Renormalization Group and Restricted Boltzmann Machines

Weilun Qiu
Department of MEAM
University of Pennsylvania
Philadelphia, PA 19104
wlqiu@seas.upenn.edu

May 2022

Abstract

Machine learning has achieved many successes in the recent years, especially in the field of information extraction. But it is still not fully clear why and how this methods work. This work tries to answer part of the problem. In this paper, we focus on one of the basics models in machine learning, the Restricted Boltzmann Machines (RBMs). Though it is not so popular these days compared to other models, it is simple enough for us to find connections between this method and physics, which may help us better understand RBMs. We introduce a crucial tool in theoretical physics, the renormalization group (RG), and build a mapping between RBMs and RG, and show that RBMs strictly obey the principles of RG. We also work out the 1d Ising model in detail to demonstrate this connection, and discuss how RBMs could be potentially useful in science and engineering applications.

Keywords Statistical physics · Renormalizaition group · Machine learning · Restricted Boltzmann Machines

1 Introduction

Machine learning has many successful applications in the recent years. One of the most important area of machine learning research is to extract important features from data. Despite the great success people have achieved, why machine learning works and how the information flows insides the architecture of the network are not fully clear. Machine learning is still somehow a black box method. Extracting important features is also an important task in physics, so it is quite natural to ask whether we can find connection between physics theory and machine learning method. In fact, some preliminary work has been done to explain machine learning methods by physical principles[1][2][9], which not only helps us better understanding machine learning, but also opens the new area of applying machine learning in science and engineering. So deepening the understanding of machine learning from the perspective of physics is of interest to people in both machine learning and science and engineering community.

In this work, we will focus on a model developed in the early days, namely, the RBMs, instead of working on the more popular models such as convolutional neural network (CNN), recurrent neural network (RNN) and transformer, because the architecture of RBMs is simpler and physics-based. We will try to understand RBMs and build the connection to RG, a very important technique in theoretical physics. RG is a tool for problems involving many scales. For example, in statistical physics, even though the interaction is between nearest neighbor, the system could still exhibit long-range behavior, which is not trivial and straight-forward to describe. The idea of RG is to integrate out the short range fluctuation, and obtain a direct description of the long-range interaction. In this way, we can extract the long-range physics from the original system. This can also be regarded as a coarse-graining process because after RG we can represent the physics of most interest with lest "degree of freedom".

RG procedure works very successfully on many problems and works for both discrete and continuous systems, such as the Ising model (which we will discuss in detail later), and the ϕ^4 theory in field theory. RG is a generic procedure rather than some specific calculation technique. Again, we limit ourselves to a simple case known as the Kadanoff's variaitional RG schemes[7] [8]. The key idea here is to introduce hidden spins, or coarse-grained spins, and the interaction energy (to be determined) between the original spins and the hidden spins. Then we could obtain the coarse-grained system by integrating out the hidden spins, and the interaction energy is determined by minimizing the difference between the free energy of the original system and the coarse-grained one. Lastly, we want point out that though RG is powerful, applying RG usually requires complicated mathematical technique and even intractable in some cases, so it is potentially useful to develop a "machine learning version" of RG.

We will build the mapping between RG and RBMs in this work. First we will introduce the idea of the RBMs, then we can immediately see that there are a lot of similarities. RBMs are stochastic models, and are usually used as generative models to model the distribution of the inputs, which are called visible variables in RBMs. There are also hidden variables in RBMs, and both visible and hidden variables are understood as random variables. RBMs model the interaction of the visible and hidden variables by an energy function, and the joint distribution is modelled as in a canonical ensemble[1]. RBMs are originally introduced to model binary random variables, but it can be generated to continuous cases without much difficulties. RBMs can also be stacked together to form the Deep Boltzmann Machines and Deep Belief Networks[12]. We can see from later sections that if we map the original spins in a statistical physics model to the visible models of RBMs, and the coarse-grained spins to the hidden variables of RBMs, then the free energy is automatically guaranteed to be invariant. If RBMs are able to reproduce the distribution of the inputs, the hidden variables can be viewed as the coarse-grained spins. It should be pointed out that in Kadanoff's variational RG scheme, the interaction energy is generally not unique. Instead, it is chosen based on physical intuition. So it is possible that the coarse-grained results produced by RBMs satisfy the variational RG scheme, but are different from the conventional treatment in physics. In fact, this means RBMs could be more flexible.

This work is basically following a former work by Mehta et al[10]. To the best of our knowledge, only preliminary work has been done on this topic. The papers by Iso et al.[6] and Funal et al.[4] also discuss this topic, but their focus on the information flows between the original data and the reconstructed data. Koch-Janusz et al.[9] construct a different machine learning algorithm, and the goal there is to maximize the mutual information between the coarse-grained spins and the surroundings.

We are interested in this topic because coarse graining is also important in mechanics. Some works have been done on application of RG in fluid mechanics[13][14]. However, there is not much RG application on solid mechanics. We are hoping to gain more insights of machine learning technique for RG, and hopefully, we can find application of the powerful RG method in solid mechanics research.

This work is organized as follows. In section 2 we introduce basic concepts and procedure in RG, and the application on one-dimensional Ising model is given in detail. In section 3 we introduce the architecture and training algorithm of RBMs. The mapping between RBMs and RG is developed in section 4, and several examples are given in section 5. This paper is finished by conclusions in section 6.

2 Renormalization Group

RG analysis is an important technique with many applications in statistical physics. For simplicity, we formulate the procedure on a binary spin system. Our formulation is basically following the paper by Mehta et al[10]. Consider a system consisting of N spins $\{v_i\}(i=1,\cdots,N)$ which take the values ± 1. Assume that a Hamiltonian $H(\{v_i\})$ is defined for a configuration (i.e., one realization of all the spins) of the system. Then we know from statistical physics that in equilibrium and in a canonical ensemble, the probability of a configuration is

$$P(\{v_i\}) = \frac{e^{-\beta H(\{v_i\})}}{Z} \tag{1}$$

where β is the inverse temperature (we set $\beta = 1$ for the rest of this paper), and $Z = \sum_{\{v_i\}} e^{-H(\{v_i\})} = \sum_{v_1,\cdots,v_N=\pm 1} e^{-H(\{v_i\})}$ (sum over all possible configurations) is the partition function, or the normalizing factor. Z is one of the most important function of the system, and the free energy is defined by

[1] A complete introduction of the ensemble theory can be found in [11]

$$F = -\log Z \tag{2}$$

Typically, the Hamiltonian is parametrized by some "material properties" $\{K\}$ of the system. For example, in the Ising model the Hamiltonian is given by

$$H(\{v_i\}) = -J \sum_{\langle ij \rangle} v_i v_j$$

where the sum is taken over all the nearest neighbors and there is only one parameter in the Hamiltonian $\{K\} = \{J\}$.

Now we try to introduce $M(M < N)$ new spins $\{h_j\}$ (called "hidden spins") to describe the system. This is motivated by two ideas. One is to coarse-grain the system so that we can represent the physics of the system with less degrees of freedom; the other is to "integrate" out the short range fluctuation so that we have a new theory which is formulated in term of the long-range interaction. (For example, recall that the Hamiltonian of the Ising model is given by the nearest neighbor iteraction.) Following the idea of Kadanoff's variational RG scheme, we introduce a function of "interaction energy" $T_\lambda(\{v_i\}, \{h_j\})$, which is parametrized by λ. Now imagine an auxiliary system consisting of $\{v_i\}$ and $\{h_j\}$, and the Hamiltonian will be $-T_\lambda(\{v_i\}, \{h_j\}) + H(\{v_i\})$ (neglecting the self-energy of the hidden spins). We can obtain the probability of a configuration of the hidden spins by integrating out the origianl spins $\{v_i\}$.

$$P_\lambda^{RG}(\{h_j\}) = \frac{e^{-H_\lambda^{RG}(\{h_j\})}}{Z_\lambda^{RG}} = \frac{\sum_{\{v_i\}} e^{T_\lambda(\{v_i\},\{h_j\}) - H(\{v_i\})}}{Z_\lambda^{RG}} \tag{3}$$

where Z_λ^{RG} is the new partition function (depending on λ). Further we have the free energy of the new system $F_\lambda^{RG} = -\log Z_\lambda^{RG}$. And we can define the optimal λ as the one that minimize the difference between the free energies $\|\Delta F\| = \|F_\lambda^{RG} - F\|$, because one can show that we can derive all properties of interest from the derivatives of the free energy. Notice that we can define the Hamiltonian of the hidden spins from such construction, and now it is parameterized by some new constants, say, $\{K^{RG}\}$. With the new Hamiltonian parameterized by the new constants, we can calculate the exact free energy just as before. Hence we say that the system flows from $\{K\}$ to $\{K^{RG}\}$ during the RG process.

The formulation above may seem abstract to those who are not familiar with statistical physics. Thus we will work out the one-dimensional Ising model in detail, which we will investigate numerically in the following sections. We assume there are 2^N spins in the system for convenience. The periodic boundary condition is adopted here and the Hamiltonian becomes

$$H(\{v_i\}) = -J \sum_{i=1}^{2^N} v_i v_{i+1}$$

Recall that we already set $\beta = 1$, then the partition function is given by

$$Z = \sum_{\{v_i\}} \exp\big\{J \sum_{i=1}^{2^N} v_i v_{i+1}\big\} = \sum_{v_1, \cdots, v_{2^N} = \pm 1} \exp\big\{J \sum_{i=1}^{2^N} v_i v_{i+1}\big\}$$

We are not going to construct $T_\lambda(\{v_i\}, \{h_j\})$ explicitly, but rather, we will integrate out half of the spins, which is more convenient. Separate the summation in the partition function into two parts, one summing over v_2, and another summing over the rest of the spins

$$Z = \sum_{v_1, v_3, \cdots, v_{2^N} = \pm 1} \sum_{v_2 = \pm 1} \exp\big\{J \sum_{i=1}^{2^N} v_i v_{i+1}\big\} = \sum_{v_1, v_3, \cdots, v_{2^N} = \pm 1} \exp\big\{J \sum_{i=1, i \neq 1,2}^{2^N} v_i v_{i+1}\big\} \sum_{v_2 = \pm 1} \exp\big\{J v_2 (v_1 + v_3)\big\}$$

Note that $\exp\{J v_2 (v_1 + v_3)\} = 2 \cosh J(v_1 + v_3)$. Let

$$A\exp\{J^{'}v_1v_3\} = 2\cosh J(v_1+v_3)$$

hold for arbitrary $v_1, v_3 \in \{\pm 1\}$, and we could solve for the constants.

$$A = 2\sqrt{\cosh 2J}, \quad J^{'} = \frac{1}{2}\ln(\cosh 2J) \tag{4}$$

Plug in these identities in the expression of the partition function

$$Z = \sum_{v_1, v_3, \cdots, v_{2^N} = \pm 1} \exp\big\{J \sum_{i=1, i\neq 1,2}^{2^N} v_i v_{i+1}\big\} \cdot A\exp\{J^{'}v_1v_3\}$$

Repeat this procedure for all even number spins, and we find out that

$$Z = A^{2^{N-1}} \sum_{v_1, v_3, \cdots, v_{2^N-1} = \pm 1} \exp\big\{J^{'} \sum_{i=1}^{2^{N-1}} v_{2i-1} v_{2i+1}\big\}$$

We successfully coarse grain the original system to a new system with only half of the original size, and with the correct partition function. The RG flow of the parameter is given by $J \to J^{'}$.

3 Restricted Boltzmann Machine

RBMs are energy-based models. For simplicity and consistency, we limit ourselves to RBMs for binary data. However, it should be pointed out that RBMs can be generated to continuous data without much difficulty. For more information on this topic, we refer to the review by Salakhutdinov[12]. RBMs are used to model probability distributions. Let $\{v_i\}$ (with some abuse of notation since we will show later that they are essentially the same as the $\{v_i\}$ defined in previous section) be binary data drawn from some distribution $P(\{v_i\})$. These are the random variables we are going to model, and we call them visible variables. RBMs also introduce hidden variables $\{h_j\}$ (again, with abuse of notation) that couple to visible variables. The interaction of the visible and hidden variables is modelled by the following energy function

$$E(\{v_i\}, \{h_j\}) = \sum_i b_j h_j + \sum_{i,j} v_i w_{ij} h_j + \sum_i c_i v_i \tag{5}$$

where $\lambda = \{b_j, w_{ij}, c_i\}$ are model parameters. Note that this energy function can be generalized to more complex form if necessary. Inspired by the principle of statistical physics, the joint probability of a configuration of the visible-hidden variables pair is

$$p_\lambda(\{v_i\}, \{h_j\}) = \frac{e^{-E(\{v_i\},\{h_j\})}}{\mathcal{Z}} \tag{6}$$

where $\mathcal{Z}$ is the normalizing factor (or partition function).

We can calculate the marginal distribution from the joint distribution. More detail about the derivation can be found in [12].

$$p_\lambda(\{v_i\}) = \sum_{\{h_j\}} p_\lambda(\{v_i\}, \{h_j\}) = \frac{1}{\mathcal{Z}} \exp\big\{\sum_i c_i v_i\big\} \cdot \prod_j \Big(1 + \exp\big\{b_j + \sum_i v_i w_{ij}\big\}\Big) \tag{7}$$

$$p_\lambda(\{h_j\}) = \sum_{\{v_i\}} p_\lambda(\{v_i\}, \{h_j\}) \tag{8}$$

Since there's no interaction between different visible variables (and between different hidden variables), the conditional probability distribution can be factorized.

$$p_\lambda(\{v_i\}|\{h_j\}) = \prod_i p_\lambda(v_i|\{h_j\}), \quad p_\lambda(\{h_j\}|\{v_i\}) = \prod_j p_\lambda(h_j|\{v_i\}) \tag{9}$$

The conditional distribution of one visible and hidden variable can also be derived analytically.

$$p_\lambda(v_i = 1|\{h_j\}) = \sigma\left(\sum_j w_{ij}h_j + c_i\right), \quad p_\lambda(h_j = 1|\{v_i\}) = \sigma\left(\sum_i v_i w_{ij} + b_j\right) \tag{10}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. Interestingly, we can see from the results above that the conditional probability is modelled by a one-layer neural network with the sigmoid activation.

Now we come to the training process of RBMs. The optiaml parameters of RBMs λ^* are those that miminize the KL-divergence between the modelled marginal distribution of $\{v_i\}$ and the true distribution $P(\{v_i\})$

$$\lambda^* = argmin_\lambda \mathbb{KL}(P(\{v_i\})||p_\lambda(\{v_i\})) = argmin_\lambda \sum_{\{v_i\}} P(\{v_i\}) \log P(\{v_i\}) - P(\{v_i\}) \log p_\lambda(\{v_i\}) \tag{11}$$

The first term is independent of λ, and the second term can be approximated by a Monte Carlo method estimate, which turns out to be the likelihood function.[3]

$$\mathcal{L} = \sum_{\{v_i\}} P(\{v_i\}) \log p_\lambda(\{v_i\}) \approx \frac{1}{N_{data}} \sum_{k=1}^{N_{data}} \log p_\lambda\left(\{v_i\}_k\right)$$

So we can also say that the optimal parameters can be found by maximizing $\mathcal{L}$. Note that RBMs can recover the exact distribution if the minimum of the KL-divergence is zero.

The training algorithm of gradient descent is not trivial because the evaluation of $\mathcal{L}$ involves the evaluation of the partition function $\mathcal{Z}$, of which the computational cost is roughly exponential in the number of variables of RBMs. A common strategy is to use sampling methods to estimate the gradient. In this work, we adopt the one-step contrast divergence (CD-1) algorithm. The basic idea is to use a one-step Gibbs sampling to approximate the gradient. The derivation of this algorithm could be tedious, so we refer to these papers[3] [5]. In practice, the update rule we use is given by (following [12])

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} \propto \mathbb{E}_{data}[v_i h_j] - \mathbb{E}_{model}[v_i h_j] \approx \sum_{k=1}^{N_{data}} \{v_i h_j\}_k - \sum_{k=1}^{N_{data}} \{\hat{v}_i h_j\}_k \tag{12}$$

where $\{h_j\}$ are sampled from $p_\lambda(\{h_j\}|\{v_i\})$, and $\{\hat{v}_i\}$ are reconstructed by sampling from $p_\lambda(\{v_i\}|\{h_j\})$ [2].

4 Mapping between RG and RBMs

We can see from the formulation of RBMs that it is inspired by statistical physics. As a result, it is not surprising that we can find a mapping between RG and RBMs. Recall from the previous section that the marginal distributions of RBMs are derived by integrating out other variables. To build the connection between RG and RBMs, we write the marginal distribution in another form.

$$p_\lambda(\{v_i\}) = \frac{e^{-H_\lambda(\{v_i\})}}{\mathcal{Z}}, \quad p_\lambda(\{h_j\}) = \frac{e^{-H_\lambda(\{h_j\})}}{\mathcal{Z}} \tag{13}$$

[2] For more explanation on the coding side, we refer to this blog http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines/

Here H_λ is the "effective" energy of the visible variables or the hidden variables. If we think of the visible variables as the original spin, and the hidden variables the new spins after RG process, we can see immediately from the above equation that the architecture of RBMs guarantees the partition functions of the original system and the new system obtain by RG method are strictly the same! Next we try to figure out the equation for $H_\lambda(\{h_j\})$

$$p_\lambda(\{h_j\}) = \frac{e^{-H_\lambda(\{h_j\})}}{\mathcal{Z}} = \sum_{\{v_i\}} \frac{e^{-E(\{v_i\},\{h_j\})}}{\mathcal{Z}} \tag{14}$$

Compare this with Equation 3, and we find out that the interaction energy function we are using is

$$T_\lambda(\{v_i\},\{h_j\}) = -E(\{v_i\},\{h_j\}) + H_\lambda(\{v_i\}) \tag{15}$$

If the architecture of RBMs is wide enough, one can show that the modelled distribution converges to to true distribution[10]. If that is the case, then we have $p_\lambda(\{v_i\}) = P(\{v_i\})$, and thus $p_\lambda(\{h_j\}) = P_\lambda^{RG}(\{h_j\})$, which means RBMs will obtain the exact RG flow. However, it should be pointed out that usually we want to coarse grain a system so the number of hidden variables is less than that of the visible variables, so usually we will not recover the exact distribution. But it is still potentially useful for us, because in many cases we cannot do the RG analysis exactly, and RBMs could become a machine-learning based numerical scheme for RG.

5 Examples

5.1 Example 0: RBMs validation

We use the MNIST data set to validate our code. RBMs are generative models, and they should be able to learn the distribution of the data, and further generate more samples. To keep things simple, we first train the RBMs model, then we sample the hidden variables from the conditional distribution $p_\lambda(\{h_j\}|\{v_i\})$, and reconstruct the visible variables by sampling from $p_\lambda(\{v_i\}|\{h_j\})$. Here $\{h_j\}$ are similar to the latent variables in autoencoders. We expect RBMs to generate similar handwritten digits.

We use 2000 images from the MNIST data set, 50% of which digit 0 and the rest digit 1. 80% of the data is used as training set, and the rest 20% is used as test set. We also change the grey scale image to binary in order to apply RBMs. The number of visible variables is the size of the image: 28×28, and we set the number of hidden variables to be 144. Two typical original-reconstructed image pairs are given in Figure 1. We define the ratio between the number of pixels that do not match the original data and the number of all pixels as the relative error. The error of reconstruction is 4.8%. RBMs are capable of learning the distribution of the input data.

5.2 Example 1: 1d Ising model

We use Metropolis algorithm to generate samples from the 1d Ising model. The probability of a certain configuration of the Ising model is proportional to e^{-H} if we set the inverse temperature $\beta = 1$ (where H is the Hamiltonian of the configuration). Thus the transition probability from state μ to ν is

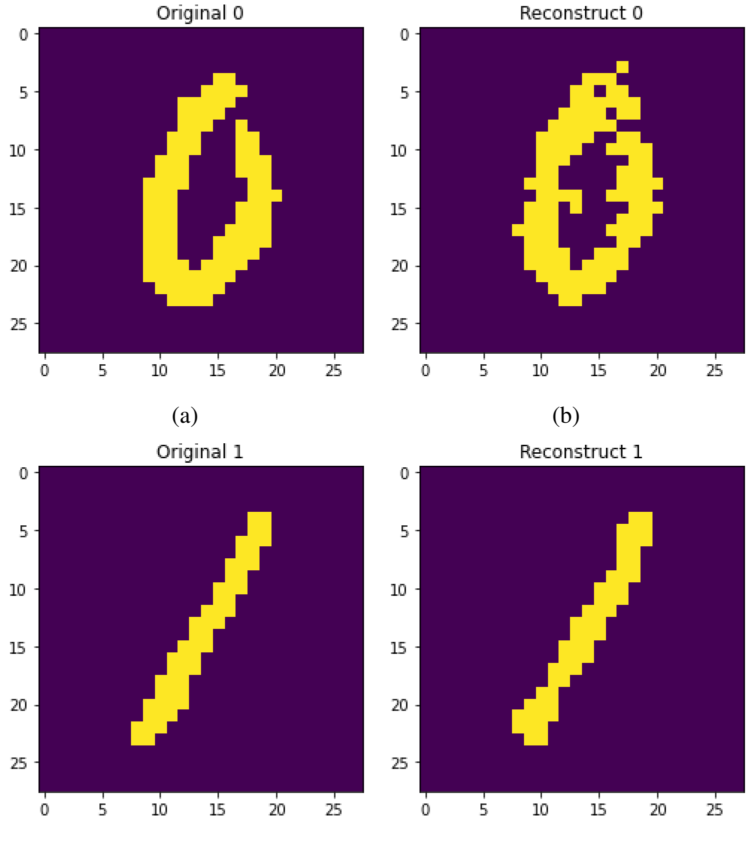
$$A(\mu \to \nu) = \begin{cases} 1 & \text{if } H_\nu < H_\mu \\ e^{-(H_\nu - H_\mu)} & \text{if } H_\nu > H_\mu \end{cases} \tag{16}$$

To make the computational process more efficient, we point out that in practice we don't need calculate the Hamiltonian of different states, but only the difference. If we only flip one spin at every step [3], then with periodic boundary condition, we have

$$\Delta H = 2Jv_i^\mu(v_{i-1}^\mu + v_{i+1}^\mu)$$

where the superscript μ stands for state μ. We generate $1,200$ samples of a system with 64 spins, where 800 samples are used as training set, and 400 samples are used as test set. Note that in Isng model, the convention is that every spin takes its value in $\{\pm 1\}$, but we assume the spin value is $\{0, 1\}$ when we formulate RBMs, so a

[3] For more discussion we refer to this blog: https://www.compphy.com/ising-model-1d-interacting-spins-in-absence-of-external-



(a) (b) (c) (d)

Figure 1: Typical original data and reconstructed samples on the MNIST data set

conversion is necessary when we apply RBMs on the Ising model data. As for the RBMs architecture, we use 64 visible variables and 32 hidden variables, so that we can make a direct comparison between RBMs and the RG decimation derived in section 2. We train the model for $170,000$ epochs, and the loss curve is shown in Figure 2. Recall that we mention above in principle, the coarse-grained results given by RBMs are not necessary to be the same as the standard RG results. To encourage the model to be consistent with the conventional "local coarse graining" (in section 2, we integrate out all the even spins), we add a L1 regularization to the loss function to impose sparsity of the interaction. Note that this is a soft constraint, so it is still possible that RBMs give us different results.

Just as in example 0, we compare the input and the reconstructed spins. Since RBMs are stochastic models and we have limited number of hidden variables, the reconstructed spins need not to be the same as the inputs. However, we can expect to see similar pattern in the reconstructed spins, which is indeed the case as shown in Figure 3. If we define the error in the same way as in example 0, we find out that the error is about 14% (depending on the reconstructed samples). This might not seem satisfactory, but note that the reconstructed samples come from a down-sample than up-sample process, certain information is loss. It is not possible to recover every detail of the original spins. It is acceptable as long as the patterns are quantitatively correct. Besides, we know from statistical physics that there is no spontaneous symmetry breaking in 1d Ising model (in plain words, when in equilibrium, the average value of the spins are zero). The numerical results of the original spins and the reconstructed spins are -0.025 and -0.0073, which shows that if the input system is in equilibrium, then so are the reconstructed spins.

Now we analyze the weights w_{ij} in the energy function. With L1 regularization in the loss function, we expect RBMs to recover the "blocking" coarse graining picture (i.e., every hidden spin is coupled to the local original spins) as in the analytical RG analysis. Indeed we observe such structures in the weights of RBMs, as shown in Figure 4. Note that it is possible that a hidden variable is coupled to multiple visible variables, because we do not build this constraint inside the architecture. If we assume that every hidden variables is basically determined by the visible variable of which the coupling weight has the greatest absolute value, then we can see that every hidden variable is determined by a

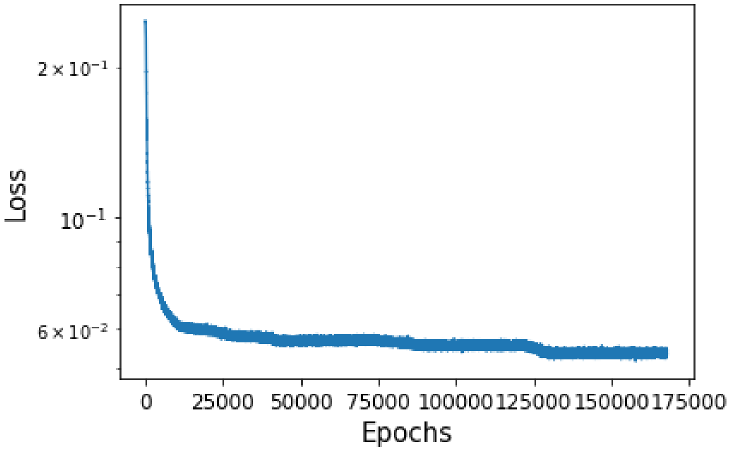
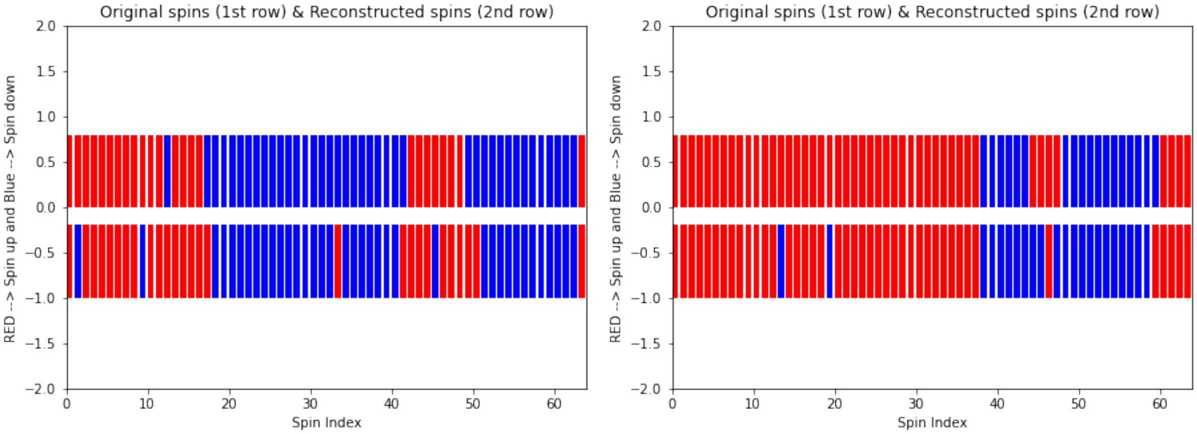


Figure 2: Loss curve of the training process



(a)

(b)

Figure 3: Typical original data and reconstructed sample on the 1d Ising model data

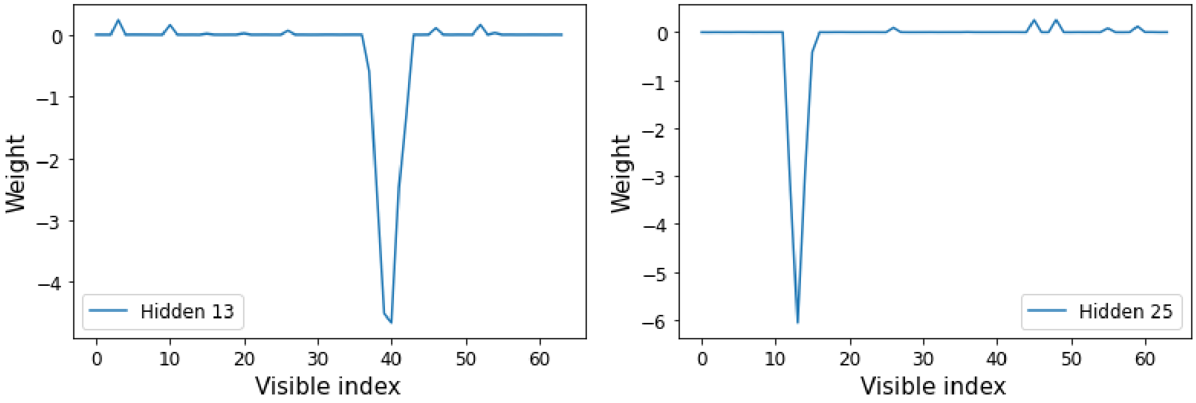
unique visible variable (with a few exception)[4]. This is discovered automatically by RBMs and consistent with physical intuition.

Recall that in Equation 4 we derive the analytical RG flow of the parameter J. We will compare this with the numerical result. Extracting J from some spins samples is not trivial, and we need more theory from statistical physics. Define the correlation of a spin system as $\langle v_i v_j \rangle$, where $\langle \cdot \rangle$ stands for expectation. Since the only parameters of 1d Ising model are inverse temperature β and coupling constant J, and we fix β to be identical, it is straight forward to see that $\langle v_i v_j \rangle$ is a function of J only. By Equation 4, we know that the analytical coupling constant after RG is

$$J^{'} = \tanh^{-1}(\tanh^2 1) \approx 0.6625$$

We simulate an 1d Ising system with 32 spins and coupling constant $J^{'}$, then we estimate the correlation function using this simulation, and compare with the correlation function obtained from the RBMs hidden variables. We would like to mention some technical detail here. Since we use periodic boundary condition here, it is natural to assume that the spin system is homogeneous, which implies that $\langle v_i v_j \rangle = \langle v_{i+k} v_{j+k} \rangle$ for arbitrary integer k. In practice we calculate $\langle v_0 v_x \rangle$ for several different x's. Besides, as we can see from Figure 4, the order of the hidden variables are random, so we need to sort them before calculating correlation function. To validate our simulation of the Ising model, we also compare the numerical results of the correlation function with theory. We can solve for the correlation function

[4] We demonstrate this in our code.



(a)

(b)

Figure 4: Typical weights of RBMs. (a) Weights coupled to hidden variable h_{13}; (b) Weights coupled to hidden variable h_{25};

analytically using a technique called transfer matrix method[5]. In general, the analytical result is complex and not easy to use, so we will use the result for infinite-many spins limit (also known as the thermodynamic limit in physics) as a reference.

$$\lim_{N\to\infty} \langle v_0 v_x \rangle = (\tanh J)^x$$

The results from RBMs hidden variables are quantitatively correct. There are two main reasons why it is not a perfect match. The first reason is that RBMs are stochastic, and we only have finite (400) samples; another reason is that there are multiple possible ways to do RG or coarse graining, we do not enforce the hidden variables in RBMs to be determined by two visible variables (just like what we do in analytical treatment), and as a result, though the weights in the energy function of RBMs are sparse, every hidden variable is still coupled to multiple visible variables. We argue that this actually makes RBMs more flexible, and we are able to discover other RG strategies. If we wish to re-discover the analytical RG result, we could hard code the constraints inside the architecture to make sure every hidden variable is determined by only two neighboring visible variables.

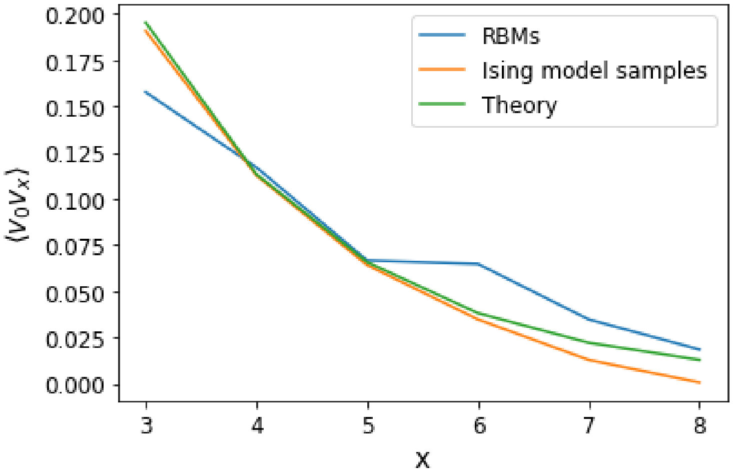


Figure 5: Correlation functions of the 1d Isng model and RBMs hidden variables

To sum up, we find a lot of resemblance between RG and RBMs, and we argue that RBMs are more flexible in some sense. Now the application is still limited to toy problems in statistical physics, because in more practical problem, it

[5] The mathematical derivation is tedious, so we skip this part. A good reference can be found fron this blog: https://stanford.edu/~jeffjar/statmech/lec4.html.

could be very difficult to define a proper Hamiltonian of a configuration. However, this discussion gives us insights on how to understand machine learning techniques from a physical viewpoint, and how to build machine learning architectures based on physical principles. There is one more interesting observation to mention. In statistical physics, people are not so interested in a configuration, because we always work with an ensemble, so a specific configuration never matters. For example, when we do RG on 1d Ising model, we can integrate out all the even spins, or all the odd spins, which is quite arbitrary. On the contrary, people in the mechanics community do care about configurations when they are studying coarse graining. And RBMs can give us samples from the coarse-grained system, so we think RBMs, or more generally, machine learning techniques for coarse graining, could be useful in mechanics.

6 Conclusions

In this paper, we introduce the theory of RG and RBMs, then build a mapping between them. This partly explain why RBMs work, and we can understand RBMs from a physical perspective. Then we apply RBMs on 1d Ising model and make a detailed analysis. The numerical results from RBMs and their physical meanings are examined and assessed. This study may not have direct applications in science and engineering, but it is of interest itself, and more importantly, this could shed light on how to build physics-based machine learning models and lead us to successful applications in the future. We were hoping to try out more examples in the beginning, such as the 2d Ising model and even some toy models in mechanics, but due to limited time and energy, we are not able to finish all these examples. Hopefully, we can combine the ideas from this work into our future research projects.

In theory, every mechanical problems can be solved using ensemble theories in statistical physics. However, in practice people tend to build phenomenalogical models instead of starting from first principles. To the best of the author's knowledge, one reason is that we do not have proper models for particles interaction and thus no proper Hamiltonian, and not to mention that calculating partition function is generally intractable or requiring many mathematical techniques. But machine learning models like RBMs, which follows the physical principles strictly, may help us tackle the latter problem.

References

[1] Alexander A Alemi and Ian Fischer. “TherML: Thermodynamics of machine learning”. In: *arXiv preprint arXiv:1807.04162* (2018).

[2] Giuseppe Carleo et al. “Machine learning and the physical sciences”. In: *Reviews of Modern Physics* 91.4 (2019), p. 045002.

[3] Asja Fischer and Christian Igel. “Training restricted Boltzmann machines: An introduction”. In: *Pattern Recognition* 47.1 (2014), pp. 25–39.

[4] Shotaro Shiba Funai and Dimitrios Giataganas. “Thermodynamics and feature extraction by machine learning”. In: *Physical Review Research* 2.3 (2020), p. 033415.

[5] Geoffrey E Hinton. “Training products of experts by minimizing contrastive divergence”. In: *Neural computation* 14.8 (2002), pp. 1771–1800.

[6] Satoshi Iso, Shotaro Shiba, and Sumito Yokoo. “Scale-invariant feature extraction of neural network and renormalization group flow”. In: *Physical review E* 97.5 (2018), p. 053304.

[7] Leo P Kadanoff. *Statistical physics: statics, dynamics and renormalization*. World Scientific, 2000.

[8] Leo P Kadanoff, Anthony Houghton, and Mehmet C Yalabik. “Variational approximations for renormalization group transformations”. In: *Journal of Statistical Physics* 14.2 (1976), pp. 171–203.

[9] Maciej Koch-Janusz and Zohar Ringel. “Mutual information, neural networks and the renormalization group”. In: *Nature Physics* 14.6 (2018), pp. 578–582.

[10] Pankaj Mehta and David J Schwab. “An exact mapping between the variational renormalization group and deep learning”. In: *arXiv preprint arXiv:1410.3831* (2014).

[11] Raj Kumar Pathria. *Statistical mechanics*. Elsevier, 2016.

[12] Ruslan Salakhutdinov. “Learning deep generative models”. In: *Annual Review of Statistics and Its Application* 2 (2015), pp. 361–385.

[13] Leslie M Smith and Stephen L Woodruff. “Renormalization-group analysis of turbulence”. In: *Annual review of fluid mechanics* 30.1 (1998), pp. 275–310.

[14] Victor Yakhot and Steven A Orszag. “Renormalization group analysis of turbulence. I. Basic theory”. In: *Journal of scientific computing* 1.1 (1986), pp. 3–51.