UNIVERSITY OF TRENTO

Bachelor's Degree in Physics

Bachelor's Degree Thesis

Emergent features: from renormalization group to artificial intelligence

Supervisor

Prof. Raffaello POTESTIO

Candidate

Guglielmo GRILLO

March 2020

“La Luna è, si potrebbe azzardare, un secondo pianeta, compagno del nostro. Gli altri pianeti del sistema solare si avvicinano e si allontanano da noi continuamente, viaggiando ognuno a una diversa velocità, ciascuno con la propria immane massa. Se non ci fosse la Luna, con la sua grande forza gravitazionale, l’asse del nostro pianeta oscillerebbe come quello di una trottola malferma. Così non avremmo un clima regolare, con temperature stabili, il ciclo delle stagioni e tutto il resto. Non esisterebbe la vita per come la conosciamo. Niente piante. Niente animali che si nutrono di piante. Niente animali che si nutrono di animali che si nutrono di piante. Invece a trecentottantamila chilometri da noi c’è la Luna, più vicina di ogni altro corpo celeste. Essa compensa queste variazioni, ne smorza l’effetto. Sì, Selene, la Luna dà equilibrio al mondo.”

R. Mercandini, Storia Perfetta dell’errore

Table of Contents

Acronyms

SLNN

Single Layer Neural Network

DL

Deep Learning

DNN

Deep Neural Network

RBM

Restricted Boltzmann Machine

BST

Block Spin Transformation

RG

Renormalization Group

VRG

Variational Renormalization Group

Summary

In recent years Machine Learning and Neural Networks have become a prominent tool with a wide range of applications ranging from product recommendations to face recognition. Despite its pervasiveness, the reasons underlying its efficiency are not completely understood. In this thesis a possible explanation based on the renormalization group theory will be presented. In particular, the link between Restricted Boltzmann Machines applied to the Ising Model and a renormalization procedure will be examined.

Chapter 1 will present some preliminary notions about Deep Learning, Restricted Boltzmann Machines, the Ising Model and the Renormalization group.

Chapter 2 will examine some efforts made in the direction of mapping the variational renormalization group to deep learning.

Chapter 1

Preliminary notions

1.1 Neural Network and Machine Learning

The term Machine Learning referees to a wide class of algorithms and procedures used to perform specific tasks without the need of explicitly coded instructions. Of all these methods Deep Neural Network have been successfully applied in multiple fields as a multipurpose black box. In this section a brief overview of what Deep Neural Networks are will be presented.

1.1.1 Single Layer Neural Network

In order to understand how a deep neural network works it is first necessary to understand how a single layer neural network (SLNN) works.

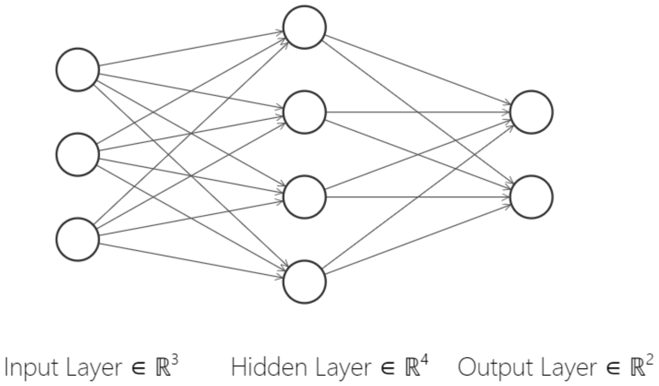

Input Layer ∈ $\mathbb{R}^3$
Hidden Layer ∈ $\mathbb{R}^4$
Output Layer ∈ $\mathbb{R}^2$

Figure 1.1: Single Layer Neural Network. The number of nodes in each layer is arbitrary.

Taking as reference *Figure* 1.1, a SLNN is made of an input layer, a hidden layer and an output layer. Information is processed in a feed-forward manner i.e. the information flows from left to right and connections between the nodes of a single layer are not allowed. Let $\boldsymbol{x} = \{x^k\}$ be the input vector, $\boldsymbol{W} = \{w^i_k\}$ the matrix associated with the weight of the connections, $\mathbf{b} = \{b^i\}$ the bias vector, σ a non linear activation function and $\boldsymbol{y} = \{y_i\}$ the value of the hidden layer. The purpose of the hidden layer is to perform the operation

$$y_i = \sigma\left(\sum_k W^i_k x^k + b^i\right)$$

The values of the output layer are generated via another affine transformation on the values of the hidden layer's neurons.
For convenience we will assume σ to be an element-wise function and incorporate

the bias vector into the the weight matrix and the input vector[1]. With this notation, a single neural network's behaviour can be written as

$$\boldsymbol{y_{out}} = \boldsymbol{W_{out}}\ \sigma(\boldsymbol{W_{hidden}}\boldsymbol{x_{input}})$$

The value of the weight and biases is not predetermined but is inferred from the training data with a training algorithm, usually back-propagation. The explanation of these algorithms is beyond the scope of this thesis but a brief introduction can be found in [1].

The universal approximation theorem states that, given enough neurons in the hidden layer, a SLNN can approximate arbitrarily well any given function. The mathematical results can be found in [1]; the general idea is that any given function admits a basis expansion in terms of a neural network.

1.1.2 Deep Learning

The term *deep learning* covers a wide range of networks with multiple layers of representations. With the increasing computational power given by modern GPUs, this architecture managed to achieve remarkable results in many fields spacing from object recognition to physics simulation. Although a broad range of architectures is available (see [1] for a complete overview) they are all variations and improvements on the basic Deep Neural Network (DNN) (*Figure* 1.2).

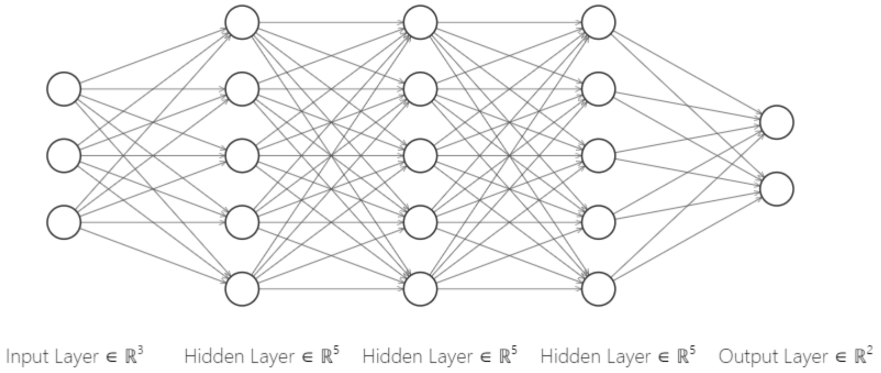


Figure 1.2: Deep Neural Network made of stacked SLNN.

A DNN is composed of many SLNN stacked one on top of the other so that the output of one becomes the input of the following. Due to their similarity with

[1] This can be easily be done by adding a new entry to the input vector with value 1 and a new column to the weight matrix representing the biases.

SLNN the basic theory follows directly from that of SLNN. DNN can be seen as an iterative composition of non linear functions and affine transformations:

$$\boldsymbol{y_{out}} = \boldsymbol{W^{out}}\ \sigma(\boldsymbol{W^{Hn}}\sigma(\boldsymbol{W^{H(n-1)}}\sigma(...\boldsymbol{W^{H2}}\sigma(\boldsymbol{W^{H1}}\boldsymbol{x_{input}})...)))$$

DNNs manifest an improvement in performance due to the composition of multiple layers which allows to reduce the number of neurons needed for each layer. On the other hand, the presence of multiple hidden layers makes it even harder to understand what is actually happening in each layer. An other open question is why DNNs perform better than SLNNs with a significant reduction in the number of neurons. A promising path is the one suggested by renormalization theory (*Section 1.3*) as examined in [2]. This hypothesis assumes that each layer of a DNN performs a single step of the renormalization procedure in order to extract key information from the input. This link will be further investigated in the following chapter.

1.1.3 Restricted Boltzmann Machine

Restricted Boltzmann Machines[2] are artificial neural networks composed of a visible layer and a hidden layer (*Figure* 1.3). Both layers can only assume binary value.

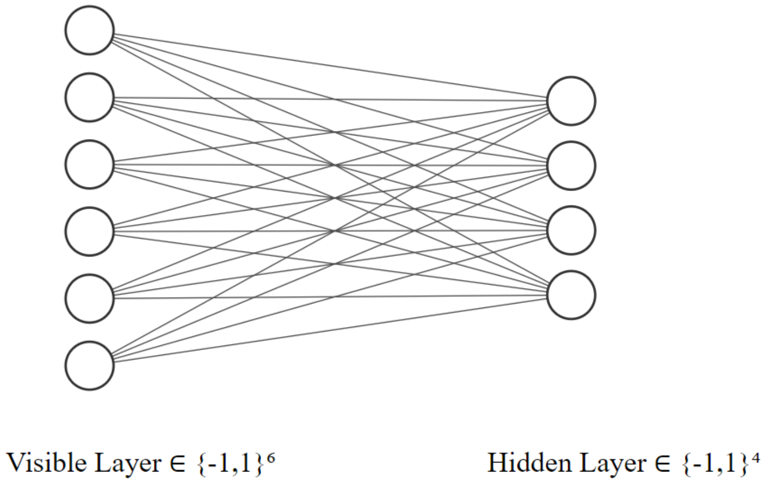


Figure 1.3: Restricted Boltzmann machine architecture

Thus, letting $\boldsymbol{v} = v_i$ be the visible layer and $\boldsymbol{h} = h_j$ the hidden layer, it is requested $v_i, h_j \in \{-1, 1\}$. The layers are connected via a weight matrix $\boldsymbol{W} = w_{ij}$ and two

[2] The adjective restricted refers to the fact that no intralayer connections are allowed, as opposed to Boltzmann Machine were nodes within a layer can be connected.

sets of biases, one related to the visible layer, call it $\boldsymbol{a} = \{a_i\}$, and one related to the hidden layer, $\boldsymbol{b} = \{b_j\}$. The activation function is restricted to have codomain $[0, 1]$. The most popular choices are a restriction of the hyperbolic tangent and the softmax function[3] for its close relation with the Boltzmann distribution.

Given a pre-trained RBM and an input vector $\boldsymbol{v}$, the working mechanism of a RBM is the following:

- The input is processed as in a standard SLNN
$$\boldsymbol{x} = \sigma(\boldsymbol{W}\boldsymbol{v} + \boldsymbol{b})$$
- The hidden vector $\boldsymbol{h}$ is constructed by setting his h_j value to 1 with probability x_j.
- The vector then undergoes a backward passage flowing from the hidden layer back to the visible layer
$$\boldsymbol{y} = \sigma(\boldsymbol{W}^T\boldsymbol{h} + \boldsymbol{a})$$
- A new visible vector $\boldsymbol{v}'$ is then generated with entries set to 1 with probability y_i.

This process allows to both extract the key features in the hidden layer and reconstruct an equivalent input vector.

In order to characterise the probability distribution it is possible to define a Hamiltonian for the RBM:

$$E = -\sum_j b_j h_j - \sum_{i,j} v_i W_{ij} h_j - \sum_i v_i a_i$$

which leads to the distribution:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E}$$

where $Z = \sum_{\{\boldsymbol{v},\boldsymbol{h}\}} e^{-E}$ is the partition function. The associated marginal distributions are

$$p(\boldsymbol{v}) = \frac{1}{Z} \sum_{\{\boldsymbol{h}\}} e^{-E} \qquad p(\boldsymbol{h}) = \frac{1}{Z} \sum_{\{\boldsymbol{v}\}} e^{-E}$$

[3] The softmax function is defined as

$$\text{softmax}(x_i, \boldsymbol{x}) = \frac{exp(x_i)}{\sum_j exp(x_j)}$$

The weights and biases are determined minimising the difference between the true distribution $q(\boldsymbol{v})$, provided by the training data, and the model distribution $p(\boldsymbol{v})$ of the RBM. The measure of distribution discrepancy is given by the Kullback-Liebler divergence:

$$D_{KL}(q||p) = \sum_i q(v_i)\Big[\log(q(v_i)) - \log(p(v_i))\Big]$$

The training algorithm is called Contrastive Divergence and consists in the following steps:

- Given a training sample $\boldsymbol{v}$ compute the corresponding compressed representation $\boldsymbol{h}$.
- Reconstruct the original vector with a backward passage and then re-sample a compressed version. Let this two vectors be $\boldsymbol{v}'$ and $\boldsymbol{h}'$.
- Compute the discrepancy between the original vectors and the re-sampled versions: $\boldsymbol{h}\boldsymbol{v}^T - \boldsymbol{h}'\boldsymbol{v}'^T$
- Update the weights according to this gradient, eventually mitigating the effect with a learning rate $\epsilon < 1$:

  $$\Delta W = \epsilon(\boldsymbol{h}\boldsymbol{v}^T - \boldsymbol{h}'\boldsymbol{v}'^T)$$

- As biases are strictly related to one of the layers the correction depends only on that specific layer:

  $$\Delta a = \epsilon(\boldsymbol{v} - \boldsymbol{v}') \qquad \Delta b = \epsilon(\boldsymbol{h} - \boldsymbol{h}')$$

This process is iterated until an acceptable error is reached.

1.1.4 Final assumption on input and output

In this thesis we will assume all the input and output vectors to be binary, $x_i = \{-1, 1\}$. This will allow us to map the input to 1D and 2D Ising Model, a model for ferromagnetism with a vast literature (*Section 1.2*), and the layers to RBMs. Furthermore this assumption is not far from practical application as any black and white image can be described by a two dimensional matrix with binary entries.

1.2 Ising Model 1D and 2D

The Ising model was developed to describe ferromagnetism but acquired greater importance as a reference model in the study of phase transition due to his simplicity and rich phenomenology.

Furthermore the Ising Model belongs to the same universality class[4] of many relevant physical systems. Only one-dimensional and two-dimensional Ising Model will be discussed. The discussion attempts to make a general overview and only relevant features are examined. These models are relevant as a lot of problems can be mapped to this scheme.

1.2.1 One-dimensional Ising Model

The one-dimensional Ising model consists of a chain of N spins. Despite not being mandatory, periodic boundary conditions are required in this thesis.

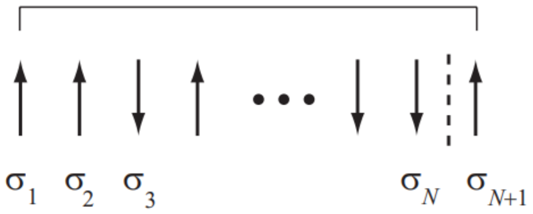


Figure 1.4: One-dimensional Ising subject to periodic boundary conditions.

The general Hamiltonian of the one-dimensional Ising model is:

$$H = -J\sum_{i=1}^{N}\sigma_i\sigma_{i+1} - h\sum_{i=1}^{N}\sigma_i$$

where $\sigma = \pm 1$ are the possible values of a single spin, J is the interaction strength between adjacent spins and h resembles the interaction with an external magnetic field. The average total magnetisation per spin is defined as:

$$m = \frac{1}{N^2}\sum_{i=1}^{N}\sigma_i$$

[4] Two systems are said to be in the same universality class if they have the same number of spatial dimensions and their order parameters (i.e. observables that enable to distinguish between phase) have the same number of dimensions.

and it is of fundamental importance due to being an order parameter for the model. As periodic boundary conditions were chosen it is possible to rewrite the Hamiltonian in the following way:

$$H = -J\sum_{i=1}^{N}\sigma_i\sigma_{i+1} - \frac{h}{2}\sum_{i=1}^{N}(\sigma_i + \sigma_{i+1})$$

which leads to the partition function:

$$Z(N,h,T) = \sum_{\sigma_1=\pm1}...\sum_{\sigma_N=\pm1}\exp\Big[\beta J\sum_{i=1}^{N}\sigma_i\sigma_{i+1} + \frac{\beta h}{2}\sum_{i=1}^{N}(\sigma_i + \sigma_{i+1})\Big]$$

The Transfer Matrix Method allows us to find an analytical solution. This method requires to write the partition function in terms of a matrix P whose elements are:

$$\langle\sigma|P|\sigma'\rangle = \exp\Big(\beta J\sigma\sigma' + \beta h(\sigma + \sigma')/2\Big)$$

Recalling that the possible values of the spins were set to be $\sigma = \pm1$, the matrix P can be explicitly written as:

$$P = \begin{pmatrix} e^{\beta(J+h)} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta(J-h)} \end{pmatrix}$$

The partition function can then be rewritten as:

$$\begin{aligned} Z(N,h,T) &= \sum_{\sigma_1=\pm1}...\sum_{\sigma_N=\pm1}\langle\sigma_1|P|\sigma_2\rangle\langle\sigma_2|P|\sigma_3\rangle...\langle\sigma_{N-1}|P|\sigma_N\rangle\langle\sigma_N|P|\sigma_1\rangle \\ &= \sum_{\sigma_1=\pm1}\langle\sigma_1|P^N|\sigma_1\rangle = Tr(P^N) \end{aligned}$$

where it was taken advantage of the completeness of the spin eigenvectors. The eigenvalues of the matrix are

$$\lambda_\pm = e^{\beta J}\Big[\cosh(\beta h) \pm \sqrt{\sinh^2(\beta h) + e^{-4\beta J}}\Big]$$

In the thermodynamic limit ($N \to \infty \Rightarrow \lambda_+^N >> \lambda_-^N$) the smallest eigenvalue can be neglected.

$$Z(N,h,T) = \lambda_+^N + \lambda_-^N \simeq \lambda_+^N = e^{N\beta J}\Big[\cosh(\beta h) + \sqrt{\sinh^2(\beta h) + e^{-4\beta J}}\Big]^N$$

All relevant thermodynamic properties can be evaluated from the free energy per spin

$$g(h,T,N) = -\frac{kT}{N}\ln\Big(Z(N,h,T)\Big) = -J - kT\ln\left(\cosh(\beta h) + \sqrt{\sinh^2(\beta h) + e^{-4\beta J}}\right)$$

The order parameter, the magnetisation, can be evaluated as

$$m = -\frac{\partial g}{\partial h} = \frac{\sinh(\beta h) + \sinh(\beta h)\cosh(\beta h)/\sqrt{\sinh^2(\beta h) + e^{-4\beta h}}}{\cosh(\beta h) + \sqrt{\sinh^2(\beta h) + e^{-4\beta h}}}$$

As $h \to 0$ the magnetisation vanishes at any given temperature, thus the nonexistence of a critical point. However in the limit of $\beta \to \infty$ ($T \to 0$) the magnetisation tends toward $m = \pm 1$ according to the sign of h.

1.2.2 Two-dimensional Ising Model

The two-dimensional Ising model consists in a two dimensional lattice of N sites whose spin values can only be $\sigma = \pm 1$. We require a square lattice ($N = n \times n$)

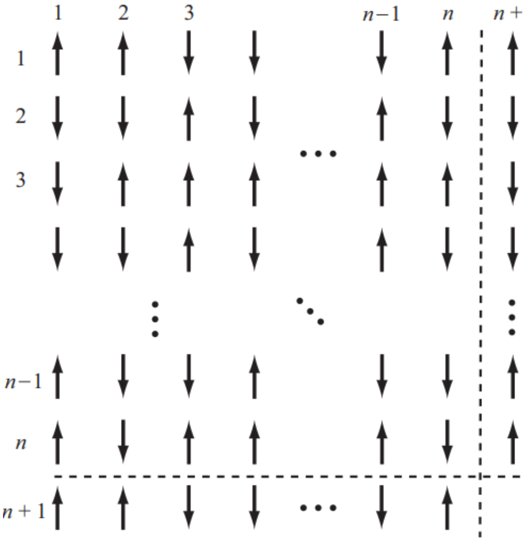


Figure 1.5: Two-dimensional Ising model subject to periodic boundary conditions.

with periodic boundary condition and restrict our discussion to nearest-neighbour interaction only. Therefore the Hamiltonian can be rewritten as

$$H = -J\sum_{i=1}^{N}\sum_{j=1}^{N}[\sigma_{i,j}\sigma_{i+1,j} + \sigma_{i,j}\sigma_{i,j+1}] - h\sum_{i=1}^{N}\sigma_{i,j}$$

The choice of periodic boundary conditions allows to write the partition function as

$$Z(N,h,T) = \sum_{\sigma_{1,1}=\pm 1} \dots \sum_{\sigma_{n,n}=\pm 1} \exp\Big[\beta J\sum_{i,j=1}^{N}[\sigma_{i,j}\sigma_{i+1,j} + \sigma_{i,j}\sigma_{i,j+1}] + \beta h\sum_{i=1}^{N}\sigma_{i,j})\Big]$$

The solution can be found using the Transfer Matrix approach. First we need to define a column vector of spins: $\mu_j = \{\sigma_{1,j}, \sigma_{2,j}, ..., \sigma_{n,j}\}$. Then we introduce two new functions:

$$E(\mu_j, \mu_k) = J \sum_{i=1}^{n} \sigma_{i,j}\sigma_{i,k}$$

$$\epsilon(\mu_j) = J \sum_{i=1}^{n} \sigma_{i,j}\sigma_{i+1,j} + h \sum_{i,j} \sigma_{i,j}$$

With this notation it is possible to rewrite the Hamiltonian and the partition function as:

$$H = -\sum_{j=1}^{n} \Big[E(\mu_j, \mu_{j+1}) + \epsilon(\mu_j) \Big]$$

$$Z(N,h,T) = \sum_{\sigma_{1,1}=\pm 1} ... \sum_{\sigma_{n,n}=\pm 1} \exp\Big[\beta E(\mu, \mu') + \frac{\beta}{2}(\epsilon(\mu) + \epsilon(\mu')) \Big]$$

The $2^n \times 2^n$ transfer matrix P can be defined as

$$\langle \mu | P | \mu' \rangle = \exp\Big[\beta E(\mu, \mu') + \frac{\beta}{2}(\epsilon(\mu) + \epsilon(\mu')) \Big]$$

The partition function then becomes

$$Z(N,h,T) = Tr(P^n) = \lambda_1^n + \lambda_2^n + ... + \lambda_{2^n}^n$$

In the thermodynamic limit only the largest eigenvalue becomes relevant. In this limit the critical temperature is given by the expression[5]:

$$2tanh^2(2J\beta) = 1 \rightarrow kT_c \simeq 2.269185J$$

The magnetisation is finally given by:

$$\begin{cases} 0 & T > T_c \\ \{1 - [\sinh(2J\beta)]^{-4}\}^{1/8} & T < T_c \end{cases}$$

[5] This and the following results are reported from [3]

1.3 Renormalization Group

The Renormalization Group (RG) is a method used to extract key features from a physical system. As in all coarse graining methods[6] there is no general method applicable to all physical systems. Nevertheless it is still possible to discuss the main points shared by any renormalization procedure. As in the rest of the thesis we will consider a lattice of spins whose value can only be $\sigma_i = \pm 1$.

The Hamiltonian of an Ising system with N spins, where many body interactions are taken in account, is:

$$H(\{\sigma_i\}) = k + \sum_{i}^{N} k_i \sigma_i + \sum_{i \leq j}^{N} k_{ij} \sigma_i \sigma_j + \sum_{i \leq j \leq k}^{N} k_{ijk} \sigma_i \sigma_j \sigma_k \ ...$$

where σ are spin values and $k_i, k_{ij}, k_{ijk},$ ... parameters that describe the interactions. The probability of a spin configuration and the partition function are defined by absorbing the factor $\beta = \frac{1}{k_b T}$ inside the Hamiltonian parameters[7]:

$$P(\{\sigma_i\}) = \frac{e^{-H(\{\sigma_i\})}}{Z}$$

$$Z = \sum_{\sigma_1, \ldots, \sigma_N = \pm 1} e^{-H(\{\sigma_i\})} = Tr_{\sigma_i} e^{-H(\{\sigma_i\})}$$

Performing a step in the renormalization procedure means finding a map T:

$$\begin{aligned} T(\sigma_i, \sigma'_j) : \quad & \{-1, 1\} \rightarrow \{-1, 1\} \\ & \qquad \sigma_i \longrightarrow \sigma'_j \end{aligned}$$

such that all the relevant statistical information is preserved. This can be done by requiring that the new Hamiltonian has the same functional form of the original:

$$H^{RG}(\{\sigma'_i\}) = k' + \sum_{i}^{N} k'_i \sigma'_i + \sum_{i \leq j}^{N} k'_{ij} \sigma'_i \sigma'_j + \sum_{i \leq j \leq k}^{N} k'_{ijk} \sigma'_i \sigma'_j \sigma'_k \ ...$$

It is mandatory to highlight that, despite being linked to the σ_i via the map $T(\sigma_i, \sigma'_j)$, the new variables σ'_j are independent from the old ones. The two probability distributions are then connected by the relation:

$$e^{-H^{RG}(\{\sigma'_j\})} = Tr_{\sigma} e^{T(\sigma_i, \sigma'_j) - H(\{\sigma_i\})} = Tr_{\sigma} \left[e^{T(\sigma_i, \sigma'_j)} \right] e^{-H(\{\sigma_i\})}$$

[6] A coarse grained model aims at simulating the behaviour of complex systems using a simplified representation which still holds all the key statistical information.

[7] This choice does not imply any loss of generality. It is still possible to take account of temperature by choosing one of the parameters as a reference for the temperature and changing the others according to it.

i.e. we required that the old partition function expressed in the new variables must be equal to the new partition function. Furthermore for an exact RG transformation the following relation holds:

$$Tr_{\sigma}\Big[e^{T(\sigma_i,\sigma'_j)}\Big] = 1$$

leading to

$$Tr_{\sigma'}\Big[e^{-H^{RG}(\{\sigma'\})}\Big] = Tr_{\sigma}\Big[e^{-H(\{\sigma\})}\Big]$$

So that the two partition functions are equal.

Variational RG: when the RG transformation can not be carried out exactly it is possible to resort to variational RG. This method uses a variational mapping $T_{\{\lambda_i\}}(\sigma_i, \sigma'_j)$ which depends on a set of parameters $\{\lambda_i\}$. The values of those parameters are set in order to minimise the discrepancy between the free energy of the original system and the free energy of the new system:

$$\min_{\lambda} \Delta F = \min_{\lambda}(F^{\sigma'}_{\lambda} - F^{\sigma})$$

Where the free energy have the standard definition:

$$F^{\sigma} = -\log(Z) = -\log(Tr_{\sigma_i} e^{-H(\{\sigma_i\})})$$

$$F^{\sigma'}_{\lambda} = -\log(Tr_{\sigma'_j} e^{-H^{RG}_{\lambda}(\{\sigma'_j\})})$$

1.3.1 Scaling Variable and RG Operators

Let $\boldsymbol{K}$ be a vector containing all the parameters of the Hamiltonian. Since the RG transformation must preserve the functional form of the Hamiltonian, it is possible to write its action in terms of a function R which acts only on $\boldsymbol{K}$:

$$\begin{aligned} T: \quad H_k(y) \longrightarrow H'(y') &= H_{K'}(y') \\ \text{with } \boldsymbol{K'} &= R(\boldsymbol{K}) \end{aligned}$$

Near a critical point a system will exhibit long-range correlation and self similarity at any scale reference. This implies that any statistical property must be invariant under length scaling and that the Hamiltonian will be invariant under a RG transformation. The equation $\boldsymbol{K'} = R(\boldsymbol{K})$ can then be linearized near the critical point $\boldsymbol{K}^*$:

$$K'_a \simeq K^*_a + \sum_b T_{ab}(K_b - K^*_b)$$

with

$$\left.\frac{\partial R_a}{\partial K_b}\right|_{K=K^*}$$

A left eigenvalue is then defined as

$$\sum_a \phi_a^i T_{ab} = \lambda^i \phi_b^i$$

and a scaling variable as

$$u_i = \sum_a \phi_a^i (K_a - K_a^*)$$

The term *scaling variable* comes from the following observation:

$$u_i' = \sum_a \phi_a^i (K_a') = \sum_a \sum_b \phi_a^i T_{ab}(K_b - K_b^*) = \sum_b \lambda^i \phi_b^i (K_b - K_b^*) = \lambda^i u_i$$

We assume that all the left eigenvalues λ^i are real and positive. Rewriting them as $\lambda^i = b^{y_i} \to y_i = \log_b(\lambda^i) = \ln(\lambda^i)/\ln(b)$ for an opportune y_i. The following relation

$$u_i' = b^{y_i} u_i$$

leads to three different cases according to the value of y_i:

- **RELEVANT** eigenvalues, $y_i > 0$. Repeated iteration of the RG transformation drives the variable away from the critical point and towards infinity.

- **IRRELEVANT** eigenvalues, $y_i < 0$. Repeated iteration of the RG transformation drives the variable away from the fixed point and toward 0.

- **MARGINAL** eigenvalues, $y_i = 0$. Repeated iteration of the RG transformation leaves the parameter value unchanged. The procedure doesn't allow to determine whether u_i will iterate towards or away from the fixed point.

After many repetitions of the RG procedures irrelevant and marginal eigenvalues can be neglected in comparison to relevant eigenvalues. Parameters associated with relevant eigenvalues are the ones that manifest a macroscopic behaviour.

Consider a two dimensional Ising model. It can be shown that both $T = 0$ ($\beta \to \infty$) and $T \to \infty$ ($\beta = 0$) are stable fixed point, i.e. any small shift away from the point tends to bring the point back to its original position. As both points are stable there must exist at least one intermediate unstable fixed point that divides the two regions. That point is the critical point. Iterated renormalization steps push the system away from the fixed point. This motion is called RG flow. *Figure* 1.6 provides a schematic representation of the process.

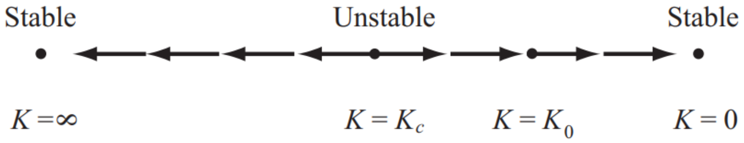


Figure 1.6: Schematic representation of the RG flow. The point K_0 start slightly on the right of $K_{critical}$ and flow to the right with every step.

1.3.2 Block Spin Transformation

One of the simplest yet most powerful methods is the one developed by Kadanoff and called Block Spin Transformation (BST)[8]. The procedure consists in the following steps:

1. Split the lattice in blocks of size $b \times ... \times b$ (according to the number of dimensions considered), eventually resorting to boundary conditions.

2. Replace each block with the mode of the spins i.e. the spin value with the majority of occurrence inside a particular block.

3. Scale the distance between spins by a factor of b.

The process is shown in *Figure* 1.7.

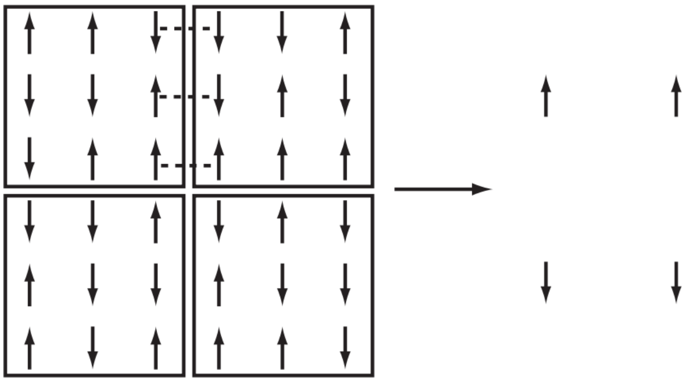


Figure 1.7: BST on a two dimensional lattice with $b = 3$. Dashed line refers to the interface interaction between blocks.

[8] An analogous method exists in deep learning literature under the name of *maxpool*. For details check the chapter *Convolutional Neural Network* in [1].

Recalling the renormalization equation $\boldsymbol{K'} = R(\boldsymbol{K})$, this relationship can be approximated at low temperature as

$$K' \simeq b^{d-1} K$$

where d represents the number of dimensions and b is called the length scaling factor. The exponent $d-1$ should not surprise as interactions between blocks are mediated by hyperplanes. In the two-dimensional Ising model the interactions is mediated by the sides of the square as shown in *Figure* 1.7. As b is greater than 1 the behaviour depends only on the exponent $y = d - 1$.

Chapter 2

A possible link between deep learning and the renormalization group

2.1 Annotation on the mapping between Variational RG and Deep Learning

This section refers to the article "*An exact mapping between variational renormalization group and deep learning*". Further reference can be found in [4]. In this article it is claimed that an exact mapping between the RG and DL exists. Here we report the main points and a counter-argument found in [2, 5].

2.1.1 The Mapping

Given an Ising lattice the probability of a configuration has the form of a Boltzmann factor[1]:

$$p(\{\sigma_i\}) = \frac{e^{-H(\{\sigma_i\})}}{Z}$$

If a variational renormalization transformation is performed, the probability distribution of the new spins and its relation with the original Hamiltonian is given by

$$p(\{\sigma'_j\}) = \frac{e^{-H^{RG}_{\lambda}(\{\sigma'_j\})}}{Z} = Tr_{\sigma_i}\Big[\frac{e^{-H(\{\sigma_i\})+T_{\lambda}(\{\sigma_i,\sigma'_j\})}}{Z}\Big]$$

Furthermore we require that the operator $T_{\lambda}(\{\sigma_i, \sigma'_j\})$ is defined in such a way that the old distribution can be reconstructed:

$$Tr_{\sigma'_j}\Big[\frac{e^{-H(\{\sigma_i\})+T_{\lambda}(\{\sigma_i,\sigma'_j\})}}{Z}\Big] = \frac{e^{-H(\{\sigma_i\})}}{Z}$$

Now, consider a RBM with an energy function defined as:

$$E(\{v_i, h_j\}) = \sum_j b_j h_j + \sum_{i,j} v_i W_{ij} h_j + \sum_i a_i v_i$$

the probability of obtaining a given configuration is:

$$p_W(\{v_i, h_j\}) = \frac{e^{-E(\{v_i,h_j\})}}{Z}$$

where W refers to both weight an biases. The marginal distributions can be obtained by summing over all possible configurations $\{v_i\}$ or $\{h_j\}$. Namely:

$$p_W(\{v_i\}) = \sum_{\{h_j\}} p_W(\{v_i, h_j\}) = Tr_{h_j}\Big[\frac{e^{-E(\{v_i,h_j\})}}{Z}\Big]$$

[1] As usual $\beta = \frac{1}{k_b T}$ is included in the Hamiltonian's parameters

If we assume that the RBM can, indeed, infer the probability distribution from the data and that the variables of the visible layer are a good representation of the spins in the data we have

$$\begin{cases} p_W(\{v_i\}) \simeq p(\{\sigma_i\}) \\ \quad \{\sigma_i\} \simeq \{v_i\} \end{cases} \qquad \Longrightarrow \qquad Tr_{h_j}\Big[\frac{e^{-E(\{v_i,h_j\})}}{Z}\Big] = Tr_{\sigma'_j}\Big[\frac{e^{-H(\{\sigma_i\})+T_\lambda(\{\sigma_i,\sigma'_j\})}}{Z}\Big]$$

This relation suggests the identity

$$-E(\{v_i,h_j\}) = -H(\{\sigma_i\}) + T_\lambda(\{\sigma_i,\sigma'_j\})$$

Substituting this equality into the relation for the RG Hamiltonian:

$$\begin{aligned} p_\lambda(\{\sigma'_j\}) = \frac{e^{-H^{RG}_\lambda(\{\sigma'_j\})}}{Z} &= Tr_{\sigma_i}\Big[\frac{e^{-H(\{\sigma_i\})+T_\lambda(\{\sigma_i,\sigma'_j\})}}{Z}\Big] \\ &= Tr_{v_i}\Big[\frac{e^{-E(\{v_i,h_j\})}}{Z}\Big] = p_W(\{h_j\}) \end{aligned}$$

This last equality implies that the probability distribution in the hidden layer matches the one generated by the renormalization procedure, hence the map between RBM and RG.

2.1.2 Annotation

The weak point of the previous argument is the deduction

$$\begin{cases} p_W(\{v_i\}) \simeq p(\{\sigma_i\}) \\ \quad \{\sigma_i\} \simeq \{v_i\} \end{cases} \qquad \Longrightarrow \qquad Tr_{h_j}\Big[\frac{e^{-E(\{v_i,h_j\})}}{Z}\Big] = Tr_{\sigma'_j}\Big[\frac{e^{-H(\{\sigma_i\})+T_\lambda(\{\sigma_i,\sigma'_j\})}}{Z}\Big]$$

This relation does not work both ways as it is possible to find different distributions still leading to the same partition function. The following argument is given in [2]. Let $p(\{v_i\})$ and $\hat{p}(\{v_i\},\{h_j\})$ be two different probability distributions:

$$p(\{v_i\}) = \frac{e^{-H(\{v_i\})}}{Z} \qquad\qquad \hat{p}(\{v_i\},\{h_j\}) = \frac{e^{-H(\{v_i\},\{h_j\})}}{Z_{tot}}$$

The joint Hamiltonian can be defined as:

$$H(\{v_i\},\{h_j\}) = H(\{v_i\}) + H(\{h_j\}) + K(\{v_i\}) + ln\Big[\sum_{\{v_i\}} e^{-H(\{v_i\})-K(\{v_i\}}\Big]$$

It is then possible to evaluate the joint partition function:

$$\begin{aligned}
Z_{tot} &= \sum_{\{v_i\},\{h_i\}} e^{-H(\{v_i\},\{h_j\})} \\
&= \frac{1}{\sum_{\{v_i\}} e^{-H(\{v_i\})-K(\{v_i\})}} \sum_{\{v_i\},\{h_i\}} e^{-H(\{v_i\})-H(\{h_j\})-K(\{v_i\})} \\
&= \frac{1}{\sum_{\{v_i\}} e^{-H(\{v_i\})-K(\{v_i\})}} \sum_{\{v_i\}} e^{-H(\{v_i\})-K(\{v_i\})} \sum_{\{h_i\}} e^{-H(\{h_j\})} \\
&= \sum_{\{h_i\}} e^{-H(\{h_j\})} = Z
\end{aligned}$$

Therefore the joint probability and the marginal probability have the same partition function. However the marginalized probability differs from the original probability distribution:

$$\begin{aligned}
\hat{p}(\{v_i\}) &= \sum_{\{h_j\}} \hat{p}(\{v_i\},\{h_j\}) \\
&= \frac{1}{Z_{tot}} \sum_{\{h_j\}} e^{-H(\{v_i\},\{h_j\})} \\
&= \frac{1}{\sum_{\{v_i\}} e^{H(\{v_i\})+K(\{v_i\})}} e^{-H(\{v_i\})-K(\{v_i\})} \neq p(\{v_i\}) = \frac{e^{-H(\{v_i\})}}{Z}
\end{aligned}$$

2.1.3 Considerations

Although the mapping is not exhaustive, as it does not take in account all probability distributions, it sheds light on a possible deeper connection between deep learning and the renormalization group.

2.2 RG Flow - Numerical Results

In this section the numerical results of the article "*Is Deep Learning an RG Flow*" [5] are reported. The sections are divided according to the sections in the article so that it would be easier to recover the original arguments. Charts and plots in this section are borrowed from the article.

2.2.1 Flows derived from learned weights

This section takes the move from the observation that, although the RBM Flow[2] progresses toward the critical point, in contrast with the RG flow, the RBM yields to impressively accurate predictions for the critical exponents of the Ising Model. An additional discrepancy consists in the flow of the lattice's size. Over the course of the RBM Flow the size of the lattice remains unchanged while the RG Flows tends to reduce it. In order to measure the compatibility with the RG flow two operators are chosen:

$$s_{ij} = \sigma_{ij} - \bar{\sigma}$$
$$\epsilon_{ij} = s_{ij}(s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1}) - \bar{\epsilon}_{ij}$$

where $\bar{\sigma}$ is the average spin in that particular configuration and

$$\bar{\epsilon}_{ij} = \frac{1}{2N} \sum_{ij} s_{ij}(s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1})$$

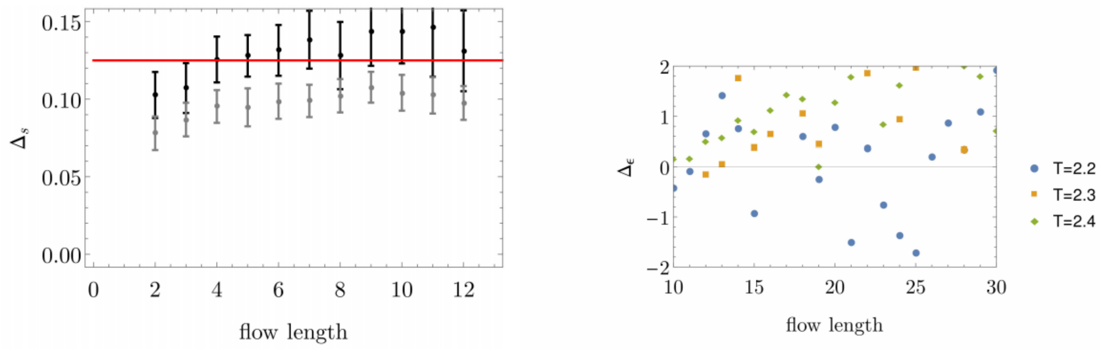
the average of the operator $\bar{\epsilon}_{ij}$. The choice felt onto these operators for their behaviour near the critical point. In a neighbourhood of T_c the correlation function between two spins can be approximated as

$$< \phi(\vec{x}_1)\phi(\vec{x}_2) >= \frac{B}{|\vec{x}_1 - \vec{x}_2|^{2\Delta}}$$

where $\vec{x}$ is the position of the spin in the lattice and B a parameter dependent on the chosen function ϕ. For the chosen observable the power law has $\Delta_s = 1/8$ and $\Delta_\epsilon = 1$. These are extremely suitable as the critical exponent $\Delta_s = 1/8$ decrease slowly and is a measure of long range correlation while $\Delta_\epsilon = 1$ decrease faster and is a measure of short range interactions.

The RBM features a 10 × 10 input layer and a 9 × 9 hidden layer. The training data include 20000 samples for each temperature between 0 and 5.9 in increments of 0.1 for a grand total of 1200000 configurations. The training used 10000 iterations of contrastive divergence.

[2] By RBM Flow it is indicated the continuous back and forth in a RBM where the reconstructed vector $\boldsymbol{v}'$ serves as input in the next step. This process is iterated many times and all the $\boldsymbol{v}$ and $\boldsymbol{h}$ generated are referred to as RBM flow steps.



(a)

(b)

Figure 2.1: (a) Evolution of the Δ_s exponent as a function of flow length. The black points correspond to a temperature of 2.3 and the grey points to 2.2 (T_c=2.269). The points converge to $0.125 = 1/8$ as expected. (b) Evolution of the Δ_ϵ exponent at various temperature. Error bars are not shown as they are larger than the y scale.

Results are shown in **Figure** 2.1. The critical exponents are evaluated through a fit. The plot on the left shows the fitted exponents Δ_s as a function of the flow length. It is possible to notice that the exponent converges to the expected value. Thus, long range correlation are correctly determined. On the other hand the right plot shows the inability of the RBM to reconstruct short range correlations as the error bar were larger than the scale taken in account.

2.2.2 Flows derived from deep learning

The connection between RG and RBM implies a correlation between the visible layer and the hidden layer. This section aims at finding, if present, such correlation. The metric used is the correlator $< v_i h_j >$. The training set consists in 30000 configuration of Ising model generated via Monte Carlo simulations near the critical temperature $T_c = 2.269$. The size of the lattice, 32×32, is chosen so that the lattice remains relevant even after two steps of 2×2 RG (see *Section* 1.3.2). In particular the lattice's dimensions after one and two RG steps are, respectively, 16×16 and 8×8. The RG transformation is done by replacing the spins value in the 2×2 square with the average of those spins. Periodic boundary conditions are chosen.

Pattern Generated by RG: Consider the plot in *Figure* 2.2. These plots were obtained by applying one or two steps of BST to the Ising configurations. Each panel corresponds to a hidden spin and its pattern represents its relations with the

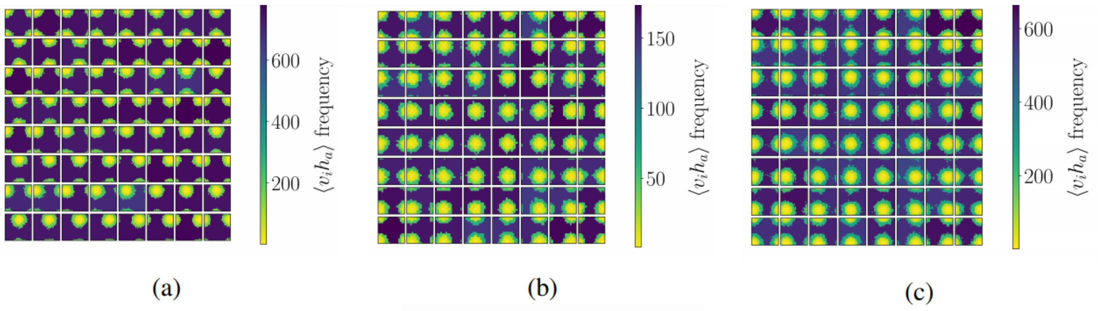


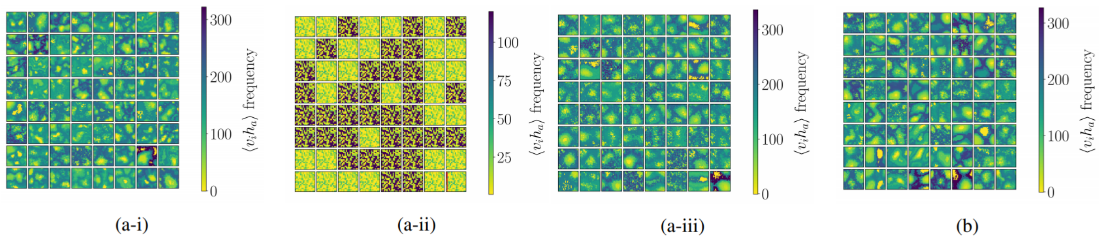
Figure 2.2: Frequency correlation plots generated through BST. (a) Analyses the correlation between the input data and the first RG passage, (b) the correlation between the output of the first RG passage and the output of the second, (c) shows the correlation between the input data and after two steps of RG.
Recall that to a higher frequency corresponds a lower correlation.

coupled spins. The colour map is chosen to reflect the number of spins with those value of correlation. As farther spins will have low (and similar) correlation values, to higher frequencies correspond lower correlations. It is clearly possible to see a brighter peak near the relative spin suggesting an higher correlation with nearby spins.

Pattern Generated by RBM: Next, two different RBMs are taken in account. One consists in a DNN made of two stacked RBM. The visible layer consists in a 32×32 matrix, the first hidden layer in a 16×16 matrix and the second hidden layer in a 8×8 matrix. The second RBM presents only a single hidden layer with 8×8 spins and a 32×32 input layer. Dimensions are chosen in order to mimic the RG steps made in the previous simulation. The correlations are reported in *Figure* 2.3 *a*. Although a distinct bright spot does not occur it is still possible to observe the emergence of bright spots between the layers. This shows that the trained RBM is, indeed, performing some kind of coarse graining. A possible explanation for the discrepancy is that the RBM ”may be choosing” a different RG procedure where a small contribution for any spin is taken in account and then averaged out. It is important to remember that, unlike the BST, a RBM presents connections between each hidden spin and all the visible spins and that weights are unlikely to be set to zero.

2.2.3 Study on the temperature flow

As a final quantitative test, the temperature flow is studied. The temperature is a relevant parameter so its value is expected to grow after each RG step. In detail, as the length of the lattice keeps halving, the temperature will roughly double at each step. The temperature of a configuration is extracted with the use of a supervised network. The supervised network is trained to measure discrete temperatures of $T = 0, 0.1, ..., 5.9$. The deep network is made of three stacked RBMs with sizes 64×64 (visible layer), 32×32 (first hidden layer), 16×16 (second hidden layer) and 8×8 (third hidden layer) and is trained on Ising data near the critical temperature. *Figure* 2.4 shows the results of this analysis. The first plot shows the results based on the analytical RG steps. It's possible to see that the temperature, corresponding to the peaks, double at each step. The (b) plots are obtained by choosing, respectively, starting configurations with temperatures of $T_{b-i} = 2.269$, $T_{b-ii} = 2$ and $T_{b-iii} = 2.7$. It is possible to notice that none of the plots resemble (a). This results excludes the possibility that each layer is linked to a renormalization step.



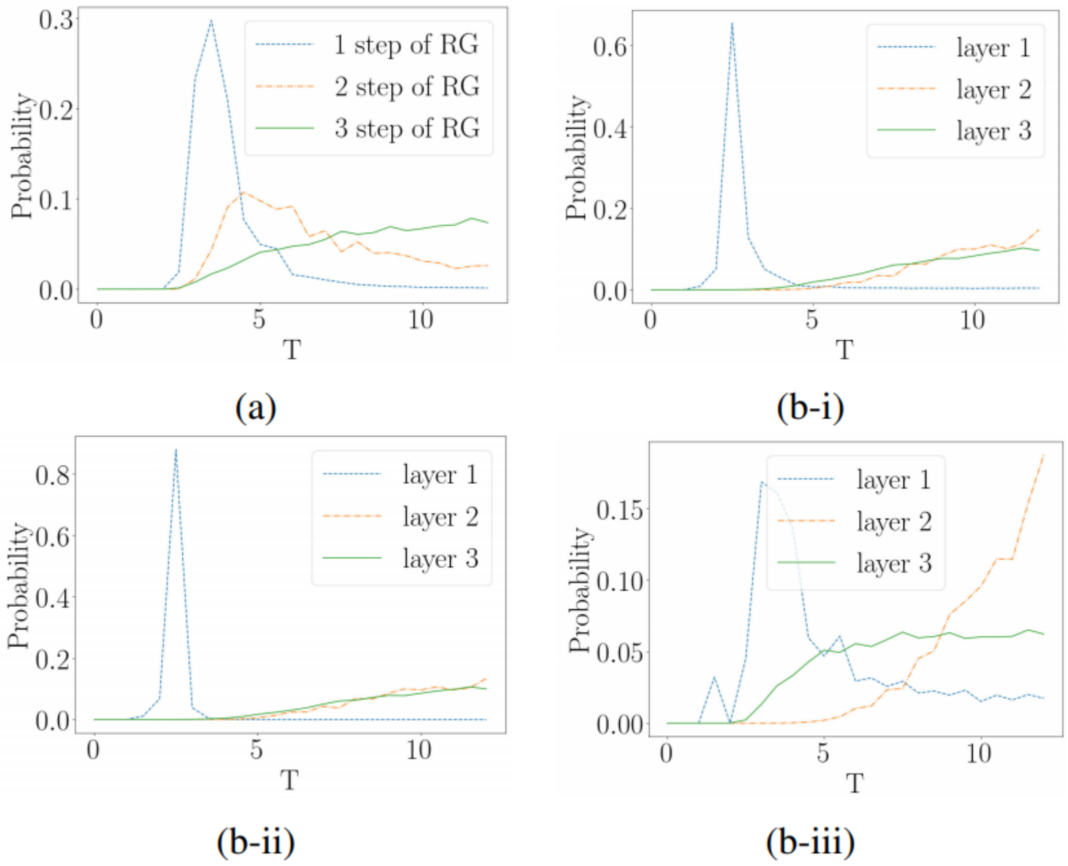
(a-i)

(a-ii)

(a-iii)

(b)

Figure 2.3: Frequency correlation plots generated through RBM. (a-i) Analyse the correlation between the input data and the first hidden layer, (a-ii) the correlation between the first hidden layer and the second, (a-ii) shows the correlation between the input data and the second hidden layer. (b) shows the correlation between the input data and the hidden layer of the second RBM.



(a)

(b-i)

(b-ii)

(b-iii)

Figure 2.4: (a) shows the temperature measured after the RG steps while (b) shows the temperatures measured on the configuration obtained thought RBM Flow. The temperatures chosen for the input configurations are $T_a = T_c = 2.269$, $T_{b-i} = 2.269$, $T_{b-ii} = 2$ and $T_{b-iii} = 2.7$ while the sharp peak is taken as reference for the layer's temperature.

Chapter 3

Conclusions

In this thesis a possible contact point between Deep Learning and Renormalization Group is investigated. After providing the essential background the articles "*An exact mapping between variational renormalization group and deep learning*"[4] and "*Is deep learning an RG Flow?*"[5] are examined. The first article studied a possible theoretical connection between the Renormalization group and deep learning. Although [2, 5] show that the mapping is not generalizable, the results mark a starting point for an in depth investigation of a possible relation between restricted Boltzmann machines and the renormalization group. The second article performed numerical simulations in order to search a correlation between the visible layer and the hidden layers of a RBM. This study shows that RBMs are indeed performing some kind of coarse graining, but failed to link the RBM flow to the RG Flow.

In conclusion an exact mapping between Deep Learning and Renormalization Group seems unlikely but deep learning and restricted Boltzmann machine seem to have the potential to shed light on RG and lead to a better understanding of coarse graining methods.

Bibliography

[1] Jianqing Fan, Cong Ma, and Yiqiao Zhong. *A Selective Overview of Deep Learning*. 2019. arXiv: `1904.05526 [stat.ML]` (cit. on pp. 3, 14).

[2] Henry W. Lin, Max Tegmark, and David Rolnick. «Why Does Deep and Cheap Learning Work So Well?» In: *Journal of Statistical Physics* 168.6 (2017), pp. 1223–1247. ISSN: 1572-9613. DOI: `10.1007/s10955-017-1836-5`. URL: `http://dx.doi.org/10.1007/s10955-017-1836-5` (cit. on pp. 4, 17, 18, 25).

[3] Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. New York: Oxford University Press Inc., 2010 (cit. on p. 10).

[4] Pankaj Mehta and David J. Schwab. *An exact mapping between the Variational Renormalization Group and Deep Learning*. 2014. arXiv: `1410.3831 [stat.ML]` (cit. on pp. 17, 25).

[5] Ellen de Mello Koch, Robert de Mello Koch, and Ling Cheng. *Is Deep Learning an RG Flow?* 2019. arXiv: `1906.05212 [cs.LG]` (cit. on pp. 17, 20, 25).