

An exact mapping between the Variational Renormalization Group and Deep Learning

Pankaj Mehta

Dept. of Physics, Boston University, Boston, MA

David J. Schwab

Dept. of Physics, Northwestern University, Evanston, IL

Deep learning is a broad set of techniques that uses multiple layers of representation to automatically learn relevant features directly from structured data. Recently, such techniques have yielded record-breaking results on a diverse set of difficult machine learning tasks in computer vision, speech recognition, and natural language processing. Despite the enormous success of deep learning, relatively little is understood theoretically about why these techniques are so successful at feature learning and compression. Here, we show that deep learning is intimately related to one of the most important and successful techniques in theoretical physics, the renormalization group (RG). RG is an iterative coarse-graining scheme that allows for the extraction of relevant features (i.e. operators) as a physical system is examined at different length scales. We construct an exact mapping from the variational renormalization group, first introduced by Kadanoff, and deep learning architectures based on Restricted Boltzmann Machines (RBMs). We illustrate these ideas using the nearest-neighbor Ising Model in one and two-dimensions. Our results suggests that deep learning algorithms may be employing a generalized RG-like scheme to learn relevant features from data.

A central goal of modern machine learning research is to learn and extract important features directly from data. Among the most promising and successful techniques for accomplishing this goal are those associated with the emerging sub-discipline of deep learning. Deep learning uses multiple layers of representation to learn descriptive features directly from training data [1, 2] and has been successfully utilized, often achieving record breaking results, in difficult machine learning tasks including object labeling [3], speech recognition [4], and natural language processing [5].

In this work, we will focus on a set of deep learning algorithms known as deep neural networks (DNNs) [6]. DNNs are biologically-inspired graphical statistical models that consist of multiple layers of “neurons”, with units in one layer receiving inputs from units in the layer below them. Despite their enormous success, it is still unclear what advantages these deep, multi-layer architectures possess over shallower architectures with a similar number of parameters. In particular, it is still not well understood theoretically why DNNs are so successful at uncovering features in structured data. (But see [7–9].)

One possible explanation for the success of DNN architectures is that they can be viewed as an iterative coarse-graining scheme, where each new high-level layer of the neural network learns increasingly abstract higher-level features from the data [1, 10]. The initial layers of the the DNN can be thought of as low-level feature detectors which are then fed into higher layers in the DNN which combine these low-level features into more abstract higher-level features, providing a useful, and at times reduced, representation of the data. By successively applying feature extraction, DNNs learn to deemphasize irrelevant features in the data while simultaneously learning relevant ones. (Note that in a supervised setting, such as classification, relevant and irrelevant are ultimately determined by the problem at hand. Here we are con-

cerned solely with the unsupervised aspect of training DNNs, and the use of DNNs for compression [6].) In what follows, we make this explanation precise.

This successive coarse-graining procedure is reminiscent of one of the most successful and important tools in theoretical physics, the renormalization group (RG) [11, 12]. RG is an iterative coarse-graining procedure designed to tackle difficult physics problems involving many length scales. The central goal of RG is to extract relevant features of a physical system for describing phenomena at large length scales by integrating out (i.e. marginalizing over) short distance degrees of freedom. In any RG sequence, fluctuations are integrated out starting at the microscopic scale and then moving iteratively on to fluctuations at larger scales. Under this procedure, certain features, called relevant operators, become increasingly important while other features, dubbed irrelevant operators, have a diminishing effect on the physical properties of the system at large scales.

In general, it is impossible to carry out the renormalization procedure exactly. Therefore, a number of approximate RG procedures have been developed in the theoretical physics community [12–15]. One such approximate method is a class of variational “real-space” renormalization schemes introduced by Kadanoff for performing RG on spin systems [14, 16, 17]. Kadanoff’s variational RG scheme introduces coarse-grained auxiliary, or “hidden”, spins that are coupled to the physical spin systems through some unknown coupling parameters. A parameter-dependent free energy is calculated for the coarse-grained spin system from the coupled system by integrating out the physical spins. The coupling parameters are chosen through a variational procedure that minimizes the difference between the free energies of the physical and hidden spin systems. This ensures that the coarse-grained system preserves the long-distance information present in the physical system. Carrying out this

procedure results in an RG transformation that maps the physical spin system into a coarse-grained description in terms of hidden spins. The hidden spins then serve as the input for the next round of renormalization.

The introduction of layers of hidden spins is also a central component of DNNs based on Restricted Boltzmann Machines (RBMs). In RBMs, hidden spins (often called units or neurons) are coupled to “visible” spins describing the data of interest. (Here we restrict ourselves to binary data.) The coupling parameters between the visible and hidden layers are chosen using a variational procedure that minimizes the Kullback-Leibler divergence (i.e. relative entropy) between the “true” probability distribution of the data and the variational distribution obtained by marginalizing over the hidden spins. Like in variational RG, RBMs can be used to map a state of the visible spins in a data sample into a description in terms of hidden spins. If the number of hidden units is less than the number of visible units, such a mapping can be thought of as a compression. (Note, however, that dimensional expansions are common [18].) In deep learning, individual RBMs are stacked on top of each other into a DNN [6, 19], with the output of one RBM serving as the input to the next. Moreover, the variational procedure is often performed iteratively, layer by layer.

The preceding paragraphs suggest an intimate connection between RG and deep learning. Indeed, here we construct an *exact mapping* from the variational RG scheme of Kadanoff to DNNs based on RBMs [6, 19]. Our mapping suggests that DNNs implement a generalized RG-like procedure to extract relevant features from structured data.

The paper is organized as follows. We begin by reviewing Kadanoff’s variational renormalization scheme in the context of the Ising Model. We then introduce RBMs and deep learning architectures of stacked RBMs. We then show how to map the procedure of variational RG to unsupervised training of a DNN. We illustrate these ideas using the one- and two-dimensional nearest-neighbor Ising models. We end by discussing the implication of our mapping for physics and machine learning.

I. OVERVIEW OF VARIATIONAL RG

In statistical physics, one often considers an ensemble of N binary spins $\{v_i\}$ that can take the values ± 1 . The index i labels the position of spin v_i in some lattice. In thermal equilibrium, the probability of a spin configuration is given by the Boltzmann distribution

$$P(\{v_i\}) = \frac{e^{-\mathbf{H}(\{v_i\})}}{Z}, \quad (1)$$

where we have defined the Hamiltonian $\mathbf{H}(\{v_i\})$, and the partition function

$$Z = \text{Tr}_{v_i} e^{-\mathbf{H}(\{v_i\})} \equiv \sum_{v_1, \dots, v_N = \pm 1} e^{-\mathbf{H}(\{v_i\})}. \quad (2)$$

Note throughout the paper we set the temperature equal to one, without loss of generality. Typically, the Hamiltonian depends on a set of couplings or parameters, $\mathbf{K} = \{K_s\}$, that parameterizes the set of all possible Hamiltonians. For example, with binary spins, the \mathbf{K} could be the couplings describing the spin interactions of various orders:

$$\mathbf{H}[\{v_i\}] = - \sum_i K_i v_i - \sum_{ij} K_{ij} v_i v_j - \sum_{ijk} K_{ijk} v_i v_j v_k + \dots \quad (3)$$

Finally, we can define the free energy of the spin system in the standard way:

$$F^v = - \log Z = - \log \left(\text{Tr}_{v_i} e^{-\mathbf{H}(\{v_i\})} \right). \quad (4)$$

The idea behind RG is to find a new coarse-grained description of the spin system where one has “integrated out” short distance fluctuations. To this end, let us introduce $M < N$ new binary spins, $\{h_j\}$. Each of these spins h_j will serve as a coarse-grained degree of freedom where fluctuations on small scales have been averaged out. Typically, such a coarse-graining procedure increases some characteristic length scale describing the system such as the lattice spacing. For example, in the block spin renormalization picture introduced by Kadanoff, each h_i represents the state of a local block of physical spins, v_i . Figure 1 shows such a block-spin procedure for a two-dimensional spin system on a square lattice, where each h_i represents a 2×2 block of visible spins. The result of such a coarse-graining procedure is that the lattice spacing is doubled at each step of the renormalization procedure.

In general, the interactions (statistical correlations) between the $\{v_i\}$ induce interactions (statistical correlations) between the coarse-grained spins, $\{h_j\}$. In particular, the coarse-grained system can be described by a new coarse-grained Hamiltonian of the form

$$\mathbf{H}^{RG}[\{h_j\}] = - \sum_i \tilde{K}_i h_i - \sum_{ij} \tilde{K}_{ij} h_i h_j - \sum_{ijk} \tilde{K}_{ijk} h_i h_j h_k + \dots, \quad (5)$$

where $\{\tilde{K}\}$ describe interactions between the hidden spins, $\{h_j\}$. In the physics literature, such a renormalization transformation is often represented as mapping between couplings, $\{K\} \rightarrow \{\tilde{K}\}$. Of course, the exact mapping depends on the details of the RG scheme used.

In the variational RG scheme proposed by Kadanoff, the coarse graining procedure is implemented by constructing a function, $\mathbf{T}_\lambda(\{v_i\}, \{h_j\})$, that depends on a set of variational parameters $\{\lambda\}$ and encodes (typically pairwise) interactions between the physical and coarse-grained degrees of freedom. After coupling the auxiliary spins $\{h_j\}$ to the physical spins $\{v_i\}$, one can then integrate out (marginalize over) the visible spins to arrive at a coarse-grained description of the physical system entirely in terms of the $\{h_j\}$. The function $\mathbf{T}_\lambda(\{v_i\}, \{h_j\})$ then naturally defines a Hamiltonian for the $\{h_j\}$ through the

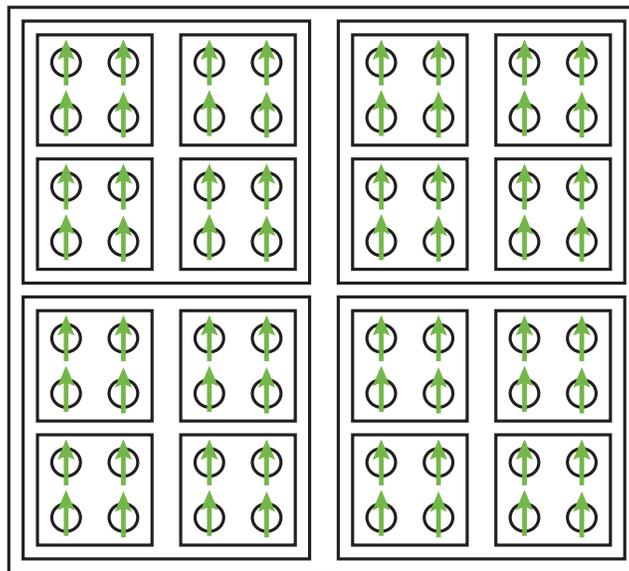


FIG. 1. **Block spin renormalization.** In block spin renormalization [14], a physical system is coarse grained by introducing new “block” variables which describe some “effective” behavior of a block of spins. For example, in the figure, four adjacent spins are grouped into 2×2 blocks. The system is then described in terms of these new block variables. This scheme is then iterated to create even new block variables that average over an even larger set of the original spins. Notice the lattice spacing doubles after each iteration.

expression

$$e^{-\mathbf{H}_\lambda^{\text{RG}}[\{h_j\}]} \equiv \text{Tr}_{v_i} e^{\mathbf{T}_\lambda(\{v_i\}, \{h_j\}) - \mathbf{H}(\{v_i\})}. \quad (6)$$

We can also define a free energy for the coarse grained system in the usual way

$$F_\lambda^h = -\log \left(\text{Tr}_{h_i} e^{-\mathbf{H}^{\text{RG}}_\lambda(\{h_i\})} \right). \quad (7)$$

Thus far we have ignored the problem of choosing the variational parameters λ that define our RG transformation $\mathbf{T}_\lambda(\{v_i\}, \{h_j\})$. Intuitively, it is clear we should choose λ to ensure that the long-distance physical observables of the system are invariant to this coarse graining procedure. This is done by choosing the parameters λ to minimize the free energy difference, $\Delta F = F_\lambda^h - F^v$, between the physical and coarse grained systems. Notice that

$$\Delta F = 0 \iff \text{Tr}_{h_j} e^{\mathbf{T}_\lambda(\{v_i\}, \{h_j\})} = 1 \quad (8)$$

Thus, for any *exact* RG transformation, we know that

$$\text{Tr}_{h_j} e^{\mathbf{T}_\lambda(\{v_i\}, \{h_j\})} = 1 \quad (9)$$

In general, it is not possible to choose the parameters λ to satisfy the condition above and various variational schemes (e.g. bond moving) have been proposed to choose λ to minimize this ΔF .

II. RBMS AND DEEP NEURAL NETWORKS

We will show below that this variational RG procedure has a natural interpretation as a deep learning scheme

based on a powerful class of energy-based models called Restricted Boltzmann Machines (RBMs) [6, 20–23]. We will restrict our discussion to RBMs acting on binary data [6] drawn from some probability distribution, $P(\{v_i\})$, with $\{v_i\}$ binary spins labeled by an index $i = 1 \dots N$. For example, for black and white images each spin v_i encodes whether a given pixel is on or off and the distribution $P(\{v_i\})$ encodes the statistical properties of the ensemble of images (e.g the set of all handwritten digits in the MNIST dataset).

To model the data distribution, RBMs introduce new hidden spin variables, $\{h_j\}$ ($j = 1 \dots M$) that couple to the visible units. The interactions between visible and hidden units are modeled using an energy function of the form

$$\mathbf{E}(\{v_i\}, \{h_j\}) = \sum_i b_j h_j + \sum_{ij} v_i w_{ij} h_j + \sum_i c_i v_i, \quad (10)$$

where $\lambda = \{b_j, w_{ij}, c_i\}$ are variational parameters of the model. In terms of this energy function, the joint probability of observing a configuration of hidden and visible spins can be written as

$$p_\lambda(\{v_i\}, \{h_j\}) = \frac{e^{-\mathbf{E}(\{v_i\}, \{h_j\})}}{\mathcal{Z}}. \quad (11)$$

This joint distribution also defines a variational distribution for the visible spins

$$p_\lambda(\{v_i\}) = \sum_{\{h_j\}} p_\lambda(\{v_i\}, \{h_j\}) = \text{Tr}_{h_j} p_\lambda(\{v_i\}, \{h_j\}) \quad (12)$$

as well as a marginal distribution for hidden spins themselves:

$$p_\lambda(\{h_j\}) = \sum_{\{v_i\}} p_\lambda(\{v_i\}, \{h_j\}) = \text{Tr}_{v_i} p_\lambda(\{v_i\}, \{h_j\}). \quad (13)$$

Finally, for future reference it will be helpful to define a ‘‘variational’’ RBM Hamiltonian for the visible units:

$$p_\lambda(\{v_i\}) \equiv \frac{e^{-\mathbf{H}_\lambda^{RBM}[\{v_i\}]}}{\mathcal{Z}}, \quad (14)$$

and an RBM Hamiltonian for the hidden units:

$$p_\lambda(\{h_j\}) \equiv \frac{e^{-\mathbf{H}_\lambda^{RBM}[\{h_j\}]}}{\mathcal{Z}}. \quad (15)$$

Since the objective of the RBM for our purposes is unsupervised learning, the parameters in the RBM are chosen to minimize the Kullback-Leibler divergence between the true distribution of the data $P(\{v_i\})$ and the variational distribution $p_\lambda(\{v_i\})$:

$$D_{KL}(P(\{v_i\})||p_\lambda(\{v_i\})) = \sum_{\{v_i\}} P(\{v_i\}) \log \left(\frac{P(\{v_i\})}{p_\lambda(\{v_i\})} \right). \quad (16)$$

Furthermore, notice that when the RBM exactly reproduces the visible data distribution

$$D_{KL}(P(\{v_i\})||p_\lambda(\{v_i\})) = 0. \quad (17)$$

In general it not possible to explicitly minimize the $D_{KL}(P(\{v_i\})||p_\lambda(\{v_i\}))$ and this minimization is usually performed using approximate numerical methods such as contrastive divergence [24]. Note that if the number of hidden units is restricted (i.e. less than 2^N), the RBM cannot be made to match an arbitrary distribution exactly [9].

In a DNN, RBMs are stacked on top of each other so that, once trained, the hidden layer of one RBM serves as the visible layer of the next RBM. In particular, one can map a configuration of visible spins to a configuration in the hidden layer via the conditional probability distribution, $p_\lambda(\{h_j\}|\{v_i\})$. Thus, after training an RBM, we can treat the activities of the hidden layer in response to each visible data sample as data for learning a second layer of hidden spins, and so on.

III. MAPPING VARIATIONAL RG TO DEEP LEARNING

In variational RG, the couplings between the hidden and visible spins are encoded by the operators $\mathbf{T}_\lambda(\{v_i\}, \{h_j\})$. In RBMs, an analogous role is played by the joint energy function $\mathbf{E}(\{v_i\}, \{h_j\})$. In fact, as we will show below, these objects are related through the equation,

$$\mathbf{T}(\{v_i\}, \{h_j\}) = -\mathbf{E}(\{v_i\}, \{h_j\}) + \mathbf{H}[\{v_i\}], \quad (18)$$

where $\mathbf{H}[\{v_i\}]$ is the Hamiltonian defined in Eq. 3 that encodes the data probability distribution $P(\{v_i\})$. This equation defines a one-to-one mapping between the variational RG scheme and RBM based DNNs.

Using this definition, it is easy to show that the Hamiltonian $\mathbf{H}_\lambda^{RG}[\{h_j\}]$, originally defined in Eq. 6 as the Hamiltonian of the coarse-grained degrees of freedom after performing RG, also describes the hidden spins in the RBM. This is equivalent to the statement that the marginal distribution $p_\lambda(\{h_j\})$ describing the hidden spins of the RBM is of the Boltzmann form with a Hamiltonian $\mathbf{H}_\lambda^{RG}[\{h_j\}]$. To prove this, we divide both sides of Eq. 6 by \mathcal{Z} to get

$$\frac{e^{-\mathbf{H}_\lambda^{RG}[\{h_j\}]}}{\mathcal{Z}} = \frac{\text{Tr}_{v_i} e^{\mathbf{T}_\lambda(\{v_i\}, \{h_j\}) - \mathbf{H}(\{v_i\})}}{\mathcal{Z}}. \quad (19)$$

Substituting Eq. 18 into this equation yields

$$\frac{e^{-\mathbf{H}_\lambda^{RG}[\{h_j\}]}}{\mathcal{Z}} = \text{Tr}_{v_i} \frac{e^{-\mathbf{E}(\{v_i\}, \{h_j\})}}{\mathcal{Z}} = p_\lambda(\{h_j\}). \quad (20)$$

Substituting Eq. 15 into the right-hand side yields the desired result

$$\mathbf{H}_\lambda^{RG}[\{h_j\}] = \mathbf{H}_\lambda^{RBM}[\{h_j\}]. \quad (21)$$

These results also provide a natural interpretation for variational RG entirely in the language of probability theory. The operator $\mathbf{T}_\lambda(\{v_i\}, \{h_j\})$ can be viewed as a variational approximation for the conditional probability of the hidden spins given the visible spins. To see this, notice that

$$\begin{aligned} e^{\mathbf{T}(\{v_i\}, \{h_j\})} &= e^{-\mathbf{E}(\{v_i\}, \{h_j\}) + \mathbf{H}[\{v_i\}]} \\ &= \frac{p_\lambda(\{v_i\}, \{h_j\})}{p_\lambda(\{v_i\})} e^{\mathbf{H}[\{v_i\}] - \mathbf{H}_\lambda^{RBM}[\{v_i\}]} \\ &= p_\lambda(\{h_j\}|\{v_i\}) e^{\mathbf{H}[\{v_i\}] - \mathbf{H}_\lambda^{RBM}[\{v_i\}]} \end{aligned} \quad (22)$$

where in going from the first the line to the second line we have used Eqs. 11 and 14. This implies that when an RG can be performed exactly (i.e. the RG transformation satisfies the equality $\text{Tr}_{h_j} e^{\mathbf{T}_\lambda(\{v_i\}, \{h_j\})} = 1$), the variational Hamiltonian is identical to the true Hamiltonian describing the data, $\mathbf{H}[\{v_i\}] = \mathbf{H}_\lambda^{RBM}[\{v_i\}]$ and $\mathbf{T}(\{v_i\}, \{h_j\})$ is exactly the conditional probability. In the language of probability theory, this means that the variational distribution $p_\lambda(\{v_i\})$ exactly reproduces the true data distribution $P(\{v_i\})$ and $D_{KL}(P(\{v_i\})||p_\lambda(\{v_i\})) = 0$.

In general, it is not possible to perform the variational RG transformation exactly. Instead, one constructs a family of variational approximations for the exact RG transform [14, 16, 17]. The discussion above makes it clear that these variational distributions work at the level of the Hamiltonians and Free Energies. In contrast, in the Machine Learning literature, these variational approximations are usually made by minimizing the KL divergence $D_{KL}(P(\{v_i\})||p_\lambda(\{v_i\})) = 0$. Thus,

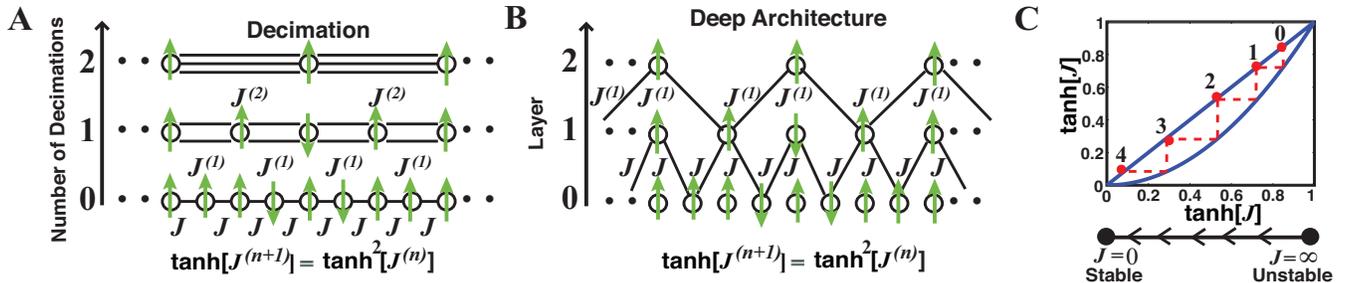


FIG. 2. **RG and deep learning in the one-dimensional Ising Model.** (A) A decimation based renormalization transformation for the ferromagnetic 1-D Ising model. At each step, half the spins are decimated, doubling the effective lattice spacing. After, n successive decimations, the spins can be described using a new 1-D Ising models with a coupling J^n between spins. Couplings at a given layer are related to couplings at a previous layer through the square of the hyperbolic tangent function. (B) Decimation-based renormalization transformations can also be realized using the deep architecture where the weights between the $n + 1$ and n -th hidden layer are given by J^n . (C) Visualizing the renormalization group flow of the couplings for 1-D Ferromagnetic Ising model. Under four successive decimations or equivalently as we move up four layers in the deep architecture, the couplings (marked by red dots) get smaller. Eventually, the couplings are attracted to stable fixed point $J = 0$.

the two approaches employ distinct variational approximation schemes for coarse graining. Finally, notice that the correspondence does not rely on the explicit form of the energy $E(\{h_j\}, \{v_j\})$ and hence holds for any Boltzmann Machine.

IV. EXAMPLES

To gain intuition about the mapping between RG and deep learning, it is helpful to consider some simple examples in detail. We begin by examining the one-dimensional nearest-neighbor Ising model where the RG transformation can be carried out exactly. We then numerically explore the two-dimensional nearest-neighbor Ising model using an RBM-based deep learning architecture.

A. One dimensional Ising Model

The one-dimensional Ising model describes a collection of binary spins $\{v_i\}$ organized along a one-dimensional lattice with lattice spacing a . Such a system is described by a Hamiltonian of the form

$$H = -J \sum_i v_i v_{i+1}, \quad (23)$$

where J is a ferromagnetic coupling that energetically favors configurations where neighboring spins align. To perform a RG transformation, we decimate (marginalize over) every other spin. This doubles the lattice spacing $a \rightarrow 2a$ and results in a new effective interaction $J^{(1)}$ between spins (see Figure 2). If we denote the coupling after performing n successive RG transformations by $J^{(n)}$,

then a standard calculation shows that these coefficients satisfy the RG equations

$$\tanh [J^{(n+1)}] = \tanh^2 [J^{(n)}], \quad (24)$$

where we have defined $J^{(0)} = J$ [14]. This recursion relationship can be visualized as a one-dimensional flow in the coupling space J from $J = \infty$ to $J = 0$. Thus, after performing RG the interactions become weaker and weaker and $J \rightarrow 0$ as $n \rightarrow \infty$.

This RG transformation also naturally gives rise to the deep learning architecture shown in Figure 2. The spins at a given layer of the DNN have a natural interpretation as the decimated spins when performing the RG transformation in the layer below. Notice that the coupled spins in the bottom two layers of the DNNs in Fig. 2B form an “effective” one-dimensional chain isomorphic to the original spin chain. Thus, marginalizing over spins in the bottom layer in the DNN is identical to decimating every other spin in the original spin systems. This implies that the “hidden” spins in the second layer of the DNN are also described by the RG transformed Hamiltonian with a coupling $J^{(1)}$ between neighboring spins. Repeating this argument for spins coupled between the second and third layers and so on, one obtains the deep learning architecture shown in Fig. 2B which implements decimation.

The advantage of the simple deep architecture presented here is that it is easy to interpret and requires no calculations to construct. However, an important shortcoming is that it contains no information about half of the visible spins, namely the spins that do not couple to the hidden layer.

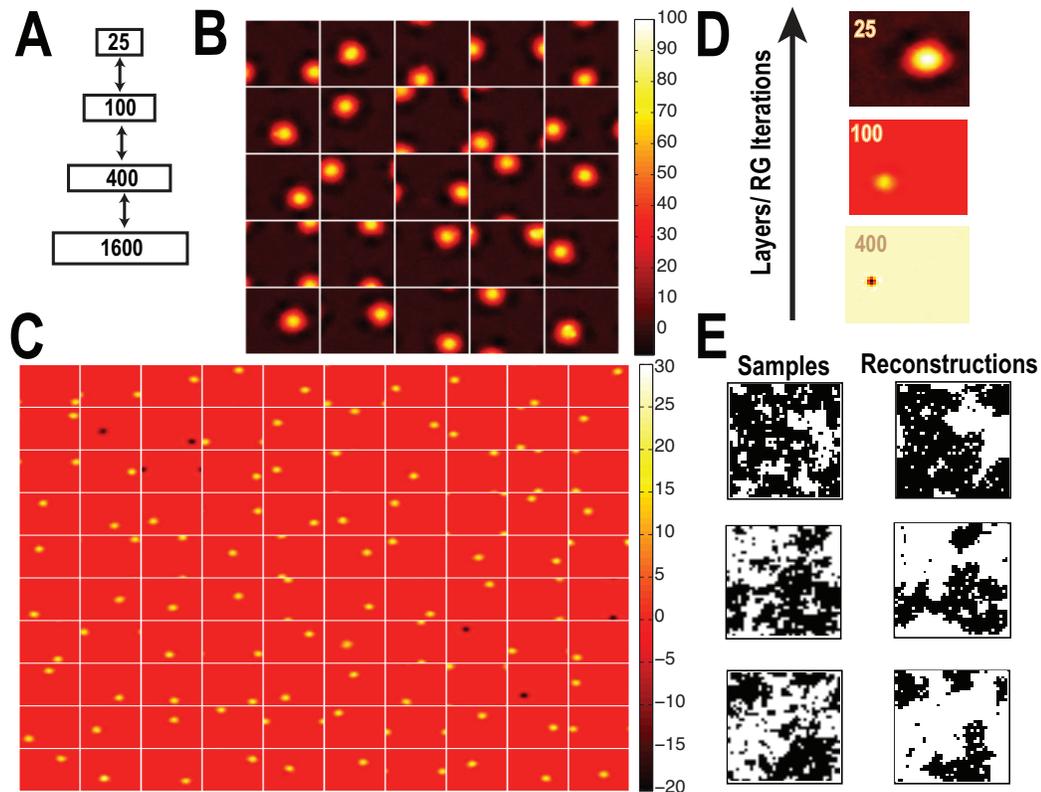


FIG. 3. **Deep learning the 2D Ising model** **A** Deep Neural Network with four layers of size 1600, 400, 100, and 25 spins was trained using samples drawn from a 2D Ising model slightly above the critical temperature, $J/(k_B T) = 0.408$. **B** Visualization of the effective receptive fields for the top layer of spins. Each 40 by 40 pixel image depicts the effective receptive field of one of the 25 spins in the top layer (see material and methods) **C** Visualization of effective receptive fields for each of the 100 spins in the middle layer calculated as in **B**. **D** The effective receptive fields get larger as one moves up the Deep Neural Network. This is consistent with what is expected from the successive application of block renormalization. **E** Three representative samples drawn from the 2D Ising model at $J = 0.408$ and their reconstruction from the trained DNN. Samples were reconstructed from DNNs as in [6].

B. Two dimensional Ising Model

We next applied deep learning techniques to numerically coarse-grain the two-dimensional nearest-neighbor Ising model on a square lattice. This model is described by a Hamiltonian of the form

$$H[\{v_i\}] = -J \sum_{\langle ij \rangle} v_i v_j, \quad (25)$$

where $\langle ij \rangle$ indicates that i and j are nearest neighbors and J is a ferromagnetic coupling that favors configurations where neighboring spins align. Unlike the one-dimensional Ising model, the two dimensional Ising model has a phase transition that occurs when $J/(k_B T) = 0.4352$ (recall we have set $\beta = T^{-1} = 1$). At the phase transition, the characteristic length scale of the system, the correlation length, diverges. For this reason, near a critical point the system can be productively coarse-grained using a procedure similar to Kadanoff's block spin renormalization (see Fig. 1) [14].

Inspired by our mapping between variational RG and DNNs, we applied standard deep learning techniques to samples generated from the 2D Ising model for $J = 0.408$, just above the critical temperature. 20,000 samples were generated from a periodic 40×40 2D Ising model using standard equilibrium Monte Carlo techniques and served as input to an RBM-based deep neural network of four layers with 1600, 400, 100, and 25 spins respectively (see Fig. 3A). We furthermore imposed an L1 penalty on the weights between layers in the RBM and trained the network using contrastive divergence [24] (see Materials and Methods). The L1 penalty serves as a sparsity promoting regularizer that encourages weights in the RBM to be zero and prevents overfitting due to the finite number of samples. In practice, it ensures that visible and hidden spins interact with only a small subset of all the spins in an RBM. (Note that we did not use a convolutional network that explicitly builds in spatial locality or translational invariance.)

The architecture of the resulting DNN suggests that it

is implementing a coarse-graining scheme similar to block spin renormalization (see Fig. 3). Each spin in a hidden layer couples to a local block of spins in the layer below. This iterative blocking is consistent with Kadanoff’s intuitive picture of how coarse-graining should be implemented near the critical point. Moreover, the size of the blocks coupling to each hidden unit in a layer are of approximately the same size (Fig. 3B,C), and the characteristic size is increasing with layer (Fig. 3D). *Surprisingly, this local block spin structure emerges from the training process, suggesting the DNN is self-organizing to implement block spin renormalization.* Furthermore, as shown in Fig. 3E, reconstructions from the coarse grained DNN can qualitatively reproduce the macroscopic features of individual samples despite having only 25 spins in the top layer, a compression ratio of 64.

V. DISCUSSION

Deep learning is one of the most successful paradigms for unsupervised learning to emerge over the last ten years. The enormous success of deep learning techniques at a variety of practical machine learning tasks ranging from voice recognition to image classification raises natural questions about its theoretical underpinnings. Here, we have demonstrated that there is a one-to-one mapping between RBM-based Deep Neural Networks and the variational renormalization group. We illustrated this mapping by analytically constructing a DNN for the 1D Ising model and numerically examining the 2D Ising model. Surprisingly, we found that these DNNs self-organize to implement a coarse-graining procedure reminiscent of Kadanoff block renormalization. This suggests that deep learning may be implementing a generalized RG-like scheme to learn important features from data.

RG plays a central role in our modern understanding of statistical physics and quantum field theory. A central finding of RG is that the long distance physics of many disparate physical systems are dominated by the same long distance fixed points. This gives rise to the idea of universality – many microscopically dissimilar systems exhibit macroscopically similar properties at long distances. Physicists have developed elaborate technical machinery for exploiting fixed points and universality to identify the salient long distance features of physics systems. It will be interesting to see, what, if any of this more complex machinery can be imported to deep learning. A potential obstacle for importing ideas from physics into the deep learning framework is that RG is commonly applied to physical systems with many symmetries. This is in contrast to deep learning which is often applied to data with limited structure.

Recently, it was suggested that modern RG techniques developed in the context of quantum systems such as matrix product states and tensor networks have a natural interpretation in terms of variational RG [17]. These new techniques exploit ideas such as entanglement entropy

and disentanglers which create a features with a minimum amount of redundancy. It is an open question to see whether these ideas can be imported into deep learning algorithms. Our mapping also suggests a route for applying real space renormalization techniques to complicated physical systems. Real space renormalization techniques such as variational RG have often been limited by their inability to make good approximations. Techniques from deep learning may represent a possible route for overcoming these problems.

Appendix A: Learning Deep Architecture for the Two-dimensional Ising Model

Details are given in the *SI Materials and Methods*. Stacked RBMs were trained with a variant of the code from [6]. This code is available at <https://code.google.com/p/matrbm/>. In particular, only the unsupervised learning phase was performed. Individual RBMs were trained with contrastive divergence for 200 epochs, with momentum 0.5 using mini-batches of size 100 on 40,000 total samples from the 2D Ising model with $J = 0.408$. Additionally, L1 regularization was implemented, with strength 2×10^{-4} , instead of weight decay. This L1 regularization strength was chosen to ensure that one could not have all-to-all couplings between layers in the DNN. Reconstructions were performed as in [6]. See Supplementary files for a Matlab variable containing the learned model.

Appendix B: Visualizing Effective Receptive Fields

The effective receptive field is a way to visualize which spins in the visible layer that coupled to a given spin in one of the hidden layers. We denote the effective receptive field matrix of layer l by $r^{(l)}$ and the number of spins in layer l by $n^{(l)}$, with the visible layer corresponding to $l = 0$. Each column in $r^{(l)}$ is a vector that encodes the receptive field of a single spin in hidden layer l . It can be computed by convoluting the weight matrices $W^{(l)}$ encoding the weights w_{ij} between the spins in layers $l - 1$ and l . To compute $r^{(l)}$ first we set $r^{(1)} = W^{(1)}$ and used the recursion relationship $r^l = r^{(l-1)}W^{(l)}$ for $l > 1$. Thus, the effective receptive field of a spin is a measure of how much that hidden spin influences the spins in the visible layer.

ACKNOWLEDGMENTS

PM is grateful to Charles K. Fisher for useful conversations. We are also grateful to Javad Noorbakhsh and Alex Lang for comments on the manuscript. This work was partially supported by Simons Foundation Investigator Award in the Mathematical Modeling of Living Sys-

tems and a Sloan Research Fellowship (to P.M). DJS was

partially supported by NIH Grant K25 GM098875.

-
- [1] Y. Bengio, Foundations and trends® in Machine Learning, **2**, 1 (2009).
- [2] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, International Conference in Machine Learning (2012).
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Advances in Neural Information Processing Systems 25, 1097 (2012).
- [4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, Signal Processing Magazine, **29**, 82 (2012).
- [5] R. Sarikaya, G. Hinton, and A. Deoras, IEEE Transactions on Audio Speech and Language Processing (2014).
- [6] G. E. Hinton and R. R. Salakhutdinov, Science, **313**, 504 (2006).
- [7] Y. Bengio and L. Yann, Large-scale kernel machines, **34**, 1 (2007).
- [8] N. Le Roux and Y. Bengio, Neural Computation, **22**, 2192 (2010).
- [9] N. Le Roux and Y. Bengio, Neural Computation, **20**, 1631 (2008).
- [10] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, Advances in neural information processing systems, **19**, 153 (2007).
- [11] K. G. Wilson and J. Kogut, Physics Reports, **12**, 75 (1974).
- [12] K. G. Wilson, Reviews of Modern Physics, **55**, 583 (1983).
- [13] J. Cardy, *Scaling and renormalization in statistical physics*, Vol. 5 (Cambridge University Press, 1996).
- [14] L. P. Kadanoff, *Statics, Dynamics and Renormalization* (World Scientific, 2000).
- [15] N. Goldenfeld, (1992).
- [16] L. P. Kadanoff, A. Houghton, and M. C. Yalabik, Journal of Statistical Physics, **14**, 171 (1976).
- [17] E. Efrati, Z. Wang, A. Kolan, and L. P. Kadanoff, Reviews of Modern Physics, **86**, 647 (2014).
- [18] Y. Bengio, A. Courville, and P. Vincent, Pattern Analysis and Machine Intelligence, IEEE Transactions on, **35**, 1798 (2013).
- [19] G. Hinton, S. Osindero, and Y.-W. Teh, Neural computation, **18**, 1527 (2006).
- [20] R. Salakhutdinov, A. Mnih, and G. Hinton, in *Proceedings of the 24th international conference on Machine learning* (ACM, 2007) pp. 791–798.
- [21] H. Larochelle and Y. Bengio, in *Proceedings of the 25th international conference on Machine learning* (ACM, 2008) pp. 536–543.
- [22] P. Smolensky, (1986).
- [23] Y. W. Teh and G. E. Hinton, Advances in neural information processing systems, 908 (2001).
- [24] G. E. Hinton, Neural computation, **14**, 1771 (2002).