Final Project Report

Quantum Annealers, Restricted Boltzmann Machines, and the Ising Model

YASHAS SHENDE

Mentors: Iván Mauricio Burbano Aldana, Marco A. Carrillo, Siddhant Mal

Project Period: 1st July, 2023 to 1st September 2023

Report Submission: 21st August, 2023

1 Abstract

Quantum Computing promises great potential in running complex simulations that would otherwise take a very long time if run on classical super computers. A lot of development is ongoing and it is worth exploring what the applications of quantum computing are. One such application is found to be in determining various observables like magnetisation, energy, specific heat, etc. in statistical models like the Ising model. While using a Restricted Boltzmann Machine (RBM)- a type of neural network consisting of symmetrically coupled stochastic, binary units or nodes- is very efficient, the conventional methods of training it have certain flaws that could potentially be mitigated by using a Quantum Annealer. In this paper, I explore how an RBM actually works and how it can be used to study the Ising model. I also study the similarities between an RBM and the Ising model- in that they are both Energy-based models. Finally, I try to determine how using a Quantum Annealer, particularly a D-Wave Quantum Annealer in place of the conventional Contrastive Divergence (CD) to train an RBM, is better [1].

2 Aim

1) To use Restricted Boltzmann Machine (RBM) in place of the conventional Markov Chain Monte Carlo (MCMC) method (see [2] for more details)to study various observables like magnetisation per spin, energy per spin, etc. in lattice field theories like the Ising model.

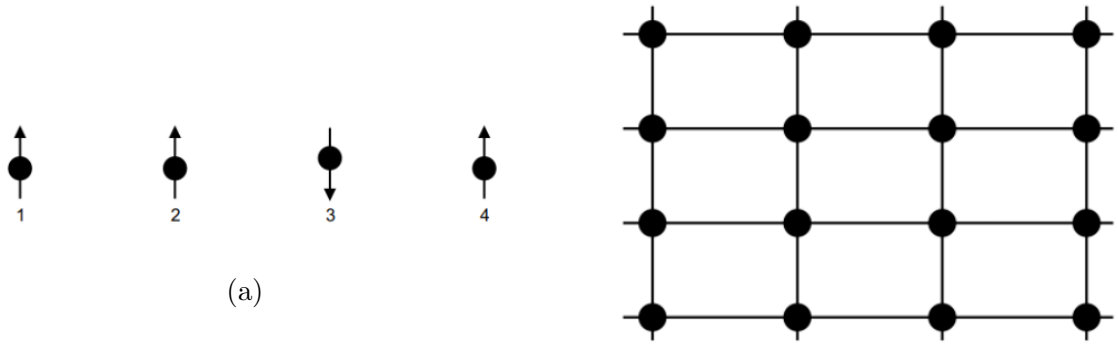
2) To explore the advantage of using a Quantum Annealer in place of the conventional Contrastive Divergence (CD) to train a Restricted Boltzmann machine.

3 Theoretical Background

3.1 The Ising Model

3.1.1 Classical Ising Model:

The classical Ising model is a mathematical model of ferromagnetism in statistical mechanics. Ferromagnetism arises when a collection of atomic spins align such that their associated magnetic moments all point in the same direction, yielding a net magnetic moment which is macroscopic in size. The model consists of discrete variables that represent magnetic dipole moments of atomic "spins" that can be in one of two states (+1 or -1). The spins are arranged in a graph, usually a lattice (where the local structure repeats periodically in all directions), allowing each spin to interact with its neighbours [3]. A 1D and 2D representation of the model is shown below:



(a) (b)

Figure 1: (a) 1D Ising Model and (b) 2D Ising Model. Source: [4]

The Hamiltonian of this system is defined as:

$$H = -J \sum_{\langle ij \rangle} s_i s_j - \mu h \sum_{i=1,N} s_i \tag{1}$$

Here, $\langle ij \rangle$ is the sum over nearest neighbouring pairs of atoms, J is the coupling constant, μ is the atomic magnetic moment, H is the external magnetic field, and s_i is the z^{th}- component of the i^{th} atomic spin.

Neighbouring spins that agree have a lower energy than those that disagree. This effect is mostly due to the Pauli exclusion principle. Electrons cannot occupy the same quantum state, so two electrons on neighbouring atoms which have parallel spins (i.e., occupy the same orbital state) cannot come close together in space. No such restriction applies if the electrons have anti-parallel spins. Different spatial separations imply different electrostatic interaction energies, and the coupling constant, J, measures this difference. The system tends to the lowest energy but heat disturbs this tendency, thus creating the possibility of different structural phases [5].

We will use a *mean field approximation*, and the energy of the i^{th} atom is:

$$e_i = -\frac{J}{2} \sum_{k=1,z} s_k s_i - \mu h s_i$$

where the sum is over the z nearest neighbours of atom i. The factor $\frac{1}{2}$ is needed to ensure that when we sum to obtain the total energy, we do not count each pair of neighbouring atoms twice [5].

Now,

$$e_i = -\mu H_{\text{eff}}\, s_i \tag{2}$$

where,

$$H_{\text{eff}} = h + \frac{J}{2\mu} \sum_{k=1,z} s_k \tag{3}$$

Here, H_{eff} is the effective magnetic field, which is made up of two components: the external field, h, and the internal field generated by neighbouring atoms.

Consider a single atom in a magnetic field h_m. Suppose that the atom is in thermal equilibrium with a heat bath of temperature T. According to the Boltzmann distribution, the mean spin of the atom is:

$$\bar{s} = \frac{\mathrm{e}^{+\beta\mu h_m} - \mathrm{e}^{-\beta\mu h_m}}{\mathrm{e}^{+\beta\mu h_m} + \mathrm{e}^{-\beta\mu h_m}}$$

where $\beta = \frac{1}{kT}$, and k is the Boltzmann constant. The above expression follows because the energy of the "spin up" state ($s = +1$) is $-\mu\, h_m$, whereas the energy of the "spin down" state ($s = -1$) is $+\mu\, h_m$.

Hence,

$$\bar{s} = \tanh(\beta\, \mu\, h_m) \tag{4}$$

Let us assume that all atoms have identical spins: i.e., $s_i = \bar{s}$. This assumption is known as the "*mean field approximation*". We can write:

$$H_{\text{eff}} = h + \frac{zJ\bar{s}}{2\mu} \tag{5}$$

Finally, setting ($h_m = H_{\text{eff}}$), we get [5]:

$$\bar{s} = \tanh\left\{\beta\mu H + \frac{\beta z J \bar{s}}{2}\right\} \tag{6}$$

Note that the heat bath in which a given atom is immersed is simply the rest of the atoms. Hence, T is the temperature of the atomic array. The critical temperature is defined as:

$$T_c = \frac{z\,J}{2\,k} \tag{7}$$

The critical magnetic field is defined as:

$$H_c = \frac{k\,T_c}{\mu} = \frac{z\,J}{2\,\mu} \tag{8}$$

Now, Equation (6) becomes:

$$\bar{s} = \tanh\left\{\frac{T_c}{T}\left(\frac{H}{H_c} + \bar{s}\right)\right\} \tag{9}$$

Finally, we define the three physical observables, starting with the net magnetisation [5]:

$$M = \mu \sum_{i=1,N} s_i = \mu\, N\, \bar{s} \tag{10}$$

Next, we have the net energy:

$$E = \sum_{i=1,N} e_i = -N\, k\, T_c \left(\frac{H}{H_c} + \bar{s}\right) \bar{s} \tag{11}$$

Lastly, the heat capacity:

$$C = \frac{dE}{dT} \tag{12}$$

Given below are three graphs depicting the net magnetisation, net energy, and the specific heat generated using an iterative algorithm in the absence of an external magnetic field ($h = 0$) [5].

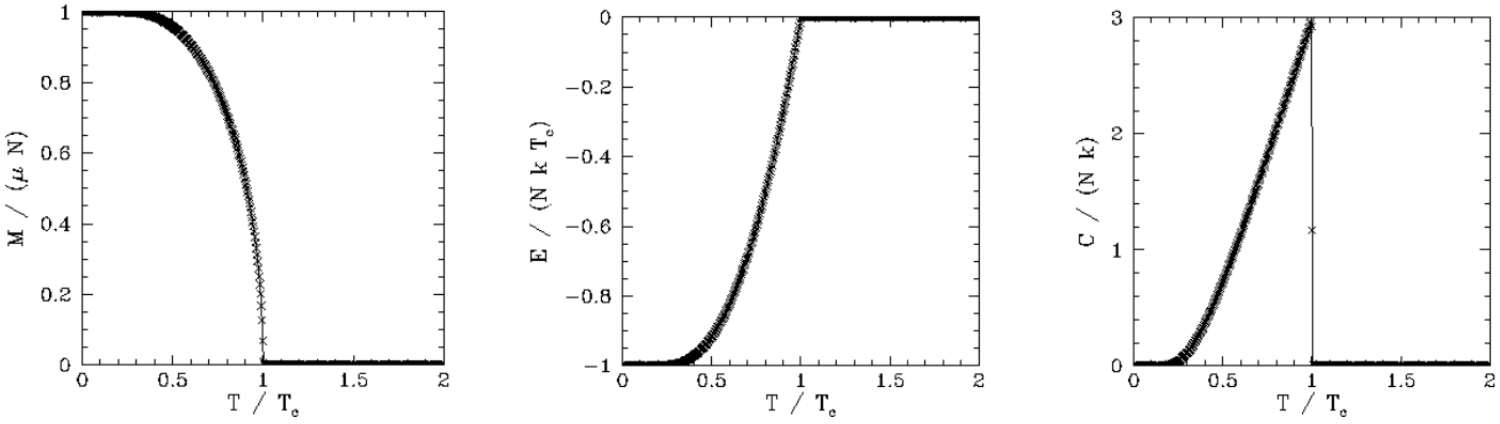


Figure 2: The magnetisation per spin M (left), the energy per spin E (centre), and the specific heat per spin C (right) when $N \to \infty$. Calculated using the mean-field method. At T_C the discontinuity that gives rise to the phase transition is visible. Source: [5]

These graphs show the following:
Below the critical temperature (T_C), there is spontaneous magnetisation, however, above it, thermal fluctuations completely eliminate any alignment. At T_C, there is a discontinuity in the first derivative of the energy, E, with respect to the temperature, T. This discontinuity generates a downward jump in the heat capacity, C, at $T = T_C$ [5] [6].

3.1.2 Quantum Ising Model:

The Quantum Ising model is quite similar to the classical one. We character its quantum state by a unit vector in the complex space $\mathbb{C}^d$, and its dynamic evolution by a Hermitian matrix of size d, which is called a quantum Hamiltonian for the quantum system. The energies

of the quantum system are defined to be the eigenvalues of the quantum Hamiltonian, and ground states refer to the eigenvectors corresponding to the smallest eigenvalue.

$$I_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_j^x \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_j^z \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad j = 1, ..., N$$

where, I_j is the Identity matrix, σ_j^x and σ_j^z are the Pauli matrices in the x and z axes respectively, and N is the number of sites in the Ising model.
The Hamiltonian of this system is as follows:

$$H_q = -J \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - \mu h \sum_{i=1,N} \sigma_i^z \tag{13}$$

The quantum Hamiltonian H_q is a diagonal matrix of size 2^N whose diagonal elements (eigenvalues) are the same as the classical Hamiltonian H in Equation (1). Therefore, finding the minimal energy of the classical Ising Hamiltonian H is equivalent to finding the minimal energy of the quantum Ising Hamiltonian H_q [7].

Quantum annealing requires an introduction of a transverse magnetic field to yield a quantum Hamiltonian in the transverse field [7]. The Hamiltonian of the transverse field is as follows:

$$H_x = -\sum_{i=1,N} \sigma_i^x \tag{14}$$

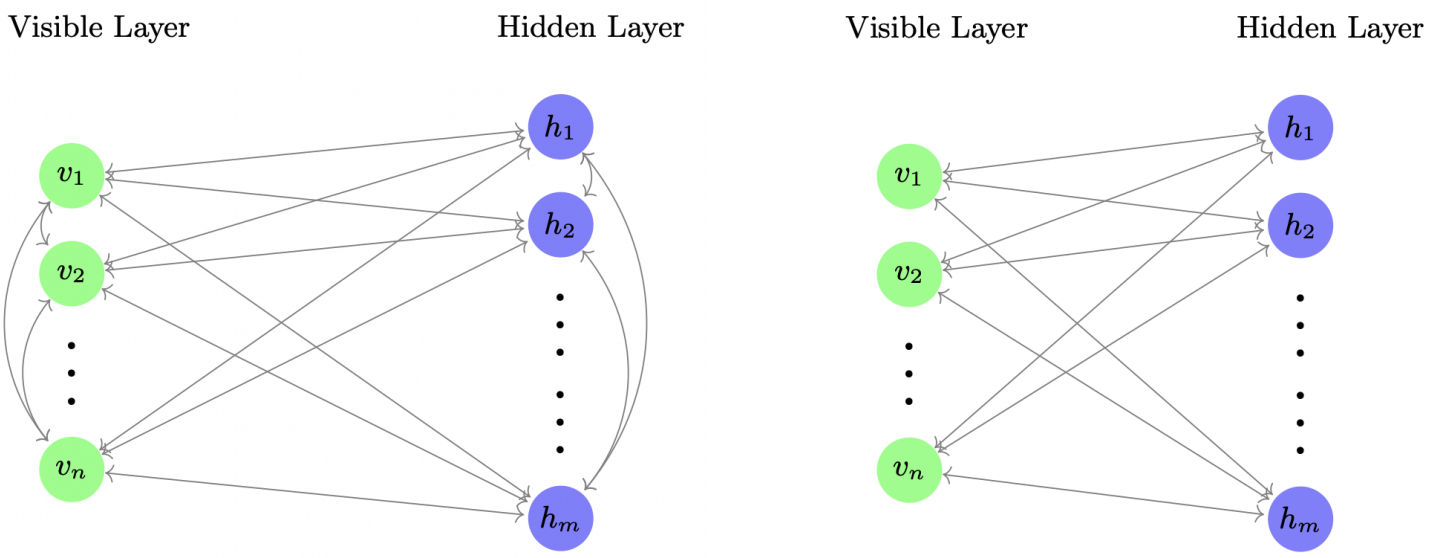
Now, the Hamiltonian for the Ising system in the transverse field is given below:

$$H_D(t) = A(t)\, H_x + B(t)\, H_q, \qquad t \in [0, t_f] \tag{15}$$

where t_f denotes the total annealing time, and time varying smooth functions $A(t)$ and $B(t)$ are called the annealing schedules which satisfy $A(t_f) = B(0) = 0$, $A(t)$ is decreasing, and $B(t)$ is increasing [7] [8]. The quantum annealing of $H_D(t)$ will be discussed later in this paper.

3.2 Boltzmann Machines (BM)

A BM is a network of symmetrically coupled stochastic binary units, which consists of a set of visible units $v \in \{0, 1\}^n$ associated with observations, and a set of hidden units $h \in \{0, 1\}^m$ used to capture dependencies between observed variables[7]. Every two nodes are connected so the model creates a fully connected undirected graph as shown in Figure 3.



(a) (b)

Figure 3: (a) Schematic representation of a Boltzmann Machine and (b) Schematic representation of a Restricted Boltzmann Machine

The learning process for general BMs is tedious which is why we may impose some restrictions on the network topology to simplify the learning problem. A Restricted Boltzmann Machine (RBM) is a special type of BM, where every visible node is connected to every hidden node, but there is no connection between two variables of the same layer as shown in Figure 3. Again, it contains n visible units $v = (v_1, \cdots, v_n)$ associated with observations and m hidden units $h = (h_1, \cdots, h_m)$ to capture dependencies between observed variables. Each node is a locus of computation that processes input and begins by making stochastic decisions about whether to transmit the input or not.

Let us now look at how an RBM is an energy-based model. Some deep learning architectures use the idea of energy as a metric for measurement of the model's quality. One purpose of deep learning models is to encode dependencies between variables; the capturing of dependencies happens through the association of a scalar energy to each configuration of the variables which serves as a measure of compatibility. A high energy means a bad compatibility; an energy based model always tries to minimise a predefined energy function [9]. The energy function for the RBM is defined using the following equation:

$$E(\mathbf{v},\, \mathbf{h}) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}\, h_i\, v_j - \sum_{j=1}^{n} a_j\, v_j - \sum_{i=1}^{m} b_i\, h_i \tag{16}$$

where v is the visible layer, h is the hidden layer, w_{ij} is the weight matrix, h_i are the hidden states, v_j are the visible states, a_j is the activation function, and b_i are the biases. The training of an RBM consists of finding the parameters for given input values so that the energy reaches a minimum.

At each point in time the RBM is in a certain state- which is basically the values of neurons in the visible and hidden layers v and h respectively. The probability that a certain state of v and h can be observed is given by the following joint distribution:

$$p(\mathbf{v},\ \mathbf{h}) = \frac{1}{Z}\, e^{-E(\mathbf{v},\ \mathbf{h})} \tag{17}$$

where,

$$Z = \sum_{\mathbf{v},\ \mathbf{h}} e^{-E(\mathbf{v},\ \mathbf{h})} \tag{18}$$

Here, Z is the partition function.

It is difficult to calculate the joint probability because of the large number of possible combinations of v and h in the partition function (Z). It is simpler to calculate the conditional probabilities of state h given state v, and vice versa, which are as follows:

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{m} p(h_i|\mathbf{v}) \tag{19}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{n} p(v_i|\mathbf{h}) \tag{20}$$

RBMs are generative models- which means that they aim to capture the underlying probability distribution of a dataset. This is a good point to introduce an indispensable function in the context of RBMs, known as the Likelihood function. It measures how likely the RBM is to generate the given data samples. The goal in training an RBM is to adjust its parameters (weights and biases) so that it can generate data samples that closely resemble the training data. This involves finding the set of parameters that maximises the likelihood of observing the given training data. The likelihood is the probability of observing the data given the model's parameters. Taking the logarithm of the likelihood simplifies the calculations and turns the product of probabilities into a sum of log probabilities, which is easier to work with mathematically. The Log Likelihood function provides a way to assess the quality of the RBM's generated samples compared to the real data. Optimising the RBM's parameters to maximise the Log Likelihood function involves using gradient-based optimisation algorithms like stochastic gradient ascent. The gradient of the Log Likelihood with respect to the model parameters guides the updates in a way that moves the parameters in a direction that increases the likelihood of the training data. Given one single training example $\bar{\mathbf{v}}$, the Log Likelihood of the RBM model with the parameter θ is given by [7]:

$$\ln\left\{L(\theta|\bar{\mathbf{v}})\right\} = \ln\left\{p(\bar{\mathbf{v}}|\theta)\right\} = \ln\left[\frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\bar{\mathbf{v}},\mathbf{h})}\right]$$

$$\implies \ln\{L(\theta|\bar{\mathbf{v}})\} = \ln \sum_{\mathbf{h}} e^{-E(\bar{\mathbf{v}},\mathbf{h})} - \ln \sum_{\mathbf{v},\ \mathbf{h}} e^{-E(\mathbf{v},\ \mathbf{h})} \tag{21}$$

The Gradient of the Log Likelihood function is given by:

$$\frac{\partial \ln\{L(\theta|\bar{\mathbf{v}})\}}{\partial\theta} = -\sum_{\mathbf{h}} p(\mathbf{h}|\bar{\mathbf{v}}) \frac{\partial E(\bar{\mathbf{v}},\mathbf{h})}{\partial\theta} + \sum_{\mathbf{v},\ \mathbf{h}} p(\mathbf{v},\ \mathbf{h}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \tag{22}$$

3.3 D-Wave Quantum Annealer

A classical annealer is a computational technique used to solve optimisation problems by simulating the physical process of annealing. Annealing is a concept from metallurgy where a material is heated and then gradually cooled to reduce defects and increase the regularity of its structure. In optimisation, the idea is to start with a high-energy state (high cost) and slowly transition to a low-energy state (low cost) by iteratively modifying the system's configuration.

A quantum annealer is a specialised type of quantum computing device designed to solve optimisation and sampling problems. Quantum annealing harnesses the principles of quantum mechanics to explore complex energy landscapes more efficiently than classical annealing.

The Hamiltonian described by Equation (15) in section 3.1.2 is the Hamiltonian that the quantum annealer implements [8]. Quantum annealing proceeds as follows. The quantum annealing system is initially driven by the transverse magnetic field H_x prepared in its ground state $(1, \cdots, 1)'$, and then we slowly drive the system from the initial Hamiltonian H_x to its final target Hamiltonian H_q. From Equation (15), $[H_D(t) = A(t)\, H_x + B(t)\, H_q]$, it is evident that at $t = 0$, $H_D(0) = A(0)\, H_x$, and at $t = t_f$, $H_D(t_f) = B(t_f)\, H_q$.

The adiabatic quantum theorem shows that during the evolution of quantum annealing, the system tends to stay in the ground states of the instantaneous Hamiltonian via quantum tunnelling. At the end of the annealing procedure, we measure the quantum system. With some probability the quantum system stays in a ground state of the final Hamiltonian H_q, and thus the measurement outcome renders a solution to the optimisation problem. That is, we can utilise the quantum annealing procedure given in Equation (15), to find the global minimum of the classical Hamiltonian H in Equation (1) and solve the original minimisation problem [7] [8].

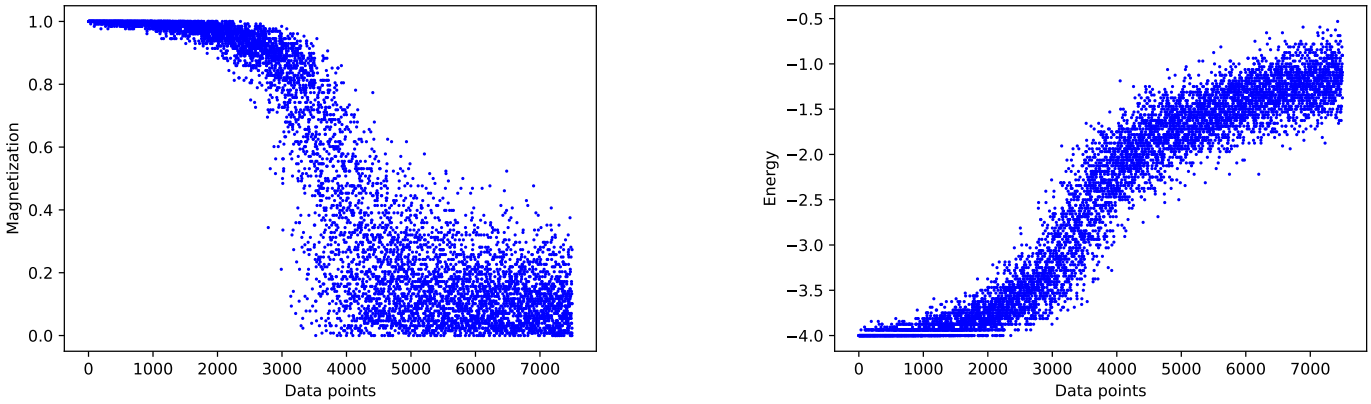
4 Results and Graphs

I have tried to estimate the three observables of the Ising system mentioned in the aim- namely the magnetisation, energy and specific heat- using an RBM. To do this, we first need to train the RBM; a collection of lattices with N spins were generated using the conventional MCMC method. Lattices were generated in a range of temperatures T with a step size ΔT, at each temperature we generate n independent lattices and use those lattices to train the Restricted Boltzmann Machines, one RBM for each temperature in the range. Details about the parameters used to generate the dataset to train the RBM are given in Table 1 below:

Parameter	Value
T_{min}	1.0
T_{max}	4.0
ΔT	0.1
N	16 x 16
n	250

Table 1: Parameters used to generate the dataset to train the RBMs.

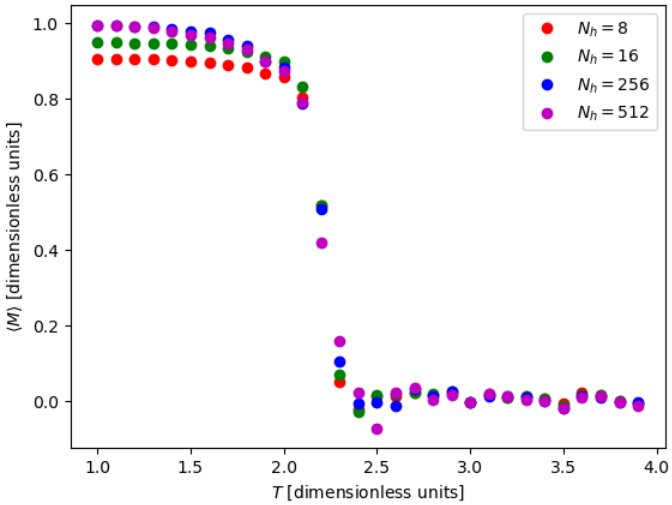
The following figures show a visualisation of this dataset. The spins are aligned in the low temperature region and random in the high temperature region.



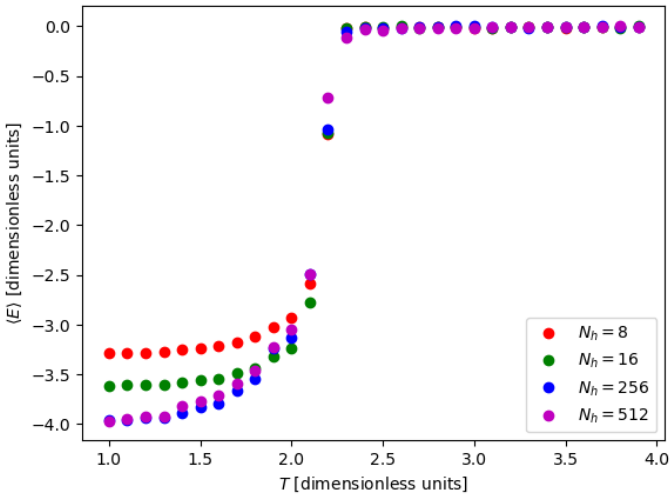
(a) (b)

Figure 4: Visualisation of the dataset used to train the RBMs, where (a) depicts the magnetisation per spin and (b) shows the energy per spin for each lattice in the dataset.

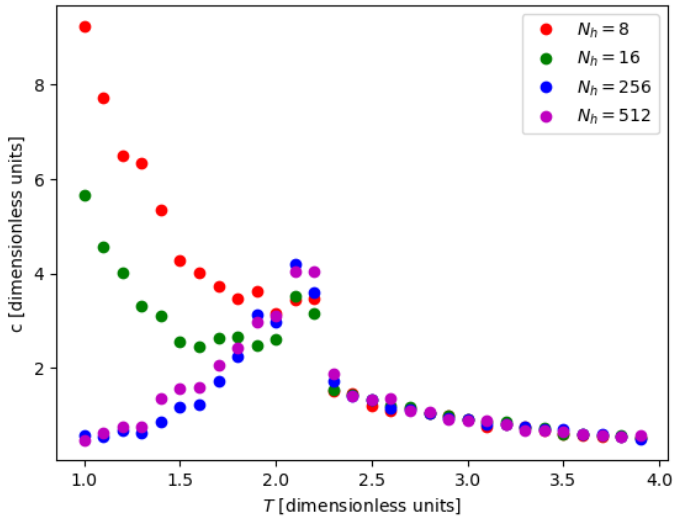
In order to train the RBMs we convert each of the lattices to vectors of length N and we change the values from -1, +1 to 0, 1 in order to make them compatible with the RBM structure. Each spin in the lattice is represented by 1 visible node in the RBM.
The main parameter we want to look at is the number of hidden nodes N_h in the network. The number of visible nodes N_v is defined by the number of spins in the lattice. The number of hidden nodes N_h can be varied.



(a)



(b)



(c)

Figure 5: The three observables calculated using the RBM, where (a) is the magnetisation per spin M, (b) is the energy per spin E, and (c) is the specific heat per spin C.

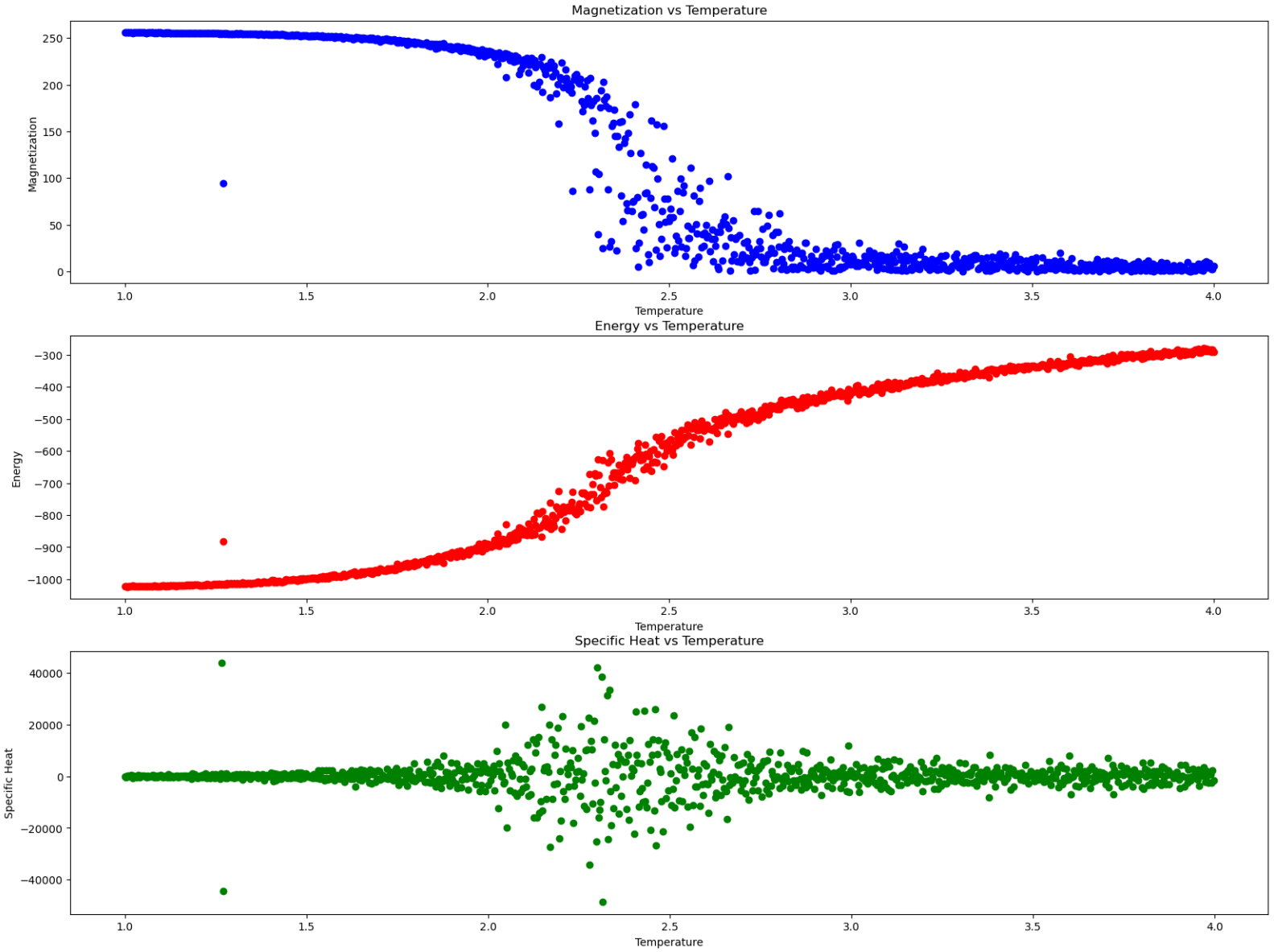


Figure 6: The magnetisation per spin M, energy per spin E and the specific heat per spin C as calculated using the MCMC method at a lattice size of 16×16 where only nearest neighbour contributions to the energy were used. These results resemble the analytical results less closely than the results of the RBMs.

5 Analysis and Conclusion

We see that the magnetisation per spin M calculated using the RBM, shown in Figure 5:(a) closely resembles the analytical graph depicted in Figure 2, which was obtained through the mean field approximation. The magnetisation tends to ± 1 at low temperatures and tends to 0 at high temperatures. There is a discontinuity around T_C.
The energy per spin E in Figure 5:(b) also agrees with the analytical result in Figure 2. For high temperatures it goes to 0 which is consistent with all spins being random. Around T_C it is has a discontinuity.
For specific heat C, in Figure 5:(c), we see that the RBM fails to model it correctly when the number of hidden nodes is lowered. It works however, for higher number of hidden nodes and agrees with the plot in Figure 2.

Figure 6 depicts the three observables of the Ising model calculated using Monte Carlo, but taking into account only nearest neighbour contributions. Comparing it with the data generated using the RBM in Figure 5, we can conclude that the data from the RBM more closely resembles the analytical results in Figure 2.

In this paper, we discussed the classical and quantum Ising models and using an RBM to calculate the three observables of the Ising model. We then compared those results to standard MC data and found that the data generated by an RBM is superior, in that it more closely resembles the analytical data. The final section of the paper dealt with Quantum Annealers and their application in training Boltzmann Machines, albeit briefly. Delving deeper into the subject of quantum annealing is beyond the scope of this paper as it requires access to quantum hardware and circuits, but let us try to discuss a bit more. A Quantum Boltzmann Machine (QBM) is a machine learning algorithm similar to a classical BM or RBM, the difference being that the former uses fundamental properties of quantum mechanics like entanglement and qubit (superposition of $|0\rangle$ and $|1\rangle$) to carry out calculations more efficiently. The Hamiltonian of a QBM is described in Equation (15). The aim of a QBM much like the RBM discussed in Section 3.2, is to maximise the Log-Likelihood function or minimise the negative Log-Likelihood function. Here, we use the density matrix formalism to calculate the aforementioned function. According to [10], in small-size examples a QBM learned the data distribution better than a BM (See [11] [12] [13] [1] [14] for further reading).

References

[1] Vivek Dixit, Raja Selvarajan, Tamer Aldwairi, Yaroslav Koshka, Mark A Novotny, Travis S Humble, Muhammad A Alam, and Sabre Kais. Training a quantum annealing based restricted boltzmann machine on cybersecurity data. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):417–428, 2021.

[2] Jos Thijssen. *Computational physics*. Cambridge university press, 2007.

[3] Ising model. `https://en.wikipedia.org/wiki/Ising_model`. [Online; accessed 20-August-2023].

[4] Ludwig Ridderstolpe. Exact solutions of the ising model. `https://www.diva-portal.org/smash/get/diva2:1139470/FULLTEXT01.pdf`, 2017.

[5] Richard fitzpatrick - the ising model. `http://farside.ph.utexas.edu/teaching/329/lectures/`. [Online; accessed 17-August-2023].

[6] L Onsager. A two-dimensional model with an order-disorder transition (crystal statistics i). *Phys. Rev*, 65(3-4):117–149, 1944.

[7] Yazhen Wang, Shang Wu, and Jian Zou. Quantum annealing with markov chain monte carlo simulations and d-wave quantum computers. *Statistical Science*, 31(3):362–398, 2016.

[8] Yazhen Wang, Shang Wu, and Jian Zou. Quantum annealing with markov chain monte carlo simulations and d-wave quantum computers. *Statistical Science*, pages 362–398, 2016.

[9] edureka! Restricted boltzmann machine — neural network tutorial — deep learning tutorial — edureka. `https://www.youtube.com/watch?v=i64KpxyaLpo`.

[10] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Phys. Rev. X*, 8:021050, May 2018.

[11] Steven H Adachi and Maxwell P Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.

[12] A Ramezanpour. Enhancing the efficiency of quantum annealing via reinforcement: A path-integral monte carlo simulation of the quantum reinforcement algorithm. *Physical Review A*, 98(6):062309, 2018.

[13] Sergei V Isakov, Guglielmo Mazzola, Vadim N Smelyanskiy, Zhang Jiang, Sergio Boixo, Hartmut Neven, and Matthias Troyer. Understanding quantum tunneling through quantum monte carlo simulations. *Physical review letters*, 117(18):180402, 2016.

[14] Siddhartha Srivastava and Veera Sundararaghavan. Generative and discriminative training of boltzmann machine through quantum annealing. *Scientific Reports*, 13(1):7889, 2023.