

Commun. Theor. Phys. **74** (2022) 095101 (6pp) https://doi.org/10.1088/1572-9494/ac7040

Realizing number recognition with simulated quantum semi-restricted Boltzmann machine

Fuwen Zhang[1,2], **Yonggang Tan**[3] **and Qing-yu Cai**[4,5]

[1] School of Physics, Zhengzhou University, Zhengzhou 450001, China
[2] Innovation Academy for Precision Measurement Science and Technology, Chinese Academy of Sciences, Wuhan 430071, China
[3] School of Physics and Electronic Information, Luoyang Normal University, Luoyang 471934, China
[4] Center for Theoretical physics, Hainan University, Haikou 570228, China
[5] School of Information and Communication Engineering, Hainan University, Haikou 570228, China

E-mail: qycai@hainanu.edu.cn

Received 27 April 2022, revised 16 May 2022
Accepted for publication 17 May 2022
Published 15 August 2022

Abstract

Quantum machine learning based on quantum algorithms may achieve an exponential speedup over classical algorithms in dealing with some problems such as clustering. In this paper, we use the method of training the lower bound of the average log likelihood function on the quantum Boltzmann machine (QBM) to recognize the handwritten number datasets and compare the training results with classical models. We find that, when the QBM is semi-restricted, the training results get better with fewer computing resources. This shows that it is necessary to design a targeted algorithm to speed up computation and save resources.

Keywords: machine learning, quantum Boltzmann machine, quantum algorithm

(Some figures may appear in colour only in the online journal)

1. Introduction

The Boltzmann machine (BM) [1] is an undirected model consisting of visible layers and hidden layers. The information is input from the visible layer and obeys the Boltzmann distribution in the hidden layer. It has been widely applied in many fields, such as phone recognition tasks [2], image recognition [3], medical health [4], the quantum many-body problem [5], and so on. In BM training, it is difficult to compute the negative phase value of the partial derivative of the average likelihood function, since its computational complexity will increase exponentially with the dimensions and the quantity of the training data. This difficulty can be solved by using Gibbs sampling to gain the expectations of the computational model.

The mixing of the Gibbs sampler becomes slow when the sampling data are complicated. Hinton proposed the contrast divergence (CD) method [6] to solve the slow sampling problem. The method assumes that there is a fantasy particle, and N steps of Gibbs sampling at the fantasy particle are run to replace the model distribution. The CD method performs well in actual training [7]. Alternatively, there is another technique to solve the model expectation named the persistent comparison divergence (PCD). The procedure is that when the state s^t of the fantasy particle and the corresponding parameter θ^t are known at time t, one can get $s^{t+\delta t}$ at the time $t+\delta t$ by transferring the operator. Then the parameters of the model are updated to the parameter $\theta^{t+\delta t}$ at the time $t+\delta t$. Different from the CD method, it requires reducing the learning rate with the update to ensure the convergence of the model [8]. Deep belief networks [9] and deep BMs [10] can be obtained by using the stack of BM, the training of which could be done with the greedy layerwise strategy. The greedy layerwise strategy contains the process of pre-training the model, which is more efficient than the random selection of parameters. The deep model [11] can learn the deep patterns and the abstract concepts of data, which has great advantages over the shallow learning in learning and interpreting complicated data [12].

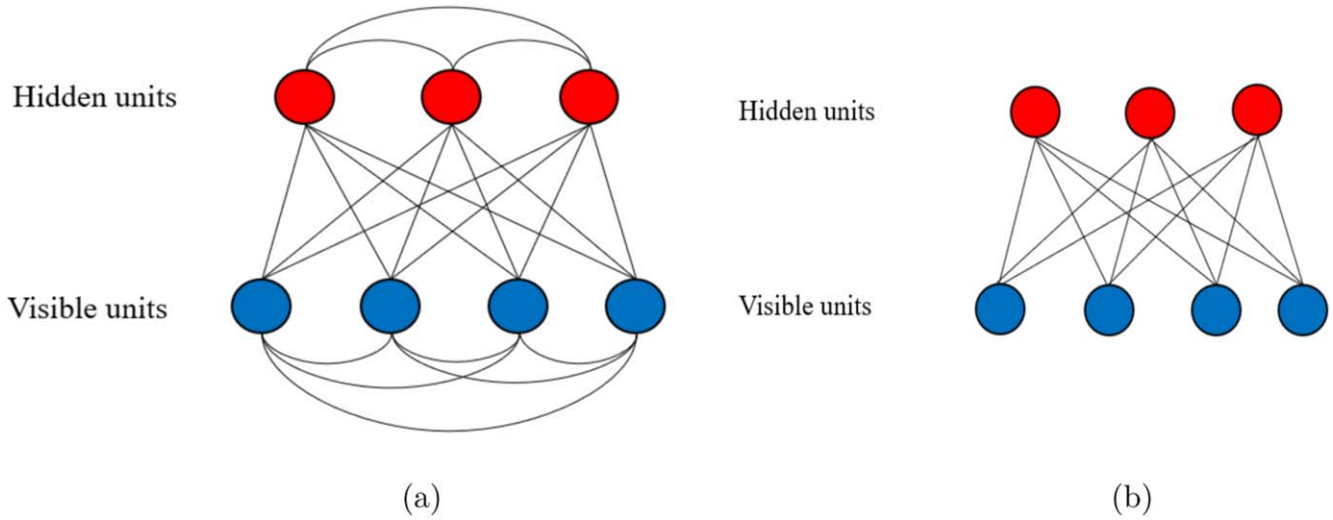


Figure 1. (a) The model of fully connected BM. The blue circles represent visible units, and the red circles represent hidden units. (b) The model of RBM without lateral connectivity both in the visible units and in the hidden units.

In practice, the dimensions and quantity of some tasks, such as computer version [13], and speech recognition [14], are usually huge, training of which requires huge computing resources. If these tasks are performed in the classical way, there are disadvantages such as slow training speed and easily falling into local minima [15]. In order to overcome these shortcomings, scientists proposed to optimize classical machine learning algorithms by exploiting the potential of quantum computing [16]. The combination of quantum theory with BM is generally called quantum Boltzmann machine (QBM) which can be mainly divided into two directions, one is based on the quantum variational principle method [17], and the other is based on the quantum annealing method [18].

In the quantum variational principle method, the Gibbs state for a given Hamiltonian is approximately generated by combining the quantum approximate optimization algorithm or the variational quantum imaginary time evolution algorithm, and then the parameters can be adjusted [19, 20]. The quantum annealing algorithm is based on the principle of the quantum tunneling effect [21]. Under the annealing condition, the quantum configuration energy will eventually evolve into the ground state and the training parameters are in the global optimal solution. Usually, annealing can be realized by using quantum annealing machines to construct a QBM model. Quantum annealing methods outperform classical methods in a number of iterations when recognizing numerical tasks [22].

The purpose of this paper is to demonstrate the possibility of digit recognition with various QBMs, particularly the quantum semi-restricted Boltzmann machines (QSRBM). We train the dataset on a QSRBM based on the method of training the lower bound of the quantum log likelihood function and give the training fidelity. This paper is organized as follows. In the second section, the principle of BM and QBM is briefly introduced. Then we show the processing of training data and the results of training. Finally, we discuss and conclude.

2. Quantum BM

BM in figure 1 (a) can be changed into various machines by adjusting the connection of layers. One can get a restricted Boltzmann machine (RBM) by disconnecting the lateral connectivity in the BM. QBM can be constructed by replacing the data units with corresponding operators.

2.1. Restricted BM

BM is a fully connected model including the lateral connection in the layers, which may cause additional computational complexity when dealing with some problems. As an improvement in some cases, there is no such connection in the layers of RBM. Without affecting the training performance, the RBM model is simpler and more efficient [23] and has been applied to some practical problems [24, 25]. The energy function of the joint configuration of RBM is given by

$$E(v, h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{ij} w_{ij} v_i h_j, \tag{1}$$

where v_i is the binary state of the visible unit i, h_j is the binary state of the hidden unit j, a_i, b_j, w_{ij} are the connection strengths between the unit layers, and $\theta \in a_i, b_j, w_{ij}$ [26]. The probability distribution of the visible unit is [27]

$$p(v) = \frac{1}{Z}\sum_h \exp(-E(v, h)), \tag{2}$$

where Z is the partition function. The machine with connections in figure 1 (b) is called RBM [28], and the hidden units of RBM are independent of given visible units. The expected value of the data can be obtained in only one parallel step, thus greatly reducing the amount of computation [27].

Usually, the training process can be performed by minimizing the negative average log likelihood function ζ

$$\zeta = -\sum_v p_v^{\text{data}} \log p_v. \tag{3}$$

The parameters can be trained by taking partial derivatives of the likelihood function

$$\frac{\partial\zeta}{\partial w_{ij}} = \langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{model}},$$
$$\frac{\partial\zeta}{\partial a_i} = \langle v_i\rangle_{\text{data}} - \langle v_i\rangle_{\text{model}},$$
$$\frac{\partial\zeta}{\partial b_j} = \langle h_j\rangle_{\text{data}} - \langle h_j\rangle_{\text{model}}, \tag{4}$$

with the gradient descent algorithm [29]

$$\Delta w_{ij} = \eta(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{model}}),$$
$$\Delta a_i = \eta(\langle v_i\rangle_{\text{data}} - \langle v_i\rangle_{\text{model}}),$$
$$\Delta b_j = \eta(\langle h_j\rangle_{\text{data}} - \langle h_j\rangle_{\text{model}}), \tag{5}$$

where η is the learning rate. η needs to be selected according to the actual problem: If η is too large, it is easy to miss the best solution, resulting in divergence; If η is too small, the number of the iteration steps will be too large. In equation (5), $\langle v\rangle_{\text{data}}$ are the clamped expectation with v fixed, which can be easily calculated by Gibbs sampling. $\langle v\rangle_{\text{model}}$ are the expected value of v over the model probability distribution, the calculation of $\langle v\rangle_{\text{model}}$ requires solving the value of the partition function Z and the parameters of the BM, which is almost incalculable and uneconomical for complicated data. The partial derivative of the negative log likelihood function can often be approximated by CD method, which uses the sampling formula [30]

$$p(h_j = 1|v) = \sigma(b_j + \sum_i w_{ij} h_j),$$
$$p(v_i = 1|v) = \sigma(v_i + \sum_j w_{ij} v_i) \tag{6}$$

to sample the data. By sampling the data n times, the update rules in equation (5) are changed into

$$\Delta w_{ij} = \eta(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_n). \tag{7}$$

The update rules which respect to the weights a_i, b_j are similar to equation (7). Even taking $n = 1$, the CD method usually works well [7].

2.2. QBM and quantum semi-restricted BM

QBM can be obtained by a quantum approximate optimization algorithm method [31]. Alternatively, it can be constructed QBM by mapping the units of BM to the qubits of the D-Wave system [32] that exploits the advantage of quantum tunneling. One can use variational quantum imaginary time evolution to obtain the Gibbs state of the QBM model and realize training [19]. One can also use the model whose visible layers are still classical and only change the hidden layers from classical variables to a quantum degree of freedom, which can be trained with the PCD method [33].

For simplicity, the Hamiltonian H of a QBM model is given by [34]

$$H = -\sum_i a_i\sigma_i^z - \sum_j b_j\sigma_j^z - \sum_{ij} w_{ij}\sigma_i^z\sigma_j^z, \tag{8}$$

where

$$\sigma_i^z \equiv I_1 \otimes \cdots \otimes I_{i-1} \otimes \sigma_z \otimes I_{i+1} \otimes \cdots \otimes I_n, \tag{9}$$

and I and σ_z are the identity operator and Pauli operator respectively. The classical $p(v)$ in equation (2) can be obtained by tracing the hidden unit of the density matrix ρ on the QBM

$$p_v = \text{Tr}[\Lambda_v\rho], \tag{10}$$

where $\Lambda_v = |v\rangle\langle v| \otimes I_h$. Then the equation (4) is replaced with

$$\frac{\partial\zeta}{\partial\theta} = \sum_i p_v^{\text{data}}\left(\frac{\text{Tr}[\Lambda_v\partial_\theta e^{-H}]}{\text{Tr}[\Lambda_v e^{-H}]} - \frac{\text{Tr}[\partial_\theta e^{-H}]}{\text{Tr}[e^{-H}]}\right). \tag{11}$$

The first term on the right side in equation (11) cannot be directly processed for derivation. This problem can be solved by finding the upper bound with Golden–Thompson inequality [35]

$$p_v = \frac{\text{Tr}[e^{-H}\cdot e^{\ln\Lambda_v}]}{\text{Tr}[e^{-H}]} \geqslant \frac{\text{Tr}[e^{-H+\ln\Lambda_v}]}{\text{Tr}[e^{-H}]}. \tag{12}$$

Setting

$$H_v = H - \ln\Lambda_v, \tag{13}$$

we have that

$$p_v \geqslant \frac{\text{Tr}[e^{-H_v}]}{\text{Tr}[e^{-H}]}, \tag{14}$$

and then

$$\zeta \leqslant \zeta_{\text{up}} \equiv -\log p_v^{\text{data}}\cdot\log\frac{\text{Tr}[e^{-H_v}]}{\text{Tr}[e^{-H}]}. \tag{15}$$

Minimizing the upper bound ζ_{up} of ζ, one can obtain

$$\frac{\partial\zeta_{\text{up}}}{\partial w_{ij}} = \langle v_i h_j\rangle_v - \langle v_i h_j\rangle,$$
$$\frac{\partial\zeta_{\text{up}}}{\partial a_i} = \langle v_i\rangle_v - \langle v_i\rangle,$$
$$\frac{\partial\zeta_{\text{up}}}{\partial b_j} = \langle h_j\rangle_v - \langle h_j\rangle. \tag{16}$$

Since the fully connected QBM is too complicated, for simplicity, we consider disconnecting the connection in the hidden layers, the quantum version of which is usually called QSRBM. The Hamiltonian in equation (13) can hence be rewritten as

$$H_v = -\sum_i(c_i\sigma_i^x + f_i^e\sigma_i^z), \tag{17}$$

where c_i is the connection strength within visible layers, $f_i^e = a_i + \sum_\mu w_{i\mu}v_\mu$, and

$$\sigma_i^x \equiv I_1 \otimes \cdots \otimes I_{i-1} \otimes \sigma_x \otimes I_{i+1} \otimes \cdots \otimes I_n. \tag{18}$$

Combining equation (11) and equation (17), one can get

$$\langle\sigma_i^z\rangle_v = \frac{f_i^e}{G_i}\tanh G_i, \tag{19}$$

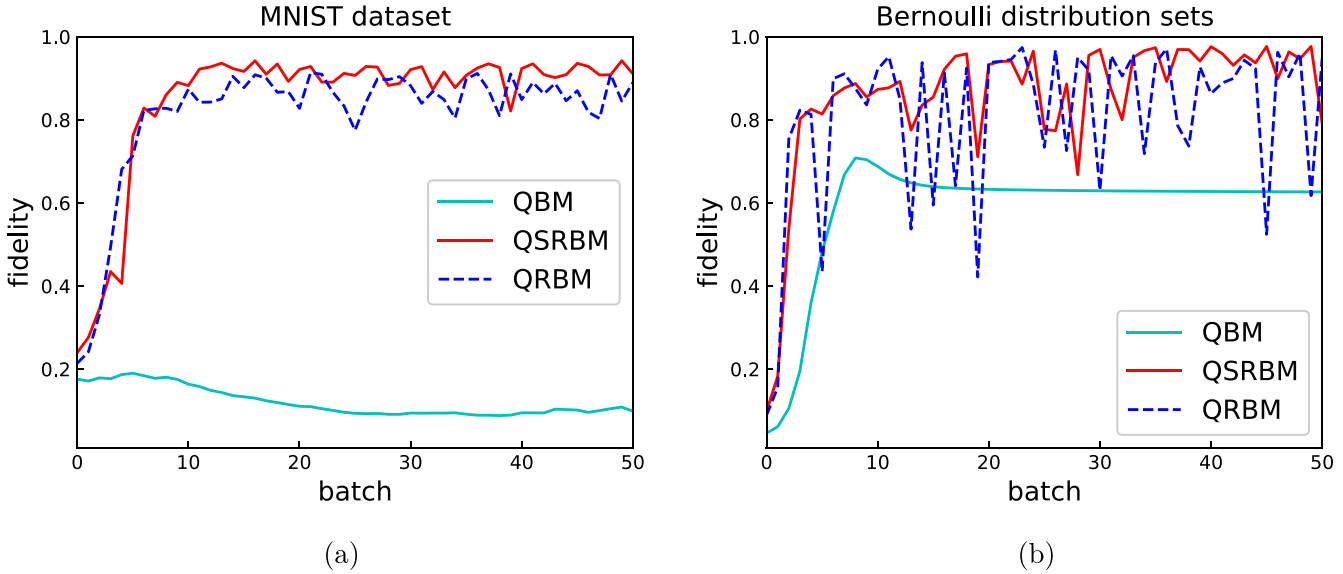


Figure 2. (a) The training of MNIST datasets in QBM, QRBM and QSRBM. The accuracy rate of MNIST datasets reaches to 0.109, 0.845, 0.942 respectively, when the number of batches is 49. (b) The training of Bernoulli distribution sets in QBM, QRBM and QSRBM. The accuracy rate of Bernoulli distribution sets reaches to 0.627, 0.618, 0.977 respectively, when the number of batch is 49.

where $G_i = \sqrt{c_i^2 + (f_i^e)^2}$. In the following, we focus on training the Bernoulli distribution sets and the MNIST dataset [36] and give the fidelities on the QBM, QRBM and QSRBM.

3. Training results

We firstly process the data to fit our models and then give the training results. Our results show that the recognition rate of the quantum algorithm can be close to the classical results even with small resources.

3.1. Data processing

For Bernoulli distribution, we use

$$f(x|p) = p^x q^{n-x} \tag{20}$$

to generate a set of binary data. Here, p, q are the probability of generating 1,0 each time, $p = 1 - q$, n is the trial count, and x is the sum of the generating number. $f(x|p)$ is the probability of obtaining x after the n trials. We set $p = 0.9$ here (We also trained the machines with other values of p).

The MNIST dataset was first introduced by LeCun *et al* [37]. It is an optical pixel set of 10 Arabic numerals, containing a total of 70 000 images with a resolution 28×28. Since the outer pixel value of the most of dataset is 0, we firstly eliminated the outer 2 pixels, and hence the pixels are reduced to 24×24 size. Next, every 8×8 pixel value is averaged to reduce the digital image to 3×3 pixels. Although 3×3 pixels could not be used to distinguish digital with our naked eyes, they still contain most of the information of digital images and are representative. The times of iterations on training Bernoulli distribution sets and MNIST dataset with RBM are 1000 and 30 000, respectively. The Bernoulli distribution sets and MNIST dataset are trained by using fully connected QBM, RBM, QRBM and QSRBM, respectively. The number of visible and hidden units here is 9 and 2, respectively.

An important improvement on the original model [34] is converting the measurement of work performance from Kullback–Leibler divergence [38] value to the fidelity between p_v^{data} and p_v. It will make our training results have an intuitive comparison with other types of QBM, the classical BM, and other models of the same data type training. The fidelity is given by

$$F = \sqrt{(p_v^{\text{data}})^{\frac{1}{2}} p_v (p_v^{\text{data}})^{\frac{1}{2}}}, \tag{21}$$

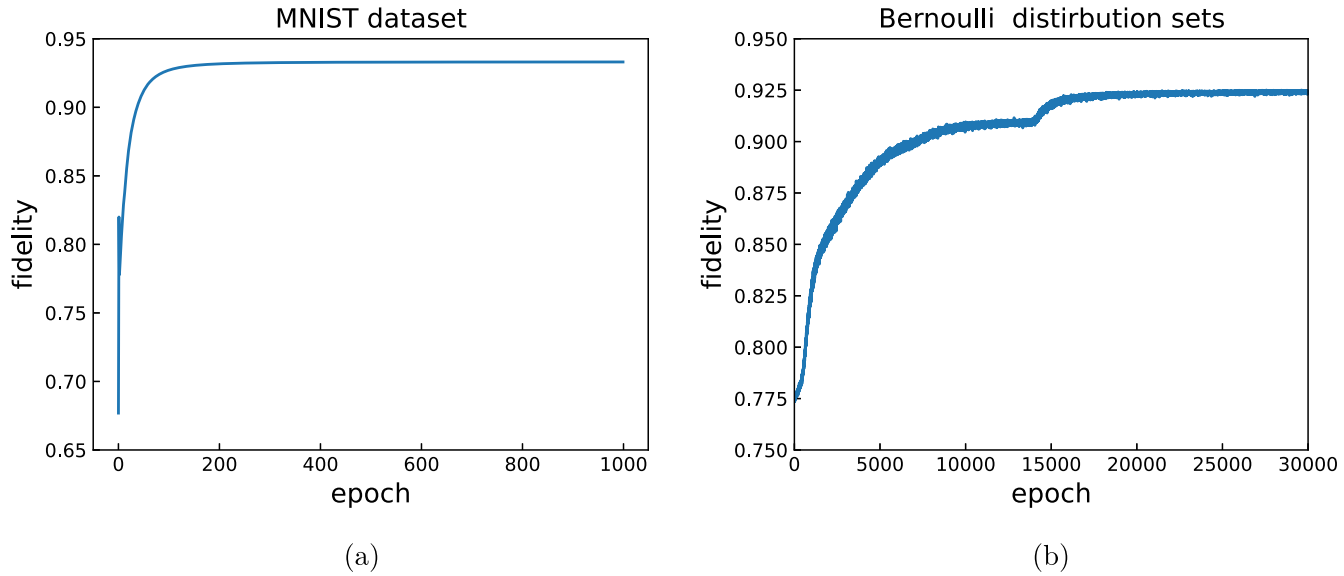
where p_v^{data} is the probability distribution of training data.

3.2. Simulation conditions

The CPU of our computer is Intel 8 cores and its model is Intel(R) Xeon(R) CPU E3-1245 v5, the main frequency is 3.50 GHZ. The operating system of our computer is Win10 64-bits. We use torch [39] developed by Facebook as the framework to build the model, and run the program using the method of synchronous execution in parallel processing. The program simulation of the above model is based on python programming language. We set $c_i = 0.3$, the learning rate $\eta_{\text{QBM}} = 0.033$, $\eta_{\text{RBM}} = 0.1$, and $\eta_{\text{QSRBM}} = 0.085$, respectively. The sampling method that we use is Quantum Monte Carlo (QMC) [40]. Results show that the QSRBM model is close to RBM in recognition rate.

3.3. Training results and analysis

The training results are shown in figure 2 and figure 3. The fidelity of training the Bernoulli distribution sets is $F_{\text{QBM}} = 0.627$, $F_{\text{RBM}} = 0.924$, $F_{\text{QRBM}} = 0.618$, and $F_{\text{QSRBM}} = 0.977$ respectively, and the training fidelity of the MNIST



(a) (b)

Figure 3. (a) The training of MNIST dataset in classical RBM. After 1000 iterations, the algorithm converges and the final accuracy reaches 0.933. (b) The training of Bernoulli distribution sets in classical RBM. After 30 000 iterations, the algorithm converges and the final accuracy reaches 0.924.

dataset is $F_{\mathrm{QBM}} = 0.109$, $F_{\mathrm{RBM}} = 0.933$, $F_{\mathrm{QRBM}} = 0.845$, and $F_{\mathrm{QSRBM}} = 0.942$, respectively.

The results of training data with QBM are worse than that of QSRBM. The reason is that in QBM, the expected values of both positive and negative phases in equation (16) can be obtained by sampling through QMC. However, due to the lateral connection in the hidden layer of QBM, the effective positive phase expected value cannot be obtained by sampling with QMC method. When running Markov chain to approach the model expected value, the Markov chain cannot be stably distributed near the model expected value. In contrast, the positive phase value of QSRBM can be calculated accurately in equation (19). Taking this exact value as the starting point, the Markov chain can approximate the expected value of the model when it reaches the steady state. Then, the gradient descent algorithm is used to adjust the parameters by using the difference between the expected values of positive and negative phases. In this way, one can get better training results with QSRBM. On the other side, the training result of QSRBM is better than that of QRBM. The reason is that the lateral connections between visible layers are disconnected in QRBM, so QSRBM can exploit quantum superposition while QRBM cannot.

The fidelity of the QSRBM in figure 2(b) is slightly higher than that of the classical model when training the MNIST dataset. In order to save the computing power, we set the number of the hidden units to very small. As shown in figure 4, when the visible units are 5 and the hidden layer units are 2, the fidelity obtained by training is the highest. When the visible units are fixed, the hidden units determine the degree of the constraint of each training data on the model parameters. In the case of weight sharing, appropriately increasing the number of the hidden units may make the model learn better. On the other side, too many hidden units can lead to severe overfitting. Practically, the number of hidden units should be determined according to the actual

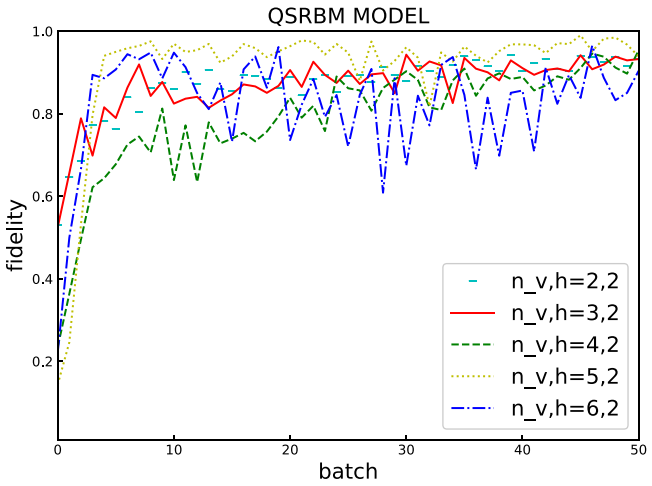


Figure 4. The change of fidelity with the configuration number of visible units and hidden units when training Bernoulli distribution sets with QSRBM.

situation. Therefore, one may infer that if the ratio of hidden units to visible units is slightly increased, the fidelity in our model on the training digital set may be probably higher.

From figures 2–4, one can see that the training curve in the quantum model is not as smooth as that in the classical model. This is because the entire training data is used to train the parameters only once. Each time when a batch is performed, the new data is trained and the fidelity may increase or decrease, but the gradient descent algorithm ensures that the trend of training results is gradual convergence.

4. Summary and discussion

In summary, we have studied training digital image sets with variable QBMs. Numerical simulations showed that the QSRBM works well, while the QBM model does not. There is still some

room for improvement in our model compared with other well-performed classifiers [41], such as increasing the number of hidden units to improve the recognition rate. Despite such few resource conditions, the results of QSRBM are still close to the classical model in training data and fulfilling tasks. This illustrates that machine learning with quantum algorithms is feasible and promising. Particularly, the training results of QSRBM are better than that of QBM with fewer computing results, which definitely means that the proper algorithm is important in quantum machine learning.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 11725524, and the Hubei Provincal Natural Science Foundation of China under Grant No. 2019CFA003.

References

[1] Max W and Hinton G E 2002 A new learning algorithm for mean field Boltzmann machines *Int. Conf. on Artificial Neural Networks* (Springer: Berlin) pp 351–7
[2] Dahl G *et al* 2010 Phone recognition with the mean-covariance restricted Boltzmann machine *Adv. Neural Inf. Process. Syst.* **1** 469–77
[3] Eslami S *et al* 2014 The shape Boltzmann machine: a strong model of object shape *Int. J. Comput. Vision* **107** 155–76
[4] Liao L, Jin W and Pavel R 2016 Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment *IEEE Trans. Ind. Electron.* **63** 7076–83
[5] Carleo G and Troyer M 2017 Solving the quantum many-body problem with artificial neural networks *Science* **355** 602–6
[6] Hinton G E 2002 Training products of experts by minimizing contrastive divergence *Neural Comput.* **14** 1771–800
[7] Carreira-Perpinan M A and Hinton G E 2005 On contrastive divergence learning *Int. Workshop on Artificial Intelligence and Statistics* (PMLR) pp 33–40 (http://proceedings.mlr.press/r5/carreira-perpinan05a.html)
[8] Tieleman T 2008 Training restricted Boltzmann machines using approximations to the likelihood gradient *Proc. 25th Int. Conf. on Machine Learning* pp 1064–71
[9] Hinton G E, Osindero S and Teh Y W 2006 A fast learning algorithm for deep belief nets *Neural Comput.* **18** 1527–54
[10] Cho K H, Raiko T and Ilin A 2013 Gaussian–Bernoulli deep Boltzmann machine *The 2013 Int. Joint Conf. on Neural Networks (IJCNN)* (Piscataway, NJ: IEEE) pp 1–7
[11] Salakhutdinov R, Tenenbaum J B and Torralba A 2012 Learning with hierarchical-deep models *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1958–71
[12] LeCun Y, Bengio Y and Hinton G E 2015 Deep learning *Nature* **521** 436–44
[13] Voulodimos A *et al* 2018 Deep learning for computer vision: a brief review *Comput. Intell. Neurosci.* **2018** 7068349
[14] Povey D *et al* 2011 The kaldi speech recognition toolkit *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* (IEEE Signal Processing Society, CONF) (https://www.fit.vut.cz/research/publication/11196)
[15] Tieleman T and Hinton G E 2009 Using fast weights to improve persistent contrastive divergence *Proc. 26th Annual Int. Conf. on Machine Learning* pp 1033–40
[16] Schuld M, Sinayskiy I and Petruccione F 2015 An introduction to quantum machine learning *Contemp. Phys.* **56** 172–85
[17] McLachlan A 1964 A variational solution of the time-dependent schrodinger equation *Mol. Phys.* **8** 39–44
[18] Kurowski K *et al* 2021 Applying a quantum annealing based restricted Boltzmann machine for mnist handwritten digit classification *Comput. Methods Sci. Technol.* **27** 99–107
[19] Zoufal C, Lucchi A and Woerner S 2021 Variational quantum Boltzmann machines *Quantum Mach. Intell.* **3** 1–15
[20] Verdon G, Broughton M and Biamonte J 2017 A quantum algorithm to train neural networks using low-depth circuits arXiv:171205304
[21] Biamonte J *et al* 2017 Quantum machine learning *Nature* **549** 195–202
[22] Adachi S H and Henderson M P 2015 Application of quantum annealing to training of deep neural networks arXiv:151006356
[23] Fischer A and Igel C 2012 An introduction to restricted Boltzmann machines *Iberoamerican Congress Pattern Recognition* (Berlin: Springer) pp 14–36
[24] Salakhutdinov R, Mnih A and Hinton G. E. 2007 Restricted Boltzmann machines for collaborative filtering *Proc. 24th Int. Conf. on Machine Learning* pp 791–8
[25] Chen C P *et al* 2015 Fuzzy restricted Boltzmann machine for the enhancement of deep learning *IEEE Trans. Fuzzy Syst.* **23** 2163–73
[26] Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc. Natl Acad. Sci.* **79** 2554–8
[27] Hinton G E 2007 Boltzmann machine *Scholarpedia* **2** 1668
[28] Larochelle H *et al* 2012 Learning algorithms for the classification restricted Boltzmann machine *J. Mach. Learning Res.* **13** 643–69
[29] Ruder S 2016 An overview of gradient descent optimization algorithms arXiv:160904747
[30] Larochelle H *et al* 2007 An empirical evaluation of deep architectures on problems with many factors of variation *Proc. 24th Int. Conf. on Machine Learning* pp 473–80
[31] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm arXiv:14114028
[32] Biamonte J D and Love P J 2008 Realizable Hamiltonians for universal adiabatic quantum computers *Phys. Rev.* A **78** 012352
[33] Lyakhova Y S, Polyakov E and Rubtsov A 2020 Effectively trainable semi-quantum restricted Boltzmann machine arXiv:200108997
[34] Amin M H *et al* 2018 Quantum Boltzmann machine *Phys. Rev.* X **8** 021050
[35] Forrester P J and Thompson C J 2014 The Golden–Thompson inequality: historical aspects and random matrix applications *J. Math. Phys.* **55** 023503
[36] Deng L 2012 The mnist database of handwritten digit images for machine learning research [best of the web] *IEEE Signal Process Mag.* **29** 141–2
[37] LeCun Y 1998 The mnist database of handwritten digits (http://yann.lecun.com/exdb/mnist/)
[38] Hu Z and Hong L J 2013 Kullback-Leibler divergence constrained distributionally robust optimization *Optimization Online* 1695–724
[39] Collobert R, Bengio S and Mariéthoz J 2002 Torch: a modular machine learning software library *Technical-Report* IDIAP (https://infoscience.epfl.ch/record/82802)
[40] Crosson E and Harrow A W 2016 Simulated quantum annealing can be exponentially faster than classical simulated annealing *IEEE 57th Annual Symp. on Foundations of Computer Science (FOCS)* (IEEE) pp 714–23
[41] Xiao H, Rasul K and Vollgraf R 2017 Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms arXiv:170807747