Logical Reasoning of Revealing the Critical Temperature through Deep Learning of Configuration Ensemble of Statistical Systems

Ken-Ichi Aoki, Tatsuhiro Fujita, and Tamao Kobayashi
Institute for Theoretical Physics, Kanazawa University, Japan
† Yonago College, National Institute of Technology, Japan

[To be published in Journal of the Physical Society of Japan (2019)]

Abstract
Recently there have been many works on the deep learning of statistical ensemble to find out the critical temperature of possible phase transition. We analyze the detailed structure of the optimized deep learning machine and prove the basic equalities among the optimized machine parameters and the physical quantities of the statistical system. According to these equalities we conclude that the bias parameters of the final full connection layer are engraved the free energy of the statistical system as a function of the temperature. We confirm these equalities in one and two dimensional Ising spin models and actually demonstrate that the deep learning machine reveals the critical temperature of the phase transition through the second difference of bias parameters, which is equivalent to the specific heat. Our results deny the previous works claiming that the weight parameters of the full connection might play a role of the order parameter like the spin expectation.

1 Introduction and Summary
Recently the deep learning has been drawing attention in various applications of the image recognition or the artificial intelligence technology, since it has been very much successful beyond expectation. In other words why the deep learning is so effective in these fields has not yet been clear.

On the other hand, the renormalization group in physics has many common characteristics with the deep learning. The input data for the deep learning corresponds to the micro variables in physics, and the convolution filtering just resembles to the renormalization transformation itself, which picks up the main structural features of the input data, that is, the relevant operators (interactions) in the renormalization group terminology.

Therefore analyzing these similarities between the deep learning and the renormalization group would benefit us in two folds: clarifying the origin and mechanism of the effectiveness of deep learning will shed light on the strategy of improving the deep learning system and of developing new types of renormalization group.

In this article we are concentrated on the recent interest that the deep learning of configuration ensemble of statistical system may tell the phase transition temperature of the system through the optimized machine parameters which may behave as the order parameter of the system. However there has been no logical argument or understanding of why the optimized machine parameter can possibly behave

1 aoki@hep.s.kanazawa-u.ac.jp
2 t_fujita@hep.s.kanazawa-u.ac.jp
3 kobayasi@yonago-k.ac.jp
as the order parameter showing the phase transition temperature. Therefore we analyze the deep learning machine in detail and will clarify what is learned by the machine and how it learns.

Here we summarize our main results referring to Fig.1, which explains the logical flow of the machine learning, its optimized parameters and information of physical quantities to be obtained. The machine structure is drawn schematically in Fig.2.

Figure 1: What does deep learning learn from statistical system configurations?

We prepare some statistical system where the physical micro variables are spin $\sigma$, and $\{\sigma\}$ denotes a configuration of the system. The statistical weight of a configuration $\{\sigma\}$ is proportional to

$$\exp(-K_{j_{\text{in}}} H(\{\sigma\})),$$

where $H$ is the Hamiltonian of the system and $K$ is (inverse) temperature.

We pick up appropriate set of temperatures $K_j, j = 1, 2, \cdots, 16$, where we take 16 values in this article. Taking one temperature, we make up a system configuration and transfer it to the machine with a label of temperature number $j_{\text{in}}$. This type of learning is called the supervised learning. Note that only the temperature number $j_{\text{in}}$ labels the configuration data. The actual value of $K_{j_{\text{in}}}$ does not matter at all. Furthermore, as will be clear later, these temperatures do not have to be equally spaced nor even ordered (e.g. increasing order).

The machine receives the configuration with label $j_{\text{in}}$ as input data, which is shown in the lowest layer in Fig.2. The target of the machine is to find the temperature number which generates the configuration. Of course there cannot be a unique answer of temperature for a configuration. In fact any configuration can occur in any temperature, ignoring its actual probability of occurrence. Therefore the machine should learn the best answer for any configuration, that is, the probabilistic correctness of the answer should be maximized.

The machine is constructed according to the standard multi layered neural network system. First we set convolution layers, which make the lower layer variables multiplied by filters to make upper layer
variables. This is perfectly similar to the renormalization group transformation. Actually we work with multiple channel filters but they are abbreviated in Fig.2.

After some number of convolution layers, we make averaging of all variables to define just one variable $x$. This averaging policy is based on the translational invariance in the original space direction of our statistical system. Since the configurations are totally translationally invariant, the optimized machine parameters must be translationally invariant. When working with multi channel convolutional layers, they are finally summed up to make the single variable $x$.

This intermediate variable $x$ is essentially important. This single variable must contain sufficient information to best predict the input temperature. As is proved in the next section, $x$ must be the Hamiltonian of the system (up to normalization and origin).

Then we set the full connection layer from $x$ to $y_j$, $j = 1, 2, \cdots 16$, through linear functions,

$$y_j = w_j x + b_j,$$  \hspace{1cm} (2)

where $w_j$ is the weight and $b_j$ is the bias for each output channel.

Finally we apply the softmax function to $y_j$ and obtain the machine output $q_j$,

$$q_j = \frac{e^{y_j}}{\sum_k e^{y_k}}.$$  \hspace{1cm} (3)

We interpret this $q_j$ is the posterior probability of temperature $K_j$ determined by the machine for the input configuration $\{\sigma\}$.

As shown in Fig.1 we define $j_{\text{out}}$ by the maximum valued $q_j$ which should be understood as the predicted temperature. If the prediction $j_{\text{out}}$ equals to $j_{\text{in}}$, then the case is judged as correct answer, which totally defines the accuracy of the machine,

$$Ac = \langle \delta_{j_{\text{in}}, j_{\text{out}}} \rangle,$$  \hspace{1cm} (4)

where $\langle \cdot \rangle$ represents the expectation value for the total input mixed temperature ensemble. The machine parameters, convolution filters and full connection parameters are optimized to lower the standard error function,

$$Er = \langle - \log q_{j_{\text{in}}} \rangle,$$  \hspace{1cm} (5)

![Figure 2: Deep learning machine structure.](image-url)
by using the stochastic gradient descent.

The level of optimization can be evaluated by comparing the machine achievement with the theoretical bounds. In models treated in this article we can calculate the theoretical upper limit of the accuracy and the theoretical lower limit of the error with sufficient precision. After the numerous leaning (optimization), we retrieve the full connection parameters $w_j$ and $b_j$.

By simple and logical argument presented in Sec.4, we get the relations that these optimized machine parameters should obey. After simple manipulation, we get the general solution for these optimized parameters as written in Fig.1:

$$\begin{align*}
    w_{j}^{\text{opt}} &= a_1K_j + a_0, \\
    b_{j}^{\text{opt}} &= F(K_j) + c_1K_j + c_0,
\end{align*}$$

where $a_1, a_0, c_1, c_0$ are $j$-independent arbitrary constants. The weight $w_j$ is a linear function in $K_j$, and the bias $b_j$ is the free energy $F$ of the original statistical system up to an arbitrary linear function in $K_j$.

In our notation, the free energy is defined by

$$F(K_j) \equiv -\log Z(K_j),$$

and it is a dimensionless quantity.

These relations are remarkable, which was reported first by the authors.[13] They relate the optimized machine parameters after learning with the physical quantities of the input statistical system. The first equation (6) gives us the actual temperature values up to the unit normalization ($a_1$) and zero origin of the inverse temperature ($a_0$).

The second equation (7) declares that the bias is nothing but the free energy of the statistical system at temperature $K_j$ up to the total normalization of statistical weight ($c_0$) and zero origin of the Hamiltonian ($c_1$).

Note that the above emergence of arbitrary constants $a_1, c_0, c_1$ is inevitable, and they cannot be determined, that is, they have no physical significance here. The parameter $a_0$ is in general physically meaningful, but in this story of deep learning determination, it becomes just irrelevant.

With these relations we can calculate the second derivative of the free energy with respect to the temperature. The temperature spacing and order can be determined by using Eq.(6). Then the second derivative of free energy, that is, the specific heat can be evaluated by the second order difference of $b_j$, without annoyed by arbitrary parameters $a_1, a_0, c_1, c_0$.

Now the phase transition, if any and if enforcing singular peak behavior of the specific heat, is revealed through the second difference of the bias $b_j$, and the critical temperature number can be read off.

In the rest of this section, we apply the above summarized strategy to the 2-dimensional nearest neighbor Ising model and present the concluding plots.

We work with $32 \times 32$ sized square lattice. As the input temperatures we take 16 values in $[0.24, 0.54]$ with equal spacing (0.02) for simplicity. We plot the optimized full connection weight parameters vs $j$ or $K_j$ in Fig.3, which clearly shows that $w_j$ is fitted by a linear function in $K_j$ very well.

Fig.4 plots the optimized full connection bias parameters vs $j$ or $K_j$. It shows the bias is a function of $K_j$ with second derivative.

To evaluate the physical quantities, we make the 1st difference of $b_j$, which corresponds to the energy expectation value $E$ at $K_j$. In Fig.5 we plot the 1st difference of $b_j$ compared with the exact energy expectation values obtained by Monte Carlo simulation (MC). Also we plot a shifted 1st difference of $b_j$ so that it is equal to the MC value at $K = 0.44$. Note that the origin of the Hamiltonian has no significance here as $c_1$ is arbitrary. The shifted 1st difference of $b_j$ well approximates the exact energy expectation values.

Then we proceed to the 2nd difference of the bias parameter which corresponds to the specific heat. In Fig.6 we compare the results with exact specific heat values given by MC. Due to the finiteness of
our system, the specific heat does not diverge. We see the characteristic increase of the specific heat near the phase transition point $K_c \approx 0.44$ (in case of infinite volume limit).[14] Also note that there is no remaining arbitrary constants, and the 2nd difference of bias approximates the exact specific heat well quantitatively. Thus the specific heat singularity is in fact engraved in the optimized bias parameters.

These results are perfectly different from the statements in the previously published materials[11,12] that the optimized weight parameter works as the order parameter like the spin expectation value as a function of the temperature. However those previous statements were just claimed without any logical or even plausible argument. As is discussed in the next section, the single unique variable $x$ in the machine must represent the Hamiltonian so far as the machine is tuned to be optimistic. Then the full connection weight parameter $w_j$, which makes a dimensionless quantity multiplied by $x$, must have the dimensionality of inverse energy. Actually in our solution, the weight $w_j$ is a linear function of $K_j$ and
satisfies the correct dimensionality. Thus, $w_j$ cannot represents the spin expectation value, which is concluded even by this dimensionality argument only.

## 2 Optimized Machine Should Know Hamiltonian

In this section we prove that the optimized machine must know the Hamiltonian of the system as a function of configuration.

The statistical system is defined by the statistical weight given in Eq. (1). In general, the target external control parameter is multiplied by the conjugate physical quantity in the exponent. Here the target parameter is the inverse temperature $K$ and the conjugate is called the energy, or the Hamiltonian.
\( H(\{\sigma\}) \).

The normalized probability of occurrence of a configuration \( \{\sigma\} \) is given by
\[
P(\{\sigma\}; K) = \frac{\exp(-K H(\{\sigma\}))}{Z(K)}, \tag{9}
\]
where the partition function \( Z(K) \) is defined by summing up all possible configurations,
\[
Z(K) = \sum_{\{\sigma\}} \exp(-K H(\{\sigma\})). \tag{10}
\]

We set some number of different values for the inverse temperature \( K_j, j = 1, 2, \ldots, J \). In this article, we take \( J = 16 \). The total ensemble is the mixed set of configurations of \( J \)-sort of temperatures. In this mixed temperature ensemble, the occurrence probability of \( \{\sigma\} \) is
\[
P(\{\sigma\}) = \frac{1}{J} \sum_{j} P(\{\sigma\}; K_j) = \frac{1}{J} \sum_{j} \frac{\exp(-K_j H(\{\sigma\}))}{Z(K_j)}, \tag{11}
\]
where we fix the number of configurations for each temperature to be equal.

Suppose we find a configuration \( \{\sigma\} \), then the posterior (conditional) probability of temperature number \( j \) is given by
\[
Q(j; \{\sigma\}) = \frac{P(\{\sigma\}; K_j)}{\sum_k P(\{\sigma\}; K_k)}. \tag{12}
\]

Consider the machine output for the input configuration \( \{\sigma\} \). If the machine outputs the temperature number \( j \), its accuracy, the rate of successful guess, is nothing but this \( Q(j; \{\sigma\}) \).

Therefore in order to maximize the accuracy, the machine should output \( j_{\text{max}} \) which maximizes \( Q(j; \{\sigma\}) \),
\[
j_{\text{max}}(\{\sigma\}) = \arg \max_j Q(j; \{\sigma\}). \tag{13}
\]

This \( j_{\text{max}} \) is a function of \( \{\sigma\} \) and the maximum accuracy \( A \) is also a function of \( \{\sigma\} \),
\[
A(\{\sigma\}) = \max_j Q(j; \{\sigma\}) = Q(j_{\text{max}}; \{\sigma\}). \tag{14}
\]

By averaging this with total input \( \{\sigma\} \), we have the maximum value of the accuracy for the total ensemble,
\[
A_{\text{max}}(A(\{\sigma\})) = \sum_{\{\sigma\}} \frac{1}{J} \left[ \sum_{j} P(\{\sigma\}; K_j) \right] \left[ \frac{\exp(-K_j H(\{\sigma\}))}{Z(K_j)} \right] \left( \max_{i} \frac{P(\{\sigma\}; K_i)}{\sum_k P(\{\sigma\}; K_k)} \right). \tag{15}
\]

The maximum accuracy \( A(\{\sigma\}) \) and the maximum guess \( j_{\text{max}}(\{\sigma\}) \) are both functions of \( \{\sigma\} \). It is important that its dependency comes only through \( P(\{\sigma\}; K) \), that is, through the Hamiltonian \( H(\{\sigma\}) \). We can denote this restricted dependency explicitly as follows:
\[
A(\{\sigma\}) \Rightarrow A(H(\{\sigma\})), \tag{16}
\]
\[
j_{\text{max}}(\{\sigma\}) \Rightarrow j_{\text{max}}(H(\{\sigma\})). \tag{17}
\]

That is, \( A \) and \( j_{\text{max}} \) are functions of a single variable: the Hamiltonian of \( \{\sigma\} \).

Now we understand that the optimized machine to achieve the theoretical upper bound of the total accuracy must be able to correctly calculate the Hamiltonian as a function of \( \{\sigma\} \). Also, there is no way of improving the accuracy by adding information about other quantity than the Hamiltonian. Though not literally correct, we would like to express this situation as that knowing the Hamiltonian is necessary and sufficient condition for the machine to accomplish the highest limit of accuracy.
Here we should note a subtle point. The deep learning machine is actually optimized by the condition of minimizing the error function in Eq. (5), not by maximizing the accuracy function in Eq. (4). Minimizing error function is much stronger condition than maximizing the accuracy. This fact is understood quickly by looking at Fig. 24 which will be explained later in Sec. 8. In fact the minimized error machine necessarily achieves the highest accuracy, but the inverse does not hold. In this sense the above statement of the necessary and sufficient condition of knowing the Hamiltonian should apply rather to the minimum error machine. However, if we prepare initial set of temperature so that the spacing between neighboring temperatures are enough small, then the difference of these two type of machines would decrease and finally vanish in the limit of infinitesimal temperature discrimination.

We extend the above argument where there exists other external control parameter in the statistical weight. For example we consider uniform external magnetic field \( f \) which interacts with the total spin of the system \( S (|\sigma|) \). Then the statistical weight reads

\[
\exp(-KH(|\sigma|) + f S(|\sigma|)).
\]  

(18)

In fact this statistical weight is unphysical, since the interactions between spin and external magnetic field should be included in the Hamiltonian of the system coupled with the temperature. Of course in case that the Hamiltonian contains external field parameter, there is no change in the previous conclusion, that is, the Hamiltonian function including external field interactions must be learned by the machine.

However, as a purely theoretical system, we can make ensemble of the above type of system. Such situation occurs in case of the spontaneous symmetry breakdown (spontaneous magnetization) where we need some virtual external field to stabilize the system ensemble. Then we should reconsider this case here.

First of all, the occurrence probability of \(|\sigma|\) for temperature \( K \) and external field \( f \) is given by

\[
P(|\sigma|; K; f) = \frac{\exp(-KH(|\sigma|) + f S(|\sigma|))}{Z(K; f)}.
\]  

(19)

The normalization factor \( Z(K; f) \) does depend also on the external field \( f \). In the mixed temperature ensemble, the posterior probability \( Q(j; |\sigma|; f) \) of labeled temperature \( K_j \) is

\[
Q(j; |\sigma|; f) = \frac{P(|\sigma|; K_j; f)}{\sum_k P(|\sigma|; K_k; f)}.
\]  

(20)

Therefore both the maximum accuracy and the guessed temperature \( j_{\text{max}} \) assuring it do depend on the external field \( f \), and the total theoretical upper limit of accuracy depends on \( f \).

Now look at the \(|\sigma|\) dependence of those quantities by substituting Eq. (19) into Eq. (20). The dependence through the total spin function \( S (|\sigma|) \) cancels out between the numerator and the denominator, and the dependence only through the Hamiltonian survives,

\[
A(|\sigma|; f) \Rightarrow A(H(|\sigma|); f), \quad \text{(21)}
\]

\[
j_{\text{max}}(|\sigma|; f) \Rightarrow j_{\text{max}}(H(|\sigma|); f). \quad \text{(22)}
\]

We reach the same conclusion again. Even in case where also the temperature independent external field controls the statistical weights, the previous conclusion holds, that is, knowing the Hamiltonian is the necessary and sufficient condition for making the best available machine.

It should be noted here that what is observed by the optimized machine does not have to be the Hamiltonian itself. Consider a procedure that the machine observes a physical quantity \( X \), gets a value of it, \( x \), and guesses the input temperature as a function of \( x \). If all states having the same value \( x \) give a single value of Hamiltonian, then the observation of \( X \) is equivalent to that of Hamiltonian. In other words this case assures existence of a map \( X \rightarrow H \). Therefore, observation of such \( X \) can achieve the highest accuracy as well.
3 Characteristics of Optimized Machine

In this section we investigate the optimization condition of the deep learning machine to minimize the error function in Eq. (5).

Let us recall back the deep learning machine structure drawn in Fig. 2. Our machine, just before the final output stage, prepares a single variable $x$ by manipulating input spin configuration $\{\sigma\}$ through multi convolutional layers.

Due to the discussion in the previous section, the optimized machine should know the Hamiltonian as a function of $\{\sigma\}$. With this structured machine, the variable $x$ is the unique variable gathering characteristic information contained in the input $\{\sigma\}$. Therefore, the variable $x$ must necessarily be the Hamiltonian itself of the input configuration, $H[\{\sigma\}]$, so that the machine might achieve the highest accuracy and the lowest error. Hereafter, for the perfectly optimized machine, we suppose the variable $x$ to represent the Hamiltonian $H[\{\sigma\}]$ itself.

As is mentioned in the previous section, the optimized machine should know the Hamiltonian or its equivalent $X$, as far as there is a map from $X$ to $H$, in order to achieve the highest ability. Therefore the key variable $x$ does not have to be the Hamiltonian itself and it can be something other. However, in later sections, we confirm that actually the variable $x$ is optimized to represent the Hamiltonian itself up to the origin and the unit normalization. This is due to the structure of final output layers and the optimization procedure of our machines, that is, the full connection layer with linear functions in $x$ is coupled to the softmax function output and they are optimized to minimize the error function discussed below, which is a very much stronger condition than to bring the highest accuracy.

Using this variable $x$, the machine makes output $q_j$, $j = 1, 2, \cdots 16$, by using the full connection layer and the softmax function. First, 16 variables $y_j$ are set by the machine as defined in Eq. (2),

$$ y_j = w_j x + b_j, \quad (23) $$

where weight $w_j$ and bias $b_j$ are the machine parameters to be optimized in this full connection layer.

Then these variables $y_j$ are transformed into $q_j$ via the softmax function as in Eq. (3),

$$ q_j = \frac{e^{y_j}}{\sum_k e^{y_k}}. \quad (24) $$

These $q_j$'s are the final machine output, which are interpreted as the machine estimate of the normalized posterior probabilities of input temperature $K_j$ as a function of input $\{\sigma\}$, respectively.

Before discussing the optimization condition, we recapitulate the basic notions of evaluating the difference between two independent probability functions. We take two probability functions $Q(i)$ and $q(i)$ both of which are normalized,

$$ \sum_i Q(i) = 1, \sum_i q(i) = 1. \quad (25) $$

We introduce the Kullback-Leibler (KL) divergence to represent the size of difference between these two probability functions,

$$ D_{KL}(Q||q) = \sum_i Q(i) \log \frac{Q(i)}{q(i)}. \quad (26) $$

In fact, the KL divergence is straightforwardly proved to be non-negative,

$$ D_{KL}(Q||q) \geq 0, \quad (27) $$

and the equality above is satisfied only at $q(i) = Q(i), \forall i$, which is the unique minimum.
We evaluate the machine by comparing the machine estimate of the posterior probabilities with the correct posterior probabilities pre-determined by the input mixed temperature ensemble. The pre-determined posterior probabilities \( Q(j; \{ \sigma \}) \) are defined in Eq. (12). Then the KL divergence comparing this \( Q \) and machine estimate \( q \) for a fixed \( \{ \sigma \} \) is given by

\[
D_{\text{KL}}(Q || q)(\{ \sigma \}) = \sum_j Q(j; \{ \sigma \}) \log \frac{Q(j; \{ \sigma \})}{q_j(\{ \sigma \})},
\]

where the machine estimate \( q_j \) is calculated for each input \( \{ \sigma \} \) and we explicitly express the \( \{ \sigma \} \) dependence of \( q_j \).

As for \( Q(j; \{ \sigma \}) \), though they are pre-determined by the input mixed temperature ensemble, the machine should not be assumed to know it. There is the standard practical way of evaluating the KL divergence in the course of optimization. We define the error function \( Er(j_{\text{in}}; \{ \sigma \}) \) for each input \( \{ \sigma \} \) with label \( j_{\text{in}} \) by

\[
Er(j_{\text{in}}; \{ \sigma \}) = -\log q_{j_{\text{in}}}(\{ \sigma \}).
\]

Averaging for input configurations automatically sums up all possible input temperature \( j_{\text{in}} \) and \( \{ \sigma \} \) with correct probabilities of the input ensemble.

Denoting the average over the input mixed temperature ensemble by \( \langle \cdot \rangle_{j_{\text{in}}, \{ \sigma \}} \), the averaged error function,

\[
Er = \langle Er(j_{\text{in}}; \{ \sigma \}) \rangle_{j_{\text{in}}, \{ \sigma \}},
\]

is rewritten as follows by separating the summation into temperature and \( \{ \sigma \} \),

\[
Er\left( \sum_j Q(j; \{ \sigma \}) Er(j; \{ \sigma \}) \right)_{\{ \sigma \}} = -\left( \sum_j Q(j; \{ \sigma \}) \log q_j(\{ \sigma \}) \right)_{\{ \sigma \}}
\]

\[
= \left( D_{\text{KL}}(Q || q)(\{ \sigma \}) \right)_{\{ \sigma \}} - \sum_j Q(j; \{ \sigma \}) \log Q(j; \{ \sigma \})_{\{ \sigma \}},
\]

where the KL divergence for a fixed \( \{ \sigma \} \) is defined in Eq. (28).

The optimization of the machine is performed to minimize the error function by adjusting the output \( q_j(\{ \sigma \}) \) while \( Q(j; \{ \sigma \}) \) is the fixed input. Due to the non-negative condition in Eq. (27), the minimum error function is realized at

\[
q_j(\{ \sigma \}) = Q(j; \{ \sigma \}), \forall \{ \sigma \},
\]

and the minimum value is

\[
Er_{\text{min}} = -\left( \sum_j Q(j; \{ \sigma \}) \log Q(j; \{ \sigma \}) \right)_{\{ \sigma \}}.
\]

Introducing the probability function \( R(\{ \sigma \}) \) for \( \{ \sigma \} \) to appear in the total mixed temperature ensemble, we have

\[
Er_{\text{min}} = -\sum_{\{ \sigma \}} R(\{ \sigma \}) \sum_j Q(j; \{ \sigma \}) \log Q(j; \{ \sigma \})
\]

\[
= -\sum_{j, \{ \sigma \}} U(j; \{ \sigma \}) \log U(j; \{ \sigma \})
\]

\[
+ \sum_{\{ \sigma \}} R(\{ \sigma \}) \log R(\{ \sigma \}),
\]

where the machine estimate \( q_j \) is calculated for each input \( \{ \sigma \} \) and we explicitly express the \( \{ \sigma \} \) dependence of \( q_j \).
where $U(j; \{\sigma\})$ is defined by
\[
U(j; \{\sigma\}) \equiv R(\{\sigma\})Q(j; \{\sigma\}).
\] (35)

This is the product probability of temperature $K_j$ and $\{\sigma\}$ in the input ensemble. Therefore the minimum error function is the Shannon entropy of the total input ensemble minus the remaining entropy ignoring the temperature information.

Here we should mention about the accuracy given by the minimum error machine. The accuracy is frequently used to evaluate the machine ability to discriminate the input objects. In our case, it is defined by the rate that the predicted temperature coincides with the labeled input temperature. To get the highest accuracy, the machine must predict the temperature $j_{\text{out}}$ so that it maximizes $Q(j; \{\sigma\})$ as in Eq.(13),
\[
j_{\text{max}}(\{\sigma\}) = \arg \max_j Q(j; \{\sigma\}).
\] (36)

This is the best effort and no machine can perform better. As explained in the above, the optimized machine should output $q_j(\{\sigma\})$ exactly equal to $Q(j; \{\sigma\})$ for all $\{\sigma\}$ and thus it achieves the highest possible accuracy.

Finally in this section, we clarify how we should evaluate the level of machine power, that is, how near the obtained machine is towards the optimized one. One way is to compare the accuracy achievement with the theoretical highest value. However, the optimization procedure is controlled by minimizing the error function. Therefore we should evaluate the machine performance by comparing the error function with the theoretical minimum error value obtained in Eq.(34).

The value $E_{\text{rmin}}$ can be evaluated by MC simulation of the system. However, in case when we know the free energy of the system, it can be calculated simply. The probability function $R(\{\sigma\})$ and $Q(j; \{\sigma\})$ in Eq.(34) are expressed by functions of $\{\sigma\}$. However, they are actually functions of the Hamiltonian only as is stressed in the previous section,
\[
R(\{\sigma\}) \Rightarrow R(H(\{\sigma\})), \quad Q(j; \{\sigma\}) \Rightarrow Q(j : H(\{\sigma\})).
\] (37)

Then defining probability density function $V(E)$ ($E$ is a value of the Hamiltonian function $H$), we have
\[
E_{\text{rmin}} = - \int dEV(E) \sum_j Q(j; E) \log Q(j; E).
\] (38)

With this expression we can evaluate $E_{\text{rmin}}$ straightforwardly if we know the energy expectation value and the specific heat at every temperature. That is, we use the normal distribution approximation and replace the density function $V(E)$ with sum of the Gaussian distributions for each temperature. Of course this is not completely correct, but it is turned out to be practically excellent approximation in our statistical models of spins.

4 Relations between Optimized Machine Parameters and Physical Quantities

In this section we prove the main statement of this article: the relations between the optimized machine parameters and the physical quantities of the statistical system.

As is proved in the previous sections, our optimized machine must have two features as follows:

1. The intermediate variable $x$ represents the Hamiltonian of the system $H(\{\sigma\})$ for all $\{\sigma\}$.
2. The output $q_j(\{\sigma\})$ is equal to $Q(j; \{\sigma\})$ for all $j$ and $\{\sigma\}$.
The purpose of this section is to prove the relations declared in Eqs. (6), (7) from these two conditions. To make the notation simple, we denote $H(\{\sigma\})$ by $E$. It should be noted again that in the second statement, the condition “for all $\{\sigma\}$” is equivalent to “for all $E$” since $q_j(\{\sigma\})$ and $Q(j; \{\sigma\})$ can be reduced to $q_j(E)$ and $Q(j; E)$ respectively.

First of all, we express the key variable $x$ in terms of $E$,
\[
x(E) = -\frac{1}{a_1}E + \frac{c_1}{a_1},
\]
where $a_1, c_1$ are constants. The reason of somewhat peculiar definition here is for making the final formula simplest. We have to allow freedom of the unit normalization and the origin of energy. These two constants are irrelevant for physics and in fact the machine can become perfect with these uncertain arbitrary parameters. This means that the optimized machine parameters have the completely flat directions for the error function.

The full connection layer set the $y_j$ variable defined in Eq.(2). The final output $q_j(E)$ has the relative ratio as follows:
\[
q_j(E) \propto \exp(w_j x(E) + b_j),\tag{40}
\]
where the proportionality means that as $j$-space vector (16 dimensional), the left hand side vector is proportional to the right hand side vector.

Next, the pre-determined probability $Q(j; E)$ defined in Eq.(12) satisfies,
\[
Q(j; E) \propto P(E; K_j) = \frac{1}{Z(K_j)} \exp(-K_j E) \]
\[
\text{Then the second statement is nothing but the following proportionality,}
\]
\[
\exp(w_j x(E) + b_j) \propto \frac{\exp(-K_j E)}{Z(K_j)}.\tag{42}
\]
The proportionality factor can depend on $E$ and we get the equality,
\[
\exp(w_j x(E) + b_j) = C(E) \frac{\exp(-K_j E)}{Z(K_j)}.\tag{43}
\]
Taking the logarithm of both sides of this equality, we have
\[
w_j x(E) + b_j = \log C(E) - K_j E + F_j,\tag{44}
\]
where $F_j$ is the free energy defined by
\[
F_j \equiv -\log Z(K_j).\tag{45}
\]
The $E$-dependent proportionality factor $C(E)$ is restricted. Since the above equality holds for any $E$, $\log C(E)$ must be a linear function in $E$. We take the following notation,
\[
\log C(E) = c_1 E + c_0 + \frac{a_0 c_1}{a_1},\tag{46}
\]
where we introduced additional constants $c_0, c_1$. These constants also represent the completely flat directions of the error function of the optimized machine.

Substituting $x$ setting (39) into Eq.(44), we have two equalities,
\[
w_j = a_1 K_j + a_0,\tag{47}
\]
\[
b_j = F_j + c_1 K_j + c_0.\tag{48}
\]
Thus we reached the key result. The full connection layer parameters of the optimized machine can be given by these solution. This solution have four free parameters, which correspond to the completely flat
directions of the error function. Of course these four constants are totally undetermined and physically insignificant parameters.

Here we should mention that the machine did not know the temperature value $K_j$ itself, and just knows the temperature label $j$. In other words there is no way nor chance to tell the machine the temperature value $K_j$. However the optimized machine learns enough to know the temperature values through $w_j$ up to the unit normalization and the origin. Those free parameters do not affect the error function at all.

Let us explain how to calculate the physical singular behavior of the specific heat of the statistical system by using the optimized machine parameters only. First of all, we reorder the temperature label $j$ so that the weight $w_j$ might be monotonic. The proper direction, increasing or decreasing, does not matter in this stage, which will be made clear afterwards. Then we calculate the difference,

$$ \Delta_j = w_{j+1} - w_j, $$

which will be proportional to the corresponding spacing of temperatures,

$$ \Delta_j = a_1 (K_{j+1} - K_j). $$

Note that we do not assume equal spacing of the input temperatures.

Then we evaluate the first derivative of the free energy as follows, e.g.,

$$ \frac{dF}{dK} \bigg|_{K_j} = -\langle E \rangle_{K_j} \implies \frac{F_{j+1} - F_{j-1}}{K_{j+1} - K_{j-1}} = \frac{b_{j+1} - b_{j-1}}{\Delta_j} - c_1. $$

Now we have the energy expectation value at temperature $K_j$. The arbitrary constant $c_1$ represents that the origin of energy cannot be determined and the arbitrary $a_1$ corresponds to the unit normalization. Although even the sign of $a_1$ is indefinite, we can fix the increasing direction of temperature so that the energy expectation values might increase when the temperature increases, assuming the normality of the input statistical system.

Finally we evaluate the second derivative of the free energy,

$$ \frac{d^2F}{dK^2} = \langle (\Delta E)^2 \rangle_{K_j} \implies 2a_1^2 \left( \frac{b_{j+1} - b_j}{\Delta_j} + \frac{b_{j-1} - b_j}{\Delta_{j-1}} \right) / (\Delta_{j-1} + \Delta_j). $$

This quantity must be non-negative and equals to the specific heat of the statistical system. Therefore we find a peaked structure around the phase transition point if any.

To summarize this section, we stress the interesting point of our results. The free energy of the statistical system is not a simple quantity to calculate. According to the definition, we need the partition function, which is usually impossible to calculate. However the deep learning machine, looking at many configurations from mixed temperature ensemble, minimizing the error of predicting the input temperature label, finally engrave the free energy onto the bias parameters, even as a function of temperature. Investigating these engraved parameters, we can obtain the specific heat as a function of temperature, and find existence of singularity related to the phase transition.

## 5 Spin Models to be Learned

Hereafter we set up deep learning machines and make them learn the input temperature from input spin configurations. The spin models we use here are 2-dimensional nearest neighbor Ising model on the square lattice (2d-NNI) and 1-dimensional long range Ising model (1d-LRI). As before, the statistical weight is given by the Hamiltonian $H(\langle \sigma \rangle)$,

$$ \exp(-K H(\langle \sigma \rangle)), $$

and we call $K$ the temperature (actually the inverse temperature).
The 2d-NNI model is the common nearest neighbor interaction model with the Hamiltonian,

$$H = - \sum_{\text{n.n.}} \sigma_i \sigma_{i+1}. \quad (54)$$

This model, in the infinite volume limit, has the second order phase transition at $K = \frac{1}{2} \log(1 + \sqrt{2}) \approx 0.44$. \cite{14} The lattice size is 32x32 and input temperatures are 16 classes in the period $K = [0.24, 0.54]$ with equal spacing of 0.02. We expect the deep learning of this finite volume system should find the specific heat singularity remnant to show the phase transition temperature. This model has the exact expression of free energy as a function of $K$. However, it is for the infinite volume system, and suffers from large difference compared to our finite volume system. Therefore to calculate theoretical key quantities to evaluate the deep learning machines, we need high precision MC calculation.

The 1d-LRI model has the long range interactions,

$$H = - \sum_{i,n} K_n \sigma_i \sigma_{i+n}. \quad (55)$$

where the coupling constant $K_n$ is defined by

$$K_n = \frac{1}{n^p}. \quad (56)$$

The constant $p$ determines the dumping rate of the interactions. It is known that for $1 < p \leq 2$, there is a phase transition\cite{16, 17, 18, 19} at finite $K$. This model is the most primitive model of quantum dissipation which is an effective theory of coupled harmonic oscillator system after the environmental oscillators are integrated out. In fact the long range interactions mean nothing but the non-local interactions in the time direction.

According to the Finite Range Scaling (FRS)\cite{20} analysis, the phase transition point can be investigated by calculating the finite range system where the maximum distance of interactions is limited to some number. Although such finite range system does not bring about any phase transition, we can extrapolate the results by increasing the maximum range to guess the infinite range phase transition point.

Here we set $p = 1.8$ and the maximum range to be 8. We take the lattice size 1024, and the input temperatures are 16 classes in the period $K = [0.2, 0.5]$ with equal spacing of 0.02. For $p = 1.8$, the infinite range and infinite volume system has phase transition at $K = 0.41$. In this finite range and finite volume system, there expected to appear a peak of the specific heat which is a remnant of the phase transition.

This type of finite ranged 1d-LRI model has a good feature. Formulating the Block Decimation Renormalization Group (BDRG),\cite{20} we can exactly calculate the free energy as a function of temperature $K$ for any finite volume system. Therefore we can calculate the key quantities to evaluate machines almost exactly without relying on MC. This is a big benefit compared to 2d-NNI case.

Also the observation of the phase transition in 1d-LRI must be harder than 2d-NNI since the specific heat singularity of 1d-LRI with this size of maximum range is very much smaller than 2d-NNI case. With these reasons we take 1d-LRI with fixed maximum range as a main model in the following sections to investigate the phase transition search by the deep learning machines.

6 Deep Learning Machine Details

In this section we briefly explain the structure of the deep learning machine. We take the standard type of convolutional neural network\cite{21, 22} with multi layers.

The input configurations are made by MC simulation. As for 1d-LRI, we have developed a new method of generating configuration data without relying on the Markov chain procedures. This new method, called Exact Restricted Boltzmann Machine (ERBM), generates configurations with exactly vanishing auto-correlation. As for 2d-NNI, we adopt the Wolff cluster algorithm\cite{23} to make the ensemble stable even below the critical temperature.
In the results reported here, we do not add external magnetic field at all. We have checked that even with external magnetic field, all the results we claim still holds as they are.

The number of input configurations is 128,000 for 1d-LRI, 64,000 for 2d-NNI, in each temperature. As for 1d-LRI, we have used two types of data, one is the spin configuration representing spin up or down at each site, the other is the domain wall configuration, the dual variable, representing the existence of domain walls (spin flips). The domain wall representation gives generally better results with quick convergence, and hereafter we show the domain wall representation results.

We work with the Tensorflow software. According to the standard discipline, we use 80% of data for training and 20% for testing.

As explained in the previous sections, we use the error (cost) function in Eq. (5) to drive the stochastic descent optimization. Also the accuracy is observed to check the machine quality.

We recapitulate the total machine structure in a very simple notation. We prepare the input data set \( X \) and machine makes output data set \( Y \), Our purpose is to design function \( M \),

\[
M : X \rightarrow Y
\]  

(57)

The machine consists of multi layers and one layer converts the input data into the output data as follows:

\[
z_i(W, b, x) = f(u_i), \quad u_i = \sum_j \omega_j x_j + b_i,
\]  

(58)

where \( f \) is called the activation function, \( b \) is bias and \( \omega \) is weight. The boldface variables, \( W, b \) and \( x \), denote matrix or vector of components \( b, \omega \) and \( x \).

Building up multi layers we have

\[
z^{(l)}_i(W^{(l)}, b^{(l)}, x) = f(u^{(l)}_i), \quad u^{(l)}_i = \sum_j \omega^{(l)}_{ij} z^{(l-1)}_j + b^{(l)}_i,
\]  

(59)

where \( l \) is the layer number and \( z^{(0)}_j = x_j \). The convolutional layer is characterized through filters as

\[
u_{i,k} = \sum_{p=0}^{H-1} h_{p,k} z_{i+p,k} + b_{i,k},
\]  

(60)

where \( h \) is a filter of size \( H \), subscript \( k \) is the channel component and \( S \) is the stride.

By convolution the input data of size \( L \) are converted into the output data of size ((\( \left\lfloor \frac{L-H}{S} \right\rfloor + 1 \)). At the final layer we take summation (average) of all variables in the direction of space and channel,

\[
x = \sum_i \sum_k z_{i,k},
\]  

(61)

where we denote this key variable by \( x \) which has no suffix any more.

Then we set the full connection layer and apply the softmax function. The output of the softmax function is interpreted as the probability that input configuration belongs to the \( j \)-th class,

\[
P(C_k|W, b, x) = \text{softmax}(u_1, u_2, \ldots, u_K)
\]  

(62)

\[
e_k = \frac{e_k}{\sum_j e_j}, \quad u_k = \omega_k x + b_k.
\]  

(63)

The output \( y \) is class \( k_{\text{max}} \) giving the maximum probability,

\[
y(W, b, x) = \arg \max_k P(C_k|W, b, x).
\]  

(64)

To optimize the machine, we minimize the error function equivalent to the cross entropy,

\[
E(W, b) = -\sum_{n=1}^{N} \sum_{k=1}^{K} \{ \delta_{y_n,k} \log (y(W, b, x_n)) \},
\]  

(65)
Table 1: The hyper parameters (1d-LRI).

| hyperparameter          | value |
|-------------------------|-------|
| Convolution layer $l$   | 4 layer |
| layer 1 Filter size $H_1$ | 8     |
| layer 1 Stride $S_1$    | 8     |
| other layer Filter size $H$ | 2     |
| other layer Stride $S$  | 2     |
| Mini batch size         | 120   |
| Learning rate           | 0.0001|
| Channel                 | 8     |
| $Na$                    | 1, 2, 4 |

where $n$ represents the configuration number and $N$ is the total number. The machine performance is also evaluated through the accuracy $A$ defined by

$$A = \frac{1}{N} \sum_{n=1}^{N} \delta (y(W, b, x_n), y_n).$$

(66)

We show our typical setting of the machine parameters (hyper parameters) in table[1]. These hyperparameters are determined based on experience, trial and error actually. The parameter $N_a$ will be explained in Sec. 7.

As for the activation function, we use the rectified linear function (ReLU),

$$f(x) = \max\{0, x\} = \begin{cases} x & (x \geq 0), \\ 0 & (x < 0). \end{cases}$$

We adopt the AdamOptimizer[25] in Tensorflow for the stochastic descent method.

7 Theoretical Evaluation of the Maximum Achievement

Before starting the machine learning, we calculate the theoretical limit of the accuracy and the error function so that we might be sure about the level of optimization.

The possible highest accuracy is given in Eq.(15). To understand the meaning intuitively we start with a plot of energy distribution for each temperature in the total ensemble.

In Fig[7] we plot the energy distribution of 16 temperatures in 1d-LRI. As is noted in the previous section, we can calculate the free energy by BDRG as a function of temperature for this finite volume system. Therefore we obtain the energy expectation value and the energy fluctuation (specific heat) for each temperature. Using these two informations we approximate the energy distribution by the Gaussian form. This figure is plotted using this Gaussian approximation, which will be understood as enough precise for our purpose.

As for 2d-NNI case, we have to do MC simulation to calculate the energy expectation and fluctuation of $32 \times 32$ finite volume system for each temperature. Then using the Gaussian approximation, we plot the energy distribution in Fig[8].

We note here that in these Figs[7][8] there already seen peak structures of the specific heat. Since each energy distribution is normalized, the lower peak position corresponds to the wider distribution, that is, the larger specific heat. Comparing these two figures, 2d-NNI singularity remnant must be better recognized than that of 1d-LRI, by human at least.
As is stressed in Section 2, to achieve the highest accuracy, the machine should know the energy of input configuration and then select the temperature having the largest posterior probability. In energy distribution plots, this procedure is performed as follows: calculate the energy of the input configuration, look at the energy distribution plot and select the largest probability temperature.

In Fig.9 we draw a schematic diagram for the logic of maximum likelihood estimate. For the configurations whose energy is in the period $[E_A, E_B]$ should output temperature $K_1$ since it is the maximum likelihood estimate. Among these configurations, those which are judged to output the right answer is only those belong to the $K_1$ distribution. Therefore the total accuracy is nothing but the area below the upper envelope of these energy distribution lines. The overlapping region contributes to reduce the accuracy.

Thus the best prediction changes at the crossing points of the largest distribution curves. This transi-
The crossing point is directly calculable. In Fig. 9, at $E = E_B$, the two neighboring probability functions ($K_1, K_2$) match at point B,

$$\frac{\exp(-K_1E_B)}{Z(K_1)} = \frac{\exp(-K_2E_B)}{Z(K_2)}.$$  \hspace{1cm} (67)

Note that the number of configurations with the same energy does not depend on $K$. Taking the logarithm of both sides, we have

$$E_B = -\frac{F(K_2) - F(K_1)}{K_2 - K_1}. \hspace{1cm} (68)$$

This crossing point is graphically understood in the plot of the free energy as a function of $K$. Recalling the energy expectation is given by

$$\langle E \rangle_K = -\frac{dF}{dK}, \hspace{1cm} (69)$$

the crossing point $E_B$ is nothing but the contact point of the tangent with slope of connecting two points of the graph at $K_1$ and $K_2$. According to the convexity of the free energy as a function of temperature, we have the inequality (we suppose $K_1 < K_2$),

$$\langle E \rangle_{K_1} > E_B > \langle E \rangle_{K_2}. \hspace{1cm} (70)$$

Using the Gaussian approximation, we can evaluate the area below the upper envelope and find the theoretical maximum accuracy: 0.43634 (1d-LRI) and 0.51177 (2d-NNI). As for 1d-LRI, we check this accuracy by alternative evaluation through the very high precision MC, ERBM, and the result is 0.4364(1). This demonstrates that the Gaussian approximation is sufficiently precise for evaluating the accuracy limit.

By the way, probably man, and also machine, tends to look at the spin itself, and may count the total spin to guess the temperature. If we use the total spin to discriminate the temperature, then the maximum accuracy is only 0.094 (1d-LRI) and 0.296 (2d-NNI), which are much less than the above upper limit. Of course, as proved in Sec.2, any method relying on physical quantities other than energy has to decrease the total accuracy.

Next we consider the minimum of error function. As is written in Eq. (38), we can calculate it with the energy distribution functions directly. In Fig. 10, the posterior probability functions $Q(j; E)$ are plotted.
for each temperature number $j$. We integrate the entropy $\sum_j Q(j; E) \log Q(j; E)$ with the total energy
probability density $R(E) dE$ to get the minimum error. The Gaussian approximation gives 1.2697 (1d-LRI) and 1.0800 (2d-NNI). The ERBM high precision MC gives 1.2704(1) for 1d-LRI. Thus also for the
error function the Gaussian approximation is extremely good.

The theoretical highest accuracies look not so high. In order to make it higher, we may bundle input
configurations, that is, the machine looks at $Na$ configurations at once. We denote each energy of these
$Na$ configurations by $E^{(i)}_k$, $k = 1, 2, \ldots Na$, and $i$ represents the configuration set number.

Then the probability of occurrence of this bundled configurations for temperature $K$ is given by the
product of probabilities of the component configurations,

$$P(\{\sigma\}_i, K) = \prod_{k=1}^{Na} \frac{\exp\left[-KE^{(i)}_k\right]}{Z(K)^{Na}} = \left(\frac{\exp\left[-KE^{(i)}_a\right]}{Z(K)^{Na}}\right)^{Na}.$$

Here we introduced the average energy $E^{(i)}_a$ for $Na$ configurations,

$$E^{(i)}_a = \frac{1}{Na} \sum_{k=1}^{Na} E^{(i)}_k.$$

We see that the average energy $E_a$ now controls the probability. Then the previous calculations hold
by replacing the energy of configuration with the average energy of the bundled configurations. The
average energy distribution functions has the same expectation value and $1/Na$ variance of the original
single energy distribution. Note that the crossing point energy between distribution curves do not change
by bundling. Increasing $Na$, the distribution becomes narrower and the accuracy must increase. For
example, for 1d-LRI, we have the upper limit for the accuracy: 0.56852 ($Na = 2$) and 0.72154 ($Na = 4$),
and the lowest error: 0.959 ($Na = 2$) and 0.640 ($Na = 4$). Thus the accuracy upper bound increase and the
lowest error decreases drastically. This means that using bundled multi configurations, its information
becomes larger, and the temperature can be discriminated more easily. However these machines with
$Na > 1$ are turned out to be worse for the phase transition search as is seen later in Sec. 8.
8 Optimizing Deep Learning Machine

We start analyzing the optimized machine parameters. Here we will report detailed results for 1d-LRI case. For 2d-NNI case, the main results are already shown in Sec. 1.

The energy distribution of the input MC configurations are plotted in Fig. 11. They reproduce well the Gaussian approximated theoretical distribution in Fig. 7.

As for the evidence of the level of optimization we refer to the accuracy and the error function achieved. For 1d-LRI, after 2M iterations of learning procedures, the error function reaches below 1.2755 and becomes stable, which should be compared with the theoretical minimum value of 1.2704. Also the accuracy becomes above 0.435, whose theoretical upper bound is 0.4364.

For 2d-NNI, after 6M iterations of learning procedures, the error function is pushed down to 1.111 where the theoretical minimum is 1.0800, and the accuracy becomes 0.501 where the theoretical upper
limit is 0.511.

According to these levels of achievement we understand the machines are optimized very well.

The resultant output temperatures are plotted in Fig.12. This figure is the energy distribution of the
configurations which are classified to each predicted temperature. The shape of distributions are far from
the Gaussian. The perfect output must show up as rectangle diagrams where the predicted temperature
changes suddenly at the crossing points of energy. Since the discrimination is not perfect, there are some
overlaps of distribution lines.

As is stressed in the previous sections, to achieve high accuracy results, we need precise energy
observation of input configurations. Fig.12 represents that the energy discrimination is well successful in
the sense of energy block. This feature is also checked as in Fig.13 where we plot the relation between
the output class and the energy of configuration. The perfect machine gives the contiguous step function
determined by the maximum likelihood estimate, which is drawn by a polygonal line there.

![Figure 13: Output temperature class vs input configuration energy.](image)

To look into the details of energy discrimination, we check the intermediate key variable $x$. We plot
the value $x$ vs the input configuration energy in Fig.14. This diagram shows the remarkable result that
the variable $x$ is in fact organized to be a linear function of the input energy. There are some broadening
of the map, which causes the remaining errors.

The accuracy itself is not so high, only 0.435. However this fact itself does not matter at all. The
central issue is that how near this value is towards the theoretical upper limit, which can be realized only
by the exact observation of the energy of input configuration. In fact, Fig.14 proves that the machine has
learned how to evaluate the energy of input configurations. This is equivalent to the fact that the machine
now knows the Hamiltonian function $H(\sigma)$ of the statistical system.

To make the point above clear, we plot the output histogram of the configurations coming out of
the single input temperature. Fig.15 shows the output for the input temperature $K = 0.24$ and 0.4
configurations.

Now we should appreciate these histograms. The central peak of each histogram corresponds to the
case that the output temperature coincides with the input temperature. The height is not so high and
actually around 40% of the total histogram, which is the accuracy itself. So this diagram might be seen
as that the machine made mistakes for so many configurations, that is, 60% of the input configurations
are assigned wrong predicted temperature.

Here comes the most important point answering the basic question about the optimized machine, that
is, what is learned by the deep learning? The above statement, though it is correct, is not right at all. Rather this diagram should be understood as *the energy spectrometer* outputs of the input temperature configurations. Thus the machine becomes the energy spectrometer after learning. By supervised learning through the temperature estimate, the machine possessed the spectrometer, analyzer of the energy, the temperature conjugate variable.

Now we claim again that the low accuracy itself does not matter. The machine has learned much more deeper information, the Hamiltonian of the statistical system. Now it is time for the machine to show its power obtained.

We define another characteristics to evaluate machine ability, the Hamiltonian Recognition Rate
Referring to Fig. 13, we define HRR $R_n$ for output class $n$ by

$$R_n \equiv \frac{M_n}{N_n},$$

(74)

where $N_n$ is the number of inputs whose best output temperature class is $n$, and $M_n$ is the number of inputs which are on the right step function line. In Fig. 16 we plot the HRR achieved by this machine. The total HRR,

$$R = \frac{\sum_n M_n}{\sum_n N_n},$$

(75)

is 93.7% for 1d-LRI, and 86.7% for 2d-NNI. By evaluating the machine from this point of view instead of the standard accuracy value itself, we can do better rating of the optimized machine.

Now we check the optimized parameters in the full connection layer. We proved in Sec. 4 that the parameters $w_j$ and $b_j$ are expected to be optimized with arbitrary linear functions of $K_j$ as Eqs. (47), (48).

In Fig. 17, the optimized weight $w_j$ in the full connection layer are plotted with respect to $K_j$. Here, for simplicity, we take equally spacing temperatures $K_j$ and diagram axis uses $K_j$ instead of their label $j$. The optimized weight $w_j$ obeys a linear function of $K_j$, just as we proved in Eq. (47) for the optimized machine.

Next we plot the optimized bias $b_j$ in Fig. 18. It must have the non-vanishing second derivative. According to our relation Eq. (48) proved in Sec. 4, we plot $b_j - F_j$ (bias minus free energy) as a function of $w_j$ in Fig. 19. It shows a linear function perfectly. Therefore the optimized machine parameters correctly satisfy the predicted formula.

We calculate the first difference of the optimized bias according to Eq. (51) to evaluate the energy expectation value as a function of the temperature. In Fig. 20 we plot the first difference and the exact values calculated by BDRG method. As is explained in Sec. 4, we see a constant shift between these two plots due to arbitrariness of the origin of energy. Apart from this constant shift, the coincidence is excellent, which is demonstrated in Fig. 21 where we shift the plot to coincide at $K = 0.44$.

Then we proceed to the second difference of bias defined in Eq. (52) and plot it in Fig. 22. We also plot the exact value of this finite system calculated by BDRG. Though there remain some statistical fluctuations, we can clearly see that the 2nd difference of bias reproduces the specific heat well quantitatively and the peak singularity appears which is the phase transition remnant. It should be noted that the right
most point looks no good, the reason of which must be the edge effect in the temperature selection. This type of edge anomaly can be seen also in Figs. 16, 10. Here we add another result using the bundled configuration method introduced in Eq. (72), where we have the higher accuracy limit. The optimized machine actually achieves higher accuracy results, 0.702, but it is somewhat low compared to the theoretical limit 0.72154, even after 2.9M leaning cycles. Also the error function remains still large value, 0.720, compared to the lowest limit 0.640. Thus, the bundled configuration learning is not effective and is not well tuned, rather the situation is definitely worse than the single configuration learning. We show the second difference of bias in Fig. 23. The shape of plot is far from the exact values, which is understandable from the error function value.

Of course, also for bundled configuration case, the perfectly optimized machine should obey our solution as well. However, due to the higher accuracy limit, the requirement in tuning machine is weaker...
Figure 19: [Optimized $b_j$ - Exact free energy] shows linear in $w_j$ (1d-LRI).

Figure 20: The first difference of $b_j$ and the exact energy expectation value.

compared to the single configuration case, and conversely the optimization procedure becomes subtler and harder, and the final physical output level goes down. This result is very interesting. We recall a legend that loose education does not make a person.

Now we look at another important functions defined in the full connection layer,

$$y_j = w_j x + b_j.$$  \(76\)

These linear functions of $x$ will be input of the softmax function to make final output. We plot these functions in Fig[24] The variable $x$ represents the energy of the input configuration. We select highest $y_j$ for each $x$, which will be the maximum likelihood estimate of the temperature. We see the contiguous change of the output temperature in Fig[24].

Now we can understand important facts about the difference between the maximum accuracy and the minimum error conditions. Maximum accuracy can be achieved if the output temperature class obeys the
Figure 21: The first difference of $b_j$ shifted for comparison.

Figure 22: The second difference of $b_j$ and the exact specific heat.

step function depicted in Fig.13. In Fig.24 this condition is seen as follows. The crossing points of 1st and 2nd upper lines represent the stepping points of energy. Keeping the $x$ coordinate of these points, we can move these $y_j$ lines without decreasing the total accuracy. What is the highest line does matter, but positions of other lines below it do not matter. Since we have many (17 in this case) degrees of freedom to move, there are many completely flat directions of the total accuracy in the parameter space of $w_j$ and $b_j$.

On the other hand we have only four arbitrary parameters for the flat directions giving the minimum error. This is because minimum error does require complete coincidence of all output temperature posterior probabilities at any energy. Therefore positions of all lines at any energy do matter to assure the minimum error. Therefore the minimum error condition is very much stronger condition than the maximum precision condition.
In Fig. 24 we also notice that there appears an envelope function of these $y_j$ lines. As is obtained in Eq. (44), the optimized machine parameter should obey the equality,

$$y_j = w_j x + b_j = -K_j E + F_j,$$

(77)

up to a linear function in $E$. At each point on the envelope, the value $E$ is nothing but the energy expectation value at the dominant temperature, $E_j^*$. Therefore on the envelope we can write

$$y_j = -K_j E_j^* + F_j = S_j.$$

(78)

Thus the envelope curve seen in Fig. 24 represents the entropy of the statistical system as a function of energy up to a linear function in $E$. This arbitrary parts do not contribute when we take the second derivative of the envelop curve, which also gives the specific heat again.
9 Concluding Remarks

We analyzed a simple model plant where the statistical system with a definite temperature generates configurations, which are transferred to the input data of the deep learning machine for the supervised learning of the statistical system temperature. We prepare 16 temperature classes. The value itself of temperature does not matter at all here. Actually we have no way of determining the temperature value of a physical realistic system without a verified thermometer. That is, we do not control the temperature value itself, but we just set up some equilibrium states with different temperatures by adjusting the “heating” power for the pan.

Thus our setup of the system is totally realistic, that is, we have a statistical system which may realize some number of equilibrium states with different temperatures. There is no need of knowledge of the temperature values themselves. Even in this situation, the deep leaning machine can detect the special temperature class which corresponds to the possible phase transition point.

To evaluate objectively the level of leaning or perfectness of the machine, we theoretically calculate the upper bound of the accuracy and the lower bound of the error function (cross entropy) in advance for the input configurations. The optimized machine actually gives the scores which are very near to these theoretical bounds.

As is proved in Sec.2, the optimized machine must know the Hamiltonian of each configuration to realize the theoretical bound. Our machine is organized so that the input data are transformed by multi layered convolutional filtering and by averaging the space direction, finally into the single variable $x$. Then this variable must be the Hamiltonian up to the origin and the unit of energy. We confirmed this property of the simple correspondence between the key variable $x$ and the energy of the input configuration as drawn in Fig.14.

Then the final section of the machine is the full connection layer connecting the variable $x$ and the output temperature classes through the machine parameters: weight $w_j$ and bias $b_j$, where $j$ labels the output temperature class. The condition of minimizing the error function defined by the cross entropy is satisfied by the linear equations for $w_j$ and $b_j$ obtained in Eqs.(47), (48).

Now the temperature values can be read off through the weights $w_j$ up to the unit scaling and the origin, and as a result $K_j$ is obtained as a linear function of $w_j$. Then the bias $b_j$ is the free energy up to a linear function of $K_j$. This is a remarkable result. The difference, a linear function, remains undetermined, but it is natural since it comes from the freedom of the total normalization of the partition function $Z$ and the origin of energy. These two quantities are irrelevant here, or impossible to determine by our machine learning method of analyzing the probability density of statistical system. Then, the second derivative of free energy, the specific heat or the fluctuation of energy, is obtained without any arbitrary parameter.

The free energy is the physical quantity which has more detailed information of the statistical system than the energy itself. Its evaluation needs the information of entropy, the number of independent states with energy fixed. Therefore it looks somewhat miraculous that the deep learning machine engrave the free energy on their optimized bias parameters.

The key logic is that the optimization is performed to make the probability density of energy of the machine output equal to that of the input mixed temperature set of configurations. In order to evaluate the probability density, we need the entropy information or the free energy in addition to the energy, since we have to normalize them properly for each temperature. Also the basic machine structure that the key variable $x$ and the softmax function connecting it to the output temperature classes makes the logic complete, which finally picks up the bias to represent the free energy.

Here we should stress again that the above remarkable results come from the optimization condition we adopted. Our machine is trained to make the cross entropy minimized. However the minimum cross entropy is not a necessary condition for the maximum accuracy, though it is a sufficient condition.

The reason of using the cross entropy instead of the accuracy to tune the machine parameter is simply that the stochastic steepest descent method cannot simply be applied to maximizing the accuracy.
Minimizing the cross entropy can be dealt straightforwardly by the stochastic steepest descent. This optimization condition of the minimum cross entropy does deduce our conclusion about the weights and biases of the full connection section.

As for the input statistical system, we adopt one dimensional Ising model with long range interactions and two dimensional Ising model with the nearest neighbor interactions. In both models, our results of the free energy extracted from the bias parameters coincide well with the exact calculations or MC results. The peaked structures correctly indicate the remnant of the phase transition in both models.

Our results show that the machine trained to guess the temperature must recognize the Hamiltonian of the system, which is the conjugate variable of the temperature in the probability density function. In other words, leaning the temperature of configurations, the machine actually becomes the spectrometer of the Hamiltonian, the conjugate variable of the temperature. As is stressed when explaining Fig.15, the prediction of the temperature is just the way of leaning, and actually the accuracy of prediction cannot be better than the theoretical upper limit which is rather low value. However, due to the learning, the machine obtains the ability to calculate the Hamiltonian, and this ability can possibly be perfect.

We should mention here that the optimized weights and biases are nothing but what is memorized in the neural network as a result of leaning. Therefore we understand the total procedure as follows. By learning the temperature of each input configuration, the machine obtained the ability to evaluate the energy (conjugate of the temperature), leaving the free energy and temperature values on the neural network couplings. This is the memory newly constructed due to the learning in addition to the energy spectrometer function engraved on the lower layer parameters.

We can generalize this structure for multi-control parameters, for example, the temperature and the external magnetic field. Suppose we have $i_M$ control parameters $K^{(i)}$ and the probability density is written as

$$P(\sigma) = \frac{\exp(-\sum_i b_i K^{(i)} H_i[\sigma])}{Z(K^{(i)})}, \quad (79)$$

where $H_i[\sigma]$ is the corresponding Hamiltonian. Note that control parameters can be understood also as individual interaction coefficients, so far as they can be controlled by the outsiders of statistical system.

We make configurations for set of parameters $K^{(i)}$ and transfer them to the input configuration of the deep learning. We denote $K^{(i)}$ value of set (class) $j$ by $K^{(i)}_j$. To achieve the highest possible scores of accuracy, the deep learning machine must know all $i_M$ functions $H_i[\sigma]$, which are nothing but the conjugate variables of $K^{(i)}$ respectively. After some convolutional layers and averaging in the space direction, we must keep $i_M$ key variables, $x_i$, which represents $H_i[\sigma]$. The final full connection layer is defined by weight $w_{ji}$ and bias $b_j$ as follows:

$$y_j = \sum_i w_{ji} x_i + b_j. \quad (80)$$

As before, the softmax function outputs the probability,

$$q_j = \frac{e^{y_j}}{\sum_k e^{y_k}}. \quad (81)$$

Following the arguments in Sec.4, we reach the conclusion that the optimized parameters must obey the analog of Eqs.(47), (48),

$$w_{ji} = a_1 K^{(i)}_j + a_0, \quad (82)$$

$$b_j = F(K^{(i)}_j) + \sum_i c_1 K^{(i)}_j + c_0, \quad (83)$$
where $a_{1i}, a_0, c_{1i}, c_0$ are arbitrary constants. Thus the optimized machine parameters have the same characteristic structures. The weight $w_{ji}$ obeys linear function of $K_j^{(i)}$ in $j$ direction. The second derivative of the free energy with respect to $K^{(i)},$

$$\frac{\partial^2 F}{\partial K^{(i)} \partial K^{(k)}},$$

(84)
can be evaluated without ambiguity.

We did not argue the machine structure in detail. Actually the simple standard form of the deep learning with multi-layered convolutional filtering worked well and achieved scores very near to the theoretical bounds. However, it is still not perfect of course. As for one dimensional nearest neighbor interaction case, we can easily make the perfect machine within this standard framework of machine structure. Although the possible perfect ability of the machine to approximate any function has been argued,[26, 27, 28, 29, 30] the actual machine with finite number of components has limited performance. As a result, we should say our simple machine fortunately obtains high enough power so that we can observe the phase transition remnant peak of the specific heat in the second difference of the bias parameters.

We guess the reason that such type of convolutional filtering succeeds in calculating the Hamiltonian is due to the fact that the Hamiltonian is constituted of the relatively local spin interactions. In this point, the renormalization group method of summing up local fluctuations to define effective macro interactions should shed light on the effectiveness of the convolutional layers. However, the detailed and precise correspondence between the renormalized variables and the variables on intermediate layers has not yet been clarified. It must be also interesting that the Hamiltonian recognition ability should be tested for various Hamiltonians including unphysical and unnatural interactions, like long range dominated ones etc.

We proved that the optimized machine should know the system Hamiltonian so that it may achieve the highest accuracy. Then one may claim that such a machine can be trained more directly by making a function-approximating machine with supervised learning. However it should be noted that for such supervised learning, we need exact Hamiltonian function to label every input configuration. This is not usually the case of realistic issues. Our input label is just the temperature class of generating the configuration. This is sufficient for the machine to learn the Hamiltonian and to engrave the free energy of the system.

As the original motivation of our research is to clarify the relationship between the deep learning and the renormalization group. However, now we notice that there exists rather big differences between the deep learning machine to learn the temperature and the renormalization group analysis, although the networking structure itself resembles very well to each other.

The temperature is the target of deep learning and the machine handles configurations generated by all classes of temperature simultaneously. On the other hand, for the renormalization group procedure, the temperature is the coefficient of the most relevant operators controlling the system. Strictly speaking, the relevant operator contains the Hamiltonian as its component and the temperature may play a role of labeling the size of relevant operator.

The renormalization group describes the change of this label, the temperature parameter, as the lower layer variables are renormalized into upper layer variables. That is, the temperature changes from lower to upper layers, and it is enlarged, which means relevant, while other operators are reduced to vanish and thus called irrelevant.

We should note another apparent difference between the deep learning machine and the renormalization group. The weights and biases defining filters in the convolutional network, are optimized independently for each layer. These parameters define the upper layer variables in terms of the lower layer variables, and are nothing but the definition of renormalization.

The renormalization group transformation is defined commonly for all layers. This is the essential feature of the renormalization group, that is, applying the same transformation many times. Then after
identifying the fixed point of the transformation which corresponds to the criticality of the system, we linearize the renormalization group transformation in the neighborhood of the fixed point. Then the transformation is characterized by its eigenvalues. Repeating the same transformation results in that there appear enhancement and reduction of operators depending on the absolute value of their eigenvalues, greater than unity (relevant) or less than unity (irrelevant).

Thus changes of operators are described by the geometric series near the criticality. The relevant operator changes with a ratio greater than unity and the ratio can be observed as the critical exponent through the experimental observation around criticality. This is nothing but the essential framework of understanding the second order phase transition by using renormalization group. This basic logic of the renormalization group analysis looks quite different from the optimized machine parameter structures. However it should be noted that this difference should not be understood as contradiction. Optimization of renormalization transformation independently for each step may give a new developing direction of the renormalization group method.

Keeping these points in mind, we consider that the restricted Boltzman machine (RBM) is expected to have tighter correspondence with the renormalization group.[6] The lower layer called the visible variables are the micro variables whose probability distribution is given by the micro Hamiltonian, and the upper layer hidden variables are regarded as the renormalized macro variables. The weights and biases defining the RBM is nothing but the rules of defining the renormalization of micro variables to give the macro variables. The upper layer macro variables should follow the renormalized Hamiltonian. These layers can be stacked deeply just as the recursive renormalization group transformations.

Suppose some RBM and we use the visible variables as input configuration of the deep learning machine. We do not know the Hamiltonian controlling the visible variables. Then can we get information about the Hamiltonian through the optimized deep learning machine parameters?

To learn something about the Hamiltonian we need a set of different values of the conjugate of the Hamiltonian, the temperature. However there is probably no simple way of adjusting the temperature with a fixed Hamiltonian by changing the RBM weight and bias parameters. The number of degrees of freedom of the Hamiltonian controlling the RBM visible variables is much larger than that of RBM machine parameters. Note that RBM defined Hamiltonian contains multi spin interactions in addition to the basic two spin interactions and thus the possible number of different sorts of interactions is quite large. Therefore we need more detailed analysis of RBM and its effective Hamiltonians before going a step forward.

Anyway, we hope that the deep mechanism of high performance of the deep learning machine should be related to the mechanism of the renormalization group to treat many degrees of freedom by coarse graining and to pick up the characteristic variables automatically as relevant operators. On the other hand we also hope that the future understanding of the origin of deep learning machine power will give us new developing ideas for the renormalization group method which are already appreciated as the universal tool in various fields.

Acknowledgment

We acknowledge fruitful, suggestive and helpful discussions with Yasuhiro Fujii, Shin-Ichiro Kumatomo. We appreciate collaborative works of ERBM by Mariko Iijima and of MC simulation by Tsubasa Jingu, and initial set up of our computing system and preparation studies in early days of this project by Yuusuke Hori and Shinnosuke Onai. Also great thanks should be given to excellent lectures given by Muneki Yasuda for motivating us to study the relationship between the deep learning and the renormalization group.
References

[1] K.G.Wilson, Phys. Rev. B 4, 3174 (1971).
[2] K.G.Wilson, Phys. Rev. B 4, 3184 (1971).
[3] K.G.Wilson and J. Kogut, Phys. Rep. 12, 75 (1974).
[4] L.P.Kadanoff, Annals Phys. 100, 359(1976).
[5] P.Mehta, and D.J.Schwab, arXiv:1410.3831 (2014).
[6] K.Aoki, and T.Kobayashi, Mod. Phys. Lett. B 30, 1650401 (2016).
[7] S.Iso, S.Shiba, and S.Yokoo, Phys. Rev. E 97, 053304 (2018).
[8] S.-H.Li and L.Wang., Phys. Rev. Lett. 121, 260601 (2018).
[9] S. Foreman, J. Giedt, Y. Meurice, and J.Unmuth-Yockey, EPJ Web Conf. 175, 8 (2018), Phys. Rev. E 98, 052129 (2018).
[10] M.Koch-Janusz, and Z. Ringel, arXiv:1704.06279 (2017).
[11] A.Tanaka, and A.Tomiya, J. Phys. Soc. Jpn. 86, 063001 (2017).
[12] S.Arai, M.Ohzeki, and K.Tanaka, J. Phys. Soc. Jpn. 87, 033001 (2018).
[13] K.I.Aoki, T.Fujita and T.Kobayashi, JSAI Jinkou Chinou 33, 420 (2018) [in Japanese].
[14] L.Onsager, Phys. Rev. 65, 117 (1944).
[15] S. Kullback and R. A. Leibler, The Annals of Mathematical Statistics 22, 79 (1951).
[16] D. Ruelle, Comm. Math. Phys. 9, 267 (1968).
[17] R.B.Griffiths, J. Math. Phys. 8, 478 (1967).
[18] R.B.Griffiths, Commun. Math. Phys. 6, 121 (1967).
[19] F.J.Dyson, Commun. Math. Phys. 12, 91 (1969).
[20] K.Aoki, T.Kobayashi, and H.Tomita, Prog. Theor. Phys. 119, 509 (2008).
[21] K.Fukushima, Biological Cybernetics 36, 193 (1980).
[22] Y. LeCun, B. Boser, J. S. Denker, and D. Henderson, Neural computation 1, 541 (1989).
[23] U.Wolff, Phys. Rev. Lett. 62, 361 (1989).
[24] M.Abadi, A.Agarwal, P.Barham, E.Brevdo, Z.Chen, C.Citro, G.S.Corrado, A.Davis, J.Dean, M.Devin, S.Ghemawat, I.Goodfellow, A.Harp, G.Irving, M.Isard, Y.Jia, R.Jozefowicz, L.Kaiser, M.Kudlur, J.Levenberg, D.Mane, R.Monga, S.Moore, D.Murray, C.Olah, M.Schuster, J.Shlens, B.Steiner, I.Sutskever, K.Talwar, P.Tucker, V.Vanhoucke, V.Vasudevan, F.Viegas, O.Vinyals, P.Warden, M.Wattenberg, M.Wicke, Y.Yu, and X.Zheng, arXiv:1603.04467.
[25] D.P.Kingma and J.Ba, arXiv:1412.6980 (2014).
[26] B. Irie, and S. Miyake, IEEE International Conference on Neural Networks, 1641 (1988)
[27] K.Funahashi, Neural Networks, 2, 183 (1989).
[28] K. Hornik, M. Stinchcombe, and H. White, Neural Networks, 2, 359 (1989)

[29] G. Cybenko, Mathematics of Control, Signals, and Systems, 2, 303 (1989)

[30] A.R. Barron, IEEE Transactions on Information Theory, 39, 930 (1993)