



Restricted Boltzmann Machine with Multivalued Hidden Variables

A Model Suppressing Over-Fitting

Yuuki Yokoyama¹ · Tomu Katsumata² · Muneki Yasuda² 

Received: 13 February 2019 / Accepted: 13 June 2019 / Published online: 27 June 2019
© The Author(s) 2019

Abstract

Generalization is one of the most important issues in machine learning problems. In this study, we consider generalization in restricted Boltzmann machines (RBMs). We propose an RBM with multivalued hidden variables, which is a simple extension of conventional RBMs. We demonstrate that the proposed model is better than the conventional model via numerical experiments for contrastive divergence learning with artificial data and a classification problem with MNIST.

Keywords Statistical machine learning · Restricted Boltzmann machine · Pattern recognition · Generalization

1 Introduction

Generalization is one of the most important goals in statistical machine learning problems [1]. In various standard machine learning techniques, given a particular data set, we fit our probabilistic learning model to the empirical distribution (or the data distribution) of the data set. When our learning model is sufficiently flexible, it can fit the empirical distribution exactly via an appropriate learning method. A learning model that is too close to the empirical distribution frequently gives poor results for new data points. This situation is known as over-fitting. Over-fitting impedes generalization; therefore, techniques that can suppress over-fitting are needed to achieve good generalizations. Regularizations, such as L_1 and L_2 regularizations or their combination (the elastic net) [18], are popular techniques used for this purpose.

✉ Muneki Yasuda
muneki@yz.yamagata-u.ac.jp

¹ ALBERT Inc., Tokyo, Japan

² Graduate School of Science and Engineering, Yamagata University, Yamagata, Japan

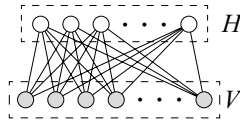


Fig. 1 Bipartite graph consisting of two layers: the visible layer and the hidden layer. V and H are the sets of indices of the nodes in the visible layer and hidden layer, respectively

Here, we focus on a restricted Boltzmann machine (RBM) [5, 15]. RBMs have a wide range of applications such as collaborating filtering [14], classification [8], and deep learning [6, 12, 13]. The suppression of over-fitting is also important in RBMs. An RBM is a probabilistic neural network defined on a bipartite undirected graph comprising two different layers: visible layer and hidden layer. The visible layer, which consists of visible (random) variables, directly corresponds to the data points, while the hidden layer, which consists of hidden (random) variables, does not. The hidden layer creates complex correlations among the visible variables. The sample space of the visible variables is determined by the range of data elements, whereas the sample space of the hidden variables can be set freely. Typically, the hidden variables are given binary values ($\{0, 1\}$ or $\{-1, +1\}$).

In this study, we propose an RBM with multivalued hidden variables. The proposed RBM is a very simple extension of the conventional RBM with binary-hidden variables (referred to as the binary-RBM in this paper). However, we demonstrate that the proposed RBM is better than the binary-RBM in terms of suppressing the over-fitting. The remainder of this paper is organized as follows. We define the proposed RBM in Sect. 2 and explain its maximum likelihood estimation in Sect. 2.1. In Sect. 2.2, we demonstrate the validity of the proposed RBM using numerical experiments for contrastive divergence (CD) learning [5] with artificial data. We give an insight on the effect of our extension (i.e., the effect of multivalued hidden variables) using a toy example in Sect. 2.3. In Sect. 3, we apply the proposed RBM to a classification problem and show that it is also effective in such type of problems. Finally, the conclusion is given in Sect. 4.

2 Restricted Boltzmann Machine with Multivalued Hidden Variables

Let us consider a bipartite graph consisting of two different layers: the visible layer and hidden layer, as shown in Fig. 1. Binary (or bipolar) visible variables, $\mathbf{v} := \{v_i \in \{-1, +1\} \mid i \in V\}$, are assigned to the corresponding nodes in the visible layer. The corresponding hidden variables, $\mathbf{h} := \{h_j \in \mathcal{X}(s) \mid j \in H\}$, are assigned to the nodes in the hidden layer, where $\mathcal{X}(s)$ is the sample space defined by

$$\mathcal{X}(s) := \{(2k - s)/s \mid k = 0, 1, \dots, s\} \quad s \in \mathbb{N}, \quad (1)$$

where \mathbb{N} is the set of all natural numbers. For example, $\mathcal{X}(1) = \{-1, +1\}$, $\mathcal{X}(2) = \{-1, 0, +1\}$, and $\mathcal{X}(3) = \{-1, -1/3, +1/3, +1\}$. Namely, $\mathcal{X}(s)$ is the set of values that evenly partition the interval $[-1, +1]$ into $(s + 1)$ parts. We define that in

the limit of $s \rightarrow \infty$, $\mathcal{X}(s)$ becomes a continuous space $[-1, +1]$, i.e., $\mathcal{X}(\infty) = [-1, +1]$. On the bipartite graph, we define the energy function for $s \in \mathbb{N}$ as

$$E_s(\mathbf{v}, \mathbf{h}; \theta) := - \sum_{i \in V} b_i v_i - \sum_{j \in H} c_j h_j - \sum_{i \in V} \sum_{j \in H} w_{i,j} v_i h_j, \tag{2}$$

where $\{b_i\}$, $\{c_j\}$, and $\{w_{i,j}\}$ are the learning parameters of the energy function, and they are collectively denoted by θ . Specifically, $\{b_i\}$ and $\{c_j\}$ are the biases for the visible and hidden variables, respectively, and $\{w_{i,j}\}$ are the couplings between the visible and hidden variables. Our RBM is defined in the form of a Boltzmann distribution in terms of the energy function given in Eq. (2):

$$P_s(\mathbf{v}, \mathbf{h} \mid \theta) := \frac{\omega(s)}{Z_s(\theta)} \exp(-E_s(\mathbf{v}, \mathbf{h}; \theta)), \tag{3}$$

where

$$Z_s(\theta) := \sum_{\mathbf{v} \in \{-1, +1\}^{|V|}} \sum_{\mathbf{h} \in \mathcal{X}(s)^{|H|}} \omega(s) \exp(-E_s(\mathbf{v}, \mathbf{h}; \theta)) \tag{4}$$

is the partition function. The multiple summations in Eq. (4) mean

$$\begin{aligned} \sum_{\mathbf{v} \in \{-1, +1\}^{|V|}} &= \sum_{v_1 \in \{0,1\}} \sum_{v_2 \in \{0,1\}} \cdots \sum_{v_{|V|} \in \{0,1\}}, \\ \sum_{\mathbf{h} \in \mathcal{X}(s)^{|H|}} &= \sum_{h_1 \in \mathcal{X}(s)} \sum_{h_2 \in \mathcal{X}(s)} \cdots \sum_{h_{|H|} \in \mathcal{X}(s)}. \end{aligned}$$

The factor $\omega(s) := \{2/(s + 1)\}^{|H|}$ appearing in Eqs. (3) and (4) is a constant unrelated to \mathbf{v} and \mathbf{h} . Although it vanishes by reducing the fraction in Eq. (3), we leave it for the sake of the subsequent analysis. The factor lets the summation over h_j be a Riemann sum and prevents the divergence of the partition function in $s \rightarrow \infty$. It is noteworthy that when $s = 1$, Eq. (3) is equivalent to the binary-RBM.

The marginal distribution of RBM is expressed as

$$\begin{aligned} P_s(\mathbf{v} \mid \theta) &= \sum_{\mathbf{h} \in \mathcal{X}(s)^{|H|}} P_s(\mathbf{v}, \mathbf{h} \mid \theta) \\ &= \frac{1}{Z_s(\theta)} \exp\left(\sum_{i \in V} b_i v_i + \sum_{j \in H} \ln \phi_s(\lambda_j(\mathbf{v}, \theta))\right), \end{aligned} \tag{5}$$

where $\lambda_j(\mathbf{v}, \theta) := c_j + \sum_{i \in V} w_{i,j} v_i$ and $\phi_s(x) := \sum_{h \in \mathcal{X}(s)} 2(s + 1)^{-1} e^{xh}$. It is noteworthy that factor $2(s + 1)^{-1}$ in the definition of $\phi_s(x)$ comes from $\omega(s)$. Using the formula of geometric series, we obtain

$$\phi_s(x) = \frac{2 \sinh\{(s + 1)x/s\}}{(s + 1) \sinh(x/s)} \tag{6}$$

for $1 \leq s < \infty$. When $s \rightarrow \infty$, we obtain

$$\phi_{\infty}(x) = \int_{-1}^{+1} e^{xh} dh = \frac{2 \sinh x}{x}. \quad (7)$$

The additive factor, $2(s+1)^{-1}$, ensures that $\lim_{s \rightarrow \infty} \phi_s(x) = \phi_{\infty}(x)$. The conditional distributions are

$$P_s(\mathbf{v} \mid \mathbf{h}, \theta) = \prod_{i \in V} P_s(v_i \mid \mathbf{h}, \theta), \quad P_s(v_i \mid \mathbf{h}, \theta) \propto \exp(\xi_i(\mathbf{h}, \theta)v_i), \quad (8)$$

$$P_s(\mathbf{h} \mid \mathbf{v}, \theta) = \prod_{j \in H} P_s(h_j \mid \mathbf{v}, \theta), \quad P_s(h_j \mid \mathbf{v}, \theta) \propto \exp(\lambda_j(\mathbf{v}, \theta)h_j), \quad (9)$$

where $\xi_i(\mathbf{h}, \theta) := b_i + \sum_{j \in H} w_{ij}h_j$. We can easily sample \mathbf{v} from a given \mathbf{h} using Eq. (8) and sample \mathbf{h} from a given \mathbf{v} using Eq. (9). Alternately repeating these two kinds of conditional samplings yields a (blocked) Gibbs sampling on the RBM. It is noteworthy that when $s \rightarrow \infty$, the conditional sampling of \mathbf{h} using Eq. (9) can be implemented using the inverse transform sampling. The cumulative distribution function of $P_{\infty}(h_j \mid \mathbf{v}, \theta)$ is

$$F(x) := \int_{-1}^x \frac{\exp(\lambda_j(\mathbf{v}, \theta)h_j)}{\phi_{\infty}(\lambda_j(\mathbf{v}, \theta))} dh_j = \frac{\exp(\lambda_j(\mathbf{v}, \theta)x) - \exp(-\lambda_j(\mathbf{v}, \theta))}{2 \sinh \lambda_j(\mathbf{v}, \theta)},$$

and therefore, its inverse function is

$$F^{-1}(u) := \frac{1}{\lambda_j(\mathbf{v}, \theta)} \ln \{ \exp(-\lambda_j(\mathbf{v}, \theta)) + 2u \sinh \lambda_j(\mathbf{v}, \theta) \}.$$

$F^{-1}(u)$ is the sampled value of h_j from $P_{\infty}(h_j \mid \mathbf{v}, \theta)$, where u is a sample point from the uniform distribution over $[0, 1]$.

2.1 Log-Likelihood Function and Its Gradients

Given N training data points for the visible layer, $D_V := \{\mathbf{v}^{(\mu)} \in \{-1, +1\}^{|V|} \mid \mu = 1, 2, \dots, N\}$, the learning of RBM is done by maximizing the log-likelihood function (or the negative cross-entropy loss function), defined by

$$l_s(\theta) := \frac{1}{N} \sum_{\mu=1}^N \ln P_s(\mathbf{v}^{(\mu)} \mid \theta), \quad (10)$$

with respect to θ (namely, the maximum likelihood estimation). The distribution in the logarithmic function in Eq. (10) is the marginal distribution obtained in Eq. (5). The log-likelihood function is regarded as the negative training error. Usually, the log-likelihood function is maximized using a gradient ascent method. The gradients of the log-likelihood function with respect to the learning parameters are as follows.

$$\frac{\partial l_s(\theta)}{\partial b_i} = \frac{1}{N} \sum_{\mu=1}^N v_i^{(\mu)} - \langle v_i \rangle_s, \tag{11}$$

$$\frac{\partial l_s(\theta)}{\partial c_j} = \frac{1}{N} \sum_{\mu=1}^N \psi_s(\lambda_j(\mathbf{v}^{(\mu)}, \theta)) - \langle h_j \rangle_s, \tag{12}$$

$$\frac{\partial l_s(\theta)}{\partial w_{ij}} = \frac{1}{N} \sum_{\mu=1}^N v_i^{(\mu)} \psi_s(\lambda_j(\mathbf{v}^{(\mu)}, \theta)) - \langle v_i h_j \rangle_s, \tag{13}$$

where $\langle \dots \rangle_s$ is the expectation of RBM, i.e.,

$$\langle A(\mathbf{v}, \mathbf{h}) \rangle_s := \sum_{\mathbf{v} \in \{-1, +1\}^{|V|}} \sum_{\mathbf{h} \in \mathcal{X}^{(s)}^{|H|}} A(\mathbf{v}, \mathbf{h}) P_s(\mathbf{v}, \mathbf{h} \mid \theta),$$

and

$$\psi_s(x) := \frac{\partial}{\partial x} \ln \phi_s(x) = \begin{cases} \frac{s+1}{s \tanh\{(s+1)x/s\}} - \frac{1}{s \tanh(x/s)} & 1 \leq s < \infty \\ \frac{1}{\tanh x} - \frac{1}{x} & s \rightarrow \infty \end{cases}. \tag{14}$$

The log-likelihood function can be maximized by a gradient ascent method with the gradients expressed in Eqs. (11)–(13). However, the evaluation of the expectations, $\langle \dots \rangle_s$, included in the above gradients is computationally hard. The computation time of the evaluation grows exponentially as the number of variables increases. Therefore, in practice, an approximate approach is used, for example, CD [5], pseudo-likelihood [10], composite likelihood [16], Kullback–Leibler importance estimation procedure (KLIEP) [17], and Thouless–Anderson–Palmer (TAP) approximation [3]. In particular, the CD method is the most popular method. In the CD method, the intractable expectations in Eqs. (11)–(13) are approximated by the sample averages of the sampled points in which each sampled point is generated from the (one-time) Gibbs sampling using Eqs. (8) and (9), starting from each data point $\mathbf{v}^{(\mu)}$.

2.2 Numerical Experiment Using Artificial Data

In the numerical experiments in this section, we used two RBMs: the generative RBM (gRBM), P_1^{gen} , and the learning RBM (tRBM), P_s^{train} . We obtained $N = 200$ artificial training data points, D_V , from the gRBM using Gibbs sampling, and subsequently, we trained the tRBM using the data points. The sizes of the visible layers of both RBMs were the same, namely, $|V| = 8$. The sizes of the hidden layers of the gRBM and tRBM were set to $|H| = 4$ and $|H| = 4 + R$, respectively. The sample space of the hidden variables in the gRBM was $\mathcal{X}(1) = \{-1, +1\}$, implying that the gRBM is the binary-RBM. The parameters of gRBM were randomly drawn: $b_i, c_j \sim G(0, 0.1^2)$ and

$$w_{i,j} \sim U[-\sqrt{6/(|V| + |H|)}, \sqrt{6/(|V| + |H|)}] \tag{15}$$

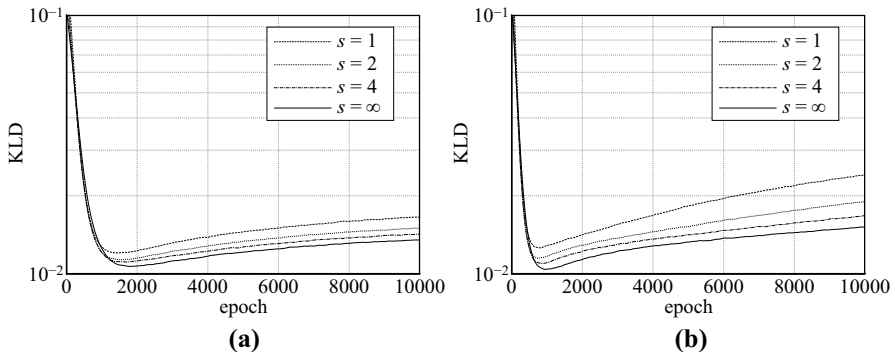


Fig. 2 KLDs against the number of parameter updates (epochs) when **a** $R = 0$ and **b** $R = 5$. We used the tRBM with $s = 1, 2, 4, \infty$. These plots show the average over 300 experiments

(Xavier’s initialization [4]), where $G(\mu, \sigma^2)$ is the Gaussian distribution and $U[\min, \max]$ is the uniform distribution.

We trained the tRBM using the CD method. In the training, the parameters of tRBM were initialized by $b_i = c_j = 0$ and Eq. (15). In the gradient ascent method, we used the full batch learning with the Adam method [7]. The quality of learning was measured using the Kullback–Leibler divergence (KLD) between the gRBM and tRBM:

$$\text{KLD} := \frac{1}{|V|} \sum_{\mathbf{v} \in \{-1,+1\}^{|V|}} P_1^{\text{gen}}(\mathbf{v}) \ln \frac{P_1^{\text{gen}}(\mathbf{v})}{P_s^{\text{train}}(\mathbf{v})}. \tag{16}$$

The KLD is regarded as the (pseudo) distance between the gRBM and tRBM. Thus, it is a type of generalization error. We can evaluate the KLD (the generalization error) and log-likelihood function in Eq. (10) (the negative training error) because the sizes of the RBMs are not large.

Figure 2a, b show the KLDs against the number of parameter updates (i.e., the number of gradient ascent updates). We observe that all KLDs increase as the learnings proceed owing to the effect of over-fitting. In Fig. 2a, because the gRBM and tRBM have the same structure (in other words, there is no model error), the effect of over-fitting is not severe. In contrast, in Fig. 2b, because the tRBM is more flexible than the gRBM, the effect of over-fitting tends to become severe. In fact, in Fig. 2b, the KLDs increase more rapidly as the learnings proceed. The increase in the KLD of higher s is evidently slower. Figure 3a, b show the log-likelihood functions divided by $|V|$ against the number of parameter updates. We observe that the log-likelihood function with lower s grows more rapidly. In other words, the training error in the tRBM with lower s decreases more rapidly. These results indicate that the multivalued hidden variables suppress over-fitting. In these experiments, the tRBM with $s = \infty$ is the best in terms of generalization.

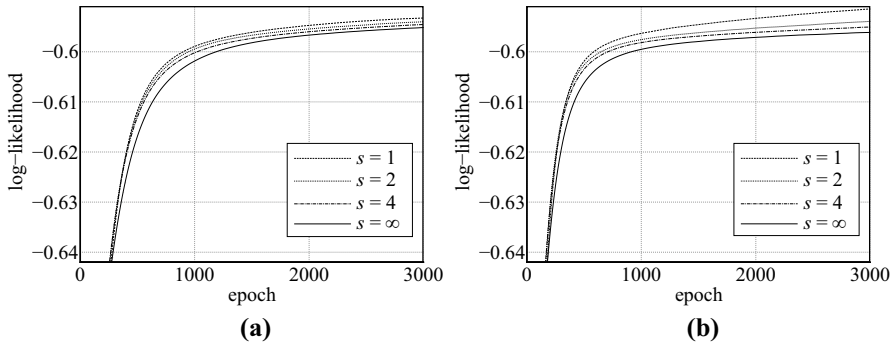


Fig. 3 Log-likelihoods (divided by $|V|$) against the number of parameter updates (epochs) when **a** $R = 0$ and **b** $R = 5$. We used the tRBM with $s = 1, 2, 4, \infty$. These plots show the average over 300 experiments

2.3 Effect of Multivalued Hidden Variables

In the numerical experiments described in the previous section, we demonstrated that the multivalued hidden variables suppress over-fitting. In this section, we provide an insight into the effect of multivalued hidden variables using a toy example. Although the consideration presented below is for a simple RBM, which is significantly different from practical RBMs, it is expected to provide an insight into the effect of multivalued hidden variables.

First, let us consider a simple RBM with two visible variables:

$$P_s(v_1, v_2, \mathbf{h} \mid w) = \frac{\omega(s)}{Z_s(w)} \exp \left(w \sum_{i=1}^2 \sum_{j \in H} v_i h_j \right). \tag{17}$$

The marginal distribution of Eq. (17) is

$$P_s(v_1, v_2 \mid w) = \frac{\exp \{ |H| \ln \phi_s(w(v_1 + v_2)) \}}{2 \exp \{ |H| \ln \phi_s(2w) \} + 2 \exp (|H| \ln 2)}, \tag{18}$$

where we used $\lim_{x \rightarrow 0} \phi_s(x) = 2$ and $\phi_s(x) = \phi_s(-x)$. Because $v_1, v_2 \in \{-1, +1\}$, Eq. (18) can be expanded as

$$P_s(v_1, v_2 \mid w) = \frac{1 + m_{s,1}(w)v_1 + m_{s,2}(w)v_2 + \alpha_s(w)v_1v_2}{4}, \tag{19}$$

where

$$\begin{aligned} m_{s,i}(w) &:= \sum_{v_1, v_2 \in \{-1, +1\}} v_i P_s(v_1, v_2 \mid w) = 0, \\ \alpha_s(w) &:= \sum_{v_1, v_2 \in \{-1, +1\}} v_1 v_2 P_s(v_1, v_2 \mid w) \\ &= \frac{\exp \{ |H| \ln \phi_s(2w) \} - \exp (|H| \ln 2)}{\exp \{ |H| \ln \phi_s(2w) \} + \exp (|H| \ln 2)} \geq 0. \end{aligned}$$

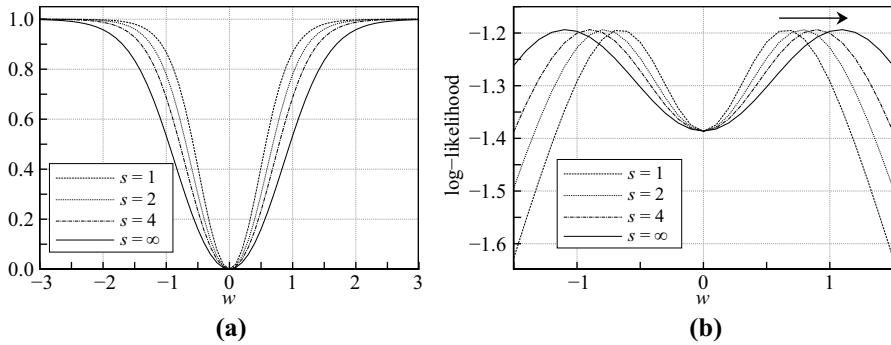


Fig. 4 **a** Plot of $\alpha_s(w)$ versus w for various s when $|H| = 2$. For $\beta = 0.6$, the values of $|w_s^*|$ for $s = 1, 2, 4$, and ∞ are approximately 0.6585, 0.7834, 0.8941, and 1.0887, respectively. **b** Plot of the log-likelihood function versus w for various s when $|H| = 2$ and $\beta = 0.6$. The shape of the peak around the global maximum point becomes sharper as s decreases

Next, we consider the empirical distribution of D_V :

$$Q_{D_V}(v_1, v_2) := \frac{1}{N} \sum_{\mu=1}^N \prod_{i=1}^2 \delta(v_i, v_i^{(\mu)}),$$

where $\delta(x, y)$ is the Kronecker delta function. Similar to Eq. (19), the empirical distribution is expanded as

$$Q_{D_V}(v_1, v_2) = \frac{1 + d_1 v_1 + d_2 v_2 + \beta v_1 v_2}{4}, \tag{20}$$

where $d_i := \sum_{\mu=1}^N v_i^{(\mu)} / N$ and $\beta := \sum_{\mu=1}^N v_1^{(\mu)} v_2^{(\mu)} / N$. For simplicity, in the following discussion, we assume that $d_1 = d_2 = 0$ and $\beta \geq 0$. Under this assumption, using the expanded forms in Eqs. (19) and (20), the log-likelihood function of the simple RBM is expressed by

$$\begin{aligned} l_s(w) &= \sum_{v_1, v_2 \in \{-1, +1\}} Q_{D_V}(v_1, v_2) \ln P_s(v_1, v_2 | w) \\ &= \sum_{v_1, v_2 \in \{-1, +1\}} \frac{1 + \beta v_1 v_2}{4} \ln \frac{1 + \alpha_s(w) v_1 v_2}{4}. \end{aligned} \tag{21}$$

Ultimately, the aim of the maximum likelihood estimation is to find the value of w that realizes $P_s(v_1, v_2 | w) = Q_{D_V}(v_1, v_2)$ or in other words, to find a value of w_s^* that satisfies $\alpha_s(w_s^*) = \beta$. The log-likelihood function in Eq. (21) is globally maximized at $w = w_s^*$ and the RBM with w_s^* over-fits the data distribution. It can be shown that the function $\alpha_s(w)$ has the following three properties: (i) it is symmetric with respect to w , (ii) it monotonically increases with an increase in $w \geq 0$, and (iii) it monotonically decreases with an increase in s when $|x| \neq 0$. The function $\alpha_s(w)$ with $|H| = 2$ is shown in Fig. 4 (a) as an example. These three properties lead to the inequality

$|w_s^*| < |w_{s+1}^*|$ for a certain $\beta > 0$, which implies that the global maximum point of the log-likelihood function in Eq. (21) moves away from the origin, $w = 0$, as s increases (see Fig. 4b).

Usually, the initial value of w is set to a value around the origin. As shown in Fig. 4b, the global maximum point moves closer to the origin and the peak becomes sharper (in other words, the global maximum point becomes the stronger attractor) as s decreases. This implies that with a gradient ascent type of algorithm, the RBM with a lower s can reach the global maximum point more rapidly and causes over-fitting during an early stage of the learning. Whereas the convergence with the global maximum point of the RBM with a higher s is slower and it prevents over-fitting during an early stage of the learning¹. In fact, the increases in the generalization error (the KLD) and negative training error (the log-likelihood function) become faster as s decreases in the numerical results obtained in the previous section (cf. Figs. 2, 3).

From the above analysis, we found that the global maximum point moves away from the origin and becomes a weaker attractor as s increases. This could lead to some expectations, for example: (i) in a more practical RBM, its log-likelihood function usually has several local maximum points, and thus, the RBM with a higher s is more easily trapped by one of the local maximum points before converging with the global maximum point (namely, the over-fitting point) and (ii) some regularization methods, such as early stopping or L_2 regularization, are more effective in the RBM with a higher s .

3 Application to Classification Problem

Let us consider a classification (or pattern recognition) problem in which an n -dimensional input vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ is classified into K different classes, C_1, C_2, \dots, C_K . It is convenient to use a 1-of- K representation (or a 1-of- K coding) to identify each class [1]. In the 1-of- K representation, each class corresponds to the K -dimensional vector $\mathbf{t} = (t_1, t_2, \dots, t_K)^T$, where $t_k \in \{0, 1\}$ and $\sum_{k=1}^K t_k = 1$, i.e., \mathbf{t} is a vector in which the value of only one element is one and the remaining elements are zero. When $t_k = 1$, \mathbf{t} indicates class C_k . For simplicity of the notation, we denote the 1-of- K vector, whose k th element is one, by $\mathbf{1}_k$. In the following section, we consider the application of the proposed RBM to the classification problem.

3.1 Discriminative Restricted Boltzmann Machine

A discriminative restricted Boltzmann machine (DRBM) was proposed to solve the classification problem [8, 9], which is a conditional distribution of the output 1-of- K vector \mathbf{t} conditioned with a continuous input vector \mathbf{x} . The conventional DRBM can be

¹ Because the value of the log-likelihood function at the global maximum point for a higher s is the same as that for a lower s , the RBM with a higher s also causes over-fitting at that point.

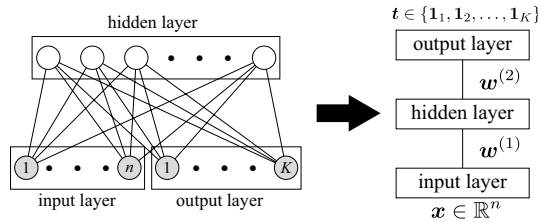


Fig. 5 Discriminative restricted Boltzmann machine is obtained to an extension of the RBM. Because the output layer corresponds to the 1-of- K vector, it takes only K different states. For distinction, the couplings between the input and hidden layers are represented by $w^{(1)}$ and those between the hidden and output layers are represented by $w^{(2)}$

obtained by a simple extension to the binary-RBM. The DRBM is obtained by the following process. The visible variables in the RBM are divided into two layers, the input and output layers. The K visible variables assigned to the output layer, t , are redefined as the 1-of- K vector with $\mathbf{1}_k$ as its realization (i.e., $t \in \{\mathbf{1}_k \mid k = 1, 2, \dots, K\}$), and the n visible variables assigned to the input layer, x , are redefined as the continuous input vector (see Fig. 5). Subsequently, we make a conditional distribution conditioned with the variables in the input layer: $P(t, h \mid x)$. Finally, by marginalizing the hidden variables out, we obtain the DRBM: $P(t \mid x) = \sum_h P(t, h \mid x)$.

By using the proposed RBM instead of the binary-RBM, we obtain an extension to the conventional DRBM, i.e., we obtain a DRBM with multivalued hidden variables. The proposed DRBM for $s \in \mathbb{N}$ is obtained by

$$P_s(t \mid x, \theta) := \frac{1}{Z_s(x, \theta)} \exp \left(\sum_{k=1}^K b_k t_k + \sum_{j \in H} \ln \phi_s(\zeta_j(t, x, \theta)) \right). \tag{22}$$

where $\zeta_j(t, x, \theta) := c_j + \sum_{k=1}^K w_{j,k}^{(2)} t_k + \sum_{i=1}^n w_{i,j}^{(1)} x_i$ and

$$Z_s(x, \theta) := \sum_{k=1}^K \exp \left(b_k + \sum_{j \in H} \ln \phi_s(\zeta_j(\mathbf{1}_k, x, \theta)) \right). \tag{23}$$

The function $\phi_s(x)$ appearing in Eqs. (22) and (23) is already defined in Eq. (6). It is noteworthy that when $s = 1$, Eq. (22) is equivalent to the conventional DRBM proposed in Ref. [8]. Eq. (22) is regarded as the class probability, indicating that $P_s(t = \mathbf{1}_k \mid x, \theta)$ is the probability of the input x belonging to class C_k . The input x should be assigned into a class that gives the maximum class probability.

Given N supervised training data points, $D := \{(t^{(\mu)}, \mathbf{x}^{(\mu)}) \mid \mu = 1, 2, \dots, N\}$, the log-likelihood function of the proposed DRBM in Eq. (22) is defined as

$$l_s^{\dagger}(\theta) := \frac{1}{N} \sum_{\mu=1}^N \ln P_s(t^{(\mu)} \mid \mathbf{x}^{(\mu)}, \theta). \tag{24}$$

The gradients of the log-likelihood function with respect to the parameters are obtained as follows.

$$\frac{\partial l_s^\dagger(\theta)}{\partial b_k} = \frac{1}{N} \sum_{\mu=1}^N [t_k^{(\mu)} - P_s(\mathbf{1}_k \mid \mathbf{x}^{(\mu)}, \theta)], \tag{25}$$

$$\frac{\partial l_s^\dagger(\theta)}{\partial c_j} = \frac{1}{N} \sum_{\mu=1}^N [\psi_s(\zeta_j(\mathbf{t}^{(\mu)}, \mathbf{x}^{(\mu)}, \theta)) - \langle \psi_s(\zeta_j(\mathbf{t}, \mathbf{x}^{(\mu)}, \theta)) \rangle_t^{(\mu,s)}], \tag{26}$$

$$\frac{\partial l_s^\dagger(\theta)}{\partial w_{ij}^{(1)}} = \frac{1}{N} \sum_{\mu=1}^N x_i^{(\mu)} [\psi_s(\zeta_j(\mathbf{t}^{(\mu)}, \mathbf{x}^{(\mu)}, \theta)) - \langle \psi_s(\zeta_j(\mathbf{t}, \mathbf{x}^{(\mu)}, \theta)) \rangle_t^{(\mu,s)}], \tag{27}$$

$$\frac{\partial l_s^\dagger(\theta)}{\partial w_{j,k}^{(2)}} = \frac{1}{N} \sum_{\mu=1}^N \psi_s(\zeta_j(\mathbf{1}_k, \mathbf{x}^{(\mu)}, \theta)) [t_k^{(\mu)} - P_s(\mathbf{1}_k \mid \mathbf{x}^{(\mu)}, \theta)], \tag{28}$$

where $\langle \dots \rangle_t^{(\mu,s)}$ denotes the expectation defined by

$$\langle A(\mathbf{t}) \rangle_t^{(\mu,s)} := \sum_{k=1}^K A(\mathbf{1}_k) P_s(\mathbf{1}_k \mid \mathbf{x}^{(\mu)}, \theta).$$

The function $\psi_s(x)$ appearing in the above gradients is already defined in Eq. (14). It is noteworthy that the gradients expressed in Eqs. (25)–(28) are computed without an approximation, unlike those in the RBM, owing to the special structure of DRBM. In the training, we maximize $l_s^\dagger(\theta)$ with respect to θ using a gradient ascent method with Eqs. (25)–(28).

3.2 Numerical Experiment Using MNIST Data Set

In this section, we show the results of the numerical experiment using MNIST. MNIST is a data set of 10 different handwritten digits, 0, 1, ..., and 9 and is composed of 60, 000 training data points and 10, 000 test data points. Each data point includes the input data, a 28×28 digit (8-bit) image, and the corresponding target digit label. Therefore, for the data set, we set $n = 784$ and $K = 10$. All input images were normalized by dividing by 255 during preprocessing.

We trained the proposed DRBM with $|H| = 200$ using $N = 1000$ training data points in MNIST and tested it using 10000 test data points. In the training, we used the stochastic gradient ascent (SGA), for which the mini-batch size was $B = 100$, with the AdaMax optimizer [7]. All coupling parameters were initialized by the Xavier method [4], and all bias parameters were initialized by zero. Figure 6 shows the plots of the missclassification rates for (a) training data set and (b) test data set versus the number of parameter updates. All input images in the test data set

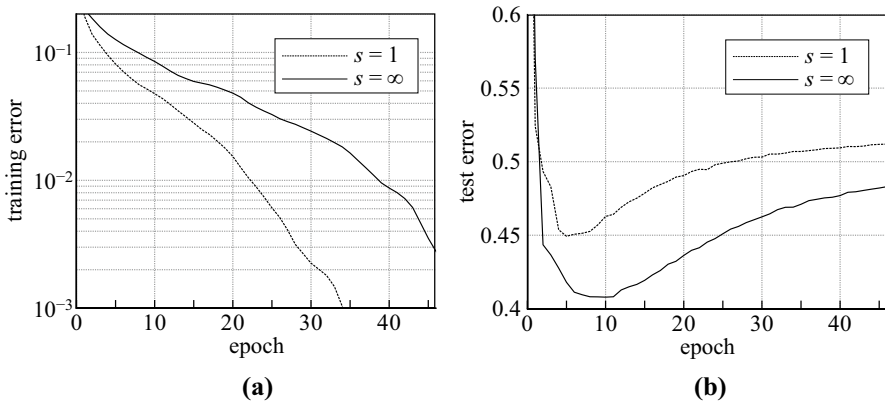


Fig. 6 Missclassification errors against the number of parameter updates (epochs): **a** training error and **b** test error. Here, one epoch consists of one full update cycle over the training data set, implying that one epoch involves $N/B = 10$ updates by the SGA in this case. We used the DRBM with $s = 1, \infty$. These plots show the average over 120 experiments

were corrupted by the Gaussian noise with $\sigma = 120$ before the normalization². We observe that the DRBM with $s = \infty$ is better in terms of generalization because it shows a higher training error and lower test error. This indicates that the multivalued hidden variables are also effective in the DRBM.

4 Conclusion

In this paper, we proposed an RBM with multivalued hidden variables, which is a simple extension to the conventional binary-RBM and showed that the proposed RBM is better than the binary-RBM in terms of the generalization property via numerical experiments conducted on CD learning with artificial data (in Sect. 2.2) and classification problem with MNIST (in Sect. 3.2).

It is important to understand the reason why the multivalued hidden variables are effective in terms of over-fitting. We provided a basic insight into it by analyzing a simple example in Sect. 2.3. However, practical RBMs are much more complex than the simple example used in this study. Therefore, we need to perform further analysis to clarify this reason. We think that a mean-field analysis [11] can be used to perform the further analysis. Moreover, a criteria for over-fitting was provided in Ref. [2]. The relationship between the criteria and our multivalued hidden variables is also interesting. These issues will be addressed in our future studies.

² Each corrupted input image $\hat{\mathbf{x}}$ was created from the corresponding original image \mathbf{x} by $\hat{x}_i = x_i + \epsilon_i$, where ϵ_i is the additive white Gaussian noise drawn from $G(0, 120^2)$. If $\hat{x}_i > 255$, we set $\hat{x}_i = 255$ and if $\hat{x}_i < 0$, we set $\hat{x}_i = 0$.

Acknowledgements This work was partially supported by JSPS KAKENHI (Grant Numbers 15K00330, 15H03699, 18K11459, and 18H03303), JST CREST (Grant number JPMJCR1402), and the COI Program from the JST (Grant number JPMJCE1312).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.
2. Coolen, A. C. C., Barrett, J. E., Paga, P., & Perez-Vicente, C. J. (2017). Replica analysis of overfitting in regression models for time-to-event data. *Journal of Physics A: Mathematical and Theoretical*, *50*(37), 375001.
3. Gabri , M., Tramel, E. W., & Krzakala, F. (2015). Training restricted boltzmann machines via the thoulless-anderson-palmer free energy. In *Proceedings of the 28th International Conference on Neural Information Processing Systems* (pp. 640–648).
4. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (Vol. 9, pp. 249–256).
5. Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*(8), 1771–1800.
6. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief net. *Neural Computation*, *18*(7), 1527–1554.
7. Kingma, D.P., & Ba, L.J. (2015). Adam: A method for stochastic optimization. In *Proceedings in the 3rd International Conference on Learning Representations* (pp. 1–13).
8. Larochelle, H., & Bengio, Y. (2008). Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 536–543).
9. Larochelle, H., Mandel, M., Pascanu, R., & Bengio, Y. (2012). Learning algorithms for the classification restricted boltzmann machine. *The Journal of Machine Learning Research*, *13*(1), 643–669.
10. Marlin, B., Swersky, K., Chen, B., & Freitas, N. (2010). Inductive principles for restricted boltzmann machine learning. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, *9*, 509–516.
11. Nishimori, H. (2001). *Statistical physics of spin glass and information processing: An introduction*. Oxford: Oxford University Press.
12. Salakhutdinov, R., & Hinton, G. E. (2009). Deep boltzmann machines. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS 2009)* (pp. 448–455).
13. Salakhutdinov, R., & Hinton, G. E. (2012). An efficient learning procedure for deep boltzmann machines. *Neural Computation*, *24*(8), 1967–2006.
14. Salakhutdinov, R., Mnih, A., & Hinton, G. E. (2007). Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning* (pp. 791–798).
15. Smolensky, P. (1986) Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland (eds.) *Parallel distributed processing: Explorations in the microstructure of cognition* vol. 1 (pp. 194–281). Cambridge, MA, USA: MIT Press.
16. Yasuda, M., Kataoka, S., Waizumi, Y., & Tanaka, K. (2012). Composite likelihood estimation for restricted boltzmann machines. In *Proceedings of 21st International Conference on Pattern Recognition* (pp. 2234–2237).
17. Yasuda, M., Sakurai, T., & Tanaka, K. (2011). Learning algorithm in restricted boltzmann machines using Kullback–Leibler importance estimation procedure. *Nonlinear Theory and Its Applications, IEICE*, *2*(2), 153–164.
18. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, *67*, 301–320.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.