aprendizado por reforço, ou ainda a análise quantitativa por medidas de entropia e correlação aprendidas pelas redes.

REFERÊNCIAS

- BAXTER, R. J. **Onsager and Kaufman's calculation of the spontaneous magnetization of the Ising model**. 2011. Disponível em: <https://arxiv.org/abs/1103.3347>. Acesso em: 10 jun. 2025.
- BHATTACHARJEE, S. M.; KHARE, A. **Fifty Years of the Exact Solution of the Two-Dimensional Ising Model by Onsager**. 1995. Disponível em: <https://arxiv.org/abs/cond-mat/9511003>. Acesso em: 11 jun. 2025.
- BRUSH, S. G. History of the lenz-ising model. **Reviews of Modern Physics**, APS, v. 39, n. 4, p. 883, 1967. Acesso em: 17 maio 2025.
- GHOJOGH, B.; GHODSI, A.; KARRAY, F.; CROWLEY, M. **Restricted Boltzmann Machine and Deep Belief Network: tutorial and survey**. 2021. Disponível em: <https://arxiv.org/abs/2107.12521>. Acesso em: 30 jun. 2025.
- HINTON, G. E.; SEJNOWSKI, T. J. Learning and relearning in boltzmann machines. **Parallel Distributed Processing (MIT Press)**, v. 1, p. 282–317, 1985. Acesso em: 17 maio 2025.
- HOLM, C.; JANKE, W. **Critical Exponents of the Classical 3D Heisenberg Model: a single-cluster monte carlo study**. 1993. Disponível em: <https://arxiv.org/abs/hep-lat/9301002>. Acesso em: 24 maio 2025.
- ISING, E. Beitrag zur theorie des ferromagnetismus. **Zeitschrift für Physik**, Springer, 1925. Acesso em: 17 maio 2025.
- IVANEYKO, D.; BERCHE, B.; HOLOVATCH, Y.; KENNA, R. **Criticality of the Random-Site Ising Model: metropolis, swendsen-wang and wolff monte carlo Algorithms**. 2006. Disponível em: https://www.researchgate.net/publication/1877490_Criticality_of_the_random-site_Ising_model_Metropolis_Swendsen-Wang_and_Wolff_Monte_Carlo_algorithms. Acesso em: 15 jun. 2025.
- KAYA, T. **An exact alternative solution method of 1D Ising model with Block-spin transformation at $H = 0$** . 2018. Disponível em: <https://arxiv.org/abs/1801.04452>. Acesso em: 25 maio 2025.
- Krauth, W. **Statistical Mechanics: algorithms and computations**. Oxford: Oxford University Press, 2006. 343 p. ISBN 978-0-19-851536-4.
- MEHTA, P.; BUKOV, M.; WANG, C. H.; DAY, A. G.; RICHARDON, C.; FISHER, C. K.; SCHWAB, D. J. **A high-bias, low-variance introduction to Machine Learning for physicists**. 2019. Disponível em: <https://arxiv.org/abs/1803.08823>. Acesso em: 21 jun. 2025.
- METROPOLIS, N.; W., R.; N., R.; H., T.; Teller, E. Equation of state calculations by fast computing machines. **Journal of Chemical Physics**, AIP Publishing, v. 21, n. 6, p. 1087–1092, 1953. Acesso em: 17 maio 2025.
- NISHIKAWA, Y.; MICHEL, M.; KRAUTH, W.; HUKUSHIMA, K. **Event-chain algorithm for the Heisenberg model: evidence for $z \simeq 1$ dynamic scaling**. 2015. Disponível em: <https://arxiv.org/abs/1508.05661>. Acesso em: 20 maio 2025.
- ONSAGER, L. Crystal statistics. i. a two-dimensional model with an order-disorder transition. **Physical Review**, APS, v. 65, n. 3–4, p. 117–149, 1944. Acesso em: 17 maio 2025.

PRADOS, A.; BREY, J. J. **Analytical solution of a one-dimensional Ising model with zero temperature dynamics**. 2001. Disponível em: <https://arxiv.org/abs/cond-mat/0106236>. Acesso em: 3 jun. 2025.

SHADRIN, A. V.; PANOV, Y. D. **Thermodynamic features of the 1D dilute Ising model in the external magnetic field**. 2023. Disponível em: <https://arxiv.org/abs/2301.11794>. Acesso em: 4 jun. 2025.

TÓTH, B. **An introduction to the theory of random processes and Monte Carlo methods**. 2009. Disponível em: <https://arxiv.org/abs/0905.1629>. Acesso em: 16 maio 2025.

WEI, R. Q. **An exact solution to the partition function of the finite-size Ising Model**. 2018. Disponível em: <https://arxiv.org/abs/1805.01366>. Acesso em: 29 maio 2025.

WOLFF, U. Collective monte carlo updating for spin systems. **Physical Review Letters**, APS, v. 62, n. 4, p. 361–364, 1989. Acesso em: 17 maio 2025.

ZHONG, W.; BARKEMA, G. T.; PANJA, D. **Super slowing down in the bond-diluted Ising model**. 2020. Disponível em: <https://arxiv.org/abs/2002.06079>. Acesso em: 18 jun. 2025.

APÊNDICE A – BIBLIOTECAS PYTHON

Bibliotecas utilizadas As bibliotecas abaixo foram utilizadas para a implementação, treinamento e visualização dos resultados:

- `numpy` — manipulação de arrays e dados numéricos;
- `torch`, `torch.nn`, `torch.nn.functional`, `torch.utils.data` — principais componentes da biblioteca PyTorch, usados para definir e treinar as redes neurais (RBMs e DBMs);
- `matplotlib.pyplot`, `matplotlib.gridspec`,
- `seaborn` — aprimoramento visual dos gráficos, especialmente mapas de calor.

APÊNDICE B – CONFIGURAÇÕES DO MODELO

Parâmetros da simulação A seguir estão listadas as principais configurações utilizadas na simulação e no treinamento dos modelos:

- **Tamanho da grade:** $L = 128$
- **Formato das imagens:** $(L \times L) = 128 \times 128$
- **Número de neurônios visíveis:** $n_{\text{visible}} = 16384$
- **Tamanho do batch:** 15
- **Número de épocas de treinamento:** 500
- **Passos de Gibbs (Contrastive Divergence):** 5
- **Passos de Gibbs para partículas fantasia:** 250
- **Número de amostras visualizadas:** 8
- **Tamanho das camadas ocultas (DBM):** 1024 e 64

Como continuidade deste trabalho, pretende-se disponibilizar o código-fonte completo em um repositório público no GitHub, permitindo reprodutibilidade e extensões futuras por parte da comunidade. <https://github.com/DaviFPassos>