

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA
TRABALHO DE CONCLUSÃO DE CURSO

Encontrando Transições de Fase a Partir da Incerteza da Regressão de Redes Neurais

Mateus Guimarães

Trabalho de Conclusão de curso apresentado como requisito parcial para obtenção do título de Bacharel em Física com ênfase em Astrofísica.

Orientador:
Prof. Dr. Heitor Carpes Marques Fernandes.

Porto Alegre, RS
Novembro de 2022

Guimarães, M.

Encontrando Transições de Fase a Partir da Incerteza da
Regressão de Redes Neurais / Mateus Guimarães. -- 2022.
56.

Orientador: Dr. Heitor Carpes Marques Fernandes.

Trabalho (Conclusão de curso) - Universidade Federal do
Rio Grande do Sul, Instituto de Física, Porto Alegre, BR-RS,
2022.

Modelo de Potts, Aprendizado de Máquina, Redes Neurais,
Transições de Fase I. Fernandes, Heitor C.M., orient.

Agradecimentos

Este trabalho é o resultado da dedicação e persistência não só minha, mas de todos que tiveram contato comigo durante a graduação, trilhei esse caminho com vocês.

Agradeço aos amigos que fiz durante o curso e, principalmente, os mais próximos. Muito obrigado Thiago Oliveira Ferreira, Pedro Henrique Mendes e Luis Gustavo Prates Cattelan, não teria chegado aqui sem vocês.

Ao professor Heitor pela paciência e suporte e conversas, não só durante a construção desse projeto, mas também nos cursos em que fui aluno durante a graduação.

Aos colegas de apartamento que tornaram mais leve e tranquila a vida fora do campus, muito obrigado João Pedro Menegott, Eduardo Godoy da Silveira e Bruno Wagner Dreger.

Agradeço a minha família também, que sempre me apoio e tornou possível a conclusão dessa etapa, mesmo estando longe vocês sempre estiveram comigo. Amo vocês!

Em especial a minha namorada, Marina, agradeço por todas as conversas e todo o apoio quando eu mais precisei, agradeço pelo incentivo a não desistir e pelos momentos felizes proporcionados que, com certeza, foram de suma importância nesta etapa final. Te amo!

Por fim, agradeço ao Instituto de Física da UFRGS pelo suporte e também aos amigos que aqui não foram citados.

“Se nada somos em tal mundo, sejamos tudo ó produtores.”

Resumo

No atual trabalho, utilizou-se de redes neurais (*ResNets*) para a obtenção da temperatura de sistemas a partir da configuração de spins na rede quadrada para o Modelo de Potts com q -estados tais que $q \in \{2, 3, 4, 6, 7, 8\}$. Fundamentou-se o uso de redes neurais para a solução do problema a partir das ferramentas do Aprendizado Estatístico em função da grande habilidade de generalização das redes neurais profundas; os passos e parâmetros para o treinamento de uma rede sob o entendimento do Aprendizado Supervisionado também são apresentados e discutidos. A partir da incerteza associada à regressão proposta na rede neural, pode-se encontrar as temperaturas de transição para todos os q -estados estudados, de forma que, todas as temperaturas de transição exatas estão contidas dentro do intervalo encontrado pelo método aplicado. Por fim, buscou-se ilustrar diferenças nas curvas das incertezas que pudessem caracterizar as diferentes ordens das transições presentes no Modelo de Potts, porém, não foi possível separar os dois tipos de transição com os resultados atuais; na tentativa de desvendar o funcionamento da rede residual, estudou-se os *feature maps* da camadas de convolução implementadas, porém tampouco foi possível embasar o funcionamento sob a ótica da física. Apresentou-se também, comentários sobre o Método de Monte Carlo utilizado nas simulações do Modelo de Potts e os algoritmos de evolução estudados: Swendsen-Wang e Metropolis; juntamente com um descritivo completo para a reprodução dos resultados.

Palavras-chave: Modelo de Potts, Aprendizado de Máquina, Redes Neurais, Transições de Fase

Abstract

The current work use neural networks (ResNets) to obtain the temperature of systems based on the spins configurations in the square lattice for the Potts Model with q -states such that $q \in \{2, 3, 4, 6, 7, 8\}$. We based the use of neural networks to solve the problem using the Statistical Learning framework due to the great generalization ability of deep neural networks; the steps and parameters for training a network under the perspective of the Supervised Learning are also presented and discussed. From the uncertainty associated with the proposed regression technique by neural network, one can find the transition temperatures for all studied q -states, which means that all exact transition temperatures are contained within the range of temperatures found by the applied method. Finally, we sought to illustrate differences in the uncertainty curves which could characterize the different orders of phase transitions in the Potts Model, however, it was not possible to separate the two types transition with current results; in an attempt to unravel the functioning of the residual neural network, we studied the *feature maps* of the convolutional layers, however it was not possible to base their performance on the perspective of physics. There is also some comments on the Monte Carlo Method used in the simulations of the Potts Model and comments on the Swendsen-Wang and Metropolis algorithms a complete description for the reproduction of the results is presented as well.

Keywords: Potts Model, Machine Learning, Neural Networks, Phase Transitions

Sumário

1	Introdução	1
2	Modelo de Potts e Transições de Fase	3
2.1	Modelo de Potts e a Mecânica Estatística	3
2.2	Transições de Fase	6
2.2.1	Transição de Primeira Ordem	8
2.2.2	Transição de Segunda Ordem	8
2.2.3	Caso Particular - Ferromagneto	8
3	Monte Carlo e Cadeias de Markov	11
3.1	Método de Monte Carlo e o Modelo de Potts	11
3.2	Algoritmo de Metropolis e Swendsen–Wang	12
4	Redes Neurais, Regressão e Classificação	15
4.1	Aprendizado de Máquina	15
4.1.1	Regressão e Aprendizado Supervisionado	17
4.2	Redes Neurais	18
4.2.1	Redes Convolucionais	21
4.2.2	ResNet e o Aprendizado Profundo	23
5	Metodologia, Desenvolvimento e Resultados	25
5.1	Metodologia	25
5.1.1	Simulações do Modelo de Potts 2D	25
5.1.2	Treinamento da Rede Neural	26
5.1.3	Pré-Processamento	26
5.1.4	Métricas de Análise	27
5.2	Resultados para o Modelo de Potts	28
5.3	Arquitetura da Rede	35
5.3.1	Exemplos de bloco <i>XResNet</i>	36
5.4	<i>Feature Maps</i> da Rede	37
6	Conclusão	41
6.1	Conclusão	41
6.1.1	Perspectivas Futuras	42
	Referências Bibliográficas	43

Capítulo 1

Introdução

Com o *boom* do aprendizado de máquina nos últimos anos [1], as mais variadas áreas da ciência apontaram os olhos com mais atenção para as ferramentas que estavam resolvendo problemas em aberto, tanto da física [2] como da biologia e da química [3] com algoritmos muitas vezes antigos ou já defasados, mas que, com atualizações e novas metodologias, poderiam acelerar descobertas [4] e inaugurar novas áreas da ciência [5]. Porém, não há como fugir, aprendizado de máquina é, e sempre será, estatística mesmo que com as ferramentas atuais [6] seja possível gerar modelos com “3 linhas de código”, compreende-se que para entender o funcionamento do modelo e como usá-lo da maneira correta irá exigir conhecimento da área. Na sobreposição da estatística e da física, o desenvolvimento da física da matéria condensada nos sistemas de spins se faz presente no estudo de materiais e superfícies na busca de encontrar e entender transições de fase nestes materiais.

Buscando solucionar problemas na intersecção da física da matéria condensada e do aprendizado de máquina, é coerente o estudo do aprendizado profundo, já que este encontra-se sob o escopo do aprendizado de representações [7], onde, fundamentalmente, o próprio algoritmo é o responsável por aprender as características do problema para resolvê-lo; como na física existem problemas em aberto [8] não explicados pelas teorias atuais, o uso do aprendizado profundo é entendível, já que talvez os algoritmos encontrem respostas para as perguntas que, atualmente, não podem ser respondidas.

Partindo dessa ideia, o atual trabalho buscou reproduzir a metodologia desenvolvida na referência [9] que faz uso de uma arquitetura específica de redes neurais: as redes residuais [10] para a obtenção da temperatura de um sistema de spins. Diferentemente da

referência, este trabalho é modelado sob a perspectiva do Modelo de Potts [11], para que, na incerteza da regressão aplicada pela rede residual, seja possível encontrar as temperaturas de transição que caracterizam as diferentes transições de fase do modelo de Potts [12]. A partir da simulação da evolução de sistemas de spins do modelo de Potts, pode-se estudar o funcionamento da matéria em condições bem definidas que, posteriormente, pode ser generalizada em modelos mais complexos que buscam representar fielmente a natureza da matéria, como por exemplos o Modelo Celular de Potts [13]. Para o problema abordado neste projeto, utilizou-se das configurações finais da rede quadrada do sistema como insumos para a rede neural, que por sua vez, tinha como problema resolver e “calcular” e temperatura desta configuração. Incentivado por [14], apresentou-se as motivações para o uso de redes neurais na solução deste tipo de problema a partir das ferramentas do aprendizado estatístico [15] e também interpretou-se os resultados das camadas de convolução da rede residual aplicada, com objetivo de relacionar os resultados da rede com aspectos físicos que possam explicar o funcionamento da rede a partir de bases físicas [16].

Na disposição deste trabalho, organizam-se comentários sobre a formulação estatística do Modelo de Potts sob a perspectiva da Mecânica Estatística e as definições de Transições de Fase (Cap. **2**). Discute-se brevemente o algoritmo de Metropolis-Hastings [17, 18] e de Swendsen-Wang [19] e suas características essenciais (Cap. **3**). São apresentadas também as técnicas utilizadas para construção da rede neural utilizada e os métodos pelos quais esta rede foi treinada, discutindo também as métricas de análise e a fundamentação teórica sobre o treinamento de algoritmos supervisionados (Cap. **4**). Na parte final são apresentados os resultados encontrados no trabalho, juntamente com comentários e discussões sobre o tema e perspectivas futuras (Cap. **5 e 6**).

Capítulo 2

Modelo de Potts e Transições de Fase

Neste capítulo serão, brevemente, discutidas relações e características do Modelo de Potts e a Mecânica Estatística, fazendo comentários sobre a função de Partição \mathcal{Z} e alguns detalhes da simulação do modelo são apresentados. Por fim, apresentam-se as transições de fase, explicando sua relação com a função de energia livre \mathcal{F} e as definições acerca das transições de primeira ordem e transições de segunda ordem.

2.1 Modelo de Potts e a Mecânica Estatística

Originado de uma defesa de doutorado do matemático australiano Renfrey B. Potts [11] na década de 1950, o Modelo de Potts ainda é muito estudado atualmente pois consegue simular o comportamento de sistemas complexos em função de propriedades internas do sistema, possuindo aplicações da biologia à sociologia como em [20, 21]. Este modelo, em função de suas características, se enquadra na abordagem dos problemas associados às áreas da Mecânica Estatística e da Física da Matéria Condensada, uma vez que consideramos uma rede quadrada¹ de elementos, *Square Lattice*, com interação entre vizinhos próximos, *Nearest Neighbours Interaction* ou interação de primeiros vizinhos. Por se tratar de um problema, frequentemente, estudado pela Mecânica Estatística, se faz usual a abordagem a partir de um Hamiltoniano \mathcal{H} que descreverá o comportamento da energia do sistema em questão favorecendo² os cálculos necessários.

¹Existem outras classes de redes que podem ser usadas, mas o atual trabalho é desenvolvido apenas para redes quadradas.

²A formulação de problemas a partir do Hamiltoniano, mesmo sendo mais densa conceitualmente, pode encurtar algumas definições.

Antes da apresentação do Hamiltoniano do sistema, começamos definindo os “estados” do Modelo de Potts. Quando fala-se do número de estados do Modelo de Potts, temos em questão a variável “ q ” dando origem ao nome mais comum do problema abordado neste trabalho: *Q-State Potts Model*. Examina-se o sistema de forma que, em cada vértice da rede quadrada está posicionado um spin em um dos q estados possíveis. Para facilitar o entendimento, pode-se considerar que cada q estado possível é uma direção no \mathbb{R}^n com $n = q$ tal que, todas as direções possíveis são ortogonais entre si³. Dada esta primeira formulação, é conveniente a apresentação do Hamiltoniano do sistema comentado anteriormente em função dos agora apresentados q estados do sistema juntamente com a formulação do sistema na rede quadrada, segue-se então:

$$\mathcal{H} = -J \sum_{\{i,j\} \in S} \delta_{\sigma_i \sigma_j} . \quad (2.1)$$

Em (2.1) temos que: J é uma constante para o fator de acoplamento dos spins; S é a rede quadrada tal que um conjunto $\{i, j\} \in S$ é o mesmo que um elemento na posição i, j desta mesma matriz S ; $\delta_{\sigma_i \sigma_j}$ nos mostra qual será a interação entre os spins σ_i e σ_j adotada na rede, ou seja:

$$\delta_{\sigma_i \sigma_j} = \begin{cases} 1, \text{ caso } \sigma_i = \sigma_j \\ 0, \text{ caso } \sigma_i \neq \sigma_j \end{cases} . \quad (2.2)$$

É importante notar que, como estamos modelando o problema com interações de vizinhos próximos, para cada spin $\{i, j\}$ da rede quadrada temos 4 vizinhos próximos que serão contabilizados para o Hamiltoniano dados por $\{i + 1, j\}$, $\{i - 1, j\}$, $\{i, j + 1\}$ e $\{i, j - 1\}$. Porém, quando calculamos o Hamiltoniano para o spin $\{i, j + 1\}$, temos dois vizinhos *compartilhados* com o spin $\{i, j\}$ e com isso acabamos com termos repetidos na soma. Corrige-se esse problema com uma divisão por 2 no valor final da energia calculada.

A expressão (2.1) com a definição de (2.2) nos apresenta um Hamiltoniano que favorece o alinhamento⁴ de spins, uma vez que, spins vizinhos em um mesmo estado q diminuem a energia do sistema e spins vizinhos com diferentes estados aumentam a energia. Despretensiosamente⁵, pode-se relacionar a temperatura T do sistema com a energia calculada pelo Hamiltoniano, com isso, é, visivelmente, perceptível a ação do

³A partir deste ponto, tratar-se-à de spins e vértices de maneira intermitente, ou seja, ambos poderão possuir um estado q associado.

⁴Alinhamento aqui entende-se como spins vizinhos de um mesmo q -estado.

⁵A relação real é mais complexa, mas neste ponto apenas a ideia de que uma maior energia está relacionado a uma maior temperatura já é suficiente.

Hamiltoniano na figura 2.1, onde temos configurações para a simulação do modelo de Potts com $q = 6$ para três diferentes temperaturas: 0.49 (esquerda), 0.81 (centro) e 1.14 (direita).

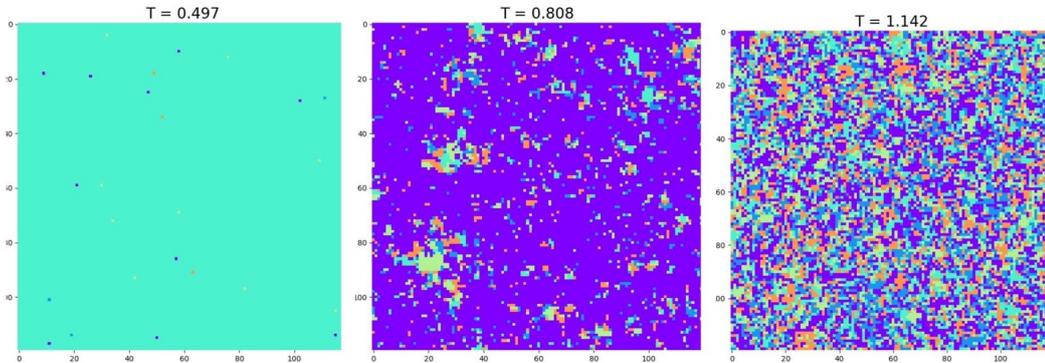


FIGURA 2.1. Configurações da rede quadrada para o modelo de Potts com $q = 6$ e temperaturas $T=0.49$, 0.81 , 1.42 , respectivamente.

No contexto da Mecânica Estatística, resolver o problema do Modelo de Potts se enquadra na solução de problemas no *ensemble canônico* já que vamos tratar a evolução do sistema considerando: uma temperatura T , o número de partículas N e um volume V fixos durante todo o processo [22]. Deste modo, pode-se introduzir a função de partição que será a responsável pela descrição estatística do sistema em questão tal que:

$$\mathcal{Z}(T, H) = \sum_i e^{-\beta E_i} . \quad (2.3)$$

Onde, em (2.3), temos⁶ $\beta = 1/k_B T$ com k_B *constante de Boltzmann* e E_i sendo a energia associada a um micro-estado⁷ i ; fazendo com que seja possível a relação entre o somatório de todos os q^N micro-estados possíveis⁸ e uma temperatura T .

Como a função de partição canônica nos apresenta a descrição estatística fundamental de um sistema, pode-se escrever outras grandezas termodinâmicas em função de

⁶Na equação em questão, H é o fator associado ao campo magnético externo do sistema e mesmo muitas vezes sendo considerado igual a zero, costuma-se deduzir as expressões e funções do sistema em função do campo magnético uma vez que o $\lim H \rightarrow 0$ pode ser tomado posteriormente.

⁷Aqui, entende-se por micro-estado a configuração dos N spins da rede S em função dos q possíveis estados, como na figura 2.1 e sua energia pode ser calculado pelo Hamiltoniano da expressão (2.1).

⁸Para a rede quadrada $N = L^2$, logo todos os estados possíveis ficam em função do tamanho da rede L e do total de q -estados.

\mathcal{Z} , como a energia livre (\mathcal{F}) e a energia interna (U):

$$\mathcal{F}(T, H) = -k_B T \ln(\mathcal{Z}) , \quad (2.4)$$

$$U = -\frac{\partial}{\partial \beta} \ln(\mathcal{Z}) . \quad (2.5)$$

Onde, em (2.4), \ln é o logaritmo natural (logaritmo na base e), T é a temperatura do sistema e k_B é novamente a constante de Boltzmann, já em (2.5) $\beta = 1/k_B T$.

É muito comum na literatura o estudo de outras características do sistema atreladas a função de energia livre \mathcal{F} , como por exemplo:

$$M = -\left(\frac{\partial \mathcal{F}}{\partial H}\right)_T , \quad (2.6)$$

$$S = -\left(\frac{\partial \mathcal{F}}{\partial T}\right)_H , \quad (2.7)$$

$$P = -\left(\frac{\partial \mathcal{F}}{\partial V}\right)_T \quad (2.8)$$

onde M , S e P são magnetização, entropia e pressão do sistema, respectivamente.

E também, a partir destas características, tomam-se novos observáveis tal como: Calor específico,

$$\mathbb{C}_X = T \left(\frac{\partial S}{\partial T}\right)_X \quad \text{onde } X = H, M . \quad (2.9)$$

Susceptibilidade magnética ou isotérmica,

$$\chi_T = \left(\frac{\partial M}{\partial H}\right)_T . \quad (2.10)$$

2.2 Transições de Fase

Uma transição de fase é uma mudança, brusca ou não, nas características físicas de um meio com a variação de um parâmetro - por exemplo, a mudança brusca de densidade quando um líquido se torna um gás com o aumento de temperatura. Uma maneira de caracterizar estas transições entre estados do sistema é a partir de observáveis já conhecidos que sejam atrelados a uma característica intrínseca do meio. Tais observáveis recebem o nome de *parâmetros de ordem*. Uma transição de fase nem sempre está associada a

uma mudança de estado físico (de sólido para líquidos, por exemplo) e também não se faz necessária a variação brusca de um observável, fazendo com que a transição não seja imediatamente visível - por exemplo, a magnetização de um ferromagnético. Porém, é estritamente necessária a “capacidade” de se medir esta transição, uma vez que, não sendo visível e não se podendo medi-la, não seria possível encontrar qualquer tipo de transição.

Mais precisamente, uma transição de fase é caracterizada por uma singularidade em um potencial termodinâmico do sistema [23], a energia livre (2.4) é apenas um comum exemplo de um potencial termodinâmico de sistemas que, ao apresentar singularidades, estão atreladas transições de fase.

Um exemplo do comportamento de um observável pode ser visto na figura 2.2, onde temos a série temporal da energia interna do modelo de Potts com $q = 10$ (a) e $q = 4$ (b).

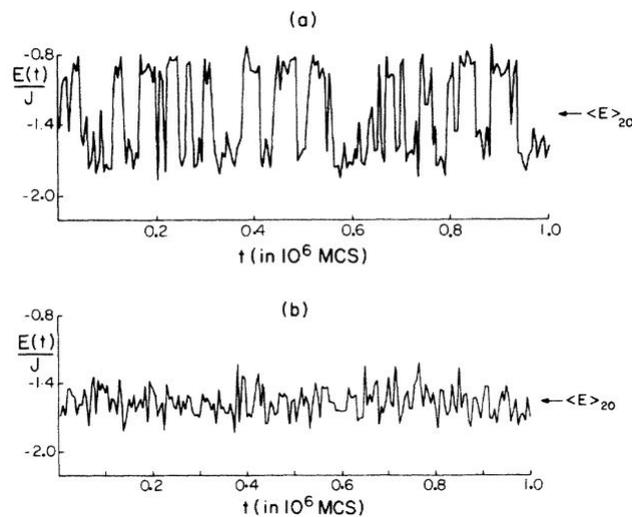


FIGURA 2.2. Flutuações na série temporal da energia interna (2.5) para o modelo de Potts com $q = 10$ (a) e $q = 4$ (b). Retirado de [24].

Acerca dos parâmetros de ordem, é importante notar que existem diversos exemplos de parâmetros de ordem associados às mais diversas áreas, como por exemplo: em um sistema ferromagnético, a magnetização é zero na fase desordenada e possui algum valor positivo na fase ordenada; em sistemas gás-líquido, o negativo da diferença de densidade entre a fase líquida e a fase gasosa é zero na fase líquida, e algum valor positivo na fase gasosa; já em sistemas de cristais líquidos, o grau da ordem de orientação.

Pode-se então, classificar as transições de fase em função da natureza da derivada da

energia livre do sistema. Seguindo a notação de Ehrenfest, temos: transições de primeira ordem e transições de segunda ordem⁹.

2.2.1 Transição de Primeira Ordem

Uma transição de primeira ordem é definida a partir de uma descontinuidade em uma função representada por derivadas primeiras da função de energia livre do sistema, ou seja, se um observável calculado a partir de uma derivada *primeira* da função de energia livre do sistema apresentar uma descontinuidade, define-se uma transição de fase de primeira ordem neste exato ponto onde encontrou-se a descontinuidade.

2.2.2 Transição de Segunda Ordem

Uma transição de segunda ordem está relacionada com descontinuidades em derivadas segundas da função de energia livre do sistema, ou seja, ao invés de apresentar descontinuidades nas primeiras derivadas, as transições de fase de segunda ordem apresentam descontinuidades em relação a derivadas *segundas* da função de energia livre.

2.2.3 Caso Particular - Ferromagneto

Aplicando os conceitos apresentados sobre as transições de fase para o caso de um ferromagneto, analisam-se os observáveis existentes com relação a função de energia livre (2.4) tal que, para observáveis têm-se: a magnetização (2.6), calor específico (2.9) e a suscetibilidade magnética (2.10). Com as definições de cada observável apresentadas, estabelece-se as seguintes afirmações: uma descontinuidade na magnetização caracterizaria uma transição de primeira ordem, já que, pela expressão que define este observável temos uma derivada primeira da função de energia livre (2.4) do sistema; já, ao encontrarmos uma descontinuidade no calor específico ou na suscetibilidade magnética do sistema, a transição associada ao evento seria de segunda ordem (ou uma transição contínua), já que os observáveis estão relacionados a uma derivada segunda da função de energia livre

⁹Em [25] é introduzida a terminologia de transições de primeira ordem e transições contínuas (para as transições de segunda ordem e/ou mais altas).

dos sistema. O Modelo de Potts é um exemplo de um modelo ferromagnético e apresenta dois [12] tipos de transições de fase em função do número de estados q dado por:

$$\text{Transição} = \begin{cases} \text{Segunda Ordem, caso } q \leq 4 \\ \text{Primeira Ordem, caso } q > 4 \end{cases} . \quad (2.11)$$

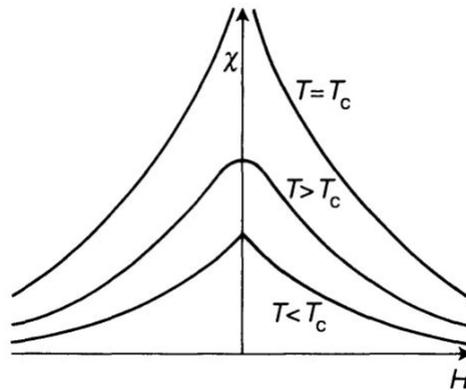


FIGURA 2.3. Gráfico da susceptibilidade magnética (2.10) em função do campo externo H mostrando a descontinuidade na temperatura crítica T_C para o Modelo de Potts com $q = 2$. Retirado de [23].

Na figura 2.3, podemos notar a descontinuidade na susceptibilidade magnética (2.10) do Modelo de Potts com $q = 2$, configurando uma transição de segunda ordem na temperatura de transição.

Capítulo 3

Monte Carlo e Cadeias de Markov

Neste capítulo serão introduzidos os conceitos básicos sobre as simulações utilizando do Método de Monte Carlo para amostragem de grandezas e suas utilidades e benefícios para as simulações de sistemas físicos. Também serão apresentados dois algoritmos de amostragem muito usuais, o algoritmo de Metropolis-Hastings e o algoritmo de Swendsen-Wang. Suas diferenças são mencionadas.

3.1 Método de Monte Carlo e o Modelo de Potts

O método de Monte Carlo baseia-se na teoria Estatística da Inferência, na qual, estimam-se valores atrelados a uma população a partir de uma amostra desta população, já que, muitas vezes o acesso a esta população é impossível ou muito trabalhoso. No contexto deste trabalho, gerar todos q^N estados possíveis é inviável¹ e, em razão disso, optou-se pela estimação dos observáveis a partir do método de Monte Carlo.

Por se tratar de uma estimativa, busca-se estimar sempre as melhores medidas possíveis no sentido de que atrelamos uma espécie de “confiança” conforme diminui-se a variância destas medidas, ou seja, vinculamos melhores estimativas a médias com menor variância [26]. Em função disso, como estamos auferindo medidas de uma amostra oriunda de uma população, metaforicamente, inacessível, quanto maior for a amostra utilizada, melhor serão as estimativas, no sentido de que, ao igualarmos amostra e população, temos a melhor estimativa possível que é exatamente igual a medida real da população.

¹Aqui temos $N = L^2$ com L sendo o tamanho da rede (número de sítios) adotada

No contexto do Modelo de Potts, enfrenta-se um problema na tentativa de calcular a função de partição (2.3), uma vez que o espaço de estados atrelados a cada energia E_i é extremamente complexo e mensurar a função em cada um destes estados é impraticável - mais especificamente, o problema está em calcularmos a função de partição para o caso de redes finitas, uma vez que [27] apresenta a solução para o caso infinito. Opta-se por utilizar do método de Monte Carlo para traçar estimativas de medidas, como por exemplo a magnetização (2.6), que podem, de certa forma, serem relacionadas com a função de partição do problema. No ambiente das simulações de problemas físicos, como a simulação do Modelo de Potts, o método de Monte Carlo é aplicado no escopo da amostragem da distribuição de Boltzmann do problema tal que os valores utilizados na simulação são na verdade os estados de energia E_i e, ao amostrarmos estes valores a partir desta distribuição, estamos acessando estados atrelados a estas energias E_i no espaço de estados do problema, ou seja, amostramos os estados a partir das energias. Pode-se considerar que, numa simulação física nos moldes citados anteriormente, a amostragem pelo método de Monte Carlo se dá quando o sistema “passeia”² pelo espaço de fases³ do sistema a partir da amostragem por importância de uma distribuição e com isso, estimam-se medidas dos observáveis sem a necessidade de “visitar” ou “passar” por todos os *pontos* no espaço de fase.

3.2 Algoritmo de Metropolis e Swendsen–Wang

Como o método de Monte Carlo é uma ferramenta da Estatística Inferencial baseada, majoritariamente, em Processos Estocásticos e Cadeias de Markov, é de se esperar que alguns critérios da probabilidade vinculada às simulações precisem ser seguidos e respeitados durante uma amostragem. Dentre os critérios necessários, sobressaem-se o Balanço Detalhado e a Hipótese da Ergodicidade [29, 30]. O balanço detalhado nos garante que, no equilíbrio, a probabilidade da passagem do sistema de um estado A para outro estado B deve ser a mesma da passagem do sistema do estado B para o estado A, endossando a reversibilidade do “passeio” intrínseco a cadeias de Markov. Consideramos também a hipótese da ergodicidade quando leva-se em consideração a capacidade de, para

²É importante ressaltar que a visitação dos estado segue uma cadeia de Markov [28].

³Espaço de Fases e Espaço de Estados podem ser usados coincidentemente sem perda de generalização.

um longo período⁴, visitarmos todos os estados, ou seja, com um longo período, todos os estados devem ter uma probabilidade de “visita” diferente de zero.

Desenvolvido na década de 50, o algoritmo⁵ de Metropolis-Hastings[17, 18] (*single spin*) revoluciona a amostragem em simulações quando implementa a geração de novos valores para a amostra a partir apenas do valor gerado anteriormente. Como a dependência do próximo valor depende apenas do valor anterior, aborda-se esse algoritmo como uma cadeia de Markov onde os estados são amostrados a partir da distribuição de Boltzmann⁶. Já o algoritmo de Swendsen-Wang [19] (*cluster*) explora a ideia de que, atualizar não só um único spin como o algoritmo de Metropolis, mas sim um grupo coerente de spins (os clusters), decorrelacionando temporalmente as medidas geradas pela cadeia de Markov e aumentando a eficiência estatística do algoritmo próximo a temperatura crítica (T_C). Com isso, torna razoável seu uso para as simulações que estudam regiões próximas a transições de fase; em contra partida, a complexidade do algoritmo de Metropolis é menor que a do algoritmo de Swendesen-Wang e a decisão de qual algoritmo usar não é tão simples [31]. A implementação do *pseudo-código* de ambos os algoritmos pode ser encontrada na referência [12].

⁴Aqui entende-se por “período” o tempo de passeio que uma simulação teria; pode-se entender também como o número de passos simulados dentro do espaço de fase do problema.

⁵Os resultados de Metropolis na década de 1950 influenciaram Hastings a formular o que conhecemos hoje por “Metropolis-Hastings”.

⁶Com isso, não depende-se mais da normalização da função de probabilidade do problema mas apenas de uma função proporcional à esta, fazendo com que não seja mais necessário o conhecimento da função de partição Z .

Capítulo 4

Redes Neurais, Regressão e Classificação

Neste capítulo encontram-se os comentários acerca do Aprendizado de Máquina no contexto deste projeto, ilustrando as características intrínsecas do trabalho que o caracterizam como um problema do Aprendizado Supervisionado sob a perspectiva do Aprendizado Estatístico. Apresenta-se também as definições iniciais de uma rede neural a partir do *Perceptron*, seguindo desta definição discorre-se a respeito dos *Perceptrons Multicamadas* e das camadas de convolução, apresentando ainda comentários sobre as arquitetura escolhida e o Aprendizado Profundo.

4.1 Aprendizado de Máquina

No escopo do aprendizado de máquina (*Machine Learning, ML*), o objetivo deste trabalho enquadra-se nos problemas de generalização pelo aprendizado supervisionado; problemas nos quais temos dados com a classificação correta definida e a intenção é classificar novos dados¹ Uma vez que temos como objetivo primário classificar configurações de sistemas de spins em determinadas temperaturas estamos tratando de um problema de regressão, no sentido de que vamos buscar relacionar variáveis independentes (spins) à uma resposta dependente destas variáveis, a temperatura do sistema [32].

Para o caso da classificação do sistema de spins na rede quadrada, particularmente,

¹Diferentemente dos problemas do aprendizado não-supervisionado, onde temos os dados, porém não temos a classificação destes dados e a intenção, majoritariamente, é agrupar estes dados para definir uma classificação.

temos L^2 spins que podem tomar q estados diferentes resultando então em q^N combinações² possíveis³ para a rede S . Buscamos uma função f tal que:

$$f(s_i) = T_i. \quad (4.1)$$

Em (4.1) temos que s_i é uma das q^N combinações possíveis para a rede S , ou seja, queremos ter uma função arbitrária que ao receber uma configuração de spins apresente como resultado uma temperatura (T_i) associada a esta configuração.

No contexto do modelo abordado, usualmente, define-se uma distribuição de energias que seja coerente com a função de partição \mathcal{Z} a partir da distribuição de Boltzmann para uma temperatura T , sendo possível então, evoluir o sistema calculando essas energias “configuracionais”⁴ esperando que a distribuição final respeite a amostragem de Boltzmann. A tarefa proposta pelo atual projeto é o inverso do usual: são amostradas configurações de spins, a partir de (4.2), e o objetivo é encontrarmos a temperatura com o maior peso de Boltzmann atrelado, ou seja, a temperatura mais provável para cada configuração.

$$\mathcal{D}_T \sim \frac{e^{-\mathcal{H}/T}}{\mathcal{Z}}. \quad (4.2)$$

Em (4.2), \mathcal{D}_T é a distribuição canônica das energias do conjunto das q^N combinações possíveis para uma temperatura T . Por se tratar de um problema expressamente probabilístico, deve-se notar que: dois conjuntos \mathcal{D}_{T_1} e \mathcal{D}_{T_2} , para $T_1 \neq T_2$, podem compartilhar uma combinação s_i uma vez que esta combinação pode representar a rede em duas temperaturas diferentes, porém as probabilidades da rede S ser representada pelo estado s_i é diferente para T_1 e T_2 , fazendo com que uma das temperaturas tenha uma maior probabilidade associada.

O aprendizado de máquina consiste em definirmos a melhor função $f \in \mathcal{F}$ encontrada no processo de minimizar o risco empírico [33]. Nos fundamentos do Aprendizado Estatístico [15], usa-se a notação de três possíveis funções: $t(x)$, $f^*(x)$ e $f_n(x)$ tal que, $t(x)$ é a melhor função em teoria e está associada ao conhecimento de toda a distribuição possível dos dados - no contexto deste trabalho, implicaria o conhecimento e uso de todas as q^N combinações; $f^*(x)$ é a melhor função da classe, uma vez que estamos limitados

²Para a rede quadrada, $N = L^2$.

³Da mesma maneira que foi abordado em 6.1, o conjunto que contém todas as combinações de estados de um sistema físico no âmbito da mecânica estatística é também chamado de Ω .

⁴Durante a simulação, a energia é calculada a partir da configuração dos spins e seus vizinhos.

a um conjunto finito \mathcal{F} de funções possíveis⁵ e esta função seria então a melhor função dentro deste conjunto definido. Já $f_n(x)$ seria a melhor função na prática, encontrada ao minimizar o risco empírico, uma vez que este risco está associado a um erro de estimativa, que por sua vez, pode ser interpretado como uma distância entre f_n e f^* ambos pertencentes a \mathcal{F} . Aplicando essas ferramentas no contexto do atual trabalho, buscamos minimizar o erro empírico da função $f(s_i)$ apresentada em (4.1) pela abordagem de redes neurais dado que \mathcal{F} seria definido pelas características da rede neural utilizada e o melhor cenário possível seria encontrarmos $f^*(s_i)$, sabendo que, encontrar uma função $t(s_i)$ não é possível visto que não serão utilizadas todas as q^N combinações.

4.1.1 Regressão e Aprendizado Supervisionado

Como apresentado em 4.1, o problema abordado neste trabalho enquadra-se nas características do aprendizado supervisionado e, de acordo com [34, 35], podemos encaixar a resolução do problema sob a perspectiva de 3 componentes: o ambiente, o professor e o aprendiz⁶. O ambiente é o encarregado de suprir os dados utilizados a partir de uma distribuição de probabilidade fixa, mas desconhecida (no contexto específico do trabalho, essa distribuição é dada por (4.2) e os dados seriam o conjunto dos $s_i \mid i \in \{1, \dots, q^N\}$ como em (4.1)); O professor funciona na ideia de apresentar a classificação correta d de uma quantidade arbitrária de s_i a partir de uma função F tal que:

$$d_i = F(s_i) \quad . \quad (4.3)$$

Onde, em (4.3), d_i é a resposta correta⁷ para a configuração de spins s_i . Já ao aprendiz é atribuída a concepção de realizar relações *input-output* no contexto de que, ao receber uma configuração s_i devolva a temperatura T_i com maior probabilidade (como por exemplo, a equação (4.1)). Porém, (4.1) não é suficiente para resolvermos o problema uma vez que não há nada a ser modelado nesta expressão e seu funcionamento nos moldes atuais assemelha-se ao do *professor*. A essência do aprendiz está relacionada ao algoritmo que

⁵Aqui entende-se essa limitação no sentido de classes de funções estudadas na tentativa de resolver o problema, como por exemplo, ao considerarmos funções do tipo $f_1(x) = ax + b$ e $f_2 = ax^2 + bx + c$ cada classe \mathcal{F}_1 e \mathcal{F}_2 teria uma função f_1^* e f_2^* associada.

⁶Diferentemente da tradução literal de *learning machine* optou-se por utilizar “aprendiz” como componente responsável pelo aprendizado.

⁷Aqui, novamente, entende-se por “resposta correta” a temperatura que possua a maior probabilidade de apresentar o estado s_i como salientado em 4.1

será utilizado para o aprendizado e, com este algoritmo definido, teríamos parâmetros⁸ \mathbf{w} a serem modelados no processo de aprendizado tal que, ao reformulamos a expressão (4.1), temos:

$$y = f(s_i, \mathbf{w}) \quad . \quad (4.4)$$

Onde, em (4.4), y seria a temperatura devolvida pela nova função f associada ao aprendiz já com os parâmetros \mathbf{w} definidos. Com as três componentes definidas, a tarefa de encontrarmos a melhor função na prática $f_n(x)$ sem uma métrica de controle seria como passear no escuro; contrariando este passeio, definimos a chamada função de custo \mathcal{L} tal que:

$$\mathcal{L} = \mathcal{L}(d, y) \quad \therefore \quad \mathcal{L}(F(s_i), f(s_i, \mathbf{w})) \quad . \quad (4.5)$$

Onde, em (4.5), d seria a temperatura correta e y a resposta devolvida pelo aprendiz para uma configuração s_i qualquer⁹. Para facilitar a busca pela melhor função na prática $f_n(x)$ no escopo do trabalho, durante a fase de treinamento (Por “fase de treinamento” entende-se o período no qual os dados são divididos em conjuntos de treino e teste para evitar *overfitting*¹⁰ e a perda de generalização da função) a métrica usada será a minimização da função de custo, que por sua vez, está definida em função dos parâmetros \mathbf{w} fazendo com que o caminho a ser seguido em busca da melhor função se dê em função dos parâmetros intrínsecos do aprendiz escolhido.

4.2 Redes Neurais

Em função das características e premissas do atual projeto, utilizar o ferramental das redes neurais para resolver o problema se faz necessário uma vez que, redes neurais artificiais (*Artificial Neural Networks, ANN ou NN*) precisam de grandes conjuntos de dados mas detêm melhores estatísticas de generalização em dados novos quando comparadas a outros algoritmos de aprendizagem [36] (como por exemplo: Florestas Aleatórias [37] e Máquinas de Vetores de Suporte [38]). Desenvolvidas na ideia de copiarmos o fun-

⁸Utiliza-se a expressão “parâmetros” na ideia de que, devido ao leque de algoritmos possíveis a seres escolhidos para o aprendizado, todos podem ser representados por, no mínimo, um parâmetro \mathbf{w}

⁹É importante notar que a expressão de \mathcal{L} costuma ser definida em função das características das respostas y e d de cada problema, seja ele de regressão ou de classificação.

¹⁰Termo relacionado aos casos onde a função encontrada apresenta resultados ótimos para os dados de treinamento e resultados muito inferiores para os dados de teste, caracterizando a perda de generalização da função e claro sobre-ajuste.

cionamento do nosso cérebro para a resolução de problemas através do computador, o desenvolvimento de novas teorias no assunto sempre esteve atrelado a tentativa de reproduzir o que acontece no corpo humano em máquinas. Com sua arquitetura básica formada por “neurônios”, as redes neurais conseguem resolver muitos problemas em função das características não-lineares¹¹ destes neurônios e seu uso para a resolução de problemas físicos é cada vez mais comum [39].

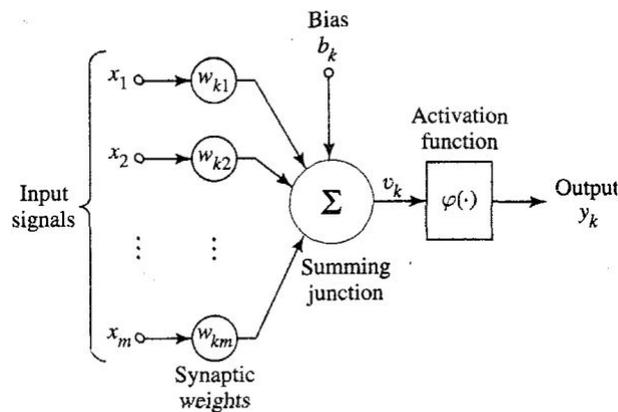


FIGURA 4.1. Estrutura do Perceptron. Retirado de [35].

Pode-se encontrar três características básicas atreladas aos neurônios que compõem uma rede neural, são estas: os pesos *sinápticos*, função de soma e função de ativação, um exemplo é apresentado na figura 4.1. O funcionamento de cada neurônio pode ser descrito então por:

$$\mu_{\mathbf{k}} = \sum_{j=1}^m w_{\mathbf{k}j} x_j \quad (4.6)$$

Tal que, em (4.6): $\mu_{\mathbf{k}}$ é estado do neurônio \mathbf{k} dado pelo valor final do somatório, $w_{\mathbf{k}j}$ são os m pesos do neurônio \mathbf{k} e x_j são os valores de entrada de cada um dos m pesos que compõem este neurônio¹². Ao somatório de (4.6), soma-se também um valor $b_{\mathbf{k}}$ chamado de viés e tem papel de fazer uma transformação afim no resultado do somatório de forma que $\mu_{\mathbf{k}} \neq \mu_{\mathbf{k}} + b_{\mathbf{k}}$. Esse valor é então atribuído a uma função de ativação¹³ $\varphi_{\mathbf{k}}$ que tem como objetivo normalizar o resultado $\mu_{\mathbf{k}} + b_{\mathbf{k}}$, geralmente, dentro do intervalo dado por

¹¹A linearidade ou não-linearidade dos neurônios é definida com respeito a função de ativação utilizada na rede como um todo.

¹²Note que $j = \{1, 2, \dots, m\}$

¹³São exemplos de funções de ativações: a função sigmoide e *Heaviside*.

$[0, 1]$. Considerando $\nu_{\mathbf{k}} = \mu_{\mathbf{k}} + b_{\mathbf{k}}$, pode-se escrever o resultado final deste neurônio \mathbf{k} em função da expressão dado por:

$$\nu_{\mathbf{k}} = \mu_{\mathbf{k}} + b_{\mathbf{k}} .$$

$$y_{\mathbf{k}} = \varphi(\nu_{\mathbf{k}}) . \quad (4.7)$$

O perceptron [40] é a arquitetura mais simples possível para uma rede neural (Figura 4.1), sendo constituída de apenas um neurônio dado por (4.7). Esta rede é capaz de separar o conjunto de dados em apenas duas classes. Porém, esta rede só alcançaria o melhor resultado para o caso em que os dados são de classes linearmente separáveis, já que o perceptron é um classificador linear¹⁴. Para resolver este problema, utilizam-se as generalizações do perceptron acumulando camadas de neurônios entre os valores de entrada da rede neural, de forma que, com mais neurônios (consequentemente, mais pesos sinápticos) seja possível a resolução de problemas que não possuem separação linear, já que, conforme a não-linearidade de sucessivas funções de ativação, a rede neural seria capaz de dominar restrições não-lineares [41]. Essa generalização proveniente do acúmulo de camadas de perceptrons recebe o nome de *Perceptrons Multicamadas* e são a base para os modelos mais complexos de redes neurais atuais, um exemplo destas estruturas mais complexas pode ser visto na figura 4.2 onde temos 3 (N_1 , N_2 e N_3) neurônios acoplados¹⁵.

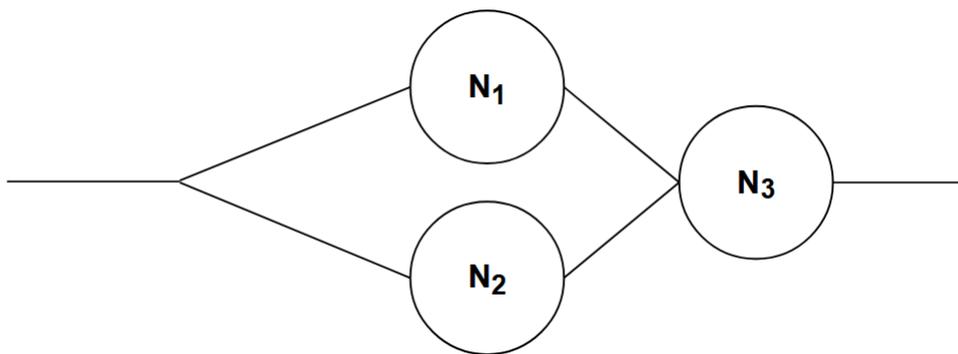


FIGURA 4.2. Exemplo de estrutura de um Perceptron Multicamadas com 3 neurônios. de [35].

¹⁴Um classificador linear funciona atrelando seus resultados a soma de uma combinação linear de variáveis.

¹⁵Note que as arquiteturas específicas de cada um dos neurônios não precisam ser idênticas, com viés ilustrativo, pode-se considerar que os 3 perceptrons da figura 4.2 possuem a arquitetura do Perceptron da figura 4.1.

4.2.1 Redes Convolucionais

As redes neurais de convolução (*Convolutional Neural Networks, CNN*) são camadas específicas de Perceptrons Multicamadas utilizadas para a solução de problemas que envolvam dados em formato de cadeias (*gride*), *e.g.*, uma matriz quadrada¹⁶, que possam ser adaptados ao contexto de uma operação de matrizes baseada na realização matemática da convolução. Pode-se considerar que essas camadas de convolução funcionam como uma espécie de filtro para os dados, de tal maneira que certas características possam ser encontradas de forma natural pela rede. Em [42], por exemplo, temos a implementação de camadas de convolução para a detecção de bordas e contornos em imagens. A capacidade de uma camada convolucional de detectar linhas, segmentos e estruturas se dá em função do núcleo (do inglês, *kernel*) intrínseco a esta camada. Na figura 4.3 é apresentado a esquematização do primeiro passo em uma convolução na matriz A_1 por um *kernel* (bloco 3x3 em azul). Pode-se variar a formatação desse *kernel* em três possíveis planos: o tamanho, o passo e o *padding*¹⁷

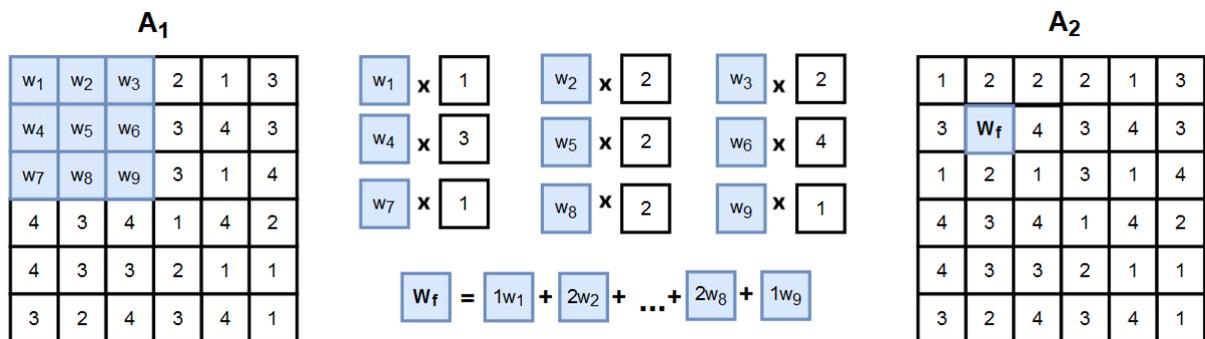


FIGURA 4.3. Exemplo do funcionamento de uma camada de convolução para uma matriz A_1 6×6 com valores aleatórios entre 1 e 4. Para encontrar o resultado final da matriz A_2 , definem-se os pesos w_i no processo de treinamento e então pode-se calcular os w_f resultados das convoluções.

O tamanho (do inglês, *size*) de um núcleo é dado pelas dimensões do núcleo em função do formato dos dados de entrada. Como por exemplo, na figura 4.4, onde a matriz B_1 possui um núcleo de tamanho 3×3 e a matriz B_2 possui um núcleo 2×2 .

¹⁶como por exemplo imagens (dados em duas dimensões) e séries temporais (dados em uma dimensão)

¹⁷Da tradução literal pode-se utilizar o termo “preenchimento”, porém não tem o mesmo sentido e, em razão disso, opta-se por utilizar o termo em inglês.

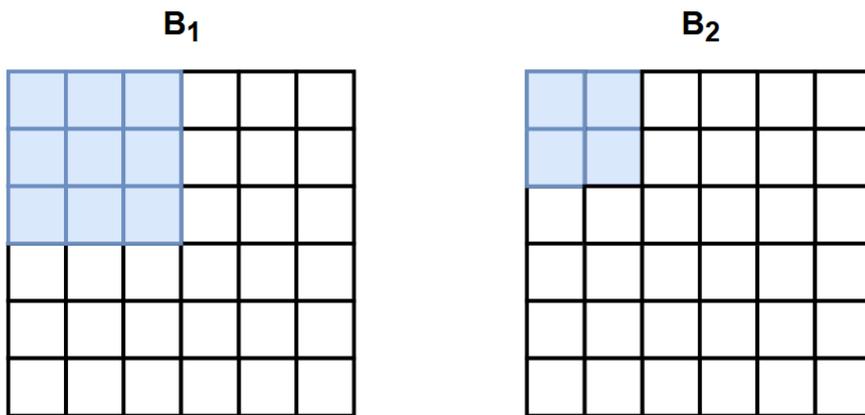


FIGURA 4.4. Diferença entre o tamanho do Kernel da matriz B_1 e Kernel B_2 .

Um kernel tem o valor do passo (do inglês, *stride*) associado aos passos que daria ao “caminhar” pela rede quadrada, como por exemplo os dois passos apresentados na figura 4.5, onde no primeiro caso (matriz C) o núcleo possui um passo de valor 1 e com isso dá um passo deste valor para a direita; já em na matriz D , temos um exemplo para o kernel com o passo de valor 2.

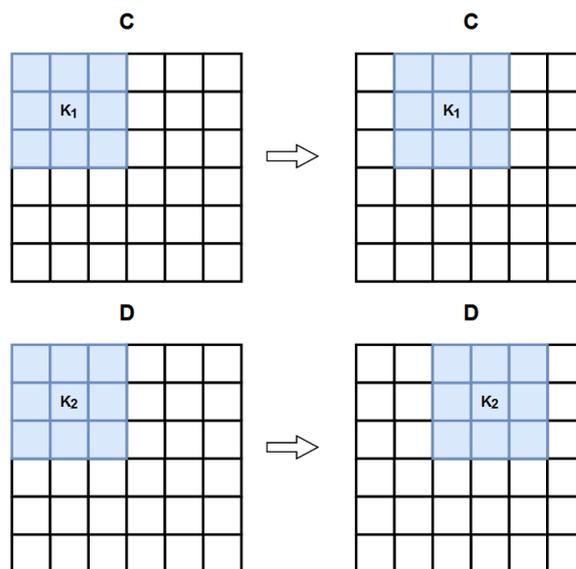


FIGURA 4.5. Diferença entre o passo do Kernel K_1 na matriz C e do Kernel K_2 na matriz D .

E para o *padding*, figura 4.6, é o valor associado a colunas ou linhas do kernel que possuem os pesos $w_i = 0$ fixos e que servem para mantermos as mesmas dimensões de entrada e saída da rede, como por exemplo o kernel \mathbf{K} da figura 4.6 que possui tamanho 3x3 porém com padding igual a 1, tornando-o em um kernel de tamanho 5x5.

K

0	0	0	0	0
0	w_1	w_2	w_3	0
0	w_4	w_5	w_6	0
0	w_7	w_8	w_9	0
0	0	0	0	0

FIGURA 4.6. Exemplo de um Kernel \mathbf{K} com colunas e linhas de *padding* que possuem os pesos $w_i = 0$.

4.2.2 ResNet e o Aprendizado Profundo

O agrupamento de camadas de Perceptrons Multicamadas recebe o nome de Aprendizado Profundo (do inglês, *Deep Learning*) e a grande quantidade de camadas utilizadas está diretamente vinculado ao trabalho desenvolvido na referência [10] com as chamadas Redes Residuais¹⁸ *ResNets*¹⁹ que resolveram o problema da Dissipação do Gradiente [43] atrelado ao uso da função sigmoide, como função de ativação, e o algoritmo de *Backpropagation*²⁰. O aprendizado profundo encontra-se sob o contexto do Aprendizado de Representações [7], assim sendo, faz parte do conjunto de algoritmos de Aprendizado de Máquina que funcionam de maneira a encontrar²¹ variáveis (do inglês, *features*) úteis para resolvermos o problema sem a necessidade de as desenvolver ou encontrá-las manualmente. É comum descrevermos um conjunto de camadas de uma rede neural em função de blocos, tal que estes blocos podem possuir camadas de convolução (vide seção 4.2.1)

¹⁸A partir deste ponto, o termo “redes residuais” e redes neurais será usado com mesmo sentido, já que a rede neural abordado no projeto é também uma rede residual.

¹⁹Do inglês, *Residual Networks*.

²⁰O algoritmo de Backpropagation é discutido em detalhe em [35] e sua explicação foge do escopo deste trabalho.

²¹Novamente, em [42], pode-se entender que as camadas de convolução “encontram” as bordas ou então que estas camadas “geram” as variáveis necessárias para a resolução do problema.

e Camadas de Agrupamento (do inglês *Pooling Layers*), estas são usadas de maneira a “condensarem” as informações encontradas pelas camadas de convolução; semelhante ao funcionamento do kernel apresentado na seção 4.2.1. Essas camadas, geralmente, funcionam no modelo de *Max Pool*²² ou *Average Pool*²³. Baseando-se na figura 4.7, uma camada de agrupamento com um filtro 3x3 na matriz E_1 6x6 (esquerda) resultaria²⁴ na matriz E_2 2x2 (direita), porém, os valores v_1, v_2, v_3 e v_4 são definidos a partir da função específica de agrupamento que seria usada. Por exemplos, para as funções de *Max Pool* e *AveragePool*, teríamos:

$$v_1 = \begin{cases} \max(w_1, 2w_2, 2w_3, \dots, w_9), & \text{para } MaxPool \\ \text{media}(w_1, 2w_2, 2w_3, \dots, w_9), & \text{para } AveragePool. \end{cases} \quad (4.8)$$

Seguindo o esquema de cores da figura 4.7, pode-se elaborar a matriz E_2 resultante da camada de agrupamento para os outros valores v_2, v_3 e v_4 .

No contexto deste trabalho, a arquitetura escolhida²⁵ para a rede neural vai ser inspirada nos conceitos das *ResNets* com as atualizações propostas em [44] nas chamadas *XResNet*.

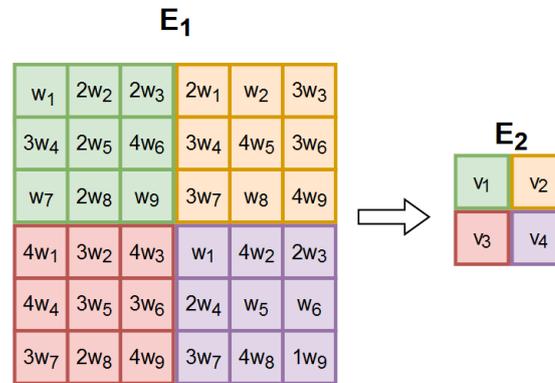


FIGURA 4.7. Esquematização de uma camada de agrupamento na matriz E_1 que resulta na matriz E_2 .

²²Função que selecionaria como resultado o maior valor coberto pelo filtro de agrupamento.

²³Função que devolveria como resultado do filtro a média dos valores contidos dentro do passo do filtro.

²⁴Uma esquematização dos passos para a geração da matriz com os pesos w_i é apresentada na figura 4.3.

²⁵Mais detalhes sobre a arquitetura escolhida são apresentados na seção 5.3.

Capítulo 5

Metodologia, Desenvolvimento e Resultados

Neste capítulo estarão dispostos os processos e métodos utilizados no projeto, tanto para a geração dos dados utilizados como também os passos de pré-processamento necessários para a reprodução dos resultados encontrados. Todos os códigos utilizados no projeto estão disponíveis no GitHub do autor [45].

5.1 Metodologia

5.1.1 Simulações do Modelo de Potts 2D

Para a elaboração das configurações finais da rede quadrada S , como na figura 2.1, utilizou-se dos seguintes aspectos do Modelo de Potts: $q \in \{2, 3, 4, 6, 7, 8\}$, $L = 120$, $\Delta T = 10^4$. Para os 10^4 passos de Monte Carlo, utilizou-se do algoritmo de clusterização de Swendsen-Wang e ainda fez-se uso de Condições de Contorno Periódicas (do inglês, *Periodic Boundary Conditions*) na estruturação da rede e os sítios vizinhos. As temperaturas de transição são conhecidas de forma exata [27] e são dadas por (5.1):

$$T_c = \frac{1}{\log(1 + q)} . \quad (5.1)$$

Onde, em (5.1), T_c é a temperatura de transição do sistema e q é o número de estados possíveis. Gerou-se 2 conjuntos para as temperaturas nas quais os sistemas seriam simulados

200 vezes para cada uma das 100 temperaturas¹ de cada, estes conjuntos foram gerados tais que: $[0,02, \dots, T_C)$ e $[T_C, \dots, 2T_C)$.

5.1.2 Treinamento da Rede Neural

Para o treinamento da rede neural, optou-se por treinar 20 redes diferentes para cada q -estado, ou seja, treinou-se no total 120 redes diferentes iniciadas com pesos aleatórios durante 100 épocas² e com tamanho do *batch*³ em 64. Para a divisão de conjuntos de treino, teste e validação, optou-se pelas seguintes porcentagens: 50%, 25% e 25%, respectivamente. O estudo de 20 séries para cada q -estado se dá na ideia de validarmos a estabilidade dos resultados da rede neural, pois, em função da forma como os pesos são inicializados⁴ e atualizados os resultados de cada série podem divergir e, com mais de uma série por q -estado, pode-se estudar os resultados, de certa forma, com mais confiança⁵ e estabilidade⁶.

5.1.3 Pré-Processamento

As simulações do modelo de Potts foram simuladas no *cluster* do Instituto de Física da Universidade Federal do Rio Grande do Sul em linguagem C [47], para diminuir o tempo de processamento e de simulação. Após os 10^4 passos de Monte Carlo, salvou-se a configuração final da rede quadrada em arquivos de texto com L linhas e L colunas. A partir do ponto no qual as simulações estavam finalizadas, todos os passos descritos na seção 5.1.2 foram implementados em linguagem *Python* [48]. Todo o agrupamento dos resultados em estruturas ideias para o treinamento de redes neurais foi feito utilizando a

¹É importante notar que, no total, cada sistema foi simulado em 200 temperaturas 200 vezes para diferentes valores de q .

²Época é o nome dado a quantidade de vezes que todo o conjunto de dados de treino será usado na fase de treino, ou seja, uma época é o mesmo que usar o conjunto de dados uma única vez.

³Termo em inglês que representa a quantidade de dados do conjunto de treino que serão usados entre as atualizações dos pesos sinápticos.

⁴Os pesos iniciais das redes são aleatoriamente definidos no início do treinamento

⁵Mais confiança no sentido de que a rede não só performou bem por conta de uma *seed* específica no gerador de números aleatórios.

⁶O passeio pelo *landscape* da função de custo é discutido na referência [46] e sua explicação foge do escopo deste trabalho.

biblioteca *Pandas* [49] e o *PyTorch* [50], sendo que a *pipeline*⁷ de treinamento foi elaborada utilizando também o *PyTorch*.

5.1.4 Métricas de Análise

No final do treinamento, pode-se elaborar uma série temporal da evolução do valor da função de custo (4.5) no contexto deste trabalho para cada série de treinamento, como a figura 5.1, que mostra a série temporal de uma rede para $q = 2$ treinada em 100 épocas, chegando a uma loss final no valor de 0.0338. Diferentemente do procedimento da referência [9], que estudou o *Clock Model*⁸ e precisou de apenas 10 épocas de treinamento⁹ para obtenção de resultados aceitáveis, o atual projeto precisou de um período 10 vezes maior, mesmo com uma queda brusca na função de custo nas épocas iniciais, vide figura 5.1.

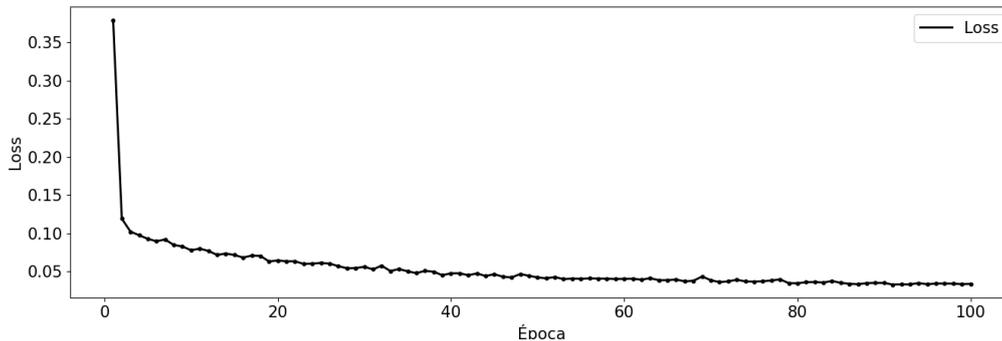


FIGURA 5.1. Série temporal da função de custo de uma rede treinada por 100 épocas com $q = 2$ com valor final de 0.0338.

É importante ressaltar que, por efeito das características do problema, a função de custo utilizada foi a raiz do erro médio quadrado, dado pela expressão (5.2).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (d - y)^2} . \quad (5.2)$$

Em (5.2), n é o número total de dados com resultados calculados pela rede e o fator $(d - y)$ é a diferença entre a temperatura correta e a temperatura encontrada pela rede.

⁷Cada série demorou em média 4h para finalizar, utilizando uma Nvidia®GTX 1050Ti Laptop.

⁸Modelo similar porém diferente quando comparado ao Modelo de Potts abordado neste trabalho.

⁹Não fica claro no texto se o processo utilizou do método de *transfer learning*.

Conforme apresentado na seção 5.1.2, a atualização dos pesos se deu a cada grupo com 64 configurações (120x120) calculados e com estes dados foi calculada a raiz do erro médio. Seguindo a referência [9], para analisarmos o “desvio padrão” do erro das temperaturas encontradas pela rede na fase de validação a partir de:

$$U(T) = \sqrt{\langle (T_r - \langle T_r \rangle_T)^2 \rangle_T} . \quad (5.3)$$

Em (5.3), T_r é a temperatura encontrada pela rede e $\langle T_r \rangle_T$ é a média das temperaturas encontradas pela rede quando a temperatura correta é igual a T . Usando essa métrica, temos uma ideia das regiões nas quais a rede possui mais “facilidade” para acertar a temperatura correta e regiões de temperatura nas quais a rede não consegue entregar bons resultados.

5.2 Resultados para o Modelo de Potts

Utilizando a métrica de (5.3), pode-se elaborar os gráficos de $U(T)$ x T para todos os q abordados neste projeto considerando cada rede treinada para um mesmo q como uma série e calculando uma vez $U(T)$ para os resultados das 20 séries comentadas na seção 5.1.2 agrupadas em um conjunto só, ou seja, os resultados para $U(T)$ nas figuras 5.2 até 5.7.

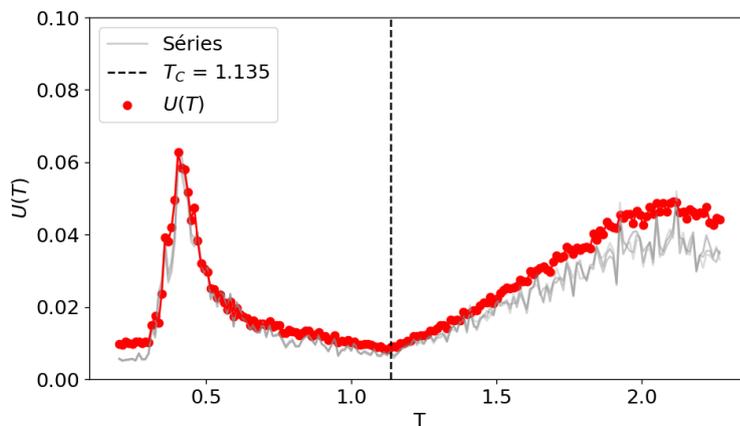


FIGURA 5.2. $U(T)$ para $q = 2$ mostrando em vermelho os dados para o agrupamento de todas as séries e em cinza 5 séries escolhidas aleatoriamente.

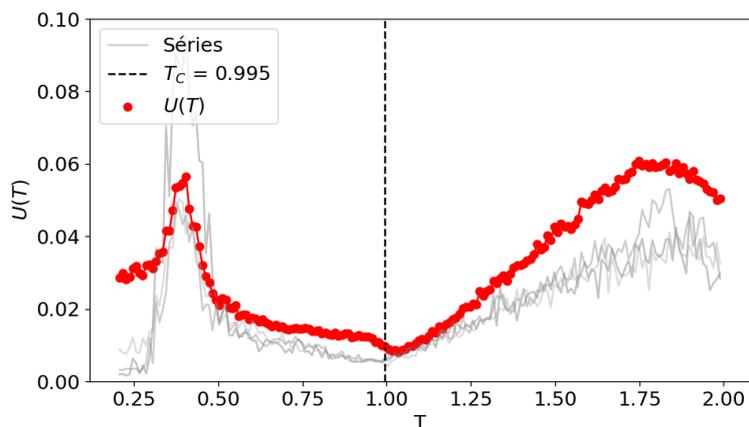


FIGURA 5.3. $U(T)$ para $q = 3$ mostrando em vermelho os dados para o agrupamento de todas as séries e em cinza 5 séries escolhidas aleatoriamente.

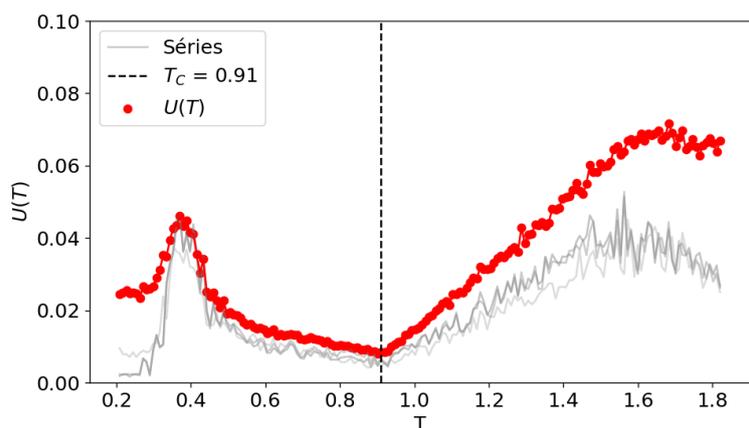


FIGURA 5.4. $U(T)$ para $q = 4$ mostrando em vermelho os dados para o agrupamento de todas as séries e em cinza 5 séries escolhidas aleatoriamente.

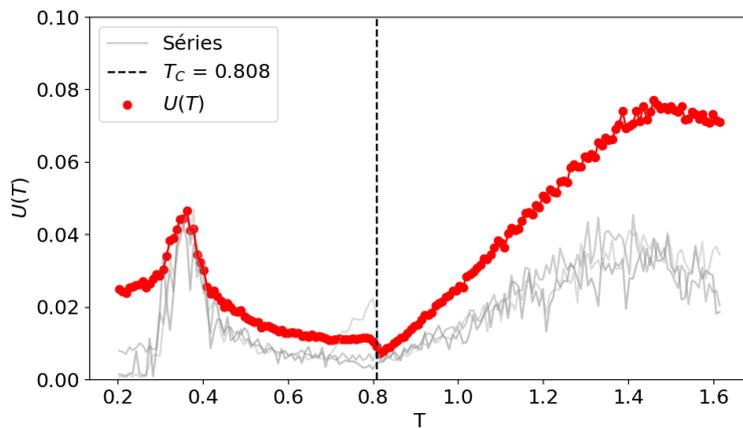


FIGURA 5.5. $U(T)$ para $q = 6$ mostrando em vermelho os dados para o agrupamento de todas as séries e em cinza 5 séries escolhidas aleatoriamente.

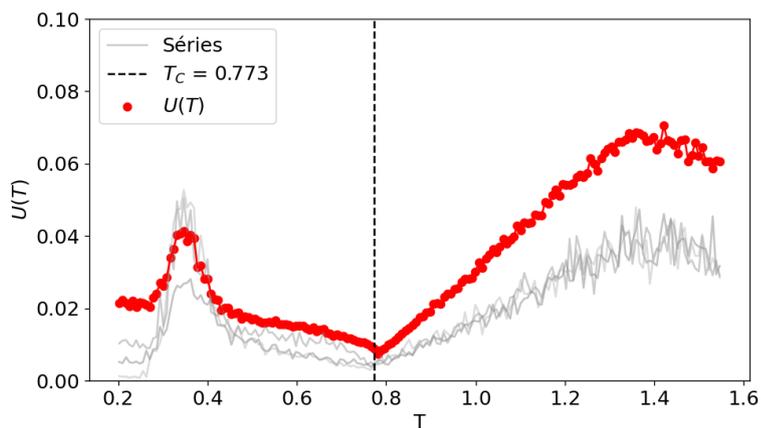


FIGURA 5.6. $U(T)$ para $q = 7$ mostrando em vermelho os dados para o agrupamento de todas as séries e em cinza 5 séries escolhidas aleatoriamente.

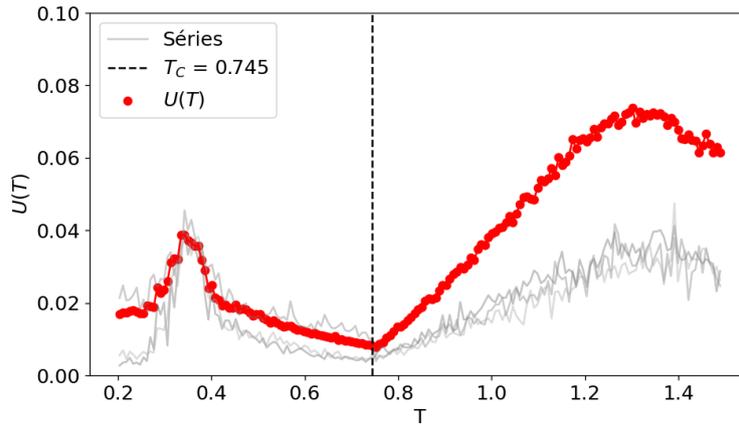


FIGURA 5.7. $U(T)$ para $q = 8$ mostrando em vermelho os dados para o agrupamento de todas as séries e em cinza 5 séries escolhidas aleatoriamente.

Como pode-se perceber nas figuras [5.2-5.7], há uma convergência dos valores mínimos de $U(T)$ nas proximidades da temperatura crítica¹⁰ do sistema de spins para todos os q abordados. A partir desta constatação, calculou-se a média e o desvio padrão da temperatura na qual $U(T)$ tinha seu valor minimizado para cada série e para cada q , pode-se assim, elaborar a tabela 5.1.

Q	T_C (J)	T_{CR} (J)
2	1.135	1.12 ± 0.0342
3	0.995	0.98 ± 0.0368
4	0.910	0.9 ± 0.034
6	0.808	0.8 ± 0.0316
7	0.773	0.75 ± 0.0339
8	0.745	0.73 ± 0.0181

TABELA 5.1. Tabela com os resultados para as temperaturas de transição média encontradas pelas redes residuais considerando 20 séries para cada q -estado.

Na tabela 5.1, pode-se perceber que considerando o intervalo de um desvio padrão, todas as temperaturas de transição exatas (T_C) estão contidas no intervalo das temperaturas de transição encontradas pela rede (T_{CR}), com isso, é aceitável interpretar que a rede residual conseguiu encontrar as temperaturas críticas teóricas com alta confiança. No

¹⁰Temperatura crítica calculado a partir da expressão (5.1).

histograma da figura 5.8, pode-se perceber que a distribuição das temperaturas de transição encontradas pela rede, estão distribuídas¹¹ nas proximidades de 1, porém, nota-se também que há um pequeno desvio da média para valores menores que 1, implicando em resultados das T_{CR} menores que as temperaturas de transição exatas (T_C); isso também poderia ser visto na tabela 5.1, já que todas as médias encontradas são menores que os valores exatos.

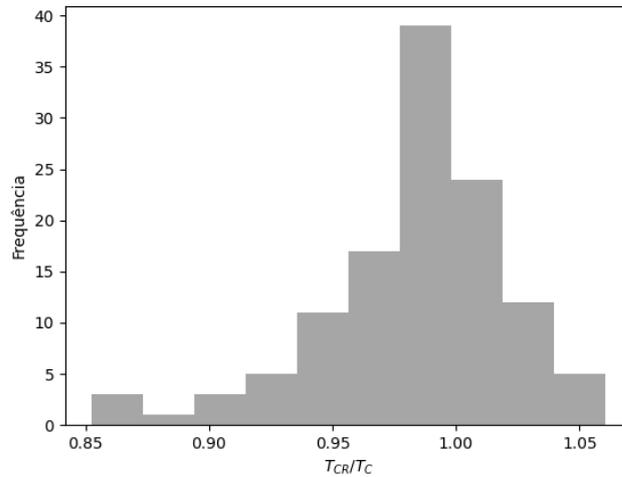


FIGURA 5.8. Distribuição das temperaturas mínimas (normalizadas pela temperatura crítica teórica de cada q -estado) encontradas pela rede residual para todas as 120 séries, 20 para cada q -estado.

Outra maneira de interpretarmos os resultados obtidos a partir do cálculo de $U(T)$ pode ser visualizado na projeção da dispersão dos valores reais de temperatura T e os valores encontrados pela rede (T_R), como na figura 5.9, onde temos um gráfico da dispersão dos valores corretos e os valores entregues pela rede para $q = 4$ para uma série. Pode-se perceber também que a dispersão no eixo y diminui nas regiões de proximidade com a temperatura de transição teórica, como era de se esperar pela figura 5.4.

¹¹Note que as temperaturas do histograma foram normalizadas em função da temperatura de transição de cada q -estado, fazendo com que o intervalo se dê em termos de T_{CR}/T_C , desta forma, pode-se agrupar os resultados em um mesmo histograma.

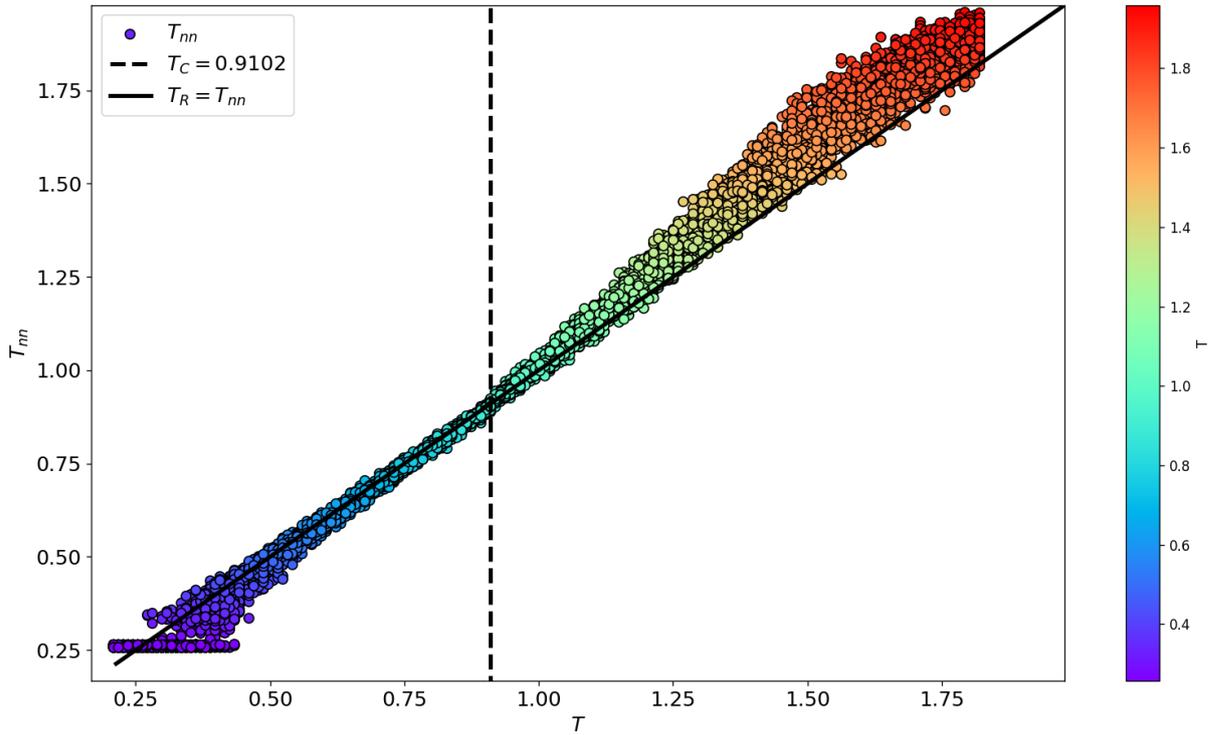


FIGURA 5.9. Projeção da dispersão de T por T_R para todos os resultados das redes treinadas no modelo de Potts com $q = 4$.

Realizando a normalização¹² das temperaturas no gráficos de $U(T)$, pode-se analisar o comportamento das curvas de $U(T)$ em função dos valores de q . O objetivo do gráfico da figura 5.10 era estudar alguma possível diferença entre as curvas para os q -estados em função da ordem de cada uma das transições (expressão (2.11)), podendo assim, além de definir a temperatura crítica, definir também a ordem da transição a partir de algum algoritmo de aprendizado de máquina.

¹²Para a normalização das temperaturas, basta dividir todos os valores pelo valor máximo de temperatura que foi simulado, conforme 5.1.1, para cada valor de q .

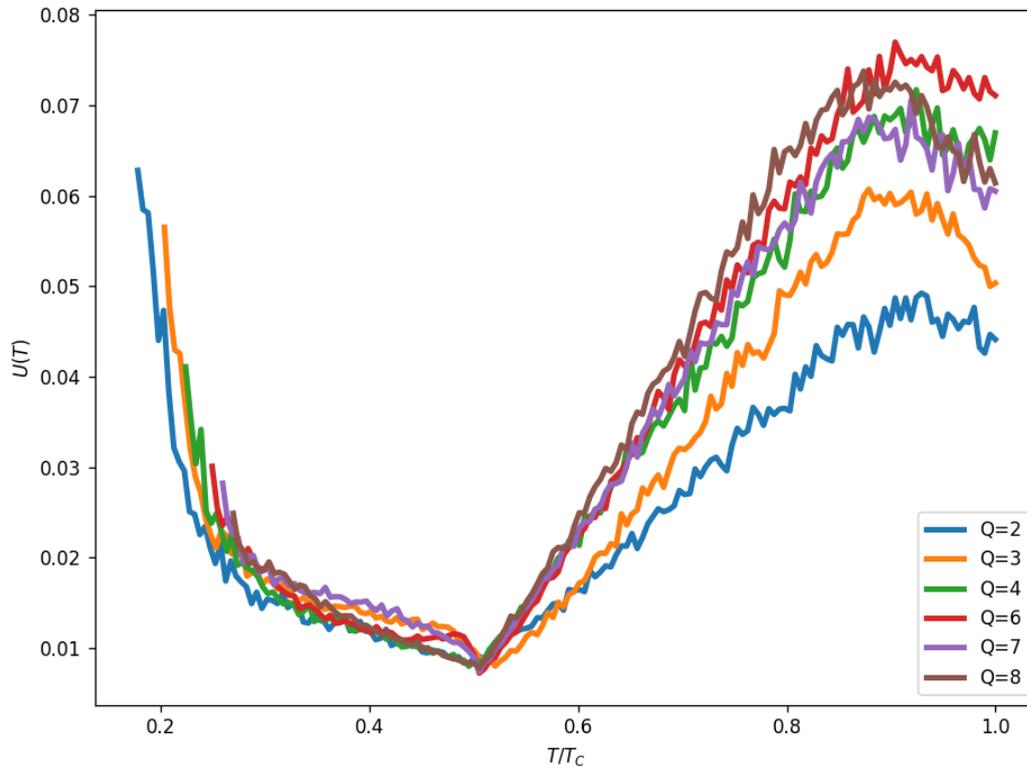


FIGURA 5.10. Curvas de $U(T)$ para diferentes q -estados normalizados pela temperatura de transição de cada q -estado.

Analisando o gráfico, pode-se notar que os q -estados de transição contínua ($q \leq 4$), possuem um pico maior na região a esquerda ponto mínimo da curva de $U(T)$; enquanto, para a região a direita do ponto de mínimo da curva, temos que os q -estados com transição de primeira ordem ($q > 4$) possuem um pico mais alto¹³. Estudou-se a média desse pico a esquerda em função dos q -estados e não foi possível separá-los estatisticamente entre si para que fosse possível estipular um corte horizontal na curva de $U(T)$ separando assim as duas ordens de transição.

¹³Nota-se que neste caso a curva de $q = 4$ está muito próxima das curvas para $q > 4$, o que já demonstra uma possível não separação.

5.3 Arquitetura da Rede

Para a definição da arquitetura da rede, seguiu-se as modificações apresentadas na referência [44]; para as funções de ativação, usou-se a função *Mish* proposta na referência [51] e como otimizador usou a referência [52]. Novamente, não fica claro em [9] qual a arquitetura utilizada pelo autor, apenas que se trata de uma *ResNet*, em função disso, optou-se pelo modelo menos complexo deste conjunto de redes residuais, a *ResNet18*; que após as modificações, passa a ser reconhecida como *XResNet18*. Uma demonstração reduzida da rede é apresentada na figura 5.11.

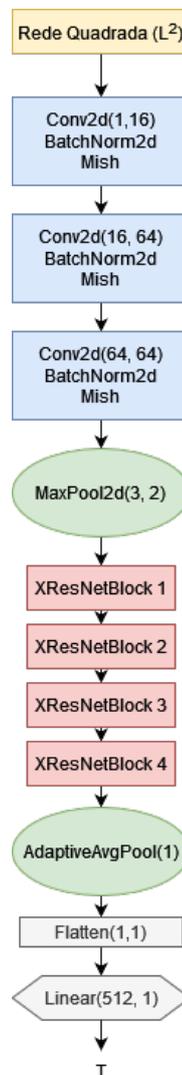


FIGURA 5.11. Arquitetura da rede usada no projeto com camadas de convolução, agrupamento e 4 blocos de *XResNet* resultando em um valor de T .

Todas as funções apresentadas nas figuras 5.11 5.12 têm seus nomes reproduzidos a partir da biblioteca de ferramentas de aprendizado de máquina *Pytorch*¹⁴ [50].

5.3.1 Exemplos de bloco *XResNet*

Como a apresentação da rede em uma só figura não é possível, apresentam-se exemplos dos blocos de *XResNet* ilustrados na figura 5.11 na figura 5.12 como forma de demonstrar o funcionamento de uma rede residual (*ResNet*).

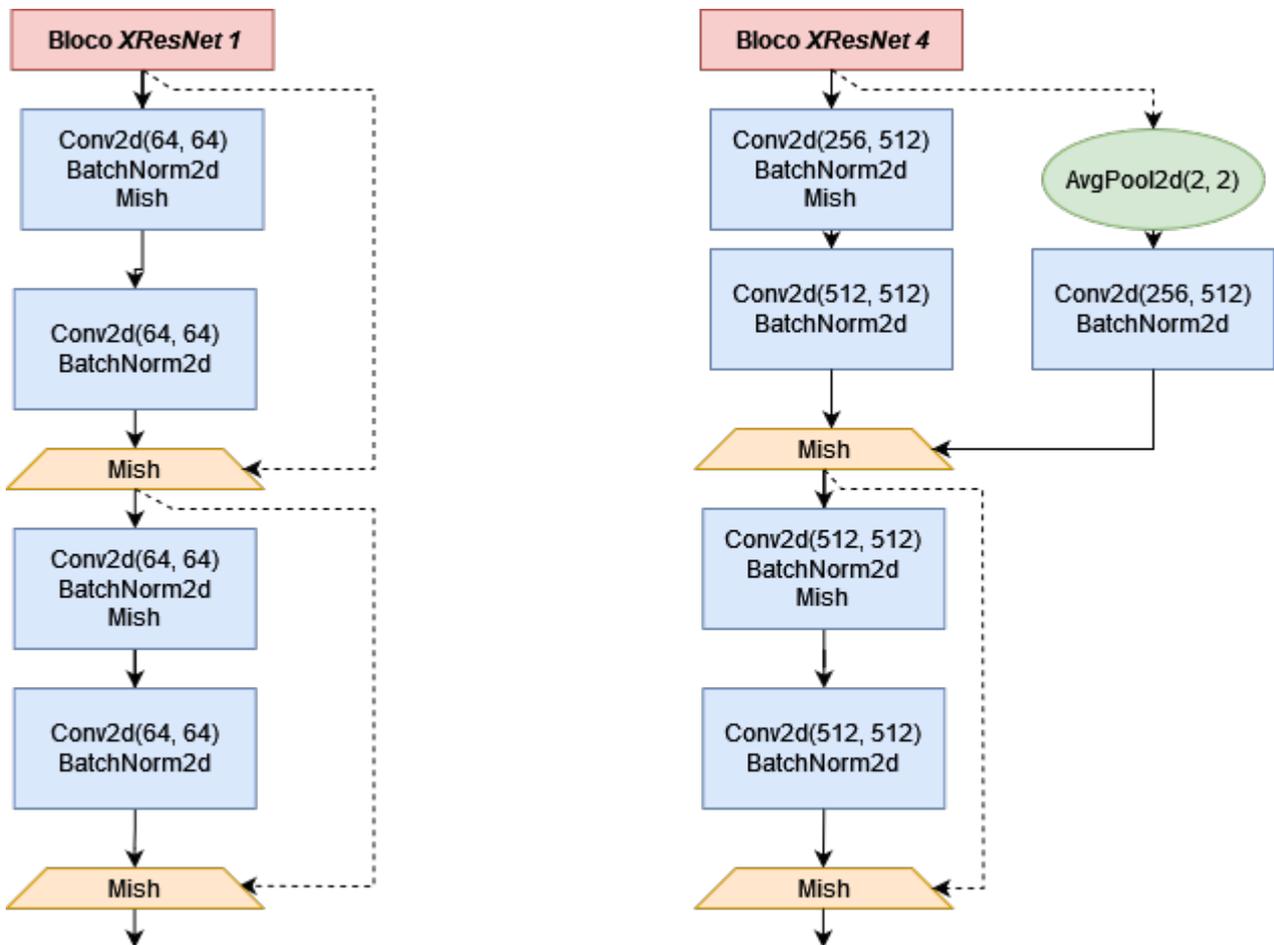


FIGURA 5.12. Arquitetura dos blocos 1 e 4 de *XResNet* mostrando somas do valor de *input* nas linhas tracejadas em paralelo à direção central da rede.

¹⁴Cada uma das funções têm seu funcionamento explicado em <https://pytorch.org/docs/> e não está no escopo do trabalho explica-lás.

5.4 *Feature Maps* da Rede

Conforme apresentado na seção 4.2.2, as camadas de convolução possuem a capacidade de “encontrar” variáveis por conta própria e ainda decidir o que é ou não uma boa variável. Em função disso, é muito comum [53, 54] o estudo dos *feature maps*¹⁵ das camadas de convolução, na ideia de ver se as variáveis que a rede acaba encontrando possuem alguma relação com o nosso entendimento humano do problema. No contexto do atual projeto, analisando a projeção dos mapas de variáveis, buscou-se na verdade entender como que o algoritmo de aprendizado de máquina consegue interpretar as relações físicas entre os spins da rede quadrada e entregar a temperatura correta da rede, sem nenhum ferramental físico envolvido. Na tentativa de encontrarmos algum mapa com alguma referência à relações físicas, foram elaboradas as figuras: 5.13, 5.14 e 5.15; nas quais temos os resultados dos *feature maps* para o Modelo de Potts com $q = 6$ e temperaturas $T = \{0, 56, 0,87, 1,28\}$, respectivamente¹⁶. Nota-se que a matriz final da rede neural possui valores menores conforme a temperatura aumenta (no mapa de calor utilizado, as cores escuras representam valores baixos e cores claras valores altos), porém não é possível retirar nenhuma informação muito concisa sobre o funcionamento do mapa de filtros, diferentemente de [55, 56], onde foi possível relacionar os resultados do algoritmo da Análise de Componentes Principais (*PCA*) com a magnetização do sistema de spins.

¹⁵Pode ser entendido como mapa de variáveis, mapa de características ou mapa de filtros também.

¹⁶Temos temperaturas antes, próxima e depois da transição, respectivamente; seguindo (5.1), $q = 6$ implica em $T_C = 0.808$.

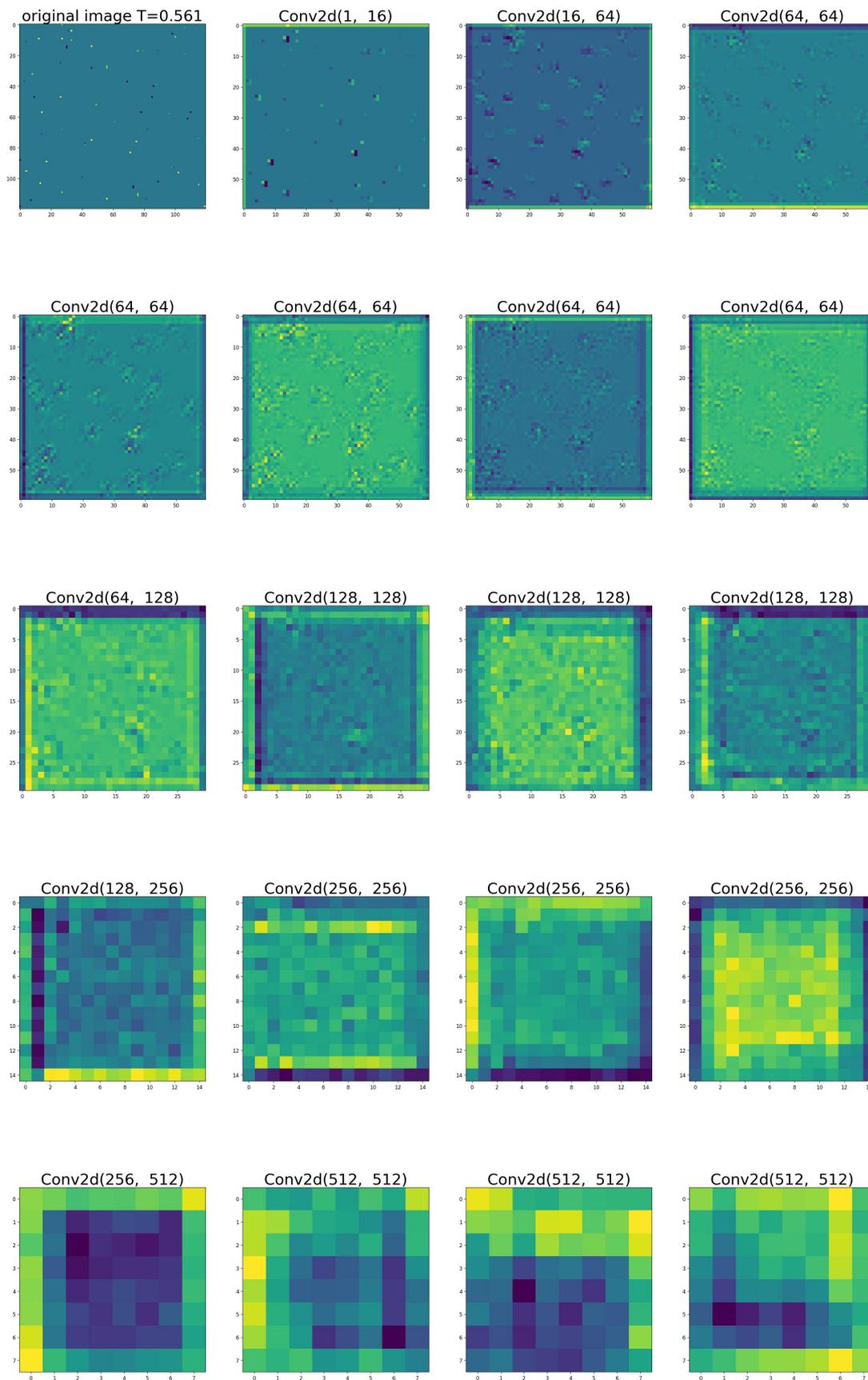


FIGURA 5.13. *Feature Map* para uma configuração do modelo de Potts com $q = 6$ e temperatura $T = 0.56$.

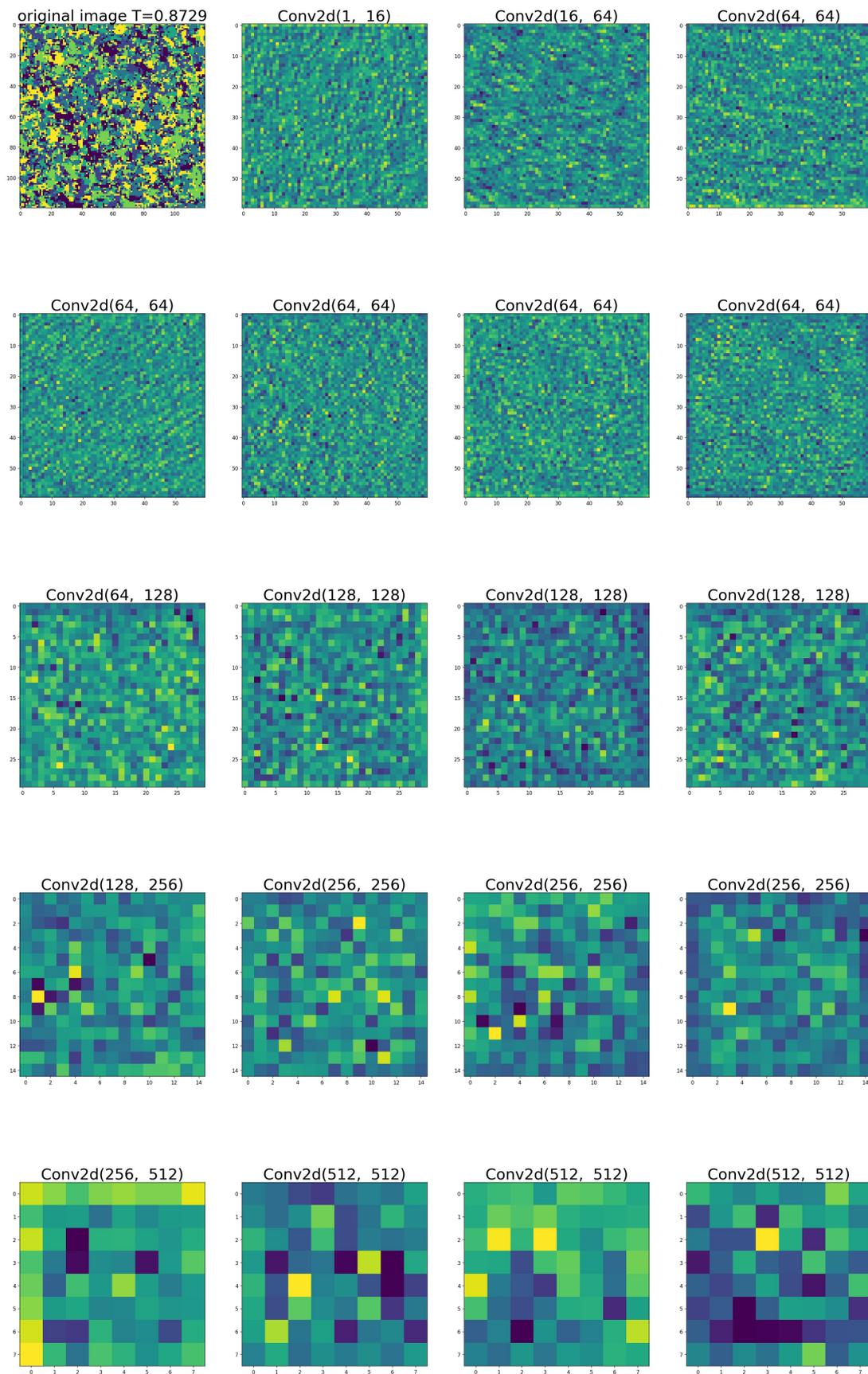


FIGURA 5.14. *Feature Map* para uma configuração do modelo de Potts com $q = 6$ e temperatura $T = 0.87$.

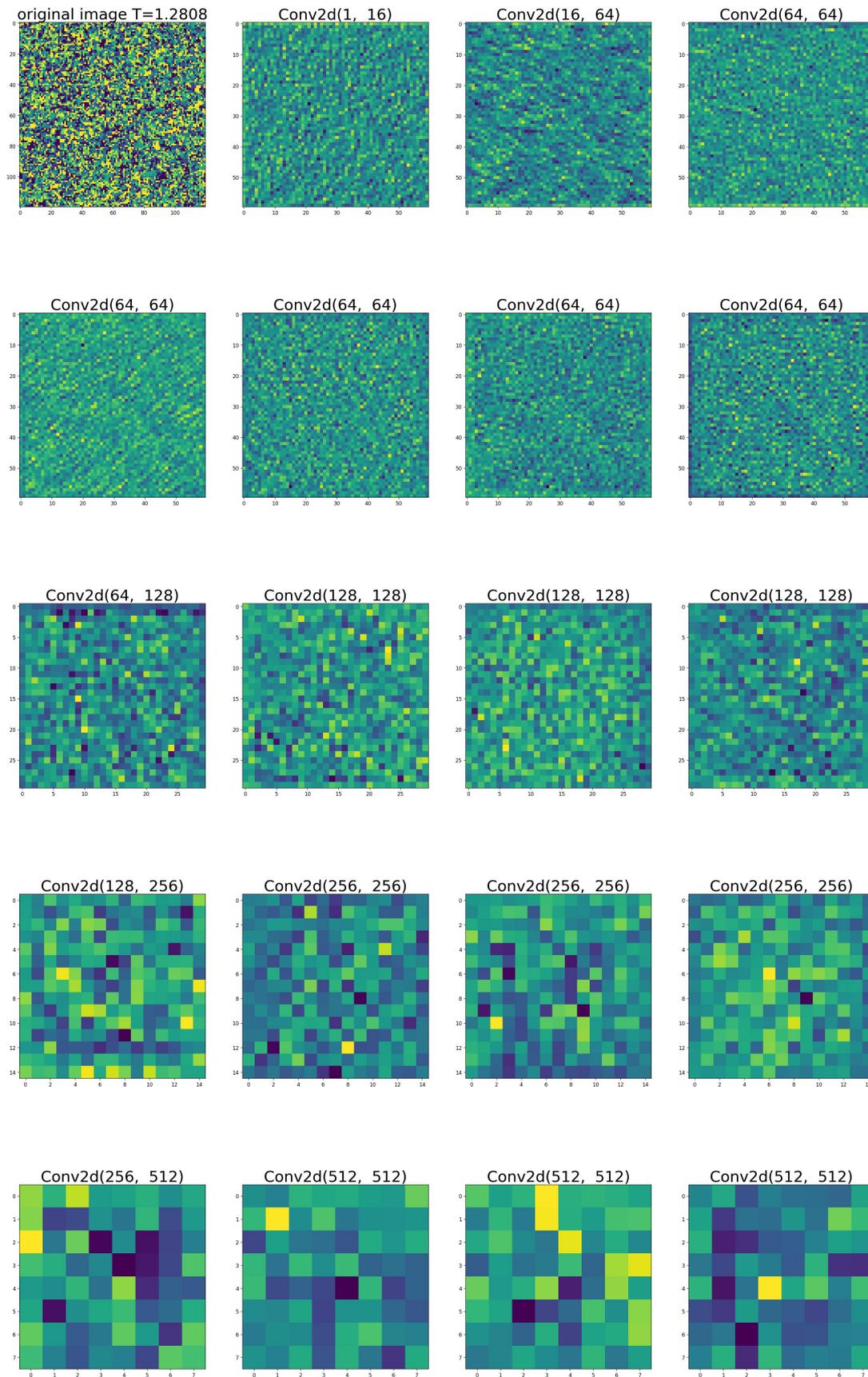


FIGURA 5.15. *Feature Map* para uma configuração do modelo de Potts com $q = 6$ e temperatura $T=1.28$.

Capítulo 6

Conclusão

Neste último capítulo são apresentados os comentários do autor sobre os resultados encontrados durante o projeto e algumas perspectivas futuras sobre a pesquisa e os próximos passos.

6.1 Conclusão

De certa forma, os resultados (re)produzidos neste trabalho estão de acordo com o estudado na literatura, como¹ em [9], onde ambos os trabalhos conseguem encontrar as transições de fase a partir da incerteza da regressão de redes residuais, demonstrando que existem elementos intrínsecos às redes neurais que demandam o estudo dos físicos tanto para o avanço da física quanto para o entendimento computacional da aprendizagem dos algoritmos. Os conhecimentos adquiridos sobre a implementação de redes neurais para a solução de problemas físicos são de suma importância para o futuro da pesquisa.

Analisando apenas o objetivo primário do projeto, percebe-se que este foi atingido, já que, para todo q abordado, a temperatura de transição T_c exata está dentro do intervalo encontrado pela rede (T_{cr}), ou seja, é possível encontrar a transição de fase do modelo de Potts em 2D a partir da incerteza de redes residuais (atualmente, um resultado inédito na literatura). Os resultados para os *feature maps* não foram, em primeiro momento, satisfatórios; uma vez que não apresentam nenhuma característica evidente que possa

¹Na referência [9], o modelo é treinado no chamado *Clock-Model*, que possui formulação diferente da tomada no atual projeto e outros comentários sobre as diferenças nos métodos são apresentados na seção 5.1.4.

ser relacionada com algum conceito físico que explique a funcionalidade da rede residual utilizada neste trabalho. Porém, estudou-se apenas uma técnica de visualização e não se pode confirmar esta “não-relação” com a física. Mesmo com os bons resultados obtidos com o parâmetro $U(T)$ no problema da regressão não se pode inferir a ordem da transição a partir das curvas. Porém, um melhor entendimento deste parâmetro é importante e pode levar ao desenvolvimento e obtenção de novos parâmetros mais apropriados; assim como pode-se estudar também a relação deste parâmetro com outros conhecido, como o PCA, que obteve como resultado as simetrias das fases de baixas temperaturas.

6.1.1 Perspectivas Futuras

Em próximas etapas, pode-se estudar novas arquiteturas para as redes e ainda as possíveis utilidades do *Transfer Learning*², como na referência [57], analisando os resultados de um rede treinada em um q mas testada em outro ou ainda treinar uma rede para todos os q -estados. Pode-se ainda estudar mais afundo os *Feature Maps*, buscando ainda entender o que a rede neural está “enxergando” para resolver a temperatura do sistema sem a fundamentação física que é, usualmente, utilizada e também a relação do parâmetro $U(T)$ com outros parâmetros já conhecidos e validados pela literatura. Na referência [58], é apresentado o estudo do uso de redes convolucionais para a obtenção da ordem da transição de fase encontrada também por redes convolucionais, demonstrando que ainda pode-se estudar o uso de redes nas duas fases do problema, na solução da transição de fase e ainda na definição da ordem dessa transição.

²Técnica na qual uma rede treinada para um problema é testada na solução de outro problema.

Referências Bibliográficas

- [1] M. Stephens, “A learning revolution,” *Physics World*, vol. 32, p. 45, mar 2019.
- [2] G. Carleo and M. Troyer, “Solving the quantum many-body problem with artificial neural networks,” *Science*, vol. 355, pp. 602–606, feb 2017.
- [3] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Šilver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, pp. 583–589, July 2021.
- [4] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, “Unsupervised word embeddings capture latent knowledge from materials science literature,” *Nature*, vol. 571, pp. 95–98, July 2019.
- [5] N. Thurey, P. Holl, M. Mueller, P. Schnell, F. Trost, and K. Um, *Physics-based Deep Learning*. WWW, 2021.
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [7] Y. Bengio, A. C. Courville, and P. Vincent, “Unsupervised feature learning and deep learning: A review and new perspectives,” *CoRR*, vol. abs/1206.5538, 2012.
- [8] J. Hansson, “The 10 biggest unsolved problems in physics,” 01 2015.
- [9] W.-c. Guo and L. He, “Learning phase transitions from regression uncertainty,” 2022.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.
- [11] R. B. Potts, “Some generalized order - disorder transformations,” *Proc. Cambridge Phil. Soc.*, vol. 48, pp. 106–109, 1952.
- [12] G. Barkema and J. de Boer, “Numerical study of phase transitions in potts models,” *Physical Review A*, vol. 44, pp. 8000–8005, Dec. 1991.
- [13] A. Balter, R. M. H. Merks, N. J. Popławski, M. Swat, and J. A. Glazier, “The glazier-graner-hogeweg model: Extensions, future directions, and opportunities for further study,” in *Single-Cell-Based Models in Biology and Medicine*, pp. 151–167, Birkhäuser Basel.

- [14] L. Zdeborová, “Understanding deep learning is also a job for physicists,” *Nature Physics*, vol. 16, pp. 602–604, May 2020.
- [15] O. Bousquet, S. Boucheron, and G. Lugosi, *Introduction to Statistical Learning Theory*, pp. 169–207. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.
- [16] A. Tanaka, A. Tomiya, and K. Hashimoto, *Deep Learning and Physics*. Springer Singapore, 2021.
- [17] 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.
- [18] W. K. Hastings, “Monte carlo sampling methods using markov chains and their applications,” *Biometrika*, vol. 57, pp. 97–109, Apr. 1970.
- [19] R. H. Swendsen and J.-S. Wang, “Nonuniversal critical dynamics in monte carlo simulations,” *Physical Review Letters*, vol. 58, pp. 86–88, Jan. 1987.
- [20] F. m. c. Graner and J. A. Glazier, “Simulation of biological cell sorting using a two-dimensional extended potts model,” *Phys. Rev. Lett.*, vol. 69, pp. 2013–2016, Sep 1992.
- [21] M. Šimėnas, S. Balčiūnas, M. Maćzka, J. Banys, and E. E. Tornau, “Structural phase transition in perovskite metal–formate frameworks: a potts-type model with dipolar interactions,” *Phys. Chem. Chem. Phys.*, vol. 18, pp. 18528–18535, 2016.
- [22] S. R. A. Salinas, *Introdução a Física Estatística Vol. 09*. Edusp, 1997.
- [23] J. Yeomans, *Statistical Mechanics of Phase Transitions*. Clarendon Press, 1992.
- [24] M. S. S. Challa, D. P. Landau, and K. Binder, “Finite-size effects at temperature-driven first-order transitions,” *Physical Review B*, vol. 34, pp. 1841–1852, Aug. 1986.
- [25] M. Fisher, *The Nature of Critical Points*. University of Colorado Press, 1965.
- [26] M. E. J. Newman and G. T. Barkema, *Monte Carlo methods in statistical physics*. Oxford: Clarendon Press, 1999.
- [27] R. J. Baxter, “Potts model at the critical temperature,” *Journal of Physics C: Solid State Physics*, vol. 6, p. L445, nov 1973.
- [28] E. Seneta, “Markov and the creation of markov chains,” p. 20, 01 2006.
- [29] L. BOLTZMANN, “On certain questions of the theory of gases,” *Nature*, vol. 51, pp. 413–415, Feb. 1895.
- [30] L. Boltzmann, *Lectures on gas theory*. Courier Corporation, 2012.
- [31] N. ITO and G. KOHRING, “CLUSTER VS. SINGLE-SPIN ALGORITHMS—WHICH ARE MORE EFFICIENT?,” *International Journal of Modern Physics C*, vol. 05, pp. 1–14, Feb. 1994.
- [32] M. Hazewinkel, ed., *Encyclopaedia of Mathematics*. Encyclopaedia of Mathematics, New York, NY: Springer, Feb. 2002.
- [33] V. Vapnik, *Estimation of dependences based on empirical data*. Springer Series in Statistics, New York, NY: Springer, Oct. 1982.

- [34] V. N. Vapnik, *Statistical learning theory*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control, Nashville, TN: John Wiley & Sons, Sept. 1998.
- [35] S. S. Haykin, *Neural networks and learning machines*. Upper Saddle River, NJ: Pearson Education, third ed., 2009.
- [36] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, “Generalization in deep learning,” 2017.
- [37] L. Breiman *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [38] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, Sept. 1995.
- [39] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, “Discovering physical concepts with neural networks,” *Phys. Rev. Lett.*, vol. 124, p. 010508, Jan 2020.
- [40] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain.,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [41] N. Kulathunga, N. R. Ranasinghe, D. Vrinceanu, Z. Kinsman, L. Huang, and Y. Wang, “Effects of the nonlinearity in activation functions on the performance of deep learning models,” 2020.
- [42] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, “Casenet: Deep category-aware semantic edge detection,” 2017.
- [43] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, Mar. 1994.
- [44] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” *CoRR*, vol. abs/1812.01187, 2018.
- [45] M. Guimaraes, “Tcc.” <https://github.com/mgteus/FisComps/tree/main/TCC>, 2022.
- [46] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, “Visualizing the loss landscape of neural nets,” 2017.
- [47] B. W. Kernighan and D. M. Ritchie, *The C programming language*. 2006.
- [48] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [49] . Pandas development team, “pandas-dev/pandas: Pandas,” Feb. 2020.
- [50] T. c. TorchVision maintainers and Torchvision, “Torchvision: Pytorch’s computer vision library,” 2016.
- [51] D. Misra, “Mish: A self regularized non-monotonic neural activation function,” *CoRR*, vol. abs/1908.08681, 2019.
- [52] Q. Tong, G. Liang, and J. Bi, “Calibrating the learning rate for adaptive gradient methods to improve generalization performance,” *CoRR*, vol. abs/1908.00700, 2019.
- [53] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *CoRR*, vol. abs/1311.2901, 2013.

- [54] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” 2013.
- [55] L. Wang, “Discovering phase transitions with unsupervised learning,” *Physical Review B*, vol. 94, nov 2016.
- [56] P. H. Mendes, “Aprendizado não supervisionado em sistemas de spins,” 2022.
- [57] K. Fukushima and K. Sakai, “Can a CNN trained on the ising model detect the phase transition of the iq/i-state potts model?,” *Progress of Theoretical and Experimental Physics*, vol. 2021, May 2021.
- [58] T. Xue, X. Li, X. Chen, L. Chen, and Z. Han, “Machine learning phases of active matter,” 2022.