Project report for 6CCP313
Third year project in physics
Spring 2021, London

Machine Learning in Quantum Mechanics

Samuel Heczko
Under the supervision of Dr. Joe Bhaseen

I declare that this report is my own work and is not plagiarised. Also, I have not assisted another student in plagiarism.

KING'S College LONDON

Abstract

In this report, the machine learning approach to many-body quantum physics is investigated. First in an analytical, the report gives an pedagogical overview of the most critical concepts in the study of Artificial Neural Networks and then proceeds to demonstrate the usefulness of such construction when applied to a quantum many-body system. Then, in a numerical part, a Restricted Boltzmann Machine is used to produce a representation of a many-body system described by a Transverse Field Ising model. The Variational Monte Carlo optimised weight matrix of the network is examined using Singular Value Decomposition to examine the relation between phase transitions of the model and the algorithm parameters.

Table of Contents

1. Introduction

Arguably, Artificial Intelligence (AI) has been the most transformative technology of the last ten years. This technology underpins some of the most commercially potent novel inventions such as social media, self-driving cars and facial recognition. In physics, the AI 'revolution' commenced in 2017 when Carleo and Troyer[1] showed that *neural networks* produce state-of-the-art accuracy estimating the ground state energy of many-body quantum systems. The field has developed rapidly, and the essential present-day applications of neural networks in condensed matter physics include modelling wave functions based on experiment[2], finding ground states of many-body systems using deep networks[3] and classifying the phases of matter[4]. An extensive and up-to-date list is given in reference[5].

In quantum physics, the main advantage of the neural network approach is evident when compared to older numerical solution methods such as Density Matrix Renormalisation Group (DMRG) and Quantum Monte Carlo. Using the traditional methods, the trouble known as the 'curse of dimensionality' arises because any quantum mechanical state is represented as elements in Hilbert space. To have a complete representation of the wave function Ψ, we need to store the probability amplitudes of a complete basis set, which for the many-body system means the amplitude per every *combination* of possible configurations. As every particle introduces new degrees of freedom into the Hilbert space, the amount of possible states grows exponentially with each particle. Imagine an array of binary variables, such as spins. Here, the number of amplitudes scales by 2^N, where N is the number of particles in the system. Thus, to store a precise snapshot of a system of 40 particles would require the presentation of 2^{40} parameters as a complex floating number. It would take 10 TB and for real-life systems with billions of atoms, any physical storage space will soon prove insufficient.

In the Artificial Neural Network (ANN) approach, we construct a function that accurately represents the system itself and from which the amplitudes of various configurations can be effectively extracted. This allows us to extend the model as effortlessly as we expand physical systems: simply by adding identical components. From a computational perspective, the exponential scaling turns into a polynomial one and, as this report shows, allows for extracting reasonable estimates of various properties of many-body systems. Increasing the system size of possible simulation gets us closer to modelling states of real life-sized systems. Such knowledge is helpful for quantum computer design, where spins are taken as the fundamental computational components and is linked to ultracold atom simulations, which in turn underpin the study of superconductivity.

In this report, a complex network Restricted Boltzmann Machine (RBM) optimised to ground state using Variational Monte Carlo (VMC)[6], will be explored in detail. The network attributes, namely the precision and computational efficiency, are well known but dubbed as 'black-box' technology little is understood how the parameters of the neural network specifically encode the information about the quantum system in question. Here

we attempt to partially illuminate the box by analytical and numerical methods building an understanding of how and in which parts of the machine the optimisation happens. First, in the analytical part, we give all the mathematical tools and an overview of the pre-requisites to understand the network and to pose meaningful questions. Then, in the numerical part, we build a neural quantum state on our own using NetKet[7] framework and show that the energy estimations produced correctly predict a phase transition for a system described by the Transverse Field Ising model. Then, we proceed to study the values of the network weights for the neural representation using Singular Value Decomposition techniques. Finally, we argue that analysis of the weight matrix itself can be used to extract, at least in our limited case, valuable knowledge of the system's physical state.

2. Analytical part

2.1 General overview

Any *machine learning* application always consist of two parts: the *machine* and *learning*.

The *machine* is the particular algorithm used in the process. Here, it refers to *Artifical Neural Networks* (ANN). Different network structures are used to satisfy the purposes of a problem on hand.

Learning is a process of calibrating or optimising such a machine using some provided data or otherwise. In general, the more we train the network, the better functionality is achieved.

2.2 The neural network

In the most general way, neural networks are based on the Kolmogorov-Arnold theorem (KAT), which states that any continuous multivariate function can be written exactly as a sum of functions of a one variable[8].

$$f(\mathbf{x}) = f(x_1,\ldots,x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \lambda_p \phi_{q,p}(x_p - \nu q) \right). \tag{1}$$

In this equation Φ and ϕ are one dimensional functions and $0 \leq \phi(y) \leq 1$, with $0 \leq y \leq 1$ and λ and ν are left to be determined. In general, neural networks are in the business of mimicking multidimensional functions as a superposition of one dimensional many variable functions. The KAT states that regardless of the complexity of the original function in principle, two simpler functions can be composed to portray the multidimensional behaviour fully. It guarantees that strangely specific functions exist: for example, there is a function that will take in a picture of a face as an input and return a probability of ones aunt being in it.

To design such a nested function as in Eq.(1), we introduce the notion of an artificial neural network (ANN). In this model, loosely based on the human brain, there are multiple layers

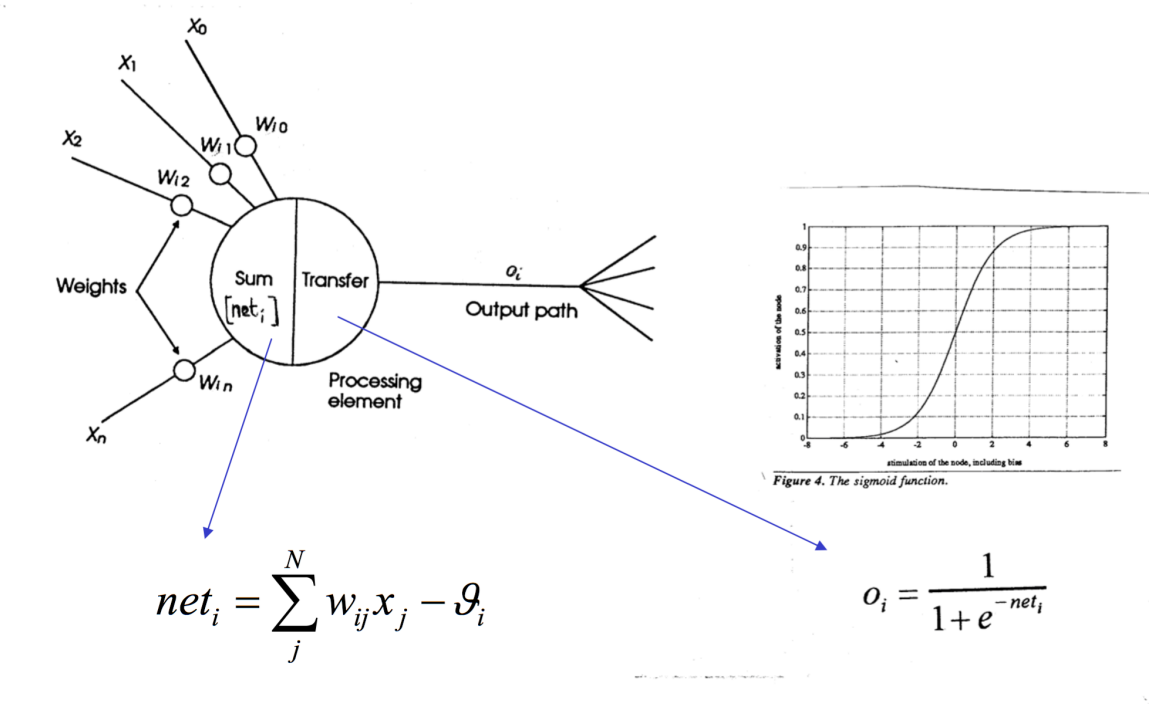


Fig. 1. A hidden neuron, the fundamental building block of neural network and a plot non-linear sigmoid activation function. In practice, the decision if a neuron is activated is made by randomly drawing a number between 0 and 1. Then its value is compared with the activation function value. (picture on courtesy of towardsdatascience.com/)

of fundamental units *neurons*. The first layer is called the *visible* layer consisting of visible neurons in general taking any value, for example the luminosity of a pixel. The deeper layers consist of hidden neurons with a binary value which are associated with an *activation functions*, *weights* $\boldsymbol{w}$ and a *biases* b. The combined vector for all the network parameters is called $\boldsymbol{p}$. The raw inputted signal gives values to the *visible* neurons from which the value of first layer *hidden* neuron is calculated using a sigmoid activation function:

$$\phi = \frac{1}{1+e^{-\varepsilon}} \tag{2}$$

which gives a probability of a hidden neuron being activated. The ε (often called energy) in the Eq.(2) is calculated using the value of visible neurons $\boldsymbol{x}$ and the weight and bias values:

$$P_j = \phi\left(\sum_i w_{ij} x_i + b_j\right) \tag{3}$$

where w_{ij} are the values of the weight matrix, x_i are the visible layer values and b_j stand for the bias values. The variables in $\boldsymbol{w}$ can be viewed to determine the strength of interactions of neurons in different layers and need to be determined in the learning process. A two-level construction is represented as a graph of connected neurons as shown in Fig. 2. A

detailed diagram of a single neuron with a plot of activation function is given in Fig. 1. In general, there can be more layers in the network. Then a new set of neurons is created with weights and biases and but the inputs are taken as the hidden neuron values. A neural network is then a way to build a nested function much like the KAT in Eq.(1) and can represent any high-dimensional function[9].

The amount of layers and hidden neurons gives the overall complexity of the function and therefore, its readiness to model complicated relations between inputs and outputs. Adding components adds to the number of parameters required to be optimised and introduces a trade-off between time and data needed for training and subtlety of the model. In this report, only two-layer networks will be considered.

For quantum physics, the introduction of a neural network means that one does not require the knowledge of the amplitude of every possible state in the Hilbert space. Instead, the values of weights and biases are stored from which the probability of a particular combination can be calculated. The wave function in this representation is seen as a 'black box', taking in configurations and giving amplitudes.

To demonstrate the usefulness of such compression, let's imagine a two-layer network commonly used in many-body physics. To store perfectly a state of 20 spins, one needs 2^{20} stored numbers. In contrast, a network used in this report can model a spin system accurately with only 20×40 weight values. Even for such a small system, the ratio of required storage space is staggering 1311.

2.3 Supervised learning

As we have seen, the neural networks require a large number of parameters $\boldsymbol{p}$ to be optimised. The conceptually simplest approach to do so is *supervised learning*. In this method, data with pre-existing information of the desired output needs to be provided. The most famous example, the MNIST dataset, contains images of handwritten digits and the value the image represents. In the jargon, the data is *labeled*. The training data is fed into the machine, and the output is compared with the label, quantified by a *loss function*:

$$\mathscr{L} = \frac{1}{N_s} \sum_{i}^{N_s} |F(\boldsymbol{x}_i; \boldsymbol{p}) - y_i| \tag{4}$$

here y_i is the label of the data, F is the value suggested by the machine and N_s is the number of samples. The loss function has a minimum value of 0 at a perfect reconstruction. To minimise the derivative *stochastic gradient descent* (SGD) is used. This method will be explored in detail in the next section. After the training process has been completed, the network will be able to represent the very complicated function based on the discreet input is has been given. The algorithm is now ready to analyse data it hasn't 'seen' before, for instance, recognising handwritten digits by a new subject. In this setting, low network parameters are desired as they are associated with 'generalisation' property[10]. This property tells how well the network can classify new data.

In physics, such networks have been used to recognise phases of matter from various "raw" data sets such as images from various devices[4].

2.4 Stochastic gradient descent

Here we establish a basic understanding of the SDG optimisation method using the supervised learning example introduced in the previous section. The idea is based on the standard gradient descent, where parameters are adjusted by going "down" the gradient hill G (illustrated in Fig. 2a:

$$\boldsymbol{p}^{(s+1)} = \boldsymbol{p}^{(s)} - \eta G(\boldsymbol{p}^{(s)}) \tag{5}$$

The update rule is based on a numerically calculated derivative G of the loss function:

$$G = \nabla_p \mathscr{L}(\boldsymbol{p}) = \frac{1}{N}\sum_i^N \nabla_p |F(\boldsymbol{x}_i; W) - y_i| \tag{6}$$

where N is the batch size, determined by how many data points from the data set are used. In the standard gradient descent, this derivative is taken exactly using all the data available. It poses two challenges: a complicated function will easily get stuck in the local minima (Figure 2a), and it is difficult, if not impossible, to sum over all the training data. In the stochastic version, samples are randomly drawn from the data such that the exact gradient is not achieved. This will introduce noise to the gradient and can be approximated the exact gradient plus some Gaussian noise with normal variance $\sigma \propto \frac{1}{\sqrt{N_s}}$:

$$G_k^{SGD} = \nabla \mathscr{L}(\boldsymbol{p}) + Normal(\sigma) \tag{7}$$

When substituted into Eq.(6) this yields:

$$p_k^{(s+1)} = p_k^{(s)} - \eta(\partial_{pk} \mathscr{L}(\boldsymbol{p}) + Normal(\frac{1}{\sqrt{N_s}}) \tag{8}$$

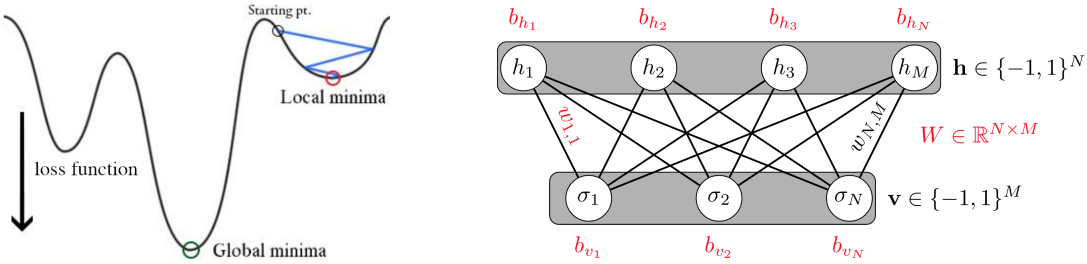
noticing the similarity to the physical Langevin equation:

$$p_k^{(s+1)} = p_k^{(s)} - \delta_t(\partial_{pk} \mathscr{L}(W) + Normal(\sqrt{2\delta T})) \tag{9}$$

where the T is the effective temperature, and δ is the small time step. This equation describes a physical system that goes to the global minimum when T is annealed to 0. Comparing Eq.(9) with Eq.(8) we see that:

$$T \propto \frac{\eta}{N_s} \tag{10}$$

The η is a small parameter called learning rate and needs to be simulated smaller such that the temperature goes to zero, resulting in the system approaching a global minimum. Another approach to get the temperature to drop would be to make the batch size larger.



(a) **(b)**

Fig. 2. (a) Exactly following the gradient might lead into local minima. This is why the stochastic method is preferred. (b) Graph of a Restricted Boltzmann machine. It's important to notice that the visible and hidden layers have no intralayer connections allowing simple extraction of conditional probabilities.

There are two major pros of using stochastic gradient descent instead of the classical. Using fewer samples is computationally faster as the gradient can be calculated using only parts of the data set. Besides, by adding noise, the gradient calculation becomes imperfect, making unfavourable moves in the energy landscape plausible. Thus there is no need to set up a complex scheme to have a probability of accepting unfavourable moves in the potential landscape. Yet as our simulations later show it is not perfect. The introduced randomness in the process makes it *stochastic*.

2.5 Restricted Boltzmann machines

For many applications, including many-body physics, the configuration of input of the network is binary, allowing us to simplify the network. Now we have both the visible and hidden neurons associated with binary values and with activation functions. This leads us to an ANN architecture called *Restricted Boltzmann machine* (RBM). RBM:s have only two layers of neurons, a visible layer with M values $\boldsymbol{\sigma}$ and a hidden layer with N values $\boldsymbol{h}$. The neurons have only vertices with neurons that are in a different level. Fig. 2 illustrates such a network. In quantum physics, σ_i and h_j can take only values $1, -1$, corresponding to a component of the spin to be up or down in a quantum system.

The importance of this construction for physics lies in theorems similar to KAT, stating that it is capable of reproducing any continuous physical state of matter[3]. Another critical observation is that by changing the number of hidden neurons, the accuracy of the model can be adjusted easily[1].

Recently, this architecture has been increasingly replaced by deeper networks that offer more precise results[11]. However, the RBM remains the most standard example, a prerequisite for understanding the more advanced constructions.

The RBM takes in the configuration of hidden and visible spins and returns probability density given by:

$$F_{rbm}(\sigma_1, \sigma_2, ...\sigma_M, h_1, h_2, ...h_N) = e^{\sum_j a_j \sigma_j^z + \sum_i b_i h_i + \sum_{ij} W_{ij} h_i \sigma_j^z}, \tag{11}$$

The values of the visible neurons are the input of the algorithm, but the values of the hidden neurons are unknown. To get a probability contribution of a state $\boldsymbol{\sigma}$ we need to sum over all the possibilities for hidden neurons:

$$F_{rbm}(\sigma_1, \sigma_2, ...\sigma_N) = \sum_{h=\pm 1} e^{\sum_j a_j \sigma_j^z + \sum_i b_i h_i + \sum_{ij} W_{ij} h_i \sigma_j^z} \tag{12}$$

This can be traced over h to give:

$$F_{rbm}(\sigma_1, \sigma_2, ...\sigma_N) = e^{\sum_j \sigma_j a_j} \times \Pi_j 2cosh\Big[\sum_i W_{ij}\sigma_i + b_j\Big] \tag{13}$$

Here, F represents a probability distribution over the possible combinations of $\boldsymbol{h}$ given the state $\boldsymbol{\sigma}$. In quantum mechanics, $\boldsymbol{\sigma}$ corresponds to the combination of basis vectors of the state. Given a certain combination of σ_i, the network returns a probability of the state from which the wave function amplitude can be calculated. The goal of the training process is to end up with network parameters for a specific Hamiltonian that the returned probabilities correspond to physical reality.
To normalise the probability, we divide by a partition function Z, the sum of all possible combinations of $\boldsymbol{\sigma}$. The sum over all the configurations is intractable as it scales exponentially with the system size and suffers from the same problem as the original, exact representation of quantum states.
Finally, for quantum mechanics, we make an analogy with the amplitude of a wave function as a square root probability such that:

$$\Psi_p = \sqrt{\frac{F_p(\boldsymbol{\sigma})}{Z_p}} \tag{14}$$

We conclude with the note that knowing the probability distribution of the RBM is analogous to knowing the entire state of a wave function it is representing.

2.6 Sampling

From the RBM, it's simple to calculate the probabilities of different combinations of inputs from Eq.(13), but calculating over all possible combinations is computationally too demanding and often impossible. A sampling scheme needs to be established. The goal is to propose a set of vectors $\boldsymbol{x}$ of h and $\boldsymbol{\sigma}$ such that they are distributed according to Eq. (11). It can be reached using Monte Carlo Markov Chain implemented on the RBM with the Gibbs sampling algorithm. Gibbs sampling is used when direct sampling from the distribution is difficult, but the conditional probabilities are achievable. The simplicity of the conditional

probability is another reason why RBM:s are widespread. In the procedure, two steps are performed:
1. Finding the set of hidden variables given σ
2. Finding the set of σ using the set of hidden variables just generated
Taking the probability of one hidden neuron, for example, we have Eq.(13), the probability summed over combinations of $\boldsymbol{h}$ and Eq.(11), probability when all the parameters are known. Conditional probability for some h, given $\boldsymbol{\sigma}$ is then:

$$P(h|\sigma) = \frac{e^{\sum_j \sigma_j a_j} \times e^{\sum_i b_i h_i + \sum_{ij} W_{ij} h_i \sigma_j^z}}{e^{\sum_i \sigma_i a_j} \times \prod_j 2cosh\Big[\sum_i W_{ij}\sigma_i + b_j\Big]} \tag{15}$$

For instance, the probability for $h_j = -1$:

$$P(h_i = -1|\sigma) = \frac{e^{\sum_j \sigma_j a_j} \times e^{-\sum_i b_i - \sum_{ij} W_{ij} \sigma_j^z}}{e^{\sum_i \sigma_i a_j} \times \prod_j 2cosh\Big[\sum_i W_{ij}\sigma_i + b_j\Big]} = \frac{1}{1+e^{2\theta}} \tag{16}$$

Where $\theta = \sum_i W_{ij}\sigma_i + b_j$. Given that in the model $h_j = -1$ corresponds to the probability of a neuron not being activated, it's not surprising that we have successfully recovered the inverse of the activation function $(1 - \frac{1}{1+e^{-x}})$ of a single neuron discussed in Section 2.2.
Then, using the hidden spins obtained, visible spins are chosen similarly, creating the first sample. Repeating the procedure, N times N samples are drawn from the machine. A long chain of samples is created whose distribution converges to the probability distribution of our machine[1]. In the Variational Monte Carlo (VMC) optimisation method used in physics, the first set of σs needs to be chosen randomly. Usually, the first couple of samples are discarded. This is not the case for unsupervised learning where a set of samples from a probability distribution is given: then one can start the sampling from a state included in the distribution speeding up the convergence. Now, even with one iteration, precise probability density estimates can be achieved. This method is called *contrastive divergence*[12]. The fact that probability density obtained using this sampling method converges to the full probability density of the machine is one of the theorems that make the whole field of machine learning possible[13].

2.7 RBMs in quantum mechanics

To modify the described RBM to be useful in quantum physics, the weights and biases are taken complex. The analysis done in the previous sections stays the same, however now the returned F Eq.(13) from the system is complex, encoding both the amplitude and phase of the wave function Ψ.

2.8 Variational Monte Carlo

To train a network to represent a ground state of Ψ we use the Variational Monte Carlo (VMC) method[6]. A more specific description is given in Appendix A. The method is based on the same idea as supervised learning, but instead the loss function is now taken to be the energy of the wave function. Having an externally given function for optimising the network is called *reward learning*. As the ground state is the lowest possible energy level, we will find the ground state wave function. The update rule used is:

$$p_k^{i+1} = p_k^i - \eta \partial_{p_k} \langle H \rangle \tag{17}$$

which can be shown to be easy to calculate from the correctly sampled RBM distribution. As before η is the learning rate of the system, and similarly to the supervised case taking the derivative using sampled distribution will lead to a noisy gradient. The method is therefore stochastic, making the system fast and reliable to find the global minima. After the optimisation, the RBM will correctly encapsulate the wave function and allowing for the ground state energy to be read form it.

In a more advanced optimisation procedure, instead of updating the parameters with a pure gradient it is weighted by a *stochastic reconfiguration* (SR) matrix. The matrix could be seen as a second-order correction to the gradient estimation, which makes the learning faster, but is not strictly essential. Its attributes in connection to the physical model have been studied extensively[10], but for our purposes, it is crucial to notice that it doesn't affect the energy valuation of the network but speeds up the optimisation. The energy convergence of the network using VMC without SR is given in Fig. 3d.

3. Numerical part

3.1 Transverse field Ising model

In this part we will examine the functionality and attributes of the neural networks in the study of 1D Transverse Field Ising model (TFI). TFI is a chain of quantum spins described by a Hamiltonian:

$$H_{TFI} = -\sum_{ij} \sigma_i^z \sigma_j^z - H \sum_i \sigma x_i \tag{18}$$

where σs are the Pauli spin matrices and the summation over i and j stands for the sum over nearest neighbours. H is the ratio of the strength of the transverse compared to the interaction between spins. Conceptually this is a lattice of spins where they interact with nearest neighbours, trying to align with each other in the z-direction. An applied transverse magnetic field will interact to counter this effect and align the spins in the x-direction. In this setting, the spin matrices in different directions don't commute and the alignment in the x-direction corresponds to a random ordering in the z-direction. It has been shown that this quantum phase transition happens at the definitive critical magnetic field $H = 1$. There are two phases; a disordered phase where the transverse applied field dominates, making the spins align randomly in the z-direction and an ordered phase where the spin-spin interaction

dominates, causing the spins to align with each other. As there are two configurations of perfectly aligned spins, all up or all down the system has $\mathbb{Z}_2$ symmetry in the ordered phase. The ordered phase is ferromagnetic and the disordered phase is paramagnetic.

The quantum phase transition of the system is a second-order transition, meaning that the second derivative of the energy experiences discontinuity. As the parameter modified is the transverse field, the discontinuity will be in a derivative with respect to it. It is shown in Appendix B how the negative of the derivative is the magnetisation so the discontinuity appears in the magnetic susceptibility.

3.2 Proving the RBM hypothesis

To empirically test the machine learning approach in quantum mechanics, we train an RBM with 20 visible and 40 hidden spins for 1000 optimisation rounds. Using the VMC optimisation method and SDG parameter update rule, we find the ground state energy for the the chain with H between 0 and 3 with a 0.1 offset. This optimisation was run on a neural network without biases to force the algorithm to store all the information about the system into the weight matrix. The VMC training system and the neural network were build using the NetKet neural quantum state framework[7].
We plot the results, and in Fig. 3, it can be observed that there is a discontinuity in the second-order derivative showing that the system has a second-order phase transition. The relative error between the ground states found by the ANN and exact values calculated by Lanczos exact diagonalisation is on average 0.01. Due to the finite system effect, the phase transition is not perfect as strictly speaking the phase transition requires an infinite system. We also find that when the probability for all the states is plotted, the network shows correctly the doubly degenerate ground state, therefore encapsulating the $\mathbb{Z}_2$ symmetry.

Furthermore, 3d shows that typically we need about 300 rounds of training for the model to find the ground state and (a,b,c) and for all cases the network ground state estimations are accurate. This qualifies our network parameters to be fit for the study of phase transitions.

3.3 Analysis of the weight matrix

Previously, we have shown that the RBM representation can find accurately the location of a quantum phase transition and the $\mathbb{Z}_2$ symmetry of the system using the 'vanilla' gradient descent VMC method. Here we examine what can be said about the physical system by the study of the weight matrix alone. Most importantly we want to examine is there a quantifiable discontinuity in some attribute of the weight matrix and can we use the analysis to find the critical field where the phase transition takes place. It is suggested that as the network depicts the energy of the system precisely, we should be able to find the direct connection between the weight matrix and the physics of the model. This hypothesis is similar to idea that the layers of a convolutional neural network show a different level of pattern recognition in image recognition applications[14]. A study with partly similar

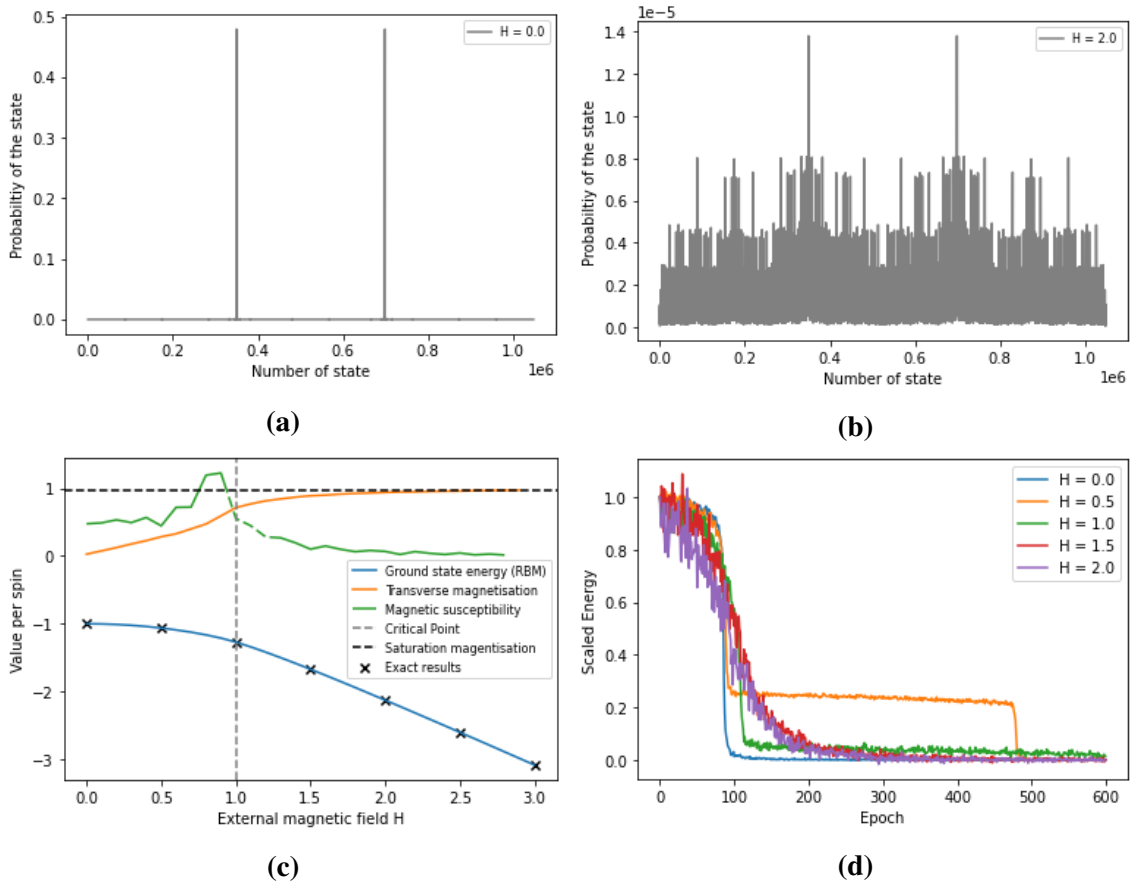


Fig. 3. (a),(b) Full probability distribution taken from the RBM after 1000 epochs. In (a) the RBM predicts the double degenerate ground state at $H = 0$ and in (b) random distribution above critical field at H = 2. Note that in general, this analysis is not possible due to the intractable amount of states. For the 20 spin system there are 10^6 possible states, plotted on the x-axis. (d) Plot showing the RBM estimated ground state energies with interval H = 0.1 and numerically obtained first and and second derivatives. When the magnetisation is positive in z, the system is paramagentic, meaning it gets magnetised with an external field. (e) The scaled energy convergence for different values of H and learning rate 0.01. Here 0 corresponds to the exact ground state energy and 1 to the initial random estimate. The kink in H = 0.5 shows the biggest weakness of the vanilla SGD, namely its proness to get stuck in the local energy minima.

ambitions was done by Park and Kastoryano in 2020[10] but not employing methods used here.

3.3.1 Methods

We analysed several properties of the weight matrix taken from the machine after training. Each value in the weight matrix is complex, and each step of the analysis was done separately for complex and real parts of the weight matrix. First, the simple average of

the absolute value of the W matrix is taken. Then the matrix is decomposed using SVD (Appendix C) to find the spectrum of the matrix. The obtained *singular values* (SR), corresponding to eigenvalues of a rectangular matrix, are scaled to compare the relative distribution between high and low values for each matrix. The spectrum of a matrix corresponds to the distribution of luminance for vector pairs each corresponding to a feature element. Large SVs correspond to a big distinctive features in the matrix and small elements present everywhere in the matrix[15] such as noise.

3.3.2 Results

The results are summarised in the Fig. 4. From the plots it can be read that average weight value reaches the highest value before the critical field, the average SV reaches maximum at the critical field. Both averages decay in a roughly parabolic fashion after the critical field. Considering the averages, imaginary values follow closely the real ones and for extracting the information from the system we can can restrict ourselves to the trends seen in both domains. Noticeable difference in trends in for the first value of average weight. Here the imaginary value is lowest of the measured situations but the in real domain it is the highest. However, we notice that for H=0 the spectra is linearly decaying corresponding to pure noise matrix meaning that the training left its features unaltered only lowered. The spectra above the critical field are almost identical both for imaginary and real values, whereas below it they feature a noticeable kink in the real domain. Neither critical spectra show any behaviour that could be simply quantified but in the real domain the kink follows a trend to get smaller with higher H.

Additionally, we show the heat map of the real part of the matrices in a Figure 5. It shows a checked pattern in the ferromagnetic phase which disappears in the paramagnetic phase.

3.3.3 Discussion

The spectrum of a matrix tells about its periodicity. Distribution dominated by higher values indicates less periodicity therefore separate chunks in the analysed matrix. The spectra in Fig. 5b shows that the matrices of weights for both imaginary and real values appear to be less uniformly ordered with increasing field. This corresponds to physical reality where spins become less correlated in the x direction with an increased transverse field. Thus the amount of interaction of the system is associated with the rate of decay of the SVs. Sometimes in a literature the decay rate is called the ”roughness” of a matrix[15]. The decrease in periodicity of the weight matrix can also be observed qualitatively by looking at the heat map of the weight matrix in Fig. 5c.

However, by examination of the spectra alone we can hardly make any conclusions about the exact location of the phase transition. In the imaginary domain all the values above $H = 1$ have an essentially identical spectrum, whereas having a look at the real values, it is not perfectly clear where the phase transition takes place. The most promising way of

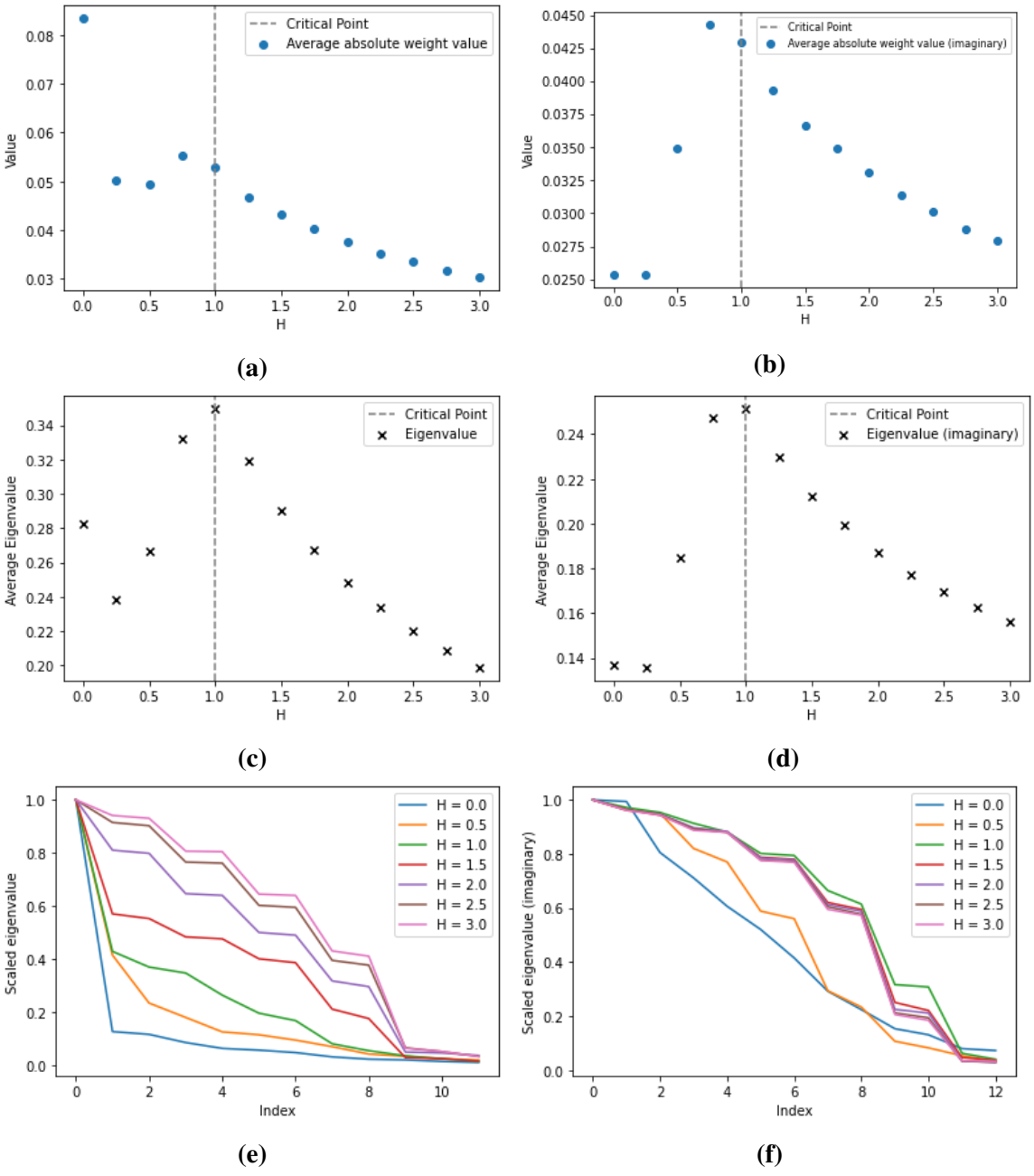


Fig. 4. (a) The average absolute real w value taken from the RBM after 1000 epochs of training. (c) The average singular value of the real weight matrix (e) The scaled spectra of the matrix trained on different H. (b),(d),(f) same analysis as on the left for the imaginary w matrix.

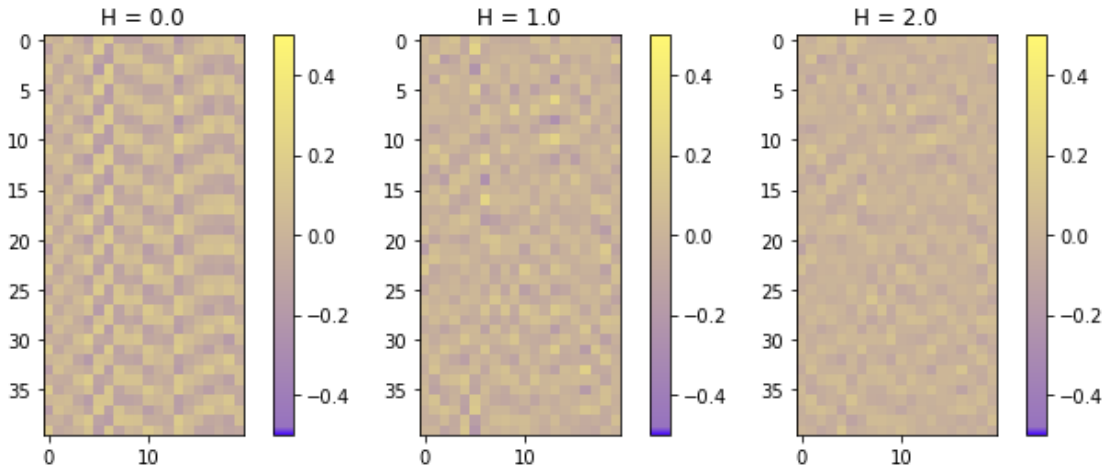


Fig. 5. (a)-(c): Heat map of the weight matrix with different h values. The chess board like pattern disappears with the interaction between spins

identifying the critical field from the weight matrix seems to average SV value, given that in our model it peaks exactly in the critical point. Each SV of the system corresponds to a luminosity of a certain structure; a high average means the presence of many structures. At the critical point, some structures of both phases are present and the matrix needs to capture both. In Fig 3d we see that the critical field the machine converges to the ground state slightly worse than for the other values of H. This supports the idea that the critical point matrix is more complicated for the machine. Above the critical point the SV:s start to decay uniformly and this could also be used to identify the exact location of phase transitions. Furthermore, the values of the imaginary matrix could potentially be used. In a strongly ferromagnetic phase the system does not have phase encoded to the wave function, and the imaginary weight values remain random and low. Nearing the critical phase gets more important reflected in the higher values of the imaginary matrix. But with the the average w value peaking before the critical point, the analysis of the average w can't be seen very informative.

4. Conclusion

In this report, we have given a brief overview and a demonstration of the simplest cases of machine learning in quantum mechanics. In the analytical part, we explored supervised learning and more complicated VMC optimisation patterns. This analysis helps us to understand the central research questions posed in the numerical part.

Then we proved such methods work correctly predict ground state energy of a quantum many-body system, namely about a Transverse Field Ising Model system of 20 spins. The

clearest success of the report was to show that the machine learning approach can accurately depict the ground state of a 1D TFI wave function with the most simple settings: the learning was done using simple SGD, we used only a two-layer network with only two hidden neurons per one visible. Then we analysed the weight matrix using methods used in image manipulation and concluded that using these we get an insight into the physical behaviour of the system. In our study of the 1D Transverse Field Ising model two main results were obtained:
1. Singular Value Decomposition applied on the weight matrix proved efficient in quantifying the complexity of the phase and the largest average Singular Value signified the critical point.
2. The roughness of the weight matrix a real matrix spectra correlates with the strong spin-spin interaction and therefore with the ferromagnetic phase.
3.The study of the weight values alone is not too helpful when analysing the properties of the physical system.
These results combined with the analytical study give an insight into the relationship between the network parameters and the physical system in question, thus satisfying the goal of the report.

Further study of these findings ought to focus on generalising the results for a diverse set of Hamiltonians and larger systems, including 2d-lattices with various shapes. It would be interesting to see if the results apply to a neural network trained using supervised and unsupervised learning. If the findings prove to hold for a large variety of set-ups, the analysis of the trained weight matrix could be used to predict phase transitions of novel materials and lattice structures.

5. Summary of the report for general audience

The main idea of the machine learning approach for the quantum many-body problem is to optimise parameters in a neural network such that the probability distribution of the network will correspond to the one of physical wave function. Redefining a quantum many-body system using a neural network reduces greatly the amount of storage space needed to have a faithful representation of the wave function.

It allows for storing wave functions for a larger amount of particles as the neural network scales up polynomially, whereas the classical representations scale up exponentially.

A two-layer neural network called Restricted Boltzmann Machine (RBM) is used to represent the many-body wave function task because just like physical many-body wave function $\Psi(\boldsymbol{\sigma})$ it takes in a binary spin configuration and returns probability. The probability density of the RBM is shown in Eq.(13) and by choosing the correct values $w_{i,j}$, b_j and b_i the probability density will correspond to the one of a wave function for a given state $\boldsymbol{\sigma}$. The advantage of RBM is that it allows for fast sampling. This means estimating the full probability density of the RBM without explicitly calculating it for every possible $\boldsymbol{\sigma}$ and therefore without the need to store every amplitude.

To find the correct parameters, that is to train such a network, we use a scheme called

Stochastic Gradient Descent. This is done by formulating a loss function, representing the quantity to be minimised with respect to the network parameters. The choice of a loss function depends on the available data and classifies the algorithm into supervised, unsupervised and reward learning. Taking the derivative of the loss function with respect to the network parameters will give a gradient, the direction towards the minima of the function as shown in Fig. 2a. The gradient is then multiplied by the constant called learning rate. After, the network parameters are updated by this quantity. To speed the calculation, the gradient of the neural network is not taken exactly but rather from a sampled distribution obtained by Gibbs Sampling. Introduced randomness makes the process *stochastic*.

To train a model in this project we use the Variational Monte Carlo optimisation, where the loss function is taken to be the Hamiltonian of the system. We minimise the energy of the chain and therefore find the ground state. The chief question asked in the numerical part was that does the physical phase transition lead to notable changes in the parameter space of the network itself. In this report we train the network parameters to find the ground state of Ψ_0 for the Transverse Field Ising model with 20 spins. This model features a chain of spins on which a transverse magnetic field is imposed. The system undergoes a quantum phase transition when transverse field of strength $H = 1$ is applied. The RBM is optimised for H between 0 and 3.0. To start an RBM trained and shown correctly find the ground state energies thus the phase transition and encode the probability distribution with $H = 0$ and $H = 3$.

Having a functional network we analysed the trained weight matrix. It is shown that that the weight matrix of the network encodes more features when the system approaches the critical transverse field and that a simple analysis of the average weight value does not tell us much about the physical system. In conclusion it can be said that from the weight matrix the qualitative behaviour of the system can be understood, but for acquiring quantitative results further study should be made. Further development if the project clearly includes a similar study with more models, larger systems and other types of learning to see if any of the results could be generalised.

References

[1] Carleo G, Troyer M (2017) Solving the quantum many-body problem with artificial neural networks. *Science* 355(6325):602–606. https://doi.org/10.1126/science.aag2302. https://science.sciencemag.org/content/355/6325/602.full.pdf Available at https://science.sciencemag.org/content/355/6325/602

[2] Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R, Carleo G (2018) Neural-network quantum state tomography. *Nature Physics* 14(5):447–450. https://doi.org/10.1038/s41567-018-0048-5. Available at https://doi.org/10.1038/s41567-018-0048-5

[3] Gao X, Duan LM (2017) Efficient representation of quantum many-body states with

deep neural networks. *Nature Communications* 8(1):662. https://doi.org/10.1038/s41467-017-00705-2. Available at https://doi.org/10.1038/s41467-017-00705-2

[4] Carrasquilla J, Melko RG (2017) Machine learning phases of matter. *Nature Physics* 13(5):431–434. https://doi.org/10.1038/nphys4035

[5] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L, Zdeborová L (2019) Machine learning and the physical sciences. *Rev Mod Phys* 91:045002. https://doi.org/10.1103/RevModPhys.91.045002. Available at https://link.aps.org/doi/10.1103/RevModPhys.91.045002

[6] Sorella S, Casula M, Rocca D (2007) Weak binding between two aromatic rings: Feeling the van der waals attraction by quantum monte carlo methods. *The Journal of Chemical Physics* 127(1):014105. https://doi.org/10.1063/1.2746035. https://doi.org/10.1063/1.2746035 Available at https://doi.org/10.1063/1.2746035

[7] Carleo G, Choo K, Hofmann D, Smith JET, Westerhout T, Alet F, Davis EJ, Efthymiou S, Glasser I, Lin SH, Mauri M, Mazzola G, Mendl CB, van Nieuwenburg E, O'Reilly O, Théveniaut H, Torlai G, Vicentini F, Wietek A (2019) Netket: A machine learning toolkit for many-body quantum systems. *SoftwareX* :100311https://doi.org/10.1016/j.softx.2019.100311. Available at http://www.sciencedirect.com/science/article/pii/S2352711019300974

[8] Sprecher DA (1965) On the structure of continuous functions of several variables. *Trans Amer Math Soc* 115:340–355. https://doi.org/https://doi.org/10.1090/S0002-9947-1965-0210852

[9] Carleo G Neural-network quantum states. Available at https://youtu.be/90OZoet_1CU.

[10] Park CY, Kastoryano MJ (2020) Geometry of learning neural quantum states. *Phys Rev Research* 2:023232. https://doi.org/10.1103/PhysRevResearch.2.023232. Available at https://link.aps.org/doi/10.1103/PhysRevResearch.2.023232

[11] Choo K, Neupert T, Carleo G (2019) Two-dimensional frustrated J_1-J_2 model studied with neural network quantum states. *Phys Rev B* 100:125124. https://doi.org/10.1103/PhysRevB.100.125124

[12] Fischer A, Igel C (2014) Training restricted boltzmann machines: An introduction. *Pattern Recognition* 47:25–39. https://doi.org/10.1016/j.patcog.2013.05.025

[13] Gelfand AE, Smith AFM (1990) Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85(410):398–409. https://doi.org/10.1080/01621459.1990.10476213. https://www.tandfonline.com/doi/pdf/10.1080/01621459.1990.10476213 Available at https://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10476213

[14] Traore BB, Kamsu-Foguem B, Tangara F (2018) Deep convolution neural network for image recognition. *Ecological Informatics* 48:257–268. https://doi.org/https://doi.org/10.1016/j.ecoinf.2018.10.002. Available at https://www.sciencedirect.com/science/article/pii/S1574954118302140

[15] Sadek RA (2012) Svd based image processing applications: State of the art, contributions and research challenges. *International Journal of Advanced Computer Science*

and Applications 3(7). https://doi.org/10.14569/IJACSA.2012.030703. Available at http://dx.doi.org/10.14569/IJACSA.2012.030703

[16] Griffiths DJ, Schroeter DF (2018) *Introduction to Quantum Mechanics* (Cambridge University Press), 3rd Ed. https://doi.org/10.1017/9781316995433

Appendix A: Variational Monte Carlo

We start by rewriting many body problem as optimisation problem, minimising well known variational energy. By definition, any energy formulated like this must be larger than the ground state energy functional[16]: $E(\psi) = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} > E_0$. We define new operator orthogonal Hamiltonian, often called local energy:

$$H_{loc}(x) = \sum_{x'} H_{xx'} \frac{\psi(x')}{\psi(x)} \tag{19}$$

Then using closure relation $\sum_x |x\rangle\langle x| = 1$ we can rewrite the energy:

$$E = \frac{\sum_{x,x'} \langle\psi|x\rangle\langle x|H_{loc}|x'\rangle\langle x'|\psi\rangle}{\sum_x \langle\psi|x\rangle\langle x|\psi\rangle} = \sum_x \frac{|\psi|^2 H_{loc}(x)}{\sum_x |\psi|^2} = \langle H_{loc}\rangle \tag{20}$$

which is the expectation value of the local Hamiltonian over the probability distribution of the RBM. Importantly, this can be extracted form the sampled distribution. To minimise the energy according to the variational principle we start by defining the derivative of the network with respect to the network parameters:

$$D_k(\boldsymbol{x}) = \frac{1}{\psi(\boldsymbol{x})} \partial_p \psi(\boldsymbol{x}) = \partial_p \ln(\psi) \tag{21}$$

and the derivative of the Hamiltonian is estimated using the method of covariances and can be show to be[1]:

$$\partial_{p_k} \langle H\rangle = \frac{d(E(\psi(\boldsymbol{x}, p_j))}{dp} = \langle D_k E_{loc}\rangle + \langle D_k\rangle\langle E_{loc}\rangle \tag{22}$$

Taking the basis vectors to be a combination of the spin variables shown in 13 and the probability relation to the wave funciton Eq.(14) and using the fact that $ln(z) = 0$ we can calculate the values variational derivatives D_k for our RBM to be:

$$D_{a_i} = \frac{1}{2}\sigma_i \tag{23}$$

$$D_{b_j} = \frac{1}{2}\tanh_j(\chi_j) \tag{24}$$

$$D_{w_{i,j}} = \frac{1}{2}\sigma_i \tanh(\chi_j) \tag{25}$$

Where $\chi_j = b_j + \sum_i w_{i,j}\sigma_i$ Now taking the derivative of the Hamiltonian as a loss function of the RBM we can finally write down the SGD update rule:

$$p_k^{i+1} = p_k^i - \eta \partial_{p_k} \langle H \rangle \tag{26}$$

Here η is the learning rate of the system. The main idea of VMC is that instead of calculating the exponential amount of sums for the partition function, statistical average over sampled distribution. Because of this, the optimisation can be performed efficiently.

Appendix B: Magnetisation as a derivative of energy

When changing the parameters of the system the change in the internal energy of the system is given by:

$$dE = TdS - MdH$$

Where H is the external field, M is its conjugate magnetisation S is entropy and T is temperature. Physically, this means we can add or subtract energy of the system by adding heat of doing magnetic work. Then we can then preform *Lagrange transform* by introducing Free energy F:

$$F = E - TS$$

Taking the infinitesimal change and using the definition of dE we see that:

$$dF = -SdT - MdH$$

At constant temperature ($dT = 0$) we have:

$$\frac{\partial F}{\partial H}\Big|_T = -M$$

As free energy is function of H in all directions the system will give different values for magnetisation in x and z.

Appendix C: Spectrum of a non-square matrix

Singular Value Decomposition (SVD) is a factorisation technique where a any sized complex or real valued matrix $\mathbf{X}$ is decomposed analogously to eigenvalue decomposition of rectangular matrices.

$$\boldsymbol{X} = \boldsymbol{USV}$$

Where the matrices have dimensions: $\mathbf{X}$; $N \times M$, $\mathbf{U}$; $N \times N$, $\mathbf{S}$; $N \times M$, $\mathbf{V}$; $M \times M$.
Despite not being rectangular the matrix $\mathbf{S}$ has i diagonal elements sv_i called *singular values* . These values are arranged in descending order, to a *spectrum* of the matrix $\mathbf{X}$. The values and decaying rates of the spectra often reveal interesting information about the matrix. For instance if $\mathbf{X}$ is an image the technique can be used to quantify the amount of noise and therefore undesired artefacts in the image. The technique is widely used in medical imaging to extract the useful information from brain scans and other high noise imaging[15].

Appendix D: Supplemental Materials

Github repository with all the code used to generate the plots and results in this report can be found: https://github.com/samuelheczko/King-s_ML_QM_Project