RESTRICTED BOLTZMANN MACHINES AND THE RENORMALIZATION GROUP: LEARNING RELEVANT INFORMATION IN STATISTICAL PHYSICS

Jesse Q. Hoogland
Amsterdam University College
Amsterdam, the Netherlands
jessequinten@gmail.com

Dr. P. Marcos Crichigno
Supervisor
University of Amsterdam
Amsterdam, the Netherlands
P.M.Crichigno@uva.nl

Prof. Dr. Max Welling
Reader
University of Amsterdam
Amsterdam, the Netherlands
M.Welling@uva.nl

Dr. Michael P. McAssey
Tutor
Amsterdam University College
Amsterdam, the Netherlands
M.P.McAssey@auc.nl

June 3, 2019
Major: Sciences
Word count: 9798

ABSTRACT

Recent work has drawn attention to the links between statistical physics and machine learning (ML) and, in particular, to comparisons between the renormalization group (RG) and deep neural networks, respectively. These have inspired renewed interest in the information-theoretic framework underpinning these fields, prompting a better understanding of what RG is. In this capstone, we introduce and expand upon these connections from the ground up. Starting with the basics of ML and RG, we work our way to an algorithm implementated on neural networks that learns optimal, model-independent RG procedures, the real-space mutual information (RSMI) algorithm. Along the way, we review the current state of the literature, clarifying misconceptions in earlier works. With the RSMI algorithm, we review a novel calculation of the Ising model critical exponent, and we generalize this approach to arbitrary lattice systems. We release an open-source library, *rgpy*, for implementing these novel procedures, and close with a discussion of the wide-ranging implications.

***K*eywords** Machine Learning, Restricted Boltzmann Machines, The Renormalization Group, Information Theory, Mutual Information

Contents

Chapter 1

Introduction

In the age of *big data*, machine learning (ML), a subset of artificial intelligence (AI), has become more than *just* another set of data analysis tools [1]. For one, ML's connections with theoretical physics are multivarious and deeply conceptual; the very success of ML may, in part, result from physical principles including symmetry, locality, and hierarchy [2]. Furthermore, ML and theoretical physics share a powerful conceptual framework in information theory [3]. Beyond data analysis, the intersection of ML and physics contains a unique set of ideas that researchers in both fields can leverage to solve tough problems.

In particular, recent work has drawn attention to the similarities between ML and a class of techniques from statistical physics known as the renormalization group (RG) [4, 5, 6, 2]. Developed in the last century, RG has been crucial in making sense of critical behavior, those phenomena characterizing phase transitions. In 2014, Mehta and Schwab published a seminal paper describing an *exact* equivalence between a technique from RG and a type of neural network (NN) from ML [4]. This, however, met criticism, and it works only under a narrow set of circumstances. The similarities between ML and RG, then, are still largely qualitative, and this remains an active area of research. Not to mention, the research landscape maintains lingering misconceptions about the details of the intersection [6, 2, 7, 8]. This warrants further investigation, and in order to facilitate and encourage such research, our first contribution is to provide a clarifying overview of the competing views. We resolve a number of inaccuracies.

In 2018, Koch-Janusz and Ringel derived an algorithm which uses neural networks to learn RG transformations on lattice systems: the *real-space mutual information* (RSMI) algorithm [5]. Notably, this method is *unsupervised* which is particularly relevant for research into poorly understood physical systems; information-theoretic approaches, like this algorithm, may guide researchers towards the locations of critical points and even calculations of critical exponents. Furthermore, Koch-Janusz and Ringel's derivation is *optimal* in a rigorous sense we define in section 5 [5, 9]. This is exciting because many well-established practices in RG lack precise justification. More exact formulations, like these, may inspire more effective implementations, not to mention a better understanding of why these ML and RG techniques work.

In our investigation, we will develop a set of tools for tackling critical phenomena. First, we consider some of the standard techniques of statistical physics [2], building towards an ML-derived implementation of RG [5]. We, then, introduce elements of ML, emphasizing their utility in a

variety of statistical physics contexts [3]. We anchor this investigation around the Ising model, one of the most important models in statistical physics. To compare these various techniques, we evaluate their ability in predicting the Ising model's correlation length critical exponent, ν.

To accomplish this, we have built and shared an open-source implementation of the RSMI algorithm [10] through *rgpy* [6]. Hereby, we provide a calculation for ν [6].Then, we describe a generalization of this algorithm to *arbitrary* lattice systems. This gives rise to a family of RSMI-inspired approaches.

It is our aim to enable and inspire researchers to build further on our results. We accomplish this by reviewing the current state of research, sharing an implementation of the RSMI algorithm, and describing avenues of future research. Although we focus on the perspective of statistical physicists, this capstone is accessible for both ML and physics researchers, even at an undergraduate level.

In section 2, we begin by introducing techniques native to statistical physics. We describe mean-field theory, and its failures bring us to the renormalization group. In section 3, we discuss two examples of ML in physical investigations. First, we use neural networks to classify Ising model phases. Then, we use the same neural networks to generate new samples of Ising models. These examples serve to introduce the basics of ML, assuming no prior knowledge (except mathematical maturity), and the same is true for the portion on statistical physics. In section 4, we explore the similarities between ML and RG, and by being explicit in our formulation of "relevant" information, we manage to avoid some of the mistakes of earlier comparisons. In section 5, we explain and justify the RSMI algorithm, following the formulation of Koch-Janusz and Ringel. In section 6, we provide our own results: a recalculation of ν and a generalization of this technique, paving the way for a new class of RG techniques. In section 7, we close with a discussion, reflecting on our comparisons of ML and statistical physics and emphasizing the wide-ranging impacts of these ideas.

1.1 Notation

To refer to single microscopic elements (e.g., spins in the Ising model or pixels in an image), we use lowercase letters with a lower index (x_i, y_j, etc.). To refer to collections of microscopic elements, microstates or images, we use boldface, lowercase letters ($\boldsymbol{x} \coloneqq \{x_i\}$, $\boldsymbol{y} \coloneqq \{y_j\}$, etc.).

To refer to collections of microstates, we use uppercase, cursive letters, $\mathcal{X} \coloneqq \{\boldsymbol{x}\}$. We will be interested in performing sums and averages over these sets. Rather than introduce an index to keep track of each term, we do so implicitly in the sums. For example, given some function $A(\boldsymbol{x})$, the following are equivalent: $\sum_n A(\boldsymbol{x}^{(n)}) \equiv \sum_{\boldsymbol{x}\in\mathcal{X}} A(\boldsymbol{x}) \equiv \sum_{\boldsymbol{x}} A(\boldsymbol{x})$. Most often, we use the last notation.

If we partition our microstates into subsets (as with block renormalization), we also use boldface, lowercase letters ($\boldsymbol{v}$, $\boldsymbol{h}$, etc.). To distinguish partitions, we may use an upper index: $\boldsymbol{v}^{(n)}$.

For partial derivatives, we typically use the shorthand $\partial_t \coloneqq \frac{\partial}{\partial t}$.

For Ising models, we will consider systems with binary units $\in \{-1, 1\}$, following standard convention. For RBMs, we use the standard notation of binary units $\in \{0, 1\}$. When using RBMs on Ising data, then, we map $-1 \to 0$.

Chapter 2

Foundations of Statistical Physics

Statistical physics emerged in the second half of the nineteenth century as an answer to unresolved questions in thermodynamics, the study of heat and work. Was heat continuous and wavelike, or might it be something else, discrete and atomic? Founding figures in the field, including Rudolf Clausius, Ludwig Boltzmann, and James Clerk Maxwell, answered the latter. Introducing the kinetic theory of gases, these scientists posited gases as large collections of tiny molecules and heat flow as the net effect of unbalanced molecular collisions. These ideas, atomic theory, were not uncontroversial. To defend these claims, their key task would be to translate such microscopic descriptions to experimentally-verifiable predictions [11].

To complicate matters, these scientists lacked equipment that could resolve the proposed microscopic length scales, and a square cubic centimeter of gas can contain upwards of a million million million molecules [11]. The key insight in statistical physics is to focus on the properties of the collection rather than on the individual components– on averages and distributions rather than microscopic details. This translating between microscopic and macroscopic is the essence of statistical physics, and it is to this task we dedicate our efforts.

Our investigation begins by defining a *microstate*, $\boldsymbol{s}$: a full description of the microscopic *degrees of freedom* of our system, $\boldsymbol{s} \coloneqq \{s_i\}$, where s_i is the i-th DOF, some fundamental way in which the system can vary.[1] Our aim, as statistical physicists, is to predict the outcomes of macroscopic measurements. A key assumption of statistical mechanics is that we can express measurement outcomes as averages over all microstates, $\mathcal{S}$ (see section A.1). If we are interested in measuring energy, E, our *expectation* $\langle E \rangle$, will be:

$$\langle E \rangle \coloneqq \sum_{\boldsymbol{s} \in \mathcal{S}} P(\boldsymbol{s}) E(\boldsymbol{s}) \tag{2.1}$$

Making macroscopic predictions boils down to evaluating probabilities of microstates and sums thereof. In the following section, we will derive the probability distribution $P(\boldsymbol{s})$ and encounter the first of the fundamental challenges in statistical physics. We follow the treatment of Domb [11] and Cardy [12].

[1] For example, *spin*, which we will encounter in the next section.

2.1 Probabilities and Partition Functions

Let us consider an example: we begin with some physical system, $\mathcal{S}$. It could be metal or really anything. As we previewed, the microscopic details will not matter to the macroscopic picture. To measure macroscopic properties of $\mathcal{S}$, we need its distribution over microstates $\boldsymbol{s}$, $P(\boldsymbol{s})$. The trick is to introduce a *reservoir*, $\mathcal{R}$, that surrounds $\mathcal{S}$. Just like $\mathcal{S}$, the reservoir could be anything: a gas, liquid, etc. We impose several conditions: $\mathcal{S}$ and $\mathcal{R}$ exchange only energy and the combined system, $\mathcal{X} = \mathcal{S} \cup \mathcal{R}$ (the *universe*) is isolated. Then, the total energy, E, is conserved. If we denote the energy of microstates, $\boldsymbol{s}$ and $\boldsymbol{r} = \{r_j\}$ as $E(\boldsymbol{s})$ and $E(\boldsymbol{r})$, respectively, this requires $E(\boldsymbol{s}) + E(\boldsymbol{r}) = E$.[2] Furthermore, we treat $\mathcal{S}$ as a much smaller fraction of $\mathcal{X}$ than $\mathcal{R}$, though $\mathcal{S}$ is still macroscopic ($1 \ll |\boldsymbol{s}| \ll |\boldsymbol{r}|$, where $|\boldsymbol{x}|$ denotes the number of degrees of freedom of $\boldsymbol{x}$).

The fundamental assumption of statistical mechanics states that, for an isolated system, each microstate is equally probable. Then, our probability distribution is $P(\boldsymbol{x}) = 1/\Omega(\boldsymbol{x})$, where $\Omega(\boldsymbol{x})$ is the number of possible microstates $\boldsymbol{x}$. Probabilities over subsets of such a system may be more complicated. Consider that the probability of $\boldsymbol{s}$ is proportional to the number of ways we can rearrange the reservoir, keeping the energy constant. If we let $\Omega_{\boldsymbol{r}}(\boldsymbol{s})$ denote the number of microstates, $\boldsymbol{r}$, with this energy $E - E(\boldsymbol{s})$:

$$P(\boldsymbol{s}) = c\Omega_{\boldsymbol{r}}(\boldsymbol{s}), \tag{2.2}$$

where c is the constant of proportionality. By the fundamental assumption of statistical mechanics, the above holds for any choice in microstate, $\boldsymbol{s}'$:

$$P(\boldsymbol{s}') = c\Omega_{\boldsymbol{r}}(\boldsymbol{s}'). \tag{2.3}$$

Although we do not have enough information to evaluate absolute probabilities, we can now compare the relative likelihood of different microstates.

$$\frac{P(\boldsymbol{s})}{P(\boldsymbol{s}')} = \frac{\Omega_{\boldsymbol{r}}(\boldsymbol{s})}{\Omega_{\boldsymbol{r}}(\boldsymbol{s}')}. \tag{2.4}$$

To transform the above into a more manageable form, we introduce the Boltzmann Entropy:

$$S(\boldsymbol{S}) \coloneqq k \log \Omega(\boldsymbol{S}), \tag{2.5}$$

where k is Boltzmann's constant, a scaling factor from the microscopic to macroscopic. Another crucial definition is that of *macrostate*: a collection of microstates, $\boldsymbol{S} \subset \mathcal{S}$, indistinguishable to the experimental observer. $\Omega(\boldsymbol{S})$ is the number of microstates corresponding to a macrostate, $\Omega(\boldsymbol{S}) = |\boldsymbol{S}|$. We interpret $S(\boldsymbol{S})$ as a measure of uncertainty: higher entropy gives a lower probability of correctly guessing the true microstate the system occupies.

Returning to the task at hand, we can apply our equation for entropy to the reservoir into equation (2.4). Then, we get:

$$\frac{P(\boldsymbol{s})}{P(\boldsymbol{s}')} = e^{\frac{1}{k}(S_{\boldsymbol{r}}(\boldsymbol{s}) - S_{\boldsymbol{r}}(\boldsymbol{s}'))}. \tag{2.6}$$

[2] Let us consider how $\boldsymbol{s}$ and $\boldsymbol{r}$ might exchange energy physically. If $\boldsymbol{s}$ is a metal and $\boldsymbol{r}$ a gas, then the two will transfer energy whenever gas particles collide against the metal, exchanging energy stored in the metal's vibrations with the kinetic energy of moving gas particles.

By our assumption that the reservoir is much larger than $\mathcal{S}$, we can approximate the above using the second law of thermodynamics, $T\Delta S \approx \Delta E$ (constant volume and number of particles), to get:[3]

$$\frac{P(\boldsymbol{s})}{P(\boldsymbol{s}')} = e^{-\beta(E_{\boldsymbol{s}}(\boldsymbol{s})-E_{\boldsymbol{s}}(\boldsymbol{s}'))}, \tag{2.7}$$

where $\beta := 1/(kT)$, the thermodynamic beta. This requires that $\mathcal{S}$ and $\mathcal{R}$ are in thermal equilibrium, i.e. their temperatures are the same (for a more thorough justification of these steps, see section A.2). Separating (and by the fact that $\boldsymbol{s}$ and $\boldsymbol{s}'$ are independent), we get that:

$$P(\boldsymbol{s}) \propto e^{-\beta E(\boldsymbol{s})}. \tag{2.8}$$

By normalizing (solving $\sum_{\boldsymbol{s}} P(\boldsymbol{s}) = 1$), we get an equation for the probability distribution, having used only the fundamental assumption of statistical mechanics.[4]. We have derived the Boltzmann distribution:

$$\boxed{P(\boldsymbol{s}) := \frac{1}{Z} e^{-\beta E(\boldsymbol{s})} \qquad Z := \sum e^{-\boldsymbol{E}(\boldsymbol{s})}} \tag{2.9}$$

where Z is the normalizing factor, *the partition function*. It holds for any system $\boldsymbol{s}$ so long as $\boldsymbol{s}$ exchanges only energy with its surroundings. Together, these conditions—fixed temperature, number of particles, and volume—form the *canonical ensemble*. We have found a relation between the energy of a microstate and its probability. If we specify an energy function, we can calculate probabilities and, from those probabilities, our desired measurement outcomes. In section 3.1, we will see this equation show up again. With the appropriate choice in energy function, equation (2.9) characterizes certain neural networks, and much of our subsequent analysis will translate readily.

Ising Model. An example of a system with a suitable energy function for the Boltzmann distribution is the Ising model depicted in figure 2.1. In this model, we require that the microscopic degrees of freedom s_i are binary-valued ($s_i \in \{1, -1\}$), and we call these degrees of freedom, *spins*. The Ising model was conceived as a minimal model for *ferromagnetism*, the phenomenon by which metals form permanent magnets. In nature, spin is an intrinsic property of particles that induces and interacts with magnetic fields. Though it is a quintessentially quantum effect, we can approximate spin classically as orienting either "up" or "down" ($+1$ and -1 respectively).

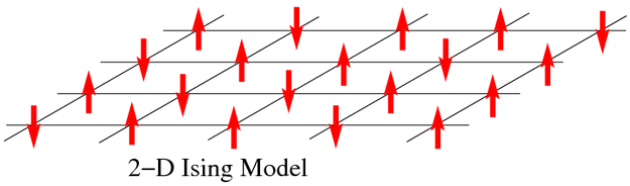


Figure 2.1: The Ising model is a minimal model for ferromagnetism. Image from [13].

[3]For other statistical ensembles, we might include other terms, like the number of particles.

[4]The other approximations become exact in the thermodynamic limit that the number of particles goes to infinity, section A.2.

We write the following *Hamiltonian* (energy function) for the Ising model:

$$E(\boldsymbol{s}) \coloneqq -B \sum_i s_i - J \sum_{\langle i,j \rangle} s_i s_j, \tag{2.10}$$

where, in the ferromagnet, B is the external magnetic field, J is the interaction energy between neighboring pairs of spins, and $\sum_{\langle i,j \rangle}$ denotes a sum over adjacent sites. We see that the system is in a lower energy state when spins s_i align with B and their neighbors $J \sum_{j \to i} s_j$, where $j \to i$ denotes the neighbors of i.

Intractable sums. For the Ising model in one and two dimensions, we can plug this Hamiltonian into equation (2.9) and derive exact solutions for expectation values. Unfortunately, for the vast majority of conceivable Hamiltonians, equation (2.9) is intractable. This stems from Z, the *partition function*. For an Ising magnet with N spins, Z will contain 2^N terms. Beyond around $N = 300$, this exceeds the number of atoms in the universe [14]. In the standard thermodynamic limit that N goes to infinity, this diverges. Even in everyday (finite) life, N is on the order of Avogadro's number, already an incredibly large number. How are we to proceed? To complicate matters further, equation (2.1) requires another sum of 2^n terms. It turns out that, with regard to this last quandary, Z will be our saving grace. Z, more than *just* a normalizing factor, contains all the relevant information about our system. From Z, we can determine any desired macroscopic parameters of interest by taking derivatives. For example, if we define the free energy, $F \coloneqq -\beta \ln Z$, then for the Ising model, our expectation for the magnetization will be $\langle M \rangle = \partial_B F$ (see section A.3), where $M(\boldsymbol{s}) \coloneqq \sum_i s_i$, the net orientation of all spins.

For most models of interest, then, our only option is approximation. One class of possibilities is Markov Chain Monte Carlo (MCMC) techniques. Rather than evaluate our sums and averages over all microstates, $\mathcal{S}$, we evaluate these over a representative, finite sets of samples, $\mathcal{S}_{data}$ (see figure 2.2). The trick in MCMC is that relative probabilities, $P(\boldsymbol{s}')/P(\boldsymbol{s})$, are much easier to evaluate than absolute probabilities, $P(\boldsymbol{s})$, since the partition functions cancel. Monte Carlo techniques proceed according to some variation of:

1. Initialize a random state, $\boldsymbol{s}$.
2. Consider a small variation to the state, $\boldsymbol{s}'$ (for example, by flipping spin s_i).
3. Decide whether to accept this variation according to $P(\boldsymbol{s}')/P(\boldsymbol{s})$.
4. Repeat steps 2 and 3 until $\boldsymbol{s}'$ converges the *equilibrium distribution*, $P(\boldsymbol{s}')$.

With modern computers, we can easily generate and average over many samples using techniques like the Metropolis-Hastings algorithm or cluster methods like the Swendsen-Wang and Wolff algorithms.[5] Crucially, computers cannot simulate infinite lattices, and the behavior of finite lattices will differ considerably from the infinite limit.[6] It was not until relatively recently (on the timeline of

[5] For more, see, for example [15] and [16]

[6] Any finite sum of analytic terms is finite, so in nature, there are no true divergences or critical points (see section 2.3). Fortunately, we can correct for these effects with finite-size scaling theory, a consequence of RG (see section B.1).

statistical physics) that we could generate appropriately large sample sizes. For these reasons, among others, physicists devised a number of other strategies. The first we will consider is mean-field theory (MFT).

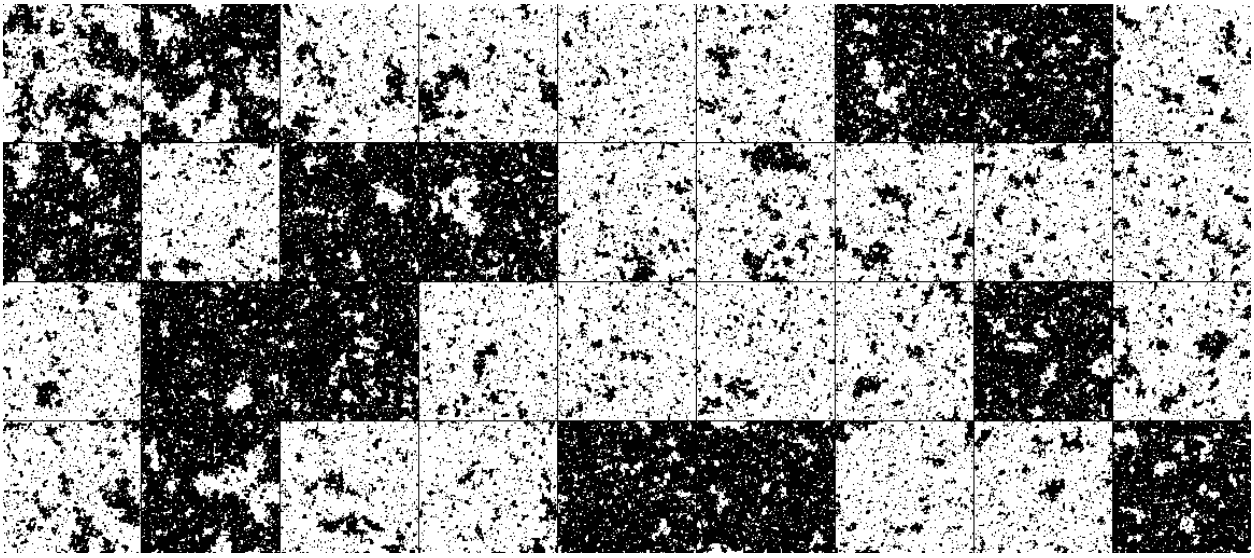


Figure 2.2: Samples of the 2D Ising model near the critical temperature generated with the Swendsen-Wang Algorithm, implemented *rgpy*.

2.2 Mean-Field Theory

To start, let us consider a simpler problem. Given a single spin s_i, and a specification of all other spins, can we calculate its partition function? We can rewrite equation (2.10) more instructively:

$$E(\boldsymbol{s}) = -\sum_i E_i(s_i), \qquad E_i(s_i) \coloneqq -\left(\sum_{j\to i} J s_j + B\right) s_i = -(nJ\langle s_{j\to i}\rangle + B)s_i, \tag{2.11}$$

where n is the number of neighbors of s_i and $\langle s_{j\to i}\rangle \coloneqq \frac{1}{n}\sum_{j\to i} s_j$ is the average spin of s_i's neighbors. Then, the probability over s_i is:

$$P(s_i) = \frac{e^{-\beta(nJ\langle s_{j\to i}\rangle + B)s_i}}{e^{-\beta(nJ\langle s_{j\to i}\rangle + B)} + e^{\beta(nJ\langle s_{j\to i}\rangle + B)}}, \tag{2.12}$$

and

$$\langle s_i\rangle = P(s_i = 1) - P(s_i = -1) = \frac{2\cosh\beta(nJ\langle s_{j\to i}\rangle + B)}{2\sinh\beta(nJ\langle s_{j\to i}\rangle + B)} = \tanh\beta(nJ\langle s_{j\to i}\rangle + B). \tag{2.13}$$

The orientation of a given spin depends on the average orientation of its neighbors, as one might expect.

If we were given $\langle s_{j\to i}\rangle$, this sum is trivial. Just as it is easy to calculate relative probabilities, it is easy to calculate conditional probabilities like $P(s_i|\{s_{j\to i}\})$. Here too, the partition functions cancel: $P(s_i|\{s_{j\to i}\}) = P(s_i, \{s_{j\to i}\})/P(\{s_{j\to i}\})$.[7]

For now, though, these observations are not of much help since we do not know $\langle s_{j\to i}\rangle$. This brings us to the mean-field approximation. Since we could have chosen any spin s_i as our starting point (including its neighbors), we expect, $\langle s_i\rangle \approx \langle s_{j\to i}\rangle$. This is the *principle of mediocrity*. In the mean-field approximation, we assume, more stringently that, $\langle s\rangle \coloneqq \langle s_i\rangle = \langle s_{j\to i}\rangle$, so:

$$\langle s\rangle = \tanh(\beta n J\langle s\rangle + B). \tag{2.14}$$

If we find a solution, then, using $\langle M(\boldsymbol{s})\rangle = \langle \sum_i s_i\rangle = \sum_i \langle s_i\rangle$, we would have our measurement outcome:

$$\langle M(\boldsymbol{s})\rangle = \sum_i \langle s_i\rangle = N\langle s\rangle, \tag{2.15}$$

and we see that $\langle s\rangle$ is nothing more than magnetization per site $m = \langle M\rangle/N$.

It turns out that we cannot solve equation (2.14) analytically (it is a transcendental equation), so we have to resort to numerical techniques. For intuition though, we can get far with a graphical approach (see figure 2.3).

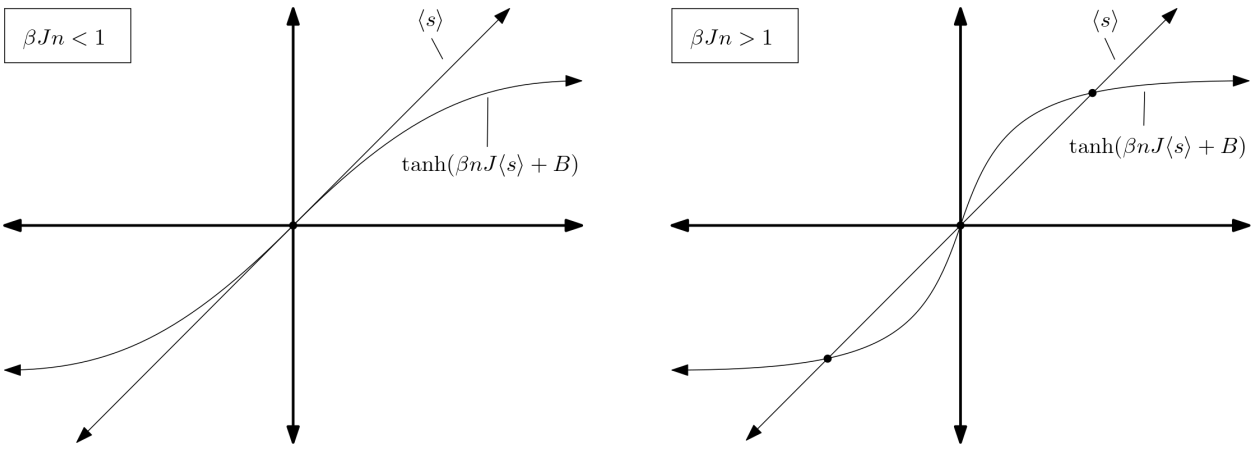


Figure 2.3: Mean-Field Theory predictions for the spontaneous magnetization $M|_{B=0}$.

Restricting to the case that $B = 0$, let us distinguish two cases:

1. $\beta Jn < 1$. There is only one solution: $m = 0$.

2. $\beta Jn > 1$. Suddenly, there are two additional solutions. Something interesting seems to happen at the point $\beta Jn = 1$; we call this a *critical point*, and it occurs at the *critical temperature*, $T_c = Jn/k$.

[7] This avoiding of joint probabilities with clever choices in conditional probabilities will be the basis for efficiently training RBMs in the next section.

Table 2.1: Macroscopic Parameters of the Ising model

Macroparameter	Description
Magnetization, $M \coloneqq \sum_i s_i$.	The strength of the magnet's field.
Spontaneous magnetization, $M\|_{B\to 0}$	Magnetization even in the absence of an external magnetic field.
Zero-field susceptibility, $\chi \coloneqq \partial_B M$	How much the magnetization changes for small changes in temperature.
Energy, $\langle E \rangle$	The average energy of our system.
Specific heat, $C \coloneqq \partial_T \langle E \rangle$	How much the average energy changes for small changes in temperature.
Correlation length, ξ	The average distance across which spins are correlated.

Taylor-expanding the right side of equation (2.14), we get that, in the vicinity of the critical point:

$$m \begin{cases} = 0 & T > T_c \\ \sim \pm(3|t|)^{-1/2} & T < T_c, \end{cases} \tag{2.16}$$

where t is the *reduced temperature*, $t \coloneqq (T - T_c)/T_c$. In fact, these equations contain more information about our system than the magnetization alone. We can use mean-field theory to derive other parameters like the magnetic susceptibility and specific heat (see table 2.2, some are particular to Ising-like models, others are more general). Furthermore, our discussion assumed an arbitrary number of neighbors, n, so we would expect this to hold for any number of dimensions. It seems we have accomplished our goal for this chapter a full section in advance.

Unfortunately, in less than four dimensions, mean-field theory provides incorrect predictions: equation (2.16) is wrong. Intuitively, systems with less than four dimensions have too little order for MFT to hold. More dimensions mean more paths between any two spins and more correlation between them. Past the *critical dimension* of four, there is enough order for our MFT approximation to hold. We have to resort to a different approach.

2.3 Critical Phenomena and the Renormalization Group

Although the quantitative predictions of the mean-field theory are incorrect, its qualitative predictions are instructive and its predictions about the existence of a critical point particularly so. From equation (2.16), we expect a phase transition between a paramagnetic, disordered phase, and a ferromagnetic, ordered phase (in which the system spontaneously magnetizes). This bears out experimentally though not at the predicted temperature, see figure 2.4.

Correlation Length. An important quantity in table 2.2 is the correlation length, ξ. This is the average distance across which spins in our lattice tend to fluctuate together. Spins farther apart than

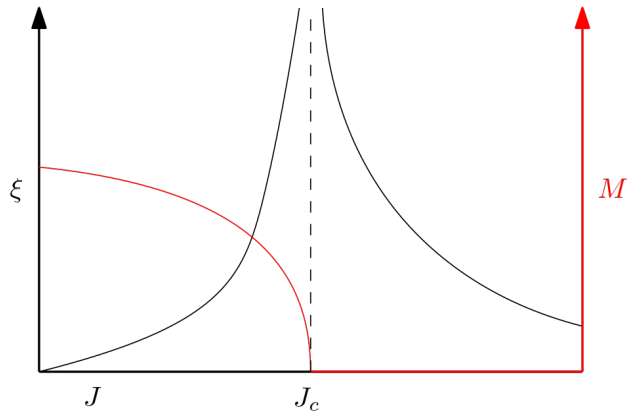


Figure 2.4: The qualitative behavior of the correlation length, ξ, and magnetization, M, around the critical point.

Table 2.2: Critical Exponents of the Ising Model.

Critical Exponent	Macroparameter	Power Law
α	Specific heat C	$C \sim A\|t\|^{-\alpha}$
β	Spontaneous magnetization M	$\lim_{B\to 0} M \propto \|t\|^{\beta}$
γ	Zero-field susceptibility χ	$\xi \equiv \partial_B M\|_{B=0} \propto \|t\|^{-\gamma}$
δ	Magnetization M	$M \propto \|B\|^{1/\delta}\|_{t=1}$
ν	Correlation length, ξ.	$\xi \propto \|t\|^{-\nu}$

ξ are effectively independent of one another, so severing such a connection has no appreciable effect on the macroscopic properties: we can think of ξ as a measure of how macroscopic our system is [12]. Mean-field theory predicts that, near the critical point, the correlation length scales as:

$$\boxed{\xi \sim |t|^{-1/2}} \tag{2.17}$$

Although the exponent does not line up with experimental results (the true value is 1), mean-field theory correctly predicts that ξ diverges at the critical point (see figure 2.4). In fact, this is the defining characteristic of critical points. When the correlation length diverges the entire system becomes correlated. Any perturbation, no matter how infinitesimal, will have macroscopic ramifications. For the statistical physicist, critical points are excellent places to test theories as they allow closer access to the microscopic realm.

Critical Exponents. Another valuable prediction of mean-field theory is that of *critical exponents*. We see from equation (2.17) and equation (2.16) that, near the critical point, the correlation length and the magnetization obey simple power-laws. These are examples of a more general trend: near critical points, macroparameters will follow power-law scaling formulas. We call the exponents that define these relations *critical exponents* (see table 2.3)

We shift our goal to the (correct) calculation of these critical exponents. To this end, we turn to the renormalization group, a set of ideas for tackling precisely these critical phenomena.

The Renormalization Group. Instead of trying to compute Z head-on, let us consider a different angle. We will try to re-express Z with a simpler set of parameters while preserving the physical, long-distance information. Repeating these transformations, we will discard more and more irrelevant, microscopic fluctuations, keeping only the macroscopic information. Formally, an RG transformation will look something like:

$$\sum_{\boldsymbol{s}'} e^{-H'(\boldsymbol{s}')} = \sum_{\boldsymbol{s}} e^{-H(\boldsymbol{s})}, \tag{2.18}$$

constraining for example $|\boldsymbol{s}'| < |\boldsymbol{s}|$. We consider H' and H parameterized with sets of couplings $\{K'\}$ and $\{K\}$, for example,

$$H(\boldsymbol{s}) = -\sum_i K_i^{(1)} s_i - \sum_{\langle i,j\rangle} K_{ij}^{(2)} s_i s_j - \sum_{\langle\langle i,j\rangle\rangle} K_{ij}^{(3)} s_i s_j \ldots, \tag{2.19}$$

where $\langle\langle i,j\rangle\rangle$ denotes next-nearest neighbors, and the continued sum will, in general, contain all possible interactions. For $K_i^{(1)} = \beta B$, $K_{ij}^{(2)} = \beta J$, we recover the Ising Hamiltonian.

Alone, equation (2.18) is not enough of a requirement. A "good" RG transformation satisfies a special set of criteria: it should preserve long-distance information while discarding short-distance information. Arbitrary transformations satisfying equation (2.18) need not extract the information we deem *relevant*. Formally, we are interested in extracting the *relevant operators*, those describing macroscopic properties, and suppressing the *irrelevant operators*, those describing microscopic properties. For example, we might accomplish this transformation by summing over even spins (known as *decimation*):

$$e^{-H'(\boldsymbol{s}')} = \sum_{s_2,s_4,\ldots,s_N} e^{-H(\boldsymbol{s})}, \tag{2.20}$$

where now $\boldsymbol{s}'$ ranges over the odd spins. In other words, we *integrate out* or *marginalize over* the short-distance degrees of freedom.

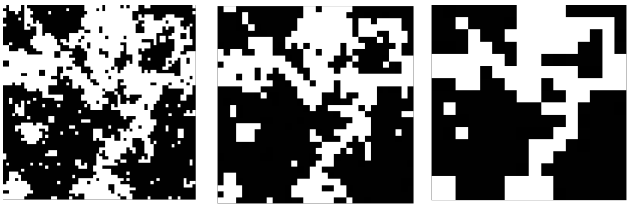


Figure 2.5: Three steps of majority-rule block-spin renormalization, preceding left to right (block size $b = 2$).

Consider, first, a descriptive example: (majority-rule) block-spin renormalization, a set of RG techniques intended for lattice systems. For a given microstate, block renormalization proceeds as follows, (see figure 2.5):

1. We partition the configuration into non-overlapping blocks. For each block, we determine which spin is in the majoriy, and we assign that value to a new, single spin. These define a new coarse-grained system.

2. We rescale the coarse-grained configuration so that each block takes the size of an original spin.

Formally, we can write the block transformation rule as:

$$e^{-H'(\boldsymbol{s}')} \coloneqq \sum_{\boldsymbol{s}} \prod_{\text{blocks}} \pi(s'; s_i) e^{-H(\boldsymbol{s})}, \tag{2.21}$$

where π is the projection operator implementing the majority rule

$$\pi(s', s_1, \ldots, s_9) \coloneqq \begin{cases} 1, & \text{if } s' = \operatorname{sgn} \sum_i s_i \\ 0 & \text{otherwise.} \end{cases} \tag{2.22}$$

Though we can easily perform this procedure for any individual configuration, equation (2.21) requires us to do this for all microstates, and this remains intractable. Ultimately, for most systems, RG will use variational schemes [12]. It is, however, the qualitative insights RG offers, in spite of these approximations, which merit our immediate attention.

Consider what happens when we apply block renormalization to Ising configurations at different temperatures as in figure 2.6. We see three trajectories of block renormalization for three different initial temperatures: below the critical point, at the critical point, and above. Our first observation is that these transformations induce flows away from the critical temperature. There are three fixed points ($T = 0$, $T = T_c$, and $T = \infty$) for the Ising model, and, indeed, macroscopically these correspond to the three phases (ordered, critical, and disordered).

Let us be more exact and determine how this behavior might arise. We will write the RG transformation rule as a function $\mathcal{R}$ of couplings: $\{K'\} = \mathcal{R}(\{K\})$. With two general assumptions, we can build a descriptive theory of RG. First, we assume that there exists a fixed point (or multiple). These are specifications of $\{K^*\}$ that are stable under our transformation rule, namely, $\{K^*\} = \mathcal{R}(\{K^*\})$. Second, we assume we can differentiate this transformation near the fixed point, so we can linearize:

$$K'_a \sim K^*_a + \sum_b \boldsymbol{\mathcal{J}}_{ab}(K_b - K^*_b), \tag{2.23}$$

where $\mathcal{J} = \frac{\partial K'_a}{\partial K_b}|_{K=K^*}$. This is the Jacobian, a generalization of the derivative to vector-valued functions of multiple variables. We denote $\boldsymbol{\mathcal{J}}$'s left eigenvectors e^i, corresponding to eigenvalues, λ^i so that

$$\sum_a e^i_a \boldsymbol{\mathcal{J}}_{ab} = \lambda^i e^i_b.^{8} \tag{2.24}$$

Next, we define *scaling variables*, $u_i \coloneqq \sum_a e^i_a (K_a - K^*_a)$, which are combinations of deviations $K_a - K^*_a$ that transform multiplicatively near the fixed point:

$$u'_i = \sum_a e^i_a (K'_a - K^*_a) = \sum_{a,b} e^i_a \boldsymbol{\mathcal{J}}_{ab}(K_b - K^*_b) \tag{2.25}$$

$$= \sum_b \lambda^i e^i_b (K_b - K^*_b) = \lambda^i u_i. \tag{2.26}$$

[8] There is no reason to assume $\boldsymbol{\mathcal{J}}$ to be symmetric or even to have real eigenvalues though we will restrict ourselves to considering real-eigenvalued Jacobians.

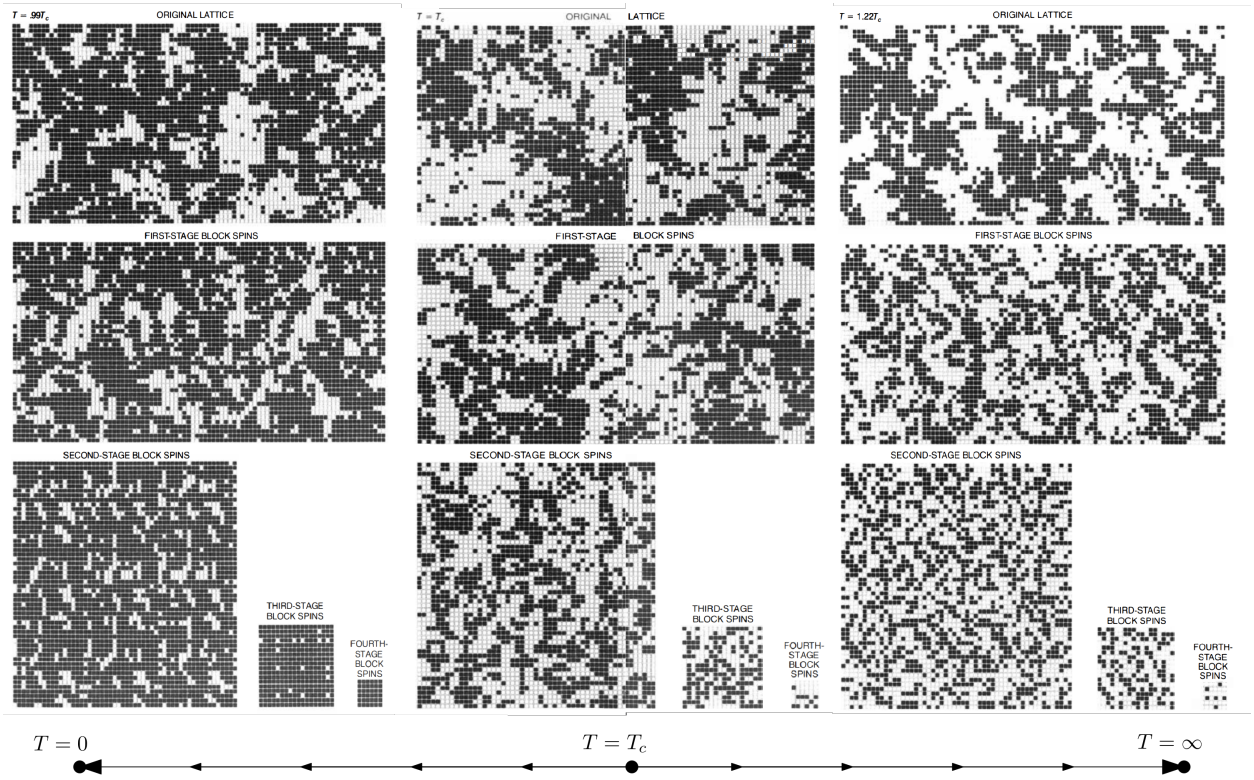


Figure 2.6: Renormalization induces changes in the effective temperature towards fixed points. Images from Wilson [17].

For later convenience, we introduce $\lambda_i \equiv b^{y_i}$, where b is the rescaling size (the width of the blocks in block-spin RG). These y_i are the *renormalization group eigenvalues* and are distinguished in three cases:

- $y_i > 0$: u_i is *relevant*. Repeated RG iterations drive u_i away from its fixed point value.
- $y_i < 0$: u_i is *irrelevant*. Repeated RG iterations drive u_i towards 0.
- $y_i = 0$: u_i is *marginal*. The linearized equations are not enough to tell us about u_i's behavior.

From this, we see that our ability to distinguish between microscopic and macroscopic is a consequence of simple dimensional analysis: there are finitely many relevant eigenvalues. Those are the ones you see when you zoom out far enough. The irrelevant eigenvalues span a *critical surface* of points which, under RG transformations, are attracted towards the critical point. Macroscopically, points on this hypersurface are indistinguishable, and their behavior is fully characterized by the critical point alone. This brings us to the remarkable principle known as *universality*.

Universality. In the last century, experimentalists faced a puzzling situation. In their efforts to measure more precise critical exponents for all manner of systems, they discovered that their experimental set-ups did not matter: all ferromagnets for a given number of dimension possessed the same critical exponents and so too for all superfluids [11]. The exponents are *universal*. The

Ising model, then, is not only a minimal model for ferromagnetism, but it can describe fluids, neural networks, metal alloys, and more.

Another key result is that we can express all of our critical exponents in terms of the relevant RG eigenvalues. First, we use RG to derive a scaling rule for the free energy (see section B.2). Then, from its derivatives, we can determine the critical exponents which even allows us to relate the exponents to one another in *scaling relations*. These relations had been postulated before the advent of RG but many only as inequalities. RG provided a rigorous means to link these different exponents.

For example, we show a derivation in (see section B.3) for the correlation length critical exponent:

$$\boxed{\nu = \frac{1}{y_t}} \tag{2.27}$$

where y_t is the thermal RG eigenvalue, which is related, as its name suggests, to the temperature of the system. From the exact solution (of the Ising model in 2D), we get that $y_t = 1$. Then, we derive for the correlation length critical exponent:

$$\boxed{\nu = 1} \tag{2.28}$$

In general, we do not have access to solutions like these. Therefore, we consider three approximate schemes.

$4 - \epsilon$ Expansion. We can rephrase the thermodynamics limit (in which we let the number of spins go to infinity) as the limit in which we hold the total size of the system fixed while letting the distance between spins go to zero. In this way, our discrete lattice becomes a continuous field, and we can formulate the Ising model as a quantum field theory (QFT). Whereas the above are examples of renormalization in real-space, in QFT, we typically perform renormalization in momentum-space. Here, we can use Wilson's $4 - \epsilon$ expansion, treating the dimensionality of our system as a perturbation [18]. Then, with perturbation theory and diagrammatics, we approximate the values of our scaling variables.

Monte Carlo Simulation. RG allows us to take advantage of the finite-size effects that dominate Monte Carlo techniques, and we can predict how our results will deviate from the infinite-size limits (see section B.1). We can combine the two approaches, performing the RG sums, like equation (2.21), over Monte Carlo samples.

Kadanoff's Variational Technique. Another approximation scheme is that of Kadanoff. He writes the renormalization transformation as:

$$e^{-H'(s')} = \sum_s e^{\boldsymbol{T}_\lambda(s',s) - H(s)}, \tag{2.29}$$

where T_λ a function coupling the original and coarse-grained systems. Kadanoff derives upper- and lower-bound estimates of the free energy that depend on T_λ. By choosing λ to tighten the

bounds Kadanoff variationally minimizes the difference in free energy between the initial and coarse-grained systems, $\Delta F = F(H') - F(H)$ (knowing the margin of error). However, this technique does not guarantee reasonable estimates of macroparameters. As Kadanoff himself observed:

> Hopefully, one might obtain good results for physical quantities by choosing the upper (lower) bound recursions that give the smallest error in the free energy... We say "hopefully" because usually one is not interested in the free energy itself. Rather its derivatives are of the major physical interest. Since the variational principles pertain to the free energy, there is no guarantee that the derivatives will be accurate [19].

This reflects one of the major challenges with RG. It is by no means "easy" to construct adequate RG schemes, and findings are rarely backed with rigorous justification. The details of appropriate transformations depend on the systems under investigation and require an amount of intuition on the part of the physicists [5]. We will see that, ultimately, being more precise in defining physically-relevant information will let us circumvent this problem, formulating, a system-independent criterion of quality for RG transformations.

Chapter 3

Machine Learning in Physical Investigations

In the previous chapter, we began with the goal of translating microscopics to macroscopics. Where MFT failed, RG made sense of critical phenomena, revealing universality, simple scaling laws, critical exponents, and relations between them. In this chapter, we turn to machine learning. Like RG, ML is a blanket term that refers to a wide range of techniques. Both involve the iterative manipulation of information with the goal of extracting "relevant" information. However, we will see that ML is more flexible in its definition of relevant. Whereas in statistical physics, relevant is synonymous with long-distance, in ML, relevance will depend on the problem at hand. We will also see that the level at which RG and ML manipulate information is different. Where statistical physics works with partition functions, ML works with probabilities. In practice, many of the challenges (namely, intractable sums), but also solutions, are the same. In spirit, however, this distinction reveals something fundamental about the types of challenges that characterize physics and ML.

In physics, our investigations are largely reductionist: we *begin* with a Hamiltonian, which, in turn, defines a partition function, and our aim is to predict something about large sets of microstates. To this end, computational techniques, like MCMC algorithms, may use $P(\boldsymbol{s})$ (more precisely, the relative probability) to generate a finite set of samples $\mathcal{S}_{data}$ that, we hope, is representative of $\mathcal{S}$, the true, complete set of microstates. By this, we mean that the statistical properties of $\mathcal{S}_{data}$ and $\mathcal{S}$ should converge as we increase the number of samples. In ML, the investigation often works in the opposite direction: we are given some $\mathcal{S}_{data}$ and assume it is representative of $\mathcal{S}$. Then, our goal is to learn something about $P(\boldsymbol{s})$. While physics concerns properties of $\mathcal{S}$, ML concerns properties of $\boldsymbol{s}$. These are not absolute distinctions, but it will be valuable to bear in mind.

Let us introduce some of the essentials of ML and show its value for statistical physics through a practical example. We follow the treatment of Hinton [20] as well as Mehta et al.'s *A high-bias, low-variance introduction to Machine Learning for physicists* [21]. We refer the interested reader to these sources for elaboration where our analysis goes quickly.

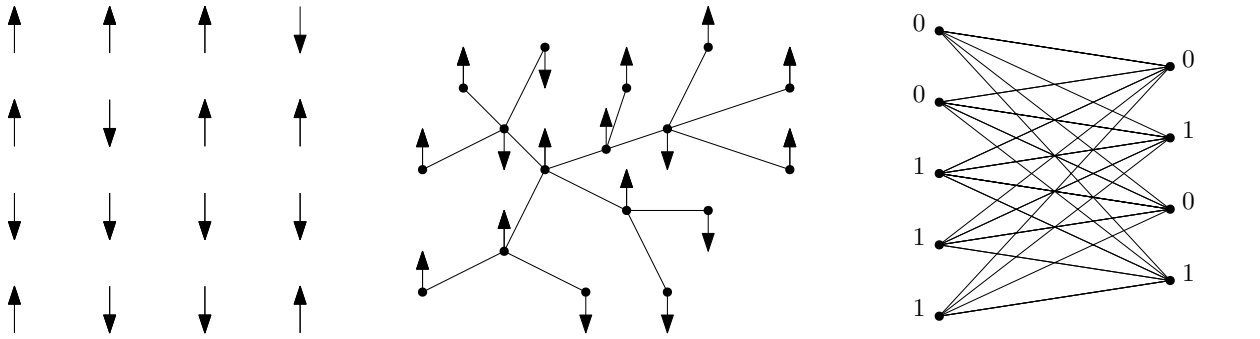
3.1 Phase Classification

Our first question, as physicists, is the following: can we train a neural network to classify the likely phase of some Ising configuration, $\boldsymbol{x}$? This immediately reflects the different spirits of ML and

physics we discussed above: our goal is to learn something about the microstate $\boldsymbol{s}$, whereas in the previous chapter we cared only for $\mathcal{S}$. Formally, our aim is to learn the conditional probability distribution $P(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{y}$ encodes the phase.

We start with a dataset. In this case, we assume we have access to a large collection of Ising configurations at different temperatures and phases, $\mathcal{X}_{data} \coloneqq \{\boldsymbol{x}\}$, as well as their phase labels $\mathcal{Y}_{data} \coloneqq \{\boldsymbol{y}\}$. We often distinguish two kinds of ML: *supervised* and *unsupervised* learning.[1] Our access to labels, $\mathcal{Y}$, means this example of classification falls under supervised learning.[2] In contrast, unsupervised learning works without labels, typically at the level of $P(\boldsymbol{x})$, detecting patterns in the raw data itself.

Before doing anything else, we randomly divide $\mathcal{X}_{data}$'s elements into a training set, $\mathcal{X}_{train}$, from which our network learns, and a testing set, $\mathcal{X}_{test}$, on which we test the trained model's results. This division is crucial because, in ML, we want our results to generalize to new samples that may not have been in our dataset $\mathcal{X}_{data}$ but that could have come from $\mathcal{X}_{true}$. A central problem in ML is the *bias-variance tradeoff*. An algorithm may *overfit* the training set which compromises its ability to generalize to new data, or it might fail to learn enough detail, performing poorly on both training and testing data (low-bias/high-variance and high-bias/low-variance, respectively). By splitting the dataset into a testing and training set, we get a good impression of the bias and variance for the models we trained. Often, we will further split the training set into cross-validation sets, trying out different kinds of models, ultimately keeping those that best accomplish a balance of low bias and variance.



The Ising Model | The Hopfield Model | Restricted Boltzmann Machine

Figure 3.1: Restricted Boltzmann machines are a binary-valued two-layer neural network. RBMs trace their origins to the Hopfield model, an early model for associative memory which itself was inspired by the Ising model [22].

A full review of the techniques available in ML is beyond the scope of this capstone. As such, we will focus our attention on one particular class of algorithms, restricted Boltzmann machines (RBMs), bidirectional artificial neural networks for modeling probability distributions. Depicted in figure 3.1, RBMs consist of two layers of binary units: a *visible* layer, $\boldsymbol{v} \coloneqq \{v_i\}_{i=0}^{N}$, and a *hidden*

[1] A third axis of differentiation, reinforcement learning is outside the scope of this capstone.

[2] We could, however, also formulate classification of phases as an unsupervised learning problem. We might try "clustering," where we specify a number of clusters, k, and try to group training examples into as many clusters, according to some notion of similarity. Then, we hope that the clusters our model learns correspond to the phases. We could also use this procedure with different choices of k to determine a likely number of phases.

layer, $\boldsymbol{h} \coloneqq \{h_j\}_{j=0}^{N}$, where $v_i, h_j \in \{0, 1\}$. The visible layer will serve as the input for our data, $\boldsymbol{x} \to \boldsymbol{v}$, and the hidden layer as our prediction, $\boldsymbol{h} \to \boldsymbol{y}$, the label. For the purposes of classifying the phases of the 2D Ising model, it will suffice to consider an RBM with one hidden unit, h. If h is 1, we predict that the configuration is in the ordered phase, and for $h = 0$, we predict disordered. From our exploration in the previous chapter, we could probably come up with some function that performs this prediction. For example, we might sum all of the spins (compute the magnetization), and if the result is close to 0 we know the system is disordered: $h \coloneqq 0$. Otherwise, we would output $h \coloneqq 0$. The power of ML will be to extract rules like these without explicit instructions.

Instead, we teach our model implicitly through the cost function, $\mathcal{C}(P_\theta(y|\boldsymbol{x}), \{\mathcal{X}_{data}, \mathcal{Y}_{data}\})$. This acts as a moderator, and it tells the model how "incorrect" its predictions (labels) are. Formally, the ML goal becomes to find the parameters θ that minimize $\mathcal{C}$. Experience from physics tells us that finding the ground state (global minima) can be really, *really* hard. Instead, we use stochastic gradient descent (SGD). For each θ, we calculate $\partial_\theta \mathcal{C}$, with which we implement the update rule:

$$\theta \to \theta - \eta \partial_\theta \mathcal{C}, \tag{3.1}$$

where η is the *learning rate*. From calculus, we know that the negative gradient of a function is the direction in which that function decreases most rapidly. This update procedure, then, adjusts our parameters so as to reduce the cost. Iterating this procedure, we end up in a local minimum of the cost function (see figure 3.2). In practice, we calculate these gradients, not over the entire data set, but over subsets, *minibatches*. This means the gradients will vary from iteration to iteration (hence, *stochastic*). This is both computationally faster and introduces noise (like temperature) that improves the chances of escaping poor local minima.

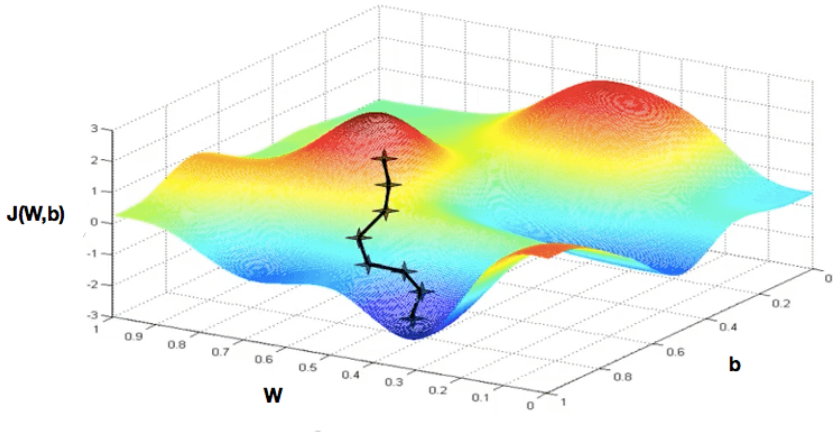


Figure 3.2: Stochastic Gradient Descent (SGD). If we view the cost function as defining an "energy" landscape, SGD allows us to find local minima (stable or metastable) energy states. Image taken from [23].

Now, if we can formulate an adequate prediction function, $P(\boldsymbol{h}|\boldsymbol{v})$, for the RBM, we can improve it with SGD. The crucial step is to define an energy function:

$$\boxed{E_\theta(\boldsymbol{v}, \boldsymbol{h}) \coloneqq -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{ij} w_{ij} v_i h_j} \tag{3.2}$$

where $\theta \coloneqq \{\{a_i\}, \{b_j\}, \{w_{ij}\}\}$. Then, we can model the system's joint probability with a Boltzmann distribution:

$$\boxed{P_\theta(\boldsymbol{v}, \boldsymbol{h}) \coloneqq \frac{1}{Z} e^{-E_\theta(\boldsymbol{v},\boldsymbol{h})}, \qquad Z \coloneqq \sum_{\boldsymbol{v}',\boldsymbol{h}'} e^{E_\theta(\boldsymbol{v}',\boldsymbol{h}')}} \tag{3.3}$$

We note that this is the exact same equation as our Ising model (2.9 and 2.10). Here, $\{a_i\}$ and $\{b_j\}$ take the role of the external magnetic field B, which now varies from site to site (hence the indices i and j). Then, $\{w_{ij}\}$ takes the role of J varying from pair to pair.

Naturally, we run into the same intractability issues. For large enough networks, we cannot evaluate Z. Instead of calculating joint distributions, we pull a trick similar to MFT and MCMC, considering instead conditional and marginal distributions. Due to RBM's bipartite structure these factor and are easy to evaluate. Explicitly, in the case of a single hidden spin h_j, we get (with a bit of algebra):

$$P(h_j|\boldsymbol{v}) = \frac{P(h_j, \boldsymbol{v})}{P(\boldsymbol{v})} = \frac{(e^{-E(\boldsymbol{v},h_j)})/Z}{(\sum_{\boldsymbol{h}'} e^{-E(\boldsymbol{v},\boldsymbol{h}')})/Z} \tag{3.4}$$

$$= \frac{e^{-h_j(\sum_i w_{ij} v_i + b_j)}}{1 + e^{(\sum_i w_{ij} v_i + b_j)}}. \tag{3.5}$$

This conditional probability is itself a Boltzmann distribution, but over only two states (tractable!). We get for the full system:

$$\boxed{P_\theta(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j=1}^{M} \frac{1}{1 + e^{-h_j(\sum_i w_{ij} v_i + b_j)}}} \tag{3.6}$$

and similarly:

$$\boxed{P_\theta(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i=1}^{N} \frac{1}{1 + e^{-v_i(\sum_j w_{ij} h_j + a_i)}}} \tag{3.7}$$

All that remains is for us to choose an adequate cost-function, and we can start training our RBMs. For the example of binary-classification, an appropriate choice is the cross-entropy loss:

$$\mathcal{C}(P_\theta(\boldsymbol{y}|\boldsymbol{x}), \{\mathcal{X}_{data}, \mathcal{Y}_{data}\}) = \sum_{\boldsymbol{x} \in \mathcal{X}_{data}} \sum_{\boldsymbol{y} \in \{0,1\}} P_{true}(\boldsymbol{y}|\boldsymbol{x}) \log P_\theta(\boldsymbol{y}|\boldsymbol{x}), \tag{3.8}$$

see (see section C.1) for elaboration. Differentiating with respect to θ and with some simple algebra, we get the training rule:

$$\partial_{b_j} \mathcal{C} = \sum_{\boldsymbol{x} \in \mathcal{X}_{batch}} P_{true}(\boldsymbol{y}|\boldsymbol{x}) \big[1 - P_\theta(\boldsymbol{y}|\boldsymbol{x})\big] \, (h_j) \,, \tag{3.9}$$

$$\partial_{w_{ij}} \mathcal{C} = \sum_{\boldsymbol{x} \in \mathcal{X}_{batch}} P_{true}(\boldsymbol{y}|\boldsymbol{x}) \big[1 - P_\theta(\boldsymbol{y}|\boldsymbol{x})\big] \, (v_i h_j) \,. \tag{3.10}$$

We have all the necessary elements of a phase classifier: a dataset, a model of $P(\boldsymbol{y}|\boldsymbol{x})$, and a means to train this model. In the next section, we will consider an example more relevant to the statistical physicist: generating samples.

3.2 Generative Modeling: Gibbs Sampling

In section 2.3, we considered how to use MCMC sampling to estimate macroparameters and even critical exponents. Here we will consider an MCMC technique called *Gibbs Sampling* which will use RBMs to generate samples of $P(\boldsymbol{x})$.

Suppose we are given an already trained RBM. Gibbs sampling, like other MCMC techniques, consists of a series of update steps. In one step, we input some initial state, transform it into a hidden state using $P_\theta(\boldsymbol{h}|\boldsymbol{v})$, and transform it back to a new visible state using $P_\theta(\boldsymbol{v}|\boldsymbol{h})$. This process is imperfect, so the output of one step will differ from the input. If we repeat this process many times, then the distribution of the outputs will converge to the equilibrium distribution $P_\theta(\boldsymbol{v})$.

This first requires $P_\theta(\boldsymbol{v})$ to be an appropriate model of $P_{true}(\boldsymbol{v})$. To get to this point, we need to derive a marginal distribution over $\boldsymbol{v}$ from $P_\theta(\boldsymbol{v}, \boldsymbol{h})$. Similar to the conditional probabilities, the architecture of the RBM allows us to perform the marginalization $P_\theta(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P_\theta(\boldsymbol{v}, \boldsymbol{h})$ explicitly. If we write $P_\theta(\boldsymbol{v})$ as a Boltzmann distribution with its own energy $E_\theta(\boldsymbol{v})$,

$$P_\theta(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P_\theta(\boldsymbol{v}, \boldsymbol{h}) \propto e^{-E_\theta(\boldsymbol{v})}, \tag{3.11}$$

then we can express $E_\theta(\boldsymbol{v})$ in terms of $E_\theta(\boldsymbol{v}, \boldsymbol{h})$ (see section D.1):

$$\boxed{E_\theta(\boldsymbol{v}) = -\sum_i a_i v_i - \sum_j \log\left(1 + \exp\left\{-\left(b_j + \sum_i v_i w_{ij}\right)\right\}\right),} \tag{3.12}$$

and we can perform an analogous computation for the marginal distribution over $\boldsymbol{h}$ to get $P_\theta(h)$ as a Boltzmann distribution in terms of energy, $E_\theta(h)$. As in the classification example, we require a cost function, in this case, the Kullback-Leibler divergence (KLD):

$$\mathcal{C}(P_\theta(\boldsymbol{x}), \boldsymbol{x}) = D_{KL}(P_{data}(\boldsymbol{x})||P_\theta(\boldsymbol{x})) \coloneqq \sum_{x \in \mathcal{X}_{data}} P_{true}(\boldsymbol{x}) \ln\left(\frac{P_{true}(\boldsymbol{x})}{P_\theta(\boldsymbol{x})}\right). \tag{3.13}$$

This is closely related to the cross-entropy, and, similarly, it is an important information-theoretic quantity (see section C.2) that provides a notion of similarity for probability distributions. It is 0 if and only if the two distributions are equal:

$$D_{KL}(P_{true}(\boldsymbol{x})||P_\theta(\boldsymbol{x})) = 0 \iff P_{true}(\boldsymbol{x}) = P_\theta(\boldsymbol{x}). \tag{3.14}$$

We claimed that this unsupervised, but in comparing $\mathcal{C}$ in equation (3.13) with equation (3.8), we might interpret $\boldsymbol{x}$ as a kind of label for itself. A more appropriate description is *self-supervised*. We emphasize this point because it has been the source of misunderstanding. ML always requires the specification of a cost function, and even in the absence of labels, the cost function guides how the model learns which information is relevant and irrelevant.

If we differentiate the KLD with respect to θ, then after a bit of algebra, we get:

$$\partial_{a_i} D_{KL} = \langle v_i \rangle_{true} - \langle v_i \rangle_\theta \tag{3.15}$$

$$\partial_{b_j} D_{KL} = \langle h_j \rangle_{true} - \langle h_j \rangle_\theta \tag{3.16}$$

$$\partial_{w_{ij}} D_{KL} = \langle v_i h_j \rangle_{true} - \langle v_i h_j \rangle_\theta \tag{3.17}$$

where $\langle \dots \rangle_{true}$ is an average over the true distribution $P_{true}(\boldsymbol{v})$ and $\langle \dots \rangle_{\theta}$ is an average over the model distribution $P_{\theta}(\boldsymbol{v})$.

If you take away one thing from this capstone, let it be this: sums and averages like these are hard. Naturally, we approximate. Wherever h_j shows up, we use $P(h_j|\boldsymbol{v})$, and for the expectations over P_{true}, we calculate a Monte Carlo average over our dataset, $\mathcal{X}_{data}$. The expectations over P_{θ} are trickier. Fortunately, we can approximate them with Gibbs sampling, initializing with samples from $\mathcal{X}_{data}$. Together, these approximations constitute the *contrastive-divergence* algorithm, see [20]. Having trained our RBM, we can generate new samples and start calculating critical exponents.

RBMs trained in this way have applications other than generation. Consider training these on black and white images: if we constrain the number of hidden nodes to be less than the number of visible nodes, then the hidden layer will learn a compact representation of the input. We can even stack multiple RBMs on top of each other to form *deep Boltzmann machines* (DBMs). In DBMs, the hidden layer of one RBM becomes the visible layer of the next. Trained with contrastive divergence, each layer learns a progressively compacter version of the input. This should remind you of RG. In the next section, we will make these similarities more exact. Ultimately, this allows us to machine learn RG transformations.

Chapter 4

Machine Learning and the Renormalization Group

In the previous chapter, we considered two examples of ML in physical investigations: neural networks used as a phase classifier and MCMC sampler. Consider the superficial similarities between these neural networks and our treatment of RG in section 2.3. The classification example calls to mind the infinite RG limit in which all disordered phases flow to one fixed point and all ordered phases to another. Here, these fixed points would correspond to the values of our label being either 0 or 1. The generative example is more immediately similar, and the very language is analogous: we speak of hidden layers that hierarchically decompose the input visible layer to coarse-grained hidden layers. We might be tempted to ask: does a DBM of, say, five layers learn to implement four iterations of some RG procedure (see figure 4.1)?

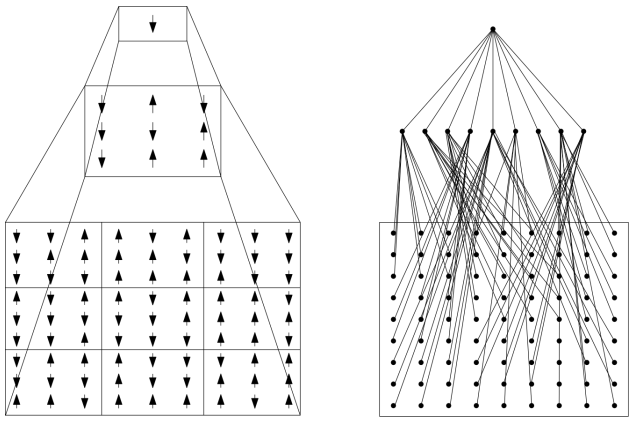


Figure 4.1: Two iterations of block renormalization and a deep Boltzmann machine of three layers.

a

However, questions like this are too general. We have already seen two examples of RBMs used for different goals, and what constituted relevant information differed in either context. In RG, relevant information is understood more narrowly: it is the long-distance information. If we are to make comparisons between RG and RBMs, we must be more specific, and this begins by being

more precise in our definition of "relevant" information. Although Claude Shannon, the founder of information theory, avoided this topic explicitly, Tishby et al. showed that information theory provides a natural and exact formulation of relevant information: relevant information is simply the information contained in one signal, $\boldsymbol{x}$, about another $\boldsymbol{y}$ [3]. For our phase classifier, the relevant information is the information contained in the Ising samples $\boldsymbol{x}$ about the labels $\boldsymbol{y}$. Once we have trained RBMs for compression, the relevant information is the information contained in the hidden layer $\boldsymbol{h}$ about the visible layer $\boldsymbol{v}$. Information theory quantifies how much information is shared between two signals with the mutual information:

$$I(\boldsymbol{x};\boldsymbol{y}) \coloneqq \sum_{\boldsymbol{x},\boldsymbol{y}} P(\boldsymbol{x},\boldsymbol{y}) \log\left(\frac{P(\boldsymbol{x},\boldsymbol{y})}{P(\boldsymbol{x})P(\boldsymbol{y})}\right), \tag{4.1}$$

We see this quantity is minimized ($I(\boldsymbol{x};\boldsymbol{y}) = 0$) when the random variables are independent ($P(\boldsymbol{x},\boldsymbol{y}) = P(\boldsymbol{x})P(\boldsymbol{y})$). It is maximized when the variables are maximally dependent. We have a quantitatively basis for extracting relevant information: maximizing mutual information between appropriately chosen signals.

In the next chapter, we will derive the correct choice for $\boldsymbol{x}$ and $\boldsymbol{y}$ on lattice systems that recovers physically-relevant (i.e. long-distance) information. We then derive a variational proxy to the mutual information that we can differentate and, thus, learn in a neural network.

With the knowledge of an exact formulation of relevant information, we can start investigating links between particular RG implementations and RBMs. Such comparisons will revolve around the question of relevance. What information does a given cost-function deem relevant? Are there examples where this overlaps with our notions of physical relevance? Let us consider a seminal example.

4.1 A Comment on Mehta and Schwab's Equivalence

As a first attempt, we will consider Mehta and Schwab's correspondence between Kadanoff's variational procedure (section 2.3) with RBMs trained by contrastive divergence (section 3.2) [4]. Specifically, we consider the narrower case, of "exact" RG and "exact" RBMs. Exact RG means that Kadanoff's transformation preserves *all* the information contained in the hidden system; i.e. the free energy of the input and coarse-grained systems is exactly the same. Similarly, an exact RBM has learnt to perfectly recreate its inputs $P_{true}(\boldsymbol{x}) = P_\theta(\boldsymbol{x})$ in equation (3.13); the RBM will have reached a global minimum of the cost function.

First, Mehta and Schwab show that, under the above conditions, the two transformations are equivalent under:

$$\boldsymbol{T}_\theta(\boldsymbol{v},\boldsymbol{h}) = -\boldsymbol{E}_\theta(\boldsymbol{v},\boldsymbol{h}) + \boldsymbol{H}(\boldsymbol{v}), \tag{4.2}$$

and we provide the derivation in section D.2.

However, the exact case is not generally possible, and when it is, it is often not particularly interesting. Exact RBM transformations likely mean overfitting which is likely opposed to the aims of the ML investigation. Exact RG transformations are few and far between, so this correspondence

would apply only under narrow circumstances. More interesting then, would be to consider a non-exact comparison.

Here, the two approaches will vary: Kadanoff's variational method operates at the level of free energies while contrastive divergence works at the level of probabilities, and optimizations at these different levels will not necessarily coincide. Still, there may still be qualitative similarities. Pursuing this line of inquiry, Mehta and Schwab train RBMs as described in section 3.2. They find that the RBMs learn to couple hidden units to local blocks of visible units.[1]

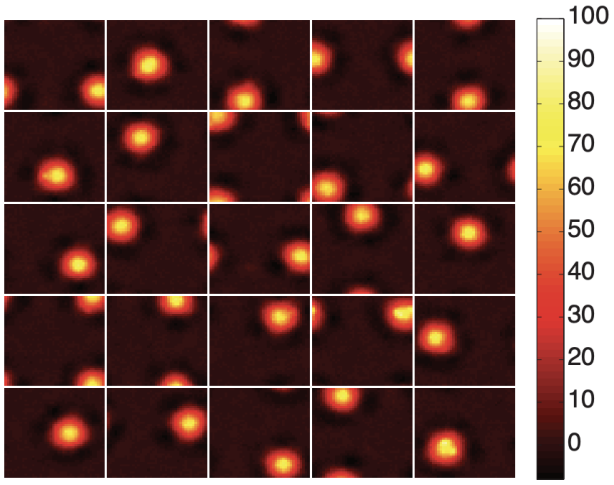


Figure 4.2: Visualization of the receptive filters of the hidden units in one of the layers of the DBM trained by Mehta and Schwab [4]. Each plot corresponds to one hidden unit, and each pixel corresponds to one spin in the initial layer. Higher intensities mean the hidden unit is more coupled to that spin. Indeed, we see that the RBM learns to couple local transformations. However, the transformations do not preserve nearest neighbor relations. This does not matter for the total amount of information stored in coarse-grained layers. In RG, these concerns do matter because they affect the practicality and interpretability of result [5].

Mehta and Schwab write:

> Surprisingly, this local block spin structure emerges from the training process, suggesting the [deep neural network] is self-organizing to implement block spin renormalization [4].

These RBMs may indeed learn a kind of block-spin transformation. However, this block-spin transformation need not be block-spin *renormalization*. As Koch-Janusz and Ringel put it:

> [T]he usefulness (and practicality) of the RG procedure depends on choosing [the transformation] . . . such that the effective Hamiltonian. . . remains as short range as possible. [5]

[1] Crucially, this was not possible without *regularization*, a technique that promotes sparse connections, between hidden and visible units see [21].

More precisely, in the Taylor expansion of our coarse-grain Hamiltonians, the shortest range terms should dominate:

$$H(\boldsymbol{s}) = -\sum_i K_i^{(1)} s_i - \sum_{\langle i,j\rangle} K_{ij}^{(2)} s_i s_j - \sum_{\langle\langle i,j\rangle\rangle} K_{ij}^{(3)} s_i s_j \ldots, \tag{4.3}$$

with $K^{(n)}$ are exponentially suppressed in n. In RG, we place additional constraints on how to organize the information in subsequent layers. Block renormalization procedures, including Kadanoff's technique, respect the symmetries and topology of the system under consideration. The locality of interactions means neighboring visible blocks of spins become neighboring hidden spins. Translational symmetry means the same transformation is applied to each block. Mehta and Schwab's RBMs will not, in general, satisfy these two conditions.

Let us be more exact, rephrasing renormalization in probabilistic terms: we parametrize our RG transformation as the conditional probability distribution $P(\boldsymbol{h}|\boldsymbol{x})$, where $\boldsymbol{x}$ is our initial configuration and $\boldsymbol{h}$ is the hidden or coarse-grained configuration. Locality and translational invariance of the Ising model mean we can factor the RG transformation into a product of local single-block transformations. We divide the system into M blocks, $\boldsymbol{x} \to \{v^{(1)}, v^{(2)}, \ldots, v^{(M)}\}$, with corresponding hidden units $\{h_1, h_2, \ldots, h_M\}$. Then we can factor $P(\boldsymbol{h}|\boldsymbol{x})$ as:

$$P(\boldsymbol{h}|\boldsymbol{x}) = \prod_{j=1}^{M} P(h_j|\boldsymbol{v}^{(j)}). \tag{4.4}$$

Consider what happens, when you permute the units in the hidden layer; i.e., we use a different transformation $P(h_{j'}|\boldsymbol{v}^{(j)})$ where $j \neq j'$. As long as our reverse rule, $P(\boldsymbol{x}|\boldsymbol{h})$ also takes this into account, such a permutation has no impact on the performance of Mehta and Schwab's contrastive divergence trained RBMs. This is *not* the case in RG. There is a unique permutation of hidden layer degrees of freedom which maximizes the above short range condition, and acceptable RG transformations identify this permutation.

Furthermore, the block transformations that the RBM learns may not be the same for each block. Compression and generation are invariant under partial flips $h_j : (0,1) \to (1,0)$, for some fixed j.[2] Blocks of spins which were perfectly correlated in the visible layer may be perfectly anticorrelated in the hidden layer. Although the information content is the same, the hidden layer Hamiltonian will take a more complicated form than is necessary, going against notions of what is a good and practical RG procedure.

Mehta and Schwab fail to mention these conditions, so their suggestion that RBMs learn block-spin renormalization falls short. However, if we make these conditions explicit, we can easily recover a stronger version of their claim. To do this, we introduce convolutional neural networks (CNNs) as depicted in figure 4.3. These are networks with a special architecture: instead of all-to-all couplings, we explicitly couple spins in the hidden layer to blocks in the visible layer. Additionally, CNNs perform the same translation for each block and respect the topology of the input system. Had they required a convolutional architecture upfront, Mehta and Schwab may, indeed, have successfully trained RBMs to learn even block renormalization.

[2] RG transformations are invariant under a collective flip of $\boldsymbol{h}$.

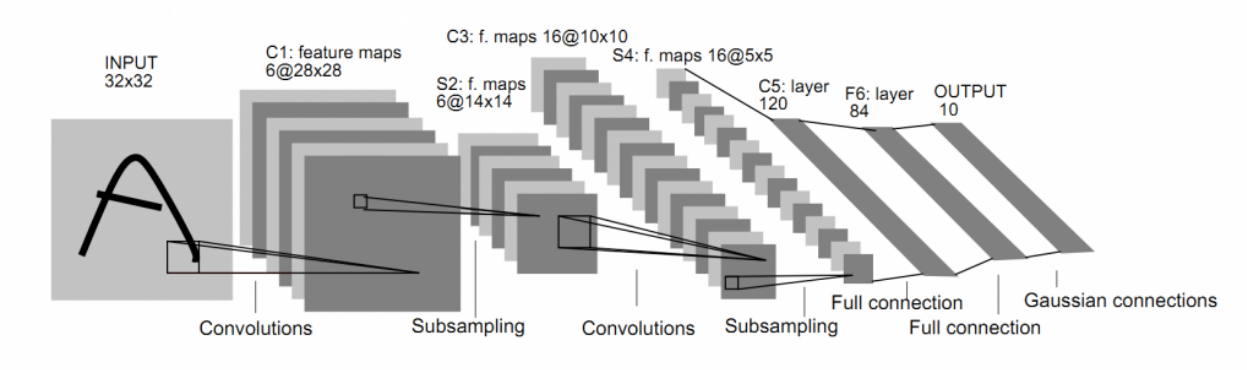


Figure 4.3: This is LeNet-5, an example of a convolutional neural network used for digit recognition and a seminal architecture [24].

From these observations, we can make the link between majority-rule block-spin RG and ML fully precise. In section D.3, we derive a equivalence between RBMs an RG, describing a parameterization which implements the majority-rule: $P(h_j|\boldsymbol{v}^{(j)}) = \operatorname{sgn} \sum_i v_i^{(j)}$.

4.2 Relevant Information

For the exact case discussed by Mehta and Schwab, we preserve the full probability measure which necessarily preserves the long-distance information. However, in the non-exact case, the KLD has no preference for long-distance information over short-distance: all information is valued equally. In the case of compression and generation, this is acceptable: we have no a priori knowledge of which features in the data are more important.

In RG, we have stronger requirements: transformations must favor long-distance information over short-distance information. We can say resolutely: RBMs trained with the KLD (even with the appropriate convolutional architecture) need not learn acceptable RG transformations. Perhaps in the case of the simple Ising model, the KLD is an adequate heuristic. However, we have no guarantee that this extends to novel systems. In fact, Koch-Janusz and Ringel showed that RBMs trained with the KLD on the dimer model, another model from statistical physics, couple to local fluctuations rather than to the correct hidden variables [5]. We recall that appropriate transformations for one system need not translate to others (see section 2.3). Critique of Mehta and Schwab's paper revolves primarily around the KLD being an inappropriate criterion for RG transformations [2, 6].

In reply, Mehta and Schwab point out that Kadanoff's method also offers no guarantee of extracting the correct physical information. We come back to the difficulty of devising appropriate RG techniques, and the need for creativity. In fact, this need for intuition in RG is mirrored quite clearly in ML. Decisions regarding model architectures and cost functions require creativity on the part of the researcher. Both fields would benefit from a clearer understanding of when techniques will and will not work. In the next chapter, we will see that information theory may provide the framework to answer questions like these, and we will encounter a system-independent formulation of RG. This might mean better, more efficient models in both disciplines.

Chapter 5

Renormalization in Information Theory

In the previous chapter, we used information theory to formalize the notion of *relevant* information. In this chapter, we will adapt this to our physical context of long-distance as relevant information. Ultimately, this allows us to devise a cost-function that measures the physical information. From this, we can derive a learning rule to train RBMs to perform *optimal* transformations.

In the previous chapter, we saw that the Ising model's locality and translational invariance conditions reduce the number of acceptable RG procedures. Rather than devise an update rule for how an entire configuration should transform under RG, we can start with equation (4.4). Then, we only need to learn the transformation for a single block, $P(h_j|\boldsymbol{v}^{(j)})$. We already know that we can model conditional probabilities like these with RBMs. This is the first insight of Koch-Janusz and Ringel, and in this chapter, we will follow their information-theoretic treatment [5].

5.1 An Information-Theoretic Formulation of RG

First, we partition a full lattice configuration $\boldsymbol{x}$ into four areas: a visible block, $\boldsymbol{v}$, that is surrounded by, in order, a buffer, $\boldsymbol{b}$, an environment, $\boldsymbol{e}$, and an outer area, $\boldsymbol{o}$ (see figure 5.1). In RG, we introduce a hidden area, $\boldsymbol{h} \coloneqq \{h_i\}$ which is a function of the degrees of freedom in $\boldsymbol{v}$. Our aim is to choose this coupling, $P(\boldsymbol{h}|\boldsymbol{v})$, so that $\boldsymbol{h}$ encodes the long-distance degrees of freedom about our system at large, $\boldsymbol{x}$[1]. By our assumption that $\boldsymbol{o}$ is farther than the correlation length, $\boldsymbol{v}$ contains no information about $\boldsymbol{o}$. We can ignore $\boldsymbol{o}$ in devising $P(\boldsymbol{h}|\boldsymbol{v})$. Furthermore, in coming up with a function for $\boldsymbol{h}$, we can reasonably ignore $\boldsymbol{b}$. The information contained in $\boldsymbol{v}$ about $\boldsymbol{b}$ is likely to be short-range. By eliminating $\boldsymbol{b}$, we throw out the shortest-range fluctuations. That which remains is the physically relevant information; we see it is the information shared between $\boldsymbol{v}$ and $\boldsymbol{e}$. Our goal is to extract this information and encode in $\boldsymbol{h}$. We choose the parameters, Λ, of our RBM that models $P(\boldsymbol{h}|\boldsymbol{v})$ to maximize the mutual information between $\boldsymbol{h}$ and $\boldsymbol{e}$:

$$I(\boldsymbol{h};\boldsymbol{e}) = \sum_{\boldsymbol{h},\boldsymbol{e}} P(\boldsymbol{h},\boldsymbol{e}) \log\left(\frac{P(\boldsymbol{h},\boldsymbol{e})}{P(\boldsymbol{h})P(\boldsymbol{e})}\right), \tag{5.1}$$

[1] We have dropped the indices for convenience, and now we let block $\boldsymbol{h}$ consist of multiple coarse-grained spins.

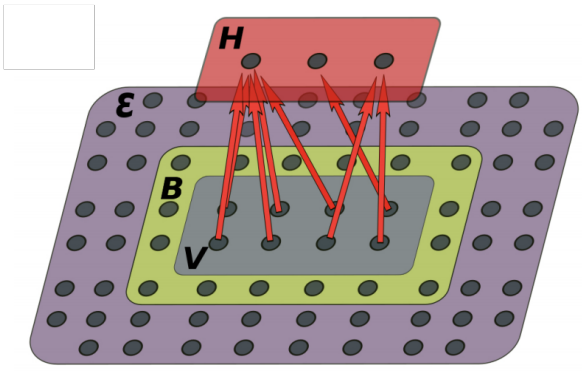


Figure 5.1: The RSMI algorithm partitions configurations into four regions, a visible block $\boldsymbol{v}$, buffer zone $\boldsymbol{b}$, environment $\boldsymbol{e}$ and outer zone, $\boldsymbol{o}$. We introduce a coarse-grained block of spins $\boldsymbol{h}$.

where these probabilities are defined as marginalizations over $P(\boldsymbol{x})$:

$$P(\boldsymbol{h}, \boldsymbol{e}) = \sum_{\boldsymbol{v}} P(\boldsymbol{h}|\boldsymbol{v}) P(\boldsymbol{v}, \boldsymbol{e}) \tag{5.2}$$

$$P(\boldsymbol{h}) = \sum_{\boldsymbol{v},\boldsymbol{e}} P(\boldsymbol{h}|\boldsymbol{v}) P(\boldsymbol{v}, \boldsymbol{e}) \tag{5.3}$$

$$P(\boldsymbol{v}, \boldsymbol{e}) = \sum_{\boldsymbol{b},\boldsymbol{o}} P(\boldsymbol{x}) \tag{5.4}$$

$$P(\boldsymbol{e}) = \sum_{\boldsymbol{b},\boldsymbol{v},\boldsymbol{o}} P(\boldsymbol{x}). \tag{5.5}$$

From previous chapters, we know that the probability measure $P(\boldsymbol{x})$ and its marginalizations are generally intractable. The key insight of Koch-Janusz and Ringel is that we can use RBMs not only to model $P(\boldsymbol{h}|\boldsymbol{v})$ but also to model these other distributions. Koch-Janusz and Ringel ultimately use three RBMs: one for $P(\boldsymbol{v}, \boldsymbol{e})$, another for $P(\boldsymbol{v})$, and finally the $P(\boldsymbol{h}|\boldsymbol{v})$ RBM already mentioned. The other distributions are calculated as MC-averages over the dataset. Hereby, Koch-Janusz and Ringel derive a proxy to the mutual information (see section E.1) which they can optimize with stochastic gradient descent.

To validate their ideas, they provide both experimental and theoretical justification. For the 1D Ising model, these marginalizations can be calculated exactly, and they show that maximizing the RSMI rederives decimation, an RG procedure known to be *optimal* for the 1D Ising model in that the procedure does not increase the range of the coarse-grained Hamiltonian. In follow-up research, Lenggenhager et al. (along with the aforementioned authors) show that RSMI coarse-graining is more generally optimal, maintaining short-distance in any number of dimensions. They further show this holds even in some cases that the mutual information is not fully saturated. We direct the interested reader to [5] and [9] for these results. In the remainder of this capstone, we discuss our own implementation and generalization.

Measuring Critical Exponents Having trained an RBM according to the RSMI algorithm, we can use finite-size scaling (see section B.1) to estimate, finally, critical exponents. An important detail will be to devise a "thermometer" that can measure the effective temperature at successive RG-iterations. We discuss several options in section E.2. These options, moreover, are intrinsic and do not require explicit knowledge of the temperature. This means the RSMI algorithm is fully unsupervised. If we encounter new systems where we do not know the proper RG transformations (or even how to measure "temperature"), then unsupervised approaches might save us considerable headache. Rather than guess and check possible transformations, we would first machine learn a renormalization procedure on the system, then, observe and interpret, only later attempting more calculation-heavy approaches. As Koch-Janusz and Ringel put it, the RSMI algorithm could form the basis of the "physical reasoning process" itself [**?**]. Instead of retroactively explaining trends in our data, ML would proactively guide our exploration.

Chapter 6

Results: Machine Learning Critical Exponents

In [10], we release *rgpy*, an open-source library for implementing ML-based renormalization group techniques. This is still in its infancy, and development is ongoing. As of now, it contains a full-stack realization of the RSMI algorithm implemented in Tensorflow: i.e., the package includes implementations of various MCMC techniques (Metropolis-Hastings, Swendsen-Wang, and Wolff algorithms), the RBMs necessary for the RSMI algorithm, and an implementation of standard block-spin renormalization. We also provide a host of already-generated samples at various lattice sizes and temperatures. We invite the reader to explore and try out these tools.

6.1 A Novel Calculation of ν for the 2D Ising Model

Our exploration of statistical physics and ML was centered around the Ising model and its macroparameters and critical exponents. The calculation of these exponents served as unifying thread, and as validation of the RSMI algorithm, we provide the following approximation of ν:

$$\boxed{\nu \approx 0.79 \pm 0.39} \tag{6.1}$$

For how we calculated this, see section E.3. This is not, by any means, an improved calculation. However, it is a highly promising first result. The quality of this calculation was ultimately limited by both time and hardware constraints. With more of both, the prediction should eventually converge to the right quantity. In fact, this result is just the start. Our investigation revealed a large number of possible improvements and generalizations. Due to the time contraints of this capstone, we have not yet implemented these ideas, but, all the same, these ideas merit attention. The generalization we discuss gives rise to a family of RSMI-inspired techniques, together constituting a new branch of RG methods.

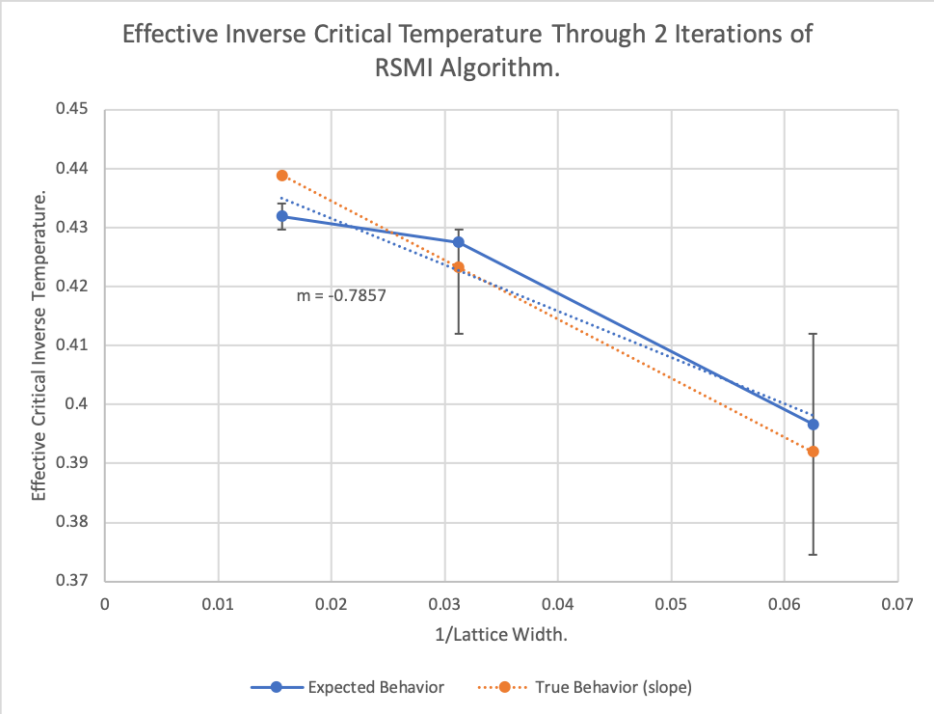


Figure 6.1: The finite-size scaling curve for the correlation length critical exponent.

6.2 A Generalization to n-Spin and $O(n)$ Systems

Koch-Janusz and Ringel claim the RSMI algorithm works for general lattice systems. However, the current formulation works only for systems with binary degrees of freedom: spin-$1/2$ Ising models. The authors avoid explicitly generalizing these results. Though the generalization is straightforward, it is important enough to warrant elaboration.

Let us consider systems with either n spins or n components of spin, then, the visible units of our RBM should be able to take n values. Accomplishing this is quite simple: we let our visible units become vectors with n components: $v_i \to \{v_{id}\}_{d=1}^{n}$. We say that v_i takes the value of spin d when:

$$v_i \equiv d \iff \begin{cases} v_{id} = 1 \\ v_{id'} = 0, d' \neq d. \end{cases} \tag{6.2}$$

In fact, we will use the RBM to model a slightly different vector:

$$v_{id} = P(d), \tag{6.3}$$

where the vector is normalized $\sum_d v_{id} = 1$. Such a vector that contains probabilities of classes is called a *one-hot encoding*, and with these probabilities, we can randomly sample values of spin to generate n-valued spins. For n-vector models, we leave the probabilities as they are. To allow our RBMs to produce these encoding, we have to introduce the index d in the parameters of our model:

$$a_i \to a_{id} \tag{6.4}$$

$$w_{ij} \to w_{ijd}. \tag{6.5}$$

To derive a one-hot encoding, we change the RBM's reverse update rule, $P(\boldsymbol{v}|\boldsymbol{h})$ (3.7). The conditional probabilities will still factor, but normalization over the scalar values $v_i = 0, 1$, becomes normalization over the components of the vector, d. If we denote $\epsilon_i \coloneqq -(\sum_j w_{ij}h_j + a_i)$, then, whereas before, we had:

$$P(v_i|\boldsymbol{h}) = \frac{e^{-\epsilon_i v_i}}{\sum_{v_i' \in \{0,1\}} e^{-\epsilon_i v_i'}}, \tag{6.6}$$

now we have, $\epsilon_{id} \coloneqq -(\sum_j w_{ijd}h_j + a_{id})$, so:

$$P(v_{id}|\boldsymbol{h}) = \frac{e^{-\epsilon_{id} v_{id}}}{\sum_{d=1}^{n} e^{-\epsilon_{id} v_{id}'}}. \tag{6.7}$$

For $n = 2$, we rederive the previous rule with a suitable choice in w_{id}, a_{id}. For reasonable values of d (anything we might encounter in practice), we can perform these calculations explicitly. If we apply this change to each of the RBMs in the RSMI algorithm, we will be able to model n-spin and $O(n)$ models. Note that since we did nothing to the hidden variables, these will still be either binary or Bernoulli-valued. If we require that the hidden spins take the same form as the input variables, as we might demand with RG, then we also need to apply a completely analogous change to the hidden unit of the $P(h_j|\boldsymbol{v}^{(j)})$ RBM, and we can apply the RSMI algorithm to generic lattice systems.

There is, in fact, more room for generalization. Consider an arbitrary Hamiltonian for a system with binary-valued data $H(\boldsymbol{x})$. If we Taylor expand this function, we would get:

$$H(\boldsymbol{x}) = a + \sum_i b_i x_i + \sum_{ij} c_{ij} x_i x_j. \tag{6.8}$$

These are *all* the terms in the expansion.[1] Then, barring the bipartite structure, the RBM energy function already has the most general form we could possibly consider. However, when we make the switch to continuous or n-ary valued data, this is no longer true: the Taylor-expansions become infinite. In these cases, we might consider energy functions with different, possibly higher-order interactions. As long as we maintain the bipartite structure, we can ensured that conditional probabilities factor. Then, we can use variants of contrastive-divergence to efficiently train these models. We need not even require that $P(\boldsymbol{v}|\boldsymbol{h})$ and $P(\boldsymbol{h}|\boldsymbol{v})$ take the same form. The first might be Gaussian and the second a gamma distribution. Formally, we can consider the extension of RBMs into the *exponential family*, and for a full discussion of the possibilities, we refer the reader to [25]. Suffice to say, there are more powerful options for modeling probability distributions. Though the current choices suffice for the spin-$1/2$ Ising model, these extensions may prove necessary in investigating more complicated systems.

For all the exact results of Koch-Janusz and Ringel, their practical implementation uses a proxy which does not fully capture the above mutual information [5]. Here we see two possibilities. First, we can improve Koch-Janusz and Ringel's approximation. The authors use a cumulant expansion

$$\langle \exp\{K(\boldsymbol{x})\} \rangle = e^{\sum_{\kappa=0}^{\infty} \frac{1}{\kappa!} C_\kappa}, \tag{6.9}$$

[1] For $p > 1$, $x_i^p = 0$ or 1.

with cumulants expressed in terms of the moments

$$C_1 = \langle K \rangle, \tag{6.10}$$
$$C_2 = \langle K^2 \rangle - \langle K \rangle^2, \tag{6.11}$$
$$C_3 = \langle K^3 \rangle - 3\langle K^2 \rangle \langle K \rangle + 2\langle K \rangle^3, \tag{6.12}$$
$$\tag{6.13}$$

and so on. In deriving the proxy, only the first term is kept (see section E.1). For better results, a first improvement could be made by introducing additional terms.[2] Second, we can consider other approximations of the mutual information, and we draw the reader's attention to a sampling of options in the literature: contrastive predictive coding [26], Deep InfoMax [27], and Mutual Information Neural Estimation [28]. All three examples use mutual information as the basis for powerful unsupervised techniques. Though we avoid details, similarities (and differences) with the RSMI algorithm warrant further investigation.

Perhaps the most exciting future course of action is extending these ideas to momentum-space renormalization. Rather than focus on lattice models, our goal would be to solve quantum field theories. A first attempt might use the quantum-to-classical mapping of the path integral formalism [5]. More powerful however would be to use the same physical and information-theoretic arguments to devise conditions equivalent to equation (5.1) from first fundamentals.

[2] We may even be able to avoid this expansion altogether, see section E.1.

Chapter 7

Discussion and Conclusions

Let us summarize what we have accomplished. We began overviewing the basic elements of statistical physics which has the goal of turning microscopic theories into concrete macroscopic observables. One quickly runs into a problem: intractable probabilites. We sampled a host of techniques that sidestep this fundamental problem. MCMC techniques avoid absolute probabilities with iterations of relative probabilities. Mean field theory uses a clever constraint on conditional distributions. RG creates new probability distributions, iteratively simpler, keeping the relevant long-distance information. After an introduction to ML, we noticed a resemblance between neural networks and RG. We identified a need for a more precise notion of "relevant" information, which one formalizes with information theory. At the level of "relevant" information, ML and RG appeared to behave similarly, extracting relevant information and suppressing irrelevant information. However, we saw that RG contained a more narrow conception of relevance: long-distance.

We evaluated Mehta and Schwab's seminal comparison of Kadanoff's RG and CD-trained RBMs. We identified a flaw in one of their claims: a "good" RG, should uncover compact, short-range hidden representations and respect the original system's symmetries, and RBMs do not necessarily favor these unless explicitly told to. By being more exact in formulating conditions, requiring convolution archictures, we derived an exact correspondence between majority-rule block-spin renormalization and restricted Boltzmann machines.

Using the information-theoretic formulation of relevance, one can derive a system-independent RG criterion that *optimally* satisfies the short-distance condition. We implemented these ideas in *rgpy*, performing a recalculation of the critical length correlation exponent. This library is available to everyone, and we will continue developing it into the future. We encourage the reader to try this out, and we hope to enable many more calculations of critical exponents. To kickstart this project, we provided an overview of possible improvements and generalizations: the next likely targets are the XY-model and the spin-1 Ising model.

We place special emphasis on the role physical reasoning played throughout our exploration. Though information theory presented us a formal notion of relevant information, it was physical reasoning that led us to an appropriate RG condition. By explicitly thinking about the properties physical systems possess, such as locality and translational invariance, we managed to peak deeper into "black box" neural networks than may otherwise have been possible. We quote Koch-Janusz and Ringel, "the internal data representations discovered by suitably designed algorithms are not just

technical means to an end, but instead are a clear reflection of the underlying structure of the physical system" [5]. Integrating physical and ML perspectives proves a promising basis for a better insight into these algorithms. Not to mention, these techniques are unsupervised, so they should translate readily to problems other than the Ising model.

Comparisons and integrations of ML and RG are at an early phase, and there remains much to be uncovered. It seems that information theory is the appropriate framework with which to lay these links. Within physics, a better understanding of this area may guide research into new systems (on disordered, glassy systems and quantum field theories), though the ramifications extend much further [5]. Ultimately, the synthesis of ML, physics, and information theory may teach us a thing or two about why these techniques work as well as they do. Until then, the links should inspire powerful new techniques in both of these disciplines.

Acknowledgements

I would like to thank Marcos Crichigno for supervising this capstone. I do not think either of us knew what we were getting into when I first approached him, and I am stunned at how much I have been able to learn during this time. Marcos provided guidance, much-needed prodding, and a watchful eye, keeping the focus on the big picture. The topics were new to both of us, ML for him and RG for me, and getting to integrate these different perspectives was a wonderful experience. I would also like to thank my dad, Jiri Hoogland, for helpful feedback on drafts and, most of all, for his expertise on on all things Monte Carlo. After being stuck with implementation details of the RSMI algorithm for two weeks, his words finally let the penny drop. As a last note, I would like to thank my friends and (other) family for putting up with me during the last weeks (when I was a little less than sociable), in particular, Nick van der Woude, for mutually infectious enthusiasm.

Appendix A

Basics of Statistical Physics

The proceeding derivations follow Cardy [12] and Domb [11].

A.1 Measurements as Averages

Suppose we want to test whether our theory, the Ising model, lines up with experimental results: what are the kinds of predictions we can make? Our first attempt might be to measure the system's magnetization, rather appropriate as this is the defining characteristic of "magnets." Before we do, however, let us be more precise. What is a measurement, specifically for magnetization? The magnetization, M, of any one microstate, $\boldsymbol{s}$ is simply the sum of the orientations of all spins, $s_i \in \boldsymbol{s}$ in a given state:

$$M(\boldsymbol{s}) = \sum_i s_i, \tag{A.1}$$

Generally, the microscopic state of a system is changing rapidly, and the system will explore many different microstates over human time-scales. Our measuring devices have finite resolution in time, so, in a laboratory, we cannot perform sums over individual microstates. A measurement becomes the average over the microstates visited during the measurement period, weighted by the time spent in each state. In statistical physics, we often work in the limit that the update time goes to zero compared to the time of measurement. In this light, the notion of a probability of a microstate is also the fraction of time the system spends in that microstate over an infinite duration. Our measurement outcome, $\mathcal{M}$, is an expectation value, $\langle M(\boldsymbol{s})\rangle$, the average over the set of all microstates, $\mathcal{S} = \{\boldsymbol{s}\}$, weighted by the amount of time spent in each state, $P(s)$:

$$\mathcal{M} = \langle M(\boldsymbol{s})\rangle = \sum_{\boldsymbol{s}\in\{\boldsymbol{s}\}} P(\boldsymbol{s})M(\boldsymbol{s}). \tag{A.2}$$

The same holds for any measurable quantity. Given some function of a microstate, $A(\boldsymbol{s})$, the macroscopic value it corresponds to is its expectation value over all microstates, $\langle A(\boldsymbol{s})\rangle$. Common examples (not restricted to the Ising model) include $E(\boldsymbol{s})$, the energy, $P(\boldsymbol{s})$, the pressure, and $V(\boldsymbol{s})$, the volume. If we can find a probability measure over microstates, we can use the above to calculate any macroparameter we choose.

A.2 Thermal Equilibrium and the Second Law of Thermodynamics

Consider the first law of thermodynamics (differential form with only one kind of particle):

$$dE = TdS - PdV + \mu dN, \tag{A.3}$$

where E is the energy, T the temperature, P the pressure, V the volume, μ the chemical potential, and N the number of particles.

For the canonical ensemble (our example), dV and dN are 0. Then, this reduces to:

$$\frac{1}{T}dE = dS. \tag{A.4}$$

From equation (2.6), we have $\Delta S = S_{\boldsymbol{r}}(\boldsymbol{s}) - S_{\boldsymbol{r}}(\boldsymbol{s}')$. If the reservoir is much larger than $\boldsymbol{s}$, this is a tiny difference, so $\Delta S \approx dS$. Then, we get $\Delta S \approx \frac{1}{T}(dE)$. Note that this requires the reservoir and system to have the same temperature (i.e., to be in thermal equilibrium, $T_{\boldsymbol{r}} = T_{\boldsymbol{s}}$). We can use the same approximation to expand this out: $dE \approx E_{\boldsymbol{r}}(\boldsymbol{s}) - E_{\boldsymbol{r}}(\boldsymbol{s}') = \Delta E$. We end up with equation (2.7).

Note that in the thermodynamic limit, $|\boldsymbol{s}|, |br| \to \infty$ while maintaining the inequality $1 \ll |\boldsymbol{s}| \ll |\boldsymbol{r}|$, these statements are precise (this is simply the central limit theorem).

A.3 Observables as Derivatives of the Partition Function

Let us define the free energy, $F = -\frac{1}{\beta}\ln Z$, where $\beta = kT$, the so-called thermodynamic beta. Consider an energy function with a dependence on parameter A according to:

$$E(\boldsymbol{s}) = E_0(\boldsymbol{s}) + \lambda A(\boldsymbol{s}) \tag{A.5}$$

Then, $\langle A \rangle = \partial_\lambda F$ The derivation proceeds as:

$$\begin{aligned}
\langle A(\boldsymbol{s}) \rangle &= \partial_\lambda \left(-\frac{1}{\beta} \ln Z \right) && \text{(A.6)} \\
&= -\frac{1}{\beta Z} \partial_\lambda Z && \text{(A.7)} \\
&= -\frac{1}{\beta Z} \sum_{\boldsymbol{s}} e^{-\beta E(\boldsymbol{s})} \partial_\lambda \left(-\beta E(\boldsymbol{s}) \right) && \text{(A.8)} \\
&= \frac{1}{\beta Z} \sum_{\boldsymbol{s}} e^{-\beta E(\boldsymbol{s})} \left(\beta A(\boldsymbol{s}) \right) && \text{(A.9)} \\
&= \sum_{\boldsymbol{s}} \frac{e^{-\beta E(\boldsymbol{s})}}{Z} A(\boldsymbol{s}) && \text{(A.10)} \\
&= \sum_{\boldsymbol{s}} P(\boldsymbol{s}) A(\boldsymbol{s}). && \text{(A.11)}
\end{aligned}$$

Consider an example, taking the Ising Hamiltonian (2.10):

$$\partial_B \left(-\log Z\right) = -\frac{1}{Z}\sum_{\boldsymbol{s}} \partial_B e^{-B\sum_i s_i - J\sum_{\langle i,j\rangle} s_i s_j} \tag{A.12}$$

$$= -\sum_{\boldsymbol{s}} \frac{e^{-E(\boldsymbol{s})}}{Z}(-\sum_i s_i) \tag{A.13}$$

$$= \sum_{\boldsymbol{s}} P(\boldsymbol{s}) M(\boldsymbol{s}) \tag{A.14}$$

$$= \langle M(\boldsymbol{s})\rangle. \tag{A.15}$$

Two others:

$$\chi = \partial_B \mathcal{M} = \partial_B^2 F \tag{A.16}$$

$$E = \partial_\beta F. \tag{A.17}$$

Appendix B

Scaling and Renormalization

B.1 Finite-Size Scaling Analysis

This section follows [29]. In our theoretic discussion, we have typically considered infinite lattices: for a fixed lattice size L, we have let the microscopic size a (the lattice width) go to zero. Then, the dimensionless $N = L/a$ (the number of lattice sites in a given dimension) diverges; this is the classical thermodynamic limit. Of course, truly physical systems are finite with correspondingly non-singular partition functions. For any such systems, there can be no divergences. An important concern is, then, determining under which conditions we can treat our systems as infinite and under which conditions finite-size effects have a non-neglibile effect.

An important tool we use to study critical systems is MCMC simulations: here, we quickly run into computational limits on the sizes of lattices we can simulate. Especially for MCMC techniques, the role of finite-size scaling is crucial.

Consider a d-dimensional Ising model infinite in all directions. If we were to decrease N of one dimension to some finite value, the system's critical behavior begins to be dominated by a $d-1$ dimensional critical point. This is an example of crossing-over behavior. By controlling some relevant parameter we change the effective dimensionality of our system and navigate between different critical points.

An example of this finite-size behavior is that of the correlation length, ξ. It will no longer diverge and depending on the boundary conditions will experience a peak at either above or below the infinte critical point. For cyclical boundary conditions, the effective critical temperature will increase, as the system is more ordered with more paths between spins. For zero-field boundary conditions, the temperature will decrease as there less paths between spins.

We have seen from table 2.3, that diverging quantities $A(t, N^{-1} = 0)$ scale as $|t - t_c|^{-\zeta}$ for some critical exponent ζ. In the large N-limit, we expect that the system will continue to behave as such, provided that N is much greater than the system's characteristic length scale, the correlation length, $\xi \sim |t - t_c|^{-\nu}$. This amounts to

$$A(t, N^{-1}) \sim |t - t_c|^{-\zeta} \sim \xi^{\zeta/\nu} \qquad (N \gg \xi, t \to t_c). \tag{B.1}$$

The system's geometry will act as a cutoff: rather than diverge, now, as $\xi \to N$, A will behave as

$$A(t, N^{-1}) \sim N^{\zeta/\nu} \qquad (N \ll \xi, t \to t_c). \tag{B.2}$$

Together, these considerations give rise to the finite-scaling ansatz:

$$A(t, N^{-1}) \sim \xi^{\zeta/\nu} \phi((N\xi)^{-1}) \qquad (N^{-1} \to 0, t \to t_c), \tag{B.3}$$

where

$$\phi(x) \{ \begin{cases} = \text{const.} & \text{for } |x| \gg 1 \\ \sim x^{\zeta/\nu} & \text{for } |x| \to 0 \end{cases}. \tag{B.4}$$

$phi(x)$ is a scaling function that controls the finite-size effects. By convention, we choose the scaling function $\tilde{\phi}(x) = x^{-\zeta}\phi(x^{\nu})$. Then, we get that

$$A(t, L) = L^{\zeta/\nu} \tilde{\phi} \left(L^{1/\nu}(t - t_c) \right), \qquad (L \to \infty, t \to t_c), \tag{B.5}$$

with

$$\tilde{\phi}(x) \begin{cases} = \text{const.} & \text{for } x \to 0 \qquad (L \ll \xi), \\ \sim L^{-\zeta/\nu}(\varrho - \varrho_c)^{-\zeta} & \text{for } |x| \gg 1 \qquad (L \gg \xi). \end{cases} \tag{B.6}$$

.This is valuable because it allows us to plot experimental data $a_{exp.}(t, L)$ collected at some temperature and lattice size, on a single curve:

$$\phi(L^{1/\nu}\tilde{(t - t_c)}) = L^{-\zeta/\nu} A(t, L) \tag{B.7}$$

.Plotting $L^{1/\nu}(t - t_c)$ against $L^{-\zeta/\nu} A(t, L)|_{a_{exp.}(t,L)}$, our experimental data will "collapse" onto a single line. In practice, it is enough to consider only the effective critical temperature at different length scales. Plotting this against $\log L$, the points should fall onto a single line with slope ν.

B.2 Scaling Rules for the Free-Energy

In this section, we will derive a transformation rule for the free energy. From section A.3, we know that we can use this transformation rule to determine our critical exponents of interest.

Starting with 2.18, we can intuit that the free energy will take a form similar to:

$$f(\{K\}) = g(\{K\}) + b^{-d} f(\{K'\}). \tag{B.8}$$

we expect a function similar to our starting point but of the updated coupling $f(\{K'\})$ and where we have rescaled the system by (for example, a block-size) b. For d dimensions, we rescale in each direction. Furthermore, we could have some (analytic) function of our coupling at a given instance, $g(\{K\})$.

The free energy of new system is equal to that of the original system with some rescaling, plus the addition of a constant term. Note that this is an inhomogenous transformation. However, for *singular* behavior (near the critical point), we only care about the second term. $g()$ originates from

summing over short degrees of freedom, and it should be an analytic (non-divergent) function of K_a even at critical point.

Considering just the transformation rule for the singular part:

$$f_s(\{K\}) = b^{-d} f_s(\{K'\}). \tag{B.9}$$

We use our knowledge that there are are only two relevant scaling variables, u_t and u_h (for the interested reader, we refer to Cardy [12]). These possess important symmetries: u_t corresponds to the even subspace of couplings (invariant a collective sign-flip $\boldsymbol{s} \to -\boldsymbol{s}$), and u_h to the odd subspace (equivariant with the collective sign-flip). Let us reexpress the above equation in terms of our scaling variables. First, we Taylor-expand the scaling variables[1]. By these symmetry arguments, odd terms vanish from u_t and even terms from u_h. Looking at the formula for u_i, we see they must vanish at $t = h = 0$.

$$u_t = \frac{t}{t_0} + O(t^2, h^2) \tag{B.10}$$

$$u_h = \frac{h}{h_0} + O(th) \tag{B.11}$$

Near the critical point, we take these scaling variables to be proportional to h and t[2]. Combining our equations for the scaling variables, (B.10) and (B.11), with the singular free energy transformation:

$$f_s(u_t, u_h) = b^{-d} f_s(b^{y_t} u_t, b^{y_h} u_h). \tag{B.12}$$

If we repeat this transformation n times, we get:

$$f_s(u_t, u_h) = b^{-nd} f_s(b^{ny_t} u_t, b^{y_h} u_h). \tag{B.13}$$

We now choose an arbitrary point $u_{t0} = |b^{ny_t} u_t|$. This constrains our n to not be too large that the linear approximation breaks down. Solving for n we get:

$$n = \frac{1}{y_t} \log_b \left(\left| \frac{u_t}{u_{t0}} \right|^{-1} \right). \tag{B.14}$$

Plugging equation (B.14) into equation (B.13):

$$f_s(u_t, u_h) = \left| \frac{u_t}{u_{t0}} \right|^{\frac{d}{y_t}} f_s\left(\pm u_{t0}, u_h \left| \frac{u_t}{u_{t0}} \right|^{-\frac{y_h}{y_t}} \right). \tag{B.15}$$

Rewriting in terms of t and h, where we incorporate u_{t0} into a new scale factor t_0 (i.e., $\frac{t}{t_0} \leftarrow \frac{t}{u_{t0} t_0}$):

$$f_s(t, h) = \left| \frac{t}{t_0} \right|^{d/y_t} \Phi\left(\frac{\frac{h}{h_0}}{|t/t_0|^{y_h/y_t}} \right) \tag{B.16}$$

[1] This is valid since we have already limited ourselves to the immediate vicinity of the critical point.

[2] In more complicated systems, and those with tricritical points, our relevant variables will not necessarily be directly proportional to the experimental variables (the knobs) we control.

We see that introducing this cutoff u_{t0} has allowed us to express the (approximately) a function of two variables, f_s, in terms of just one variable in the *scaling function* Φ.

This scaling function may appear to include a u_{t0} dependency, but since the l.h.s. does not, this will cancel. These scaling functions turn out to be universal, it only depends on the system through t_0 and h_0.

B.3 The Correlation Length Critical Exponent

As an example of using free energy scaling to solve critical exponents, let us attempt correlation length critical exponent. Consider the two-point correlation function:

$$G(r_1 - r_2, H) \equiv \langle s(r_1)s(r_2\rangle_H - \langle s(r_1)\rangle_H \langle s(r_2)\rangle_H \tag{B.17}$$

Our first step will be to express this in terms of (derivatives of) the free energy. We introduce a non-uniform magnetic field to our Hamiltonian:

$$H \to H - \sum_r h(r)s(r) \tag{B.18}$$

and differentiate the free energy ($f = \ln Z$) twice:

$$\begin{aligned}
G(r_1 - r_2, H) =& \frac{\partial^2}{\partial h(r_1)\partial h(r_2)} \ln Z(\{H\})|_{h(r)=0} && \text{(B.19)}\\
=& \frac{\partial}{\partial h(r_1)} \left(-\frac{1}{Z} \sum_s s(r_2) e^{H(s) - \sum_r h(r)s(r)} \right)|_{h=0} && \text{(B.20)}\\
=& -\frac{1}{Z^2} \sum_{s'} s'(r_1) e^{H(s') - \sum_r h(r)s'(r)} \sum_s s(r_2) e^{H(s) - \sum_r h(r)s(r)} && \text{(B.21)}\\
& - \frac{1}{Z} \sum_s s(r_1)s(r_2) e^{H(s) - \sum_r h(r)s(r)}|_{h=0} && \text{(B.22)}\\
=& -\left(\frac{1}{Z} \sum_{s'} s'(r_1) e^{H(s')} \right) \left(\frac{1}{Z} \sum_s s(r_1) e^{H(s)} \right) + \frac{1}{Z} \sum_s s(r_1)s(r_2) e^{H(s)} && \text{(B.23)}\\
=& \langle s(r_1)s(r_2)\rangle_H - \langle s(r_1)\rangle_H \langle s(r_2)\rangle_H. && \text{(B.24)}
\end{aligned}$$

We suppose that $h(r)$ varies over scales much larger than block size ba. Applying block renormalization we can effectively ignore the varying of $h(r)$ within a given block. The, for a specified block, it transforms as a uniform field would. The renormalization Hamiltonian should be of the same form.

$$H'(s') - \sum_{r'} h'(r')s'(r') \tag{B.25}$$

Since RG preserves the partition function, we can write:

$$\frac{\partial^2 \ln Z'(h')}{\partial h'(r_1')\partial h'(r_2')} = \frac{\partial^2 \ln Z(h)}{\partial h'(r_1')\partial h'(r_2')} \tag{B.26}$$

Now the l.h.s. is just the correlation function of the RG system with the new Hamiltonian. In terms of our original correlation function, we have to rescale the distance by a factor b.

$$G((r_1 - r_2)/b, H') \tag{B.27}$$

Onto the r.h.s. If we make a change in $h'(r')$, we will change *all* the spins contained in that block:

$$\delta h(r_i) = b^{-y_h} \delta h'(r_1'), \tag{B.28}$$

so this reduces to:

$$b^{-2y_h} \langle (s_1^{(1)} + s_2^{(1)} + \dots)(s_1^{(2)} + s_2^{(2)} + \dots) \rangle_H. \tag{B.29}$$

Each block contains b^d spins, so expanding this gives us b^{2d} 2-point correlations. For our assumption that $r = |r_1 - r_2|$ is large, each will give about the same result. For isotropic systems (respecting rotational symmetries of the lattice), the correlation function will only depend on distance between points, not on orientation.

We end up with the transformation rule:

$$G((r_1 - r_2)/b, H') = b^{2(d-y_h)} G(r_1 - r_2, H). \tag{B.30}$$

For simplicitly, we set $h = 0$ and iterate n times:

$$G(r, t) = b^{-2(d-y_h)} G(r/b, b^{y_t} t) = b^{-2n(d-y_h)} G(r/b^n, b^{ny_t} t), \tag{B.31}$$

stopping at some fixed point where $b^{ny_t}(t/t_0) = 1$ (as we did for the free energy in section B.2). Then,

$$n = -\frac{1}{y_t} \log_b(t/t_0), \tag{B.32}$$

and plugging this in:

$$G(r, t) = |t/t_0|^{2(d-y_h)/y_t} \Psi \left(\frac{r}{|t/t_0|^{-1/y_t}} \right). \tag{B.33}$$

By definition, $G(r) \propto e^{-r/\xi}$ for $r \gg \xi$, near the critical point (see argument in, for example, [12]). From the argument of the scaling function, Psi, we can identify that: $\xi \propto |t|^{-1/y_t}$.

Then,

$$\nu = 1/y_t \tag{B.34}$$

Appendix C

Basics of Information Theory

We begin with a random variable $x \in \mathcal{X}$, with the probability distribution $P(x)$. The *information* in x is:

$$I(x) = -\log_2 P(x). \tag{C.1}$$

To understand why this is more useful than $P(x)$, consider another random variable $y \in \mathcal{Y}$ with distribution $P(y)$. If the two variables are independent ($P(x,y) = P(x)P(y)$), we get that their information *adds*:

$$I(x,y) = I(x) + I(y). \tag{C.2}$$

The logarithm is a powerful tool that converts multiplication (hard) to addition (easy). Intuitively, it makes sense. An event which is 100P(x)=1, $isguaranteedtohappen, sowhenithappens, itcarriesnoinformation. Aneventwhichisinfinitelyu$

In this light, entropy is expectation value of information: where we to sample $P(x)$ an infinite number of times, what is the average information we glean from any one sample. In this limiting case of infinite samples, we perform a sum of the information of each state weighted by the probability of that state.

$$S(\mathcal{X}) = \langle I(x) \rangle = \sum_{x \in \mathcal{X}} P(x) I(x). \tag{C.3}$$

C.1 Cross-Entropy

Assume we have one set of events, $\mathcal{X}$, but two distributions $P(x)$ and $Q(x)$. We have devised an optimal encoding of the events in $\mathcal{X}$ using $Q(x)$. This means we need the least amount of bits that is possible to identify events x from $Q(x)$. If instead, the events suddenly come from $P(x)$ and $P(x) \neq Q(x)$, we will need, on average, more bits to identify x. The *average* number of bits we need is the *cross-entropy*, H, identified as the expectation over P of the information over Q:

$$H(P,Q) = \langle I_Q(x) \rangle_P = \sum_{x \in \mathcal{X}} P(x) I_Q(x) = -\sum_x P(x) \log Q(x). \tag{C.4}$$

Choosing Q so as to minimize this quantity, we derive an encoding closer to $P(x)$.

C.2 Kullback-Leibler Divergence

If we rewrite the KLD (3.13) as

$$D_{KL}(P||Q) = H(P,Q) - S(P), \tag{C.5}$$

we see it simply the cross entropy (section C.1) between P and Q minus the entropy over P. In other words this is the average number of *additional* bits we need to identify x from P when using the improper coding Q. This amount is 0 if and only if $H(P,Q) = S(P)$. From equation (C.4) and equation (C.3), we see, by the positive semi-definiteness of entropy, this immediately implies that P=Q.

Appendix D

Restricted Boltzmann Machines

D.1 Factoring of the Marginal Distribution

Our aim is to solve for $E_\theta(\boldsymbol{v})$, the energy function describing the marginalized system of visibles of an RBM, equation (3.12), in terms of the joint energy function of the full system, equation (2.10). Combining these two equations, we get the following.

$$e^{-E_\theta(\boldsymbol{v})} = \sum_{\boldsymbol{h}} e^{-E_\theta(\boldsymbol{v},\boldsymbol{h})} \tag{D.1}$$

$$\iff E_\theta(\boldsymbol{v}) = -\log\left(\sum_{\boldsymbol{h}} e^{-E_\theta(\boldsymbol{v},\boldsymbol{h})}\right), \tag{D.2}$$

where the partition functions have cancelled out. We can rewrite the right hand side:

$$\sum_{\boldsymbol{h}} e^{-E_\theta(\boldsymbol{v},\boldsymbol{h})} = \sum_{\boldsymbol{h}} e^{-\sum_i a_i v_i - \sum_j b_j h_j - \sum_{ij} v_i w_{ij} h_j} \tag{D.3}$$

$$= e^{-\sum_i a_i v_i} \sum_{\boldsymbol{h}} e^{-\sum_j \left(b_j + \sum_i v_i w_{ij}\right) h_j}. \tag{D.4}$$

$$\tag{D.5}$$

Furthermore,

$$\sum_{\boldsymbol{h}} e^{-\sum_j \left(b_j + \sum_i v_i w_{ij}\right) h_j} = \sum_{\boldsymbol{h}} \prod_j e^{-\left(b_j + \sum_i v_i w_{ij}\right) h_j} \tag{D.6}$$

$$= \prod_j \sum_{h_j \in \{0,1\}} e^{-\left(b_j + \sum_i v_i w_{ij}\right) h_j} \tag{D.7}$$

$$= \prod_j \left(1 + e^{-\left(b_j + \sum_i v_i w_{ij}\right)}\right). \tag{D.8}$$

Plugging in, we see,

$$E_\theta(\boldsymbol{v}) = -\log\left(e^{-\sum_i a_i v_i}\prod_j\left(1+e^{-\left(b_j+\sum_i v_i w_{ij}\right)}\right)\right) \tag{D.9}$$

$$= \sum_i a_i v_i - \sum_j \log\left(1+e^{-\left(b_j+\sum_i v_i w_{ij}\right)}\right). \tag{D.10}$$

D.2 Correspondence between Kadanoff's Variational RG and RBMs

When we lay a comparison with restricted Boltzmann machines, we will consider the limited case of exact transformations (i.e., $\Delta F = 0 \iff F(\boldsymbol{v}) = F_\theta(\boldsymbol{h})$). This implies that:

$$F(\boldsymbol{v}) = F_\lambda(\boldsymbol{h}) \tag{D.11}$$

$$-\ln\left(\sum_{\boldsymbol{v}} e^{-\boldsymbol{H}(\boldsymbol{v})}\right) = -\ln\left(\mathrm{Tr}_{h_j,v_i}\, e^{\boldsymbol{T}_\lambda(\boldsymbol{v},\boldsymbol{h})-\boldsymbol{H}(\boldsymbol{v})}\right) \tag{D.12}$$

$$\sum_{\boldsymbol{v}} e^{-\boldsymbol{H}(\boldsymbol{v})} = \mathrm{Tr}_{h_j,v_i}\, e^{\boldsymbol{T}_\lambda(\boldsymbol{v},\boldsymbol{h})-\boldsymbol{H}(\boldsymbol{v})} \tag{D.13}$$

$$= \sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}\in\mathcal{H}} e^{\boldsymbol{T}_\lambda(,\boldsymbol{h})} e^{-\boldsymbol{H}(\boldsymbol{v})} \tag{D.14}$$

$$= \sum_{\boldsymbol{v}} e^{-\boldsymbol{H}(\boldsymbol{v})}\sum_{\boldsymbol{h}\in\mathcal{H}} e^{\boldsymbol{T}_\lambda(\boldsymbol{v},\boldsymbol{h})}, \tag{D.15}$$

which is true if and only if: v

$$\mathrm{Tr}_{h_j}\, e^{\boldsymbol{T}_\lambda(\boldsymbol{v},\boldsymbol{h})} = 1. \tag{D.16}$$

We know that $P^{(RBM)}(\boldsymbol{h})$ is a marginalization over the energy function $E(\boldsymbol{v},\boldsymbol{h})$. Combining this with equation (D.16):

$$P(\boldsymbol{h}) = \frac{e^{-\boldsymbol{H}^{RBM}(\boldsymbol{h})}}{Z^h} = \sum_{v_i}\frac{e^{-\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h})}}{Z} \tag{D.17}$$

$$= \sum_{v_i}\frac{e^{\boldsymbol{T}(\boldsymbol{v},\boldsymbol{h})-\boldsymbol{H}(\boldsymbol{v})}}{Z} \tag{D.18}$$

$$= \frac{e^{-\boldsymbol{H}^{RG}(\boldsymbol{h})}}{Z^h}. \tag{D.19}$$

We identify:

$$\boldsymbol{H}^{RBM}(\boldsymbol{h}) = \boldsymbol{H}^{RG}(\boldsymbol{h}) \tag{D.20}$$

The Hamiltonian describing our hidden spins after block-spin renormalization also describes the configuration of hidden spins in our RBM. Furthermore, we can expand:

$$e^{\boldsymbol{T}(\boldsymbol{v},\boldsymbol{h})} = e^{-\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h})+\boldsymbol{H}(\boldsymbol{v})} \tag{D.21}$$

$$= \frac{e^{-\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h})}}{\sum_{v_i,h_j} e^{-\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h})}} \cdot \left(\sum_{v_i,h_j} e^{-\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h})} \right) \cdot \frac{P(\boldsymbol{v})}{P(\boldsymbol{v})} \cdot e^{-\boldsymbol{H}(\boldsymbol{v})} \tag{D.22}$$

$$= \frac{P(\boldsymbol{v},\boldsymbol{h})}{P(\boldsymbol{v})} \cdot \frac{\sum_{v_i,h_j} e^{-\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h})}}{\sum_{v_i} e^{-\boldsymbol{H}^{RBM}(\boldsymbol{v})}} e^{-\boldsymbol{H}^{RBM}(\boldsymbol{v})-\boldsymbol{H}(\boldsymbol{v})} \tag{D.23}$$

$$= P(\boldsymbol{h}|\boldsymbol{v}) \cdot \frac{\sum_{v_i} e^{-\boldsymbol{H}^{RBM}(\boldsymbol{v})}}{\sum_{v_i} e^{-\boldsymbol{H}^{RBM}(\boldsymbol{v})}} e^{-\boldsymbol{H}^{RBM}(\boldsymbol{v})-\boldsymbol{H}(\boldsymbol{v})} \tag{D.24}$$

$$= P(\boldsymbol{h}|\boldsymbol{v}) \cdot e^{-\boldsymbol{H}^{RBM}(\boldsymbol{v})-\boldsymbol{H}(\boldsymbol{v})}. \tag{D.25}$$

This lends Kadanoff's visible-hidden coupling $\boldsymbol{T}$ an intepretation as a kind of variational conditional probability distribution. Using the exact case, equation (D.16), we show:

$$1 = \sum_{h_j} e^{\boldsymbol{T}_\lambda(\boldsymbol{v},\boldsymbol{h})} \tag{D.26}$$

$$= \sum_{h_j} P(\boldsymbol{v}|\boldsymbol{h}) \frac{P(\boldsymbol{v})}{P_{true}(\boldsymbol{v})} \tag{D.27}$$

$$= \frac{P(\boldsymbol{v})}{P_{true}(\boldsymbol{v})}. \tag{D.28}$$

.

Therefore, $P(\boldsymbol{v}) = P_{true}(\boldsymbol{v})$. This derivation trivially goes both ways.

D.3 An Exact Correspondence between Majority-Rule RG and Convolutional RBMs

In this section, we will see that it is possible to implement a majority-rule block update with an RBM. Note that this does not concern whether or not any given RBM will actually learn this rule. For arguments considered in section 4, we need only consider the update function for a single block, $P(h_j|\boldsymbol{v}^j)$. Ignoring the index on $\boldsymbol{v}^{(j)}$, let us remind ourselves of the RBM's conditional distribution:

$$P(h_j|\boldsymbol{v}) = \frac{1}{1+e^{-h_j(\sum_i w_{ij}v_i+b_j)}.} \tag{D.29}$$

For block renormalization, we want each spin v_i to contribute equally, so we can consider the simpler $w_{ij} \to w_j$, let

$$-h_j(w_j \sum_i v_i + b_j). \tag{D.30}$$

If we have N, visible spins (i.e. $i \in \{1 \dots N\}$), then we can rewrite the sum over visibles in terms of their average $\langle v_i \rangle$ as

$$-h_j w_j \left(N\langle v_i \rangle + b_j\right). \tag{D.31}$$

If we choose $b_j = -Nw_j/2$, we get:

$$-h_j w_j N \left(\langle v_i \rangle - \frac{1}{2}\right). \tag{D.32}$$

Restricting our attention to $\langle v_i \rangle - \frac{1}{2}$, we see this is positive if and only if more than half of the spins in v_i are up, 1, and negative if and only if half the spins are down, 0. Narrowing our attention to $\langle h_j \rangle = P(h_j|\boldsymbol{v})$, the trick to recover the majority rule transformation will be to let $w_j \to -\infty$.

$$\langle h_j \rangle = \lim_{w_j \to \infty} \frac{1}{1 + e^{-w_j N\left(\langle v_i \rangle - \frac{1}{2}\right)}} \tag{D.33}$$

We distinguish three cases:

$$\langle h_j \rangle = \begin{cases} 0 & \iff \langle v_i \rangle < 1/2 \\ 0.5 & \iff \langle v_i \rangle = 1/2 \\ 1 & \iff \langle v_i \rangle > 1/2. \end{cases} \tag{D.34}$$

This is exactly the majority-rule, including even the probabilistic update rule for blocks of even numbers of spins.

Appendix E

The Real-Space Mutual Information Maximization Algorithm

E.1 A Proxy for the Mutual Information

As a first step, let us factor the joint distribution in equation (5.1) (the dependence of $\boldsymbol{h}$ on $\boldsymbol{e}$ is mediated entirely through $\boldsymbol{v}$):

$$P(\boldsymbol{h}, \boldsymbol{e}) = \sum_{\boldsymbol{v}} P(\boldsymbol{h}|\boldsymbol{v}) P(\boldsymbol{v}, \boldsymbol{e}). \tag{E.1}$$

In section 6, we saw that we can interpret an RG transformation as a conditional probability distribution $P(\boldsymbol{h}|\boldsymbol{v})$, equation (4.4) (dropping the index j and allowing multiple hidden units per block). As we discussed in section 5, we model this distribution with an RBM, with parameters Λ, such that:

$$P_\Lambda(\boldsymbol{h}, \boldsymbol{v}) = \frac{e^{-E_\Lambda(\boldsymbol{h}, \boldsymbol{v})}}{\sum_{\boldsymbol{h}', \boldsymbol{v}'} e^{-E_\Lambda(\boldsymbol{h}', \boldsymbol{v}')}}, \tag{E.2}$$

$$E_\Lambda(\boldsymbol{h}, \boldsymbol{v}) = -\left(\sum_{ij} w_{ij}^{(\Lambda)} v_i h_j + \sum_i a_i^{(\Lambda)} v_i + \sum_j b_j^{(\Lambda)} h_j \right). \tag{E.3}$$

As we saw, we can use the above to derive easy to evaluate equations for $P_\Lambda(\boldsymbol{h}|\boldsymbol{v})$, $P_\Lambda(\boldsymbol{h})$, and $P_\Lambda(\boldsymbol{v})$. In fact, we can also derive an equation for $P_\Lambda(\boldsymbol{v}|\boldsymbol{h})$. However, for RG, we only care about transformations in one direction, $\boldsymbol{v} \to \boldsymbol{h}$. Therefore, for simplicity, we can set $a_i^{(\Lambda)} = 0$ (this term factors out in $P_\Lambda(\boldsymbol{h}|\boldsymbol{v})$). We get that:

$$P_\Lambda(\boldsymbol{v}) = \sum_{\boldsymbol{h} P_\Lambda(\boldsymbol{h}, \boldsymbol{v})} = \frac{e^{-E_\Lambda(\boldsymbol{v})}}{\sum_{\boldsymbol{v}} e^{-E_\Lambda(\boldsymbol{v})}} \tag{E.4}$$

$$E_\Lambda(\boldsymbol{v}) = -\sum_j \log\left(1 + e^{-\left(b_j + \sum_i v_i w_{ij}\right)}\right), \tag{E.5}$$

and

$$P_\Lambda(\boldsymbol{h}|\boldsymbol{v}) = \frac{P_\Lambda(\boldsymbol{h},\boldsymbol{v})}{P_\Lambda(\boldsymbol{v})} = e^{-E_\Lambda(\boldsymbol{h},\boldsymbol{v})+E_\Lambda(\boldsymbol{v})}. \tag{E.6}$$

Combining with our mutual information condition, equation (5.1), we get:

$$A_\Lambda(\boldsymbol{h}:\boldsymbol{e}) = \sum_{\boldsymbol{h},\boldsymbol{v},\boldsymbol{e}} P_\Lambda(\boldsymbol{h}|\boldsymbol{v})P(\boldsymbol{v},\boldsymbol{e})\log\left(\frac{\sum_{\boldsymbol{v}} P(\boldsymbol{v},\boldsymbol{e})P_\Lambda(\boldsymbol{h}|\boldsymbol{v})}{\sum_{\boldsymbol{v}',\boldsymbol{e}} P(\boldsymbol{v}',\boldsymbol{e})P_\Lambda(\boldsymbol{h}|\boldsymbol{v}')}\right) \tag{E.7}$$

We assume that all of these distributions (not only those defined by the Λ-RBM) are of Boltzmann form. Then, we can factor our the partition functions, keeping an equation of Boltzmann form: Since these probabilities are all of Boltzmann form, we can factor out partition functions, leaving behind an equation with Boltzmann terms:

$$\frac{\sum_{\boldsymbol{v}} P(\boldsymbol{v},\boldsymbol{e})P_\Lambda(\boldsymbol{h}|\boldsymbol{v})}{\sum_{\boldsymbol{v}'} P(\boldsymbol{v}')P_\Lambda(\boldsymbol{h}|\boldsymbol{v}')} \to \frac{\sum_{\boldsymbol{v}} e^{-E(\boldsymbol{v},\boldsymbol{e})-E_\Lambda(\boldsymbol{h},\boldsymbol{v})+E_\Lambda(\boldsymbol{v})}}{\sum_{\boldsymbol{v}'} e^{-E(\boldsymbol{v}')-E_\Lambda(\boldsymbol{h},\boldsymbol{v}')+E_\Lambda(\boldsymbol{v}')}}. \tag{E.8}$$

We can reexpress the argument of the logarithm as follows:

$$\frac{\sum_{\boldsymbol{v}} e^{-E(\boldsymbol{v},\boldsymbol{e})-E_\Lambda(\boldsymbol{h},\boldsymbol{v})+E_\Lambda(\boldsymbol{v})}}{\sum_{\boldsymbol{v}'} e^{-E(\boldsymbol{v}')-E_\Lambda(\boldsymbol{h},\boldsymbol{v}')+E_\Lambda(\boldsymbol{v}')}} = \frac{\sum_{\boldsymbol{v}} e^{-E(\boldsymbol{v})-E_\Lambda(\boldsymbol{h},\boldsymbol{v})+E_\Lambda(\boldsymbol{v})}e^{-E(\boldsymbol{v},\boldsymbol{e})+E(\boldsymbol{v})}}{\sum_{\boldsymbol{v}'} e^{-E(\boldsymbol{v}')-E_\Lambda(\boldsymbol{h},\boldsymbol{v}')+E_\Lambda(\boldsymbol{v}')}}. = \frac{\sum_{\boldsymbol{v}} e^{-E_\Lambda(\boldsymbol{h},\boldsymbol{v},\boldsymbol{e})}e^{-\Delta E(\boldsymbol{v},\boldsymbol{e})}}{\sum_{\boldsymbol{v}'} e^{-E_\Lambda(\boldsymbol{h},\boldsymbol{v},\boldsymbol{e})}}, \tag{E.9}$$

where

$$E_\Lambda(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h}) = E(\boldsymbol{v},\boldsymbol{e}) + E_\Lambda(\boldsymbol{h},\boldsymbol{v}) - E_\Lambda(\boldsymbol{v}) \tag{E.10}$$

$$\Delta E(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h}) = E(\boldsymbol{v},\boldsymbol{e}) - E(\boldsymbol{v}). \tag{E.11}$$

This is the expectation of $e^{-\Delta E(\boldsymbol{v},\boldsymbol{h},\boldsymbol{e})}$ over the Boltzmann distribution with energy $E_\Lambda(\boldsymbol{h},\boldsymbol{v},\boldsymbol{e})$, where $\boldsymbol{h}$ and $\boldsymbol{e}$ are clamped. We write $\langle e^{-\Delta E(\boldsymbol{v},\boldsymbol{h},\boldsymbol{e})}\rangle_\Lambda[\boldsymbol{e},\boldsymbol{h}]$, where $[\boldsymbol{e},\boldsymbol{h}]$ denotes the dependence of this expectation value on the clamped values. Although ΔE has no dependence on Λ, its expectation value gains a dependence through $P_\Lambda(\boldsymbol{h},\boldsymbol{v},\boldsymbol{e})$. We get the following expression for the mutual information proxy:

$$A_\Lambda(\boldsymbol{h}:\boldsymbol{e}) = \sum_{\boldsymbol{h},\boldsymbol{v},\boldsymbol{e}} P_\Lambda(\boldsymbol{h}|\boldsymbol{v})P(\boldsymbol{v},\boldsymbol{e})\log\left(\langle e^{-\Delta E(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h})}\rangle_\Lambda[\boldsymbol{e},\boldsymbol{h}]\right), \tag{E.12}$$

Here we see that the clamped values of $\boldsymbol{e}$ and $\boldsymbol{h}$ are given by the outside sum, so we write $[\boldsymbol{e},\boldsymbol{h}]$ after the expectation value to denote its dependence on these variables

To further simplify this expression, we use a cumulant expansion:

$$\langle e^{K(\boldsymbol{x})}\rangle = e^{\sum_{\kappa=0}^{\infty}\frac{1}{\kappa!}C_\kappa}, \tag{E.13}$$

with the cumulants expressed in terms of the moments. The first three terms are:

$$C_1 = \langle K\rangle \tag{E.14}$$

$$C_2 = \langle K^2\rangle - \langle K\rangle^2 \tag{E.15}$$

$$C_3 = \langle K^3\rangle - 3\langle K^2\rangle\langle K\rangle + 2\langle K\rangle^3. \tag{E.16}$$

This extends to distributions that include a dependence on other variables (i.e., $\boldsymbol{e}$, $\boldsymbol{h}$). Then, we can approximate:

$$A_\Lambda(\boldsymbol{h}:\boldsymbol{e}) \approx \sum_{\boldsymbol{h},\boldsymbol{v},\boldsymbol{e}} P_\Lambda(\boldsymbol{h}|\boldsymbol{v})P(\boldsymbol{v},\boldsymbol{e})\langle -\Delta E(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h})\rangle[\boldsymbol{e},\boldsymbol{h}], \tag{E.17}$$

where now

$$\langle \Delta E_\Lambda(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h})\rangle[\boldsymbol{e},\boldsymbol{h}] \equiv \frac{\sum_{\boldsymbol{v}} \left(\Delta E_\Lambda(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h})\right) e^{-E_\Lambda(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v}'} e^{-E_\Lambda(\boldsymbol{v}',\boldsymbol{h})}}. \tag{E.18}$$

In general, we will not have access to the energy functions $E(\boldsymbol{v})$ and $E(\boldsymbol{v},\boldsymbol{e})$. Even if we were to have access to $P(\boldsymbol{x}=\{\boldsymbol{v},\boldsymbol{b},\boldsymbol{e},\boldsymbol{o}\})$, the necessary marginalizations ($P(\boldsymbol{v},\boldsymbol{e})=\sum_{\boldsymbol{b},\boldsymbol{o}} P(\boldsymbol{x})$ and $P(\boldsymbol{v})=\sum_{\boldsymbol{b},\boldsymbol{e},\boldsymbol{o}} P(\boldsymbol{x})$) are not necessarily tractable calculations. Therefore, we approximate $E(\boldsymbol{v},\boldsymbol{e})$ and $E(\boldsymbol{v})$ with two other RBMs with parameters Θ and Ψ, respectively:

$$E(\boldsymbol{v},\boldsymbol{e}) \approx E_\Theta(\boldsymbol{v},\boldsymbol{e}) \tag{E.19}$$

$$E(\boldsymbol{v}) \approx E_\Psi(\boldsymbol{v}). \tag{E.20}$$

Note that $(\boldsymbol{v},\boldsymbol{e})$ and $(\boldsymbol{v})$ are the inputs, the visible layers of these two RBMs. We introduce a second, hidden layer, which we marginalize over to get the above quantities.

Plugging in our learned distributions, we get:

$$A_{\Lambda,\Theta,\Psi}(\boldsymbol{h}:\boldsymbol{e}) \approx \sum_{\boldsymbol{h},\boldsymbol{v},\boldsymbol{e}} P_\Lambda(\boldsymbol{h}|\boldsymbol{v})P(\boldsymbol{v},\boldsymbol{e})\langle -\Delta E_{\Theta,\Psi}(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h})\rangle_{\Lambda,\Theta,\Psi}[\boldsymbol{e},\boldsymbol{h}], \tag{E.21}$$

We have derived an expression which we evaluate using Monte Carlo averages.

$$A_{\Lambda,\Theta,\Psi}(\boldsymbol{h}:\boldsymbol{e}) \approx \frac{1}{N^{(\boldsymbol{v},\boldsymbol{e})}} \sum_{\boldsymbol{h},\boldsymbol{v},\boldsymbol{e}} P_\Lambda(\boldsymbol{h}|\boldsymbol{v})\langle -\Delta E_{\Theta,\Psi}(\boldsymbol{v},\boldsymbol{e},\boldsymbol{h})\rangle_{\Lambda,\Theta,\Psi}[\boldsymbol{e},\boldsymbol{h}], \tag{E.22}$$

First, we generate samples of $\boldsymbol{e}$ and $\boldsymbol{v}$ simply from partitions of our dataset $\boldsymbol{x}$[1]. Then, we use our Λ-RBM to translate samples of $\boldsymbol{v}$ into samples of $\boldsymbol{h}$. For each combination of $\boldsymbol{e}$ and $\boldsymbol{h}$, we generate wholly new samples, $\boldsymbol{v}'$, with the energy function $E_{\Lambda,\Psi}(\boldsymbol{v},\boldsymbol{e})$, over which we perform the internal MC-average.

In fact, we are not interested in the quantity, $A_{\Lambda,\Theta,\Psi}(\boldsymbol{h}:\boldsymbol{e})$ as much as we are in its derivative. With some simple (though exceedingly tedious)[2] algebra, we can evaluate an expression for the derivative with respect to Λ of our mutual information proxy. It is crucial we calculate this explicitly, because through the stochastic nature of MC-sampling, our proxy A_Λ will often have a zero-gradient. We would not be able to use stochastic gradient descent.

Comment on the Cumulant Expansion. It is not entirely clear why the authors felt this expansion was necessary. The expectations are ultimately calculated over Monte Carlo samples, and $\langle K(\boldsymbol{x})\rangle$ is not much of an improvement over $\langle e^{K(\boldsymbol{x})}\rangle$ in computational complexity. Furthermore, it

[1] We could have generated these samples de novo using the Θ-RBM, but that would introduce needless time-complexity. Instead, we use that data we already have access to

[2] I mean whiteboards and whiteboards of it.

throws away information about higher order terms. It may be that this step manages to suppress irrelevant fluctuations, but a more rigorous understanding is needed. In the future, we may implement the algorithm, including additional terms in this expansion, comparing the ultimate performance against an un-expanded baseline. Then, we can more rigorously justify or critique this assumption.

E.2 Intrinsic Thermometer

In order to measure critical exponents, we need a means of measuring the values of macrovalues through our iterations. If our aim is to measure temperature, we may, however, not have access to an explicit thermometer. To further complicate matters, as we saw previously, RG transformations, generally, introduce higher order correlations. From data alone, it is not necessarily possible to identify the contributions from different terms in the Hamiltonian. For our approach to be general, then, we need a means of implicit macroparameter calculations. With regard to temperature, we have several options.

Koch-Janusz and Ringel considered three proxies to the temperature:

1. First, they looked at using the MC configurations at given iterations. We can compute expectations of functions like the nearest-neighbor and next-nearest-neighbor correlation. Since these depend monotonically on the temperature near the critical point, we can use these to recreate an effective temperature at successive length scales.

2. Next, they used the mutual information as a proxy to the temperatue. This too depends monotonically on the temperature near the critical point.

3. Finally, they mentioned the possibility of using the Λ- and Ψ-RBMs. These we can also use to measure expectations of correlations, and we can even intrinsically evaluate effective temperatures.

There are yet other options, not considered by Koch-Janusz and Ringel. We can formulate the task of measuring the effective temperature of a set of samples as a supervised learning problem. Then, we can further leverage the power of neural networks to act as our thermometers[3]. So far, we have only considered neural networks which take a fixed number of inputs. Our aim for a NN thermometer would be that it could accurately measure the temperature of systems with different numbers of spins (at different RG steps). Two immediate solutions come to mind. First, we could train the networks on subsets of the $\boldsymbol{x}$ samples. Then, we would train a separate network for each successive step. Second, we could make use of recursive neural networks[4]. These recursive neural networks employ, as their name suggests, recursive weight-sharing, which explicitly allows for variable-sized inputs.

However, this would reduce the extent to which the RSMI algorithm is truly unsupervised. In practice, this need not be an issue: we assume that we have access to some data, and more often

[3] Accomplished for example by Iso et al. [6]

[4] Not to be confused with recurrent neural networks, a particular kind of recursive neural network used for processing 1-dimensional (typically temporal) data [].

than not, this will be given to us by MCMC techniques. Then, we, necessarily, have a means of measuring parameters like these, at least, in the non-coarse-grained systems.

Though we did not have the time to implement these ideas, we will continue developing *rgpy*, and we intend to introduce these features in later releases.

For our implementation, we used the next-nearest correlation function. For each value of this function we derive in successive iterations, we assign the temperature that it is closest too in our sample set. In the future, we will use more complicated functions: this "nearest-neighbors" approach introduces significant error margins.

E.3 Experimental Realization

Ultimately, our calculation of the correlation length critical exponent, ν, proceeded as follows. We generated samples (1000 per temperature) of Ising configurations of various lattice widths (8, 16, 32, 64). From values for the next-nearest neighbor correlation function, we devised a "thermometer", see preceding section, for each length-scale. We trained the RSMI algorithm on samples of 64-by-64 spins for a total of 3 RG iterations. Each RBM we trained for 30 epochs. For the rest, our experimental set-up mirrored that of Koch-Janusz and Ringel [5]. Having generated results, and measurements of the magnetic susceptibility, χ, for each step in each temperature sequence. From the peaks in susceptibility, we identified the effective critical temperature. Plotting this on one curve against $1/L$, using arguments from finite-size scaling (see section B.1), we collapsed the data onto a single curve whose slope equals ν.

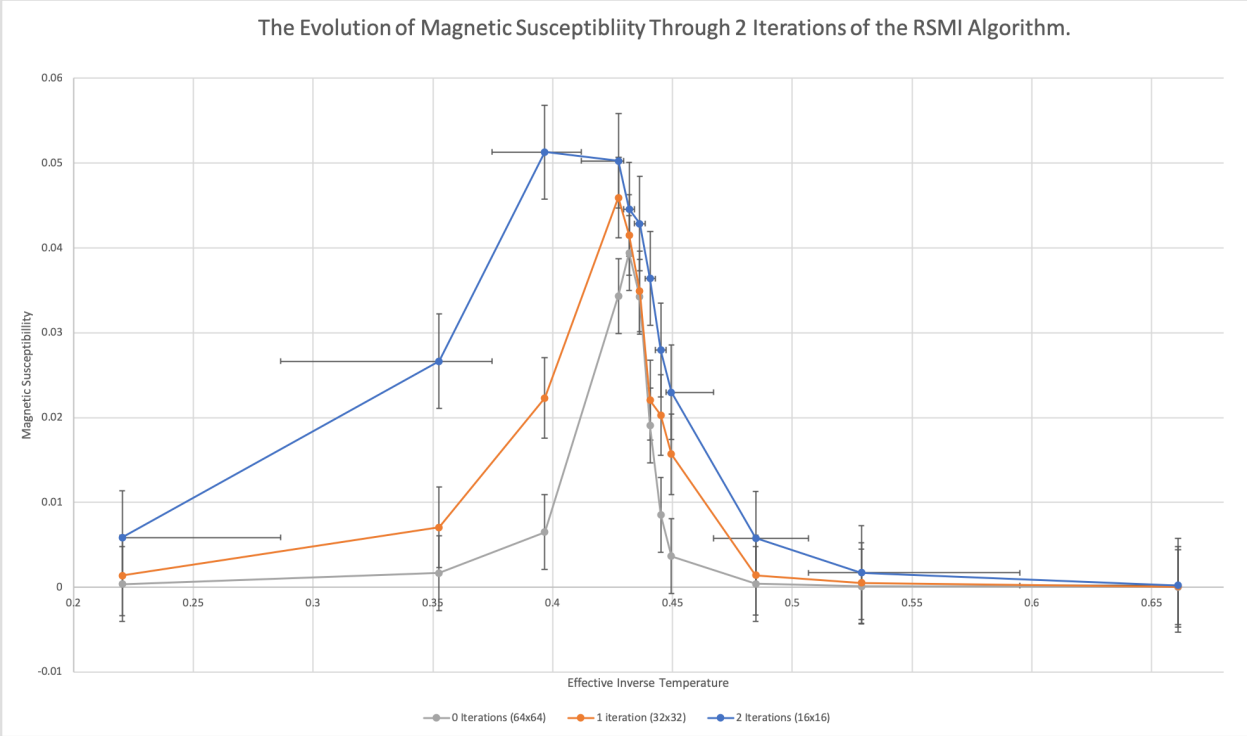


Figure E.1: The Magnetic Susceptiblity, χ, at different iterations of the RSMI algorithm.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[2] Henry W. Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168:1223–1247, Sep 2017.

[3] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[4] Pankaj Mehta and David J. Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv e-prints*, page arXiv:1410.3831, Oct 2014.

[5] Maciej Koch-Janusz and Zohar Ringel. Mutual information, neural networks and the renormalization group. *Nature Physics*, 14:578–582, Jun 2018.

[6] Satoshi Iso, Shotaro Shiba, and Sumito Yokoo. Scale-invariant feature extraction of neural network and renormalization group flow. *arXiv e-prints*, 97:053304, May 2018.

[7] David J. Schwab and Pankaj Mehta. Comment on "why does deep and cheap learning work so well?" [arxiv:1608.08225]. *arXiv e-prints*, page arXiv:1609.03541, Sep 2016.

[8] Jaco ter Hoeve. Renormalization group connected to neural networks, 2018.

[9] Patrick M. Lenggenhager, Zohar Ringel, Sebastian D. Huber, and Maciej Koch-Janusz. Optimal renormalization group transformation from information theory. *arXiv e-prints*, page arXiv:1809.09632, Sep 2018.

[10] Jesse Hoogland. rgpy. `https://github.com/jqhoogland/rgpy`, 2019.

[11] Cyril Domb. *The critical point: a historical introduction to the modern theory of critical phenomena*. CRC Press, 1996.

[12] John Cardy. *Scaling and Renormaliztion in Statistical Physics*. Cambridge University Press, Cambridge, United Kingdom, 1996.

[13] JGTechSol. Optical computing, Feb 2017.

[14] Alan H Guth. Time since the beginning. *arXiv preprint astro-ph/0301199*, 2003.

[15] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[16] Jian-Sheng Wang and Robert H Swendsen. Cluster monte carlo algorithms. *Physica A: Statistical Mechanics and its Applications*, 167(3):565–579, 1990.

[17] Kenneth G Wilson. Problems in physics with many scales of length. *Scientific American*, 241(2):158–179, 1979.

[18] Michael E Peskin. *An introduction to quantum field theory*. CRC Press, 2018.

[19] Leo P Kadanoff, Anthony Houghton, and Mehmet C Yalabik. Variational approximations for renormalization group transformations. *Journal of Statistical Physics*, 14(2):171–203, 1976.

[20] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[21] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *arXiv e-prints*, page arXiv:1803.08823, Mar 2018.

[22] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[23] Mukul Rathi. Learning through gradient descent, Aug 2018.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] Max Welling, Michal Rosen-Zvi, and Geoffrey E Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in neural information processing systems*, pages 1481–1488, 2005.

[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[27] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[28] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[29] pyfissa.