Scale-invariant feature extraction of neural network and renormalization group flow

Satoshi Iso,[1,2] Shotaro Shiba,[1] and Sumito Yokoo[1,2]
[1]*Theory Center, High Energy Accelerator Research Organization (KEK), Tsukuba, Ibaraki 305-0801, Japan*
[2]*Graduate University for Advanced Studies (SOKENDAI), Tsukuba, Ibaraki 305-0801, Japan*

(Received 13 February 2018; published 8 May 2018)

Theoretical understanding of how a deep neural network (DNN) extracts features from input images is still unclear, but it is widely believed that the extraction is performed hierarchically through a process of coarse graining. It reminds us of the basic renormalization group (RG) concept in statistical physics. In order to explore possible relations between DNN and RG, we use the restricted Boltzmann machine (RBM) applied to an Ising model and construct a flow of model parameters (in particular, temperature) generated by the RBM. We show that the unsupervised RBM trained by spin configurations at various temperatures from $T = 0$ to $T = 6$ generates a flow along which the temperature approaches the critical value $T_c = 2.27$. This behavior is the opposite of the typical RG flow of the Ising model. By analyzing various properties of the weight matrices of the trained RBM, we discuss why it flows towards T_c and how the RBM learns to extract features of spin configurations.

DOI: 10.1103/PhysRevE.97.053304

I. INTRODUCTION

Machine learning has attracted interdisciplinary interest as the core method of artificial intelligence, particularly for big data science, and is now widely used to discriminate subtle images by extracting specific features hidden in complicated input data. A deep neural network (DNN), which is motivated by human brains, is one of the well-known algorithms [1]. Despite its enormous successes, it is still unclear why the DNN works so well and how a DNN can efficiently extract specific features. In discriminating images, we first provide samples of input images with assigned labels, such as a cat or a dog, and then train the neural network (NN) so as to correctly predict the labels of new, previously unseen, input images: this is the supervised-learning algorithm, and its ability of prediction depends on how many relevant features the NN can extract. On the other hand, in unsupervised-learning algorithms, a NN is trained without assigning labels to data, but trained so as to generate output images that are as close to the input ones as possible. If the NN is successfully trained to reconstruct the input data, it must have acquired specific features of the input data. With this in mind, unsupervised learnings are often adopted for pretraining of supervised NNs.

How can a DNN efficiently extract features? Specific features characteristic of input data usually have hierarchical structures. An image of a cat can still be identified as an animal in a very low resolution image but one may not be able to distinguish it from a dog. Thus, it is plausible that depth of neural networks reflects such a hierarchy of features. Namely, a DNN learns low-level (microscopic) characteristics in the upper stream of the network and gradually extracts higher-level (macroscopic) characteristics as the input data flow downstream. In other words, the initial data will get coarse grained toward output. This viewpoint is reminiscent of the renormalization group (RG) in statistical physics and quantum field theories, and various thoughts and studies are given [2–9] based on this analogy. In particular, in a seminal paper [4], Mehta and Schwab proposed an explicit mapping between the RG and the restricted Boltzmann machine (RBM) [1,10–13].

The RG is the most important concept and technology to understand the critical phenomena in statistical physics and also plays an essential role to constructively define quantum field theories on a lattice. It is based on the idea (and proved by Wilson [14]) that the long-distant macroscopic behavior of a many body system is universally described by relevant operators (*relevant information*) around a fixed point and not affected by microscopic details in the continuum limit. Through reduction of degrees of freedom in the RG, the relevant information is emphasized while other irrelevant information is discarded. In particular, suppose that the statistical model is described by a set of parameters $\{\lambda_\alpha\}$, and that the parameters are mapped to a different set $\{\lambda'_\alpha\}$ by RG transformations.[1] Repeating such RG transformations, we can draw a flow diagram in the parameter space of the statistical model, $\{\lambda_\alpha\} \to \{\lambda'_\alpha\} \to \{\lambda''_\alpha\} \to \cdots$. These RG flows control the behavior of the statistical model near the critical point where a second order phase transition occurs.

The simplest version of a RBM is a NN consisting of two layers, a visible layer with variables $\{v_i = \pm 1\}$ and a hidden layer with variables $\{h_a = \pm 1\}$, that are coupled to each other through the Hamiltonian

$$\Phi(\{v_i\},\{h_a\}) = -\left(\sum_{i,a} W_{ia} v_i h_a + \sum_i b_i^{(v)} v_i + \sum_a b_a^{(h)} h_a\right). \tag{1}$$

A probability distribution of a configuration $\{v_i, h_a\}$ is given by

$$p(\{v_i\},\{h_a\}) = \frac{1}{\mathcal{Z}} e^{-\Phi(\{v_i\},\{h_a\})}, \tag{2}$$

[1]In order to describe the RG transformation exactly, infinitely many parameters are necessary to be introduced. But it can be usually well approximated by a finite number of parameters.

where we defined the partition function by $\mathcal{Z} = \sum_{\{v_i,h_a\}} e^{-\Phi(\{v_i\},\{h_a\})}$. No intralayer couplings are introduced in the RBM. Now suppose that the RBM is already trained and the parameters of the Hamiltonian in Eq. (1), namely, $\{W_{ia}, b_i^{(v)}, b_a^{(h)}\}$, are already fixed through a process of training. The probability distribution $p(\{v_i\},\{h_a\})$ also provides the following conditional probabilities for $\{h_a\}$ (or $\{v_i\}$) with the other variables being kept fixed:

$$p(\{h_a\}|\{v_i\}) = \frac{p(\{h_a\},\{v_i\})}{\sum_{\{h_a\}} p(\{h_a\},\{v_i\})}, \tag{3}$$

$$p(\{v_i\}|\{h_a\}) = \frac{p(\{h_a\},\{v_i\})}{\sum_{\{v_i\}} p(\{h_a\},\{v_i\})}. \tag{4}$$

As we see in the following, these conditional probabilities generate a flow of distributions, and consequently a flow of parameters $\{\lambda_\alpha\}$ of the corresponding statistical model. Suppose that we have a set of $N(\gg 1)$ initial configurations $\{v_i = \sigma_i^A\}$ $(A = 1,\ldots,N)$, which are generated by a statistical model with parameters $\lambda_\alpha^{(0)}$, such as the Ising model at temperature $T^{(0)}$. In the large-N limit, the distribution function

$$q_0(\{v_i\}) = \frac{1}{N}\sum_A \delta\left(v_i - \sigma_i^A\right) \tag{5}$$

faithfully characterizes the statistical model with parameters $\lambda_\alpha^{(0)}$. Multiplying $q_0(\{v_i\})$ by the conditional probabilities of Eqs. (3) and (4) iteratively, we can generate a flow of probability distributions as

$$q_0(\{v_i\}) \to r_1(\{h_a\}) = \sum_{\{v_i\}} p(\{h_a\}|\{v_i\}) q_0(\{v_i\}), \tag{6}$$

$$r_1(\{h_a\}) \to q_1(\{v_i\}) = \sum_{\{h_a\}} p(\{v_i\}|\{h_a\}) r_1(\{h_a\}), \tag{7}$$

and so on for $q_n(\{v_i\}) \to r_{n+1}(\{h_a\})$ and $r_{n+1}(\{h_a\}) \to q_{n+1}(\{v_i\})$. By combining these transformations, we can generate a flow of the probability distributions $q_0(\{v_i\}) \to q_1(\{v_i\}) \to q_2(\{v_i\}) \to \cdots$. Then, if the probability distribution $q_n(\{v_i\})$ is well approximated by the Boltzmann distribution of the statistical model with different parameters $\lambda_\alpha^{(n)}$, we can say that the RBM generates a flow of parameters[2]

$$\lambda_\alpha^{(0)} \to \lambda_\alpha^{(1)} \to \cdots \to \lambda_\alpha^{(n)} \to \cdots, \tag{8}$$

where n is the number of iterations. We call it the RBM flow of statistical parameters.

Mehta and Schwab [4] pointed out similarity between RG transformations and the above flows of parameters in the unsupervised RBM. But in order to show that the transformation of parameters $\{\lambda_\alpha\}$ in the RBM indeed generates the conventional RG transformation, it is necessary to show that the weight matrix W_{ia} and the biases $b_i^{(v)}, b_a^{(h)}$ of the RBM must be appropriately chosen so as to generate the correct RG transformation that performs coarse-graining of input configurations. In Ref. [4], a multilayer RBM is employed as an unsupervised-learning NN, and the weights and the biases are chosen by minimizing the Kullback-Leibler (KL) divergences (relative entropy) between the input probability distribution and the reconstructed distribution by integrating (marginalizing) over the hidden variables. The authors suggested the similarity by looking at the local spin structures in the hidden variables, but they did not show explicitly that the weights determined by the unsupervised learning actually generate the flow of RG transformations.

The arguments [4] and misconception in the literature are criticized by Ref. [6].[3] In a wider context, the criticism is related to the following question: what determines whether a specific feature of input data is relevant or not? In RG transformations of statistical models, long-wavelength (macroscopic) modes are highly respected while short-wavelength modes are discarded as noise. In this way, RG transformations can extract universal behavior of the model at long wavelength. But, of course, it is so because we are interested in the macroscopic behavior of the system: if we are instead interested in short-wavelength physics, we need to extract opposite features of the model. Thus, we may say that extraction of *relevant features* needs preexisting biases to judge, and supervised learning is necessary to give such biases to the machine. However, this does not mean that unsupervised learnings do not have anything to do with the RG. Even in unsupervised learnings, a NN automatically may notice and extract some kind of features of the input data, and the flow generated by the trained NN will reflect such features.

In the present paper, in order to investigate further the relationship between the RBM and the RG, we train an RBM by using spin configurations of an Ising model and construct a flow of statistical parameters, i.e., temperature T. Here notice that in defining the flow of Eqs. (6) and (7), we need to specify how we have trained the RBM because the training determines the properties of the weights and biases and, accordingly, the behavior of the flow. In this paper we use the following three different trainings. One type of RBM (which we call type V) is trained by configurations at various temperatures from low to high. The other two types (H and L) are trained by configurations only at high and low temperatures. The RBM flow of temperature is constructed for each type of the RBMs:

$$T^{(0)} \to T^{(1)} \to \cdots \to T^{(n)} \to \cdots. \tag{9}$$

In order to measure temperature, we prepare another NN trained by a supervised learning. The main observation in our numerical simulations is that, in the type-V RBM with an appropriate number of neurons, the temperature *approaches the critical point* $T \to T_c$ along the RBM flow, irrespective of the initial temperature $T^{(0)}$ of configurations. The behavior is opposite to the RG flow of the Ising model.

The paper is organized as follows. In Sec. II, we explain the basic settings and the methods of our investigations. We

[2]The situation is similar to Footnote 1, and infinitely many parameters are necessary to represent the probability distribution $p(\{h_a\})$ in terms of the statistical model.

[3]The authors of Ref. [9] gave another counterargument against the claim by Ref. [4]. In the paper, they showed that in order to reproduce the ordinary RG flow, it is necessary to train the RBM by a different method which optimizes the mutual information so that it takes the relationship between a specific region of spin configurations and its surroundings into account.

prepare sample images of the spin configurations of the Ising model and train RBMs by the configurations without assigning labels of temperature. Then we construct flows of parameters (i.e., temperature) generated by the trained RBM.[4] In Sec. III, we show various results of the numerical simulations, including the RBM flows of parameters. In Sec. IV, we analyze properties of the weight matrices W_{ia} using the method of singular value decomposition. The final section is devoted to summary and discussions. Our main results of the RBM flow and conjectures about the feature extractions of the unsupervised RBM are written in Sec. III B.

II. METHODS

In this section we explain various methods for numerical simulations to investigate relations between the unsupervised RBM and the RG of the Ising model. Though most methods in this section are standard and well known, we explain them in detail to make the paper self-contained. In Sec. II C, we explain the central method of generating the RBM flows. Basic materials of the RBM are given in Sec. II B. The other two sections, Secs. II A and II D, can be skipped over unless one is interested in how we generate the initial spin configurations and measure the temperature of a set of configurations.

A. Monte Carlo simulations of Ising model

We first construct samples of configurations of the two-dimensional Ising model by using Monte Carlo simulations. The spin variables $\sigma_{x,y} = \pm 1$ are defined on a two-dimensional lattice of size $L \times L$. The index (x,y) represents each lattice site and takes $x,y = 0,1,\ldots,L-1$. The Ising model Hamiltonian is given by

$$H = -J \sum_{x,y=0}^{L-1} \sigma_{x,y}(\sigma_{x+1,y} + \sigma_{x-1,y} + \sigma_{x,y+1} + \sigma_{x,y-1}). \quad (10)$$

It describes a ferromagnetic model for $J > 0$ and an antiferromagnetic model for $J < 0$. Here we impose the periodic boundary conditions for the spin variables,

$$\sigma_{L,y} := \sigma_{0,y}, \quad \sigma_{-1,y} := \sigma_{L-1,y},$$
$$\sigma_{x,L} := \sigma_{x,0}, \quad \sigma_{x,-1} := \sigma_{x,L-1}. \quad (11)$$

Generations of spin configurations at temperature T are performed by the method of Metropolis Monte Carlo (MMC) simulation. In the method, we first generate a random configuration $\{\sigma_{x,y}\}$. We then choose one of the spins $\sigma_{x,y}$ and flip its spin with the probability

$$p_{x,y} = \begin{cases} 1 & (dE_{x,y} < 0) \\ e^{-dE_{x,y}/k_B T} & (dE_{x,y} > 0), \end{cases} \quad (12)$$

where $dE_{x,y}$ is the change of energy of this system:

$$dE_{x,y} = 2J\sigma_{x,y}(\sigma_{x+1,y} + \sigma_{x-1,y} + \sigma_{x,y+1} + \sigma_{x,y-1}). \quad (13)$$

[4]The two-dimensional Ising model is the simplest statistical model to exhibit the second order phase transition, and there are many previous studies of the Ising model using machine learnings. See, e.g., Refs. [15–20].

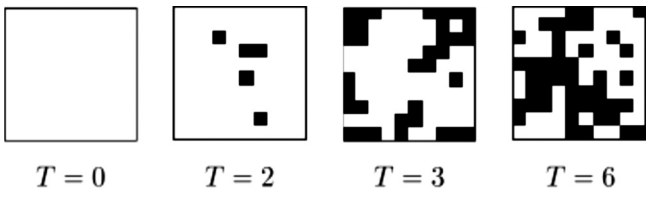

FIG. 1. Examples of spin configurations at temperatures $T = 0,2,3,6$.

The probability of flipping the spin, Eq. (12), satisfies the detailed balance condition $P_{s\to s'}\rho_s = P_{s'\to s}\rho_{s'}$ where $\rho_s \propto e^{-E_s/k_B T}$ is the canonical distribution of the spin configuration $s = \{\sigma_{x,y}\}$ at temperature T. Thus, after many iterations of flipping all the spins, the configuration approaches the equilibrium distribution at T. Since all physical quantities are written in terms of a combination of $J/k_B T$, we can set the Boltzmann constant k_B and the interaction parameter J to be equal to 1 without loss of generality.

In the following analysis, we set the lattice size $L^2 = 10 \times 10$ and repeat the procedure of MMC simulations $100L^2 = 10\,000$ times to construct spin configurations. In our simulations, we generated spin configurations at various temperatures $T = 0,0.25,0.5,\ldots,6$.[5] The number of sample spin configurations to train the RBM is 1000 for each temperature. Some of typical spin configurations are shown in Fig. 1.

B. Unsupervised learning of the RBM

Our main motivation in the present paper is to study whether the RBM is related to the RG in statistical physics. In this section, we review the basic algorithm of the RBM [1,10–13], which is trained by configurations constructed by the MMC method of Sec. II A.

As explained in the Introduction, the RBM consists of two layers as shown Fig. 2(a). The initial configurations $\{\sigma_{x,y}\}$ of the Ising model generated at various temperatures are input into the visible layer $\{v_i\}$. The number of neurons in the visible layer is fixed at $N_v = L^2 = 100$ $(i = 1,\ldots,N_v)$ to represent the spin configurations of the Ising model. On the other hand, the hidden layer can take an arbitrary number of neurons, N_h. In the present paper, we consider seven different sizes: $N_h = 16$, 36, 64, 81, 100, 225, and 400. The N_h spin variables in the hidden layer are given by $\{h_a\}$ for $a = 1,\ldots,N_h$.

The RBM is a generative model of probability distributions based on Eq. (2). We first explain how we can train the RBM by optimizing the weights W_{ia} and the biases $b_i^{(v)}, b_a^{(h)}$. The goal of the training is to represent the given probability distribution $q_0(\{v_i\})$ in Eq. (5), as faithfully as possible, in terms of a model probability distribution defined by

$$p(\{v_i\}) = \frac{1}{\mathcal{Z}} \sum_{\{h_a\}} e^{-\Phi(\{v_i\},\{h_a\})}. \quad (14)$$

[5]For $T = 0$, we practically set $T = 10^{-6}$ for numerical calculations.

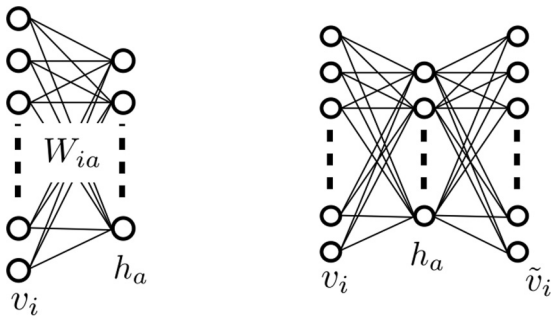

FIG. 2. (a) Two-layer neural network of the RBM with a visible layer $\{v_i\}$ and a hidden layer $\{h_a\}$. These two layers are coupled but there are no intralayer couplings. (b) The RBM generates reconstructed configurations from $\{v_i\}$ to $\{\tilde{v}_i\}$ through the hidden configuration $\{h_a\}$.

The partition function $\mathcal{Z} = \sum_{\{v_i,h_a\}} e^{-\Phi(\{v_i\},\{h_a\})}$ is difficult to evaluate,[6] but summations over only one set of spin variables (e.g., over $\{v_i\}$) are easy to perform because of the absence of the intralayer couplings. It also makes the conditional probabilities in Eqs. (3) and (4) to be rewritten as products of probability distributions of each spin variable:

$$\begin{aligned} p(\{h_a\}|\{v_i\}) &= \prod_a p(h_a|\{v_i\}) \\ &= \prod_a \frac{1}{1+\exp\left[-2h_a\left(\sum_i W_{ia}v_i + b_a^{(h)}\right)\right]}, \end{aligned} \tag{15}$$

$$\begin{aligned} p(\{v_i\}|\{h_a\}) &= \prod_i p(v_i|\{h_a\}) \\ &= \prod_i \frac{1}{1+\exp\left[-2v_i\left(\sum_a W_{ia}h_a + b_i^{(v)}\right)\right]}. \end{aligned} \tag{16}$$

Then the expectation values of spin variables in the hidden (or visible) layer in the background of spin configurations in the other layer are calculated as

$$\langle h_a\rangle_{\{v_i\}} = \tanh\left(\sum_i W_{ia}v_i + b_a^{(h)}\right), \tag{17}$$

$$\langle v_i\rangle_{\{h_a\}} = \tanh\left(\sum_a W_{ia}h_a + b_i^{(v)}\right). \tag{18}$$

Now the task is to train the RBM so as to minimize the distance between two probability distributions of $q(\{v_i\})$ and $p(\{v_i\})$ by appropriately choosing the weights and the biases. The distance is called the KL divergence, or relative entropy, and is given by

$$\begin{aligned} \mathrm{KL}(q \parallel p) &= \sum_{\{v_i\}} q(\{v_i\}) \log \frac{q(\{v_i\})}{p(\{v_i\})} \\ &= \text{const} - \sum_{\{v_i\}} q(\{v_i\}) \log p(\{v_i\}). \end{aligned} \tag{19}$$

If the two probabilities are equal, the KL divergence vanishes. Otherwise, derivatives of $\mathrm{KL}(q \parallel p)$ with respect to the weight W_{ia} and the biases $b_i^{(v)}, b_a^{(h)}$ are given by

$$\begin{aligned} \frac{\partial\,\mathrm{KL}(q \parallel p)}{\partial W_{ia}} &= \langle v_i h_a\rangle_{\text{data}} - \langle v_i h_a\rangle_{\text{model}}, \\ \frac{\partial\,\mathrm{KL}(q \parallel p)}{\partial b_i^{(v)}} &= \langle v_i\rangle_{\text{data}} - \langle v_i\rangle_{\text{model}}, \\ \frac{\partial\,\mathrm{KL}(q \parallel p)}{\partial b_a^{(h)}} &= \langle h_a\rangle_{\text{data}} - \langle h_a\rangle_{\text{model}}, \end{aligned} \tag{20}$$

where averages are defined by

$$\langle A(\{v_i\})\rangle_{\text{data}} = \sum_{\{v_i\}} q(\{v_i\})A(\{v_i\}), \tag{21}$$

$$\langle A(\{v_i\},\{h_a\})\rangle_{\text{model}} = \sum_{\{v_i\},\{h_a\}} p(\{v_i\},\{h_a\})A(\{v_i\},\{h_a\}), \tag{22}$$

and h_a in $\langle\cdots\rangle_{\text{data}}$ is replaced by $\langle h_a\rangle_{\{v_i\}}$ of Eq. (17). In training the RBM, we change the weights and biases so that the KL divergence is reduced. Using the method of maximum likelihood learning, we renew values of the weights and biases as

$$\begin{aligned} W &\to W_{\text{new}} = W + \delta W, \\ b^{(v)} &\to b^{(v)}_{\text{new}} = b^{(v)} + \delta b^{(v)}, \\ b^{(h)} &\to b^{(h)}_{\text{new}} = b^{(h)} + \delta b^{(h)}, \end{aligned} \tag{23}$$

where

$$\begin{aligned} \delta W_{ia} &= \epsilon(\langle v_i h_a\rangle_{\text{data}} - \langle v_i h_a\rangle_{\text{model}}), \\ \delta b_i^{(v)} &= \epsilon(\langle v_i\rangle_{\text{data}} - \langle v_i\rangle_{\text{model}}), \\ \delta b_a^{(h)} &= \epsilon(\langle h_a\rangle_{\text{data}} - \langle h_a\rangle_{\text{model}}). \end{aligned} \tag{24}$$

Here ϵ denotes the learning rate, which we set to 0.1. The first terms $\langle\cdots\rangle_{\text{data}}$ are easy to calculate, but the second terms $\langle\cdots\rangle_{\text{model}}$ are difficult to evaluate since it requires the knowledge of the full partition function $\mathcal{Z}$.

To avoid this difficulty, we need to use the method of Gibbs sampling to approximately evaluate these expectation values $\langle\cdots\rangle_{\text{model}}$. Practically we employ a more simplified method, which is called the method of contrastive divergence (CD) [22–24]. Given the input data of the visible spin configurations $\{v_i^{A(0)} = \sigma_i^A\}$, the expectation value of the hidden spin variable h_a can be easily calculated as Eq. (17). We write the expectation value as

$$h_a^{A(1)} := \langle h_a\rangle_{\{v_i^{A(0)}\}} = \tanh\left(\sum_i W_{ia}v_i^{A(0)} + b_a^{(h)}\right). \tag{25}$$

Then in this background of the hidden spin configurations, the expectation value of v_i can be again easily calculated by using

[6] An efficient way to evaluate the partition function using the mean-field method is proposed in Ref. [21].

Eq. (18). We write it as

$$v_i^{A(1)} := \langle v_i \rangle_{\{h_a^{A(1)}\}} = \tanh\left(\sum_a W_{ia} h_a^{A(1)} + b_i^{(v)}\right). \quad (26)$$

Then we obtain $h_a^{A(2)} = \langle h_a \rangle_{\{v_i^{A(1)}\}}$, and so on. We can iterate this procedure many times and replace the model-dependent terms in Eq. (20) by the expectation values generated by this method.

In doing the numerical simulations in the present paper, we adopt the simplest version of CD, called CD_1. In the CD_1 method, averages over the initial data are given as

$$\begin{aligned} \langle v_i \rangle_{\text{data}} &= \frac{1}{N}\sum_A \sigma_i^A, \\ \langle h_a \rangle_{\text{data}} &= \frac{1}{N}\sum_A h_a^{A(1)}, \\ \langle v_i h_a \rangle_{\text{data}} &= \frac{1}{N}\sum_A \sigma_i^A h_a^{A(1)}, \end{aligned} \quad (27)$$

where σ_i^A denotes input spin configuration generated by the method of Sec. II A. On the other hand, $\langle \cdots \rangle_{\text{model}}$ are replaced by the following approximate formulas:

$$\begin{aligned} \langle v_i \rangle_{\text{model}} &= \frac{1}{N}\sum_A v_i^{A(1)}, \\ \langle h_a \rangle_{\text{model}} &= \frac{1}{N}\sum_A h_a^{A(2)}, \\ \langle v_i h_a \rangle_{\text{model}} &= \frac{1}{N}\sum_A v_i^{A(1)} h_a^{A(2)}. \end{aligned} \quad (28)$$

In the numerical simulations, we generated 1000 spin configurations for each of 25 different temperatures $T = 0, 0.25, \ldots, 6$ for training the (type-V) RBM, in which the index A runs from 1 to $N = 25\,000$. In training other types of RBMs (type H or L), we use only a restricted set of configurations at high or low temperatures. We repeat the renewal procedure of Eq. (23) many times (5000 epochs) and obtain adjusted values of the weights and biases.

The training of the RBM is performed so as to minimize the KL divergence, but it does not mean that the reconstructed configuration becomes exactly the same as the initial one since it is the distance of probability distributions that is minimized. Moreover, the KL divergence itself cannot become zero unless the number of the hidden neurons is infinite.

C. Generation of RBM flows

As discussed in the Introduction, once the RBM is trained and the weights and biases are fixed, the RBM can be used to generate a sequence of probability distributions in Eq. (7). Then we translate it into a flow of parameters (i.e., temperature). In generating the RBM flow, the initial set of configurations should be prepared separately in addition to the configurations that are used to train the RBM.[7]

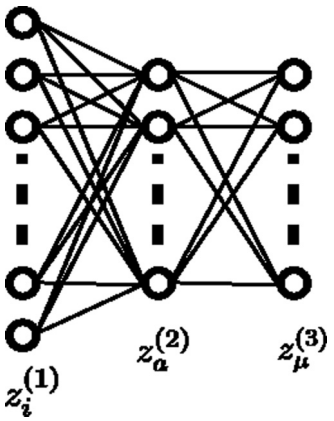

FIG. 3. Three-layer neural network for supervised learning with an input layer $\{z_i^{(1)}\}$, a hidden layer $\{z_a^{(2)}\}$, and an output layer $\{z_\mu^{(3)}\}$.

Let us see more explicitly how we can generate a flow of parameters. For an initial configuration $v_i = v_i^{(0)}$, we can define a sequence of configurations following Eqs. (25) and (26) as

$$\{v_i^{(0)}\} \to \{h_a^{(1)}\} \to \{v_i^{(1)}\} \to \{h_a^{(2)}\} \to \{v_i^{(2)}\} \to \cdots. \quad (29)$$

Fig. 2(b) shows a generation of new configurations from $\{v_i\}$ to $\{\tilde{v}_i\}$ through $\{h_a\}$. Since each value of $v_i^{(n)}$ and $h_a^{(n)}$ (for $n > 0$) is defined by an expectation value as in Eqs. (25) and (26), it does not take an integer ± 1 but a fractional value between ± 1. In order to get a flow of spin configurations, we need to replace these fractional values by ± 1 with a probability $(1 \pm \langle v_i^{(n)} \rangle)/2$ or $(1 \pm \langle h_a^{(n)} \rangle)/2$. It turns out that the replacement is usually a good approximation since the expectation values are likely to take values close to ± 1 owing to the property of the trained weights $|W_{ia}| \gg 1$. In this way, we obtain a flow of spin configurations

$$\{v_i^{(0)}\} \to \{v_i^{(1)}\} \to \{v_i^{(2)}\} \to \cdots \to \{v_i^{(n)}\} \quad (30)$$

starting from the initial configuration $\{v_i^{(0)}\}$. It is transformed to a flow of temperature distributions by using the method explained in Sec. II D.

D. Temperature measurement by a supervised-learning NN

Next we design a NN to measure the temperature of spin configurations. The NN for the supervised learning has three layers with one hidden layer in the middle (see Fig. 3).

The input layer $\{z_i^{(1)}\}$ consists of $L^2 = 100$ neurons in which we input spin configurations of the Ising model. The output layer $\{z_\mu^{(3)}\}$ has 25 neurons which correspond to 25 different temperatures that we want to measure. The number of neurons in the hidden layer $\{z_a^{(2)}\}$ is set to 64. We train this three-layer NN by a set of spin configurations, each of which has a label of temperature. Thus, this is the supervised learning. As input data

[7]Thus we generate 25 000 spin configurations in addition to the 25 000 configurations used for training the RBM.

to train the NN, we use the same $N = 25\,000$ configurations which were used to train the RBM.[8]

The training of the NN is carried out as follows. Denote the input data as

$$Z^{(1)}_{Ai} = \sigma^A_i, \tag{31}$$

where $A = 1, \ldots, N$, and σ^A_i are the spin configurations $\{\sigma_{x,y} = \pm 1\}$ as in Sec. II B. The input data are transformed to $Z^{(2)}_{Aa}$ in the hidden layer by the following nonlinear transformation:

$$Z^{(2)}_{Aa} = f\left(\sum_{i=1}^{100} Z^{(1)}_{Ai} W^{(1)}_{ia} + b^{(1)}_a\right) =: f\left(U^{(1)}_{Aa}\right), \tag{32}$$

where $W^{(1)}_{ia}$ is a weight matrix and $b^{(1)}_a$ is a bias. The activation function $f(x)$ is chosen as $f(x) = \tanh(x)$. $Z^{(2)}_{Aa}$ is transformed to $Z^{(3)}_{A\mu}$ in the output layer, which corresponds to the label, namely, temperature, of each configuration. The output $Z^{(3)}_{A\mu}$ is given by

$$Z^{(3)}_{A\mu} = g\left(\sum_{a=1}^{64} Z^{(2)}_{Aa} W^{(2)}_{a\mu} + b^{(2)}_\mu\right) =: g\left(U^{(2)}_{A\mu}\right), \tag{33}$$

where $W^{(2)}_{a\mu}$ and $b^{(2)}_\mu$ are another weight matrix and bias. The function $g(x)$ is the softmax function

$$g\left(U^{(2)}_{A\mu}\right) = \frac{\exp U^{(2)}_{A\mu}}{\sum_{\nu=1}^{25} \exp U^{(2)}_{A\nu}}, \tag{34}$$

so that $Z^{(3)}_{A\mu}$ can be regarded as a probability since $\sum_\mu Z^{(3)}_{A\mu} = 1$ is satisfied for each configuration A. Thus, the NN transforms an input spin configuration $Z^{(1)}_{Ai}$ to the probability $Z^{(3)}_{A\mu}$ of the configuration to take the μth output value (i.e., temperature).

Each of the input configurations $Z^{(1)}_{Ai}$ is generated by the MMC method at temperature T. T takes one of the 25 discrete values $T = \frac{1}{4}(\nu - 1)$ $(\nu = 1, \ldots, 25)$. If the Ath configuration is labeled by ν, we want the NN to give an output $Z^{(3)}_{A\mu}$ as close as the following one-hot representation:

$$d^{(\nu)}_A = (0, \ldots, 0, \check{1}^{(\nu)}, 0, \ldots, 0)_A, \tag{35}$$

or its μth component is given by $d^{(\nu)}_{A\mu} = \delta_{\mu\nu}$. It can be interpreted as a probability of the configuration A to take the μth output. Then the task of the supervised training is to minimize the cross entropy, which is equivalent to the KL divergence of the desired probability $d^{(\nu)}_{A\mu}$ and the output probability $Z^{(3)}_{A\mu}$. The loss function is thus given by the cross entropy,

$$E_A = \mathrm{KL}\left(d^{(\nu)}_{A\mu} \| Z^{(3)}_{A\mu}\right) = -\sum_\mu d^{(\nu)}_{A\mu} \log Z^{(3)}_{A\mu}. \tag{36}$$

Then, using the method of back propagation [25], we renew values of the weights and biases from the lower to the upper stream:

$$W^{(\ell)} \to W^{(\ell)}_{\mathrm{new}} = W^{(\ell)} + \delta W^{(\ell)},$$
$$b^{(\ell)} \to b^{(\ell)}_{\mathrm{new}} = b^{(\ell)} + \delta b^{(\ell)}. \tag{37}$$

The variations of $\delta W^{(\ell)}, \delta b^{(\ell)}$ at the lower stream are given by

$$\delta W^{(2)}_{a\mu} = -\frac{\epsilon}{N} \sum_A (Z^{(2)})^T_{aA} \Delta^{(3)}_{A\mu},$$
$$\delta b^{(2)}_\mu = -\frac{\epsilon}{N} \sum_A \Delta^{(3)}_{A\mu}, \tag{38}$$

where $\Delta^{(3)}_{A\mu} = Z^{(3)}_{A\mu} - d^{(\nu)}_{A\mu}$. The learning rate ϵ is set to 0.1. Then using these lower stream variations, we change the upper stream weights and biases as

$$\delta W^{(1)}_{ia} = -\frac{\epsilon}{N} \sum_A (Z^{(1)})^T_{iA} \Delta^{(2)}_{Aa},$$
$$\delta b^{(1)}_a = -\frac{\epsilon}{N} \sum_A \Delta^{(2)}_{Aa}, \tag{39}$$

where

$$\Delta^{(2)}_{Aa} = \sum_\mu \Delta^{(3)}_{A\mu} (W^{(2)})^T_{\mu a} f'\left(U^{(1)}_{Aa}\right). \tag{40}$$

We repeat this renewal procedure many times (7500 epochs) for the training of the NN to obtain suitably adjusted values of the weights and biases.

Finally we note how we measure temperature of a configuration. If the size of a configuration generated at temperature T is large enough, say, $L = 10^{10^{100}}$, the trained NN will reproduce the temperature of the configuration quite faithfully. However, our configurations are small sized with only $L = 10$. Thus, we instead need an ensemble of many spin configurations and measure a temperature distribution of the configurations. The supervised learning gives us this probability distribution of temperature.

III. NUMERICAL RESULTS

In this section we present our numerical results for the flows generated by the unsupervised RBM and discuss a relation with the renormalization group flow of the Ising model. Our main results of the RBM flows are written in Sec. III B.

A. Supervised learning for temperature measurement

Before discussing the unsupervised RBM, let us first see how we trained the NN to measure temperature.

In Fig. 4, we plot behaviors of the loss function in Eq. (36) as we iterate renewals of the weights and biases, Eq. (37). The lower line shows the training error, namely, values of the loss function Eq. (36) after iterations of training using 25 000 configurations. It is continuously decreasing, even after 7500 epochs. On the other hand, the upper line shows the test error, namely, values of the loss function for an additional 25 000 configurations which are not used for the training. This is also decreasing at first, but after 6000 epochs it becomes almost constant. After 7500 epochs, in fact, it begins to increase. This means the machine becomes overtrained; therefore, we stopped the learning at 7500 epochs.

[8] In order to check the performance of the NN, namely, to see how precisely the machine can measure the temperature of a new set of configurations, we use another 25 000 configurations that are prepared for generating the sequence of probability distributions of the RBM in Sec. II C. We show the results of the performance in Sec. III A.

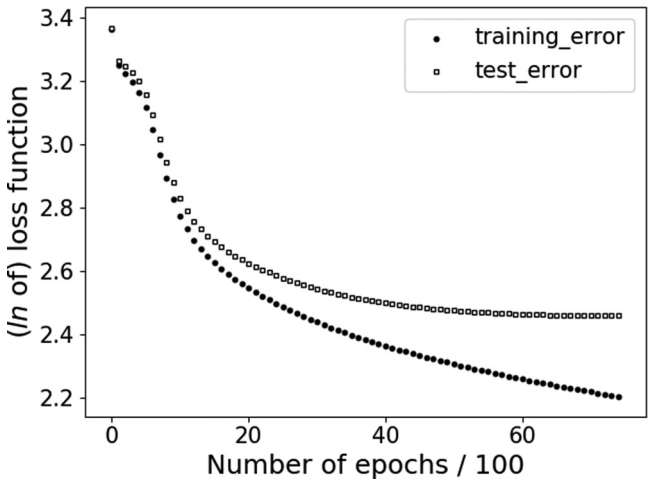

FIG. 4. Training error and test error (up to 7500 epochs).

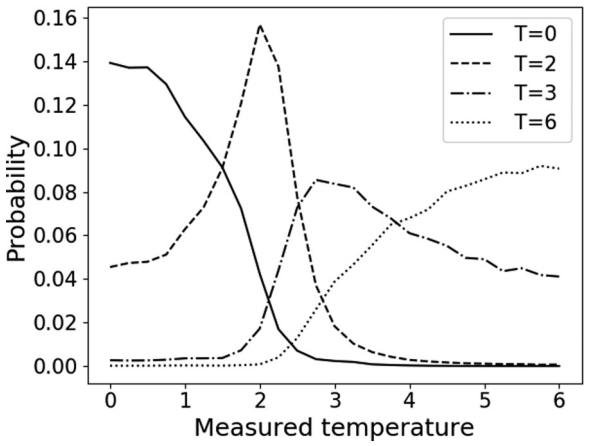

FIG. 5. Probability distributions of measured temperatures for various sets of configurations generated at $T = 0$, 2, 3, and 6. Temperature of the configurations can be distinguished by looking at the shapes of the distributions.

In Fig. 5 we show the probability distributions of temperature this NN measures. Here we use configurations at $T = 0,2,3,6$ which are not used for the training. Though they are not sharply peaked at the temperatures where the configurations are generated,[9] each of them has characteristic shape that is different temperature by temperature. Thus, it is possible to distinguish the temperature of the input configurations by looking at the shape of the probability distribution.[10]

In the following, by using this NN, we measure the temperature of configurations that are generated by the RBM flow.

B. Unsupervised RBM flows

Now we present the main results of our numerical simulations, namely, the flows generated by the unsupervised RBM. As discussed in the Introduction, if the RBM is similar to the ordinary RG in that it possesses a function of coarse graining, the RBM flow must go away from the critical point $T_c = 2.27$. In order to check it, using the method of Sec. II B, we construct three different types of unsupervised RBMs, which we call type V, type L, and type H, respectively. Each of them is trained by a different set of spin configurations generated at different sets of temperatures. We then, following the methods of Secs. II C and II D, generate flows of temperature distributions by using these trained RBMs.

1. Type-V RBM: Trained by various configurations at $T = \{0,0.25,0.5,\ldots,6\}$

First we construct the type-V RBM, which is trained by configurations at temperatures ranging widely from low to high, $T = 0,0.25,\ldots,6$. The temperature range includes the temperature $T = 2.25$ near T_c. After training is completed, this unsupervised RBM will have learned features of spin configurations at these temperatures.

Once the training is finished, we then generate a sequence of reconstructed configurations as in Eq. (30) using the methods in Sec. II C. For this, we prepare two different sets of initial configurations. One is a set of configurations at $T = 0$, and another at $T = 6$. These initial configurations are not used for the training of the RBM. Then by using the supervised NN in Sec. III A, we measure temperature and translate the flow of configurations to a flow of temperature distributions.

In Figs. 6 and 7, we plot temperature distributions of configurations that are generated by iterating the RBM reconstruction in Sec. II C. The "itr" in the legends means the numbers of iterations n by the unsupervised RBM. Fig. 6 shows a flow of temperature distributions starting from spin configurations generated at $T = 0$. Fig. 7 starts from $T = 6$. In all of the figures, the solid lines are the measured temperature distributions of the initial configurations.[11] The other lines show temperature distributions of the reconstructed configurations $\{v_i^{(n)}\}$ after various numbers of iterations. Figs. 6(a) and 7(a) show the temperature distributions at small iterations (up to 10 in Fig. 6 and 50 in Fig. 7), while Figs. 6(b) and 7(b) are at larger iterations up to 1000. These results indicate that the critical temperature T_c is a stable fixed point of the flows in type-V RBM. It is apparently different from a naive expectation that the RBM flow should show the same behavior as the RG

[9] There are two reasons for this broadening of the distributions. One is due to the finiteness of the size of a configuration $N = L \times L = 10 \times 10$. Another is due to the limit of measuring temperature by the NN. If the size of a configuration was infinite and if the ability of discriminating subtle differences of different temperature configurations was limitless, we would have obtained a very sharp peak at the labeled temperature.

[10] The sharpness of distribution is different for different temperatures, and accuracy is worse for very low or very high temperatures. This is because our lattice has IR and UV cutoffs. Namely, for low temperature, most spin configurations are almost uniform below $T < 1$ and difficult to distinguish from each other. If lattice size is bigger than 10×10, low-temperature configurations will be more sharply distinguishable. For high temperature, the correlation length becomes shorter than the unit lattice length and again spin configurations are hard to distinguish above $T > 3$.

[11] As discussed in Footnote 9, these distributions are not sharply peaked at the temperature at which the configurations are generated.

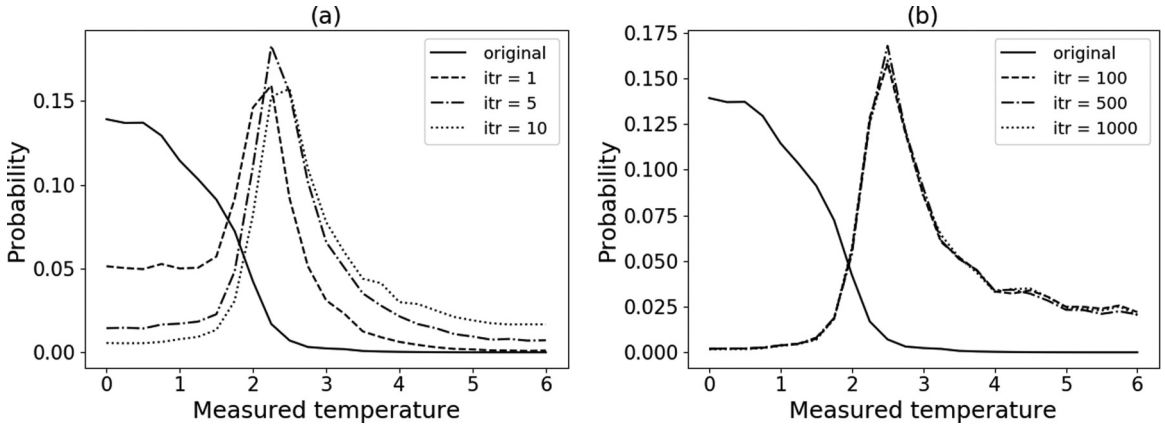

FIG. 6. Temperature distributions after various numbers of iterations of type-V RBM, which is trained by the configurations at $T = 0,0.25,\ldots,6$. The original configurations are generated at $T = 0$. After only several iterations, the temperature distribution peaks around T_c, and stabilizes there: T_c is a stable fixed point of the flow.

flow. Indeed it is in the opposite direction. From whichever temperature $T = 0$ or $T = 6$ we start the RBM iteration, the peak of the temperature distributions approaches the critical point ($T = 2.27$). A reason for this behavior is discussed in Summaries and Conjectures at the end of this section.

In order to further confirm the above behavior, we provide another set of configurations at $T = 2.25$ as initial configurations and generate the flow of temperature by using the same trained RBM. The flow of temperature distributions is shown in Fig. 8. We can see that the temperature distribution of the reconstructed configurations remains near the critical point and never flows away from there.[12] If the process of the unsupervised RBM corresponds to coarse graining of spin configurations, the temperature distributions of the reconstructed configurations must flow away from T_c. Though the direction of the flow is opposite to the RG flow, both flows have the same property in that the critical point $T = T_c$ plays an important role in controlling the flows.

So far, in obtaining the above results of Figs. 6–8, we used an unsupervised RBM with 64 neurons in the hidden layer. We also trained other RBMs with different sizes of the hidden layer ($N_h = 16,36,81,100,225,400$), but by the same set of spin configurations. When the size of the hidden layer is smaller than (or equal to) that of the visible layer $N_v = 100$, namely, $N_h = 100$, 81, 64, 36, or 16, we find that the temperature distribution approaches the critical point. A difference is that for smaller N_h, the speed of the flow to approach T_c becomes faster (i.e., the flow arrives at T_c by smaller numbers of iterations).

In contrast, when the RBM has more than 100 neurons in the hidden layer, $N_h > N_v$, we obtain different results. Fig. 9 shows the case of $N_h = 225$ neurons. Until about ten iterations, the measured temperature distribution behaves similarly to the case of $N_h \leqslant 100$; i.e., it approaches the critical temperature. However, afterward it passes the critical point and flows away to higher temperature. In the case of 400 neurons, it moves toward high temperature at faster speed. This behavior suggests that, if the hidden layer has more than a "necessary" size, the NN tends to learn a lot of noisy fluctuations specific to higher temperature configurations. We come back to this conjecture in later sections.

2. Type-H or -L RBM: Trained by configurations at higher or lower temperatures

Next we construct another type of RBM, which is trained by configurations at higher temperatures $T = 4,4.25,\ldots,6$ than $T_c \sim 2.25$. We call it type-H RBM. The results of the flows of temperature distributions in type-H RBM are drawn in Fig. 10. In this case, the measured temperature passes the critical point and goes away toward higher temperature. The behavior is understandable since the RBM must have learned only the features at higher temperatures. We also find that, if the number of neurons in the hidden layer is increased, the flow moves more slowly.

Finally, we construct type-L RBM, which is trained by configurations only at the lowest temperature $T = 0$. Fig. 11 shows the numerical results of flows in the type-L RBM. Similarly to the type-H RBM, the measured temperature passes the critical point, but flows toward lower temperature instead of higher temperature. It is, of course, as expected because the type-L RBM must have learned the features of spin configurations at $T = 0$. In the type-L RBM, as far as we have studied, the flow never goes back to higher temperature even for large N_h. It will be because the $T = 0$ configurations used for training do not contain any noisy fluctuations specific to high temperatures. This also suggests that the RBM does not learn features that are not contained in the configurations used for the trainings.

3. Summaries and conjectures

Here we summarize the numerical results. For the type-V RBM,

(i) When $N_h \leqslant 100 = N_v$, the measured temperature T approaches T_c (Figs. 6–8).

[12] We also trained the RBM using configurations of a wider range of temperatures: $T = 0,0.25,\ldots,10,\infty$. The results are very similar, and the temperature distributions of reconstructed configurations always approach the critical point $T_c = 2.27$.

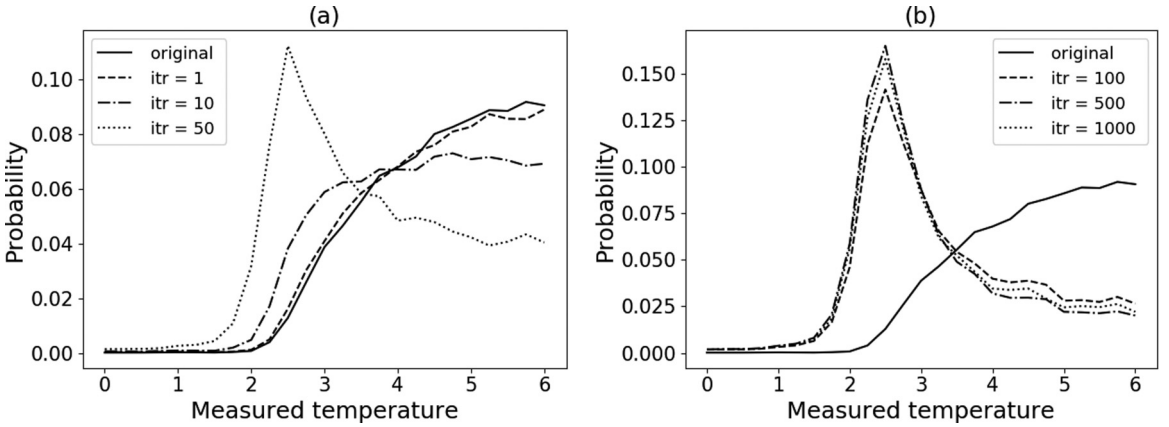

FIG. 7. Temperature distributions after various numbers of iterations of the same RBM as Fig. 6. The original configurations are generated at $T = 6$. After ∼50 iterations, the distribution stabilizes at T_c.

(ii) However, for $N_h > 100 = N_v$, the flow eventually goes away toward $T = \infty$ (Fig. 9).

(iii) The speed of flow is slower for a larger N_h.

For the type-H or -L RBM,

(i) The temperature T flows toward $T = \infty / T = 0$, respectively (Figs. 10 and 11).

(ii) The speed of flow is slower for a larger N_h.

These behaviors are, of course, controlled by the properties of the weights and biases that the unsupervised RBMs have learned in the process of trainings.

Understanding the above behaviors is nothing but answering the question of what the unsupervised RBMs have learned in the process of training. The most important question will be why the temperature approaches T_c in the type-V RBM with $N_h \leqslant N_v$, instead of, e.g., broadening over the whole region of temperature from $T = 0$ to $T = 6$. Note that we did not teach the NN about the critical temperature or the presence of phase transition. We just have trained the NN by configurations at various temperatures, from $T = 0$ to $T = 6$. Nevertheless, the numerical simulations show that the temperature distributions are peaked at T_c after some iterations of the RBM reconstruction. Thus, we are forced to conclude that the RBM has automatically learned *features* specific to the critical temperature T_c.

An important feature at T_c is the scale invariance. We have generated spin configurations at various temperatures by the Monte Carlo method, and each configuration has typical fluctuations specific to each temperature. At very high temperature, fluctuations are almost random at each lattice site and there are no correlations between spins at distant positions. At lower temperature, they become correlated: the correlation length becomes larger as $T \to T_c$ and diverges at T_c. On the other hand, at $T \ll T_c$, spins are clustered and in each domain all spins take $\sigma_{x,y} = +1$ or -1. At low temperature configurations have only big clusters, and as temperature increases small-sized clusters appear. At T_c, spin configurations begin to have clusters of various sizes in a scale-invariant way.

Now let us come back to the question of why the type-V RBM generates a flow approaching T_c and does not broaden the temperature distribution over the whole region. We have trained the type-V RBM by using configurations at various temperatures with different-sized clusters, and in the process the machine must have simultaneously acquired features at

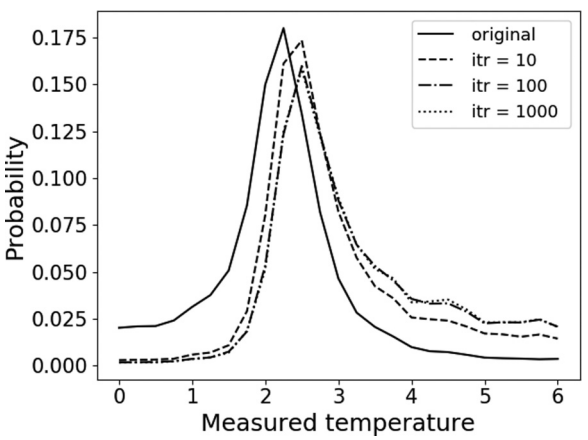

FIG. 8. Temperature distributions after various numbers of iterations of the same RBM as Figs. 6 and 7. The original configurations are generated at $T = 2.25$. The distribution is stable at around T_c.

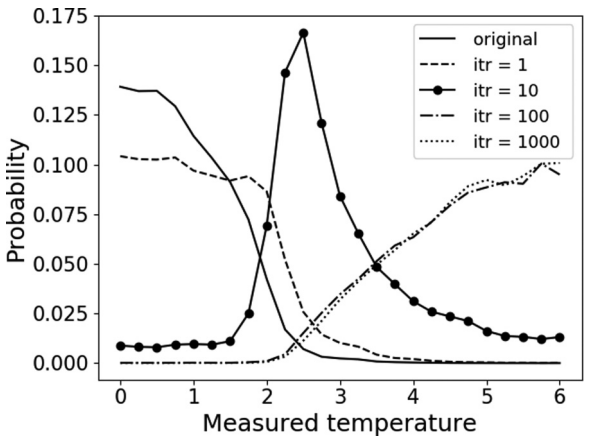

FIG. 9. Temperature distribution after various numbers of iterations of type-V RBM with 225 neurons in the hidden layer; i.e., $N_h > N_v$. The original configurations are generated at $T = 0$. The distribution has a peak at $T = T_c$ after ten iterations, but then moves toward $T = \infty$.

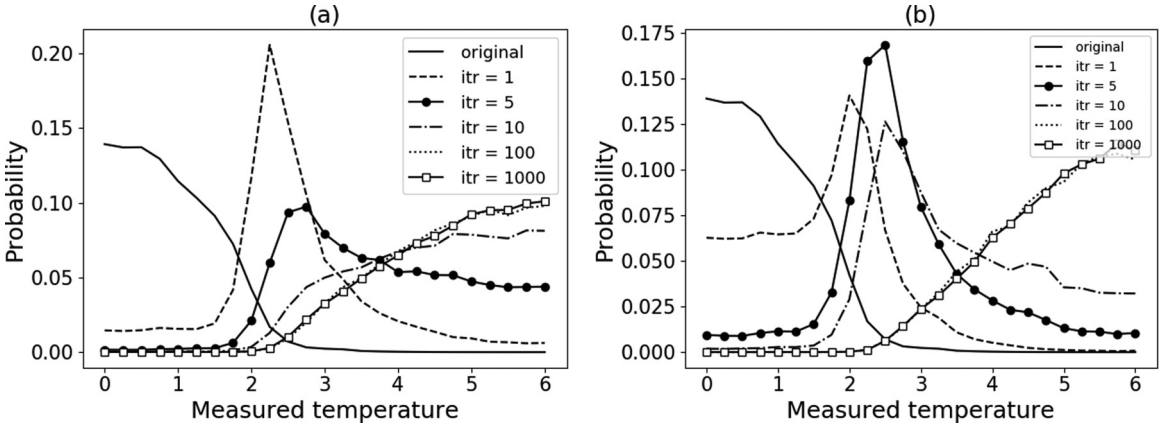

FIG. 10. Flow of temperature distributions starting from $T = 0$ in type-H RBM that is trained by configurations at only $T = 4, 4.25, \ldots, 6$. The NN has (a) $N_h = 64$ neurons and (b) $N_h = 225$ neurons, respectively, in the hidden layer. Both flows move to higher temperature, and the speed is slower for the larger N_h.

various temperatures. Consequently the process of the RBM reconstruction adds various features that the machine has learned to a reconstructed configuration. If only a single feature at a specific temperature was added to the reconstructed configuration, the distribution would have a peak at this temperature, just as in the type-L RBM. But in the type-V RBM it cannot happen because various features of different temperatures will be added to a single configuration by iterations of reconstruction processes. Then one may ask if there is a configuration that is stable under additions of features at various different T.

Our first conjecture about this question is that a set of configurations at T_c is a stabilizer (and even more an attractor) of the type-V RBM with $N_h \leqslant N_v$. It must be due to the scale-invariant properties of the configurations at T_c. Namely since these configurations are scale invariant, they have all the *features* of various temperatures simultaneously, and consequently they can be the stabilizer of this RBM. This sounds plausible since the scale invariance means that the configurations have various different characteristic length scales. However, we notice that this does not mean that the RBM has forgotten the features of configurations away from the critical point. Rather it means that the RBM has learned features of all temperatures simultaneously. This does not mean either that the configurations at $T = T_c$ have had a particularly strong influence on the machine in the process of training. It can be confirmed as follows. Suppose we have trained an RBM by configurations at temperatures excluding $T = T_c$, e.g., trained it by configurations at all temperatures *except* $T = 2.25$ and 2.5. We found in the numerical simulations that such an RBM also generates a flow toward the critical point though we did not provide configurations at $T = T_c$ for the process of training.[13] Therefore, we can say that the type-V RBM has learned the features at all the temperatures and that configurations at T_c are special because they contain all the features of various temperatures in the configurations.

Our second conjecture, which is related to the behavior of the type-V RBM with $N_h > N_v$, is that RBMs with unnecessarily large-sized hidden layers tend to learn lots of

[13]Even when an RBM has learned configurations only at temperatures excluding those around T_c (e.g., $T = 0, \ldots, 1$ and $4, \ldots, 6$), the flow again approaches $T = T_c$.

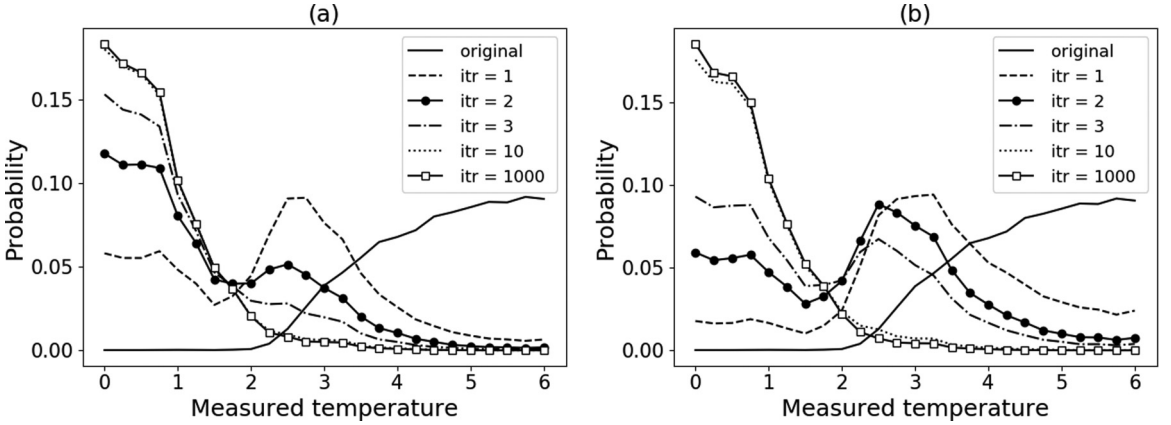

FIG. 11. Flow of temperature distributions starting from $T = 6$ in type-L RBM. Type-L RBM is trained by configurations at only $T = 0$. (a) $N_h = 64$ and (b) $N_h = 225$.

irrelevant features.[14] In the present case, they are noisy fluctuations of configurations at high temperatures. High-temperature configurations have only short distance correlations, whose behavior is similar to the typical behavior of noise. The conjecture will be partially supported by the similarity of the RBM flows between the type-V RBM with $N_h > N_v$ and the type-H RBM. Namely, both RBM flows converge on $T = \infty$. The similarity indicates that the NN with a larger number N_h may have learned too many noiselike features of configurations at higher temperatures. The above considerations will teach us that the moderate size of the hidden layer, $N_h < N_v$, is the most efficient to properly extract the features of input configurations.

IV. ANALYSIS OF THE WEIGHT MATRIX

In the previous section, we showed our numerical results for the flows generated by unsupervised RBMs, and proposed two conjectures. One is that the scale-invariant $T = T_c$ configurations are stabilizers (and attractors) of the type-V RBM flow. Another conjecture is that the RBM with an unnecessarily large-sized hidden layer $N_h > N_v$ tends to learn too many irrelevant noises. In this section, to further understand the theoretical basis of the feature extractions and to give supporting evidence for our conjectures, we analyze various properties of the weight matrices and biases of the trained RBMs. In particular, we study properties of WW^T by looking at spin correlations in Sec. IV B, magnetization in Sec. IV C, and the eigenvalue spectrum in Sec. IV D.

A. Why WW^T is important

All the information that the machine has learned is contained in the weights W_{ia} and the biases $b_i^{(v)}$,$b_a^{(h)}$. Since the biases take smaller values than the weights (at least in the present situations), we concentrate on the weight matrix W_{ia} ($i = 1, \ldots, N_v = L^2$, $a = 1, \ldots, N_h$) in the following.

Let us first note that the weight matrix W_{ia} transforms as

$$W_{ia} \to \tilde{W}_{ia} = \sum_{j,b} U_{ij} W_{ib} (V^T)_{ba} \tag{41}$$

under transformations[15] of exchanging the basis of neurons in the visible layer (U_{ij}) and in the hidden layer (V_{ab}). Since the choice of basis in the hidden layer is arbitrary, relevant information in the visible layer is stored in a combination of W_{ia} that is invariant under transformations of V_{ab}. The simplest combination is a product:

$$(WW^T)_{ij} = \sum_a W_{ia} W_{aj}. \tag{42}$$

It is an $N_v \times N_v = 100 \times 100$ matrix, and independent of the size of N_h. But its property depends on N_h because the rank of WW^T must be always smaller than $\min(N_v, N_h)$. Thus, if $N_h < N_v$, the weight matrix is strongly constrained; e.g., a unit matrix $WW^T = 1$ is not allowed.

The product (42) plays an important role in the dynamics of the flow generated by the RBM. It can be shown as follows. If the biases are ignored, the conditional probability (15) and the expectation value (17) for h_a in the background of v_i become

$$p(\{h_a\}|\{v_i\}) = \frac{e^{\sum_i v_i W_{ia} h_a}}{2\cosh\left(\sum_i v_i W_{ia}\right)},$$
$$\langle h_a \rangle = \tanh\left(\sum_i v_i W_{ia}\right). \tag{43}$$

In $p(\{h_a\}|\{v_i\})$, a combination $\sum_i v_i W_{ia} =: B_a$ can be regarded as an external magnetic field for h_a. Thus, these two variables, B_a and h_a, tend to correlate with each other. Namely, the probability $p(\{h_a\}|\{v_i\})$ becomes larger when they have the same sign. Moreover, for $|B_a| < 1$, $\langle h_a \rangle$ is approximated by B_a and we can roughly identify these two variables:

$$\langle h_a \rangle \sim B_a := \sum_i v_i W_{ia}. \tag{44}$$

It is usually not a good approximation since weights can have larger values. For a large value of $|B_a| \gg 1$, $\langle h_a \rangle$ is saturated at $\langle h_a \rangle = B_a/|B_a|$, which is further strongly correlated to B_a. For simplicity let us assume Eq. (44) for the moment.

Suppose that the input configuration is given by $\{v_i^{(0)}\} = \{\sigma_i^A\}$. If Eq. (44) is employed, we have $h_a^{(1)} = B_a^{(0)} = \sum_i v_i^{(0)} W_{ia}$. Then the conditional probability (16) in the background of $h_a^{(1)}$ with $b_a^{(h)} = 0$,

$$p\left(\{v_i\}\middle|\{h_a^{(1)}\}\right) = \prod_i \frac{e^{\sum_a v_i W_{ia} h_a^{(1)}}}{2\cosh\left(\sum_a W_{ia} h_a^{(1)}\right)}, \tag{45}$$

can be approximated as [27,28]

$$p\left(\{v_i\}\middle|\{h_a^{(1)}\}\right) \sim \prod_i \frac{e^{\sum_a v_i W_{ia} \sum_j W_{ja} v_j^{(0)}}}{2\cosh\left(\sum_a W_{ia} \sum_j W_{ja} v_j^{(0)}\right)}$$
$$= \prod_i \frac{e^{\sum_j v_i (WW^T)_{ij} v_j^{(0)}}}{2\cosh\left(\sum_j (WW^T)_{ij} v_j^{(0)}\right)}. \tag{46}$$

The RBM learns the input data $\{v_j^{(0)}\}$ so that the probability distribution p reproduces the probability distribution of the initial data, $q(\{v_i\}) = \frac{1}{N}\sum_A \delta(v_i - \sigma_i^A)$. Therefore, training of the RBM will be performed so as to enhance the value $\sum_{A,i,j} \sigma_i^{A(0)} (WW^T)_{ij} \sigma_j^{A(0)}$. This means that W is chosen so that $(WW^T)_{ij}$ will reflect the spin correlations of the input configurations $\{\sigma_i^A\}$ at site i and j. The tendency is not spoiled by a large value of $|B_a|$.

In this simplified discussion, learning of the RBM is performed through the combination WW^T. Of course, we neglected the nonlinear property of the neural network and the above statement cannot be justified as it is. Nevertheless, we find below that the analysis of WW^T is quite useful to understand how the RBM works.

[14] Ref. [26] points out that a similar phenomenon occurs in the deep belief network.

[15] Since spin variables on each lattice site are restricted to take values ± 1, the matrices, U_{ij} and V_{ab}, are elements of the symmetric group, not the orthogonal Lie group.

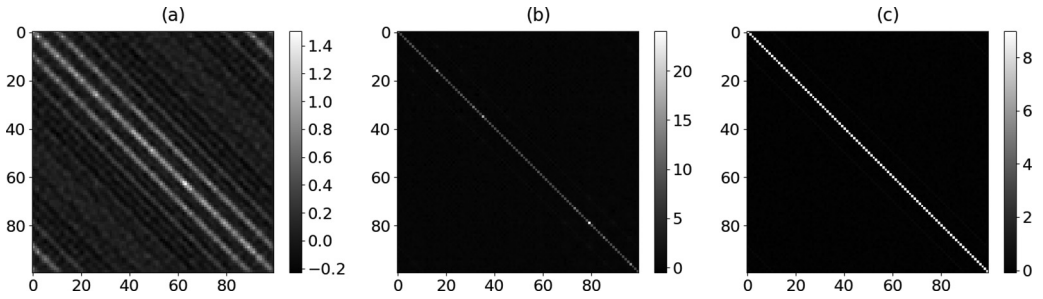

FIG. 12. Elements of WW^T when the hidden layer has (a) 16, (b) 100, and (c) 400 neurons.

B. Spin correlations in WW^T

In Fig. 12, we plot values of matrix elements of the 100×100 matrix WW^T. These three figures correspond to the RBMs with different sizes of N_h. We can see that they have large values in the diagonal and near diagonal elements. Note that the spin variables in the visible layer, $\sigma_{x,y}$ with $x,y=1,\ldots,L=10$, are lined up as $(\sigma_{1,1},\sigma_{1,2},\ldots,\sigma_{1,L},\sigma_{2,1},\ldots,\sigma_{2,L},\sigma_{3,1},\ldots,\sigma_{L,L})$, and named $(\sigma_1,\sigma_2,\ldots,\sigma_N)$. Hence, lattice points i and j of σ_i $(i=1,\ldots,L^2)$ are adjacent to each other when $j=i\pm1$ and $j=i\pm L$. In the following, we mostly discuss the type-V RBM unless otherwise stated.

As discussed above, the product of weight matrices WW^T must reflect correlations between spin variables of the input configurations used for the training of the RBM. The strongest correlation in $v_i^{(0)}(WW^T)_{ij}v_j^{(0)}$ is of course the diagonal component, $i=j$. Thus, we expect that the matrix WW^T will have large diagonal components. Indeed, such behavior can be seen in Fig. 12. In particular, for $N_h=400>N_v=100$ [Fig. 12(c)], WW^T is clearly close to a diagonal matrix. It is almost true for the case of $N_h=100=N_v$ [Fig. 12(b)]. However, for $N_h=16<N_v=100$ [Fig. 12(a)], it is different from a unit matrix, and off-diagonal components of $(WW^T)_{ij}$ also have large values, particularly at $j=i+1$ and $j=i+2$.

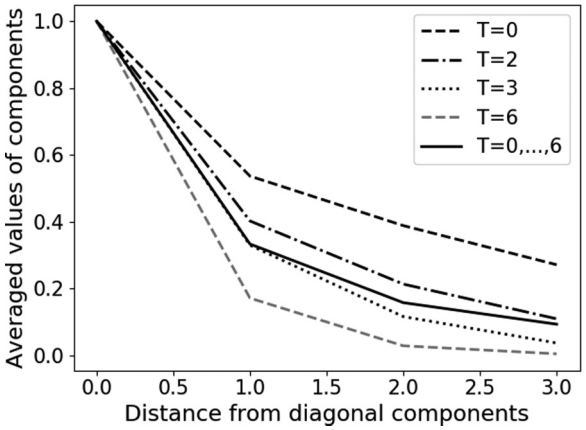

FIG. 13. Averaged values of the off-diagonal components of WW^T (normalized by the diagonal components). The solid line (the most middle line) is the behavior of the type-V RBM that has learned all the temperatures $T=0,\ldots,6$. Each of the other lines corresponds to the RBM that has learned configurations at a single temperature $T=0$, 2, 3, or 6, respectively.

This behavior must be a reflection of the spin correlations of the input configurations.[16] It is also a reflection of the fact that the rank of WW^T is smaller than N_h and WW^T cannot be a unit matrix if $N_h<N_v$. Thus, even though only less information can be stored in the weight matrix for a smaller number of hidden neurons, the relevant information of the spin correlations is well encoded in the weight matrix of the RBM with $N_h<N_v$ compared with the RBM with larger N_h. Then we wonder why such relevant information is lost in the RBM with $N_h>N_v$. This question might be related to our second conjecture proposed at the end of Sec. III B that the RBM with very large N_h will learn too much irrelevant information, namely, noises of the input configurations. It is interesting and a bit surprising that the RBM with fewer hidden neurons seems to learn more efficiently the relevant information of the spin correlations.

In order to further confirm the relation between the correlations in the combination of the weight matrix WW^T and the spin correlations of the input configurations, we study structures of the weight matrices of other types of RBMs. In Fig. 13, we plot behaviors of the off-diagonal components of WW^T for various RBMs. Each RBM is trained by configurations at a single temperature $T=0$ (type L), $T=2$, $T=3$, and $T=6$ respectively. The size of the hidden layer is set to $N_h=16$. For comparison, we also plot the behavior of the off-diagonal components for the type V RBM.

Fig. 13 shows that the correlation of WW^T decays more rapidly at higher temperature, which is consistent with the expected behavior of spin correlations. Therefore, the RBM seems to have learned correctly about the correlation length, or the size of clusters, which becomes smaller at higher temperature. Furthermore, we find that, for the type-V RBM that has learned all temperatures $T=0,\ldots,6$, the off-diagonal elements decrease with the decay rate between the $T=2$ case and the $T=3$ case. This indicates that the type-V RBM has acquired similar features to those of the configurations around $T_c=2.27$. It is consistent with the numerical results of Figs. 6–8, and gives another circumstantial evidence supporting the first conjecture in Sec. III B.

[16]Off-diagonal components of $j=i+L$ or $j=i+2L$ are also large, which corresponds to correlations along the y direction. Large off-diagonal components at $j=i+1$ and $j=i+2$ mean correlations along the x direction.

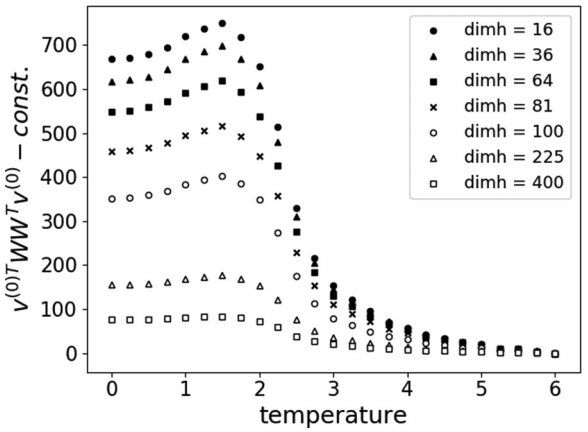

FIG. 14. Averaged values of $v^{(0)T}WW^Tv^{(0)}$ over the 1000 input configurations at each temperature. Different lines correspond to type-V RBMs with different numbers of hidden neurons N_h. In this figure, the values at $T = 6$ are subtracted for comparison between different RBMs.

C. Magnetization and singular value decomposition (SVD)

Information of the weight matrix W can be inferred by using the method of the singular value decomposition.[17] Suppose that the matrix WW^T has eigenvalues λ_a ($a = 1, \ldots, N_v$) with corresponding eigenvectors u_a:

$$WW^T u_a = \lambda_a u_a. \tag{47}$$

Decomposing an input configuration vector $v^{(0)}$ in terms of the eigenvectors u_a as $v^{(0)} = \sum_a c_a u_a$ with a normalization condition $\sum_a (c_a)^2 = 1$, we can rewrite $v^{(0)T}WW^Tv^{(0)}$ as

$$v^{(0)T}WW^Tv^{(0)} = \sum_a c_a^2 \lambda_a. \tag{48}$$

[17]In Refs. [29–31] and references therein, Matsueda found various interesting properties of snapshot images, including Ising spin configurations, based on the method of SVD of image data. He especially found that the snapshot entropy changes drastically at $T = T_c$.

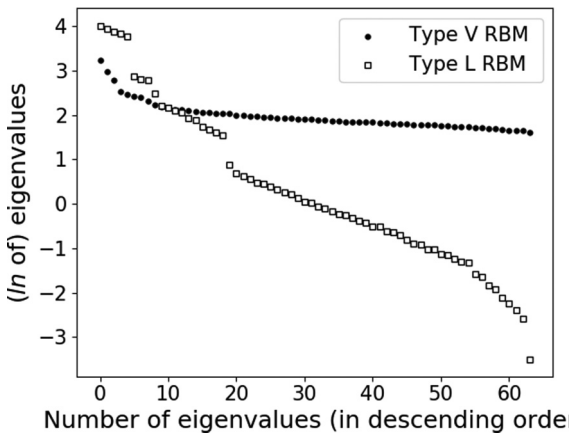

FIG. 16. Eigenvalues of WW^T for type-V RBM (a smooth line) and type-L RBM (a steplike line). Both RBMs have 64 neurons in the hidden layer.

Thus, if a vector $v^{(0)}$ contains more components with larger eigenvalues of WW^T, the quantity $v^{(0)T}WW^Tv^{(0)}$ becomes larger.

Fig. 14 shows averaged values of $v^{(0)T}WW^Tv^{(0)}$ over the 1000 configurations $\{v^{(0)}\}$ at each temperature. For comparison between different RBMs, we subtracted the values at $T = 6$. The figure shows a big change near the critical point, which is reminiscent of the magnetization of the Ising model. Since $v^{(0)T}WW^Tv^{(0)}$ should contain more information than the magnetization itself, the behavior cannot be exactly the same. But it is quite intriguing that Fig. 14 shows similar behavior to the magnetization.[18] It might be because the quantity contains much information about the lower temperature after subtraction of the values at higher temperature.[19]

[18]The behavior indicates that the principal eigenvectors with large eigenvalues might be related to the magnetization, and information about the phase transition is surely imported in the weight matrix. Thus, we investigated properties of the eigenvectors but so far we have not gotten any physically reasonable pictures. We want to come back to this problem in future work.

[19]It suggests that the subtraction may correspond to removing the contributions of the specific features at higher temperature.

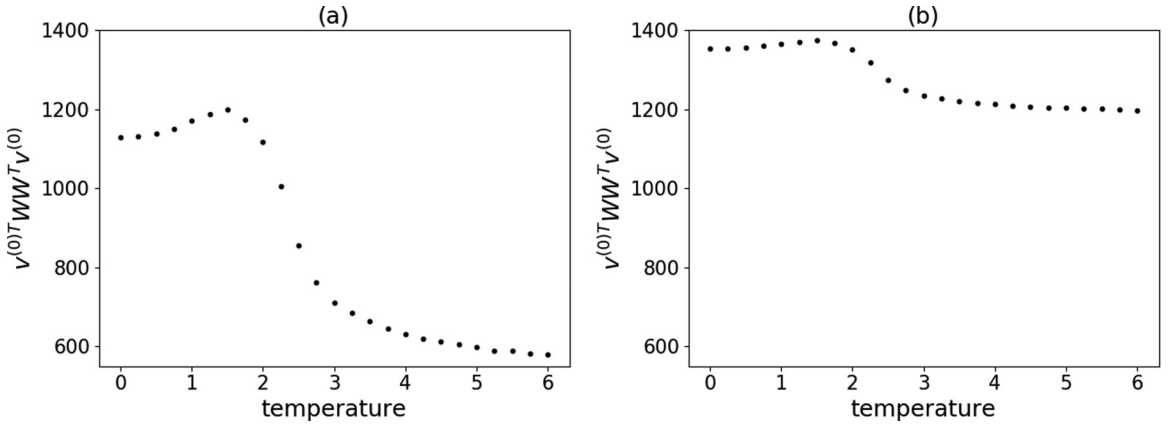

FIG. 15. Averaged values of $v^{(0)T}WW^Tv^{(0)}$ over the 1000 configurations $\{v^{(0)}\}$ at each temperature. Quantities shown for the type-V RBM with (a) $N_h = 64$ and (b) $N_h = 225$.

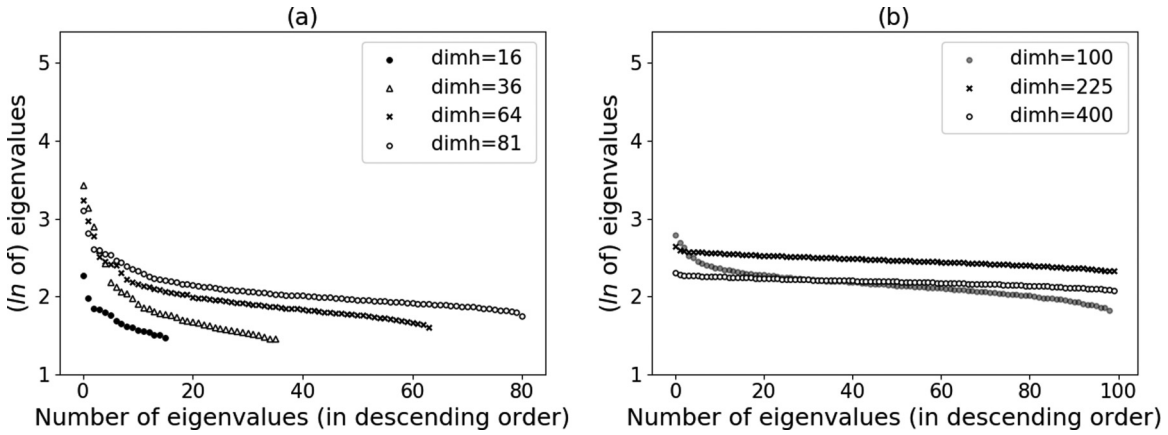

FIG. 17. Eigenvalues of WW^T for type-V RBM. The legend shows the number of hidden neurons, N_h.

In order to see the properties of $v^{(0)T}WW^Tv^{(0)}$ more than the magnetization in Fig. 14, we plot the same quantities but without subtracting the values at $T = 6$. Fig. 15 shows two cases for $N_h = 64$ and $N_h = 225$. These figures show that, at high temperature, the RBM with large N_h in Fig. 15(b) has larger components of the principal eigenvectors compared to the RBM with small N_h in Fig. 15(a). The difference must have caused the different behaviors in the RBM flows shown in Fig. 6 ($N_h = 64$) and Fig. 9 ($N_h = 225$). Namely, the former RBM flow approaches the critical temperature T_c, while the latter eventually goes towards higher temperature. The difference of two figures in Fig. 14 indicates that the RBM with larger N_h seems to have learned more characteristic features at high temperatures than the RBM with fewer N_h. Then, does the RBM with small N_h fail to learn the features of high temperatures? Which RBM is more adequate for feature extractions? Although it is difficult to answer which is more adequate without specifying what we want the machine to learn, we believe that the RBM with $N_h < N_v$ properly learns all the features of various temperatures while the RBM with $N_h > N_v$ has learned too many irrelevant features of high temperature. This is nothing but the second conjecture in Sec. III B, and it is supported by the behaviors of correlations in WW^T discussed in Sec. IV B.

D. Eigenvalue spectrum and information stored in W

Finally we study the eigenvalue spectrum λ_a of the matrix WW^T. Figs. 16 and 17 show the eigenvalues in descending order. In Fig. 16, the smooth dotted line shows the eigenvalues of the type-V RBM trained by configurations at all the temperatures ($T = 0,0.25,\ldots,6$), while the steplike dotted line denotes the eigenvalues of the type-L RBM (only $T = 0$).

These are obviously different. For the type-L RBM, only several eigenvalues are especially large, and the rest are apparently smaller. On the other hand, for the type-V RBM, the eigenvalues decrease gradually and there are no jumps or big distinctions between larger and smaller eigenvalues. The behavior indicates that, in the type-L RBM, the weight matrix holds only small relevant information and only a small number of neurons is sufficient in the hidden layer. In the type-V RBM, however, since it is trained by configurations at various different temperatures, all the eigenvectors are equally utilized to represent relevant features of spin configurations at various temperatures. Namely, in order to learn features of a wide range of temperatures, larger number of neurons in the hidden layer are necessary.[20] Using such larger degrees of freedom, the weight matrix has learned configurations with various characteristic scales at various temperatures so that the RBM can grasp the rich properties of these configurations.

The difference of the eigenvalues between type V and type L is also phrased that type V has a scale-invariant eigenvalue spectrum.[21] In contrast, the eigenvalues of the type-L RBM are separated into distinct regions in which the corresponding eigenvalues might represent features with different scales. It might be related to our previous numerical results, shown in Figs. 6–8, that the type-V RBM generates a flow toward the critical point where the configurations have scale invariance.

Finally in Figs. 17 and 18 we show differences in eigenvalue spectrum between RBMs with different numbers of hidden neurons, N_h.

As shown in Fig. 17, in the type-V RBM with $N_h > N_v = 100$ [Fig. 17(b)], there are no gaps in the eigenvalue spectrum and most eigenvalues have similar values. In contrast, for a smaller N_h [Fig. 17(a)], large and small eigenvalues are very different and the spectrum has a hierarchical structure. The type-L RBM shows similar behaviors as shown in Fig. 18. It might indicate that the RBM with larger N_h ($>N_v$) has learned too many details of the input configurations and the most relevant features are weakened. In other words, it has learned too many irrelevant features which are especially specific to configurations at higher temperature. This view is consistent with the discussion at the end of Sec. IV C and supports the second conjecture in Sec. III B.

To summarize, we find that the type-V RBM with smaller N_h ($<N_v$) can adequately learn configurations at a wide range

[20] However, too many hidden neurons ($N_h > N_v$) are not appropriate because lots of irrelevant features are acquired.

[21] Exactly speaking, it is not completely scale invariant, but compared to type L, at least there is no jump between larger and smaller eigenvalues. Such power law behavior is also seen in the SVD of Ising model configurations [30].

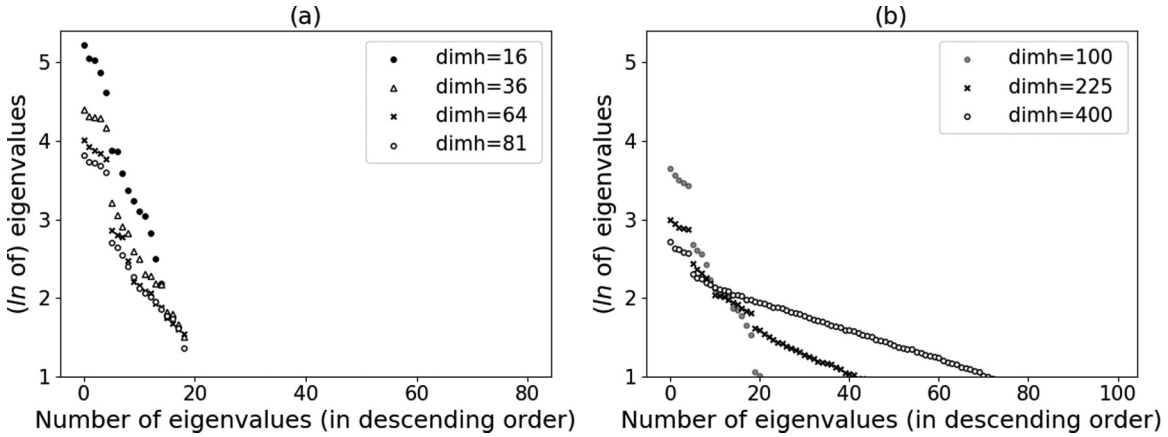

FIG. 18. Eigenvalues of WW^T for type-L RBM. The legend shows the number of hidden neurons, N_h.

of temperatures, without learning too many features at higher temperature. All the neurons in the hidden layer are efficiently used to represent features of various temperatures in a scale-invariant way. As a result, the RBM flow approaches the critical temperature T_c. The RBM flow is the same as the RG flow in that T_c is a fixed point, but the direction is opposite. Thus, a naive analogy between them does not hold.

V. DISCUSSIONS

In this paper, in order to see what the RBM learns in the process of training, we investigated the flow of configurations generated by the weight matrices of the RBM. In particular, we studied the Ising model and found that, if the RBM is trained by spin configurations at various different temperatures (we call it type-V RBM), the temperature of an initial configuration flows toward the critical point $T = T_c$ where the system becomes scale invariant. The result suggests that the configurations at $T = T_c$ are *attractors* of the RBM flow. In order to understand the numerical results of the RBM flows and to find a clue of what the machine has learned, we explored properties of the weight matrix W_{ia}, especially those of the product WW^T, by looking at the eigenvalue spectrum.

There are still many unsolved issues left for future investigations. Eigenvectors of WW^T must represent "features" that the RBM has learned, and the magnitude of the corresponding eigenvalue is an indicator of how much influence the feature affects. It will be interesting to pursue this analogy further and to extract more information from the eigenvalue spectrum, such as the amount of information stored in the NN [30]. As we saw in the present paper, the RBM flow gives an important clue of what the machine has learned. If an RBM is trained by configurations at a single specific temperature, it generates a flow toward that temperature, which confirms the hypothesis that the unsupervised RBM indeed extracts relevant features of the configurations at the temperature. The RBM flow should be described by the Langevin equation, whose drift term is controlled by the relevant features of the configurations that the machine has learned. We want to come back to this in future investigations.

In the present paper, we studied the Ising model, the simplest statistical model with the second order phase transition, and found that the critical point is an attractor of the RBM flow. Then we wonder what happens in the case of the first order phase transition. A simple example is the Blume-Capel model on a two-dimensional lattice. The Hamiltonian is given by $\beta H = -J \sum_{ij} s_i s_j + \Delta \sum_i s_i^2$, where $s_i = \pm 1, 0$ is a spin at site i. The model undergoes the first order phase transition that separates the parameter space (J, Δ) and the tricritical point at (J_c, Δ_c). If we train an RBM by configurations at various different parameters (such as the type-V RBM of the Ising model), the flow of parameters will be attracted to the tricritical point. On the other hand, if we use only a restricted set of configurations for training, e.g., various J with a fixed Δ ($<\Delta_c$), where is the RBM flow attracted? It is under investigation and we want to report the numerical results in future publications.

Finally we would like to comment on a possible relation between structures of RBM flows and how we recognize the world around us. Our finding is that there is an attractor of the RBM flow which characterizes the relevant feature that machine has learned before. We human beings also meet similar phenomena, namely, we can recognize more easily and comfortably what we have already learned many times than what we first experience. We here propose a conjecture that cognition at higher levels might be also related to fixed points of neural networks to which the input images are attracted. Our conjecture is the following:

(1) In the process of training, multiple fixed points are formed in the neural network.

(2) Input data (images) are attracted to one of them in the process of RBM flow.

(3) Each fixed point of RBM flow will correspond to what the NN "recognizes."

As a simple toy model, we have trained a neural network (RBM) by teaching various sizes of circles. Then we found that random input images are attracted to a circle with a typical radius. More interestingly, when we trained the network using circles and triangles, two fixed points of RG flow are formed: a circle and a triangle [32]. It would be interesting to further investigate wide varieties of examples in the scope of RBM flows and their fixed points to understand nature of cognition (e.g., the category recognition [33]). More systematic studies are under investigations.

ACKNOWLEDGMENTS

We would like to thank participants of the I-URIC frontier colloquium, especially Shunichi Amari, Taro Toyoizumi, and Shinsuke Koyama for fruitful discussions. We also thank Masato Taki for his intensive lectures on machine learning at KEK, and Hiroaki Matsueda for valuable comments and discussions on the properties of the weight matrices. Numerical computations were partially carried out using the work stations of Koutarou Kyutoku supported by JSPS Leading Initiative for Excellent Young Researchers. This work of S.I. and S.S. is supported in part by Grants-in-Aid for Scientific Research (No. 16K05329 and No. 16K17711, respectively) from the Japan Society for the Promotion of Science (JSPS).

[1] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science **313**, 504 (2006).

[2] C. Bény, Deep learning and the renormalization group, arXiv:1301.3124.

[3] S. Saremi and T. J. Sejnowski, Hierarchical model of natural images and the origin of scale invariance, Proc. Natl. Acad. Sci. USA **110**, 3071 (2013).

[4] P. Mehta and D. J. Schwab, An exact mapping between the variational renormalization group and deep learning, arXiv:1410.3831.

[5] A. Paul and S. Venkatasubramanian, Why does deep learning work? A perspective from group theory, arXiv:1412.6621.

[6] H. W. Lin, M. Tegmark, and D. Rolnick, Why does deep and cheap learning work so well? J. Stat. Phys. **168**, 1223 (2017).

[7] M. Sato, Renormalization group transformation for Hamiltonian dynamical systems in biological networks, arXiv:1609.02981.

[8] K.-I. Aoki and T. Kobayashi, Restricted Boltzmann machines for the long range Ising models, Mod. Phys. Lett. B **30**, 1650401 (2016).

[9] M. Koch-Janusz and Z. Ringel, Mutual information, neural networks and the renormalization group, arXiv:1704.06279.

[10] R. Salakhutdinov, A. Mnih, and G. Hinton, Restricted Boltzmann machines for collaborative filtering, in *Proceedings of the 24th International Conference on Machine Learning* (ACM Press, New York, 2007), pp. 791–798.

[11] H. Larochelle and Y. Bengio, Classification using discriminative restricted Boltzmann machines, in *Proceedings of the 25th International Conference on Machine Learning* (ACM Press, New York, 2008), pp. 536–543.

[12] G. E. Hinton, A practical guide to training restricted Boltzmann machines, in *Neural Networks: Tricks of the Trade* (Springer, New York, 2012), pp. 599–619.

[13] D. S. P. Salazar, Nonequilibrium thermodynamics of restricted Boltzmann machines, Phys. Rev. E **96**, 022131 (2017).

[14] K. G. Wilson, Renormalization group and critical phenomena. 1. Renormalization group and the Kadanoff scaling picture, Phys. Rev. B **4**, 3174 (1971); Renormalization group and critical phenomena. 2. Phase space cell analysis of critical behavior, **4**, 3184 (1971).

[15] L. Wang, Discovering phase transitions with unsupervised learning, Phys. Rev. B **94**, 195105 (2016).

[16] G. Torlai and R. G. Melko, Learning thermodynamics with Boltzmann machines, Phys. Rev. B **94**, 165134 (2016).

[17] A. Tanaka and A. Tomiya, Detection of phase transition via convolutional neural network, J. Phys. Soc. Jpn. **86**, 063001 (2017).

[18] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, Phys. Rev. E **96**, 022140 (2017).

[19] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions and crossovers through unsupervised machine learning: A critical examination, Phys. Rev. E **95**, 062122 (2017).

[20] A. Morningstar and R. G. Melko, Deep learning the Ising model near criticality, arXiv:1708.04622.

[21] H. Huang and T. Toyoizumi, Advanced mean field theory of restricted Boltzmann machine, Phys. Rev. E **91**, 050101 (2015).

[22] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. **14**, 1771 (2002).

[23] M. Á. Carreira-Perpiñán and G. E. Hinton, On contrastive divergence learning, in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)* (The Society for Artificial Intelligence and Statistics, New Jersey, 2005), pp. 33–40.

[24] Y. Bengio and O. Delalleau, Justifying and generalizing contrastive divergence, Neural Comput. **21**1601 (2009).

[25] S. Amari, Theory of adaptive pattern classifiers, IEEE Trans. Electron. Comput. **EC-16**, 299 (1967).

[26] H. Huang, Mean-field theory of input dimensionality reduction in unsupervised deep neural networks, arXiv:1710.01467.

[27] E. Agliari, A. Barra, A. Galluzzi, F. Guerra, and F. Moauro, Multitasking Associative Networks, Phys. Rev. Lett. **109**, 268101 (2012).

[28] H. Huang and T. Toyoizumi, Unsupervised feature learning from finite data by message passing: Discontinuous versus continuous phase transition, Phys. Rev. E **94**, 062310 (2016).

[29] C. H. Lee, Y. Yamada, T. Kumamoto, and H. Matsueda, Exact mapping from singular value spectrum of fractal images to entanglement spectrum of one-dimensional quantum systems, J. Phys. Soc. Jpn. **84**, 013001 (2015).

[30] H. Matsueda and D. Ozaki, Proper encoding for snapshot-entropy scaling in two-dimensional classical spin models, Phys. Rev. E **92**, 042167 (2015).

[31] T. Kumamoto, M. Suzuki, and H. Matsueda, Singular-value-decomposition analysis of associative memory in a neural network, J. Phys. Soc. Jpn. **86**, 024005 (2017).

[32] S. Iso, S. Shiba, and S. Yokoo (unpublished).

[33] See, e.g., R. Fergus, Visual bbject category recognition, Ph.D. thesis, University of Oxford, 2005, http://www.bmva.org/thesis-archive/2005/2005-fergus.pdf