



UNIVERSIDAD NACIONAL DE COLOMBIA

Exploring in and out-of-equilibrium learning regimes of restricted Boltzmann machines

Alfonso de Jesús Navas Gómez

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Física
Bogotá, Colombia
2022

Trabajo final de maestría

Exploring in and out-of-equilibrium learning regimes of Restricted Boltzmann Machines

Alfonso de Jesús Navas Gómez

Director:

Ph.D. José Jairo Giraldo Gallo

Codirectora:

Ph.D. Beatriz Seoane Bartolomé

Línea de Investigación:

Física estadística de sistemas desordenados, machine learning

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Física
Bogotá, Colombia
2022

Abstract

Exploring in and out-of-equilibrium learning regimes of Restricted Boltzmann Machines

Although machine learning based artificial intelligence is considered as one of the most disruptive technologies of our age, the understanding of many of these methods lies behind their practical success. Statistical physics of disordered systems has a long history studying inference problems and learning processes with its own tools, shedding light on the underlying mechanisms of many machine learning models. Following this tradition, in this master's thesis we studied how the training protocol affects the model and the features extracted by an unsupervised machine learning method called Restricted Boltzmann Machine. In particular, we trained machines in and out-of-equilibrium learning regimes with Ising Model samples and then, using a novel pattern extraction protocol developed in this work, we inferred the coupling matrix of the effective Ising model learned in each case. Such experiment allowed us to elucidate some consequences of equilibrium and non-equilibrium training regimes. Additionally, we explored the potential use of restricted Boltzmann machine as an inference tool for Ising model-like sample data, being the first step towards to tackle more complex problems.

Keywords: Artificial intelligence, Machine Learning, Statistical physics of disordered systems, Restricted Boltzmann machines, Monte-Carlo methods.

Resumen

Explorando los regímenes de aprendizaje dentro y fuera del equilibrio de las máquinas de Boltzmann restringidas

Aunque los métodos de inteligencia artificial basados en aprendizaje automatizado son considerados como una de las tecnologías disruptivas de nuestros tiempos, el entendimiento de estas herramientas yace muy por detrás de su éxito práctico. La física estadística de sistemas desordenados goza de una larga historia estudiando problemas de inferencia y aprendizaje usando sus propias herramientas. Siguiendo con esta tradición, en este trabajo final de maestría se estudió cómo el protocolo de aprendizaje afecta a los patrones extraídos por una Máquina Restringida de Boltzmann. En particular, se entrenaron máquinas dentro y fuera del equilibrio con muestras del modelo de Ising en 1 y 2 dimensiones para luego, usando un nuevo método de inferencia, extraer la matriz de acoplamientos del modelo efectivo aprendido en cada caso. Este experimento permitió dilucidar algunas consecuencias de los regímenes de entrenamiento dentro y fuera de equilibrio. Adicionalmente, se exploró el potencial del uso de las Máquinas Restringidas de Boltzmann para la extracción automática de patrones para muestras similares a las del modelo de Ising, siendo este el primer paso para abordar problemas más complejos.

Palabras clave: Inteligencia artificial, Aprendizaje automatizado, Física estadística de sistemas desordenados, Máquinas restringidas de Boltzmann, Métodos de Monte-Carlo.

Contents

1	Introduction	2
2	Theoretical framework	4
2.1	Overview of machine learning and RBMs	4
2.2	The RBM model	5
2.2.1	General definition	6
2.2.2	RBM Training and Gibbs sampling	7
2.3	Learning the Ising Model with RBMs	10
2.3.1	Reproducing the thermodynamic observables of the Ising model . . .	10
2.3.2	Extracting couplings from the RBM	11
3	Materials and methods	15
3.1	Ising model simulations and dataset generation	15
3.2	Simulation data analysis	16
3.3	RBM training	20
4	Results	22
4.1	Extracting the effective couplings from the RBM	22
4.2	Learning the 1D Ising model	24
4.2.1	Evolution of the observables of generated samples with the sampling time	25
4.2.2	Inferring the effective model of in and out-of-equilibrium trained ma- chines	29
4.3	Learning the 2D Ising model	29
4.3.1	Observable predictions at all temperatures	30
4.3.2	Inferring the effective model at all temperatures	32
5	Conclusions and perspectives	35
6	Acknowledgements	36

1 Introduction

Restricted Boltzmann Machines (RBMs) are stochastic neural networks [14], known for being able to learn a latent representation of the data and generate statistically similar new data, but different from those of the training set [12]. From the statistical physicist's point of view, an RBM is an extremely familiar object: a disordered Ising spin Hamiltonian, in which the spins are distributed on a bipartite lattice [12]. The training process of RBMs consists in finding a set of good parameters for the model (i.e., the couplings between the spins), so that the Gibbs-Boltzmann distribution associated with the RBM resembles as much as possible to the original data distribution. In other words, the training process is, essentially, an inverse Ising problem in Statistical Physics, i.e., instead of searching the most probable configurations of a model at the equilibrium, we look for the model that has the dataset samples as typical configurations. Additionally, the effective Ising spin Hamiltonian of the trained machine is accessible for further investigation and examination. Given such interpretability potential, which is rare in Machine Learning, RBMs are beginning to be used in natural sciences beyond purely classification and data generation tasks [8, 15, 25, 34, 33, 37].

In the standard set-up, RBMs are trained via log-likelihood maximization using stochastic gradient ascent dynamics, which requires costly Monte Carlo (MC) simulations for the estimation of the gradient of the log-likelihood. In practice, working with such systems with little experience could be tricky and unpredictable. One of the major issues with RBMs is that it is hard to evaluate whether the learning is progressing or not [13]. In order to overcome these difficulties, for decades, practitioners have been proposing more or less successful training recipes [10, 14, 19], approximating equilibrium distributions with just some few and fixed, typically 1-10, MC steps for computational reasons, but without studying the crucial quantity of the problem: the mixing time [13]. Only until very recently [13], there were no studies estimating how long these MC chains should be in order to provide reliable estimations of the log-likelihood gradient. Decelle *et al.* [13] found that the origin of most of the difficulties of the training process of RBMs are because the number of MC steps used is smaller than the mixing time of the algorithm in the model. It was also observed that machines trained in such non-equilibrium regime learn to reproduce the dynamical processes used for the training, and not the distribution of the data. Moreover, the most unpleasant finding of this work was noticing that most of the studies discussed in the literature using RBMs have been analyzing out-of-equilibrium machines, which raises reasonable concerns about the reliability of the features extracted.

Given the increasing attention that RBMs are receiving lately, it is more important than ever to establish reproducible protocols and evaluation tools to guarantee the reliability of the features extracted by the machine. Therefore, the objective of this master's thesis is understanding how the training protocol affects the effective model and the features extracted by the machine. With this goal in mind, we will start with a very controlled set-up, where the true model generating the samples is known: the Ising Model. We will use RBM in and out-of-equilibrium training protocols to learn Ising model samples. Once we have trained the RBMs, it is possible to infer from them the coupling matrix of the effective model learned in each training [10]. Then, by comparing the extracted features to the original ones, it was possible to assess the differences between the features extracted by an RBM trained in and out-of-equilibrium learning regimes. Since the procedure used to infer the couplings from the trained RBMs was developed in this thesis, we also explore and discuss how it can be used as an inference tool for more complex structured data.

2 Theoretical framework

In the following sections, some generalities and concepts from Machine Learning and RBMs are introduced. Then, a detailed description of the model and its training algorithm is presented. Finally, we will review how the Ising model has been used by physicists to study the generative power and the inferential potential of RBMs.

2.1 Overview of machine learning and RBMs

Machine Learning (ML) is a subfield of Artificial Intelligence with the goal of developing computer algorithms capable of recognizing patterns in data and to perform tasks such as automatically classifying the data into different categories [7, 24]. In an ML set-up, a large dataset is used to tune the parameters of an adaptive model. This process is called training or learning [7]. Learning paradigms can be divided into three broad categories: *supervised*, *unsupervised*, and *reinforcement learning* [12, 24]. Supervised learning deals with learning a specific task, usually classification or regression, by giving the machine a large set of labeled data and fitting the prediction of the model by minimizing a conveniently chosen loss function [7, 13, 24]. In unsupervised learning, no label is assigned to the data and the result depends only on the structure of the assumed model and that of the dataset. Such a protocol consists in finding a representation of the data given an explicit or implicit probability distribution by fitting a likelihood function on the data [12]. Common unsupervised learning tasks include *clustering*, *dimensionality reduction*, and *generative modeling* [24]. Finally, reinforcement learning neither requires labeled data. In this case, an agent learns by interacting with its environment and changing its behavior to maximize a reward [24]. Tasks performed within this paradigm can be as sophisticated as learning how to play strategy games, such as chess or Go, well enough to outperform human champions in these activities [30].

An example of an unsupervised ML model are Boltzmann Machines (BMs), which are bidirectionally fully connected networks of stochastic processing elements, called units, nodes or variables [1]. From a physicist's point of view, such models are Ising models in which spins (i.e., units) are coupled by infinite-ranged and stochastic interactions, i.e., a Sherrington-Kirkpatrick model [29] (see Fig. 2-1(a)). A BM is capable of performing unsupervised learning tasks, i.e., capturing the underlying structure of the probability distribution of the data used during its training [1], but implementing such training protocol is slow and computationally demanding. However, the training procedure can be simplified by constraining the

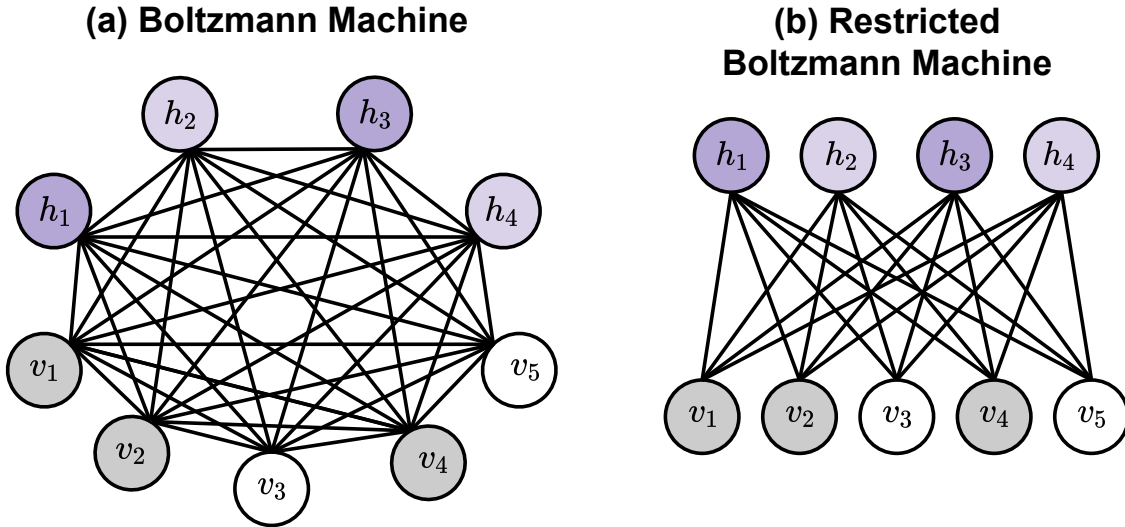


Figure 2-1: Diagram of the lattice structure of a Boltzmann Machine (a) and a Restricted Boltzmann Machine (b).

BM network topology to a bipartite structure, which leads us to RBMs [14] (see Fig. 2-1(b)).

Originally introduced under the name of Harmonium by Smolensky in 1986 [31], RBMs started to receive attention by the ML community in the mid-2000, after Hinton *et al.* created a fast learning algorithm for them, the *contrastive divergence* (CD) [18]. Using this algorithm, RBMs are able to learn a non-trivial data set such as images of handwritten digits (MNIST) [12, 23]. Since then, RBMs have found many applications performing unsupervised tasks, for instance, as deep auto-encoders to pre-train deep neural networks [20]. Later, it was showed that RBMs are indeed universal approximators of discrete distributions [22], i.e., an RBM with sufficiently large number of hidden nodes can approximate arbitrarily well any discrete distribution, which led to many rigorous results about the power representation of these models [27]. Such representational power, combined with its potential for interpretability and its ability to manage learning from small data sets, has drawn attention to RBMs for use in a scientific context [8, 15, 21, 25, 34, 33, 37].

2.2 The RBM model

In this part, we will discuss the main features of the RBM model and its training procedure. A full detailed description of an RBM is presented in Ref. [14]. For a presentation of a generalized model and a revision of recent advances from Statistical Physics, we refer the

reader to Ref. [12].

2.2.1 General definition

As it was stated before, an RBM is a disordered Ising model defined on a bipartite and undirected lattice. In such architecture, every visible node v_j , $j \in \{1, \dots, m\}$ is connected to all hidden nodes h_i , $i \in \{1, \dots, n\}$ and no connection between any pair of visible or hidden nodes occurs. The visible and the hidden nodes will be denoted collectively as \mathbf{v} and \mathbf{h} , respectively. That said, it is important to remark that a major difference between the traditional definition of the RBM and the Ising model commonly studied in Statistical Physics is that, in the former, both visible and hidden nodes are binary variables defined over $\{0, 1\}$, i.e., bits, instead of $\{-1, 1\}$, which is the sample space of spins variables in the latter. However, this does not constitute a fundamental distinction, since one representation can always be mapped to the other. The weight of the interactions between the nodes in the visible and hidden layers is determined by the coupling matrix \mathbf{W} , whose elements are denoted by W_{ij} . Additionally, each variable can have a local magnetic field or bias, in ML in the jargon. We will use b_j and c_i to denote the biases of visible and hidden nodes, respectively. Similarly, we will collectively call the biases as \mathbf{b} and \mathbf{c} . Thus, we can introduce the Hamiltonian, or energy function, of the joint state $\{\mathbf{v}, \mathbf{h}\}$ of the machine:

$$\mathcal{H}(\mathbf{v}, \mathbf{h}) = - \sum_{i=1}^n \sum_{j=1}^m h_i W_{ij} v_j - \sum_{j=1}^m b_j v_j - \sum_{i=1}^n c_i h_i. \quad (2-1)$$

The probability of such given state of the RBM follows the Gibbs-Boltzmann distribution:

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})}}{\mathcal{Z}}, \quad (2-2)$$

where the partition function \mathcal{Z} is defined as

$$\mathcal{Z} = \sum_{\mathbf{v}, \mathbf{h}} e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})}, \quad (2-3)$$

Where the sum $\sum_{\mathbf{v}, \mathbf{h}}$ runs over all the possible configurations of the model. From the above expression, it is immediate that the Gibbs-Boltzmann distribution defined in (2-2) is correctly normalized. In addition, we introduce the average over the RBM measure denoted by

$$\langle f(\mathbf{v}, \mathbf{h}) \rangle_{\mathcal{H}} = \sum_{\mathbf{v}, \mathbf{h}} f(\mathbf{v}, \mathbf{h}) p(\mathbf{v}, \mathbf{h}). \quad (2-4)$$

The advantage of having a bipartite structure is that the hidden variables are statistically independent given the state of the visible layer, and conversely. In other words,

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^n p(h_i|\mathbf{v}) \quad \text{and} \quad p(\mathbf{v}|\mathbf{h}) = \prod_{j=1}^m p(v_j|\mathbf{h}). \quad (2-5)$$

Such property is crucial since it will be used in the learning and sample generating procedures of the RBM to parallelize the computations. In order to formally derive the property (2-5), let us calculate the probability mass function of the visible layer by marginalizing over the hidden variables h_i ,

$$\begin{aligned}
p(\mathbf{v}) &= \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{h}} e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})} \\
&= \frac{1}{\mathcal{Z}} \sum_{h_1} \sum_{h_2} \cdots \sum_{h_n} \exp \left(\sum_{i=1}^n \sum_{j=1}^m h_i W_{ij} v_j + \sum_{j=1}^m b_j v_j + \sum_{i=1}^n c_i h_i \right) \\
&= \frac{1}{\mathcal{Z}} \exp \left(\sum_{j=1}^m b_j v_j \right) \prod_{i=1}^n \sum_{h_i} \exp \left(h_i \left\{ c_i + \sum_{j=1}^m W_{ij} v_j \right\} \right) \\
&= \frac{1}{\mathcal{Z}} \prod_{j=1}^m \exp(b_j v_j) \prod_{i=1}^n \left\{ 1 + \exp \left(c_i + \sum_{j=1}^m W_{ij} v_j \right) \right\}. \tag{2-6}
\end{aligned}$$

Using the Bayes' theorem and the above result, we can compute the conditional probability $p(\mathbf{h}|\mathbf{v})$:

$$\begin{aligned}
p(\mathbf{h}|\mathbf{v}) &= \frac{p(\mathbf{v}, \mathbf{h})}{p(\mathbf{v})} = \frac{e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})}}{\sum_{\mathbf{h}} e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})}} \\
&= \prod_{i=1}^n \frac{\exp \left(h_i \left\{ c_i + \sum_{j=1}^m W_{ij} v_j \right\} \right)}{1 + \exp \left(c_i + \sum_{j=1}^m W_{ij} v_j \right)}. \tag{2-7}
\end{aligned}$$

Following an analogous procedure, we can obtain

$$p(\mathbf{v}|\mathbf{h}) = \prod_{j=1}^m \frac{\exp \left(v_j \left\{ b_j + \sum_{i=1}^n h_i W_{ij} \right\} \right)}{1 + \exp \left(b_j + \sum_{i=1}^n h_i W_{ij} \right)}. \tag{2-8}$$

From the results in equations (2-7) and (2-8), it is clear that the property (2-5) holds. Also, we can readily compute the probability of a single node being one:

$$p(h_i = 1|\mathbf{v}) = \sigma \left(\sum_{j=1}^m W_{ij} v_j + c_i \right) \quad \text{and} \quad p(v_i = 1|\mathbf{v}) = \sigma \left(\sum_{i=1}^n h_i W_{ij} + b_j \right), \tag{2-9}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function.

2.2.2 RBM Training and Gibbs sampling

The training process of RBM consists in tuning the weight matrix \mathbf{w} and the biases \mathbf{b} and \mathbf{c} , so that the Gibbs-Boltzmann distribution (2-2) resembles as much as possible to the empirical data distribution. The classical procedure to do so is by maximizing the likelihood

function of the dataset, which is achieved by implementing the gradient ascent of the log-likelihood (LL) function.

Considering a set of data points $\mathcal{D} = \{v_j^{(d)}\}$, where $d \in \{1, \dots, M\}$ is the index of each data point, the LL function over \mathcal{D} is given by

$$\mathcal{L} = \frac{1}{M} \sum_{d=1}^M \log p(\mathbf{v}^{(d)}) = \frac{1}{M} \sum_{d=1}^M \log \sum_{\mathbf{h}} e^{-\mathcal{H}(\mathbf{v}^{(d)}, \mathbf{h})} - \log \mathcal{Z}. \quad (2-10)$$

Since we need to compute the gradient to maximize the LL, let us begin by deriving such function with respect to the elements of the weight matrix W_{ij} :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W_{ij}} &= \frac{1}{M} \sum_{d=1}^M \sum_{\mathbf{h}} h_i v_j^{(d)} p(\mathbf{h}|\mathbf{v}^{(d)}) - \sum_{\mathbf{v}, \mathbf{h}} h_i v_j p(\mathbf{v}, \mathbf{h}) \\ &= \frac{1}{M} \sum_{d=1}^M \sum_{\mathbf{h}} h_i v_j^{(d)} p(\mathbf{h}|\mathbf{v}^{(d)}) - \langle h_i v_j \rangle_{\mathcal{H}}. \end{aligned} \quad (2-11)$$

Introducing the following definition:

$$\langle f(\mathbf{v}, \mathbf{h}) \rangle_{\mathcal{D}} = \frac{1}{M} \sum_{d=1}^M \sum_{\mathbf{h}} f(\mathbf{v}^{(d)}, \mathbf{h}) p(\mathbf{h}|\mathbf{v}^{(d)}), \quad (2-12)$$

for the average over the dataset, and replacing it in (2-11), we obtain

$$\frac{\partial \mathcal{L}}{\partial W_{ij}} = \langle h_i v_j \rangle_{\mathcal{D}} - \langle h_i v_j \rangle_{\mathcal{H}}. \quad (2-13)$$

Similarly, for the biases we can obtain

$$\frac{\partial \mathcal{L}}{\partial b_j} = \langle v_j \rangle_{\mathcal{D}} - \langle v_j \rangle_{\mathcal{H}}, \quad (2-14)$$

$$\frac{\partial \mathcal{L}}{\partial c_i} = \langle h_i \rangle_{\mathcal{D}} - \langle h_i \rangle_{\mathcal{H}}. \quad (2-15)$$

On the simplest implementation of learning, once these derivatives are computed, the parameters of the model can be updated according to:

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \gamma \left. \frac{\partial \mathcal{L}}{\partial W_{ij}} \right|_{\phi^{(t)}}, \quad (2-16)$$

$$b_j^{(t+1)} = b_j^{(t)} + \gamma \left. \frac{\partial \mathcal{L}}{\partial b_j} \right|_{\phi^{(t)}}, \quad (2-17)$$

$$c_i^{(t+1)} = c_i^{(t)} + \gamma \left. \frac{\partial \mathcal{L}}{\partial c_i} \right|_{\phi^{(t)}}, \quad (2-18)$$

where we denoted all the parameters of the model at t step as $\phi^{(t)} = \{\mathbf{W}^{(t)}, \mathbf{b}^{(t)}, \mathbf{c}^{(t)}\}$. Additionally, it was introduced the parameter γ , which is called the *learning rate* and regulates the pace at which the parameters are updated in the steepest ascent direction.

The hardest part of implementing the above procedure resides in the calculation of the gradient of the LL function. However, let us note that the positive part of the gradient is easily computable using the results given above:

$$\langle h_i v_j \rangle_{\mathcal{D}} = \frac{1}{M} \sum_{d=1}^M \sum_{h_i} h_i v_j^{(d)} p(h_i | \mathbf{v}^{(d)}) = \frac{1}{M} \sum_{d=1}^M v_j^{(d)} p(h_i = 1 | \mathbf{v}^{(d)}) \quad (2-19)$$

$$\langle v_j \rangle_{\mathcal{D}} = \frac{1}{M} \sum_{d=1}^M v_j^{(d)} \quad (2-20)$$

$$\langle h_i \rangle_{\mathcal{D}} = \frac{1}{M} \sum_{d=1}^M \sum_{h_i} h_i p(h_i | \mathbf{v}^{(d)}) = \frac{1}{M} \sum_{d=1}^M p(h_i = 1 | \mathbf{v}^{(d)}), \quad (2-21)$$

where $p(h_i = 1 | \mathbf{v}^{(d)})$ is given in (2-9). Thus, the difficulty lies in the computation of the negative part of the gradient, which is not exactly solvable and depends on the model at each stage. However, it can be estimated from equilibrium samples of the RBM. Such samples are generated via a Markov chain Monte-Carlo (MCMC) method, often referred to as *Gibbs sampling*. The Gibbs sampling consists in implementing the following Markov chain:

$$\begin{aligned} \mathbf{v}^{(0)} &\rightarrow \mathbf{h}^{(0)} \sim p(\mathbf{h} | \mathbf{v}^{(0)}), \\ \mathbf{h}^{(0)} &\rightarrow \mathbf{v}^{(1)} \sim p(\mathbf{v} | \mathbf{h}^{(0)}), \\ \mathbf{v}^{(1)} &\rightarrow \mathbf{h}^{(1)} \sim p(\mathbf{h} | \mathbf{v}^{(1)}), \\ &\vdots \\ \mathbf{v}^{(k)} &\rightarrow \mathbf{h}^{(k)} \sim p(\mathbf{h} | \mathbf{v}^{(k)}). \end{aligned}$$

This is possible because of the statistical independence of the conditional distributions, which is indeed a consequence of the bipartite structure of the machine. To sample the equilibrium configurations of the RBM is crucial to obtain effective models whose equilibrium distribution fits the empirical distribution of the data [13]. Although equilibrium can be reached, no matter the initial condition, for a sufficiently large k , implementing long MCMC is computationally expensive. Therefore, part of the success of RBM training resides in finding a clever starting point for the MC chains in order to reduce the number of k steps needed to reach the equilibrium. In this regard, the *contrastive divergence* (CD) method [18] was introduced as an efficient approximated scheme, and has been used extensively since its introduction by Hinton in the middle '00s. Such method uses a stochastic gradient ascent approach i.e., instead of updating the parameters by calculating the gradient using the entire training dataset at each iterative step, the training data is divided into small subsets called

minibatches, over which the gradient ascent is implemented sequentially following a random order. A cycle over all training minibatches is called training epoch, which results in a number of parameters updates equal to the number of minibatches. The principle of CD consists in running many MC chains in parallel, using each sample of the minibatch as the initial condition for one of those chains. CD is based upon the idea that the dataset must be a good approximation of the equilibrium samples of a well-trained RBM. However, such assumption does not hold at initial stages of the training, where typical equilibrium samples of the machine may be distant from that of the dataset [13]. In a variation of CD, known as *persistent contrastive divergence* (PCD) [32], the last configuration of each Markov chain is saved and used as the initial condition for that of the subsequent training iteration. The logic of this implementation relies on that, as parameters change smoothly during the learning, the same is expected to happen for the equilibrium configurations of the model.

2.3 Learning the Ising Model with RBMs

Since the Ising model is a simple well known system, but complex enough to capture interesting dynamics, e.g., phase transitions, it has been used by physicists to study the generative power of RBM [10, 33]. Moreover, once an RBM is trained, it is possible, not only to generate statistically similar new data, but to use it as an effective model of the training data. One natural question that arises in this context is if one can use trained RBMs to infer the Hamiltonian that generated the training data [10]. In this section, we will review some recent work where both the generative and the inferential potential of RBMs has been studied using the Ising Model case.

2.3.1 Reproducing the thermodynamic observables of the Ising model

It has been demonstrated that an RBM is capable of modeling thermodynamic observables of the Ising model [10, 33]. In particular, Torlai and Melko [33] trained RBMs with samples of a finite-size square lattice ($L^2 = 64$) Ising system in at distinct temperatures above, below, and at the critical temperature T_c . For all training instances, the values of the training hyperparameters were fixed, varying only the number of hidden nodes. Once trained, they evaluated the magnetization, energy, specific heat, magnetic susceptibility and correlation length on the samples generated by their machines and compare them to those calculated directly from the training set. For $T > T_c$ and $T < T_c$, it was found that RBMs with few hidden nodes ($n = 4$) can readily reproduce the thermodynamics of the Ising model. However, near $T = T_c$, the number of hidden nodes required to faithfully reproduce the specific heat increases ($n = 64$). This growth of the number of hidden nodes needed was attributed to the increase of fluctuations at criticality. Following a similar procedure to that of the Ref. [33], Cossu *et al.* [10] evaluated the magnetization, energy, specific heat and magnetic susceptibility of samples generated by trained machines. In their work, it was

concluded that the RBM is, in general, able to obtain precise values for the first moments of the underlying distribution regardless of the temperature, i.e., for the magnetization and energy, however, the agreement for the second moments, i.e., for the susceptibility and specific heat, is not always as precise.

2.3.2 Extracting couplings from the RBM

Besides performing quality tests of the generative power of RBMs, Cossu *et al.*, in the Ref. [10], introduce an expression for extracting the values of couplings for every N -point interaction between the visible nodes of an RBM. This allows to reconstruct the effective model learned by the machine and, if the machine is well-trained, such model should resemble the Hamiltonian of the training data, i.e., the Ising model. In other words, with such method would be possible to infer the coupling matrix of the Ising system used to train the machine. Now, let us reproduce the calculations presented in Ref. [10] to extract the Ising couplings from an RBM, and then point out a problem in their assumptions. This development will be useful to introduce ours in Section 4.1.

Coupling extraction by Cossu et al. [10]

Marginalizing the Gibbs-Boltzmann distribution for an RBM, given by (2-2), over the hidden nodes gives

$$p(\mathbf{v}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{h}} e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})}. \quad (2-22)$$

Also, we can define a marginal energy $\mathcal{H}(\mathbf{v})$ using

$$p(\mathbf{v}) = \frac{e^{-\mathcal{H}(\mathbf{v})}}{\mathcal{Z}}. \quad (2-23)$$

Combining (2-22), (2-23) and solving for $\mathcal{H}(\mathbf{v})$, we obtain

$$\begin{aligned} \mathcal{H}(\mathbf{v}) &= -\ln \sum_{\mathbf{h}} e^{-\mathcal{H}(\mathbf{v}, \mathbf{h})} \\ &= -\ln \sum_{\mathbf{h}} \exp \left(\sum_j b_j v_j + \sum_i c_i h_i + \sum_{i,j} h_i W_{ij} v_j \right) \\ &= -\sum_j b_j v_j - \sum_i \ln \sum_{h_i} \exp \left(c_i h_i + \sum_j h_i W_{ij} v_j \right) \\ &= -\sum_j b_j v_j - \sum_i \ln \sum_{h_i} q(h_i) e^{t h_i}, \end{aligned} \quad (2-24)$$

where we have defined $q(h_i) \equiv e^{c_i h_i}$ and $t \equiv \sum_j W_{ij} v_j$. This leads us to introduce the following cumulant generating function:

$$K_i(t) \equiv \ln \sum_{h_i} q(h_i) e^{t h_i} = \sum_n \frac{\kappa_i^{(n)} t^n}{n!}, \quad (2-25)$$

where the n th cumulant is obtained from

$$\kappa_i^{(n)} = \left. \frac{\partial^n K_i(t)}{\partial t^n} \right|_{t=0}.$$

Expanding (2-25) in (2-24) gives

$$\begin{aligned} \mathcal{H}(\mathbf{v}) &= - \sum_j b_j v_j - \sum_i \sum_n \frac{\kappa_i^{(n)} t^n}{n!} \\ &= - \sum_j b_j v_j - \sum_i \kappa_i^{(0)} - \sum_i \kappa_i^{(1)} t - \sum_i \frac{\kappa_i^{(2)} t^2}{2!} - \dots \\ &= - \sum_i \kappa_i^{(0)} - \sum_j \left(b_j + \sum_i \kappa_i^{(1)} W_{ij} \right) v_j - \frac{1}{2!} \sum_{j_1, j_2} \left(\sum_i \kappa_i^{(2)} W_{ij_1} W_{ij_2} \right) v_{j_1} v_{j_2} \dots \end{aligned} \quad (2-26)$$

Since the RBM has learned the physics of the Ising model, Cossu *et al.* made the following assumption:

$$\mathcal{H}(\mathbf{v}) = \mathcal{H}_{\text{Ising}}(\mathbf{v}), \quad (2-27)$$

up to a constant. Therefore, we are not interested in the first term in (2-26) because these two energies may differ by an additive constant without affecting the physical observables. However, $\mathcal{H}(\mathbf{v})$ and $\mathcal{H}_{\text{Ising}}(\mathbf{v})$ should be equal order by order in \mathbf{v} .

For the standard Ising interactions, with no external field, one has

$$\mathcal{H}_{\text{Ising}}(\boldsymbol{\sigma}) = \sum_{j_1, j_2} H_{j_1 j_2} \sigma_{j_1} \sigma_{j_2}, \quad (2-28)$$

where \mathbf{H} is the appropriate Ising matrix with nearest neighbors interactions. In this model, nearest neighbors couplings were set to $-1/T$ where T is the temperature of the model. Thus, $H_{j_1 j_2}$ is zero except for components which correspond to nearest neighbors interactions, which are equal to $-1/2T$.

Also, it should be noted that the Ising variables $\sigma_j \in \{-1, 1\}$. To obtain the Ising Hamiltonian in terms of bits, one can apply the following transformation:

$$\sigma_j \equiv 2v_j - 1, \quad (2-29)$$

to (2-28), which gives,

$$\begin{aligned}
\mathcal{H}_{\text{Ising}}(\boldsymbol{\sigma}) &= \sum_{j_1, j_2} H_{j_1 j_2} (2v_{j_1} - 1)(2v_{j_2} - 1) \\
&= 4 \sum_{j_1, j_2} H_{j_1 j_2} v_{j_1} v_{j_2} - 2 \sum_{j_1, j_2} H_{j_1 j_2} v_{j_2} - 2 \sum_{j_1, j_2} H_{j_1 j_2} v_{j_1} + \sum_{j_1, j_2} H_{j_1 j_2} \\
&= 4 \sum_{j_1, j_2} H_{j_1 j_2} v_{j_1} v_{j_2} - 4 \sum_{j_1} \left(\sum_{j_2} H_{j_1 j_2} \right) v_{j_1} + \sum_{j_1, j_2} H_{j_1 j_2} \tag{2-30}
\end{aligned}$$

Now, we can compare (2-26) and (2-30) to extract the coupling from the trained RBM. First, let us note that $v_j^n = v_j$ for any $n > 0$. This implies that in the expansion of the cumulant generating function, higher order terms in n also contribute to the N_v -point function, where N_v is the number of distinct variables. We can make the above statement clearer by rewriting (2-26) as

$$\mathcal{H}(\mathbf{v}) = - \sum_i \kappa_i^{(0)} - \sum_j \left(b_j + \sum_i \kappa_i^{(1)} W_{ij} \right) v_j - \sum_{n>1} \frac{1}{n!} \sum_{j_1 \dots j_n} \left(\sum_i \kappa_i^{(n)} W_{ij_1} \dots W_{ij_n} \right) v_{j_1} \dots v_{j_n}, \tag{2-31}$$

and noting that n can be thought of as counting powers of W_{ij} . In other words, for each n , the contributions from the sum over the visible nodes can be grouped by the number N_v of distinct nodes being multiplied. Thus, the 2-point contributions from all orders in n is given by

$$\begin{aligned}
& \sum_{n>1} \frac{1}{2(n!)} \sum_{0<k<n} \sum_{j_1 \neq j_2} \left(\sum_i \kappa_i^{(n)} \binom{n}{k} W_{ij_1}^k W_{ij_2}^{n-k} \right) v_{j_1} v_{j_2} \\
&= \frac{1}{2} \sum_{n>1} \frac{1}{n!} \sum_{j_1 \neq j_2} \left(\sum_i \kappa_i^{(n)} [(W_{ij_1} + W_{ij_2})^n - (W_{ij_1})^n - (W_{ij_2})^n] \right) v_{j_1} v_{j_2}. \tag{2-32}
\end{aligned}$$

By comparing the 2-point contribution to $H_{\text{Ising}}(\mathbf{v})$ in (2-30), which can be written as $4 \sum_{j_1 \neq j_2} H_{j_1 j_2} v_{j_1} v_{j_2}$, to that in (2-32), it is found that

$$\begin{aligned}
H_{j_1 j_2} &= \frac{1}{8} \sum_{n>1} \frac{1}{n!} \sum_i \kappa_i^{(n)} [(W_{ij_1} + W_{ij_2})^n - (W_{ij_1})^n - (W_{ij_2})^n] \\
&= \frac{1}{8} \sum_{n>1} \frac{1}{n!} \sum_i [(W_{ij_1} + W_{ij_2})^n - (W_{ij_1})^n - (W_{ij_2})^n] \left. \frac{\partial^n K_i(t)}{\partial t^n} \right|_{t=0} \\
&= \frac{1}{8} \sum_i \sum_n \frac{1}{n!} [(W_{ij_1} + W_{ij_2})^n - (W_{ij_1})^n - (W_{ij_2})^n] \left. \frac{\partial^n K_i(t)}{\partial t^n} \right|_{t=0} + \frac{1}{8} \sum_i K_i(0) \\
&= \frac{1}{8} \sum_i (e^{(W_{ij_1} + W_{ij_2})\partial_t} - e^{W_{ij_1}\partial_t} - e^{W_{ij_2}\partial_t} + 1) K_i(t)|_{t=0}. \tag{2-33}
\end{aligned}$$

Substituting the shift operator $e^{a\partial_x} f(x) = f(a+x)$ in the above expression finally gives,

$$\begin{aligned}
H_{j_1 j_2} &= \frac{1}{8} \sum_i \left(K_i(W_{ij_1} + W_{ij_2}) - K_i(W_{ij_1}) - K_i(W_{ij_2}) + K_i(0) \right) \\
&= \frac{1}{8} \sum_i \left(\ln(1 + e^{c_i + W_{ij_1} + W_{ij_2}}) - \ln(1 + e^{c_i + W_{ij_1}}) - \ln(1 + e^{c_i + W_{ij_2}}) + \ln(1 + e^{c_i}) \right) \\
&= \frac{1}{8} \sum_i \ln \left[\frac{(1 + e^{c_i + W_{ij_1} + W_{ij_2}})(1 + e^{c_i})}{(1 + e^{c_i + W_{ij_1}})(1 + e^{c_i + W_{ij_2}})} \right]. \tag{2-34}
\end{aligned}$$

By following an analogous procedure, Cossu *et al.* could obtain the expression for couplings for an arbitrary N -body interaction between visible nodes [10]:

$$H_{j_1 \dots j_N} = \frac{1}{N!} \sum_{l=0}^N (-1)^l \sum_{\alpha_1 < \dots < \alpha_{N-l}} \sum_i K_i(W_{ij_{\alpha_1}} + \dots + W_{ij_{\alpha_{N-l}}}). \tag{2-35}$$

The strong limitation of these results

The inconvenience of this approach lies in assuming that the latent representation of the Ising Hamiltonian learned by the machine, which is what appears on the right-hand side of (2-27), has up to two point correlations, as in shown in (2-28). Since RBMs are able to model N -body couplings between visible nodes at every N , a more general approach should assume that the Ising-like Hamiltonian inferred by the RBM could have the following general form:

$$\mathcal{H}_{\text{Ising}}(\boldsymbol{\sigma}) = \sum_{j_1} H_{j_1} \sigma_{j_1} + \sum_{j_1, j_2} H_{j_1 j_2} \sigma_{j_1} \sigma_{j_2} + \sum_{j_1, j_2, j_3} H_{j_1 j_2 j_3} \sigma_{j_1} \sigma_{j_2} \sigma_{j_3} + \dots \tag{2-36}$$

In this thesis, such assumption will be relaxed and the corresponding expression for $H_{j_1 j_2 \dots j_N}$ will be derived in Section 4.1.

3 Materials and methods

Evaluating the generative and inferential potential of RBMs with the Ising model requires datasets of equilibrium samples of such models. In the following sections, we will explain the simulation techniques used, followed by the data analysis performed in order to generate the datasets of independent and in equilibrium Ising samples. Finally, we will expose how the RBMs trainings were implemented, giving some general features and the hyperparameters used in all instances.

3.1 Ising model simulations and dataset generation

To train the RBMs, we generated Ising spin configurations in a 1D chain and in a 2D square lattice, both with periodic boundary conditions. The Hamiltonian of the model is

$$\mathcal{H}_{\text{Ising}}(\boldsymbol{\sigma}) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \quad (3-1)$$

where $\langle i, j \rangle$ indicates that the sum is over nearest neighbors spins. The probability measure of any configuration of spins $\boldsymbol{\sigma}$ is given by the Gibbs-Boltzmann distribution:

$$p(\boldsymbol{\sigma}) = \frac{1}{\mathcal{Z}_{\text{Ising}}} e^{-\beta \mathcal{H}_{\text{Ising}}(\boldsymbol{\sigma})}. \quad (3-2)$$

Where, $\beta = (k_b T)^{-1}$, denoting the Boltzmann constant as k_b and the temperature as T . In all simulation instances, it was set that $J/k_b = 1$. In (3-2), the Ising partition function was introduced, which is simply given by

$$\mathcal{Z}_{\text{Ising}} = \sum_{\boldsymbol{\sigma}} e^{-\beta \mathcal{H}_{\text{Ising}}(\boldsymbol{\sigma})}. \quad (3-3)$$

In such a model, each spin can take a binary values $\sigma_i \in \{1, -1\}$, but the visible v_j and hidden variables h_i used in the machines satisfy that $v_j, h_i \in \{0, 1\}$. Thus, to use the generated samples of the Ising systems as the training dataset, the following transformation was applied:

$$v_j^{(d)} \equiv \frac{1}{2}(\sigma_j^{(d)} + 1). \quad (3-4)$$

The samples of this model were generated using the Metropolis-Hastings algorithm [16, 26] for the 1D system and the Wolff algorithm [36] for the 2D system. These algorithms were

implemented in Cython [5], a superset of the Python programming language [35] designed to provide a C-like performance with code that resembles the Python syntax. To generate each training set, we started a long run and saved configurations with a fixed frequency of elementary MC steps (EMCS). An EMCS refers to an attempt of updating $N = L^D$ spins. The time we set to wait until save the first configuration was, at least, 25τ , where τ is the estimated integrated autocorrelation time of the corresponding simulation (see Section 3.2). Once the first configuration is saved, we start to save configurations, at most one every 2.5τ .

3.2 Simulation data analysis

In order to guarantee that the simulation algorithms are properly implemented and the samples correspond to independent configurations of an Ising system at equilibrium, we analyzed the ergodic behavior of our simulation following the recipes found in Ref. [2]. To do such analysis, we ran a long simulation, 10^6 or 10^7 EMCS and measured different observables at every EMCS t . The observables that we analyzed were the magnetization m , the absolute magnetization m_{abs} and the energy per spin e , and we also verified the Schwinger-Dyson equalities for the Ising model [4], whose corresponding exponential expression is denoted by A . Such measures were calculated using the following expressions:

$$m^{(t)} = \frac{1}{L^D} \sum_i \sigma_i^{(t)}, \quad (3-5)$$

$$m_{\text{abs}}^{(t)} = \frac{1}{L^D} \left| \sum_i \sigma_i^{(t)} \right|, \quad (3-6)$$

$$e^{(t)} = -\frac{1}{L^D} \sum_{\langle ij \rangle} \sigma_i^{(t)} \sigma_j^{(t)}, \quad (3-7)$$

$$A^{(t)} = \frac{1}{L^D} \sum_i \exp \left(-2\beta \sigma_i^{(t)} \sum_j J_{ij} \sigma_j^{(t)} \right). \quad (3-8)$$

Where $J_{ij} = 1$ in (3-8) if and only if site i and site j are neighbors, otherwise $J_{ij} = 0$. Since, the Ising system in 1D and 2D square lattice are exactly solvable in the thermodynamic limit, we made sure that the time series obtained for each observable fluctuates around its theoretical expectation (Fig. **3-1**). This was verified quantitatively by taking the mean value of the time series for each observable and compared it to their corresponding theoretical values (Table **3-1**). For an observable O , its estimated mean \bar{O} is given by,

$$\bar{O} \equiv \frac{1}{N_t} \sum_{t=1}^{N_t} O_t, \quad (3-9)$$

where N_t is the total number of measures in the time series. It should be mentioned that, to be completely sure that what we are analyzing is indeed the ergodic behavior of the simula-

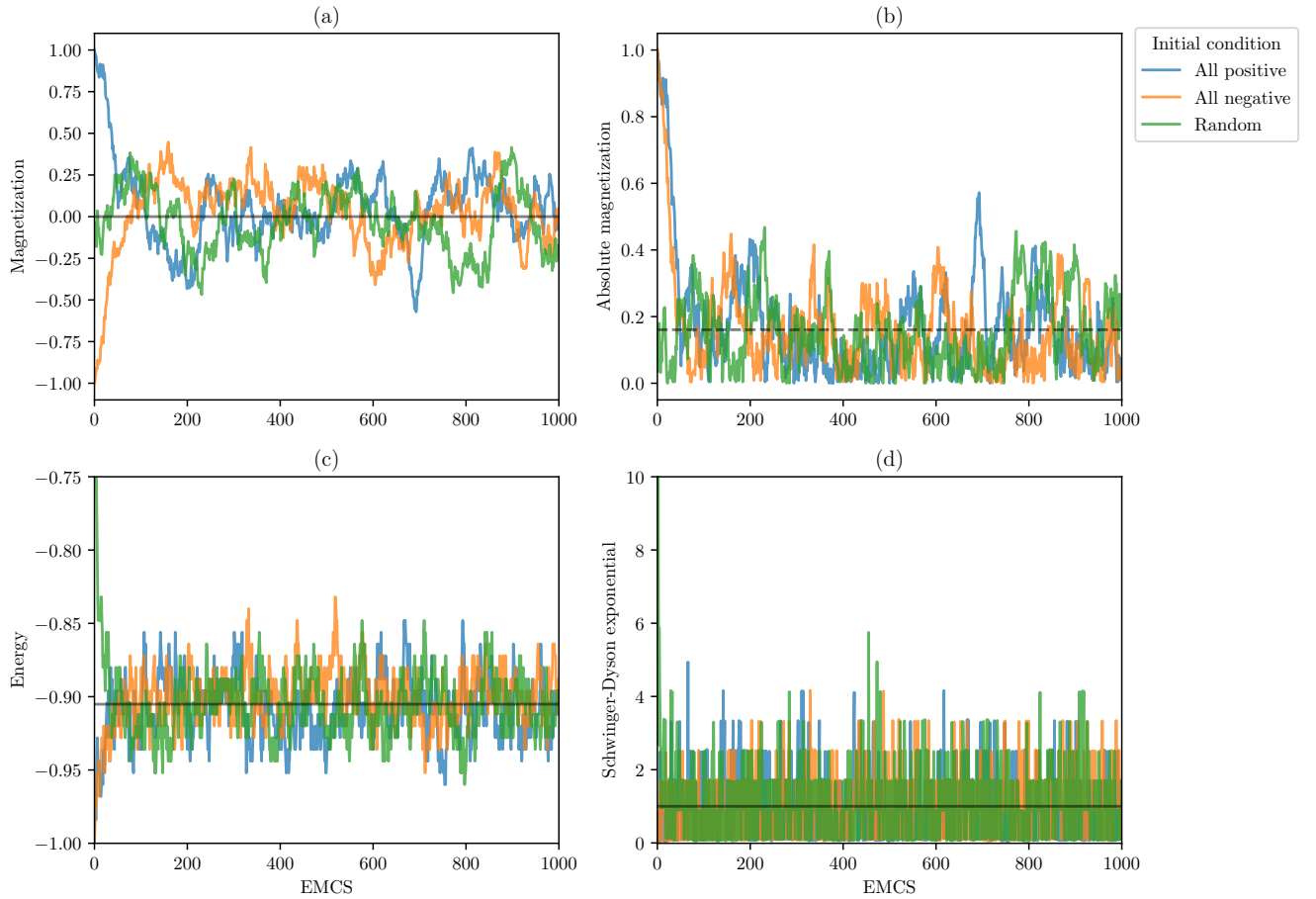


Figure **3-1**: Time-series of the observables for the first 1000 EMCS of a simulation of a 1D Ising system ($L = 500$, $\beta = 1.5$, $J = 1$). The Observables measured were the magnetization per spin (a), the absolute magnetization (b), the energy per spin (c) and the Schwinger-Dyson exponential (d). In (a), (c), and (d), the black line indicates their theoretical mean values (see Table **3-1**). In (b) the punctuated black line indicates the mean value of the time series. It is also observed that the ergodic behavior of our simulations is independent of its initial condition, which are all positive and all negative spins, given by the blue and orange line, respectively, and random configuration given by the green line.

Table **3-1**: Comparison between the mean of the time series of observables and their calculated theoretical mean. These values were obtained from the simulation of an 1D Ising system ($L = 500$, $\beta = 1.5$, $J = 1$), with all positive spins as initial condition (see Fig. **3-1**). It is observed that the theoretical value lies in the 2σ interval centered in the estimated mean. Since, the 1D Ising model only has a paramagnetic phase, one may expect absolute magnetization to vanish. However, this only effectively occurs when $L \rightarrow \infty$. Thus, the presence of a small but finite value for this observable in our simulation is due to the fact that L is finite.

Observables	Estimated mean	Theoretical mean
Magnetization	$1 (3) \times 10^{-3}$	0.0
Abs. Magnetization	$1.598 (11) \times 10^{-1}$	-
Energy	$-9.0519 (11) \times 10^{-1}$	-9.0515×10^{-1}
Schwinger-Dyson	$9.973 (17) \times 10^{-1}$	1.0

tion, we discard the first half of the simulation in all instances. We verified later that this time was long enough to converge to equilibrium.

Once the time series of the observables are calculated, we estimated the corresponding autocorrelations functions (Fig. **3-2**). For an observable O , the unnormalized autocorrelation function is estimated using

$$C_{OO}(t) = \frac{1}{N_t - |t|} \sum_{s=1}^{N_t - |t|} (O_s - \bar{O})(O_{s+|t|} - \bar{O}), \quad (3-10)$$

where \bar{O} in the above expression can be replaced by the theoretical mean if it is known. For the normalized autocorrelation function, we have

$$\rho_{OO}(t) = \frac{C_{OO}(t)}{C_{OO}(0)}. \quad (3-11)$$

Such autocorrelation functions are needed to estimate the correlation time, which was used in the sample generation to guarantee the independence of the samples (see Section 3.1). The integrated correlation time $\tau_{\text{int},O}$ controls the effective number of independent measurements of an observable O in a series of observations obtained from MCMC methods. Given N consecutive observations of O , it can be shown that the number of independent measurements O_t is given by $N/(2\tau_{\text{int},O})$. To estimate the integrated autocorrelation time τ_{int} , we use the following relation:

$$\tau_{\text{int},O} = \frac{1}{2} + \sum_{t=1}^{t_{\text{max}}} \rho_{OO}(t). \quad (3-12)$$

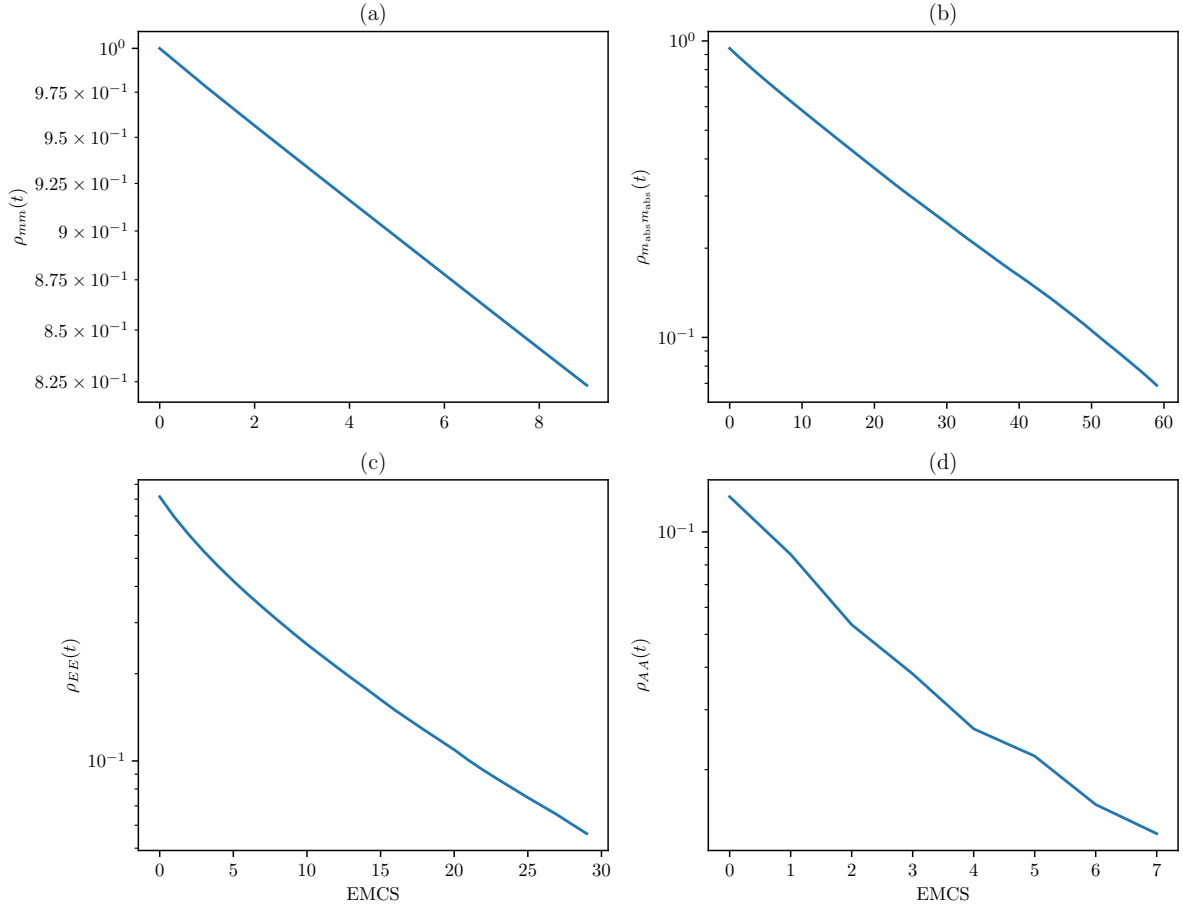


Figure 3-2: Normalized autocorrelation function for the magnetization (a), the absolute magnetization (b), the energy (c) and the Schwinger-Dyson exponential (d). This simulation corresponds to an 1D Ising system ($L = 500$, $\beta = 1.5$, $J = 1$), with all positive spins as initial conditions. Here the reader should note two things. First, the decay of the normalized autocorrelation functions is exponential. Second, in this case, the observable with the slowest dynamics is the magnetization. The decay of the normalized autocorrelation function is given by: $\rho_{OO}(t) \sim e^{-t/\tau_{\text{exp},O}}$. Indeed, when the linear relation was long enough, such a relation was used to estimate the exponential time $\tau_{\text{exp},O}$, which was always a value close to $\tau_{\text{exp},O}$.

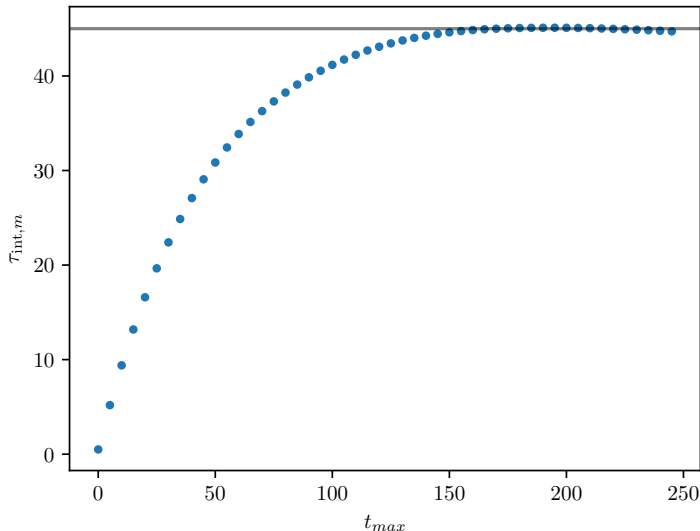


Figure **3-3**: Convergence of the integrated time $\tau_{\text{int},m}$ of magnetization with respect to values of t_{max} in eq. (3-12). The black line indicates the solution for $t_{\text{max}} = 6 \tau_{\text{int},m}$, $\tau_{\text{int},m} \sim 45$. These results correspond to the simulation of an 1D Ising system ($L = 500$, $\beta = 1.5$, $J = 1$), with all positive spins as initial condition

To avoid summing up just error, we see that the sum in the above relation is truncated at some value t_{max} , which is chosen typically as an integer multiple of t_{int} , i.e., $t_{\text{max}} \equiv \alpha \tau_{\text{int}}$. Here, as in Ref. [2], we chose $\alpha = 6$. We show in Fig. **3-3** that the precise value of α is not important unless it is too large. It should be noted that the correlation time of the simulation is taken as that of the observable with the slowest dynamics.

3.3 RBM training

The code regarding RBM training was adapted from `TorchRBM` [11], a PyTorch implementation of RBM available online. In training instances, the RBMs were trained with $N = 10^4$ Ising configurations, applying the stochastic gradient ascent dynamics explained in the section 2.2.2. The size of minibatches was set to $M = 500$ and the learning rate to $\gamma = 0.005$. The matrix \mathbf{W} was initialized from a Gaussian distribution centered around zero and with variance $\sigma = 10^{-4}$.

In addition, to monitor the learning process, we save trained machines in different epochs, and we used them to generate samples from which we measured observables as a function of the EMCS during the Gibbs sampling to evaluate the convergence of these time series to the values calculated directly from the training set. The observables that we used in this part

were: the average absolute magnetization, the magnetic susceptibility, the energy, and heat capacity whose formulas are given by:

$$\langle m \rangle = \frac{1}{L^D} \left\langle \left| \sum_i \sigma_i \right| \right\rangle, \quad (3-13)$$

$$\left\langle \frac{\chi}{L} \right\rangle = \beta (\langle m^2 \rangle - \langle m \rangle^2), \quad (3-14)$$

$$\langle e \rangle = -\frac{1}{L^D} \left\langle \sum_{\langle ij \rangle} \sigma_i \sigma_j \right\rangle, \quad (3-15)$$

$$\langle C \rangle = \beta^2 (\langle e^2 \rangle - \langle e \rangle^2), \quad (3-16)$$

$$\langle A \rangle = \frac{1}{L^D} \left\langle \sum_i \exp \left(-2\beta \sigma_i \sum_j J_{ij} \sigma_j \right) \right\rangle. \quad (3-17)$$

Here, we denote for an observable O its estimated mean $\langle O \rangle$ as

$$\langle O \rangle = \frac{1}{M} \sum_{d=1}^M O(\boldsymbol{\sigma}^{(d)}, \beta), \quad (3-18)$$

where M denotes the number of independent samples $\boldsymbol{\sigma}^{(d)}$. It is important to remember that the visible variables v_j are such that $v_j \in \{0, 1\}$, thus, to calculate the observables mentioned above from samples generated by the machine, one must apply the following transformation

$$\sigma_j^{(d)} \equiv 2v_j^{(d)} - 1. \quad (3-19)$$

4 Results

In this chapter, we discuss the results obtained in this thesis. In Section 4.1, we present the derivation of the relation used to extract the values of couplings for N -body interaction between the visible nodes of an RBM in terms of Ising variables. Then, we use this method to investigate the influence of the training regime in the latent model created by the machine. In Section 4.2 and 4.3, we use samples of an 1D and 2D Ising systems, respectively, to train RBMs with different training schemes. Once machines are trained, we can extract the spin coupling matrices learned and compare it to that of the model that generate the dataset.

4.1 Extracting the effective couplings from the RBM

Here, we introduce a novel method to reconstruct the effective Ising model learned by the machine, overcoming the limitations mentioned in the Section 2.3.2. Let us start by rewriting the expression (2-24) for the marginalization of the Hamiltonian of the machine as

$$\mathcal{H}(\mathbf{v}) = - \sum_j b_j v_j - \sum_i \ln \sum_{h_i} \exp \left(c_i h_i + \sum_j h_i W_{ij} v_j \right). \quad (4-1)$$

Our goal is to expand the above expression as a generalized Ising Hamiltonian, that also contains higher order interactions, of the form:

$$\mathcal{H}(\boldsymbol{\sigma}) = - \sum_j h_j \sigma_j - \sum_{j_1 < j_2} H_{j_1 j_2} \sigma_{j_1} \sigma_{j_2} - \sum_{j_1 < j_2 < j_3} H_{j_1 j_2 j_3} \sigma_{j_1} \sigma_{j_2} \sigma_{j_3} + \dots \quad (4-2)$$

Applying the following change of variables:

$$v_j \equiv \frac{1}{2}(\sigma_j + 1), \quad h_i \equiv \frac{1}{2}(\tau_i + 1), \quad (4-3)$$

to (4-1), one can obtain,

$$\begin{aligned}
\mathcal{H}(\boldsymbol{\sigma}) &= -\frac{1}{2} \sum_j b_j - \frac{1}{2} \sum_i c_i - \frac{1}{4} \sum_{i,j} W_{ij} - \frac{1}{2} \sum_j \left(b_j + \frac{1}{2} \sum_i W_{ij} \right) \sigma_j \\
&\quad - \sum_i \ln \sum_{\tau_i} \exp \left(\frac{1}{2} c_i \tau_i + \frac{1}{4} \sum_j \tau_i W_{ij} + \frac{1}{4} \sum_j \tau_i W_{ij} \sigma_j \right) \\
&= -\frac{1}{2} \sum_j b_j - \frac{1}{2} \sum_i c_i - \frac{1}{4} \sum_{i,j} W_{ij} - n \ln 2 - \frac{1}{2} \sum_j \left(b_j + \frac{1}{2} \sum_i W_{ij} \right) \sigma_j \\
&\quad - \sum_i \ln \cosh \left[\frac{1}{2} \left(c_i + \frac{1}{2} \sum_j W_{ij} \right) + \frac{1}{4} \sum_j W_{ij} \sigma_j \right]. \tag{4-4}
\end{aligned}$$

Since (4-4) and (4-2) may differ by an additive constant without affecting the physical observables, the above expression is equivalent to

$$\begin{aligned}
\mathcal{H}(\boldsymbol{\sigma}) &= -\frac{1}{2} \sum_j \left(b_j + \frac{1}{2} \sum_i W_{ij} \right) \sigma_j \\
&\quad - \sum_i \ln \cosh \left[\frac{1}{2} \left(c_i + \frac{1}{2} \sum_j W_{ij} \right) + \frac{1}{4} \sum_j W_{ij} \sigma_j \right]. \tag{4-5}
\end{aligned}$$

By defining,

$$\eta_j \equiv \frac{1}{2} \left(b_j + \frac{1}{2} \sum_i W_{ij} \right), \quad \theta_i \equiv \frac{1}{2} \left(c_i + \frac{1}{2} \sum_j W_{ij} \right), \quad w_{ij} \equiv \frac{1}{4} W_{ij}, \tag{4-6}$$

it allows us to rewrite (4-5) as

$$\mathcal{H}(\boldsymbol{\sigma}) = - \sum_j \eta_j \sigma_j - \sum_i \ln \cosh \left(\sum_j w_{ij} \sigma_j + \theta_i \right). \tag{4-7}$$

To make explicit the expansion of non-linear terms in the above expression, one can use the following artifact:

$$\mathcal{H}(\boldsymbol{\sigma}) = - \sum_j \eta_j \sigma_j - \frac{1}{2^m} \sum_{\boldsymbol{\sigma}'} \prod_j (1 + \sigma_j \sigma'_j) \sum_i \ln \cosh \left(\sum_j w_{ij} \sigma'_j + \theta_i \right), \tag{4-8}$$

where new variables $\sigma'_j \in \{-1, 1\}$ were introduced. To see why (4-7) and (4-8) are equivalent, let us point out that

$$\frac{1}{2} (1 + \sigma_j \sigma'_j) = \delta_{\sigma_j \sigma'_j}, \tag{4-9}$$

where δ is the Kronecker delta. Therefore, one can rewrite (4-8) as

$$\mathcal{H}(\boldsymbol{\sigma}) = - \sum_j \eta_j \sigma_j - \sum_{\boldsymbol{\sigma}'} \prod_j \delta_{\sigma_j \sigma'_j} \sum_i \ln \cosh \left(\sum_j w_{ij} \sigma'_j + \theta_i \right). \quad (4-10)$$

After computing the sum over all the possible configurations of $\boldsymbol{\sigma}'$ in (4-10), we note only the term having $\sigma_j = \sigma'_j$, for all j , survives, which automatically leads us to (4-7). Returning to (4-8) and by comparing it to (4-2), the following identification is obtained:

$$H_{j_1 \dots j_N} = \frac{1}{2^N} \sum_{\boldsymbol{\sigma}'} \sum_i \sigma'_{j_1} \dots \sigma'_{j_N} \ln \cosh \left(\sum_j w_{ij} \sigma'_j + \theta_i \right). \quad (4-11)$$

To reduce the above expression to a more transparent and suitable form for evaluation, let us introduce the following random variable:

$$X_i^{(j_1 \dots j_N)} \equiv \sum_{j \neq j_1, \dots, j_N} w_{ij} \sigma'_j, \quad (4-12)$$

where each σ'_j is a random variable uncorrelated from w_{ij} that can take the values 1 and -1 with equal probability. Thus, (4-11) can be rewritten in terms of an expected value of $x \sim X_i^{(j_1 \dots j_N)}$:

$$H_{j_1 \dots j_N} = \frac{1}{2^N} \sum_i \mathbb{E}_{x \sim X_i^{(j_1 \dots j_N)}} \left[\sum_{\sigma'_{j_1}, \dots, \sigma'_{j_N}} \sigma'_{j_1} \dots \sigma'_{j_N} \ln \cosh \left(\sum_{l=1}^N w_{ij_l} \sigma'_{j_l} + x + \theta_i \right) \right]. \quad (4-13)$$

Finally, we assumed that the central limit theorem can be used to approximate the distribution of $X_i^{j_1 \dots j_N}$, therefore:

$$X_i^{j_1 \dots j_N} \rightarrow \mathcal{N} \left(0, \sum_{j \neq j_1, \dots, j_N} w_{ij}^2 \right). \quad (4-14)$$

4.2 Learning the 1D Ising model

In this section, it will be studied how the training scheme affects the generative power and the latent model learned by an RBM. For the latter, the inference method presented in Section 4.1 was used. All RBMs presented in this section were trained with samples of an 1D Ising chain ($L = 500, J = 1, \beta = 0.5$) (see Table 4-1). We chose to start with such a system because it was previously demonstrated that RBMs are able to model the probability distribution of the 1D Ising model with great precision [10, 33]. Additionally, 1D Ising systems are well suited to investigate further applications of RBMs. In particular, we will be interested to apply the results of this thesis to inference in large dataset of protein sequences, which are indeed chains of ~ 400 amino acids [9] and 1D Ising model has found applications

Table 4-1: Estimated mean of the thermodynamics observables of the generated samples of an 1D Ising chain ($L = 500, \beta = 0.5, J = 1$). These samples were used as the training set for treatments presented in Section 4.2.

Observables	Estimated mean
Abs. magnetization	$5.82 (4) \times 10^{-2}$
M. susceptibility	$4.82 (3) \times 10^{-1}$
Energy	$-4.626 (4) \times 10^{-1}$
Specific heat	$1.195 (3) \times 10^{-1}$
Schwinger-Dyson	$9.982 (10) \times 10^{-1}$

modelling such systems [3, 17].

Since the objective of this computational experiment is to evaluate the effects of the convergence or not to equilibrium of the MCMC methods used during the estimation of the negative part of the gradient, as in the Ref. [13], we initialized those chains randomly, denoting such scheme as Rdm- k (with $k = 10, 20, 50, 100$). In our case this is a convenient set-up mainly for two reasons: (i) The initial conditions of the MCMC are uncorrelated with the data set, (ii) the Gibbs sampling protocol used during the training will be identical to that used for samples generation in the trained machines. Additionally, it was included treatments trained in PCD-50 and CD-50 in order to tell if these training schemes have any effect in reducing the number of k steps needed to reach the equilibrium. In all instances showed in this part, it was implemented the training prescription mentioned in Section 3.3 and the machines were trained for 8000 epochs.

4.2.1 Evolution of the observables of generated samples with the sampling time

As in Ref. [13], we know that equilibrium or out-of-equilibrium training of the RBMs can be unveiled by the time evolution of the observables during the Gibbs sampling. In Fig. 4-1, it can be observed two features of out-of-equilibrium learning regime previously reported by Decelle *et al.* [13]. First, memory effects displayed by out-of-equilibrium trained machines are clearly observed ($k = 10, 20$) for all measured observables. Such effects consist in that RBMs trained in an out-of-equilibrium regime learns to reproduce data statistically similar to the training data after $\sim k$ Gibbs sampling steps, where k is the number of Gibbs steps used in their training to estimate the negative part of the gradient. This effect is clearly observed in Fig. 4-1, where the minimum of the absolute error is obtained after about 10 and 20 Gibbs steps for Rdm-10 and Rdm-20 training schemes, respectively. The second

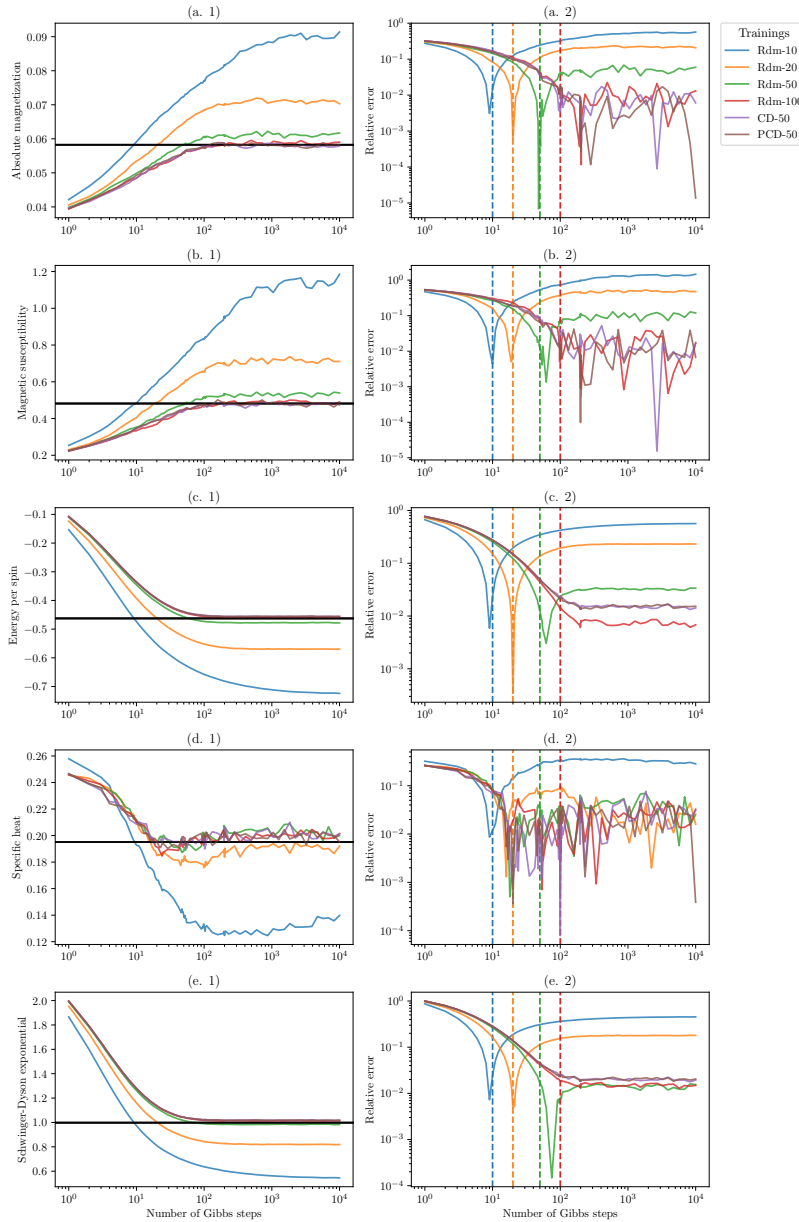


Figure 4-1: Evolution of the observables of generated samples with the sampling time. It is shown the absolute magnetization (a), magnetic susceptibility (b), energy (c), specific heat (d) and the value of Schwinger-Dyson exponential (e) averaged over 10^4 samples of RBMs, with their corresponding relative error, as a function of the number of Gibbs steps. Each color represents a machine trained in a different training scheme. The horizontal black lines in (1) represent the mean value of such observables calculated directly from the training data (see Table 4-1). In (2), blue, orange, green, and red vertical lines are indicates 10, 20, 50, and 100 Gibbs sampling steps, respectively. Such numbers are the number of steps used to estimate the gradient during the training.

Table 4-2: Statistics of inference in each training. In the first column, we present the mean of inferred fields. In the second column, we have the root-mean-squared error for inferred fields δ_h . Third and fourth columns show the mean of 2-body couplings for values greater a lesser than 0.25, respectively. Finally, we show the error Δ_H in the fifth column. For the original Ising model, one simply has $h_j^* = 0$, $H_{j_1 j_2}^{*\geq 0.25} = 0.5$, $H_{j_1 j_2}^{*\leq 0.25} = 0$, for all j , j_1 , and j_2 .

Training	$\overline{h_j}$	δ_h	$\overline{H_{j_1 j_2}^{\geq 0.25}}$	$\overline{H_{j_1 j_2}^{\leq 0.25}}$	Δ_H
Rdm-10	7.07×10^{-4}	2.45×10^{-2}	7.44×10^{-1}	6.37×10^{-4}	8.76×10^{-1}
Rdm-20	2.54×10^{-4}	1.80×10^{-2}	5.87×10^{-1}	3.03×10^{-4}	6.26×10^{-1}
Rdm-50	-7.96×10^{-6}	1.50×10^{-2}	5.22×10^{-1}	7.60×10^{-5}	5.68×10^{-1}
Rdm-100	1.66×10^{-4}	1.43×10^{-2}	5.12×10^{-1}	4.23×10^{-6}	5.62×10^{-1}
PCD-50	-2.16×10^{-4}	1.39×10^{-2}	5.11×10^{-1}	-1.41×10^{-5}	5.58×10^{-1}
CD-50	-2.62×10^{-4}	1.40×10^{-2}	5.12×10^{-1}	-5.26×10^{-6}	5.62×10^{-1}

effect that we evidence is the slow convergence towards an equilibrium value when sampling machines trained in a far from equilibrium regime. In particular, we note that the farther from equilibrium the machine is trained, the slower its dynamics will be during Gibbs sampling. For instance, the sampling of Rdm-10 still evolves after 10^4 steps.

Additionally, it is observed that the equilibrium measures for the Rdm-100 training scheme converges towards the values of the training data set in all cases, except in that of the specific heat, which suggests this is a case of equilibrium training. The discrepancy between the specific heat capacity of RBM samples and that of the training set is a feature also reported by Cossu *et al.* [10], and further investigation is needed to elucidate the reasons.

Finally, let us comment what we found for the treatments with the same $k = 50$. On the one hand, we observe memory effects in Rdm-50, which are more evident in the observables with slower dynamics, e.g. the absolute magnetization or the magnetic susceptibility, as a sharp peak in the error. However, the equilibrium values of the samples of this treatment are not as far from ones measured from the training set as that of Rdm-10 or Rdm-20. Such mixed results suggest that Rdm-50 represents a near equilibrium training regime. On the other hand, we see that the sampling behavior of PCD-50 and CD-50 is similar to that of Rdm-100, which implies that PCD and CD training schemes can decrease the number of k steps needed to ensure equilibrium training.

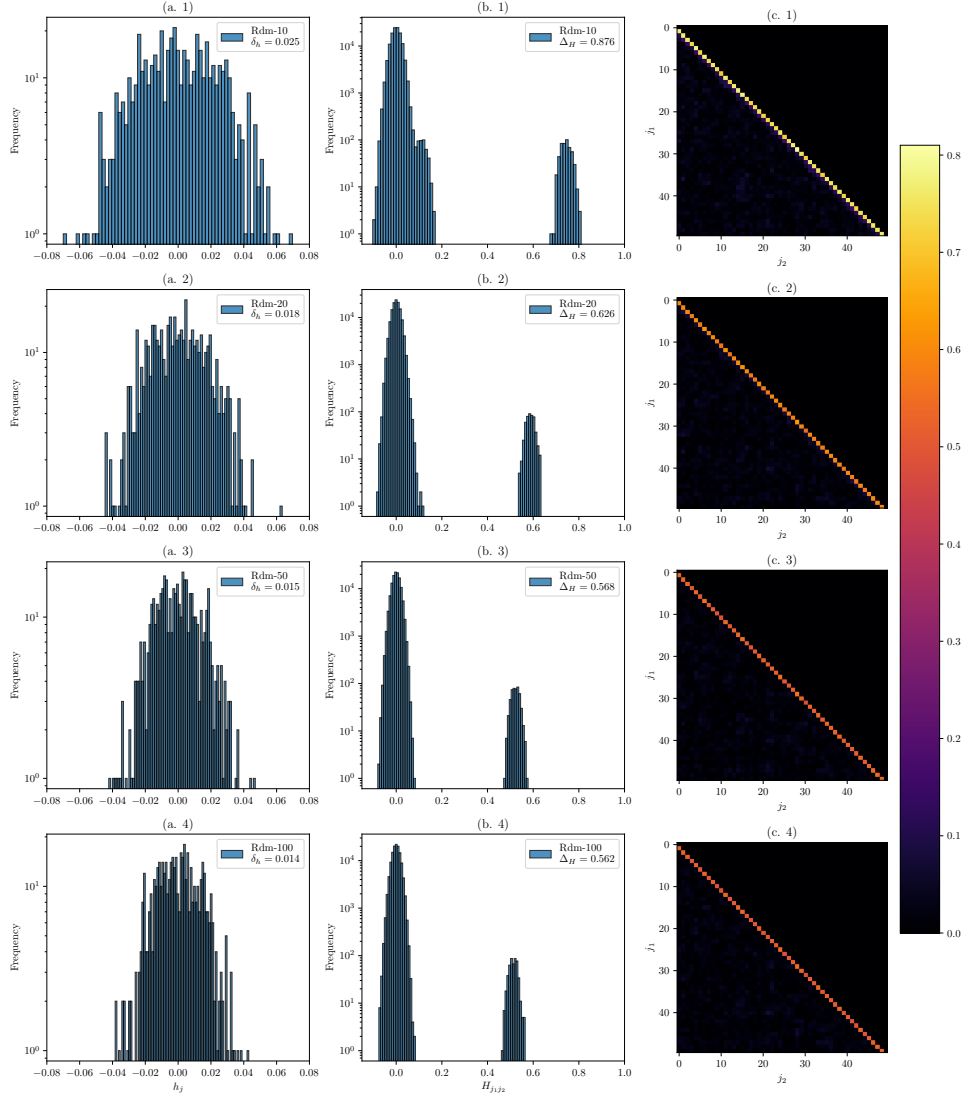


Figure 4-2: (a) histogram of extracted fields, (b) histogram of extracted 2-body couplings, (c) matrix of extracted 2-body couplings, where it is shown the absolute values of the couplings $H_{j_1 j_2}$, where $j_1 > j_2$, for the first 50 indexes. In each row, we present the results for a different training procedure, Rdm- k with $k = 10, 20, 50, 100$ from top to bottom. For the original Ising model, one has $h_j = 0$, for all j , and $H_{j_1 j_2} = 0.5$, if j_1 and j_2 are first neighbors, otherwise $H_{j_1 j_2} = 0$.

4.2.2 Inferring the effective model of in and out-of-equilibrium trained machines

As it was mentioned above, in this case, it was possible to reconstruct the effective model learned during all learning instances. In Fig. 4-2 (a) the distribution of inferred external fields, i.e. the interaction constant in the first term of eq. (4-2), is shown. In both equilibrium and out-of-equilibrium training regimes, estimated couplings are distributed around 0. However, we observe that the inference is better as equilibrium training is reached, since the root-mean error squared decreases with k (see Table 4-2). Besides, in Fig. 4-2 (b, c) the distribution of inferred 2-body couplings and its respective matrix is displayed. To measure the quality of the inference of 2-body couplings, we calculated the mean of inferred couplings greater and lesser than 0.25 (see Table 4-2), where the firsts correspond to the couplings between first neighbors and the latter, the rest of all other possible 2-body interactions. Additionally, we introduce the error in inferred couplings $H_{j_1 j_2}$ with respect to the true ones $H_{j_1 j_2}^*$ [28]:

$$\Delta_H = \sqrt{\frac{\sum_{j_1 > j_2} (H_{j_1 j_2} - H_{j_1 j_2}^*)^2}{\sum_{j_1 > j_2} H_{j_1 j_2}^{*2}}}. \quad (4-15)$$

In general, since the $\overline{H_{j_1 j_2}^{>0.25}}$ is closer to 0.5 as k increases and the decrease of Δ_H in equilibrium and near to equilibrium training schemes, we conclude that the quality of inference is improved as equilibrium training is implemented. In particular, it is observed that the RBM trained in a far-from-equilibrium regime (Rdm-10) not only overestimates the value of inferred couplings, but also learns nonexistent couplings between second neighbors. However, if we compare the inference performed by machines trained in Rdm-50 and Rdm-100 schemes, we note that the inference is only slightly better in the latter, which suggests that near equilibrium training could be enough for some inference purposes.

4.3 Learning the 2D Ising model

To study the influence of the temperature in the effective model learned by the machine, we trained RBMs with 2D Ising samples ($L^2 = 49$, $J = 1$) with β ranging from 0 to 0.65 in intervals of 0.01. In this case, working with the square lattice Ising model is interesting since it presents a phase transition at $T_c = 2/\ln(1 + \sqrt{2}) \approx 2.269\dots$, so the performance of the inference done by the machines can be evaluated in both phases and at T_c . In this test, we used symmetrized RBMs, where the marginalized Hamiltonian in terms of visible Ising

variables $\sigma_j \in \{-1, 1\}$ in eq. (4-5),

$$\mathcal{H}(\boldsymbol{\sigma}) = -\frac{1}{2} \sum_j \left(b_j + \frac{1}{2} \sum_i W_{ij} \right) \sigma_j - \sum_i \ln \cosh \left[\frac{1}{2} \left(c_i + \frac{1}{2} \sum_j W_{ij} \right) + \frac{1}{4} \sum_j W_{ij} \sigma_j \right],$$

has the \mathbb{Z}_2 symmetry. This lead us to the following effective Hamiltonian:

$$\mathcal{H}(\boldsymbol{\sigma}) = - \sum_i \ln \cosh \left(\frac{1}{4} \sum_j W_{ij} \sigma_j \right), \quad (4-16)$$

and the following constraints:

$$b_j \equiv -\frac{1}{2} \sum_i W_{ij}, \quad c_i \equiv -\frac{1}{2} \sum_j W_{ij}. \quad (4-17)$$

All the machines presented in this section were trained for 10^4 epochs.

4.3.1 Observable predictions at all temperatures

Following the Refs. [33, 10], we used trained machines to generate samples from which we calculated thermodynamic observables and compare them to those calculated directly from the dataset. We do not use Onsager's exact solution because we need to take into account finite size effects. In Fig. **4-3**, we recognize that for first moment observables, i.e. the absolute magnetization and the energy per spin, a far for equilibrium trained machines, i.e., Rdm-10, gives the estimation values is close to the target values, although not negligible in the energy case. However, for the second moments, i.e., the magnetic susceptibility and the specific heat, the divergence between the values calculated directly from the dataset and the estimation done using RBM samples is considerable. We attribute such discrepancy partly to the fact that the number of Gibbs steps at which samples were generated was fixed, in this case at $k = 10$, and the number of k steps needed by far from equilibrium trained RBMs to reproduce data statistically similar to the training data may fluctuate.

On the other hand, there are no considerable differences among the CD-50, PCD-50, and PCD-150 training schemes. Since we do not observe any dependence of k in the quality, we conclude that equilibrium, or at least near equilibrium, was implemented in such training schemes at all temperatures. Similarly to what was obtained by Cossu *et al.* in Ref. [10], we observed that there is a better agreement between RBM samples and training data in first moments, i.e., the magnetization and the energy, compared to the second moments, i.e., the susceptibility and the specific heat. An important discrepancy is found in the specific heat

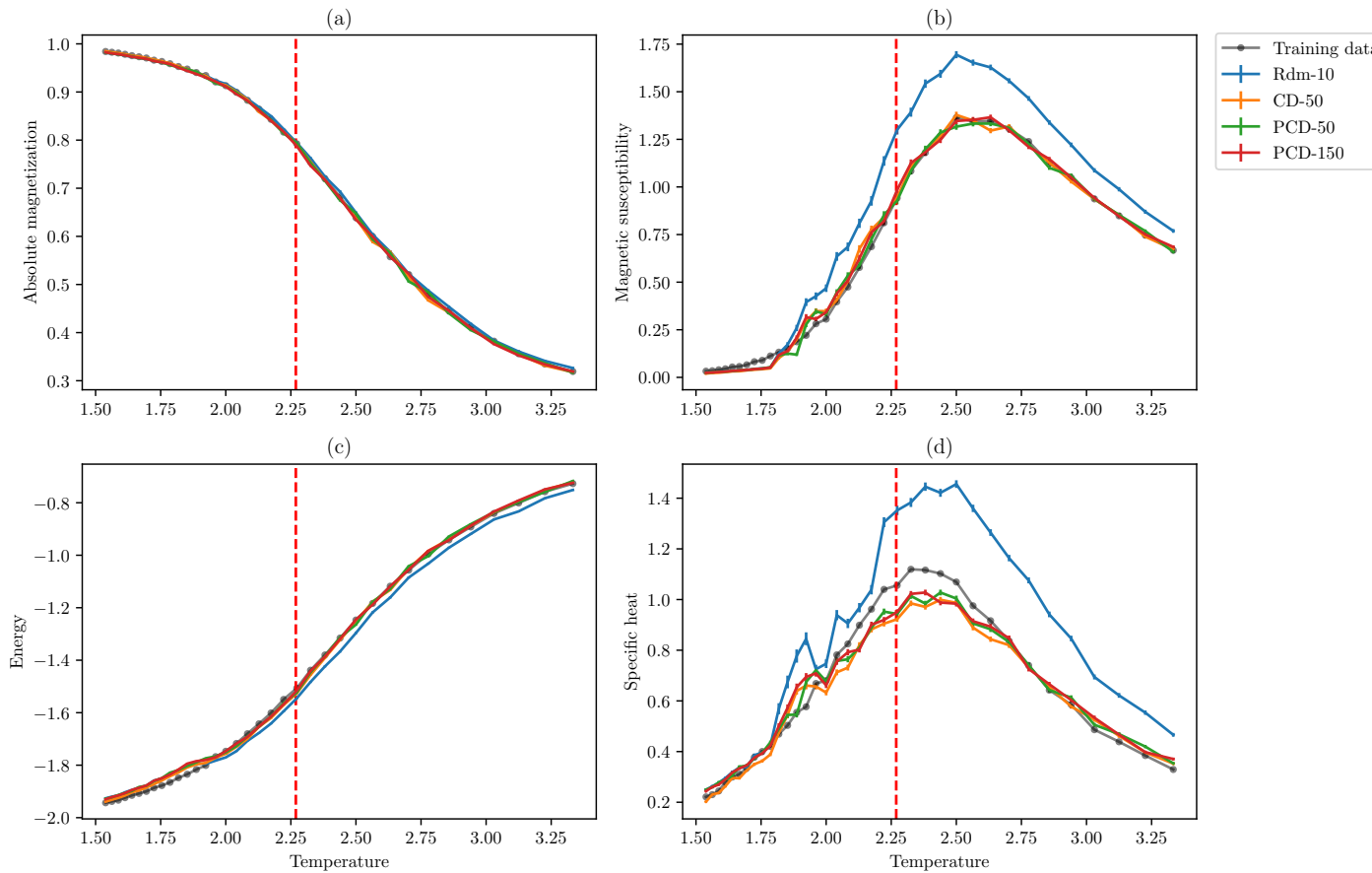


Figure 4-3: Observables at every temperature. Black pointed lines represent the values calculated directly from the training dataset. Each color line represents a different training scheme. The results for CD-50, PCD-50 and PCD-150 were obtained from 10^4 samples after 10^3 Gibbs steps. The results for Rdm-10, were obtained after 10 Gibbs steps. The punctuated red line indicates the critical point T_c .

near to the critical point, and as it was suggested in Section 4.2.1 other factors, additionally to the implementation of equilibrium or out-of-equilibrium regimes, must be taken into account in order to overcome this problem.

4.3.2 Inferring the effective model at all temperatures

As in the previous case, we can also infer the effective model learned by the machines in all training instances. Then, we can compute the error couplings, Δ_H whose formula is given in (4-15). In Fig. 4-4 (a) we present the value of δ_H for all training instances. Remarkable differences between PCD and CD were not observed in the effective models of the machines. We also note that the performance of inference is best at intermediate temperature values in paramagnetic phase. This can be explained if we consider that, at high temperature, thermal fluctuations may hinder the learning of the couplings by the machine. Also, near the critical point, the increase of fluctuations may impede that machine to construct a good latent model for the dataset. This is also confirmed by the increase of the variance of the inferred couplings near the critical point in Fig. 4-4 (b) and (c). Additionally, the overestimation of inferred couplings and the prediction of non-existing couplings near the critical value are features that suggest that out-of-equilibrium features are present (see Fig. 4-5 (2)), therefore, critical slowdown in sampling dynamics of the machine may be present. However, more experiments addressing specifically this question are needed to give a clear answer to this question. At low temperatures, the problem with the latent model generated by the machine is most likely due to the impossibility to properly thermalize during the Gibbs sampling. However, this problem could be overcome simply by using a *tethered sampling method* [6].

Finally, it should be pointed out that other factors beyond the training scheme may increase the quality of the model learned by the machines. For instance, if we compare the matrices and histograms in Fig. 4-5 with those presented by Cossu *et al.*, one may find more noise in the former case. However, it should be pointed out that the datasets used in the latter were at least 10 times bigger than those we used. Therefore, the size of training data may have also played an important role in the latent model created by the machine.

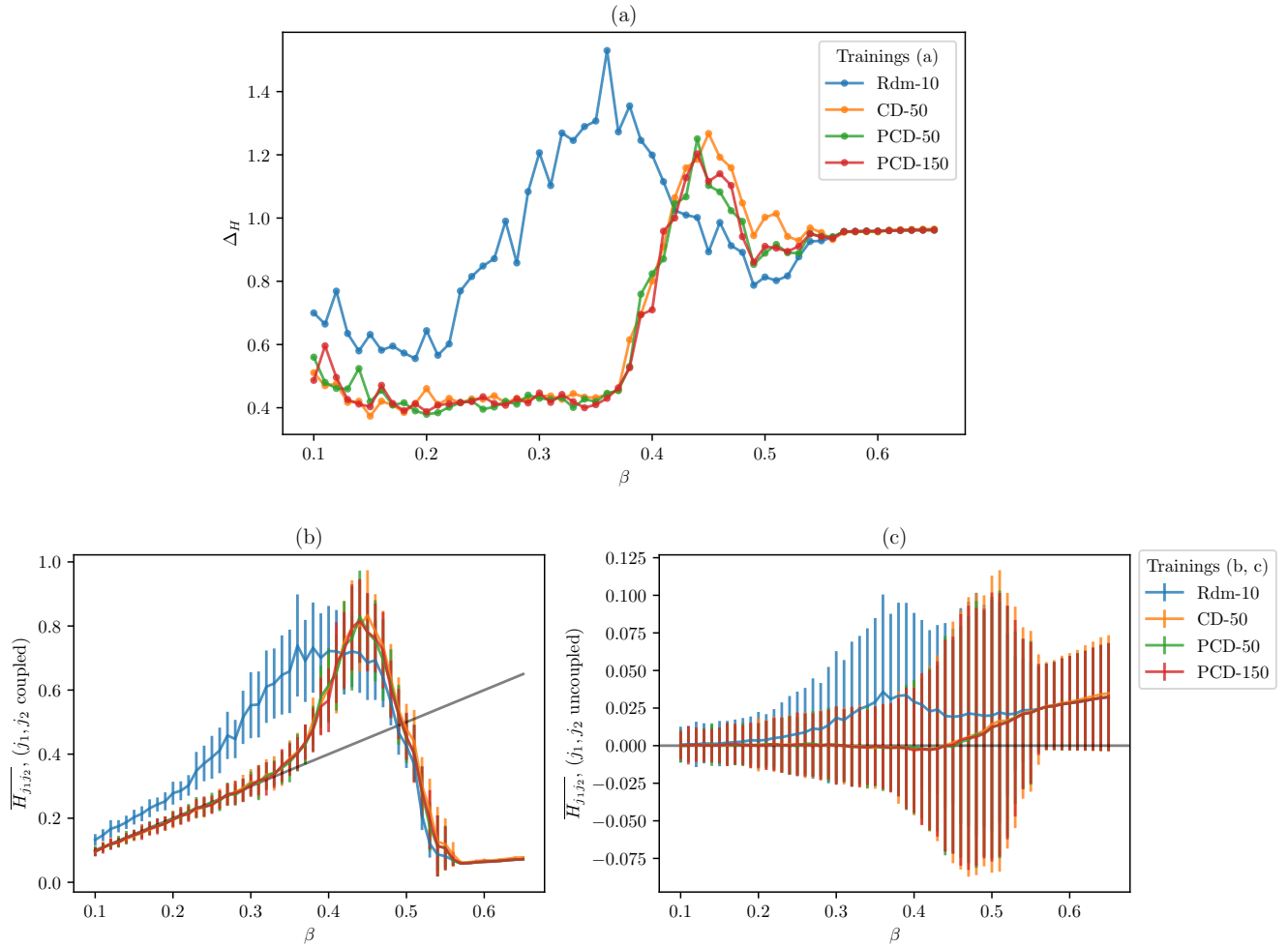


Figure 4-4: Quality measures for inference at all temperatures. Error in inferred couplings Δ_H is shown in (a). The mean of inferred couplings for coupled and uncoupled sites is presented in (b) and (c) respectively. In (b) and (c), The error bars are given by the standard deviation of inferred couplings at each temperature and the black line indicates the value of the couplings in the corresponding Ising model.

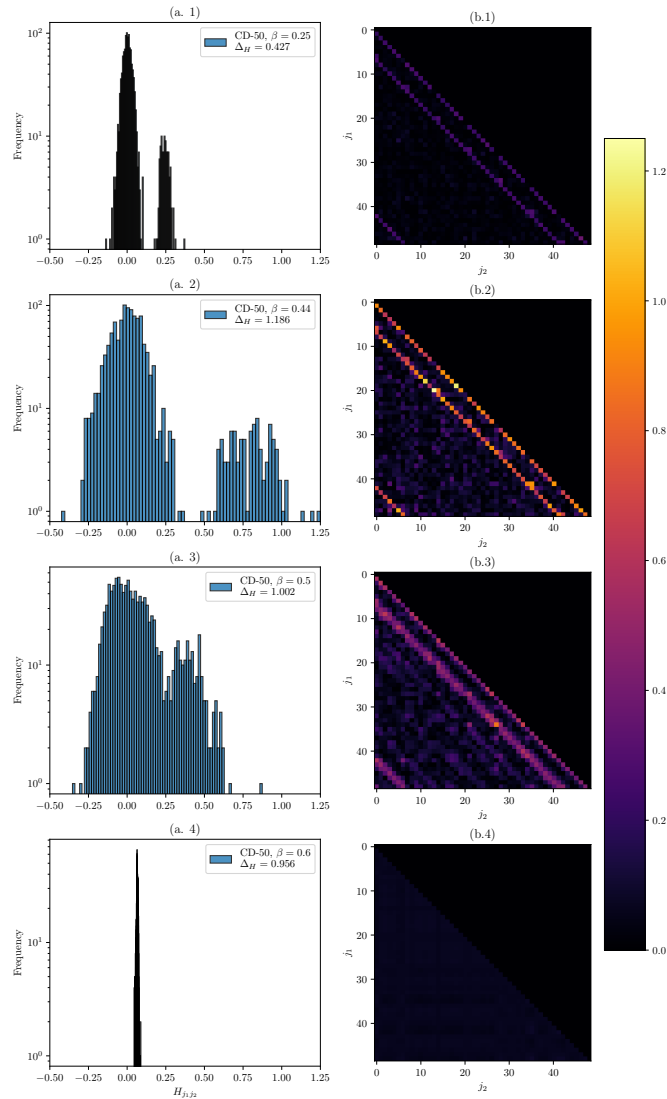


Figure 4-5: For an RBM trained in PCD-50 scheme with Ising in the square lattice samples ($L^2 = 49$, $J = 1$), we show the histogram of extracted 2-body couplings (a) and the matrix of extracted 2-body couplings (b), where it is shown the absolute values of the couplings $H_{j_1 j_2}$, where $j_1 > j_2$. In each row, we present the results for different temperatures, $\beta = 0.25, 0.44, 0.5, 0.6$, from top to bottom.

5 Conclusions and perspectives

In this Master’s thesis, we use samples of the Ising model to train RBM to investigate how in and out-of-equilibrium training regime affects the features that the machine can extract. To achieve this goal, it was needed to introduce a novel method to represent the latent model of interactions between visible variables of an RBM in terms of Ising variables $\sigma \in \{-1, 1\}$. With this method, it was possible to reconstruct the effective model of RBMs trained in different learning regimes and instances of the Ising model.

During a first experiment, we used samples of 1D Ising chain of $L = 500$, $J = 1$, and $\beta = 0.5$ to train machines in different learning regimes. Besides finding previously reported features of out-of-equilibrium learning regimes, such as memory effects and slow dynamics [13], using the above-mentioned novel inference method it was possible to observe that RBMs trained in far-from-equilibrium regime overestimate the value of inferred couplings and learns non-existent couplings, in this case, between second neighbors. Additionally, we suggest that in equilibrium learning regime is not necessary to obtain satisfactory inference results, since with a near equilibrium regime one may get almost as good results. Then, we train RBMs with samples of 2D Ising square lattice system to assess how the temperature of the sample affects the inference performed by the machine. In this case, we found that best results are obtained at intermediate β range in the paramagnetic phase. However, these results are still very incipient and more research is needed to improve the inference power of this ML method.

Furthermore, it should be pointed out that this is the first work where the reconstruction of the latent representation of the RBM into a physically interpretable model is used to study features of its learning. An analogous methodology may be implemented to understand the impact of any other training variables in the latent representation of the model. In general, such inference method is useful whenever one is interested to translate an already existent RBM model into a more physically interpretable effective model between the visible variables.

Additionally, once an RBM model could capture the underlying distribution of a dataset with great accuracy, one may use an analogous method to that proposed here to actually infer things regarding the data used during the RBM training. Although, RBM training is still a challenge, such a model could, in theory, capture arbitrary N -body couplings interactions among visible nodes (see eq. (4-13)), which leaves the door open to RBMs to be used for inference of beyond 2-body couplings, a little explored area in the literature.

6 Acknowledgements

I would like to express my deep gratitude to Professors Aurélien Decelle and Professor Beatriz Seoane for their close guidance, patient instruction, enthusiastic encouragement, constructive advice of this research work.

I would also like to thank Professor José Jairo Giraldo for his support, advice and feedback during the conducting of this research and writing of this manuscript.

I would also want to thanks to the group of Disordered Systems Group for providing me the hardware needed to carry out this research and the Department of Theoretical Physics of the Complutense University of Madrid for their reception.

Finally, I wish to thank my parents, my aunt, my brothers and friends –in Cartagena, Bogotá, Madrid and all over the world– for their support and encouragement throughout my study.

Bibliography

- [1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.
- [2] D. J. Amit and V. Martín-Mayor. *Field theory, the renormalization group, and critical phenomena: graphs to computers*. World Scientific Publishing Company, 2005.
- [3] A. Bakk and J. S. Høye. One-dimensional ising model applied to protein folding. *Physica A: Statistical Mechanics and its Applications*, 323:504–518, 2003.
- [4] H. Ballesteros and V. Martín-Mayor. Test for random number generators: Schwinger-dyson equations for the ising model. *Physical Review E*, 58(5):6787, 1998.
- [5] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.
- [6] N. Béreux, A. Decelle, C. Furtlehner, and B. Seoane. Learning a restricted boltzmann machine using biased monte carlo sampling. *arXiv preprint arXiv:2206.01310*, 2022.
- [7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.
- [8] B. Bravi, J. Tubiana, S. Cocco, R. Monasson, T. Mora, and A. M. Walczak. Rbm-mhc: A semi-supervised machine-learning method for sample-specific prediction of antigen presentation by hla-i alleles. *Cell systems*, 12(2):195–202, 2021.
- [9] L. Brocchieri and S. Karlin. Protein length in eukaryotic and prokaryotic proteomes. *Nucleic acids research*, 33(10):3390–3400, 2005.
- [10] G. Cossu, L. Del Debbio, T. Giani, A. Khamseh, and M. Wilson. Machine learning determination of dynamical parameters: The ising model case. *Physical Review B*, 100(6):064304, 2019.
- [11] A. Decelle. Torchrbm. <https://github.com/AurelienDecelle/TorchRBM>, 2021. Accessed: 20-07-2022.
- [12] A. Decelle and C. Furtlehner. Restricted boltzmann machine: Recent advances and mean-field theory. *Chinese Physics B*, 30(4):040202, 2021.

-
- [13] A. Decelle, C. Furtlehner, and B. Seoane. Equilibrium and non-equilibrium regimes in the learning of restricted boltzmann machines. *arXiv preprint arXiv:2105.13889*, 2021.
- [14] A. Fischer and C. Igel. An introduction to restricted boltzmann machines. In *Iberoamerican congress on pattern recognition*, pages 14–36. Springer, 2012.
- [15] M. Harsh, J. Tubiana, S. Cocco, and R. Monasson. ‘place-cell’ emergence and learning of invariant data with restricted boltzmann machines: breaking and dynamical restoration of continuous symmetries in the weight space. *Journal of Physics A: Mathematical and Theoretical*, 53(17):174002, 2020.
- [16] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1953.
- [17] T. L. Hill. Generalization of the one-dimensional ising model applicable to helix transitions in nucleic acids and proteins. *The Journal of Chemical Physics*, 30(2):383–387, 1959.
- [18] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- [19] G. E. Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.
- [20] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
- [21] R. D. Hjelm, V. D. Calhoun, R. Salakhutdinov, E. A. Allen, T. Adali, and S. M. Plis. Restricted boltzmann machines for neuroimaging: an application in identifying intrinsic networks. *NeuroImage*, 96:245–260, 2014.
- [22] N. Le Roux and Y. Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural computation*, 20(6):1631–1649, 2008.
- [23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [24] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 810:1–124, 2019.
- [25] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac. Restricted boltzmann machines in quantum physics. *Nature Physics*, 15(9):887–892, 2019.

-
- [26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [27] G. Montúfar. Restricted boltzmann machines: Introduction and review. In *Information Geometry and Its Applications IV*, pages 75–115. Springer, 2016.
- [28] F. Ricci-Tersenghi. The bethe approximation for solving the inverse ising problem: a comparison with other inference methods. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(08):P08015, 2012.
- [29] D. Sherrington and S. Kirkpatrick. Solvable model of a spin-glass. *Physical review letters*, 35(26):1792, 1975.
- [30] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [31] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, chapter 6, pages 194–281. MIT Press, Cambridge, 1986.
- [32] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071, 2008.
- [33] G. Torlai and R. G. Melko. Learning thermodynamics with boltzmann machines. *Physical Review B*, 94(16):165134, 2016.
- [34] J. Tubiana, S. Cocco, and R. Monasson. Learning protein constitutive motifs from sequence data. *Elife*, 8:e39397, 2019.
- [35] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [36] U. Wolff. Collective monte carlo updating for spin systems. *Physical Review Letters*, 62(4):361, 1989.
- [37] B. Yelmen, A. Decelle, L. Ongaro, D. Marnetto, C. Tallec, F. Montinaro, C. Furtlehner, L. Pagani, and F. Jay. Creating artificial human genomes using generative neural networks. *PLoS genetics*, 17(2):e1009303, 2021.