1 Theory

1.1 The Variational Method

The variational method is a powerful tool in quantum mechanics, and is used to estimate the ground state energy of a system. This method is based on the variational principle, and is an effective way of approximating the ground state which is very often difficult to find, analytically but also numerically. We start with an initial ansatz wavefunction, and calculate the expectation value of our system Hamiltonian

$$E = \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\int d\sigma \Psi^*(\sigma,\theta) H \Psi(\sigma,\theta)}{\int d\sigma \Psi^*(\sigma,\theta)\Psi(\sigma,\theta)}. \tag{1}$$

where θ are the variational parameters, and σ is the spin configuration of the system. The expectation value of the Hamiltonian is then minimized with respect to the variational parameters, and the optimal parameters are used to estimate the ground state energy of the system.

1.1.1 The Variational Principle

The variational principle states that for any given Hamiltonian, the expectation value of the Hamiltonian in any state is always greater than or equal to the ground state energy of the Hamiltonian. This means that if we have a trial wavefunction, $|\Psi\rangle$, we can use this to estimate the ground state energy of the Hamiltonian.

$$E_0 \leq \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle}. \tag{2}$$

As we can see, if we work to minimize the expectation value of the Hamiltonian, we will always find a "better" estimate to the ground state energy - while never going below it.

1.2 The Ising model

The Ising model is a well-studied model in statistical mechanics, and is used to describe the behavior of ferromagnetic materials. The model was first introduced by Wilhelm Lenz in 1920, and later solved by Ernst Ising in 1925. The model consists of a lattice of spins, where each spin can take on one of two values, $s_i = \pm 1$. The energy of the system is given by the Hamiltonian

$$H = -J \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - \mu \sum_i \sigma_i^z, \tag{3}$$

where J is the coupling constant, σ_i is the spin at site i, and μ is the magnetic moment. The first sum runs over all nearest neighbor pairs of spins, and the second sum runs over all spins in the lattice. In our system, we will assume periodic boundary conditions - meaning we have a spin-chain lattice, with a constant non-zero temperature $T_0 = 1.0$.
The simplest 1D Ising model does not have a phase transition, and can be solved analytically.

1.2.1 Long-range interaction

Introducing long-range interactions between spins in the lattice will allow us to study how the ground state changes when more interactions are added. The Hamiltonian for the long-range Ising model is given by

$$H = -J \sum_{i \neq j} \frac{\sigma_i^z \sigma_j^z}{|i-j|^\alpha} - \mu \sum_i \sigma_i^z, \tag{4}$$

where the coupling constant now is also a function of the distance between the spins, and the first sum runs over all pairs of spins in the lattice. The parameter α determines the range of the interaction, and for $\alpha = 0$ we have the nearest neighbor Ising model. This means we have a splitwise-function to express the interaction between spins, as presented in the article by Martinez et al. [**martinez2022critical**].

$$J = \begin{cases} J_0 \frac{1}{|i-j|} & \text{if } |i-j| < n_v, \\ 0 & \text{otherwise} \end{cases}$$

where n_v is the number of neighbouring spins to include in the long-range interaction, i and j the indices of the spins, and J_0 is the coupling constant.

1.3 The Monte Carlo method

Initially equation (**??**) does not look like something we can use Markov-Chain Monte Carlo methods (MCMC) to evaluate, however with some clever rewriting it can be reformulated into a more useful expression. We begin by re-expressing the integrand in the numerator

$$\Psi_T^*(\sigma,\theta)H\Psi_T(\sigma,\theta) = \Psi_T^*(\sigma,\theta)\frac{\Psi_T(\sigma,\theta)}{\Psi_T(\sigma,\theta)}H\Psi_T(\sigma,\theta).$$

In this expression we recognise both the expression for the local energy E_L and the expression of the un-normalised probability density $\tilde{P} = \Psi_T^*\Psi_T$ formed from the wavefunction. Applying this to equation (**??**), and using the normalising factor in the denominator to normalise $\tilde{P}$, which we will denote by simply P, we get

$$E[H] = \int d\sigma P(\sigma,\theta)E_L(\sigma,\theta) \approx \frac{1}{N}\sum_i n_i(E_L)_i. \tag{5}$$

Where N is the total number of samples, n_i is the number of samples where the local energy has the value $(E_L)_i$ and

$$P(\sigma,\theta) = \frac{|\Psi_T(\sigma,\theta)|^2}{\int d\sigma \Psi_T^*(\sigma,\theta)\Psi_T(\sigma,\theta)}, \quad E_L = \frac{1}{\Psi_T}H\Psi_T.$$

This form of the local energy now allow us to solve the integral and get an estimate of the ground state energy of the system, for a given value of variational parameters using MCMC methods by sampling the local energy throughout spin-configuration space. The Monte Carlo method can be summarised into the following points

1: Initialise the system: Pick an initial spin-configuration $\sigma_0 = \sigma$ and a randonmly initialized variational parameter vector θ. Also fix the number N of MC samples the system should run over.
2: Select a proposed spin-configuration x_p, and let x_n be the current spin-configuration.
3: Check with some condition to see if the proposed move is accepted
4: **if** The proposed move is accepted **then**
5: Set the new spin-configuration as the proposed $\sigma_{n+1} = \sigma_p$
6: **else**
7: Set the new spin-configuration as the old spin-configuration $\sigma_{n+1} = \sigma_n$
8: **end if**
9: With the spin-configuration σ_{n+1} save or update any new values of importance and return to point (2) until we have N samples.

There are multiple ways of finding both a candidate for the new spin-configuration and accepting or rejecting the proposed move. We will use the Metropolis-Hastings algorithm, which is a widely used algorithm for MCMC methods.

1.3.1 Metropolis-Hastings

Here we will give a brief overview of the steps in the Metropolis-Hastings MCMC algorithm, but a more detailed explanation can be found in [**Metropolis**].

- *Proposal of spin-configuration*: The Metropolis algorithm is a Markov Chain Monte Carlo method, and as such we need to propose a new spin-configuration based on the current spin-configuration. This is done by "swapping" a random spin in the current lattice configuration, σ_i with it's opposite spin, $-\sigma_i$, i.e if the spin is up, we propose a new configuation where the spin is down.

- *Acceptance algorithm*: Given a new spin configuration, we need to decide whether to accept or reject the proposed re-configuration. This is done by calculating the ratio of the probability of the new spin-configuration and the old spin-configuration, and comparing this to a random number between 0 and 1. If the ratio is greater than the random number, we accept the configuration, if not we reject it. The acceptance probability is given by

$$P_{\text{accept}} = \min\left(1, \left|\frac{P(\sigma_{\text{new}})}{P(\sigma_{\text{old}})}\right|^2\right).$$

1.4 Restricted Boltzmann Machines

Solving for the ground state of our hamiltonian, it is important that we make a good wavefunction ansatz. In this ansatz, it is of the essence that we manage to properly capture the entanglement (or other physical properties) in the system - this is where the Restricted Boltzmann Machine (RBM) comes in. The generative nature of the probabilities produced by the RBM makes it a good candidate for modeling the wavefunction of a quantum system, and we will use this to our advantage in this project.

1.4.1 Architecture

The RBM is a type of neural network, and consists of two layers of nodes, a visible layer and a hidden layer. The nodes in the visible layer represent the physical degrees of freedom in the system, and the nodes in the hidden layer are used to model the correlations between the visible nodes. The nodes in the visible layer are connected to the nodes in the hidden layer, but there are no connections between nodes in the same layer. This is what makes the RBM a "restricted" Boltzmann machine. A visualization of the network is shown in figure **??**. Working with our 1D Ising Hamiltonian, we require a discete spin

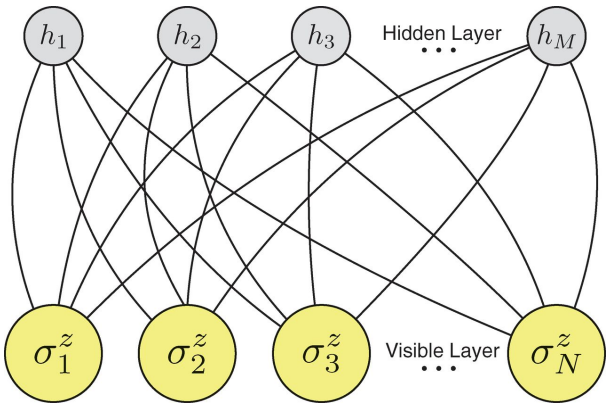

Figure 1: Visualization of the Restricted Boltzmann Machine.[**carleo2017solving**]

lattice model - and thus our RBM will be a *binary-binary* RBM. This means that the nodes in the visible layer and the hidden layer can only take on binary values, $\sigma_i, h_j \in \{0, 1\}$, and the RBM is described in detail by the Boltzmann distribution

$$P(\sigma, \mathbf{h}, \theta) = \frac{1}{Z(\theta)} e^{\frac{1}{T} - E(\sigma, \mathbf{h}, \theta)}, \tag{6}$$

where $Z(\theta)$ is the partition function, and θ are the variational parameters, T the temperature of the RBM (usually put to 1.0), and $E(\sigma, \mathbf{h}, \mathbf{a})$ is the energy-function of the RBM, expressed in a binary-binary RBM as

$$E(\sigma, \mathbf{h}, \theta) = -\sum_i a_i \sigma_i - \sum_j b_j h_j - \sum_{i,j} w_{ij} \sigma_i h_j, \tag{7}$$

where σ is the visible layer, $\mathbf{h}$ is the hidden layer, and $\mathbf{a}, \mathbf{b}, \mathbf{w}$ are the variational parameters.

1.4.2 Initialization

As with any neural network architecture, the visible and hidden nodes are connected, as seen in figure (**??**), and these connections have their weights and biases. Furthermore, the RBM has a bias on both layers - and a "running" an RBM means to give an input to the visible layer, run it "through" the network to the hidden layer, and then take the trip back to the visible layer. The goal is then for the machine to "re-generate" a desired output, given some input.

For our purposes, we only will make use of the probability distribution supplied by the RBM - but the weights and biases play an important role nonetheless, and making a good initial starting point for the variational parameters is important.
Following the article [**carleo2017solving**], we will initialize the parameters as follows

$$W_{ij}, a_i, b_j \in (-0.1, 0.1) \tag{8}$$

which will be samples from the standard normal distribution with $\mu = 0$, $\sigma = 0.1$.

1.4.3 Marginal probability

In order to find the probability of any given configuartion of the visible units (the spin-configuration), we need to marginalize over the hidden units. This is done by summing over all possible hidden unit configurations, and the marginal probability is given by

$$P(\sigma) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} e^{-E(\sigma, \mathbf{h}, \mathbf{a})}, \tag{9}$$

In our binary-binary model, we will only allow the hidden units to take on values of -1 and 1. With that, and no intra-layer interactions, we can trace out the hidden units completely [**carleo2017solving**], and the marginal probability is then given by

$$P_\sigma = e^{\sum_j a_j \sigma_j} \times \prod_i^M F_i(\sigma), \tag{10}$$

$$F_i(\sigma) = 2 \cosh \left[b_i + W_{ij} \sigma_j \right]. \tag{11}$$

This marginal probability is what will constitute our wavefunction, and we will use this to calculate the local energy of the system. As we can see, this is not dependent on the hidden units - which gives us freedom in how we initialize our RBM. Meaning,

$$\Psi(\sigma, \theta) = e^{\sum_j a_j \sigma_j} \times \prod_i^M F_i(\sigma), \tag{12}$$

where we've excluded the partition function, since this is usually intractable - and will cancel in our expression for calculating acceptance probabilities (see (**??**)).

1.5 Optimization

By use of the variational principle, we can find the optimal variational parameters by minimizing the expectation value of the local energy. The "optimal" variational parameters will then be the ones that yields the lowest expectation value, and by (**??**), this will be our closest estimate to the ground state energy (and ground state configuration). For our project, we will make use of steepest gradient descent for optimization - since our problem is a mild minimization problem, this simple method will still yield strong results.

1.5.1 Steepest Gradient Descent

The steepest gradient descent method is a simple optimization algorithm, where we iteratively update the variational parameters in the direction of the steepest gradient. The gradient descent algorithm is as follows:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \nabla_\theta \langle E_L \rangle, \tag{13}$$

where η is the learning rate, θ the variational parameters, and $\nabla_\theta \langle E_L \rangle$ is the gradient of the local enerrgry w.r.t. the variational parameters at the current iteration.
I.e we calculate the gradient of the parameter(s) we are using to minimize the expectation value of the Hamiltonian, and then update the parameters by taking a step in the direction of the negative gradient. Finding the gradient of the local energy is non-trivial, and will also have to be estimated using the monte-carlo sampler. Following the derivations found in [**RBM˙gradients**], the gradients are found by

$$\frac{\partial \langle E_L \rangle}{(\partial \theta_i)} = 2\big(\langle E_L \frac{1}{\Psi} \frac{\partial \Psi}{\partial \theta_i} \rangle - \langle E_L \rangle \langle \frac{1}{\Psi} \frac{\partial \Psi}{\partial \theta_i} \rangle \big) \tag{14}$$

Calculating the gradients, $\frac{1}{\Psi}\frac{\partial \Psi}{\partial \theta_i}$, we will follow the results presented in app. C of [**carleo2017solving**], and use the expression for the gradient of the local energy w.r.t. the variational parameters as

$$\begin{aligned}\frac{1}{\Psi}\partial_{a_i}\Psi &= \sigma_i,\\ \frac{1}{\Psi}\partial_{b_j}\Psi &= \tanh[\gamma_j],\\ \frac{1}{\Psi}\partial_{W_{ij}}\Psi &= \sigma_i \tanh[\gamma_j],\end{aligned}$$

where $\gamma_j = b_j + \sum_i W_{ij}\sigma_i$. These derivatives will be evaluated for each spin configuration in the sampling process, and found the same way an estimate for the local energy is found, see (**??**), and used to update the variational parameters in the gradient descent algorithm.