

AMSI VACATION RESEARCH SCHOLARSHIPS 2020–21

Get a Thirst for Research this Summer



Quantum Boltzmann Machines

Saleh Naghdi

Supervised by Charles Hill & Thomas Quella

The University of Melbourne

Vacation Research Scholarships are funded jointly by the Department of Education, Skills and Employment
and the Australian Mathematical Sciences Institute.

Contents

1	Abstract	1
2	Introduction	2
3	Boltzmann Machines	3
3.1	Classical Boltzmann Machines	3
3.2	Quantum Boltzmann Machines	4
4	Model Parameters	5
4.1	Bias Field	5
4.2	Weight	7
4.3	Transverse Field	7
5	Implementation	8
5.1	Gibbs State Preparation	8
5.2	Gradient Descent	11
5.3	Results	11
6	Discussion	11
7	Conclusion	13
8	Acknowledgements	14
9	Statement of Authorship	14
10	Code Availability	14

1 Abstract

The ever-growing field of machine learning has recently reckoned with the potential of quantum computing to make strides in computational speedup. One such model is the Quantum Boltzmann Machine (QBM) which is based on the classical analogue of the Classical Boltzmann Machine. We explore the structure of the QBM as a probabilistic graphical model inspired by the physical Ising model, a framework which we use to intuit the effect of the model parameters on training quality. As a proof of concept, we implement a simple QBM on a quantum computer simulator and experiment its capacity for generative learning – one important task in machine learning.

2 Introduction

Many problems in machine learning, and particularly generative learning, reduce to the goal of approximating some true, but inaccessible, probability distribution $q(x)$ from which only a sampling set D_n may be obtained. It is with this aim that any potential ML framework requires two objects (Watanabe 2009):

- A model, parametrised by model parameters $\vec{\theta}$, which produces a probability distribution $p(x) \equiv p(x|\vec{\theta})$
- An error function $\mathcal{L}(p, q)$ quantifying how closely $p(x)$ fits $q(x)$

Given these, Figure 1 outlines what a generic ML procedure would look like. Of key importance is the distinction that the independent variable is the model parameters $\vec{\theta}$ whose effect we observe on the dependent variable, the error. Training a model may then be formulated as an optimisation problem, in which the ideal set of model parameters θ_{best} is found satisfying $\mathcal{L}(p(x|\vec{\theta}_{best}), q) = \min_{\vec{\theta}} \mathcal{L}(p(x|\vec{\theta}), q)$.

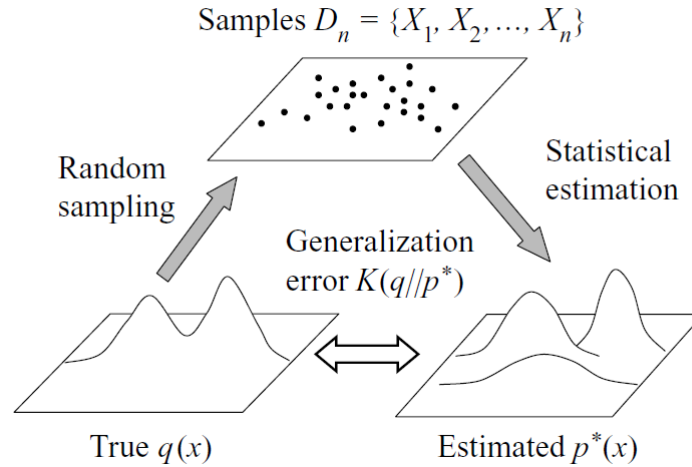


Figure 1: General procedure of statistical machine learning (Watanabe 2009)

The gamut of machine learning models spans widely, so much so that recently some classical models have been adapted to a quantum framework (Biamonte et al. 2017). The Quantum Boltzmann Machine (QBM) is one such model, whose model parameters are inspired by the Transverse Field Ising Model (Amin et al. 2018). Our research aims to evaluate the efficacy of the QBM, both theoretically, and practically as model to be implemented on a quantum computer. It is highly recommended for the reader to first read sections 1.2-3 of Nielsen and Chuang 2010 to understand quantum computing formulations mentioned in future sections.

We investigate how exactly the model parameters influence the capacity of the QBM for fitting a given $q(x)$, or mathematically, determine the topology of the error landscape which is critical for training. The results indicated that the topology, in just low-dimensional space is prone to the occurrence of flat regions which may interrupt conventional methods of training such as gradient descent.

Moreover, we set out to run a QBM on IBM's superconducting quantum computing framework Qiskit and test how it manages in generative learning of a simple probability distribution. While somewhat successful, it

was found that the precision of learning was stymied by a few technical caveats.

3 Boltzmann Machines

3.1 Classical Boltzmann Machines

In its most generic form, a Boltzmann Machine (BM) is an undirected graph $G = (V, E)$ such that each vertex $v_i \in V$ represents a random binary variable X_i taking values in $\Lambda = \{0, 1\}$. In this way, a BM is recognised as a probabilistic graphical model exhibiting a total of $2^{|V|}$ states (Fischer and Igel 2014).

More specifically, the graphical model describing BMs may be classified as a Markov Random Field (MRF), whose discrete probability distribution describing the state $\mathbf{x} \in \Lambda^{|V|}$ satisfies the following form (Fischer and Igel 2014):

$$p(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})}. \quad (1)$$

Here, $Z = \sum_{\mathbf{x} \in \Lambda^{|V|}} e^{-E(\mathbf{x})}$ is the normalisation constant and importantly, $E(x)$ is the *energy function*. Due to its correspondence to the same distribution that occurs in statistical mechanics, equation (1) is often called the *Gibbs Distribution*, a convention which will be adopted from here onward.

Bar satisfying the conditions mentioned in Fischer and Igel 2014, the energy function $E(\mathbf{x})$ may be defined liberally, but for the case of the BM it is strictly based on the energy function introduced by the Ising Model.

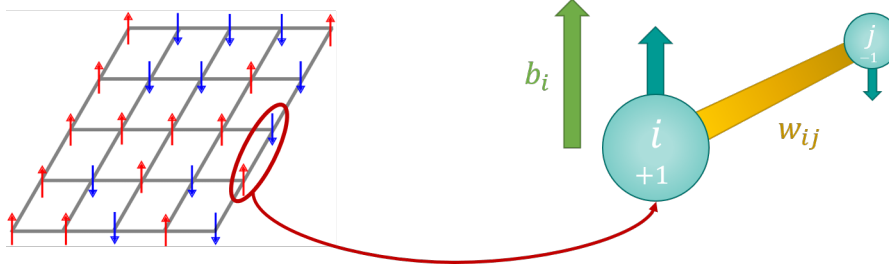


Figure 2: The classical Ising model

The Ising Model describes the energy associated with spin- $\frac{1}{2}$ configurations of a lattice of interacting dipoles subject to a magnetic field. Already, this physical system is similar in structure to the MRF, with the dipoles as vertices and spin as the random variable ($0 := \uparrow$ and $1 := \downarrow$). For a given spin configuration state \mathbf{x} , the Ising Model ascribes the following energy

$$\begin{aligned} E(\mathbf{x} | \mathbf{b}, \mathbf{W}) &= -\mathbf{b} \cdot \mathbf{x} - \mathbf{x}^T \mathbf{W} \mathbf{x} \\ &= -\sum_i b_i x_i - \sum_i \sum_j w_{ij} x_i x_j. \end{aligned} \quad (2)$$

Physically, the bias field b_i represents the magnetic field acting on the i^{th} dipole, inclining (biasing) its spin towards whichever of the two directions it points in. The weight term w_{ij} then represents the magnetic interaction between any two (neighbouring) dipoles, such that a large positive value would incline the spins of the pair into alignment and vice versa.

As a machine learning model, it is the set of biases and weights (\mathbf{b}, \mathbf{W}) that comprises the model parameters of the BM, $\vec{\theta}$. Recalling the goal of generative learning, training the BM to learn some true discrete distribution $q(\mathbf{x})$, involves finding the best instance of the aforementioned physical variables that would minimise the distance of the Gibbs distribution from it.

3.2 Quantum Boltzmann Machines

The classical Ising Model, and by extension, the BM, does not capture quantum phenomena such as entanglement or superposition. For that, a reformalism is required.

Despite the infamous unpredictability of quantum systems, there are still many physical quantities that can be measured from them. Such quantities are called *observables*, which mathematically correspond to Hermitian operators, or matrices acting on a Hilbert space (Nielsen and Chuang 2010) and whose spectrum contains the set of possible outcomes following measurement of that quantity.

If the Ising Model describing a lattice of n dipoles is reformulated as a quantum system, the energy of its 2^n possible states must now be expressed in terms of the energy observable \hat{H} , also known as the *Hamiltonian*. For the simple case of the classical Ising model described in the previous section in which there is no account of superposition nor entanglement, it is easy to see that \hat{H} will have eigenstates $\mathbf{X} = \{|0\rangle, |1\rangle\}^n$, with energy spectrum $\{E(\mathbf{x}) | \forall \mathbf{x} \in \mathbf{X}\}$, lending to the form

$$\begin{aligned} \text{diag}(\{E(\mathbf{x}) | \forall \mathbf{x} \in \mathbf{X}\}) &= \hat{H} \\ &= - \sum_i b_i \sigma_i^z - \sum_i \sum_j w_{ij} \sigma_i^z \sigma_j^z. \end{aligned} \quad (3)$$

Where σ_i^z is the Pauli Z matrix acting on the i^{th} spin, as described in further detail in (Amin et al. 2018)

In the last line of equation (3), the earlier contrived setup is replaced with an equivalent and much more organic expression reminiscent of the classical Ising Model.

With this generalised definition of the energy in the Ising model as a Hamiltonian observable, a new quantum mechanical variable may be introduced known as the transverse field $\mathbf{\Gamma}$ (Amin et al. 2018):

$$\hat{H} = - \sum_i \Gamma_i \sigma_i^x - \sum_i b_i \sigma_i^z - \sum_i \sum_j w_{ij} \sigma_i^z \sigma_j^z. \quad (4)$$

Unlike the bias field b_i , Γ_i is the magnetic field acting on the i^{th} dipole in the direction transverse to the spins. The effect of this is that it induces a spin superposition (Nielsen and Chuang 2010) within the dipole, so that the eigenstates of the Hamiltonian are no longer simple computational basis states, but superpositions thereof.

As with any ML model, it must be noted that the ultimate goal of extracting a Gibbs distribution $p(\mathbf{x} | \mathbf{b}, \mathbf{W}, \mathbf{\Gamma})$ still remains. Conveniently, $p(\mathbf{x})$ over the 2^n computational basis states correspond to the diagonal entries of the density matrix ρ (Nielsen and Chuang 2010) of the system

$$\rho = \frac{e^{-H}}{Z} = \frac{e^{-H}}{\text{Tr}[e^{-H}]} \quad (5)$$

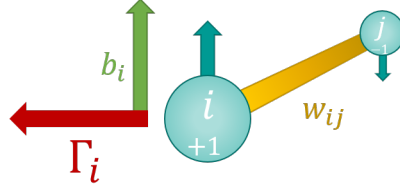


Figure 3: The Transverse Field Ising Model

4 Model Parameters

As mentioned in the introduction, the effect of the model parameters on generative learning can be quantified through the error function $\mathcal{L}(p, q)$. One commonly used error function is the negative average log-likelihood, which reads

$$\mathcal{L}(p, q) = - \sum_{\mathbf{x} \in \Lambda^{|V|}} q(\mathbf{x}) \log p(\mathbf{x}) \quad (6)$$

and is minimum wherever the model parameters $\vec{\theta}$ yield p that best approximates q . The next few sections observe the topology of this error (distance) landscape when conditioned on two model parameters. The true distributions $q(x)$ that are to be learned in subsequent sections are shown in Figure 4.

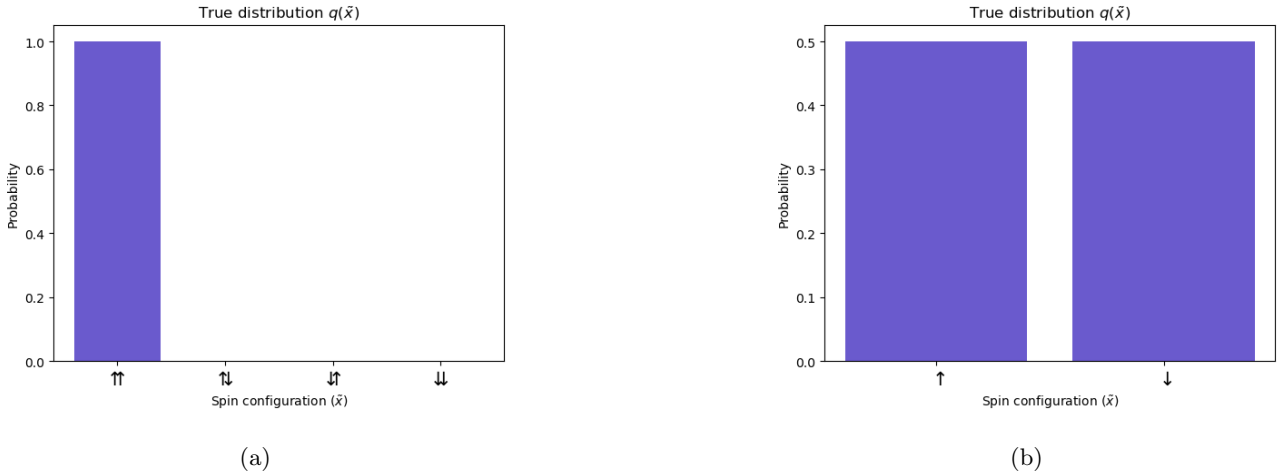


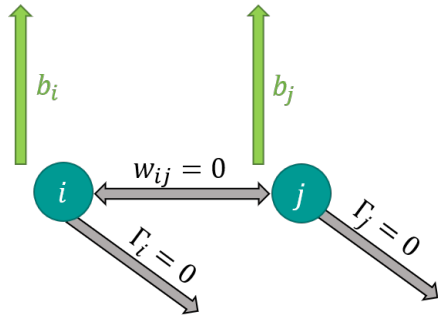
Figure 4: True distributions being learned by the (a) Bias, Weight, and (b) Transverse models

4.1 Bias Field

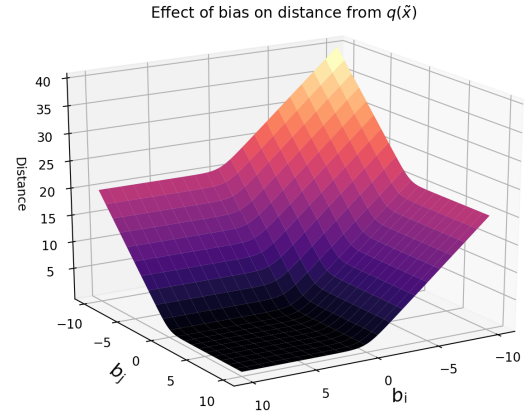
To investigate the effect of the bias field on learning the true distribution $q(x)$ shown in Figure 4a), the QBM in Figure 5a was used.

Shown in green are the two bias parameters b_i and b_j that will be varied. All other parameters (in grey) besides the pair of biases have been fixed to a value of 0. From here onward, unless otherwise stated, any parameter that is not shown is set to 0.

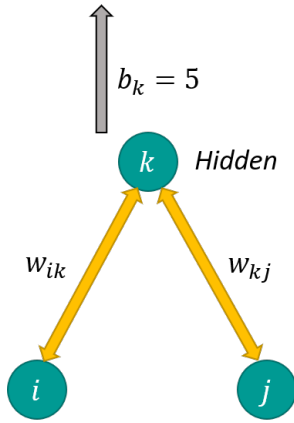
The distance landscape as a function of these two bias fields can be seen in Figure 5b.



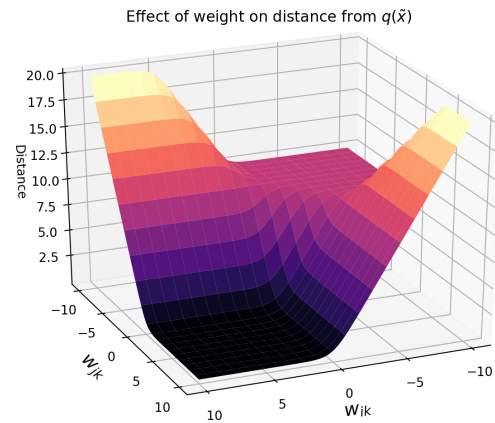
(a) Bias model



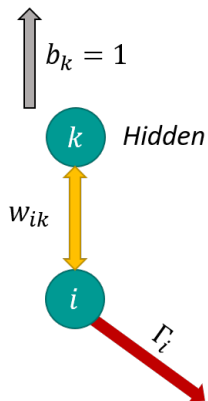
(b)



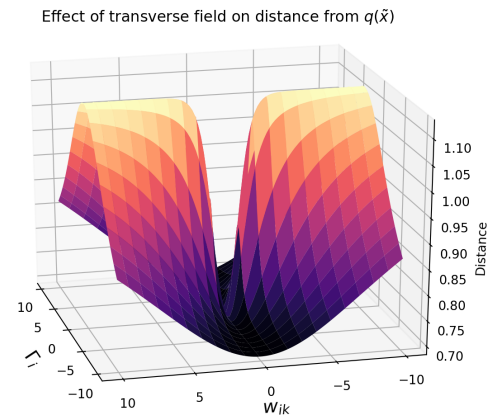
(c) Weight model



(d)



(e) Transverse model



(f)

Figure 5: The QBMs used (left column) and the respective distance landscape produced when two of their model parameters are varied (right)

The most notable feature of Figure 5b is that it is partitioned into four quadrants, with the quadrant of minimum distance being the one in which both biases are positive. This is expected as under these conditions, both dipoles have their spins biased in the positive ($|\uparrow\rangle$) direction, and not in any other configuration - exactly what is required by the true distribution. On the other hand, the opposite quadrant has neither dipole's spin biased in the desired direction, thus conferring it the highest distance away from $q(x)$.

4.2 Weight

For the weight model, Figure 5c was used to model the same distribution shown in Figure 4a).

Note that there is now the inclusion of a hidden dipole, whose value is not recorded in the final readout of the QBM state \mathbf{x} . Instead, it serves as a linking dipole encoding dependencies between the visible nodes i and j . This time the weight parameters between each visible and hidden dipole pair, w_{ij} and w_{jk} , are being varied, producing the distance landscape in Figure 5d.

Again, the first quadrant is that of the lowest distance from $q(x)$. Here, both weights are positive, meaning that dipoles i and j are coaxed into alignment with k . And since k is already biased upwards, the two visible dipoles to which it is connected will be too. Notably, however, there is a plateau on the opposite quadrant as well, albeit at a higher elevation. Although both weights being negative would incline the dipoles into anti-alignment to k , the fact that they are the same sign means that there will still be some (small) probability of the dipoles being in $|\uparrow\rangle$, thus the somewhat higher distance. This is as opposed to the remaining quadrants, in which i and j are in any case antiparallel to each other, and thus even further away from the true distribution.

The presence of flat plateaus for such a low-dimensional system might transfer over to the landscape of larger QBMs, which would severely disrupt any training by gradient descent.

4.3 Transverse Field

This time, the distribution being learned is the maximally mixed state shown in Figure 4b), for which the QBM model Figure 5e will be used.

Here, the transverse field Γ_i on i is varied, as well as the weight between the two dipoles w_{ik} , where k is once again, a hidden node. The two most important loci in figure Figure 5f are when either parameter is 0. Note when $w_{ik} = 0$, the visible node i is independent of k and so is in its naturally mixed state. Thus, the distance is consistently at a minimum, producing a connected valley. Conversely, when $\Gamma_i = 0$, the visible node i remains purely coupled to k which is already biased upwards, thus offsetting it from its mixed state and adding distance away from $q(x)$.

It is important to note two additional features of this graph. Firstly, it is symmetric about the Γ_i axis, which reflects the fact that whether it be left or right, the effect of the transverse field remains the same on the visible dipole as inclining it into superposition. Secondly, for any fixed weight, the distance decreases as the transverse field increases in magnitude. In this case, the effect of the transverse field dominates over the coupling on i , leading it to progressively resemble the true uniform distribution in Figure 4b).

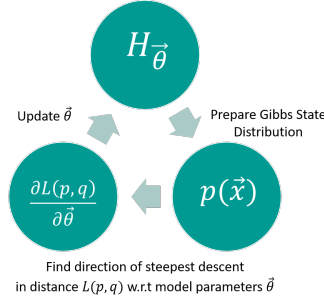


Figure 6: One cycle of training the QBM (Zoufal, Lucchi, and Woerner 2021)

5 Implementation

Thus far, only a conceptual investigation of QBMs has been conducted. The main goal of QBMs in practice is implementing one on a real device and observing its performance. The procedure outlined in Figure 6 illustrates the overall training process consisting of three main steps:

1. Physically simulate the Gibbs state ρ (and thus $p(\mathbf{x})$) from the Hamiltonian $H(\vec{\theta})$ on a quantum computer.
2. Obtain the gradient of the distance function $\mathcal{L}(p, q)$ with respect to the model parameters $\vec{\theta}$.
3. Update $\vec{\theta}$ and repeat until convergence

5.1 Gibbs State Preparation

In the previous section, the Gibbs distribution extracted to produce the loss landscape had been obtained numerically by explicitly calculating the matrix exponential (Amin et al. 2018). This, of course, defeats the purpose of the QBM as a model conferring some speed-up.

To forgo explicit calculation, the principle of Imaginary Time Evolution (ITE) may be used (McArdle et al. 2019). For a system in state $|\psi_0\rangle$ with Hamiltonian $\hat{H} = \sum_p \theta_p h_p$ where h_p are Pauli terms as defined in (Zoufal, Lucchi, and Woerner 2021), ITE propagates the state as,

$$|\psi(\tau)\rangle = C e^{-\hat{H}\tau} |\psi_0\rangle \quad (7)$$

where the normalisation constant is,

$$C(\tau) = \frac{1}{\sqrt{\text{Tr}[e^{-2\hat{H}\tau} |\psi_0\rangle \langle \psi_0|]}}.$$

Observe that when the initial mixed state is maximally mixed $\rho_0 = |\psi_0\rangle \langle \psi_0| = \frac{1}{d}I$, the propagated mixed state at $\tau = \frac{1}{2}$ is the desired Gibbs distribution in (5).

Equation (7) may be produced by means of an appropriate variational quantum circuit encoding an Ansatz

state (Zoufal, Lucchi, and Woerner 2021),

$$\begin{aligned} |\psi(\omega)\rangle &= U_q(\omega_q) \dots U_1(\omega_1) |00\dots 0\rangle \\ &= V(\vec{\omega}) |\tilde{0}\rangle \end{aligned} \quad (8)$$

Then, the projection of $|\psi(\tau)\rangle$ onto $|\psi(\vec{\omega}(\tau))\rangle$ may be obtained – which is hoped to be a close approximation – by using McLachlan’s variational principle,

$$\delta \|(d/d\tau + H - \langle \psi(\vec{\omega}(\tau)) | H | \psi(\vec{\omega}(\tau)) \rangle) |\psi(\vec{\omega}(\tau))\rangle\| = 0$$

which ultimately yields the system of equations (McArdle et al. 2019),

$$\mathbf{A}\dot{\boldsymbol{\omega}} = \mathbf{C} \quad (9)$$

where

$$\begin{aligned} A_{ij} &= \text{Re} \left(\text{Tr} \left[\frac{\partial \langle \psi(\vec{\omega}(\tau)) |}{\partial \omega(\tau)_i} \frac{\partial |\psi(\vec{\omega}(\tau))\rangle}{\partial \omega(\tau)_j} \right] \right) \\ C_i &= - \sum_p \theta_p \text{Re} \left(\text{Tr} \left[\frac{\partial \langle \psi(\vec{\omega}(\tau)) |}{\partial \omega(\tau)_i} h_p |\psi(\vec{\omega}(\tau))\rangle \right] \right). \end{aligned} \quad (10)$$

Solving (9) iteratively allows for the Ansatz parameters to be propagated by Euler’s method to $\tau = 1$, yielding the state parameters required to approximate the Gibbs state,

$$\vec{\omega}(\tau) = \vec{\omega}(0) + \sum_{n=0}^{\tau/\delta\tau-1} \dot{\boldsymbol{\omega}}(n\delta\tau) \delta\tau.$$

5.1.1 Computing C

Note the entries of C can be rewritten as (Schuld et al. 2019),

$$\begin{aligned} C_i &= -\frac{1}{2} \left(\langle \psi(\vec{\omega}) | H \partial_{\omega_i} |\psi(\vec{\omega})\rangle + h.c. \right) \\ &= -\frac{1}{2} \left(\langle \tilde{0} | V(\vec{\omega})^\dagger H \partial_{\omega_i} V(\vec{\omega}) | \tilde{0} \rangle + h.c. \right). \end{aligned}$$

Now, assuming only the gate U_i in $V(\vec{\omega}) = U_q(\omega_q) \dots U_i(\omega_i) \dots U_1(\omega_1)$ depends on ω_i ,

$$\begin{aligned} \partial_{\omega_i} V(\vec{\omega}) &= U_q(\omega_q) \dots \partial_{\omega_i} U_i(\omega_i) \dots U_1(\omega_1) \\ &:= M \partial_{\omega_i} U_i N \end{aligned}$$

where the adjacent terms have been redefined as M and N respectively, and the independent variable w_i is implied by U_i for clarity. By conjugating H with M so that $\hat{Q} := M^\dagger \hat{H} M$, and defining $|\psi\rangle := N |\tilde{0}\rangle$, the expression for C_i simplifies to,

$$C_i = -\frac{1}{2} \left(\langle \psi | U_i^\dagger \hat{Q} \partial_{\omega_i} U_i | \psi \rangle + h.c. \right). \quad (11)$$

Moreover, further factorisation yields,

$$\begin{aligned} C_i &= -\frac{1}{4} \left(\langle \psi | (U_i + \partial_{\omega_i} U_i)^\dagger \hat{Q} (U_i + \partial_{\omega_i} U_i) | \psi \rangle \right. \\ &\quad \left. - \langle \psi | (U_i - \partial_{\omega_i} U_i)^\dagger \hat{Q} (U_i - \partial_{\omega_i} U_i) | \psi \rangle \right). \end{aligned} \quad (12)$$

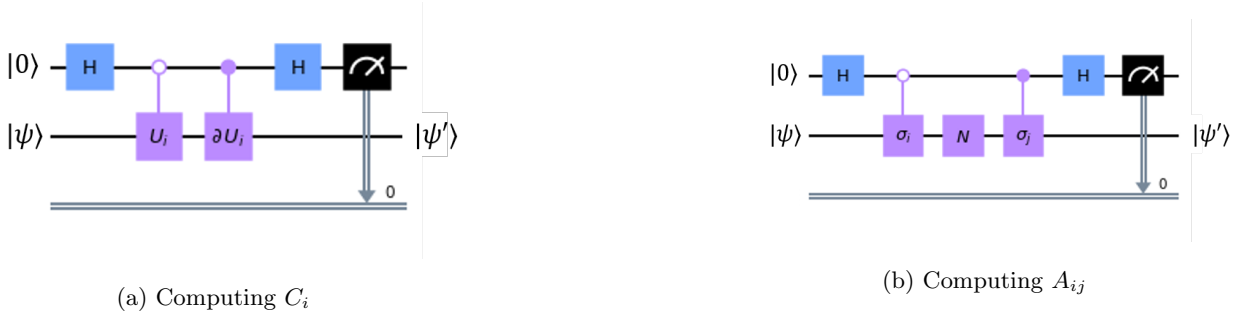


Figure 7: Circuits for computing (a) C and (b) A

As shown in (McArdle et al. 2019), for a general unitary gate, $\partial_{\omega_i} U_i$ can be decomposed to a linear combination of unitary gates also, which means it is easily implementable on a quantum computer.

With the help of an ancilla, equation (12) can be computed using the circuit shown in Figure 7a). Let p_0 and p_1 be the probabilities of measuring 0 and 1 on the ancilla, and thus, having the main state collapse to $|\psi'_0\rangle$ and $|\psi'_1\rangle$ respectively. If $E_0 = \langle \psi'_0 | \hat{Q} | \psi'_0 \rangle$ and $E_1 = \langle \psi'_1 | \hat{Q} | \psi'_1 \rangle$ (Nielsen and Chuang 2010), it can be shown that Equation (12) is equivalent to (Schuld et al. 2019)

$$C_i = -(p_0 E_0 - p_1 E_1). \quad (13)$$

5.1.2 Computing A

Conveniently, the entries of A can be massaged into resembling those of C :

$$\begin{aligned} A_{ij} &= \frac{1}{2} (\langle \partial_{\omega_j} \psi(\vec{\omega}) | \partial_{\omega_i} \psi(\vec{\omega}) \rangle + h.c.) \\ &= \frac{1}{2} (\langle \tilde{0} | U_1^\dagger \dots \partial_{\omega_i} U_i^\dagger \dots U_q^\dagger U_q \dots \partial_{\omega_j} U_j \dots U_1 | \tilde{0} \rangle + h.c.) \end{aligned}$$

assuming $i < j$,

$$= \frac{1}{2} (\langle \tilde{0} | U_1^\dagger \dots \partial_{\omega_i} U_i^\dagger \dots U_j^\dagger \partial_{\omega_j} U_j \dots U_1 | \tilde{0} \rangle + h.c.)$$

letting $|\psi\rangle := U_{i-1} \dots U_1 |\tilde{0}\rangle$,

$$= \frac{1}{2} (\langle \psi | \partial_{\omega_i} U_i^\dagger \dots U_j^\dagger \partial_{\omega_j} U_j \dots U_i | \psi \rangle + h.c.).$$

As mentioned in (McArdle et al. 2019), assuming U is a Pauli gate, $\partial U = \alpha \sigma U$ where α is a known phase factor and σ a Pauli gate such that $[\sigma, U] = 0$,

$$\begin{aligned} &= \frac{1}{2} \alpha_i \alpha_j (\langle \psi | \sigma_i^\dagger U_i^\dagger \dots U_j^\dagger U_j \sigma_j \dots U_i | \psi \rangle + h.c.) \\ &= \frac{1}{2} \alpha_i \alpha_j (\langle \psi | \sigma_i^\dagger U_i^\dagger \dots U_{j-1}^\dagger \sigma_j U_{j-1} \dots U_i | \psi \rangle + h.c.). \end{aligned}$$

Now by labelling $N := U_{j-1} \dots U_i$,

$$= \frac{1}{2} \alpha_i \alpha_j (\langle \psi | \sigma_i^\dagger N^\dagger \sigma_j N | \psi \rangle + h.c.). \quad (14)$$

Note that the form of (14) closely resembles that of (11), with the added simplification of $\hat{Q} = I$. It is no surprise, then, that the entries may be computed with Figure 7b) following the same procedure as in (13).

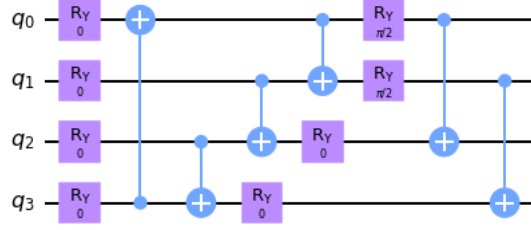


Figure 8: The Ansatz used to simulate ρ from \hat{H} (Zoufal, Lucchi, and Woerner 2021)

Moreover, because $\hat{Q} = I$, the expectation values $E_0 = E_1 = 1$, so (McArdle et al. 2019)

$$\begin{aligned} A_{ij} &= \alpha_i \alpha_j (p_0 - p_1) \\ &= \alpha_i \alpha_j (2p_0 - 1). \end{aligned} \quad (15)$$

5.2 Gradient Descent

While there exists an analytical method (Zoufal, Lucchi, and Woerner 2021) to computing the gradients $\frac{\partial \mathcal{L}(p_{\bar{\theta}}, q)}{\partial \bar{\theta}}$, a finite differences approach was adopted for simplicity:

$$\partial_{\theta_i} \mathcal{L}(p_{\bar{\theta}}, q) = \frac{\mathcal{L}(p_{\theta_i + \epsilon}, q) - \mathcal{L}(p_{\theta_i - \epsilon}, q)}{2\epsilon}. \quad (16)$$

This was repeated for all Hamiltonian parameters to build $\nabla \mathcal{L}$ which was used to update $\bar{\theta}$ accordingly:

$$\bar{\theta} \leftarrow \bar{\theta} - \eta \nabla \mathcal{L}.$$

5.3 Results

The Ansatz in Figure 8 was used to simulate the Gibbs distribution described by the following Hamiltonian

$$\hat{H}(\mathbf{b}, \mathbf{W}, \mathbf{\Gamma}) = b_1 \sigma_z^1 + b_2 \sigma_z^2 + w_{12} \sigma_z^1 \sigma_z^2 + \Gamma_1 \sigma_x^1 + \Gamma_2 \sigma_x^2 \quad (17)$$

One instance of VarQITE can be seen in Figure 9:

Figures 10 and 11 display the model’s performance in learning a simple Bell state distribution on a quantum computing simulator.

6 Discussion

Using the Ansatz in Figure 8, the circuits in Figure 7 were implemented on the QISKIT statevector simulator in order to reproduce the true probability distribution shown in Figure 10. While Figure 11 clearly shows that the QBM is learning the desired distribution, tapering off beyond 4 steps, Figure 10 reveals how the trained Gibbs distribution still exhibits inaccuracies, such as the nonzero values for the mixed spin states, and the underestimation of the maximally excited state.

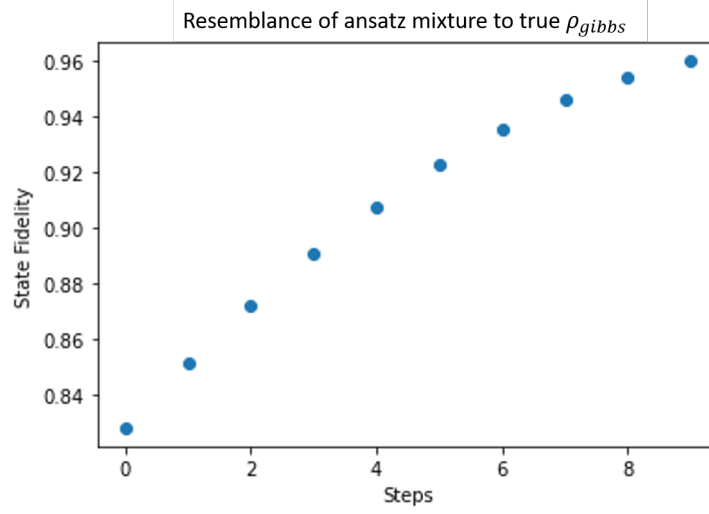


Figure 9: State fidelity of Ansatz mixture to desired ρ_{gibbs} over ten steps of VarQITE

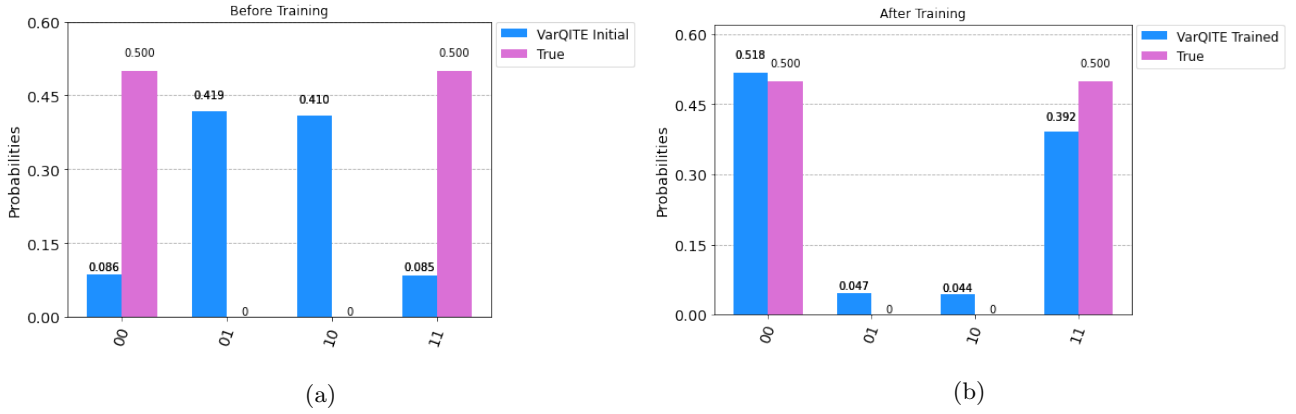


Figure 10: Gibbs distribution $p(\mathbf{x})$ a) before and b) after training

These inadequacies hint at the presence of multiple confounding factors, the most salient of which is that the VarQITE package (Zoufal, Lucchi, and Woerner 2021) used to prepare the Gibbs states had an error which disrupted the fidelity of the final mixture to the theoretical Gibbs distribution defined by the Hamiltonian. The effect of this error can be observed in Figure 9, where despite there being a general increasing trend in state fidelity, by the end of VarQITE, the simulator only caps at 0.96 which is lower than that produced by Zoufal, Lucchi, and Woerner 2021. Apart from technical issues, there also remains the fact that VarQITE aims to obtain the *projection* of the desired Gibbs state $|\psi(\tau)\rangle$ onto the subspace spanned by the variational state $|\psi(\omega(\tau))\rangle$, and depending on the (potentially limited) representational capacity of the Ansatz this could easily reduce the maximum state fidelity reached. Much more research is needed in devising metrics that could assess the adequacy of an Ansatz for this intrinsic setback to be potentially curbed.

The fact that training was successful in Figure 11 implies the distance landscape was smooth and not too flat which would've otherwise lead to no significant decrease in the loss. However, no conclusion can be made about

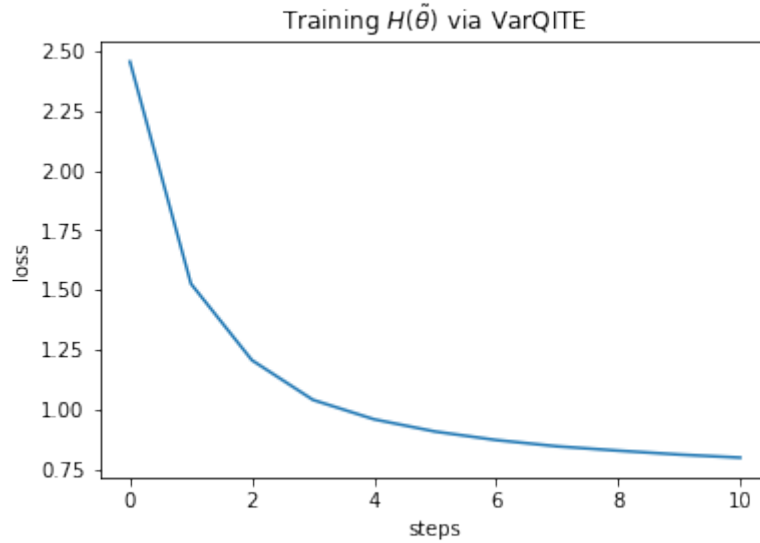


Figure 11: Loss (distance) over training steps

the general topology of the distance landscape, so whether this smoothness scales with size isn't guaranteed. In fact, research has shown that sufficiently parameterised variational circuits are prone to the occurrence of plateau barrens in their distance function (Marrero, Kieferová, and Wiebe 2020), which would suggest training quality gets worse with increasing size.

There also exist certain computational limitations in this implementation of the QBM. By having to solve (9) at each step, Gibbs state preparation by VarQITE is very time-consuming. However, recent strides in quantum machine learning have revealed efficient quantum algorithms to help with the exact problem of solving a linear system of equations. Perhaps, this could be incorporated into the model to make VarQITE more efficient. Additionally, the finite differences method used for gradient descent quickly becomes unfeasible for a large number of Hamiltonian parameters - a standard which is expected for practical implementations in the future.

7 Conclusion

Having defined the QBM architecture as a Markov Random Field with an energy function based on the Ising model, this project first set out to explore the effect of the different model parameters on the topology of the distance landscape used in training. This led to the discovery of notable regions such as extremely flat plateaus and valleys.

As an ML model, the QBM was implemented semi-classically on a quantum computer using the VarQITE algorithm. While the model shows some potential in generative learning, its effectiveness is mostly set back by lingering confounding factors. Theoretical issues include the representational (in)capacity of the Ansatz used in VarQITE, as well as the occurrence of plateaus in the distance landscape which would render gradient descent obsolete. Additionally, there is the practical issue of computational overhead with both VarQITE,

and finite difference calculations, especially for larger systems consisting of highly parameterised Hamiltonians and variational circuits. Future research should systematise the process of selecting an adequate Ansatz for approximating a desired state. Moreover, currently classical subroutines in VarQITE (e.g. solving a system of equations) could be replaced with quantum counterparts that have been shown to result in speedup.

8 Acknowledgements

Special thanks to Christa Zoufal for her consistent help in navigating about the VarQITE algorithm. And of course, I extend my gratitude to AMSI for making this project possible in the first place, as well as my supervisors Charles Hill and Thomas Quella without whom I would've been completely adrift.

9 Statement of Authorship

The workload was distributed as follows:

- Saleh Naghdi produced the code used in both modelling the distance landscape, and implementation as well as reported and interpreted the findings into the write-up of this report.
- Charles Hill and Thomas Quella inspired the research investigation, supervised this project by assisting with demystifying points of concern, and proofread this report.

10 Code Availability

All code related to either graphing the distance landscape of an arbitrary QBM or implementing one on a Qiskit quantum computer can be obtained here.

References

- [1] Mohammad H. Amin et al. “Quantum Boltzmann Machine”. In: *Physical Review X* 8.2 (May 2018). ISSN: 2160-3308. DOI: 10.1103/physrevx.8.021050. URL: <http://dx.doi.org/10.1103/PhysRevX.8.021050>.
- [2] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (Sept. 2017), pp. 195–202. ISSN: 1476-4687. DOI: 10.1038/nature23474. URL: <http://dx.doi.org/10.1038/nature23474>.
- [3] Asja Fischer and Christian Igel. “Training restricted Boltzmann machines: An introduction”. In: *Pattern Recognition* 47.1 (Jan. 2014), pp. 25–39. ISSN: 00313203. DOI: 10.1016/j.patcog.2013.05.025.
- [4] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. *Entanglement Induced Barren Plateaus*. 2020. arXiv: 2010.15968 [quant-ph].

- [5] Sam McArdle et al. “Variational ansatz-based quantum simulation of imaginary time evolution”. In: *npj Quantum Information* 5.1 (Sept. 2019). ISSN: 2056-6387. DOI: 10.1038/s41534-019-0187-2. URL: <http://dx.doi.org/10.1038/s41534-019-0187-2>.
- [6] M.A. Nielsen and I.L. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [7] Maria Schuld et al. “Evaluating analytic gradients on quantum hardware”. In: *Physical Review A* 99.3 (Mar. 2019). ISSN: 2469-9934. DOI: 10.1103/physreva.99.032331. URL: <http://dx.doi.org/10.1103/PhysRevA.99.032331>.
- [8] Sumio Watanabe. *Algebraic geometry and statistical learning theory*. Vol. 25. Cambridge University Press, 2009.
- [9] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. “Variational quantum Boltzmann machines”. In: *Quantum Machine Intelligence* 3.1 (Feb. 2021). ISSN: 2524-4914. DOI: 10.1007/s42484-020-00033-7. URL: <http://dx.doi.org/10.1007/s42484-020-00033-7>.