Quantum Machine Intelligence (2021) 3: 7
https://doi.org/10.1007/s42484-020-00033-7

RESEARCH ARTICLE

Variational quantum Boltzmann machines

Christa Zoufal[1,2] · Aurélien Lucchi[2] · Stefan Woerner[1]

Received: 9 July 2020 / Accepted: 17 December 2020 / Published online: 22 February 2021

Abstract
This work presents a novel realization approach to quantum Boltzmann machines (QBMs). The preparation of the required Gibbs states, as well as the evaluation of the loss function's analytic gradient, is based on variational quantum imaginary time evolution, a technique that is typically used for ground-state computation. In contrast to existing methods, this implementation facilitates near-term compatible QBM training with gradients of the actual loss function for arbitrary parameterized Hamiltonians which do not necessarily have to be fully visible but may also include hidden units. The variational Gibbs state approximation is demonstrated with numerical simulations and experiments run on real quantum hardware provided by IBM Quantum. Furthermore, we illustrate the application of this variational QBM approach to generative and discriminative learning tasks using numerical simulation.

Keywords Quantum machine learning · Variational quantum imaginary time evolution · Generative learning · Discriminative learning

1 Introduction

Boltzmann machines (BMs) (Ackley et al. 1985; Du and Swamy 2019) offer a powerful framework for modelling probability distributions. These types of neural networks use an undirected graph structure to encode relevant information. More precisely, the respective information is stored in bias coefficients and connection weights of network nodes, which are typically related to binary spin-systems and grouped into those that determine the output, the visible nodes, and those that act as latent variables, the hidden nodes. Furthermore, the network structure is linked to an energy function which facilitates the definition of a probability distribution over the possible node configurations by using a concept from statistical mechanics, i.e., Gibbs states (Boltzmann 1877; Gibbs 1902). The aim of BM training is to learn a set of weights such that the resulting model approximates a target probability distribution which is implicitly given by training data. This setting can be formulated as discriminative as well as generative learning task (Liu and Webb 2010). Applications have been studied in a large variety of domains such as the analysis of quantum many-body systems, statistics, biochemistry, social networks, signal processing, and finance; see, e.g., (Torlai et al. 2018; Carleo et al. 2018; Carleo and Troyer 2017; Nomura et al. 2017; Anshu et al. 2020; Melko et al. 2019; Hrasko et al. 2015; Tubiana et al. 2019; Liu et al. 2013; Mohamed and Hinton 2010; Assis et al. 2018). However, BMs are complicated to train in practice because the loss function's derivative requires the evaluation of a normalization factor, the partition function, that is generally difficult to compute. Usually, it is approximated using Markov chain Monte Carlo methods which may require long runtimes until convergence (Carreira-Perpinan and Hinton 2005; Murphy 2012). Alternatively, the gradients could be estimated approximately using contrastive divergence (Hinton 2002) or pseudo-likelihood (Besag 1975) potentially leading to inaccurate results (Tieleman 2008; Sutskever and Tieleman 2010).

Quantum Boltzmann machines (QBMs) (Amin et al. 2018) are a natural adaption of BMs to the quantum computing framework. Instead of an energy function with nodes being

✉ Stefan Woerner
wor@zurich.ibm.com

Christa Zoufal
ouf@zurich.ibm.com

[1] IBM Quantum, IBM Research – Zurich, Zurich, Switzerland
[2] ETH Zurich, Zurich, Switzerland

represented by binary spin values, QBMs define the underlying network using a Hermitian operator, a parameterized Hamiltonian:

$$H_\theta = \sum_{i=0}^{p-1} \theta_i h_i,$$

with $\theta \in \mathbb{R}^p$ and $h_i = \bigotimes_{j=0}^{n-1} \sigma_{j,i}$ for $\sigma_{j,i} \in \{I, X, Y, Z\}$ acting on the j^{th} qubit. The network nodes are hereby characterized by the Pauli matrices $\sigma_{j,i}$. This Hamiltonian relates to a quantum Gibbs state, $\rho^{\text{Gibbs}} = e^{-H_\theta/(\text{k}_\text{B}\text{T})}/Z$ with k_B and T denoting the Boltzmann constant and the system temperature, and $Z = \text{Tr}\left[e^{-H_\theta/(\text{k}_\text{B}\text{T})}\right]$. It should be noted that those qubits which determine the model output are referred to as visible and those which act as latent variables as hidden qubits. The aim of the model is to learn Hamiltonian parameters such that the resulting Gibbs state reflects a given target system. In contrast to BMs, this framework allows the use of quantum structures which are potentially inaccessible classically. Equivalently to the classical model, QBMs are suitable for discriminative as well as generative learning.

We present here a QBM implementation that circumvents certain issues which emerged in former approaches. The first paper on QBMs (Amin et al. 2018) and several subsequent works (Anschütz and Cao 2019; Kieferová and Wiebe 2017; Kappen 2020; Wiebe and Wossnig 2019) are incompatible with efficient evaluation of the loss function's analytic gradients if the given model has hidden qubits and

$$\exists j : \left[H_\theta, \frac{\partial H_\theta}{\partial \theta_j}\right] \neq 0.$$

Instead, the use of hidden qubits is either avoided, i.e., only fully visible settings are considered (Kieferová and Wiebe 2017; Kappen 2020; Wiebe and Wossnig 2019), or the gradients are computed with respect to an upper bound of the loss (Amin et al. 2018; Anschütz and Cao 2019; Kieferová and Wiebe 2017), which is based on the Golden-Thompson inequality (Thompson 1965; Golden 1965). It should be noted that training with an upper bound renders the use of transverse Hamiltonian components, i.e., off-diagonal Pauli terms, difficult and imposes restrictions on the compatible models.

Furthermore, we would like to point out that, in general, it is not trivial to evaluate a QBM Hamiltonian with a classical computer, i.e., using exact simulation with quantum Monte Carlo methods (Troyer and Wiese 2005), because the underlying Hamiltonian can suffer from the so-called sign-problem (Hangleiter et al. 2020; Okunishi and Harada 2014; Li et al. 2016; Alet et al. 2016; Li et al. 2015). As already discussed in Ortiz et al. (2001), evaluations on quantum computers can avoid this problem.

Our QBM implementation works for generic Hamiltonians H_θ with real coefficients θ and arbitrary Pauli terms h_i, and furthermore, is compatible with near-term, gate-based quantum computers. The method exploits *Variational Quantum Imaginary Time Evolution* (McArdle et al. 2019; Yuan et al. 2019) (VarQITE), which is based on McLachlan's variational principle (McLachlan 1964), to not only prepare approximate Gibbs states, $\rho_\omega^{\text{Gibbs}}$, but also to train the model with gradients of the actual loss function. During each step of the training, we use VarQITE to generate an approximation to the Gibbs state underlying H_θ and to enable automatic differentiation for computing the gradient of the loss function which is needed to update θ. This variational QBM algorithm (VarQBM) is inherently normalized which implies that the training does not require the explicit evaluation of the partition function.

We focus on training quantum Gibbs states whose sampling behavior reflects a classical probability distribution. However, the scheme could be easily adapted to an approximate quantum state preparation scheme by using a loss function which is based on the quantum relative entropy (Kieferová and Wiebe 2017; Kappen 2020; Wiebe and Wossnig 2019). Hereby, the approximation to ρ^{Gibbs} is fitted to a given target state ρ^{data}. Notably, this approach is not necessarily suitable for learning classical distributions. More precisely, we do not need to train a quantum state that captures all features of the density matrix ρ^{data} but only those which determine the sampling probability. It follows that fitting the full density matrix may impede the training.

The remainder of this paper is structured as follows. Firstly, we review classical BMs and VarQITE in Section 2. Then, we outline VarQBM in Section 3. Next, we illustrate the feasibility of the Gibbs state preparation and present QBM applications in Section 4. Finally, a conclusion and an outlook are given in Section 5.

2 Preliminaries

This section introduces the concepts which form the basis of our VarQBM algorithm. First, classical BMs are presented in Section 2.1. Then, we discuss VarQITE, the algorithm that VarQBM uses for approximate Gibbs state preparation, in Section 2.2.

2.1 Boltzmann machines

Here, we will briefly review the original concept of classical BMs (Ackley et al. 1985). A BM represents a network model that stores the learned knowledge in connection weights between network nodes. More explicitly, the connection weights are trained to generate outcomes according to a probability distribution of interest, e.g., to generate samples which are similar to given training samples or to output correct labels depending on input data samples.

Typically, this type of neural network is related to an Ising-type model (Ising 1925; Peierls 1936) such that each node i corresponds to a binary variable $z_i \in \{-1, +1\}$. Now, the set of nodes may be split into visible and hidden nodes representing observed and latent variables, respectively. Furthermore, a certain configuration $z = \{v, h\}$ of all nodes—visible and hidden—determines an energy, which is given as

$$E_{z=\{v,\,h\}} = -\sum_i \tilde{\theta}_i z_i - \sum_{i,j} \theta_{ij} z_i z_j,$$

with $\tilde{\theta}_i, \theta_{ij} \in \mathbb{R}$ denoting the weights and z_i representing the value taken by node i. It should be noted that the parameters θ_{ij} correspond to the weights of connections between different nodes. More explicitly, if two nodes are connected in the network, then a respective term appears in the energy function. The probability to observe a configuration v of the visible nodes is defined as

$$p_v^{BM} = \frac{\sum_n e^{-E_{z=\{v,\,h\}}/k_\mathrm{B}\mathrm{T}}}{Z}, \tag{1}$$

where k_B is the Boltzmann constant, T the system temperature and Z the canonical partition function

$$Z = \sum_{z=\{v,\,h\}} e^{-E_z/(k_\mathrm{B}\mathrm{T})}.$$

We would like to point out that BMs adopt a concept from statistical mechanics. Suppose a closed system that is in thermal equilibrium with a coupled heat bath at constant temperature. The possible configuration space is determined by the canonical ensemble, i.e., the probability for observing a configuration is given by the Gibbs distribution (Boltzmann 1877; Gibbs 1902) which corresponds to Eq. 1.

Now, the goal of a BM is to fit the target probability distribution p^{data} with p^{BM}. Typically, this training objective is achieved by optimizing the cross-entropy

$$L = -\sum_v p_v^{\mathrm{data}} \log p_v^{BM}. \tag{2}$$

In theory, fully connected BMs have interesting representation capabilities (Ackley et al. 1985; Younes 1996; Fischer and Igel 2012a), i.e., they are universal approximators (Roux and Bengio 2010). However, in practice they are difficult to train as the optimization easily gets expensive. Thus, it has become common practice to restrict the connectivity between nodes which relates to Restricted Boltzmann machines (RBMs) (Montúfar 2018). Furthermore, several approximation techniques, such as contrastive divergence (Hinton 2002), have been developed to facilitate BM training. However, these approximation techniques typically still face issues such as long computation time due to a large amount of required Markov chain steps or poor compatibility with multimodal probability distributions (Murphy 2012). For further details, we refer the interested reader to Hinton (2012), Fischer and Igel (2012b), and Fischer (2015).

2.2 Variational quantum imaginary time evolution

Imaginary time evolution (ITE) (Magnus 1954) is an approach that is well known for (classical) ground-state computation (McArdle et al. 2019; Gupta et al. 2002; Auer et al. 2001) but—as suggested by some prior literature on ITE (Matsui 1998; Khalkhali and Marcolli 2008; Motta and et al. 2020; McArdle et al. 2019; Yuan et al. 2019)—may also be used for Gibbs state preparation.

Suppose a starting state $|\psi_0\rangle$ and a time-independent Hamiltonian $H = \sum_{i=0}^{p-1} \theta_i h_i$ with real coefficients θ_i and Pauli terms h_i. Then, the normalized ITE propagates $|\psi_0\rangle$ with respect to H for time τ according to

$$|\psi_\tau\rangle = C(\tau)\, e^{-H\tau} |\psi_0\rangle,$$

where $C(\tau) = 1/\sqrt{\mathrm{Tr}\left[e^{-2H\tau}|\psi_0\rangle\langle\psi_0|\right]}$ is a normalization. The differential equation that describes this evolution is the Wick-rotated Schrödinger equation:

$$\frac{d|\psi_\tau\rangle}{d\tau} = -(H - E_\tau)\,|\psi_\tau\rangle, \tag{3}$$

where $E_\tau = \langle\psi_\tau|H|\psi_\tau\rangle$ originates from the normalization of $|\psi_\tau\rangle$. The terms in $e^{-H\tau}$, corresponding to small eigenvalues of H, decay slower than the ones corresponding to large eigenvalues. Due to the continuous normalization, the smallest eigenvalue dominates for $\tau \to \infty$. Thus, $|\psi_\tau\rangle$ converges to the ground state of H given that there is some overlap between the ground and starting states. Furthermore, if ITE is only evolved to a finite time, $\tau = 1/2\,(k_\mathrm{B}\mathrm{T})$, then it enables the preparation of Gibbs states; see Section 3.1.

As introduced in McArdle et al. (2019) and Yuan et al. (2019), an approximate ITE can be implemented on a gate-based quantum computer by using McLachlan's variational principle (McLachlan 1964). The basic idea of the method is to introduce a parameterized trial state $|\psi_\omega\rangle$ and to project the temporal evolution of $|\psi_\tau\rangle$ to the parameters, i.e., $\omega := \omega(\tau)$. We refer to this algorithm as VarQITE and, now, discuss it in more detail.

First, we define an input state $|\psi_{\mathrm{in}}\rangle$ and a quantum circuit $V(\omega) = U_q(\omega_q) \cdots U_1(\omega_1)$ with parameters $\omega \in \mathbb{R}^q$ to generate the parameterized trial state

$$|\psi_\omega\rangle := V(\omega)\,|\psi_{\mathrm{in}}\rangle.$$

Now, McLachlan's variational principle

$$\delta\,\|(d/d\tau + H - E_\tau)\,|\psi_\omega\rangle\| = 0 \tag{4}$$

determines the time propagation of the parameters $\omega(\tau)$. This principle aims to minimize the distance between the right-hand side of Eq. 3 and the change $d|\psi_\omega\rangle/d\tau$. Equation 4 leads to a system of linear equations for $\dot{\omega} = d\omega/d\tau$, i.e.,

$$A\dot{\omega} = C \tag{5}$$

with

$$\begin{aligned} A_{pq}\left(\tau\right) &= \mathrm{Re}\left(\mathrm{Tr}\left[\frac{\partial V^\dagger(\omega(\tau))}{\partial\omega(\tau)_p}\frac{\partial V(\omega(\tau))}{\partial\omega(\tau)_q}\rho_{\mathrm{in}}\right]\right) \\ C_p\left(\tau\right) &= -\sum_i \theta_i \mathrm{Re}\left(\mathrm{Tr}\left[\frac{\partial V^\dagger(\omega(\tau))}{\partial\omega(\tau)_p} h_i V\left(\omega\left(\tau\right)\right)\rho_{\mathrm{in}}\right]\right), \end{aligned} \tag{6}$$

where $\mathrm{Re}\left(\cdot\right)$ denotes the real part and $\rho_{\mathrm{in}} = |\psi_{\mathrm{in}}\rangle\langle\psi_{\mathrm{in}}|$. The vector C describes the derivative of the system energy $\langle\psi_\omega|H|\psi_\omega\rangle$ and A is proportional to the classical Fisher information matrix, a metric tensor that reflects the system's information geometry (Koczor and Benjamin 2019). To evaluate A and C, we compute expectation values with respect to quantum circuits of a particular form which is illustrated and discussed in Appendix A.

This evaluation is compatible with arbitrary parameterized unitaries in $V\left(\omega\right)$ because all unitaries can be written as $U\left(\omega\right) = e^{iM(\omega)}$, where $M\left(\omega\right)$ denotes a parameterized Hermitian matrix. Furthermore, Hermitian matrices can be decomposed into weighted sums of Pauli terms, i.e., $M\left(\omega\right) = \sum_p m_p\left(\omega\right)h_p$ with $m_p\left(\omega\right) \in \mathbb{R}$ and $h_p = \bigotimes_{j=0}^{n-1}\sigma_{j,p}$ for $\sigma_{j,p} \in \{I, X, Y, Z\}$ (Nielsen and Chuang 2010) acting on the j^{th} qubit. Thus, the gradients of $U_k\left(\omega_k\right)$ are given by

$$\frac{\partial U_k\left(\omega_k\right)}{\partial\omega_k} = \sum_p i\frac{\partial m_{k,p}\left(\omega_k\right)}{\partial\omega_k}U_k\left(\omega_k\right)h_{k_p}.$$

This decomposition allows us to compute A and C with the techniques described in Somma et al. (2002), McArdle et al. (2019), and Yuan et al. (2019). Furthermore, it should be noted that Eq. 5 is often ill-conditioned and may, thus, require the use of regularized regression methods; see Section 4.1.

Now, we can use, e.g., an explicit Euler method to evolve the parameters as

$$\omega\left(\tau\right) = \omega\left(0\right) + \sum_{j=1}^{\tau/\delta\tau}\dot{\omega}\left(_j\delta\tau\right)\delta\tau.$$

3 Quantum Boltzmann Machine algorithm

A QBM is defined by a parameterized Hamiltonian $H_\theta = \sum_{i=0}^{p-1}\theta_i h_i$ where $\theta \in \mathbb{R}^p$ and $h_i = \bigotimes_{j=0}^{n-1}\sigma_{j,i}$ for $\sigma_{j,i} \in \{I, X, Y, Z\}$ acting on the j^{th} qubit. Equivalently to classical BMs, QBMs are typically represented by an Ising model (Ising 1925), i.e., a 2-local system (Bravyi et al. 2006) with nearest-neighbor coupling that is defined with regard to a particular grid. In principle, however, any Hamiltonian compatible with Boltzmann distributions could be used.

In contrast to BMs, the network nodes, given by the Pauli terms $\sigma_{j,i}$, do not represent the visible and hidden units. These are defined with respect to certain sub-sets of qubits. More explicitly, those qubits which determine the output of the QBM are the visible qubits, whereas the others correspond to the hidden qubits. Now, the probability to measure a configuration v of the visible qubits is defined with respect to a projective measurement $\Lambda_v = |v\rangle\langle v| \otimes I$ on the quantum Gibbs state

$$\rho^{\mathrm{Gibbs}} = \frac{e^{-H_\theta/(\mathrm{k_B T})}}{Z}$$

with $Z = \mathrm{Tr}\left[e^{-H_\theta/(\mathrm{k_B T})}\right]$, i.e., the probability to measure $|v\rangle$ is given by

$$p_v^{\mathrm{QBM}} = \mathrm{Tr}\left[\Lambda_v\rho^{\mathrm{Gibbs}}\right].$$

For the remainder of this work, we assume that Λ_v refers to projective measurements with respect to the computational basis of the visible qubits. Thus, the configuration v is determined by $v_i \in \{0, 1\}$. It should be noted that this formulation does not require the evaluation of the configuration of the hidden qubits.

Our goal is to train the Hamiltonian parameters θ such that the sampling probabilities of the corresponding ρ^{Gibbs} reflect the probability distribution underlying given classical training data. For this purpose, the same loss function as described in the classical case (see Eq. 2) can be used

$$L = -\sum_v p_v^{\mathrm{data}}\log p_v^{\mathrm{QBM}}, \tag{7}$$

where p_v^{data} denotes the occurrence probability of item v in the training dataset.

To enable efficient training, we want to evaluate the derivative of L with respect to the Hamiltonian parameters. Unlike existing QBM implementations, VarQBM facilitates the use of analytic gradients of the loss function given in Eq. 7 for generic QBMs. The presented algorithm involves the following steps. First, we use VarQITE to approximate the Gibbs state; see Section 3.1 for further details. Then, we compute the gradient of L to update the parameters θ with automatic differentiation, as is discussed in Section 3.2. The parameters are trained with a classical optimization routine where one training step consists of the Gibbs state preparation with respect to the current parameter values and a consecutive parameter update, as illustrated in Fig. 1.

In the remainder of this section, we discuss Gibbs state preparation with VarQITE in Section 3.1 and VarQBM in more detail in Section 3.2.

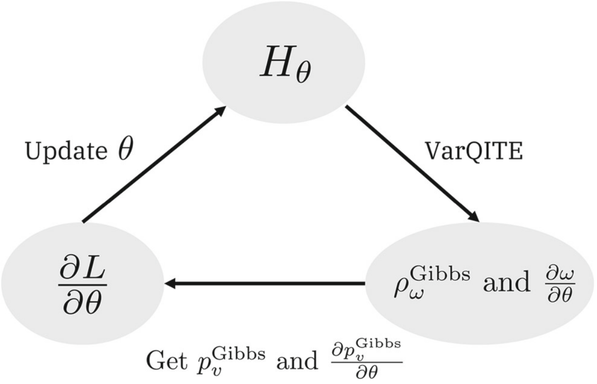
H_θ
Update θ
VarQITE
$\frac{\partial L}{\partial \theta}$
$\rho_\omega^{\text{Gibbs}}$ and $\frac{\partial \omega}{\partial \theta}$
Get p_v^{Gibbs} and $\frac{\partial p_v^{\text{Gibbs}}}{\partial \theta}$

Fig. 1 The VarQBM training includes the following steps. First, we need to fix the Pauli terms for H_θ and choose initial parameters θ. Then, VarQITE is used to generate $\rho_\omega^{\text{Gibbs}}$ and compute $\partial\omega/\partial\theta$. The quantum state and the derivative are needed to evaluate p_v^{QBM} and $\partial p_v^{\text{QBM}}/\partial\theta$. Now, we can find $\partial L/\partial\theta$ to update the Hamiltonian parameters with a classical optimizer

3.1 Gibbs state preparation with VarQITE

The Gibbs state ρ^{Gibbs} describes the probability density operator of the configuration space of a system in thermal equilibrium with a heat bath at constant temperature T (Gibbs 2010). Originally, Gibbs states were studied in the context of statistical mechanics but, as shown in (Pauli 1927), the density operator also facilitates the description of quantum statistics.

Gibbs state preparation can be approached from different angles. Hereby, different techniques not only have different strengths but also different drawbacks. Some schemes (Temme et al. 2011; Yung and Aspuru-Guzik 2012; Poulin and Wocjan 2009) use quantum phase estimation (Abrams and Lloyd 1999) as a subroutine, which is likely to require error-corrected quantum computers. Other methods enable the evaluation of quantum thermal averages (Motta and et al. 2020; Brandão and Kastoryano 2019; Kastoryano and Brandão 2016) for states with finite correlations. However, since QBM-related states may exhibit long-range correlations, these methods are not the first choice for the respective preparation. A thermalization-based approach is presented in Anschütz and Cao (2019), where the aim is to prepare a quantum Gibbs state by coupling the state register to a heat bath given in the form of an ancillary quantum register. Correct preparation requires a thorough study of suitable ancillary registers for a generic Hamiltonian as the most useful ancilla system is not a priori known. Furthermore, variational Gibbs state preparation methods which are based on the fact that Gibbs states minimize the free energy of a system at constant temperature have been presented in Wu and Hsieh (2019) and Chowdhury et al. (2020). The goal is to fit a parameterized quantum state that minimizes the free energy. Here, the difficulty is to estimate the von Neumann entropy in every step of the state preparation. An algorithm which aims to enable efficient von Neuman entropy estimation is given in Chowdhury et al. (2020). However, it requires the application of quantum amplitude estimation (Brassard et al. 2002), as well as matrix exponentiation of the input state, and thus is not well suited for near-term quantum computing applications.

In contrast to these Gibbs state preparation schemes, VarQITE is compatible with near-term quantum computers, and neither is limited to states with finite correlations nor requires ambiguous ancillary systems. In the following, we discuss how VarQITE can be utilized to generate an approximation of the Gibbs state ρ^{Gibbs} for a generic $n-$qubit Hamiltonian $H_\theta = \sum_{i=0}^{p-1}\theta_i h_i$ with $\theta \in \mathbb{R}^p$ and $h_i = \bigotimes_{j=0}^{n-1}\sigma_{j,i}$ for $\sigma_{j,i} \in \{I, X, Y, Z\}$ acting on the j_i^{th} qubit.

First, we need to choose a suitable variational quantum circuit $V(\omega)$, $\omega \in \mathbb{R}^q$, and set of initial parameters $\omega(0)$ such that the initial state is

$$|\psi_0\rangle = V(\omega(0))\,|0\rangle^{\otimes 2n} = |\phi^+\rangle^{\otimes n} \tag{8}$$

where $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ represents a Bell state. We define two n-qubit sub-systems a and b such that the first and second qubits of each $|\phi^+\rangle$ are in a and b, respectively. Accordingly, an effective $2n$-qubit Hamiltonian $H_{\text{eff}} = H_\theta^a + I^b$, where H_θ and I act on sub-system a and b, is considered. It should be noted that tracing out sub-system b from $|\psi_0\rangle$ results in an n-dimensional maximally mixed state:

$$\text{Tr}_b\left[|\phi^+\rangle^{\otimes n}\right] = \frac{1}{2^n}I.$$

Now, the Gibbs state approximation $\rho_\omega^{\text{Gibbs}}$ can be generated by propagating the trial state with VarQITE with respect to H_{eff} for $\tau = 1/2\,(\text{k}_\text{B}\text{T})$. The resulting state

$$|\psi_\omega\rangle = V(\omega(\tau))\,|0\rangle^{\otimes 2n}$$

gives an approximation for the Gibbs state of interest

$$\rho_\omega^{\text{Gibbs}} = \text{Tr}_b\left[|\psi(\omega(\tau))\rangle\langle\psi(\omega(\tau))|\right] \approx \frac{e^{-H_\theta/(\text{k}_\text{B}\text{T})}}{Z}$$

by tracing out the ancillary system b. We would like to point out that the VarQITE propagation relates ω to θ via the energy derivative C given in Eq. 6.

Notably, this is an approximate state preparation scheme that relies on the representation capabilities of $|\psi_\omega\rangle$. However, since the algorithm is employed in the context of machine learning, we do not necessarily require perfect state preparation. The noise may even improve the training, as discussed, e.g., in (Noh et al. 2017).

McLachlan's variational principle is not only the key component for Gibbs state preparation. It also enables the QBM training with gradients of the actual loss function for generic Pauli terms in H_θ, even if some of the qubits are hidden. Further details are given in Section 3.2.

3.2 Variational QBM

In the following, VarQBM and the respective utilization of McLachlan's variational principle and VarQITE are discussed. We consider training data that takes at most 2^n different values and is distributed according to a discrete probability distribution p^{data}. The aim of a QBM is to train the parameters of H_θ such that the sampling probability distribution of the corresponding $\rho_\omega^{\text{Gibbs}} = e^{-H_\theta/(\text{k}_\text{B}\text{T})}/Z$ for $|v\rangle,\ v \in 0, \ldots, 2^n - 1$ with

$$p_v^{\text{QBM}} = \text{Tr}\left[\Lambda_v \rho_\omega^{\text{Gibbs}}\right],$$

approximates p^{data}. The QBM model is trained to represent p^{data} by minimizing the loss, given in Eq. 7, with respect to the Hamiltonian parameters θ, i.e.,

$$\min_\theta L = \min_\theta \left(-\sum_v p_v^{\text{data}} \log p_v^{\text{QBM}}\right).$$

Now, VarQBM facilitates gradient-based optimization with the derivative of the actual loss function

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial\left(-\sum_v p_v^{\text{data}} \log p_v^{\text{QBM}}\right)}{\partial \theta_i} = -\sum_v p_v^{\text{data}} \frac{\partial p_v^{\text{QBM}}/\partial\theta_i}{p_v^{\text{QBM}}} \tag{9}$$

by using the chain rule, i.e., automatic differentiation. More precisely, the gradient of L can be computed by using the chain rule for

$$\frac{\partial p_v^{\text{QBM}}}{\partial \theta_i} = \frac{\partial p_v^{\text{QBM}}}{\partial \omega(\tau)} \frac{\partial \omega(\tau)}{\partial \theta_i} = \sum_{k=0}^{q-1} \frac{\partial p_v^{\text{QBM}}}{\partial \omega_k(\tau)} \frac{\partial \omega_k(\tau)}{\partial \theta_i}. \tag{10}$$

Firstly, $\partial p_v^{\text{QBM}}/\partial \omega_k(\tau) = \partial \text{Tr}\left[\Lambda_v \rho_\omega^{\text{Gibbs}}\right]/\partial \omega_k(\tau)$ can be evaluated with quantum gradient methods discussed in Farhi and Neven (2018), Mitarai et al. (2018), Dallaire-Demers and Killoran (2018), Schuld et al. (2019), and Zoufal et al. (2019) because the term has the following form:

$$\partial \text{Tr}\left[\hat{O}|\phi(\alpha)\rangle\langle\phi(\alpha)|\right]/\partial\alpha.$$

Secondly, $\partial\omega_k(\tau)/\partial\theta_i$ is evaluated by computing the derivative of Eq. 5 with respect to the Hamiltonian parameters:

$$\frac{\partial A\dot{\omega}(\tau)}{\partial\theta_i} = \frac{\partial C}{\partial\theta_i}.$$

This gives the following system of linear equations:

$$A\left(\frac{\partial\dot{\omega}(\tau)}{\partial\theta_i}\right) = \frac{\partial C}{\partial\theta_i} - \left(\frac{\partial A}{\partial\theta_i}\right)\dot{\omega}(\tau). \tag{11}$$

Equation 11 is just as Eq. 5 prone to being ill- conditioned. Thus, the use of regularization schemes may be required.

Now, solving for $\partial\dot{\omega}(\tau)/\partial\theta_i$ in every time step of the Gibbs state preparation enables the use of, e.g., an explicit Euler method to get

$$\begin{aligned}\frac{\partial\omega_k(\tau)}{\partial_{\theta_i}} &= \frac{\partial\omega_k(\tau-\delta\tau)}{\partial_{\theta_i}} + \frac{\partial\dot{\omega}_k(\tau-\delta\tau)}{\partial_{\theta_i}}\delta\tau \\ &= \frac{\partial\omega_k(0)}{\partial_{\theta_i}} + \sum_{j=1}^{\tau/\delta\tau} \frac{\partial\dot{\omega}_k(j\delta\tau)}{\partial_{\theta_i}}\delta\tau.\end{aligned} \tag{12}$$

We discuss the structure of the quantum circuits used to evaluate $\partial_{\theta_i} A$ and $\partial_{\theta_i} C$, in Appendix A.

In principle, the gradient of the loss function could also be approximated with a finite difference method. If the number of Hamiltonian parameters is smaller than the number of trial state parameters, this requires less evaluation circuits. However, given a trial state that has less parameters than the respective Hamiltonian, the automatic differentiation scheme presented in this section is favorable in terms of the number of evaluation circuits. A more detailed discussion on this topic can be found in Appendix B.

An outline of the Gibbs state preparation and evaluation of $\partial\omega_k(\tau)/\partial_{\theta_i}$ with VarQITE is presented in Algorithm 1.

Algorithm 1 VarQITE for VarQBM

```
input
$H_{\text{eff}} = H_\theta^a + I^b$
$\tau = 1/2\,(\text{k}_\text{B}\text{T})$
$|\psi(\omega(0))\rangle = V(\omega(0))\,|0\rangle^{\otimes 2n} = |\phi^+\rangle^{\otimes n}$
with $|\phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$
procedure
for $t \in \{\delta\tau, 2\delta\tau, \ldots, \tau\}$ do
    Evaluate $A(t)$ and $C(t)$
    Solve $A\dot{\omega}(t) = C$
    for $i \in \{0, \ldots, p-1\}$ do
        Evaluate $\partial_{\theta_i} C$ and $\partial_{\theta_i} A$
        Solve $A\left(\partial_{\theta_i}\dot{\omega}(t)\right) = \partial_{\theta_i} C - \left(\partial_{\theta_i} A\right)\dot{\omega}(t)$
        Compute $\partial_{\theta_i}\omega(t) = \partial_{\theta_i}\omega(t-\delta\tau) + \partial_{\theta_i}\dot{\omega}(t)\,\delta\tau$
    end for
    Compute $\omega(t+\delta\tau) = \omega(t) + \dot{\omega}(t)\,\delta\tau$
end for
return $\omega(\tau),\ \partial\omega(\tau)/\partial\theta$
```

Now, using a classical optimizer, such as truncated Newton (Dembo and Steihaug 1983) or Adam (Kingma and Ba 2015), allows the parameters θ to be updated according to $\partial L/\partial\theta$ from Eq. 9. The VarQBM training is illustrated in Fig. 1.

4 Results

In this section, the Gibbs state preparation with VarQITE is demonstrated using numerical simulation as well as the quantum hardware provided by IBM Quantum (IBM Q Experience 2020). Furthermore, we present numerically simulated QBM training results for a generative and a discriminative learning task. First, aspects which are relevant for the practical implementation are discussed in Section 4.1. Next, experiments of quantum Gibbs state preparation with VarQITE are shown in Section 4.2. Then, we illustrate the training of a QBM with the goal to generate a state which exhibits the sampling behavior of a Bell state (see Section 4.3), and to classify fraudulent credit card transactions (Section 4.4).

4.1 Methods

To begin with, we discuss the choice of a suitable parameterized trial state consisting of $V(\omega)$ and $|\psi_{\text{in}}\rangle$. Most importantly, the initial state $|\psi_{\text{in}}\rangle$ must not be an eigenstate of $V(\omega)$ as this would imply that the circuit could only act trivially onto the state and $|\psi_{\text{in}}\rangle$ as well as $V(\omega(0))$ must be chosen such that the initial trial state suffices the form given in Eq. 8. As there is currently little known about what types of Hamiltonians map particularly well to what types of trial states, this work employs generic hardware-efficient Ansätze. These are chosen such that the number of hidden nodes and the entanglement between the qubits are not too large as these might lead to barren plateaus (Marrero et al. 2020). Furthermore, the state needs to be able to represent a sufficiently accurate approximation of the target state. If we have to represent, e.g., a non-symmetric Hamiltonian, the chosen trial state needs to be able to generate non-symmetric states. Numerical experiments indicate that states which exhibit complex correlations require Ansätze with higher depth, i.e., more parameters. Moreover, $V(\omega)$ should not exhibit too much symmetry as this may lead to a singular A which in turn causes ill-conditioning of Eq. 5. Assume, e.g., that all entries of C are zero and, thus, that Eq. 5 is homogeneous. If A is singular, infinitely many solutions exist and it is difficult for the algorithm to estimate which path to choose. If A is non-singular, the solution is $\dot{\omega} = 0$ and the evolution stops although we might have only reached a local extreme point. Another possibility to cope with ill-conditioned systems of linear equations are least-squares methods in combination with regularization schemes. We test Tikhonov regularization (Tikhonov et al. 1995) and Lasso regularization (Tibshirani 2011) with an automatic parameter evaluation based on L-curve fitting (Hansen 2000), as well as an ϵ-perturbation of the diagonal, i.e., $A \rightarrow A + \epsilon I$. It turns out that all regularization methods perform similarly well.

The results discussed in this section employ Tikhonov regularization. Furthermore, we use trial states which are parameterized by Pauli-rotation gates. Therefore, the gradients of the QBM probabilities with respect to the trial state parameters

$$\frac{\partial p_v^{\text{QBM}}}{\partial \omega_k(\tau)} = \frac{\partial \text{Tr}\left[\Lambda_v \rho_\omega^{\text{Gibbs}}\right]}{\partial \omega_k(\tau)}$$

can be computed using a $\pi/2-$shift method which is, e.g., described in Zoufal et al. (2019). All experiments employ an additional qubit $|0\rangle_{\text{add}}$ and parameter ω_{add} to circumvent a potential phase mismatch between the target $|\psi_\tau\rangle$ and the trained state $|\psi(\omega(\tau))\rangle$ (McArdle et al. 2019; Yuan et al. 2019; Koczor and Benjamin 2019) by applying

$$R_Z(\omega_{\text{add}})\,|0\rangle_{\text{add}}.$$

Notably, the additional parameter increases the dimension of A and C by 1. The effective temperature, which in principle acts as a scaling factor on the Hamiltonian parameters, is set to $(\text{k}_\text{B}\text{T}) = 1$ in all experiments.

4.2 Gibbs state preparation

We verify VarQITE is able to generate suitable approximations to Gibbs states by illustrating the convergence of the fidelity between the prepared and the target state for several example Hamiltonians. This experiment is run for a one- and a two-qubit Hamiltonian using simulation as well as an actual quantum computer and for Hamiltonians with more qubits using only a simulated backend. The results for the latter are given in Appendix C.

The Gibbs state preparation for the following Hamiltonians

$$\begin{aligned} H_1 &= 1.0Z, \\ H_2 &= 1.0ZZ - 0.2ZI - 0.2IZ + 0.3XI + 0.3IX. \end{aligned}$$

corresponding to

$$\begin{aligned} \rho_1^{\text{Gibbs}} &= \begin{pmatrix} 0.12 & 0. \\ 0. & 0.88 \end{pmatrix}, \\ \rho_2^{\text{Gibbs}} &= \begin{pmatrix} 0.10 & -0.06 & -0.06 & 0.01 \\ -0.06 & 0.43 & 0.02 & -0.05 \\ -0.06 & 0.02 & 0.43 & -0.05 \\ 0.01 & -0.05 & -0.05 & 0.05 \end{pmatrix} \end{aligned}$$

is executed for 10 time steps on an ideal simulator and the *ibmq_johannesburg 20-qubit* backend. The results are computed using the parameterized quantum circuit shown in Fig. 2. Notably, readout error-mitigation (Aleksandrowicz and et al. 2019; Dewes et al. 2012; Stamatopoulos et al. 2020) is used to obtain the final results run on real quantum hardware. Figure 3 depicts the results considering the fidelity between the trained and the target Gibbs states for each time step. It should be noted that the fidelity for the

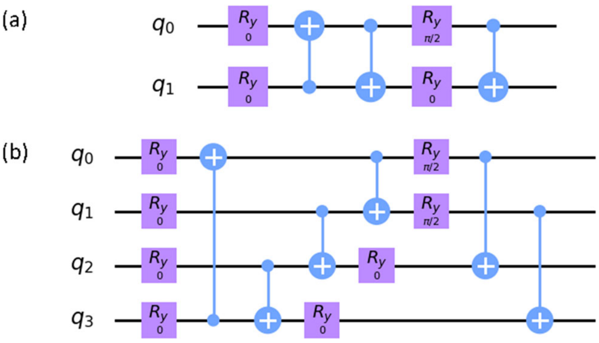

Fig. 2 The depicted circuits illustrate the initial trial state for the Gibbs state preparation of **a** ρ_1^{Gibbs} and **b** ρ_2^{Gibbs} using VarQITE

quantum backend evaluations employs state tomography. The plots illustrate that the method approximates the states, we are interested in reasonably well and that also the real quantum hardware achieves fidelity values over 0.99 and 0.96, respectively.

4.3 Generative learning

Now, the results from an illustrative example of a generative QBM model are presented. More explicitly, the QBM is trained to mimic the sampling statistics of a Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$, which is a state that exhibits non-local correlations. Numerical simulations show that the distribution can be trained with a fully visible QBM which is based on the following Hamiltonian:

$$H_\theta = \theta_0 ZZ + \theta_1 IZ + \theta_2 ZI.$$

We draw the initial values of the Hamiltonian parameters θ from a uniform distribution on $[-1, 1]$. The optimization runs on an ideal simulation of a quantum computer using AMSGrad (Reddi et al. 2018) with initial learning rate 0.1, maximum number of iterations 200, first momentum 0.7, and second momentum 0.99 as optimization routine. The Gibbs state preparation uses the initial trial state shown in Fig. 4 and 10 steps per state preparation.

The training is run 10 times using different randomly drawn initial parameters. The averaged values of the loss function as well as the distance between the target distribution $p^{\text{data}} = [0.5, 0., 0., 0.5]$ and the trained distribution p^{QBM} with respect to the ℓ_1 norm are illustrated over 50 optimization iterations in Fig. 5. The plot shows that loss and distance converge toward the same values for all sets of initial parameters. Likewise, the trained parameters θ converge to similar values. Furthermore, Fig. 6 illustrates the target probability distribution and for the best and worst of the trained distributions. The plot reveals that the model is able to train the respective distribution very well.

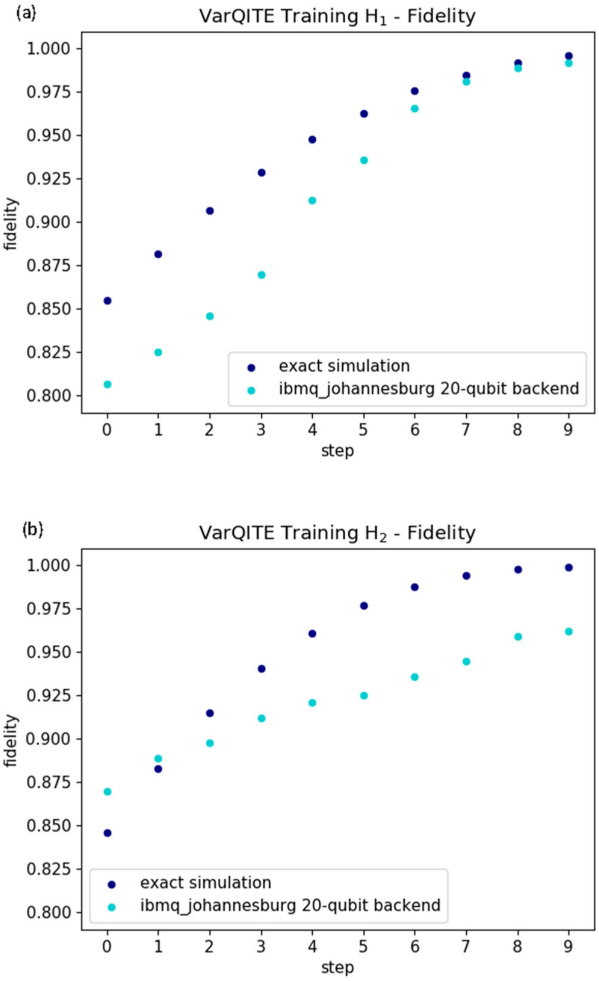

Fig. 3 Fidelity between trained and target Gibbs states with VarQITE for **a** ρ_1^{Gibbs} and **b** ρ_2^{Gibbs} trained with an ideal simulator and real quantum hardware, i.e., the *ibmq_johannesburg 20-qubit* backend. Each simulation uses 10 time steps

We verify the scalability of this method by training equivalent QBM models for 3 and 4 qubit Greenberg-Horne-Zeilinger (GHZ) states (Greenberger et al. 1989). The respective results are presented in Appendix D.

4.4 Discriminative learning

QBMs are not only applicable for generative but also for discriminative learning. We discuss the application to a classification task: the identification of fraudulent credit card transactions.

To enable discriminative learning with QBMs, we use the input data points x as bias for the Hamiltonian weights. More explicitly, the parameters of the Hamiltonian

$$H_\theta(x) = \sum_i f_i(\theta, x) h_i \tag{13}$$

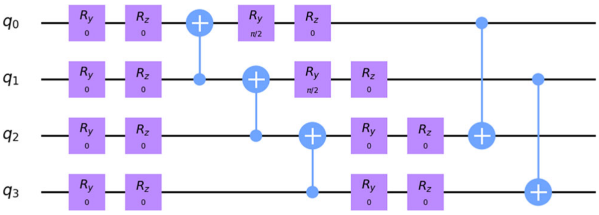

Fig. 4 We train a QBM to mimic the sampling behavior of a Bell state. The underlying Gibbs state preparation with VarQITE uses the illustrated parameterized quantum circuit to prepare the initial trial state. The first two qubits represent the target system and the last two qubits are ancillas needed to generate the maximally mixed state as starting state for the evolution

are given by a function $f_i(\theta, x)$ which maps θ and x to a scalar in $\mathbb{R}$. Now, the respective loss function reads:

$$L = -\sum_x p_x^{\text{data}} \sum_v p_{v|x}^{\text{data}} \log p_{v|x}^{\text{QBM}}$$

with

$$p_{v|x}^{\text{QBM}} = \text{Tr}\left[\Lambda_v \rho\,(x)_\omega^{\text{Gibbs}}\right],$$

where $\rho\,(x)_\omega^{\text{Gibbs}}$ denotes the approximate Gibbs state corresponding to $H_\theta\,(x)$. The model encodes the class labels in the measured output configuration of the visible qubits v of $\rho\,(x)_\omega^{\text{Gibbs}}$. Now, the aim of the training is to find Hamiltonian parameters θ such that, given a data sample x, the probability of sampling the correct output label from $\rho\,(x)_\omega^{\text{Gibbs}}$ is maximized.

The training is based on 500 artificially created credit card transactions (Altman 2019) with about 15% fraudulent instances. To avoid redundant state preparation, the training is run for all unique item instances in the dataset and the results are averaged according to the item's occurrence counts. The dataset includes the following features: location (ZIP code), time, amount, and Merchant Category Code (MCC) of the transactions. To facilitate the training, the features of the given dataset are discretized and normalized as follows. Using k-means clustering, each of the first three features are independently discretized to 3 reasonable bins. Furthermore, we consider MCCs $< 10,000$ and group them into 10 different categories. The discretization is discussed in more detail in Table 1. Furthermore, for each feature, we map the values x to $x' = \frac{x-\mu}{\sigma}$ with μ denoting the mean and σ denoting the standard deviation.

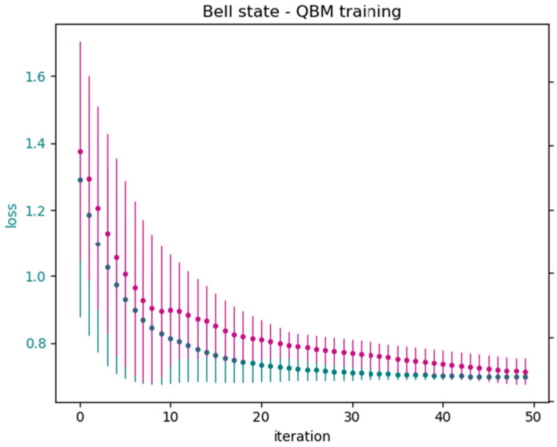

Fig. 5 The figure illustrates the training progress of a fully visible QBM model which aims to represent the measurement distribution of a Bell state. The green function corresponds to the loss and the pink function represents the distance between the trained and target distributions with respect to the ℓ_1 norm at each step of the iteration. Both measures are computed for 10 different random seeds. The points represent the mean and the error bars the standard deviation of the results

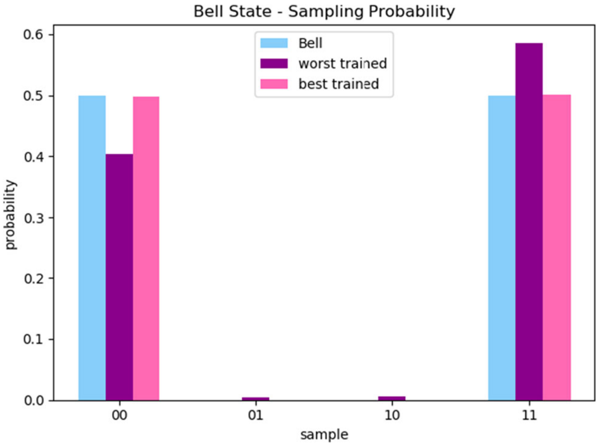

Fig. 6 The figure illustrates the sampling probability of the Bell state (blue), as well as the best (pink) and worst (purple) probability distributions achieved from 10 different random seeds

The complexity of this model demands a Hamiltonian that has sufficient representation capabilities. We investigate a diagonal Hamiltonian:

$$\begin{aligned} H_\theta^{(0)}\,(x) = \; & f_0^{(2)}\,(\theta, x)\,ZZ + f_1^{(2)}\,(\theta, x)\,ZI \\ & + f_2^{(2)}\,(\theta, x)\,IZ, \end{aligned} \tag{14}$$

a Hamiltonian with off-diagonal terms whose parameters are fixed to 0.1

$$\begin{aligned} H_\theta^{(1)}\,(x) = \; & f_0^{(1)}\,(\theta, x)\,ZZ + f_1^{(1)}\,(\theta, x)\,ZI \\ & + f_2^{(1)}\,(\theta, x)\,IZ + 0.1XI + 0.1IX, \end{aligned} \tag{15}$$

and a Hamiltonian with parameterized off-diagonals

$$\begin{aligned} H_\theta^{(2)}\,(x) = \; & f_0^{(2)}\,(\theta, x)\,ZZ + f_1^{(2)}\,(\theta, x)\,ZI \\ & + f_2^{(2)}\,(\theta, x)\,IZ + f_3^{(2)}\,(\theta, x)\,XI \\ & + f_4^{(2)}\,(\theta, x)\,IX, \end{aligned} \tag{16}$$

where $f_i^{(j)}\,(\theta, x) = \vec{\theta}_i^{(j)} \cdot \vec{x}$ corresponds to the dot product of the vector corresponding to the data item $\vec{x}$ and

Table 1 The table discusses the binning of a transaction fraud dataset which is used to train a discriminative QBM model

Feature	Condition	Value
Time	0AM–11AM	0
	11AM–6PM	1
	6PM–0AM	2
Amount	Amount $<$ \$50	0
	Amount in \$50–150	1
	Amount $>$ \$150	2
ZIP	East	0
	Central	1
	West	2

MCC refers to the merchant category code and *ZIP* to zone improvement plan. Notably, the given values are approximate

a parameter vector $\vec{\theta}_i^{(j)}$ of equal length. Additionally, the first and second qubits of all systems correspond to hidden and visible qubits, respectively.

The training uses a conjugate gradient optimization routine with a maximum iteration number of 100.

The initial values for the Hamiltonian parameters are drawn from a uniform distribution on $[-1, 1]$.

Given a test dataset consisting of 250 instances, with about 10% fraudulent transactions, the Gibbs states, corresponding to the unique items of the test data, are approximated using VarQITE with the trained parameters θ and the trial state shown in Fig. 7. To predict the labels of the data instances, we sample from the states $\rho_{\omega}^{\text{Gibbs}}$ and choose the label with the highest sampling probability. These results are, then, used to evaluate the accuracy (Acc), recall (Rec), precision (Pre), and F_1 score. It should be noted that we choose a relatively simple quantum circuit to keep the simulation cost small. However, it can be expected that a more complex parameterized quantum circuit might lead to further improvement in the training results.

The resulting values are compared to a set of standard classifiers defined in a *scikit-learn* (Pedregosa et al. 2011) classifier comparison tutorial (Classifier Comparison 2020) as well as a classifier consisting of a restricted Boltzmann machine and a logistic regression (RBM) (Restricted Boltzmann Machine for classification 2020); see Table 2. The respective classifiers are used with the hyperparameters defined in the tutorials. Neither the linear SVM nor the RBM classifies any test data item as fraudulent, and thus, the classifier sets precision and recall score to 0. The comparison reveals that all variational QBMs perform similarly well to the classical classifiers considering accuracy and are competitive regarding precision. In terms of recall and F_1 score, the VarQBM models without off-diagonal elements and with fixed off-diagonal elements perform well. Notably, the addition of fixed-off diagonal terms in the Hamiltonian seems not to improve the model performance compared to a Hamiltonian without these terms in this case. Finally, the VarQBM with trained off-diagonal elements achieves the best results in terms of recall and F_1 score.

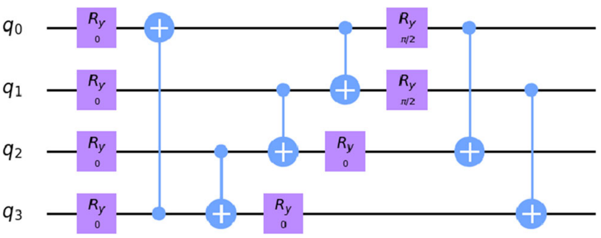

Fig. 7 Given a transaction instance, the measurement output of the QBM labels it as being either fraudulent or valid. The underlying Gibbs state preparation with VarQITE uses the illustrated parameterized quantum circuit as initial trial state. The first qubit is the visible unit that determines the QBM output, the second qubit represents the hidden unit, and the last two qubits are ancillas needed to generate the maximally mixed state as starting state for the evolution

Table 2 This table presents performance measures for scikit-learn standard classifiers, as well as the trained variational QBM models

Model	Acc	Rec	Pre	F_1
Nearest Neighbors	0.94	0.54	0.72	0.62
Linear SVM	0.90	0	0	0
RBF SVM	0.94	0.42	0.83	0.56
Gaussian Process	0.94	0.46	0.85	0.60
Gaussian Naive Bayes	0.91	0.42	0.56	0.48
Decision Tree	0.94	0.42	0.83	0.56
Random Forrest	0.93	0.29	1.00	0.45
Multi-layer Perceptron	0.94	0.38	0.9	0.53
AdaBoost	0.94	0.54	0.81	0.65
QDA	0.92	0.46	0.61	0.52
RBM	0.90	0	0	0
VarQBM$^{(0)}$	**0.93**	**0.50**	**0.67**	**0.57**
VarQBM$^{(1)}$	**0.93**	**0.50**	**0.67**	**0.57**
VarQBM$^{(2)}$	**0.95**	**0.58**	**0.88**	**0.70**

The Nearest Neighbors classifier uses a 3 nearest neighbors vote. The Linear and RBF Support Vector Machine (*SVM*) are based on a linear and radial kernel, respectively. The Linear SVM uses a regularization term of 0.25 and for the RBF SVM the kernel coefficient is set to 2. The maximum depth of the Decision Tree as well as the Random Forrest is set to 5. Furthermore, the Random Forrest classifier uses 10 trees and uses 1 feature to search for the best spit. The Multi-layer Perceptron uses ℓ_2 regularization with coefficient 1 and a maximum iteration number of 1000. *QDA* refers to Quadratic Discriminant Analysis. It should be noted that the remaining classifier properties are default settings

5 Conclusion and outlook

This work presents the application of McLachlan's variational principle to facilitate VarQBM, a variational QBM algorithm, that is compatible with generic Hamiltonians and can be trained using analytic gradients of the actual loss function even if some of the qubits are hidden. Suppose a sufficiently powerful variational trial state, the presented scheme, is not only compatible with local but also long-range correlations and for arbitrary system temperatures.

We outline the practical steps for utilizing VarQITE for Gibbs state preparation and verify that it can train states which are reasonably close to the target using simulation as well as real quantum hardware. Moreover, applications to generative learning and classification are discussed and illustrated with further numerical results. The presented model offers a versatile framework which facilitates the representation of complex structures with quantum circuits.

An interesting question for future research to conduct is a thorough analysis of trial state Ansätze which would help to improve our understanding of the model's representation capabilities and enable more informed choices of Ansatz states. Furthermore, QBMs are not limited to the presented applications. They could also be utilized to train models for data from experiments with quantum systems. This is a problem that has recently gained interest; see e.g., Gentile et al. (2020). Additionally, they might be employed for combinatorial optimization. Classical BMs have been investigated in this context (Spieksma 1995) and developing and analyzing quantum algorithms for combinatorial optimization is an active area of research (Farhi et al. 2014; Barkoutsos et al. 2020). All in all, there are many possible applications which still have to be explored.

Appendix 1: Evaluation of A, C, and their gradients

The elements of the matrix A and the vector C (see Eq. 6) are of the following form:

$$\mathrm{Re}\left(e^{i\alpha}\mathrm{Tr}\left[U^{\dagger}V\rho_{\mathrm{in}}\right]\right) \tag{17}$$

with $\mathrm{Re}\,(\cdot)$ denoting the real part and $\rho_{\mathrm{in}} = |\psi_{\mathrm{in}}\rangle\langle\psi_{\mathrm{in}}|$. As discussed in McArdle et al. (2019), Yuan et al. (2019), and Somma et al. (2002), such terms can be computed by sampling the expectation value of an observable Z with respect to the quantum circuit shown in Fig. 8.

Notably, the phase $e^{i\alpha}$ in the first qubit is needed to include phases which may occur from gate derivatives. In our case, α needs to be set to 0 respectively $\pi/2$ when computing the terms of A or C. More precisely, the first qubit is initialized by an H gate for A and H followed by an S gate for C. These phases come from the fact that the trial states, used in this work, are constructed via Pauli rotations, i.e., $U\,(\omega) = R_{\sigma_l}\,(\omega)$ with $\sigma_l \in \{X, Y, Z\}$, which leads to

$$\frac{\partial U\,(\omega)}{\partial\omega} = -\frac{i}{2}\sigma_l R_{\sigma_l}\,(\omega)\,. \tag{18}$$

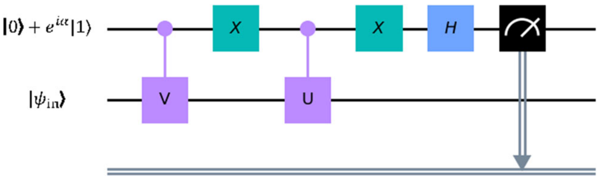

Fig. 8 Quantum circuit to evaluate $\mathrm{Re}\left(e^{i\alpha}\mathrm{Tr}\left[U^{\dagger}V\rho_{\mathrm{in}}\right]\right)$ with $\rho_{\mathrm{in}} = |\psi\rangle\langle\psi_{\mathrm{in}}|$

Furthermore, this method can be applied for the evaluation of $\partial A/\partial\theta$ and $\partial C/\partial\theta$, i.e., the respective terms can be written in the form of Eq. 17. More precisely,

$$\begin{aligned}\partial_{\theta_i} A_{p,q}\,(\tau) = \sum_s \frac{\partial\omega_s(\tau)}{\partial\theta_i}\mathrm{Re}\left(\mathrm{Tr}\left[\left(\frac{\partial^2 V^{\dagger}(\omega(\tau))}{\partial\omega_p(\tau)\partial\omega_s(\tau)}\frac{\partial V(\omega(\tau))}{\partial\omega(\tau)_q}\right.\right.\right.\\ \left.\left.\left.+\frac{\partial V^{\dagger}(\omega(\tau))}{\partial\omega(\tau)_p}\frac{\partial^2 V(\omega(\tau))}{\partial\omega_q(\tau)\partial\omega_s(\tau)}\right)\rho_{\mathrm{in}}\right]\right)\end{aligned}$$

and

$$\begin{aligned}&\partial_{\theta_j} C_p =\\ &-\mathrm{Re}\left(\mathrm{Tr}\left[\frac{\partial V^{\dagger}(\omega(\tau))}{\partial\omega(\tau)_p}h_j V\,(\omega\,(\tau))\,\rho_{\mathrm{in}}\right]\right)\\ &-\sum_{i,s}\theta_i\frac{\partial\omega_s(\tau)}{\partial\theta_j}\mathrm{Re}\left(\mathrm{Tr}\left[\left(\frac{\partial V^{\dagger}(\omega(\tau))}{\partial\omega_p(\tau)}h_i\frac{\partial V(\omega(\tau))}{\partial\omega_s(\tau)}\right.\right.\right.\\ &\left.\left.\left.+\;\frac{\partial^2 V^{\dagger}(\omega(\tau))}{\partial\omega_p(\tau)\partial\omega_s(\tau)}h_i V\,(\omega\,(\tau))\right)\rho_{\mathrm{in}}\right]\right).\end{aligned}$$

Hereby, α must be set to $\pi/2$ for all terms in $\partial A/\partial\theta$ and the first term in $\partial C/\partial\theta$ and 0 for the remaining terms in $\partial C/\partial\theta$. This is achieved with the same gates as mentioned before.

Appendix 2: Complexity analysis

To compute the gradient $\partial L/\partial\theta$ of the loss function, given in Eq. 7, we could use either a numerical finite differences method (Kardestuncer 1975), or the analytic, automatic differentiation approach that is presented in this paper. In the following, we discuss the number of circuits that have to be evaluated for those gradient implementations for a trial state with q parameters, an n-qubit Hamiltonian with p parameters, and VarQITE for Gibbs state preparation using t steps. In near-term quantum devices, the run of every circuit requires compilation, queuing for a position to be run on quantum hardware and, preferably, error mitigation/correction. Thus, the number of circuits strongly impacts the execution time and resources.

Table 3 Comparing the number of circuits needed to train a QBM with VarQITE using either finite differences or automatic differentiation

Method	Number circuits
Finite Diff	$\Theta\left(tqp(q+p)\right)$
Automatic Diff	$\Theta\left(tq^2(q+p)\right)$

The number of Hamiltonian parameters is p, the number of trial state parameters is q, and the number of time steps during the Gibbs state preparation is t

The number of circuits that need to be evaluated for Gibbs state preparation with VarQITE are $\Theta\left(tq^2\right)$ and $\Theta\left(tqp\right)$ for A and C, respectively. Therefore, the overall number of circuits is $\Theta\left(tq(q+p)\right)$. Now, computing the gradient with forward finite differences reads:

$$\frac{\partial L}{\partial \theta} \approx \frac{L\left(\theta+\epsilon\right)-L\left(\theta\right)}{\epsilon},$$

for $0<\epsilon \ll 1$. For this purpose, VarQITE must be run once with θ and p times with an ϵ-shift which leads to a total number of $\Theta\left(tpq(q+p)\right)$ circuits.

The automatic differentiation gradient, given in Eq. 9, corresponds to:

$$\frac{\partial L}{\partial \theta} = -\sum_{v}\sum_{k=0}^{q-1}\frac{p_v^{\text{data}}}{\langle\Lambda_v\rangle}\frac{\partial\langle\Lambda_v\rangle}{\partial\omega_k}\frac{\partial\omega_k}{\partial\theta}$$

with $\langle\ldots\rangle = \text{Tr}\left[\rho_{\omega}^{Gibbs}\ldots\right]$. VarQITE needs to be run once to prepare $\rho_{\omega}^{\text{Gibbs}}$. Furthermore, the evaluation of $\partial\omega_k/\partial\theta$ requires that $\partial A/\partial\theta$ and $\partial C/\partial\theta$ are computed for every step of the Gibbs state preparation. This leads to $\Theta\left(tq^2(q+p)\right)$ circuits. The resulting overall complexity of the number of circuits is $\Theta\left(tq^2(q+p)\right)$. The results are summarized in Table 3. Automatic differentiation is more efficient than finite differences if $q<p$. For $q>p$, on the other hand, focusing mainly on computational complexity, one should rather use finite differences. Considering, e.g., a k-local Ising model that corresponds to a Hamiltonian with $\mathcal{O}\left(n^k\right)$ parameters. Suppose that we can find a reasonable variational n-qubit trial state with $\mathcal{O}\left(n\right)$ layers of parameterized and entangling gates, which results in $q=\mathcal{O}\left(n^2\right)$ parameters, then automatic differentiation would outperform finite differences for $k>2$.

Appendix 3: Numerical Gibbs state preparation

In order to verify that VarQITE enables approximate Gibbs state preparation, we conduct exact simulation experiments for Hamiltonians with $n>2$ qubits. The states are prepared for the following Hamiltonians:

$$\begin{aligned} H_3 &= 2.\cdot ZZI + 1.\cdot IZZ - 0.5\cdot IZI \\ H_4 &= 2.\cdot ZZII + 1.\cdot IZZI - 0.5\cdot IZIZ \\ H_5 &= 2.\cdot ZZIII + 1.\cdot IZZII - 0.5\cdot IZIIZ, \end{aligned}$$

temperature $k_BT=1$ and a discretization of 10 time steps. All Gibbs states are prepared using a shallow, depth 2 Ansatz that consists of parameterized R_Y and R_Z gates as well as CX gates. Figure 9 illustrates the respective quantum circuit for H_3, whereby the given parameters prepare the initial state. The states for H_4 and H_5 are prepared equivalently.

The effectiveness of VarQITE is confirmed with a plot (see Fig. 10), of the fidelities between the trained and the target Gibbs states throughout the simulated time evolution. It should be noted that all final states approximate the targets with fidelity >0.936.

Appendix 4: Generative QBM training for larger systems

To investigate the trainability of QBMs for larger systems, we extend the learning setting discussed in Section 4.3 to $n=\{3,4\}$ qubits. More explicitly, given a 3- and 4-qubit Greenberger-Horne-Zeilinger (GHZ) state, a QBM model is trained to represent the respective sampling probabilities of these states:

$$\begin{aligned} p_3^{\text{data}} &= [0.5, 0., 0., 0., 0., 0., 0., 0.5] \\ p_4^{\text{data}} &= [0.5, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., \\ &\quad 0., 0., 0.5] \end{aligned}$$

using Hamiltonians of the form

$$H_{\theta} = \sum_{i=0}^{n-1}\theta_i Z_i + \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\theta_{ij}Z_iZ_j,$$

where Z_i acts on the i^{th} qubit. The temperature is again set to $k_BT=1$ and the time is discretized into 10 steps. Optimizer and initial states are chosen equivalently to the ones from Section 4.3. Notably, the number of qubits in the Ansatz state shown in Fig. 4 needs to match to the number of qubits in the respective Hamiltonian. Figure 11 illustrates the progress of the loss function as well as the ℓ_1 norm between the trained and the target distributions for 30 training steps. Both measures show a good convergence behavior and thus indicate that the method scales well.

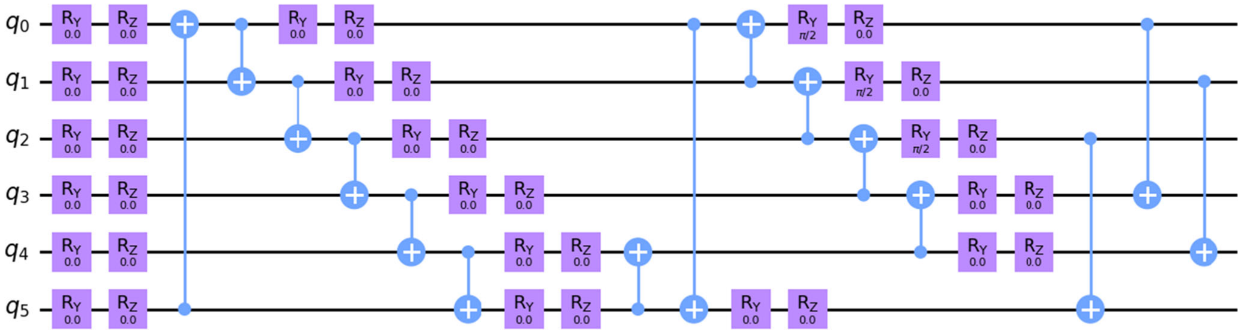

Fig. 9 This quantum circuit is used as Ansatz for the Gibbs state preparation of H_3. The given parameter set prepares the initial state, i.e., tracing out q_3 to q_5 results in a maximally mixed state

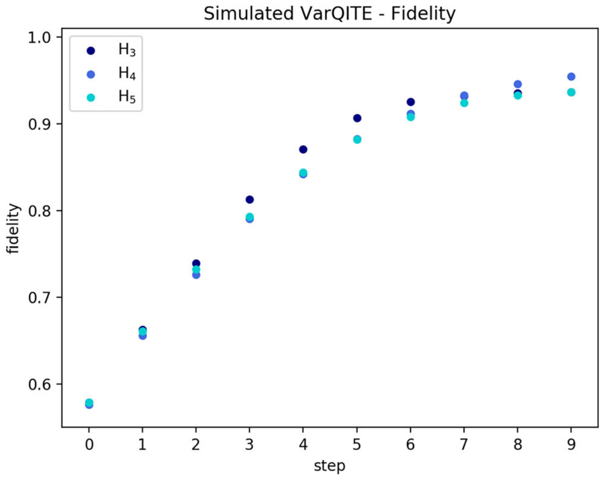

Fig. 10 This figure illustrates the fidelity between the state that is prepared using VarQITE and the target Gibbs state preparation for H_3, H_4, and H_5 during each step of the simulated imaginary time evolution

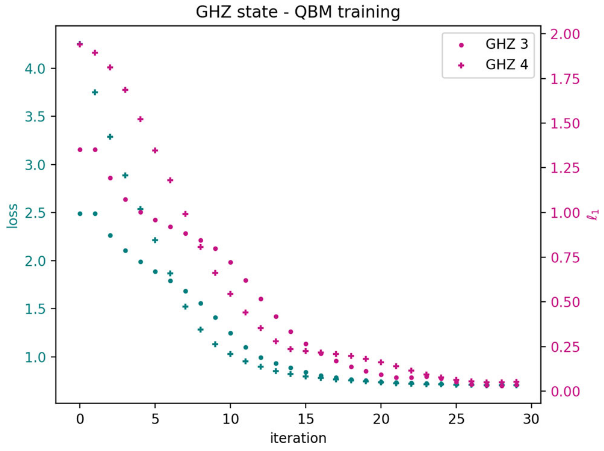

Fig. 11 The figure illustrates the training progress of a fully visible QBM model which aims to represent the measurement distribution of a 3- and a 4-qubit GHZ state. The green and pink dots (crosses) correspond to the loss and the distance, respectively, between the trained and target distributions with respect to the ℓ_1 norm at each step of the iteration for the 3 (4)qubit GHZ state

Acknowledgments We would like to thank Erik Altman for making the synthetic credit card transaction dataset available to us. Moreover, we are grateful to Pauline Ollitrault, Guglielmo Mazzola, and Mario Motta for sharing their knowledge and engaging in helpful discussion. Furthermore, we thank Julien Gacon for his help with the implementation of the algorithm and all of the IBM Quantum team for its constant support.

Also, we acknowledge the support of the National Centre of Competence in Research *Quantum Science and Technology* (QSIT).

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at https://www.ibm.com/legal/copytrade.

Author contributions All authors researched, collated, and wrote this paper.

Funding Open Access funding provided by ETH Zurich

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Classifier Comparison (2020). https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

IBM Q Experience (2020). https://quantumexperience.ng.bluemix.net/qx/experience

Restricted Boltzmann Machine for classification (2020). https://scikit-learn.org/stable/auto_examples/neural_networks/plot_rbm_logistic_classification.html

Abrams DS, Lloyd S (1999) Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. Phys. Rev. Lett. 83:5162

Ackley DH, Hinton GE, Sejnowski TJ (1985) A learning algorithm for Boltzmann machines. Cogn Sci 9(1):147–169

Aleksandrowicz G, et al. (2019) Qiskit: An open-source framework for quantum computing

Alet F, Damle K, Pujari S (2016) Sign-problem-free Monte Carlo simulation of certain frustrated quantum magnets. Phys. Rev. Lett. 117:197203

Altman E (2019) Synthesizing credit card transactions. arXiv:1910.03033

Amin MH, Andriyash E, Rolfe J, Kulchytskyy B, Melko R (2018) Quantum Boltzmann machine. Phys. Rev. X 8:021050

Anschütz ER, Cao Y (2019) Realizing Quantum Boltzmann Machines Through Eigenstate Thermalization. arXiv:1903.01359

Anshu A, Arunachalam S, Kuwahara T, Soleimanifar M (2020) Sample-efficient learning of quantum many-body systems. arXiv:2004.07266

Assis CAS, Pereira ACM, Carrano EG, Ramos R, Dias W (2018) Restricted Boltzmann machines for the prediction of trends in financial time series. In: International Joint Conference on Neural Networks

Auer J, Krotscheck E, Chin SA (2001) A fourth-order real-space algorithm for solving local Schrödinger equations. The Journal of Chemical Physics 115(15):6841–6846

Barkoutsos P, Nannicini G, Robert A, Tavernelli I, Woerner S (2020) Improving variational quantum optimization using CVaR. Quantum 4:256

Besag J (1975) Statistical analysis of non-lattice data. Journal of the Royal Statistical Society. Series D (The Statistician) 24(3):179–195

Boltzmann L (1877) Über die Natur der Gasmoleküle. In: Wissenschaftliche Abhandlungen, Vol. I, II, and III

Brandão FGSL, Kastoryano MJ (2019) Finite correlation length implies efficient preparation of quantum thermal states. Commun Math Phys 365(1):1–16

Brassard G, Hoyer P, Mosca M, Tapp A (2002) Quantum amplitude amplification and estimation. Contemp Math 305:53–74

Bravyi S, DiVincenzo D, Oliveira R, Terhal B (2006) The complexity of Stoquastic local Hamiltonian problems. Quantum Information and Computation, 8

Carleo G, Nomura Y, Imada M (2018) Constructing exact representations of quantum many-body systems with deep neural networks. Nat Commun 9:1–11

Carleo G, Troyer M (2017) Solving the quantum many-body problem with artificial neural networks. Science 355:602–606

Carreira-Perpinan M, Hinton G (2005) On contrastive divergence learning. Artificial Intelligence and Statistics

Chowdhury A, Low GH, Wiebe N (2020) A Variational Quantum Algorithm for Preparing Quantum Gibbs States. arXiv:2002.00055

Dallaire-Demers P-L, Killoran N (2018) Quantum generative adversarial networks. Phys. Rev. A 98:012324

Dembo RS, Steihaug T (1983) Truncated-Newton algorithms for large-scale unconstrained optimization. Math Program 26(2):190–212

Dewes A, Ong FR, Schmitt V, Lauro R, Boulant N, Bertet P, Vion D, Esteve D (2012) Characterization of a two-transmon processor with individual single-shot Qubit readout. Phys. Rev. Lett. 108:057002

Du K-L, Swamy MNS (2019) Boltzmann machines. In: Neural networks and statistical learning. Springer London, London

Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. arXiv:1411.4028

Farhi E, Neven H (2018) Classification with quantum neural networks on near term processors. arXiv:1802.06002

Fischer A (2015) Training restricted Boltzmann machines. KI - Künstliche Intelligenz 29(4):25–39

Fischer A, Igel C (2012) An Introduction to Restricted Boltzmann Machines. In: Alvarez L, Mejail M, Gomez L, Jacobo J (eds) Progress in pattern recognition, image analysis, computer vision, and applications. Springer Berlin Heidelberg

Fischer A, Igel C (2012) An introduction to restricted Boltzmann machines. In: Alvarez L, Mejail M, Gomez L, Jacobo J (eds) Progress in pattern recognition, image analysis, computer vision, and applications. Springer Berlin Heidelberg, Berlin, Heidelberg

Gentile AA, Flynn B, Knauer S, Wiebe N, Paesani S, Granade C, Rarity J, Santagati R, Laing A (2020) Learning models of quantum systems from experiments. arXiv:2002.06169

Gibbs JW (1902) Elementary principles in statistical mechanics. Cambridge University Press, Cambridge

Gibbs JW (2010) Elementary principles in statistical mechanics: Developed with especial reference to the rational foundation of thermodynamics. Cambridge Library Collection - Mathematics. Cambridge University Press, Cambridge

Golden S (1965) Lower bounds for the helmholtz function. Phys. Rev. 137:B1127

Greenberger DM, Horne MA, Zeilinger A (1989) Going beyond bell's theorem. Springer Netherlands, Dordrecht

Gupta N, Roy AK, Deb BM (2002) One-dimensional multiple-well oscillators: A time-dependent quantum mechanical approach. Pramana 59(4):575–583

Hangleiter D, Roth I, Nagaj D, Eisert J (2020) Easing the Monte Carlo sign problem. Science Advances 6(33):eabb8341. arXiv:1906.02309

Hansen PC (2000) The L-Curve and its use in the numerical treatment of inverse problems. WIT Press, Southampton

Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14:1771–1800

Hinton GE (2012) A practical guide to training restricted Boltzmann machines. In: Montavon G, Orr GB, Müller K-R (eds) Neural networks: Tricks of the trade: Second edition. Springer Berlin Heidelberg, Berlin, Heidelberg

Hrasko R, Pacheco AGC, Krohling RA (2015) Time series prediction using restricted Boltzmann machines and backpropagation. 3rd International Conference on Information Technology and Quantitative Management 55:990–999

Ising E (1925) Beitrag zur Theorie des Ferromagnetismus. Zeitschrift für Physik 31(1):253–258

Kappen H (2020) Learning quantum models from quantum or classical data. Journal of Physics A: Mathematical and Theoretical 53(21). arXiv:1803.11278

Kardestuncer H (1975) Finite differences. In: Discrete Mechanics A Unified Approach. Springer Vienna

Kastoryano MJ, Brandão FGSL (2016) Quantum Gibbs samplers: The commuting case. Commun Math Phys 344(3):915–957

Khalkhali M, Marcolli M (2008) An invitation to noncommutative geometry. WORLD SCIENTIFIC, Singapore

Kieferová M, Wiebe N (2017) Tomography and generative training with quantum Boltzmann machines. Phys. Rev. A 96:062327

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y (eds) 3rd International Conference on Learning Representations

Koczor B, Benjamin S (2019) Quantum natural gradient generalised to non-unitary circuits. arXiv:1912.08660

Li Z-X, Jiang Y-F, Yao H (2015) Solving the fermion sign problem in quantum Monte Carlo simulations by Majorana representation. Phys. Rev. B 91:241117. https://doi.org/10.1103/PhysRevB.91.241117

Li Z-X, Jiang Y-F, Yao H (2016) Majorana-time-reversal symmetries: A fundamental principle for sign-problem-free quantum monte carlo simulations. Phys Rev Lett 117:267002

Liu B, Webb GI (2010) Generative and discriminative learning. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning. Springer US, Boston, MA

Liu F, Liu B, Sun C, Liu M, Wang X (2013) Deep learning approaches for link prediction in social network services. Springer Berlin Heidelberg, Berlin

Magnus W (1954) On the exponential solution of differential equations for a linear operator. Commun Pur Appl Math 7(4):649–673

Marrero CO, Kieferová M, Wiebe N (2020) Entanglement Induced Barren Plateaus. arXiv:2010.15968

Matsui T (1998) Quantum statistical mechanics and Feller semigroup. In: Quantum probability communications

McArdle S, Jones T, Endo S, Li Y, Benjamin SC, Yuan X (2019) Variational ansatz-based quantum simulation of imaginary time evolution. npj Quantum Information 5(1):1–6

McLachlan AD (1964) A variational solution of the time-dependent Schrödinger equation. Mol Phys 8(1):39–44

Melko RG, Carleo G, Carrasquilla J, Cirac JI (2019) Restricted Boltzmann machines in quantum physics. Nat Phys 15(9):887–892

Mitarai K, Negoro M, Kitagawa M, Fujii K (2018) Quantum circuit learning. Phys. Rev. A 98:032309

Mohamed A-R, Hinton G (2010) Phone recognition using Restricted Boltzmann machines. IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings

Montúfar G (2018) Restricted Boltzmann Machines: Introduction and Review. In: Ay N, Gibilisco P, Matúš F (eds) Information geometry and its applications. Springer International Publishing

Motta M, et al. (2020) Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. Nat Phys 16(2):205–210

Murphy KP (2012) Machine learning: a probabilistic perspective. MIT Press, Cambridge

Nielsen MA, Chuang IL (2010) Quantum computation and quantum information. Cambridge University Press, Cambridge

Noh H, You T, Mun J, Han B (2017) Regularizing deep neural networks by noise: Its interpretation and optimization. In: NIPS

Nomura Y, Darmawan A, Yamaji Y, Imada M (2017) Restricted-Boltzmann-machine learning for solving strongly correlated quantum systems. Phys Rev B 96:205152

Okunishi K, Harada K (2014) Symmetry-protected topological order and negative-sign problem for SO(n) bilinear-biquadratic chains. Phys. Rev. B 89:134422

Ortiz G, Gubernatis J, Knill E, Laflamme R (2001) Quantum algorithms for fermionic simulations. Phys Rev A 64:022319

Pauli W (1927) Über Gasentartung und Paramagnetismus. Zeitschrift für Physik 41(2):81–102

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830

Peierls R (1936) On Ising's model of ferromagnetism. Math Proc Camb Philos Soc 32(3):477–481

Poulin D, Wocjan P (2009) Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. Phys Rev Lett 103:220502

Reddi SJ, Kale S, Kumar S (2018) On the convergence of adam and beyond. In: International conference on learning representations

Roux N, Bengio Y (2010) Deep belief networks are compact universal approximators. Neural computation 22:2192–2207

Schuld M, Bergholm V, Gogolin C, Izaac J, Killoran N (2019) Evaluating analytic gradients on quantum hardware. Phys Rev A 99:032331

Somma R, Ortiz G, Gubernatis JE, Knill E, Laflamme R (2002) Simulating physical phenomena by quantum networks. Phys. Rev. A 65:042323

Spieksma FCR (1995) Boltzmann machines. In: Braspenning PJ, Thuijsman F, Weijters AJMM (eds) Artificial neural networks: an introduction to ANN theory and practice. Springer Berlin Heidelberg

Stamatopoulos N, Egger DJ, Sun Y, Zoufal C, Iten R, Shen N, Woerner S (2020) Option pricing using quantum computers. Quantum 4(291). arXiv:1905. 02666

Sutskever I, Tieleman T (2010) On the convergence properties of contrastive divergence. Proceedings of the thirteenth international conference on artificial intelligence and statistics 9:789–795

Temme K, Osborne TJ, Vollbrecht KGH, Poulin D, Verstraete F (2011) Quantum metropolis sampling. Nature 471:87–90

Thompson CJ (1965) Inequality with applications in statistical mechanics. J Math Phys 6(11):1812–1813

Tibshirani R (2011) Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73(3):273–282

Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. Proceedings of the 25th International Conference on Machine Learning

Tikhonov AN, Goncharsky AV, Stepanov VV, Yagola AG (1995) Numerical methods for the solution of ill-posed problems. Mathematics and Its Applications. Springer, Berlin

Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R, Carleo G (2018) Neural-network quantum state tomography. Nat Phys 14(5):447–450

Troyer M, Wiese U-J (2005) Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. Phys Rev Lett 94:170201

Tubiana J, Cocco S, Monasson R (2019) Learning compositional representations of interacting systems with restricted Boltzmann machines: Comparative study of lattice proteins. Neural Comput 31:1671–1717

Wiebe N, Wossnig L (2019) Generative training of quantum Boltzmann machines with hidden units. arXiv:1905.09902

Wu J, Hsieh TH (2019) Variational thermal quantum simulation via Thermofield double states. Phys Rev Lett 123:220502

Younes L (1996) Synchronous Boltzmann machines can be universal approximators. Appl Math Lett 9(3):109–113

Yuan X, Endo S, Zhao Q, Benjamin S, Li Y (2019) Theory of variational quantum simulation. Quantum 3, 191:191

Yung M-H, Aspuru-Guzik A (2012) A quantum–quantum Metropolis algorithm. Proceedings of the National Academy of Sciences 109(3):754–759

Zoufal C, Lucchi A, Woerner S (2019) Quantum generative adversarial networks for learning and loading random distributions. npj Quantum Information 5(1):1–9

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