Effective fine-tuning training of deep Boltzmann machine based on spatial Monte Carlo integration

Tomu Katsumata\textsuperscript{1} and Muneki Yasuda\textsuperscript{1a)}

\textsuperscript{1} Graduate School of Science and Engineering, Yamagata University
4-3-16 Jyounan, Yonezawa, Yamagata 992-8510, Japan

\textsuperscript{a)} muneki@yz.yamagata-u.ac.jp

Received December 24, 2020; Revised March 13, 2021; Published July 1, 2021

Abstract: A deep Boltzmann machine (DBM) is a probabilistic deep learning model; DBM learning consists of pretraining and fine-tuning stages. This study focuses on the fine-tuning stage, and it develops a new and effective fine-tuning method based on spatial Monte Carlo integration (SMCI), which is an extension of the standard Monte Carlo integration (MCI). It has been proved that SMCI is statistically more accurate than the standard MCI. Fine-tuning methods based on first-order and semi-second-order SMCI methods are formulated. The numerical experiments demonstrate that the proposed fine-tuning methods are superior to the conventional method in terms of both training and generalization errors.

Key Words: deep Boltzmann machine, approximate learning, fine-tuning, spatial Monte Carlo integration

1. Introduction

Recent studies on deep learning [1] have introduced two types of models: deterministic and stochastic models. Deterministic deep learning models include deep neural networks and constitutional neural networks, whereas stochastic deep learning models include deep belief networks (DBNs) [2] and deep Boltzmann machines (DBMs) [3]. DBM is an extension of DBN, which can be viewed as a stacked model of restricted Boltzmann machines (RBMs). This study focuses on the learning algorithm of DBM. DBM is a probabilistic deep learning model based on a pairwise Boltzmann machine [3, 4]; it has been applied to various tasks, for example, 3D image recognition [5], speech recognition [6], emotion recognition [7], feature extraction [8], generative model [9], multimodal learning [10], and prediction of total organic carbon content [11]. A heuristic optimization method for the hyperparameters of DBM such as the sizes of hidden layers and the learning rate was proposed [12].

DBM learning comprises pretraining and fine-tuning stages; pretraining is layer-wise (unsupervised) training based on an RBM, while fine-tuning is a maximum likelihood estimation (MLE) for the entire DBM. The exact evaluations of pretraining and fine-tuning are computationally difficult to perform because they include the evaluation of intractable multiple summations, and therefore, they need an approximation. In the original paper [3], contrastive divergence [13] was used in the pretraining
stage, and an approximation based on a variational approximation (called mean-field approximation) and Monte Carlo integration (MCI) was used in the fine-tuning stage. Pretraining has been actively developed from both technical [14, 15] and theoretical perspectives [16]. However, in contrast, fine-tuning has been insufficiently developed. In fact, many studies have used the original method for fine-tuning [4, 6, 7, 9, 10, 14, 15, 17]. In the best of the authors’ knowledge, a more effective method for fine-tuning has not been proposed. In Ref. [17], although parallel tempering is used instead of the standard Gibbs sampling, it is essentially the same as the original method. This study aims to propose an effective fine-tuning method; it should be noted that “fine-tuning” focused on in Ref. [11] is essentially different from that in this study.

One of the present authors proposed an effective Monte Carlo integration (MCI) for Markov random field (MRF), which is called spatial Monte Carlo integration (SMCI) [18, 19]. In SMCI, the expectation of target variables is evaluated as follows: multiple summation over variables in a region involving target variables is precisely executed, while multiple summations over variables outside of the region is evaluated by the sampling approximation. It has been proved that SMCI is statistically more accurate than MCI, and it is effective for Boltzmann-machine learnings [18–20]. SMCI can be immediately applied to the objective fine-tuning. In the present paper, a new and effective fine-tuning method for DBM is proposed based on SMCI with an appropriate sampling procedure.

The remainder of this paper is organized as follows. DBM and its MLE, i.e., the fine-tuning, are briefly explained in section 2. The SMCI for DBM is described in section 3, in which two different SMCI methods are presented: the first-order (1-SMCI) and semi-second-order SMCI (s2-SMCI) methods; the 1-SMCI method is the simplest version of SMCI and the latter is a higher-order method [19]. The proposed fine-tuning procedure based on SMCI and Gibbs sampling is presented in section 4. In section 5, the proposed fine-tuning methods are validated using numerical experiments with artificial datasets. The numerical results show that the proposed fine-tuning methods based on the 1-SMCI and s2-SMCI methods are superior to the conventional method [3] in terms of both training and generalization errors. Finally, a summary and some discussions are presented in section 6. This paper is an extension of our previous study [21] that includes new improvements: (i) the previous study focused on only the 1-SMCI method, whereas the s2-SMCI method is formulated in this paper; and (ii) more experiments were conducted.

2. Deep Boltzmann machine

Consider a DBM comprising of a visible layer and L hidden layers. The visible layer consists of |V| visible variables \( v \) := \( \{v_i \in \{-1, +1\} \mid i \in V\} \), and the \( \ell \)th (\( 1 \leq \ell \leq L \)) hidden layer consists of \(|H_\ell|\) hidden variables \( h_\ell \) := \( \{h_{\ell,j} \in \{-1, +1\} \mid j \in H_\ell\} \), where \( V \) and \( H_\ell \) are the sets of indices of the variables in the visible layer and in the \( \ell \)th hidden layer, respectively. Figure 1 illustrates a DBM of \( L = 3 \). The energy function is defined by

\[
E_{DBM}(v, h; \theta) := - \sum_{i \in V} b_i v_i - \sum_{\ell=1}^{L} \sum_{j \in H_\ell} c^{(\ell)}_{j} h_{\ell,j} - \sum_{i \in V} \sum_{j \in H_1} w^{(1)}_{j,i} h_{1,j} v_i - \sum_{\ell=2}^{L} \sum_{j \in H_{\ell-1}} \sum_{k \in H_\ell} w^{(\ell)}_{k,j} h_{\ell,k} h_{\ell-1,j},
\]

where \( b_i \) and \( c^{(\ell)}_{j} \) are the bias parameters for \( v_i \) and \( h_{\ell,j} \), respectively, and \( w^{(\ell)}_{k,j} \) is the connection-weight (or interaction) parameter between \( h_{\ell-1,j} \) and \( h_{\ell,k} \) (when \( \ell = 1 \), \( w^{(\ell)}_{k,j} \) is between \( v_j \) and \( h_{1,k} \)). Here, the sets of all parameters and hidden variables are denoted by \( \theta \) and \( h \), respectively. DBM, which is a fully joint distribution over all variables, is the Markov random field with the energy function in Eq. (1),

\[
P_{DBM}(v, h \mid \theta) := \frac{1}{Z_{DBM}(\theta)} \exp \left( - E_{DBM}(v, h; \theta) \right).
\]

Here, \( Z(\theta) \) is the partition function defined by

\[
Z_{DBM}(\theta) := \sum_{v} \sum_{h} \exp \left( - E_{DBM}(v, h; \theta) \right),
\]
Fig. 1. Illustration of DBM with $L = 3$. The undirected connections represent interactive connections.

where $\sum_v$ denotes the sum over all possible realizations of $v$ and $\sum_h := \sum_{h_1} \cdots \sum_{h_L}$ is the sum over all possible realizations of $h$. When $L = 1$, DBM is identical to an RBM [13, 22]; therefore, $L > 1$ is assumed in DBM.

Given a dataset for the visible variables comprising $N$ data points, $D := \{v(\mu) \in \{-1, +1\}^{|V|} | \mu \in \Omega := \{1, 2, \ldots, N\}\}$, the log likelihood of DBM is expressed by

$$
\phi(\theta) := \frac{1}{N} \sum_{\mu \in \Omega} \ln \sum_h p_{DBM}(v(\mu), h | \theta).
$$

The negative of the log likelihood is regarded as the training error. The fine-tuning for DBM (i.e., MLE for DBM) is performed by maximizing the log likelihood with respect to $\theta$. The gradients of the log likelihood with respect to $b_i$ and $c_j^{(\ell)}$ are

$$
\frac{\partial \phi(\theta)}{\partial b_i} = \frac{1}{N} \sum_{\mu \in \Omega} v_i^{(\mu)} - \mathbb{E}_{fr}[v_i | \theta],
$$

$$
\frac{\partial \phi(\theta)}{\partial c_j^{(\ell)}} = \frac{1}{N} \sum_{\mu \in \Omega} \mathbb{E}_{cl}[h_{\ell,j} | v^{(\mu)}, \theta] - \mathbb{E}_{fr}[h_{\ell,j} | \theta],
$$

respectively. The gradient of log likelihood with respect to $w_{k,j}^{(\ell)}$ is

$$
\frac{\partial \phi(\theta)}{\partial w_{k,j}^{(\ell)}} = \frac{1}{N} \sum_{\mu \in \Omega} v_j^{(\mu)} \mathbb{E}_{cl}[h_{1,k} | v^{(\mu)}, \theta] - \mathbb{E}_{fr}[v_j h_{1,k} | \theta]
$$

and it is

$$
\frac{\partial \phi(\theta)}{\partial w_{k,j}^{(\ell)}} = \frac{1}{N} \sum_{\mu \in \Omega} \mathbb{E}_{cl}[h_{\ell,k} h_{\ell-1,j} | v^{(\mu)}, \theta] - \mathbb{E}_{fr}[h_{\ell,k} h_{\ell-1,j} | \theta]
$$

for $2 \leq \ell \leq L$, where $\mathbb{E}_{fr}[\cdots | \theta]$ denotes the expectation of DBM (called free expectation),

$$
\mathbb{E}_{fr}[\cdots | \theta] := \sum_v \sum_h (\cdots) p_{DBM}(v, h | \theta),
$$

and $\mathbb{E}_{cl}[\cdots | v^{(\mu)}, \theta]$ denotes the expectation of the conditional DBM where the visible layer is clamped by $v^{(\mu)}$ (called clamped expectation),

$$
\mathbb{E}_{cl}[\cdots | v^{(\mu)}, \theta] := \sum_h (\cdots) p_{DBM}(h | v^{(\mu)}, \theta),
$$

where

$$
p_{DBM}(h | v^{(\mu)}, \theta) = \frac{p_{DBM}(v^{(\mu)}, h | \theta)}{\sum_h p_{DBM}(v^{(\mu)}, h | \theta)}
$$

is the conditional distribution, which is also a DBM comprising $L$ layers, for which the energy function is
\[
E_{\text{cl}}(h; \psi, \theta) := - \sum_{j \in H_1} \left( c_j^{(1)} + \sum_{i \in V} w_j^{(1)} \psi_i \right) h_{1,j} - \sum_{\ell=2}^L \sum_{j \in H_\ell} c_j^{(\ell)} h_{\ell,j} - \sum_{\ell=2}^L \sum_{k \in H_\ell} \sum_{j \in H_{\ell-1}} w_{k,j}^{(\ell)} h_{\ell,k} h_{\ell-1,j}.
\]

The maximization of Eq. (3) is computationally difficult because the gradients in Eqs. (4)–(7) involve intractable summations over variables. Therefore, this maximization must be performed using an approximation. In the conventional method [3, 4], a variational approximation (called mean-field approximation) and the standard MCI were used: the clamped expectations are evaluated using a variational approximation, while the free expectations are evaluated using the standard MCI with the sample set obtained through a Gibbs sampling procedure similar to persistent contrastive divergence (PCD) [23].

3. Approximation of free and clamped expectations based on SMCI

In SMCI, for the subregion \( T \) is the energy function of the pairwise Boltzmann machine, wherein \( h_i \) is the bias on \( i \in X \) and \( J_{i,j} \) is the symmetric interaction on \( \{i, j\} \in e \) (i.e., \( J_{i,j} = J_{j,i} \)). This pairwise Boltzmann machine includes DBM as the special case. Here, an approximation of the expectation of \( f(x_T) \) is considered, where \( T \) is a (connected) subregion of \( X \) and \( x_T := \{x_i \mid i \in T \subseteq X\} \) denotes the variables in \( T \). In the standard MCI, the expectation of \( f(x_T) \) is approximated by the sample average over a sample set comprising \( m \) sampled points, \( S := \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\} \) generated from \( P(x) \), i.e.,

\[
\mathbb{E}[f(x_T)] := \sum_x f(x_T)P(x) \approx \frac{1}{m} \sum_{s=1}^m f(x_T^{(s)}),
\]

where \( x_T^{(s)} \) is the \( s \)th sampled point on region \( T \).

In SMCI, for the subregion \( T \), a (connected) subregion \( U \) such that \( T \subseteq U \subseteq X \) is selected. The two subregions \( T \) and \( U \) are called “target region” and “sum region,” respectively. For the sum region, a conditional distribution on \( P(x) \):

\[
P(x_U \mid x_{\partial U}) = \frac{P(x)}{\sum_{x_U} P(x)}
\]

is considered, where \( \partial U \) (called “sample region”) denotes the first-nearest-neighboring region of \( U \), i.e., \( \partial U = \{i \mid \{i, j\} \in e, j \in U, i \not\in U\} \). The relationship among the three subregions is illustrated in Fig. 2. With the sample set \( S \), the expectation is approximated by
Fig. 2. Illustration of the target, sum, and sample regions of SMCI.

Fig. 3. Illustration of the sum regions of 1-, s2-, and 2-SMCI methods (the sum regions are shaded): (a) the 1-SMCI method, (b) an example of the s2-SMCI method, and (c) the 2-SMCI method. The variables in region $U \setminus T$ are an independent set in the region in the s2-SMCI method. In general, the choice of the sum region in the s2-SMCI method is not unique.

$$\mathbb{E}[f(x_T)] \approx \frac{1}{m} \sum_{s=1}^{m} \sum_{x_U} f(x_T)P(x_U | x_{\partial U}^{(s)}),$$

where $x_{\partial U}^{(s)}$ denotes the $s$th sampled point corresponding to the sample region. When the sum region covers up to $(k-1)$th-nearest-neighboring regions of $T$, Eq. (14) is referred to as the $k$th-order SMCI ($k$-SMCI) method [18]. The 1-SMCI method is the simplest version of SMCI in which the sum region is identical to the target region; it is expressed as

$$\mathbb{E}[f(x_T)] \approx \frac{1}{m} \sum_{s=1}^{m} \sum_{x_T} f(x_T)P(x_T | x_{\partial T}^{(s)}).$$

Two important properties of SMCI have been proved: for a given $S$, (i) SMCI is statistically more accurate than the standard MCI [18], and (ii) approximation accuracy of SMCI will monotonically increase as the size of the selected sum region increases. The two properties are maintained in general MRFs including higher-order MRFs [18, 19].

However, SMCI has some fundamental drawbacks. SMCI demands to execute multiple summation (or integration) over the sum region. Therefore, the sum region cannot easily expand in dense graphs; only the 1-SMCI and s2-SMCI [19] methods are usable in dense graphs. The s2-SMCI method is an intermediate method between the 1- and 2-SMCI method in which $U \setminus T$ covers the variables that do not have any interactions between them in the first-nearest-neighboring region of $T$; in other words, the variables in region $U \setminus T$ are an independent set in the region (see Fig. 3). It should be noted that the 1-SMCI method cannot be used when the target region is significantly large, with the exception of some special cases (e.g., the target region is a tree).

In the following sections, approximations for the free and clamped expectations appearing in the gradients in Eqs. (4)–(7) based on the 1-SMCI (section 3.2) and s2-SMCI (section 3.3) methods, respectively, are provided. In these methods, the visible layer is regarded as the zeroth hidden layer, i.e., $H_0 = V$, $h_0 = v$, and $c^{(0)} = b$, for simplicity of notation.
3.2 Approximation with the 1-SMCI method

The free and clamped expectations are approximated using the 1-SMCI method given in Eq. (15) [21]. Suppose that $m_{fr}$ sampled points generated from the DBM in Eq. (2),

$$\hat{S}_{fr} := \{\hat{h}_{fr}^{(s)}, \hat{h}_{fr}^{(1)}, \ldots, \hat{h}_{fr}^{(s)} \mid s = 1, 2, \ldots, m_{fr}\},$$

is obtained; the free expectations $E_{fr}[h_{\ell,j} \mid \theta]$ for $0 \leq \ell \leq L$ are approximated as

$$E_{fr}[h_{\ell,j} \mid \theta] \approx M_{\ell,j}(\hat{S}_{fr}) := \frac{1}{m_{fr}} \sum_{s=1}^{m_{fr}} h_{\ell,j}P_{DBM}(h_{\ell,j} \mid \hat{h}_{fr}^{(s)} \mid \theta),$$

(17)

where $\hat{h}_{fr}^{(s)}$ denotes the $s$th sample point corresponding to the sample region in which the target region $T$ consists of only $h_{\ell,j}$, and therefore, the sample region $\partial T$ consists of the first-nearest-neighboring variables of $h_{\ell,j}$ (see Fig. 4(a)). The conditional distribution in Eq. (17) is expressed by

$$P_{DBM}(h_{\ell,j} \mid h_{\partial T}, \theta) \propto \exp \{\{c_{\ell,j}^{(f)} + \beta_{j}^{(f)}(h_{\ell-1}) + \gamma_{j}^{(f)}(h_{\ell+1}))h_{\ell,j}\},$$

(18)

where $h_{\partial T}$ denotes the variables in the sample region; here,

$$\beta_{j}^{(f)}(h_{\ell-1}) := \begin{cases} \sum_{i \in h_{\ell-1}} w_{j;i}^{(f)}h_{\ell-1;i} & (0 \leq \ell \leq L) \\ 0 & \text{(otherwise)} \end{cases}$$

and

$$\gamma_{j}^{(f)}(h_{\ell+1}) := \begin{cases} \sum_{k \in h_{\ell+1}} w_{j;k}^{(f)}h_{\ell+1;k} & (0 \leq \ell < L) \\ 0 & \text{(otherwise)} \end{cases}$$

$\beta_{j}^{(f)}(h_{\ell-1})$ and $\gamma_{j}^{(f)}(h_{\ell+1})$ denote the effective biases from the lower and upper layers to the target variable, respectively. From Eqs. (17) and (18),

$$M_{\ell,j}(\hat{S}_{fr}) = \frac{1}{m_{fr}} \sum_{s=1}^{m_{fr}} \tanh (c_{j}^{(f)} + \beta_{j}^{(f)}(\hat{h}_{\ell-1}) + \gamma_{j}^{(f)}(\hat{h}_{\ell+1}))$$

(19)

is obtained. For the sake of simple formulation, the expressions in Eqs. (18) and (19) (and the expressions appearing in the following and subsequent sections) include undefined variables such as $h_{-1}$ and $h_{L+1}$; they are just dummy variables and should be ignored.

The free expectations $E_{fr}[h_{\ell,k}, h_{\ell-1,j} \mid \theta]$ for $0 < \ell \leq L$ are approximated as

$$E_{fr}[h_{\ell,k}, h_{\ell-1,j} \mid \theta] \approx C_{\ell,k,j}(\hat{S}_{fr}) := \frac{1}{m_{fr}} \sum_{s=1}^{m_{fr}} \sum_{h_{\ell,k}, h_{\ell-1,k}} h_{\ell,k}h_{\ell-1,j}P_{DBM}(h_{\ell,k}, h_{\ell-1,j} \mid \hat{h}_{fr}^{(s)} \mid \theta),$$

(20)

where the target region consists of $h_{\ell,k}$ and $h_{\ell-1,j}$ (see Fig. 4(b)). The conditional distribution in Eq. (20) is expressed by

$$P_{DBM}(h_{\ell,k}, h_{\ell-1,j} \mid h_{\partial T}, \theta) \propto \exp \{\{c_{\ell,k}^{(f)} + \beta_{\ell,k}^{(f)}(h_{\ell-1}) + \gamma_{\ell,k}^{(f)}(h_{\ell+1}))h_{\ell,k} + (c_{\ell-1,k}^{(f)} + \beta_{\ell-1,k}^{(f)}(h_{\ell-2}) + \gamma_{\ell-1,k}^{(f)}(h_{\ell}))h_{\ell-1,j} + w_{\ell,k}^{(f)}h_{\ell,k}h_{\ell-1,j}\},$$

(21)

where

$$\beta_{\ell,k}^{(f)}(h_{\ell-1}) := \begin{cases} \beta_{\ell,k}^{(f)}(h_{\ell-1}) - w_{\ell,k}^{(f)}h_{\ell-1,j} & (0 \leq \ell \leq L) \\ 0 & \text{(otherwise)} \end{cases}$$

and

$$\gamma_{\ell,k}^{(f)}(h_{\ell+1}) := \begin{cases} \gamma_{\ell,k}^{(f)}(h_{\ell+1}) - w_{\ell+1,k}^{(f)}h_{\ell+1,k} & (0 \leq \ell < L) \\ 0 & \text{(otherwise)} \end{cases}$$

Note that $\beta_{\ell-1,k}^{(f)}(h_{\ell-1})$ does not depend on $h_{\ell-1,j}$, and $\gamma_{\ell+1,k}^{(f)}(h_{\ell+1})$ does not depend on $h_{\ell+1,k}$. From Eqs. (20) and (21),
as follows: the sum regions consist of this section. In DBM, the sum region can be expanded without an exponential increase in the
Approximations of the free and clamped expectations using s2-SMCI method \cite{19} are described in 3.3 Approximation with the s2-SMCI method
\[
E_{\ell,j}^{(\mu)} \text{ according to Eq. (14),}
\]
are approximated as
\[
\tilde{E}_{\ell,j}^{(\mu)} = \frac{1}{m_{\text{fr}}} \sum_{s=1}^{m_{\text{fr}}} \text{tanh} \left( \text{atanh} \left( \text{tan} \left( \Gamma_{k,j}^{(\ell)}(\hat{h}_{\ell-1}^{(s)}, \hat{h}_{\ell+1}^{(s)}) \right) \right) \right) + w_{\ell,j}^{(\ell)},
\]
is obtained, where \text{atanh} denotes the inverse hyperbolic tangent function, and
\[
\Gamma_{k,j}^{(\ell)}(\hat{h}_{\ell-1}, \hat{h}_{\ell+1}) := c_j^{(\ell)} + \beta_{k,j}^{(\ell)}(\hat{h}_{\ell-1}) + \gamma_{k,j}^{(\ell)}(\hat{h}_{\ell+1}),
\]
\[
B_{k,j}^{(\ell)}(\hat{h}_{\ell-1}, \hat{h}_{\ell+1}) := c_j^{(\ell)} + \beta_{k,j}^{(\ell)}(\hat{h}_{\ell-1}) + \gamma_{k,j}^{(\ell)}(\hat{h}_{\ell+1}).
\]
The clamped expectations are approximated in a similar manner to the above approximations. Given \( m_{\text{cl}} \) sampled points generated from the conditional DBM conditioned on \( \nu^{(\mu)} \) in Eq. (8),
\[
\tilde{E}_{\ell,j}^{(\mu)} := \left\{ \tilde{h}_1^{(\mu,s)}, \tilde{h}_2^{(\mu,s)}, \ldots, \tilde{h}_L^{(\mu,s)} \right\} \quad s = 1, 2, \ldots, m_{\text{cl}},
\]
the clamped expectations, \( E_{\text{cl}}[h_{\ell,j} | \nu^{(\mu)}, \theta] \) for \( 1 \leq \ell \leq L \) and \( E_{\text{cl}}[h_{\ell,j} | h_{\ell-j,1} | \nu^{(\mu)}, \theta] \) for \( 1 < \ell \leq L \), are approximated as
\[
E_{\text{cl}}[h_{\ell,j} | \nu^{(\mu)}, \theta] \approx M_{\ell,j}(\tilde{E}_{\ell,j}^{(\mu)}),
\]
\[
E_{\text{cl}}[h_{\ell,j} | h_{\ell-j,1} | \nu^{(\mu)}, \theta] \approx C_{\ell,j}(\tilde{E}_{\ell,j}^{(\mu)}),
\]
respectively. In Eqs. (24) and (25), \( \tilde{h}_0^{(\mu,s)} \) is fixed to \( \nu^{(\mu)} \) for all \( s \) and \( m_{\text{fr}} \) appearing in the expressions (19) and (22) is replaced with \( m_{\text{cl}} \).

3.3 Approximation with the s2-SMCI method

Approximations of the free and clamped expectations using s2-SMCI method \cite{19} are described in this section. In DBM, the sum region can be expanded without an exponential increase in the computational cost. In the following, only the resultant expressions are described; for the detailed derivations, refer to Ref. [19].

For the free expectations \( E_{\text{fr}}[h_{\ell,j} | \theta] \) for \( 0 \leq \ell \leq L \), the corresponding sum regions \( U \) are defined as follows: the sum regions consist of \( h_{0,j} \) and \( h_1 \) for \( \ell = 0 \), these consist of \( h_{\ell-1}, h_{\ell,j}, \) and \( h_{\ell+1} \) for \( 0 < \ell < L \), and these consist of \( h_{L,j} \) and \( h_{L-1} \) for \( \ell = L \) (see Fig. 5(a)). With these sum regions, and according to Eq. (14), \( E_{\text{fr}}[h_{\ell,j} | \theta] \) are approximated by
\[
M_{\ell,j}(\hat{S}_{\text{fr}}) := \frac{1}{m_{\text{fr}}} \sum_{s=1}^{m_{\text{fr}}} \text{tanh} \left( c_j^{(\ell)} + \Phi_j^{(\ell)}(\hat{h}_{\ell-2}, \hat{h}_{\ell+2}) + \Psi_j^{(\ell)}(\hat{h}_{\ell-2}, \hat{h}_{\ell+2}) \right),
\]
for \( 0 \leq \ell \leq L \), where \( \hat{S}_{\text{fr}} \) denotes the sample set defined in Eq. (16), and
\[
\Phi_j^{(\ell)}(h_{\ell-2}, h_{\ell}) := \left\{ \sum_{i \in H_{\ell-1}} \text{atanh} \left[ \text{tan} \left( w_{j,i}^{(\ell)} \right) \right] \left[ \text{tan} \left( B_{i,j}^{(\ell)}(h_{\ell-2}, h_{\ell}) \right) \right] \right\} \quad (0 < \ell \leq L),
\]
\[
\Psi_j^{(\ell)}(h_{\ell}, h_{\ell+2}) := \left\{ \sum_{k \in H_{\ell+1}} \text{atanh} \left[ \text{tan} \left( w_{j,k}^{(\ell+1)} \right) \right] \left[ \text{tan} \left( \Gamma_{k,j}^{(\ell+1)}(h_{\ell}, h_{\ell+2}) \right) \right] \right\} \quad (0 \leq \ell < L),
\]}
Equation (26) is regraded as the second-order SMCI method (2-SMCI) \cite{18}.

For the free expectations, \( E_f[h_{ℓ,k}h_{ℓ−1,j} | θ] \) for \( 0 < ℓ ≤ L \), the corresponding sum regions \( U \) are defined as follows: the sum regions consist of \( h_{ℓ−1,j}, h_{ℓ,k} \), and \( h_{ℓ+1} \) for \( ℓ = 1 \), these consist of \( h_{ℓ−2}, h_{ℓ−1,j}, h_{ℓ,k} \), and \( h_{ℓ+1} \) for \( 1 < ℓ < L \), and these consist of \( h_{ℓ−2}, h_{ℓ−1,j}, h_{ℓ,k} \) for \( ℓ = L \) (see Fig. 5(b)). With these sum regions, and according to Eq. (14), \( E_f[h_{ℓ,k}h_{ℓ−1,j} | θ] \) are approximated by

\[
C_{ℓ,k,j}(\hat{S}_f) := \frac{1}{m_{tr}} \sum_{s=1}^{m_{tr}} \tanh \left( \text{atanh} \left( \tanh \left( c_{ℓ,j}^{(ℓ−1)} + \gamma_{ℓ,j,k}^{(ℓ−1)}(\hat{h}_ℓ^{(s)} + \Phi_{ℓ}^{(ℓ−1)}(\hat{h}_{ℓ−3}^{(s)}, \hat{h}_{ℓ−1}^{(s)})) \right) \right) \right.
\]

\[
\times \tanh \left( \gamma_{k,j}^{(ℓ)}(\hat{h}_{ℓ−1}^{(s)}) + \Psi_{ℓ}^{(ℓ)}(\hat{h}_{ℓ}^{(s)}, \hat{h}_{ℓ+2}^{(s)}) \right) \right) + w_{k,j}^{(ℓ)}, \tag{27}
\]

for \( 0 < ℓ ≤ L \).

The clamped expectations \( E_{cl}[h_{ℓ,j} | v(μ), θ] \) for \( 1 ≤ ℓ ≤ L \) and \( E_{cl}[h_{ℓ,k}h_{ℓ−1,j} | v(μ), θ] \) for \( 1 < ℓ ≤ L \) are approximated as

\[
E_{cl}[h_{ℓ,j} | v(μ), θ] \approx M_{ℓ,j}(\hat{S}_{cl}^{(μ)}), \tag{28}
\]

\[
E_{cl}[h_{ℓ,k}h_{ℓ−1,j} | v(μ), θ] \approx C_{ℓ,k,j}(\hat{S}_{cl}^{(μ)}), \tag{29}
\]

respectively, where \( \hat{S}_{cl}^{(μ)} \) denotes the sample set defined in Eq. (23). In Eqs. (28) and (29), \( \hat{h}_{0}^{(μ,s)} \) is fixed to \( v(μ) \) for all \( s \) and \( m_{tr} \) appearing in the expressions (26) and (27) is replaced with \( m_{cl} \).

The s2-SMCI method allows different choices for the sum regions. The presented sum regions are selected to let the regions be larger. However, the presented region-selection may not be appropriate in cases such as when the sizes of layers vary widely. In addition, in a three-layered case, the s2-SMCI method for \( E_{cl}[h_{ℓ,k}h_{ℓ−1,j} | v(μ), θ] \) will be identical as the 1-SMCI method with the presented region-selection because the conditional DBM in Eq. (8) has only two layers; thus, an alternative region-selection-strategy is required to apply the s2-SMCI method to the three-layered case.

4. Proposed fine-tuning procedure

The MLE of DBM is computationally difficult because of the intractable expectations (the free and clamped expectations) in the gradients of the log likelihood. To obtain a practical algorithm, the evaluation of the intractable expectations is avoided using the approximations presented in sections 3.2 and 3.3; for example, based on the 1-SMCI method presented in section 3.2, the gradients in Eqs. (5) and (6) can be approximated as

\[
\frac{∂φ(θ)}{∂c_{ℓ,j}^{(ℓ)}} \approx \frac{1}{N} \sum_{μ∈Ω} M_{ℓ,j}(\hat{S}_{cl}^{(μ)}) - M_{ℓ,j}(\hat{S}_f),
\]

\[
\frac{∂φ(θ)}{∂w_{k,j}^{(ℓ)}} \approx \frac{1}{N} \sum_{μ∈Ω} C_{ℓ,k,j}(\hat{S}_{cl}^{(μ)}) - C_{ℓ,k,j}(\hat{S}_f),
\]

384
Algorithm 1 Pseudocode of the proposed algorithm (based on the 1-SMCI method)

1: Input training dataset \( D := \{v(\mu) \in \{-1, +1\}^{|V|} | \mu \in \Omega\} \)
2: Divide \( D \) into \( R \) different mini-batches: \( B_1, B_2, \ldots, B_R \subseteq \Omega \)
3: Initialize \( \theta \)
4: Initialize \( \hat{S}_{fr} \) and \( \bar{S}_{cl}(\mu) \) for \( \mu \in \Omega \)
5: \( T = 0 \)
6: repeat
7: \hspace{1em} for \( r = 1, 2, \ldots, R \) do
8: \hspace{2em} Update \( \theta \) by using a stochastic gradient ascent method with the following gradients:
9: \hspace{3em} \( \frac{\partial \phi(\theta)}{\partial b_i} \approx \frac{1}{|B_r|} \sum_{\mu \in B_r} v_i^{(\mu)} - M_{0,i}(\hat{S}_r) \),
10: \hspace{3em} \( \frac{\partial \phi(\theta)}{\partial c_{j}^{(l)}} \approx \frac{1}{|B_r|} \sum_{\mu \in B_r} M_{\ell,j}^{(l)}(\bar{S}_{cl}^{(\mu)}) - M_{\ell,j}(\hat{S}_r) \) \( (0 < \ell \leq L) \),
11: \hspace{3em} \( \frac{\partial \phi(\theta)}{\partial w_{k,j}^{(1)}} \approx \frac{1}{|B_r|} \sum_{\mu \in B_r} v_j^{(\mu)} M_{1,k}^{(l)}(\bar{S}_{cl}^{(\mu)}) - C_{1,k,j}^{(l)}(\hat{S}_r) \),
12: \hspace{3em} \( \frac{\partial \phi(\theta)}{\partial w_{k,j}^{(\ell)}} \approx \frac{1}{|B_r|} \sum_{\mu \in B_r} C_{\ell,k,j}^{(l)}(\bar{S}_{cl}^{(\mu)}) - C_{\ell,k,j}^{(l)}(\hat{S}_r) \) \( (1 < \ell \leq L) \).
13: \hspace{2em} end for
14: \hspace{1em} \( T \leftarrow T + 1 \)
15: until \( T \leq T_{max} \)
16: Output \( \theta \)

respectively. This is the basic idea of our proposed method. The proposed approximations based on the 1-SMCI and s2-SMCI methods require the sampled sets \( \hat{S}_r \) and \( \bar{S}_{cl}^{(\mu)} \) in Eqs. (16) and (23), respectively. Based on the original study [3], a Gibbs sampling procedure similar to PCD is employed. Algorithm 1 shows the pseudocode of the proposed fine-tuning procedure based on a mini-batch-type stochastic gradient. The one-cycle Gibbs sampling in Algorithm 1 comprises the forward Gibbs sampling from the bottom layer to the top layer and backward Gibbs sampling from the top layer to the bottom layer (see Fig. 6). Different sample points in the sample set (\( \hat{S}_r \) or \( \bar{S}_{cl}^{(\mu)} \)) are generated from parallel Gibbs sampling procedures starting from different initial states. For example, \( m_{fr} \) parallel one-cycle Gibbs samplings are performed to update \( \hat{S}_r \). The pseudocodes of the one-cycle Gibbs sampling procedures for \( \hat{S}_r \) and \( \bar{S}_{cl}^{(\mu)} \) are shown in Algorithm 2 and 3, respectively.

The pseudocode in Algorithm 1 is described based on the 1-SMCI method. If the s2-SMCI method is used instead of the 1-SMCI method, \( M_{\ell,j}(\cdots) \) and \( C_{\ell,k,j}(\cdots) \) in Eqs. (26) and (27) are used instead.
of $M _ { f,i} ( \cdot \cdot \cdot )$ and $C _ { l,k,j} ( \cdot \cdot \cdot )$ in Eqs. (19) and (22), respectively. The order of the computational costs of the proposed methods based on both 1-SMCI and s2-SMCI methods are the same as that of the original method [3].

5. Numerical experiment

This section presents the results of numerical experiments using an artificial dataset $\mathcal{D}$ consisting of $N = 500$ data points, which was generated from a generative DBM (gDBM), $P _ { \text{gen}}(v, h)$, using Gibbs sampling. The procedure for generating the artificial dataset is as follows. Using Gibbs sampling (which is the same procedure as used for Algorithm 2) on gDBM, $N$ independent sample points for $v$ and $h$, $\{(v^{(\mu)}, h^{(\mu)}) \mid \mu = \Omega\}$, were generated. The artificial dataset was obtained by taking the sample points corresponding only to the visible layer, i.e., $\mathcal{D} = \{v^{(\mu)} \mid \mu \in \Omega\}$. In gDBM, the interaction parameters between $\ell$th and $(\ell - 1)$th hidden layers were randomly generated from a normal distribution with mean zero and variance $10/(H_\ell + H_{\ell-1})$ (here, the visible layer is regarded as the zeroth hidden layer), and all bias parameters were fixed to zero. With the artificial dataset, a training DBM (tDBM) $P _ { \text{DBM}}(v, h \mid \theta)$ was trained using different learning algorithms: the conventional method [3] and the proposed methods (based on the 1-SMCI and s2-SMCI methods). In tDBM, all interaction parameters were initialized using Glorot’s method [24], and all bias parameters were fixed to zero during training for simplicity.

In the proposed methods, the sizes of sample sets $\hat{\mathcal{S}}_f$ and $\hat{\mathcal{S}}_{c_l}^{(\mu)}$ for each $\mu \in \Omega$ were set to $m_{f} = 500$ and $m_{c_l}^{(\mu)} = 1$, respectively, and the sample sets were initialized using the 1000-cycle Gibbs sampling relaxations. Further, $\hat{\mathcal{S}}_f$ was used in the conventional method to evaluate the free expectations. During the training of all methods, the adamax optimizer [25] with a mini-batch size of 100 was used.

The accuracy of the learning was measured with both training and generalization errors. The training error is obtained from the log likelihood given in Eq. (3), and the generalization error is obtained from the Kullback–Leibler divergence (KLD) between gDBM and tDBM, i.e.,
require a computational cost of clamped expectations can be precisely evaluated. The exact evaluations of the errors and expectations depend not only on the approximation accuracy but also on the quality of the dataset. When the sizes (the number of variables) of gDBM and tDBM are large, these errors are not evaluated precisely because of the high computational cost; therefore, in the experiments, small-sized models were used wherein the free and clamped expectations can achieve a higher log likelihood does not necessarily achieve a lower KLD (this is known as the over-fitting problem). The improvement in approximation accuracy of the learning algorithm should reduce the training error; however, whether it reduces the generalization error depends not only on the approximation accuracy but also on the quality of the dataset. When the sizes (the number of variables) of gDBM and tDBM are large, these errors are not evaluated precisely because of the high computational cost; therefore, in the experiments, small-sized models were used wherein the free and clamped expectations can be precisely evaluated. The exact evaluations of the errors and expectations require a computational cost of $O(2^n)$, where $n$ is the total number of variables; therefore, $n \approx 30$ is the limit in terms of the computational resource.

First, the results for three-layered models, i.e., $L = 2$, are shown, wherein the sizes of the layers of gDBM were set to $|V| = |H_1| = |H_2| = 5$ and tDBM had the same structure, i.e., both models are matched. Figure 7(a) presents KLD (namely, generalization error) against the training epoch, while Fig. 7(b) presents the log likelihood (namely, negative training error) against the training epoch. In the figures, “exact” denotes the results obtained using exact gradients, and “conventional” denotes those obtained using the conventional method [3]; “1-SMCI” denotes the results obtained using the proposed methods based on the 1-SMCI method. The results of the proposed method based on the 1-SMCI method are superior to those obtained using the conventional method in terms of both training and generalization errors.

Next, the results for four-layered models, i.e., $L = 3$, are shown, in which the sizes of the layers of gDBM were set to $|V| = 5$ and $|H_1| = |H_2| = |H_3| = 8$, and those of tDBM were set to $|V| = |H_1| = |H_2| = |H_3| = 5$, i.e., both models are mismatched. The results are shown in Figs. 8(a) and (b). In the figures, “s2-SMCI” denotes the results obtained using the proposed methods based on the s2-SMCI method. The method based on the s2-SMCI method improves the training error (while, in

### Algorithm 3 One-cycle Gibbs sampling for $S_{cl}^{(μ)}$

1. **Input** $S_{cl}^{(μ)} = \{h_1^{(μ,s)}, h_2^{(μ,s)}, \ldots, h_L^{(μ,s)} \mid s = 1, 2, \ldots, m_{cl}\}$

2. **procedure** **Forward Gibbs Sampling**:

   3. **for** $s = 1, 2, \ldots, m_{cl}$ **do**
      
      \[
      \hat{h}_1^{(μ,s)} \leftarrow P_{DBM}(h_1 \mid v_1^{(μ,s)}, h_2^{(μ,s)}, \theta)
      \]
      \[
      \hat{h}_2^{(μ,s)} \leftarrow P_{DBM}(h_2 \mid h_1^{(μ,s)}, \theta)
      \]
      \[
      \hat{h}_L^{(μ,s)} \leftarrow P_{DBM}(h_L \mid h_{L-1}^{(μ,s)}, \theta)
      \]

   4. **end for**

5. **procedure** **Backward Gibbs Sampling**:

   7. **for** $s = 1, 2, \ldots, m_{cl}$ **do**
      
      \[
      \hat{h}_{L-1,b}^{(μ,s)} \leftarrow P_{DBM}(h_{L-1} \mid \hat{h}_{L-2,b}^{(μ,s)}, \hat{h}_L^{(μ,s)}, \theta)
      \]
      \[
      \hat{h}_{L,b}^{(μ,s)} \leftarrow P_{DBM}(h_{L} \mid h_{L-1}^{(μ,s)}, \theta)
      \]
      \[
      \hat{h}_{1,b}^{(μ,s)} \leftarrow P_{DBM}(h_{1} \mid \hat{h}_{2,b}^{(μ,s)}, \theta)
      \]

   8. **end for**

9. **end procedure**

10. **Update** $S_{cl}^{(μ)} \leftarrow \{h_1^{(μ,s)}, h_2^{(μ,s)}, \ldots, h_{L-1,b}^{(μ,s)}, \hat{h}_L^{(μ,s)} \mid s = 1, 2, \ldots, m_{cl}\}$

11. **Output** $S_{cl}^{(μ)}$

\[
\text{KLD} := \sum_v P_{gen}(v) \ln \frac{P_{gen}(v)}{P_{DBM}(v \mid \theta)},
\]

where $P_{gen}(v)$ and $P_{DBM}(v \mid \theta)$ are the marginal distribution of gDBM and tDBM, respectively. The training and generalization errors measure different properties, which means that an algorithm that can achieve a higher log likelihood does not necessarily achieve a lower KLD (this is known as the over-fitting problem). The improvement in approximation accuracy of the learning algorithm should reduce the training error; however, whether it reduces the generalization error depends not only on the approximation accuracy but also on the quality of the dataset. When the sizes (the number of variables) of gDBM and tDBM are large, these errors are not evaluated precisely because of the high computational cost; therefore, in the experiments, small-sized models were used wherein the free and clamped expectations can be precisely evaluated. The exact evaluations of the errors and expectations require a computational cost of $O(2^n)$, where $n$ is the total number of variables; therefore, $n \approx 30$ is the limit in terms of the computational resource.

First, the results for three-layered models, i.e., $L = 2$, are shown, wherein the sizes of the layers of gDBM were set to $|V| = |H_1| = |H_2| = 5$ and tDBM had the same structure, i.e., both models are matched. Figure 7(a) presents KLD (namely, generalization error) against the training epoch, while Fig. 7(b) presents the log likelihood (namely, negative training error) against the training epoch. In the figures, “exact” denotes the results obtained using exact gradients, and “conventional” denotes those obtained using the conventional method [3]; “1-SMCI” denotes the results obtained using the proposed methods based on the 1-SMCI method. The results of the proposed method based on the 1-SMCI method are superior to those obtained using the conventional method in terms of both training and generalization errors.

Next, the results for four-layered models, i.e., $L = 3$, are shown, in which the sizes of the layers of gDBM were set to $|V| = 5$ and $|H_1| = |H_2| = |H_3| = 8$, and those of tDBM were set to $|V| = |H_1| = |H_2| = |H_3| = 5$, i.e., both models are mismatched. The results are shown in Figs. 8(a) and (b). In the figures, “s2-SMCI” denotes the results obtained using the proposed methods based on the s2-SMCI method. The method based on the s2-SMCI method improves the training error (while, in
Fig. 7. Results for the three-layered case: (a) KLD (lower is better) and (b) log likelihood (higher is better) versus the training epoch. These plots present the average values over 500 experiments.

Fig. 8. Results for the four-layered case: (a) KLD and (b) log likelihood versus the training epoch. These plots present the average values over 1000 experiments.

terms of the generalization error, this method had almost the same results as the method based on the 1-SMCI method. The training error of the conventional method drastically degrades as the training proceeds; this might support the result obtained in Ref. [3], that is, the performance of four-layered DBM tends to be lower than that of three-layered DBM in several classification tasks.

In Fig. 8(b), the likelihood obtained by the conventional method monotonically increased in the early stage and drastically decreased in the later stage. This behavior might be caused by the mean-field approximation in the conventional method. The mean-field approximation is a variational approximation that only considers rough structures and ignores precise structures in the model. In the early stage of learning, the learning parameters are optimized to fit the rough structures; subsequently, the optimization for fitting to the precise structures starts. The information of precise structures, that the mean-field approximation ignores, causes the log-likelihood degradation in the later stages. However, such behavior is not seen in Fig. 7(b). This means that the relative importance of the precise structures increases with a higher DBM.

The standard Gibbs sampling is used in this study; however, there exist more sophisticated sampling methods such as parallel tempering (or replica exchange MCMC) [26, 27] or Suwa-Todo method [28] and belief-propagation-guided MCMC [29]. These methods are expected to improve the performance of the proposed fine-tuning methods.

6. Conclusions
In this paper, new and effective fine-tuning methods for DBM were proposed based on the 1- and s2-SMCI methods. Their validities were demonstrated via numerical experiments using an artificial
The results of the numerical experiments indicated that both proposed methods are superior to the conventional method in terms of both training and generalization errors, and the method based on the s2-SMCI method is superior to that based on the 1-SMCI method. Although the proposed methods are the same as the conventional one in terms of the order of computational cost with respect to the sizes of the system and of the dataset, the method based on the s2-SMCI method is particularly slower compared to the others in terms of CPU time. This is an obvious drawback of the s2-SMCI method. In practical situations, a combination of the 1-SMCI and s2-SMCI methods might be effective (for example, after specific epochs of learning using the 1-SMCI method, the s2-SMCI methods is used for fine-tuning the learning).

This study focused on only the fine-tuning stage. However, the pretraining stage is also important to complete DBM learning, and it is believed that combining the proposed fine-tuning methods with an appropriate pretraining method will lead to effective DBM learning. This is an important future research direction. An improvement of the sampling procedure is another alternative research direction.

Acknowledgments

This work was partially supported by JSPS KAKENHI (Grant Numbers 18K11459 and 18H03303), JST CREST (Grant Number JPMJCR1402), and the COI Program of JST (Grant Number JP-MJCE1312).

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[2] G.E. Hinton, S. Osindero, and Y.W. Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.
[3] R. Salakhutdinov and G.E. Hinton, “Deep Boltzmann machines,” Proc. 12th International Conference on Artificial Intelligence and Statistics, pp. 448–455, 2009.
[4] R. Salakhutdinov and G.E. Hinton, “An efficient learning procedure for deep Boltzmann machines,” Neural Computation, vol. 24, no. 8, pp. 1967–2006, 2012.
[5] B. Leng, X. Zhang, M. Yao, and Z. Xiong, “A 3D model recognition mechanism based on deep Boltzmann machines,” Neurocomputing, vol. 151, pp. 593–602, 2015.
[6] M.R. Alam, M. Bennamoun, R. Togneri, and F. Sohel, “A joint deep Boltzmann machine (jDBM) model for person identification using mobile phone data,” IEEE Transactions on Multimedia, vol. 19, pp. 317–326, 2017.
[7] S. Wang, M. He, Z. Gao, S. He, and Q. Ji, “Emotion recognition from thermal infrared images using deep Boltzmann machine,” Frontiers of Computer Science, vol. 8, pp. 609–618, 2014.
[8] A. Taherkhani, G. Cosma, and T.M. McGinnity, “Deep-FS: A feature selection algorithm for deep Boltzmann machines,” Neurocomputing, vol. 322, pp. 22–37, 2018.
[9] S.M. A. Eslami, N. Heess, C.K.I. Williams, and J. Winn, “The shape Boltzmann machine: A strong model of object shape,” International Journal of Computer Vision, vol. 107, pp. 155–176, 2014.
[10] N. Srivastava and R. Salakhutdinov, “Multimodal learning with deep Boltzmann machines,” Journal of Machine Learning Research 1, vol. 15, pp.949–2980, 2014.
[11] L. Zhu, C. Zhang, C. Zhang, Z. Zhang, X. Nie, X. Zhou, W. Liu, and X. Wang, “Forming a new small sample deep learning model to predict total organic carbon content by combining unsupervised learning with semisupervised learning,” Applied Soft Computing, vol. 83, 105596, 2019.
[12] L.A. Passos and J.P. Papa, “A metaheuristic-driven approach to fine-tune deep Boltzmann machines,” Applied Soft Computing, vol. 97, 105717, 2020.
[13] G.E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Computation, vol. 14, no. 8, pp. 1771–1800, 2002.
[14] G.E. Hinton and R. Salakhutdinov, “A better way to pretrain deep Boltzmann machines,” Proceedings of the Advances in Neural Information Processing Systems 25, pp. 2447–2455, 2012.

[15] K. Cho, T. Raiko, A. Ilin, and J. Karhunen, “A two-stage pretraining algorithm for deep Boltzmann machines,” Proceedings of the 23rd International Conference on Artificial Neural Networks, pp. 106–113, 2013.

[16] M. Yasuda, “Relationship between pretraining and maximum likelihood estimation in deep Boltzmann machines,” Proc. 19th International Conference on Artificial Intelligence and Statistics, pp. 582–590, 2016.

[17] K. Cho, T. Raiko, and A. Ilin, “Gaussian-Bernoulli deep Boltzmann machine,” Proc. 2013 International Joint Conference on Neural Networks, pp. 1–7, 2013.

[18] M. Yasuda, “Monte Carlo integration using spatial structure of Markov random field,” Journal of the Physical Society of Japan, vol. 84, no. 3, 034001, 2015.

[19] M. Yasuda and K. Uchizawa, “A generalization of spatial Monte Carlo integration,” Neural Computation, vol. 33, no. 4, pp. 1037–1062, 2021.

[20] M. Yasuda, “Learning algorithm of Boltzmann machine based on spatial Monte Carlo integration method,” Algorithms, vol. 11, no. 4, p. 42, 2018.

[21] T. Katsumata and M. Yasuda, “Effective fine-tuning algorithm for deep Boltzmann machine,” Proc. 2020 International Symposium on Nonlinear Theory and its Applications, pp. 29–32, 2020.

[22] P. Smolensky, “Information processing in dynamical systems: foundations of harmony theory,” Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, pp. 194–281, 1986.

[23] T. Tieleman, “Training restricted Boltzmann machines using approximations to the likelihood gradient,” Proc. 25th International Conference on Machine Learning, pp. 1064–1071, 2008.

[24] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” Proc. 13th International Conference on Artificial Intelligence and Statistics, vol. 9, pp. 249–256, 2010.

[25] D.P. Kingma and L.J. Ba, “Adam: A method for stochastic optimization,” Proc. 3rd International Conference on Learning Representations, pp. 1–13, 2015.

[26] R.H. Swendsen and J.S. Wang, “Replica monte carlo simulation of spin-glasses,” Phys. Rev. Lett., vol. 57, no. 21, pp. 2607–2609, 1986.

[27] K. Hukushima and K. Nemoto, “Exchange monte carlo method and application to spinglass simulations,” J. Phys. Soc. Jpn., vol. 65, no. 6, pp. 1604–1608, 1996.

[28] H. Suwa and S. Todo, “Markov chain monte carlo method without detailed balance,” Phys. Rev. Lett., vol. 105, 20603, 2010.

[29] A. Decelle and F. Krzakala, “Belief-propagation-guided Monte-Carlo sampling,” Phys. Rev. B, vol. 89, no. 21, 214421, 2014.