University of Oslo

FYS4411

Computational physics II: Quantum Mechanical Systems

Project 2: The restricted Boltzmann machine applied to the quantum many-body problem

Author:
Elisabeth Christensen

Author:
Håkon D. Fossheim

June 15, 2020

In this project, we employ the restricted Boltzmann machine to find the expected ground-state energy of interacting electrons confined in an isotropic harmonic oscillator potential. The neural-network quantum state is optimized by use of stochastic gradient descent. Three different Marcov chain Monte Carlo methods are used to estimate said energy, and their performance is compared. These methods consist of the Metropolis, Metropolis-Hastings and Gibbs sampling procedures.

To give an estimate of the error in our results, while accounting for the correlations in the Markov chain produced by these MCMC procedures, the blocking method is applied. We found that the Metropolis algorithm gave the lowest blocking error, by an order of magnitude compared to the other two MCMC procedures.

Lastly, the RBM was found to produce an NQS which mostly gave accurate energy measurements in all dimension for varying particle numbers. For two interacting electrons in two dimensions, Gibbs sampling was found to give a ground state energy measurement of 2.89 a.u., which most closely match the analytical value of 3 a.u. compared to the two other sampling procedures when using a common learning rate of $\eta = 0.01$ and 4 hidden nodes.

I. Introduction

G. Carleo and M. Troyer[1] were the first to demonstrate that the many-body wave function of the Ising model could be successfully modeled by the restricted Boltzmann machine. In this project, we investigate its applicability to various numbers of electrons confined in an isotropic harmonic oscillator potential in one, two and three dimensions. Of particular interest is the system consisting of two interacting electrons confined in the two dimensional potential, as here we have analytical values for the ground state energy for selected values of the oscillator frequency [7], in particular $E_{gs} = 3$ a.u. for $\omega = 1$, which is what will be relevant in this project.

The Pauli exclusion principle states that particles of

spin $s = (1/2 + n)$, $n \in \mathbb{N}$, viz. fermions, cannot occupy the exact same state. There is however no such restriction for integer spin particles, viz. bosons.

Modifying the trial wave function to include more particles (and dimensions) is therefore rather straightforward in the bosonic case, as was seen in the previous project, because they can keep being added to the lowest energy level of the system.

Due to the Pauli exclusion principle, one cannot keep placing electrons into the lowest energy level of the system. Instead, they will start filling up levels/shells of higher energy and Ψ_T cannot be adjusted in a simple manner to account for this.

So rather than tediously (and perhaps unsuccessfully) constructing new guesses for Ψ_T by hand, this task is delegated to the RBM. The RBM keeps providing new guesses for Ψ_T until they are deemed satisfactory. The criterion by which a guess is judged, is the expected local energy it yields, $\langle E_L \rangle$.

Because $\langle E_L \rangle$ can be impossible to calculate analytically, this is done through one of the three mentioned MCMC algorithms. Now that $\langle E_L \rangle$ is found, the variational principle states that the Ψ_T which most adequately (or even exactly) matches the true ground state wave function of the system is given by implementing the optimization algorithm (stochastic gradient descent) in such a way so as to minimize $\langle E_L \rangle$.

Carleo and Troyer coin the trial wave function produced by the RBM as the *neural quantum state*, but we will mostly keep referring to it as simply Ψ_T.

The integration techniques used in this project build to a large degree on what was covered in project 1. We repeat the theory behind Monte-Carlo integration, but now with a larger emphasis on aspects which weren't thoroughly covered in the previous project, like the optimal step-size and the variational principle. Doing this also makes it easier to contrast the brute-force and importance sampling algorithms known from the previous project with the novel concept of Gibbs sampling, which we will introduce. We refer to the previous project for a discussion on the blocking method [2].

II. Theory

A. The Physical System

The system consists of a number of electrons confined in a two-dimensional isotropic harmonic oscillator potential. Using natural units ($\hbar = c = m = e = 1$) and measuring energy in units of $\hbar\omega$, the Hamiltonian of the system is given by [1]

$$\hat{H} = \sum_{p=1}^{P} -\frac{1}{2}\nabla_p^2 + \frac{1}{2}\omega^2 r_p^2 + \sum_{p<q} \frac{1}{r_{pq}} \tag{1}$$

Here p is the particle index and so ∇_p and r_p are D-component vectors, where D is the dimensionality of the system. The first and second terms represent the kinetic and potential energies respectively, while the last term stems from the Coulomb interaction. Here $r_{pq} = \frac{1}{|r_p - r_q|}$, i.e. the modulus of the the distance between particles p and q. The last sum iterates over every way to couple two electrons of the system (counting each coupling once). Ignoring Coulomb interaction, the spatial part of the wave function for one electron in the 2D harmonic oscillator potential is

$$\phi_{n_x,n_y}(x,y) = AH_{n_x}(\sqrt{\omega}x)H_{n_y}(\sqrt{\omega}y)e^{-\frac{\omega(x^2+y^2)}{2}} \tag{2}$$

with an energy eigenvalue of

$$\epsilon_{n_x,n_y} = \omega(n_x + n_y + 1). \tag{3}$$

The spin part of the wave function are two-component spinors, which we denote by $\chi_\uparrow$ for spin up and $\chi_\downarrow$ for spin down. The total wave function for a spin up and a spin down electron are then respectively

$$\Psi_\uparrow(r_i) = \phi_{n_x,n_y}(r_i) \cdot \chi_\uparrow \tag{4}$$

and

$$\Psi_\downarrow(r_i) = \phi_{n_x,n_y}(r_i) \cdot \chi_\downarrow \tag{5}$$

For the ground state ($n_x = n_y = 0$), the spatial part reduces to

$$\phi_{0,0} = Ae^{-\frac{\omega(x^2+y^2)}{2}} \tag{6}$$

Where A is the normalization constant. We now consider two electrons in the ground state of the 2D harmonic oscillator potential. Their wave function can be written in terms of products of single-particle wave functions. The Pauli principle dictates that the wave function must be anti-symmetric under interchange of the electrons. This requirement is fulfilled by constructing the wave function from the Slater determinant:

$$\begin{aligned}\Psi(1,2) &= \frac{1}{\sqrt{2!}}\begin{vmatrix}\Psi_\uparrow(1) & \Psi_\uparrow(2)\\ \Psi_\downarrow(1) & \Psi_\downarrow(2)\end{vmatrix}\\ &= \frac{1}{\sqrt{2!}}\Big[\Psi_\uparrow(1)\Psi_\downarrow(2) - \Psi_\uparrow(2)\Psi_\downarrow(1)\Big]\end{aligned} \tag{7}$$

and so $\Psi(1,2) = -\Psi(2,1)$ as desired. We can write out the wave function in terms of its spatial components and its and spin components:

$$\Psi(1,2) = \frac{1}{\sqrt{2}}\phi_{0,0}(1)\phi_{0,0}(2)\Big[\chi_\uparrow(1)\chi_\downarrow(2) - \chi_\uparrow(2)\chi_\downarrow(1)\Big]. \tag{8}$$

[1] The scaling process is explained in project 1, appendix F .

Here we used the fact that $\phi_{0,0}(1)\phi_{0,0}(2) = \phi_{0,0}(2)\phi_{0,0}(1)$, viz. the spatial part of the wave function is symmetric under interchange of two particles. The anti-symmetry must therefore fully reside in the spin component, thereby making this a singlet state. It therefore has a spin quantum number $s = 0$.

Furthermore, the Hamiltonian given in eq. 1 does not act on the spinors[2], and so the ground state energy of the two-electron system is

$$\begin{aligned}\hat{H}\Psi(1,2) &= \frac{1}{\sqrt{2}}\hat{H}\Big[\phi_{0,0}(1)\phi_{0,0}(2)\Big]\Psi_\chi \\ &= \frac{1}{\sqrt{2}}\Big[\phi_{0,0}(2)\hat{H}\phi_{0,0}(1) + \phi_{0,0}(1)\hat{H}\phi_{0,0}(2)\Big]\Psi_\chi \\ &= \frac{1}{\sqrt{2}}\left[\phi_{0,0}(2)E_1\phi_{0,0}(1) + \phi_{0,0}(1)E_2\phi_{0,0}(2)\right]\Psi_\chi \\ &= (E_1+E_2)\frac{1}{\sqrt{2}}\phi_{0,0}(1)\phi_{0,0}(2)\Psi_\chi \\ &= 2\omega\Psi(1,2) \end{aligned} \tag{9}$$

Where Ψ_χ denotes the spin part of the wave function. We have therefore shown that the ground state energy for two electrons in the 2D harmonic oscillator potential is 2ω when ignoring interactions. Analytical solutions are also obtainable when including interaction, and the ground state energy in this case with two electrons and $\omega = 1$ is shown in [7] to be 3 a.u. .

B. Artificial Neural Networks

Analogous to biological neural networks, artificial neural networks consist of connected neurons/nodes, which activate when their input exceed a certain threshold. The "brain" learns through synaptic plasticity in a process called long-term potentiation where neural pathways get strengthened through repeated stimulation. Artificial neural nets on the other hand learn by continually adjusting their weights and biases so as to minimize a cost function in a process known as back-propagation.

A neural networks consist of layers of nodes interconnected by weights. In feed-forward neural networks you have an input layer, one or more hidden layers and an output layer, as shown in figure 1. As an example, consider a neural network designed to categorize images of handwritten digits between 7 and 9 from the MNIST database[3].

[2] Nor should it, as the energy of the system is independent of spin.
[3] This database actually contains handwritten digits between 0 and 9, but the figure for this is quite unpleasant.

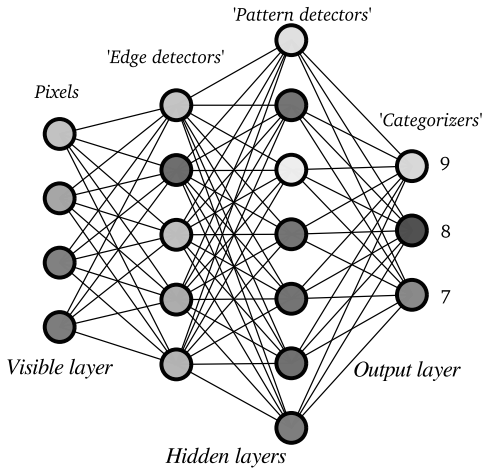

FIG. 1. Depiction of the neural network in our toy example. Each line corresponds to a weight, and each node has a bias and an activation value associated with it. The activation value is represented by the brightness of the node. The brightness of a given node in the output layer indicates how confident the neural net is in this classification.

Each node of the input layer corresponds to a pixel in the image, encoding its gray-scale value in an activation value between 0 and 1. By connecting every node in the input layer to every node in the adjacent hidden layer, a given node of this hidden layer can adjust its weights so as to pick up e.g. an edge in a certain part of the image. The different hidden nodes of this layer detect edges in different parts of the picture. Additional hidden layers then pick up patterns in these edges (e.g. do the edges in the image combine to make a circle?). We could keep adding hidden layers, which would pick up patterns in these patterns and so on. In this way, each additional hidden layer allows for greater abstraction of the data encoded in the image.

The nodes in the output layer are then able to categorize the image by adjusting its weights to pick up certain combinations of patterns in the image. In our example, we would have 3 nodes in the output layer, each corresponding to a digit between 7 and 9. The node corresponding to 9 might activate when one node of the last hidden layer detects a circle-like shape and another detects a line right below it.

At the start, the neural network will not be able to classify the digits correctly. It needs to be provided with correctly classified samples (training data) so as to compare how well an image was classified via a cost function, and adjust its weights to decrease this function. The process of making these adjustments is what's referred to as back-propagation.

In addition to weights and activation values, the neural network also has a bias associated with every node. These biases are also adjusted in back-propagation, thereby allowing each node to adjust its activation threshold. This gives the neural network greater flexibility when trying to fit the task at hand.

Note that, in actuality, the network will probably not configure itself to detect these specific patterns, but it serves as an illustrative example. The terms in quotation marks in figure 1 have been made up for this example.

C. Gaussian-Binary Restricted Boltzmann Machines

The Boltzmann machine is a Markov Random Field with stochastic visible and hidden units [6]. We may represent these visible and hidden units as $\mathbf{X} = (X_1, X_2, ..., X_M)^T$ and $\mathbf{h} = (h_1, h_2, ..., h_N)^T$ respectively. Similarly to neural networks, every node has a bias and each hidden node is connected to every visible node by weights as shown in figure 2. Generally, one may also have intra-layer connections in addition to the inter-layer ones. The exlusion of these intra-layer connections is what makes our Boltzmann machine restricted. This restriction makes the conditional probabilities of the nodes independent of one another[4] and allows for more efficient training algorithms than what is available for the BM. Furthermore, this independence is an assumption on which our Gibbs sampling algorithm rests.

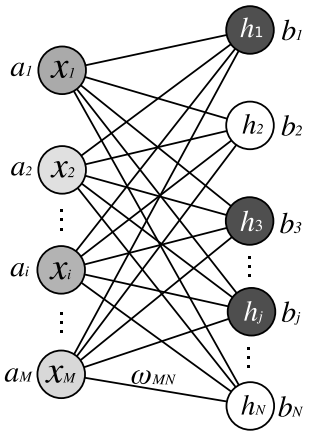

FIG. 2. Depiction of the Gaussian-binary RBM. The left layer shows the visible nodes while the right layer shows the hidden ones. Their biases are also shown, in addition to the weights which form inter-layer connections only. The activation values of the nodes are depicted through their brightness. The visible nodes take on continuous values while the hidden ones take on binary values, i.e. 0 or 1.

However, the hidden and visible nodes are **not** independent of one another. The set of values they can take, $\{\mathbf{X}, \mathbf{h}\}$, can therefore be described by a joint probability distribution function.

In RBM's, this joint PDF is set to

$$P(\mathbf{X}, \mathbf{h}) = \frac{1}{Z} e^{-\beta E(\mathbf{X}, \mathbf{h})} \tag{10}$$

Here $\beta = 1/(k_B T)$. In this project, we set $T = 1$. Furthermore, since $k_B = 1$ in natural units, we can drop β altogether.

Z is the partition function, which sums up the probability of the system being in each energy state, thereby ensuring that $P(\mathbf{X}, \mathbf{h})$ is normalized:

$$Z = \int \sum_{\{\mathbf{h}\}} e^{-E(\mathbf{X}, \mathbf{h})} dx \tag{11}$$

The energy of a particular configuration of the node values is defined as

$$E(\mathbf{X}, \mathbf{h}) = \sum_{i=1}^{M} \frac{(X_i - a_i)^2}{2\sigma^2} - \sum_{j=1}^{N} b_j h_j - \sum_{i,j}^{M,N} \frac{X_i \omega_{ij} h_j}{\sigma^2} \tag{12}$$

Now, the marginal distribution of $\mathbf{X}$, namely $P(\mathbf{X})$ is what will be used to model the wave function. It is shown in appendix A that

$$\begin{aligned} \Psi_T(\mathbf{X}) &= P(\mathbf{X}) \\ &= \frac{1}{Z} e^{-\sum_{i=1}^{M} \frac{(X_i - a_i)^2}{2\sigma^2}} \prod_{j=1}^{N} (1 + e^{b_j + \sum_{i=1}^{M} \frac{X_i w_{ij}}{\sigma^2}}) \end{aligned} \tag{13}$$

Where $\alpha = (a_1, ..., a_M, b_1, ..., b_N, \omega_{11}, ..., \omega_{MN})$ are all the weights and biases. The job of the RBM is to give us a good[5] trial wave function, and all these free parameters give it a lot of flexibility in doing so.

D. The Metropolis Algorithm

The variational principle[6] states that for a system with ground state energy E_{gs}, any trial wave function we can come up with satisfies the following relation

$$E_{gs} \leq \langle \hat{H} \rangle = \frac{\int \Psi_T^*(\mathbf{X}) \hat{H} \Psi_T(\mathbf{X}) d\tau}{\int \Psi_T^*(\mathbf{X}) \Psi_T(\mathbf{X}) d\tau} \tag{14}$$

By defining the local energy

$$E_L = \frac{1}{\Psi_T(\mathbf{X})} \hat{H} \Psi_T(\mathbf{X}) \tag{15}$$

[4] Specifically, $P(h_i|\mathbf{X})$ and $P(h_{\neq i}|\mathbf{X})$ as well as $P(X_j|\mathbf{h})$ and $P(X_{\neq j}|\mathbf{h})$ are independent. This is further discussed in section II F.

[5] What is meant by good is discussed in section II G.

[6] Discussed in appendix C.

We may rewrite eq. 14 as

$$E_{gs} \leq \frac{\int |\Psi_T(\mathbf{X})|^2 E_L d\tau}{\int |\Psi_T(\mathbf{X})|^2 d\tau} = \int E_L(\mathbf{X}) \rho_T(\mathbf{X}) d\tau = \langle E_L \rangle \tag{16}$$

Where

$$\rho_T(\mathbf{X}) = \frac{|\Psi_T(\mathbf{X})|^2}{\int |\Psi_T(\mathbf{X})|^2 d\tau} \tag{17}$$

Is the probability density of the trial wave function.

Thus the expectation value of the local energy will be an upper bound to the system's true ground state energy. We therefore want to find a Ψ_T which minimizes $\langle E_L \rangle$, which is the task we assign the RBM.

Because we are unable to calculate $\langle E_L \rangle$ analytically, we must resort to numerical integration. Since the integral in question can be of very high dimensionality, the technique of choice will be Monte-Carlo integration.

The integral may then be approximated as

$$\langle E_L \rangle = \int E_L(\mathbf{X}) \rho_T(\mathbf{X}) d\tau \approx \sum_{i=1}^{N} E_L(\mathbf{X}_i) \rho_T(\mathbf{X}_i) \tag{18}$$

Where N is the number of Monte Carlo samples. However, we are not able to find a closed form expression for $\rho_T(\mathbf{X})$ due to $\int |\Psi_T(\mathbf{X})|^2 d\tau$ being intractable. We therefore use Metropolis sampling. This is done by first picking a random starting point, and iteratively proposing new moves in a random direction within a specified distance of the current position in configuration space;

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{r} \Delta X. \tag{19}$$

Here $\mathbf{r}$ is a random direction on the unit sphere centered on $\mathbf{X}_i$ and ΔX is a user defined step length. If the following holds:

$$\omega = \frac{\rho_T(\mathbf{X}_{i+1})}{\rho_T(\mathbf{X}_i)} > 1 \tag{20}$$

, the proposed move is accepted and E_L is sampled. Otherwise, we reject the move if $w < s$ where $s \in [0, 1]$ is a continuous random variable. In this way, the initially discarded move is accepted with a probability proportional to ω. This ensures that the walker moves toward regions of high probability, thereby prioritizing regions of interest whilst not getting stuck at a local/global maximum. This sampling procedure gives samples of E_L that are distributed according to $\rho_T(\mathbf{X})$ and removes the need to calculate the normalization constant analytically, since it cancels in the fraction. The expectation value of the local energy is then simply

$$\langle E_L \rangle \approx \frac{1}{N} \sum_{i=1}^{N} E_L(\mathbf{X}_i). \tag{21}$$

E. Importance Sampling

The Metropolis algorithm, also known as brute force Metropolis, is suboptimal due to proposing moves in arbitrary directions. This leads to a lot of rejected steps, thus wasting CPU-cycles. The Metropolis-Hastings algorithm, also known as importance sampling, seeks to remedy this by proposing moves that are in the direction of the gradient of the distribution [4]. This "pushes" the walker in the right direction, leading to fewer rejected steps given the same "step length" as the brute force algorithm.

New moves are proposed in the following manner:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \chi + \mathcal{D} F(\mathbf{X}_i) \Delta t \tag{22}$$

Here χ is a random variable picked from a normal distribution with mean 0 and variance $2\mathcal{D}\Delta t$. Δt is called the time step, and lets us adjust the "step length"[7] of the walker. $\mathcal{D}$ is the diffusion constant, which in our case is 1/2, and F is the so-called quantum force. It is given by

$$F(\mathbf{X}_i) = \frac{2}{\Psi_T(\mathbf{X}_i)} \nabla_p \Psi_T(\mathbf{X}_i) \tag{23}$$

Where we are taking the gradient with respect to the coordinate variables of the particle we propose to move. From eq. (A.8) in the appendix, we have

$$\begin{aligned} F = \frac{2}{\Psi} \nabla_p \Psi_T &= 2 \nabla_p \ln \Psi_T = 2 \sum_{k=1}^{D} \hat{e}_k \frac{\partial}{\partial X_k} \ln \Psi_T \\ &= \frac{2}{\sigma^2} \sum_{k=1}^{D} \hat{e}_k \left[(a_k - X_k) \sum_{j=1}^{N} \frac{\omega_{kj}}{1 + e^{-b_j - \sum_{i=1}^{M} \frac{X_i \omega_{ij}}{\sigma^2}}} \right] \end{aligned} \tag{24}$$

Where D is the dimensionality of the system and N is the number of hidden nodes. The transition suggestion rule for importance sampling is given by

$$\omega = \frac{G(\mathbf{X}_i, \mathbf{X}_{i+1}) \rho_T(\mathbf{X}_{i+1})}{G(\mathbf{X}_{i+1}, \mathbf{X}_i) \rho_T(\mathbf{X}_i}. \tag{25}$$

Otherwise, the moves are accepted or rejected in exactly the same way as with brute force. $G(\mathbf{X}_i, \mathbf{X}_{i+1})$ is the Greens function of the Fokker Planck equation, and is up to a constant given by

[7] Note that there is no fixed step length in importance sampling due to the random variable χ. However, the probability that the proposed move lies within a given distance of the previous position can be increased (decreased) by decreasing (increasing) Δt.

$$G(\mathbf{X}_{i+1}, \mathbf{X}_i) \propto \exp\left\{ -\left(\frac{(\mathbf{X}_{i+1} - \mathbf{X}_i - \mathcal{D}\Delta t F(\mathbf{X}_i))^2}{4\mathcal{D}\Delta t} \right) \right\} \tag{26}$$

The step length of the walker is of crucial importance when tuning the sampling method of choice, be it Metropolis or Importance sampling. Everything else being constant, one would want as high of an acceptance rate as possible, since the rejected proposals are nothing but wasted CPU-cycles. One can achieve a high acceptance rate by making the step length sufficiently small. However, this is computationally expensive since it increases the amount of steps needed to map out the wave function. In addition, the samples will get more correlated. One might then try to remedy this by making the step length large, but at a certain point we will start getting far too many proposals to regions of low probability density, which will just get rejected. A general rule of thumb is to opt for a step length with an acceptance rate of about 50% [4].

F. Gibbs Sampling

Gibbs sampling is a method for estimating a multivariate probability distribution when calculating it directly is impractical or impossible. In our case, evaluating the joint probability distribution $P(\mathbf{X}, \mathbf{h})$ is made intractable by the partition function Z. Instead, we repeatedly sample from the conditional probabilities $P(\mathbf{h}|\mathbf{X})$ and $P(\mathbf{X}|\mathbf{h})$ and use these to set the hidden and visible nodes respectively.

By applying Bayes' theorem, the conditional probabilities are shown in [5] to be

$$P(\mathbf{h}|\mathbf{X}) = \frac{P(\mathbf{X}, \mathbf{h})}{P(\mathbf{X})} = \prod_j^N S(v_j) \tag{27}$$

and

$$P(\mathbf{X}|\mathbf{h}) = \frac{P(\mathbf{X}, \mathbf{h})}{P(\mathbf{h})} = \prod_i^M \mathcal{N}(X_i; a_i + \mathbf{w}_{i*}\mathbf{h}, \sigma^2) \tag{28}$$

Here $S(v_j)$ denotes the logistic function of v_j, given in equations (A.9) and (A.10). We see that equations (27) and (28) are written as products of the conditional probabilities of each individual node, namely

$$P(h_j|\mathbf{X}) = S(v_j) = \begin{cases} S(v_j), & , h_j = 1. \\ S(-v_j), & , h_j = 0. \end{cases} \tag{29}$$

and

$$P(X_i|\mathbf{h}) = \mathcal{N}(X_i; a_i + \mathbf{w}_{i*}\mathbf{h}, \sigma^2) \tag{30}$$

They must therefore be independent, which allows us to sample in parallel. Meaning that for a given $\mathbf{X}$ we loop over and set each value of h_j **separately** before calculating $\mathbf{X}$ conditioned on the acquired value of $\mathbf{h}$ in the same manner. We then repeat this process and sample $\mathbf{X}$ for each iteration, as shown in Figure 3.

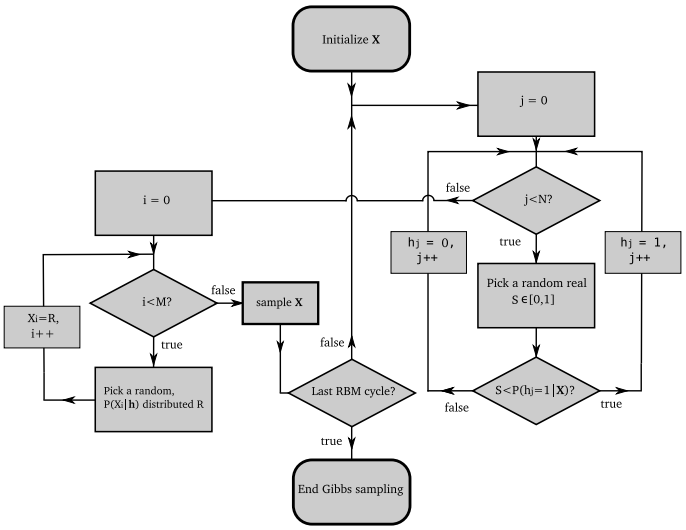

FIG. 3. Flowchart of the Gibbs sampling algorithm.

For each iteration, we use the sampled $\mathbf{X}$ to evaluate the wave function which in turn is used to calculate a local energy "measurement". Similarly to the Metropolis and Metropolis-Hastings algorithms, these measurements constitute a Markov chain. The advantage of using Gibbs sampling is that it's more efficient at exploring configuration space, since we are not wasting computational resources on rejected moves. However, the conditions for being able to use Gibbs sampling are more stringent as we need to know the conditional probabilities and be able to sample from them.

If we assume the wave function is positive definite, we can set it as

$$\Psi_T(\mathbf{X}) = \sqrt{P(\mathbf{X})} \tag{31}$$

Which means that the probability density of the system is equal to the marginal distribution of $\mathbf{X}$:

$$|\Psi_T(\mathbf{X})|^2 = [\Psi_T(\mathbf{X})]^2 = P(\mathbf{X}) \tag{32}$$

Thus the restricted Boltzmann machine now directly models the position of the particles in our system, which was not the case in brute force and importance sampling. We must be a little cautious though, as the expressions used for gradient descent and local energy will now differ, as shown in the appendix.

G. Stochastic Gradient Descent

Choosing a good trial wave function for our sampling procedure can be quite difficult. Regardless, our results

very much depend on a good choice. If we happened to pick the correct wave function, then we would get the exact[8] local energy for the system. The variational principle C states that any other choice of wave function will give a higher local energy.

Knowing this, we can let the RBM do the heavy lifting. We initialize the free parameters of the wavefunction[9], and let it adjust them so as to come up with a wave function which gives a lower local energy. It does not magically know how to do this. Rather, we tell it by specifying which direction in parameter space will yield a lower local energy than what was produced by its previous guess.

The local energy of a given guess is found by employing the sampling algorithm of choice. The procedure of generating the trial wave function and calculating the local energy is what's referred to as an RBM cycle. When we are satisfied with the local energy output of these cycles, we stop the RBM and use the output from the last cycle as our measurement.

So how do we know in which direction the RBM should move in parameter space? As you may have guessed, we find the gradient with respect to E_L in this space. To do this, we need to differentiate E_L with respect to each of the free parameters[10]. Lastly, we specify how far to move in the direction opposite to the gradient through the learning rate, η. We move to this new location by adjusting the free parameters according to $\alpha_i \rightarrow \alpha_i + \frac{\partial <E_L>}{\partial \alpha_i}\eta$.

Now you may ask: What's left for the RBM to do? The answer is **nothing**. The RBM is just a black box put around the process of finding the gradient, moving in the direction opposite to it and adjusting the joint PDF of $\mathbf{X}$ and $\mathbf{h}$ accordingly[11], with this joint PDF set to the Boltzmann distribution. Figure 2 simply provides a nice visualization of the correlation of $\mathbf{X}$ and $\mathbf{h}$, and this correlation is specified by the functional form of the energy given in eq. (12). For example, if the cross terms were set to zero via the weights, we could write $P(\mathbf{X}, \mathbf{h}) = P(\mathbf{X})P(\mathbf{h})$. $\mathbf{X}$ and $\mathbf{h}$ would no longer be correlated and Figure 2 would no longer contain any lines. Of course, this would defeat the whole purpose of the hidden variables $\mathbf{h}$, which is to produce a hypersurface defined by the closed-form expression for E_L in which the global minimum is likely to be close or equal to the ground state. Removing the dependence on the hidden variables would reduce the dimensionality of this hypersurface, thereby reducing the chance of it containing a minimum sufficiently close to the ground state.

[8] Excluding errors stemming from the sampling.
[9] How we chose to do this is discussed in section III A.
[10] This is done in appendix B.
[11] From which the trial wave function can be found by marginalizing over $\mathbf{h}$.

III. Results

A. Brute force method

We attempted to first look at the effects of initializing the weights with different distributions, namely uniform and normal. The results in the figures within this section reflects the energies found where we have also taken into account different values for the variation σ for the normal distribution, and different intervals for the uniform distribution. Figures 4 and 5 show the convergence of the energy according to the initialization of the weights. Although both distributions lead to the expected analytical value of E_L, there is still a clear difference between the two when the intervals for the uniform distribution increases and the standard deviation for the gaussian distribution increases. Here, we see that an initialization of the weights using a normal distribution results in a more stable convergence towards the analytical value of the local energy. However, it appears also that a uniform initialization results in a faster convergence towards E_L. We will therefore, from this point on, initialize weights using a normal distribution with $\sigma = 0.01$.

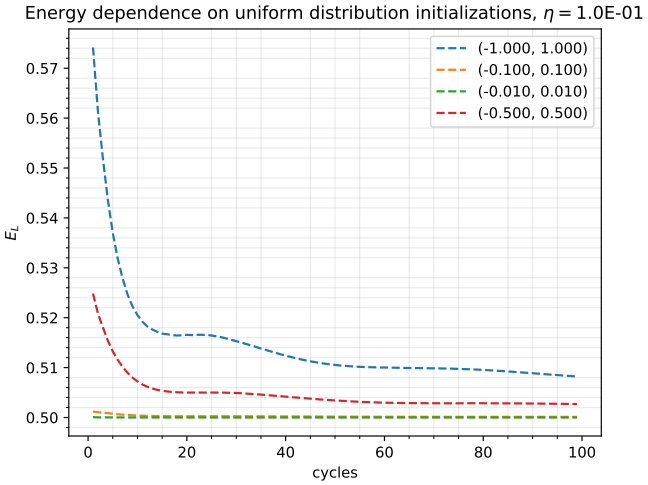

FIG. 4. Energy vs. RBM cycles, with weights initialized according to a uniform distribution for different intervals. Performed using the brute force method.

TABLE I. Local energies for one particle in 1D as a function of number of hidden nodes, nh, with corresponding blocking error, for both a uniform and normal initialization of weights. Uniform weights are initialized between (-0.01, 0.01), while normal weights are initialized with $\sigma = 0.01$. This was run using $\eta = 0.1$.

	Uniform		Normal	
nh	E_L[a.u.]	σ_b	E_L[a.u.]	σ_b
4	0.5000	2.09623e-06	0.5000	7.35366e-07
6	0.5000	2.39605e-06	0.5000	2.10714e-05
8	0.5000	3.44103e-06	0.5000	1.17843e-05
10	0.5000	4.19890e-07	0.5000	1.09455e-06
12	0.5000	1.06411e-05	0.5000	1.87983e-05
14	0.5000	1.35619e-05	0.5000	4.68948e-05
16	0.5000	6.06849e-06	0.5000	3.96287e-05
18	0.5000	1.06353e-06	0.5000	7.91902e-06
20	0.5000	6.71991e-06	0.5000	3.21553e-05

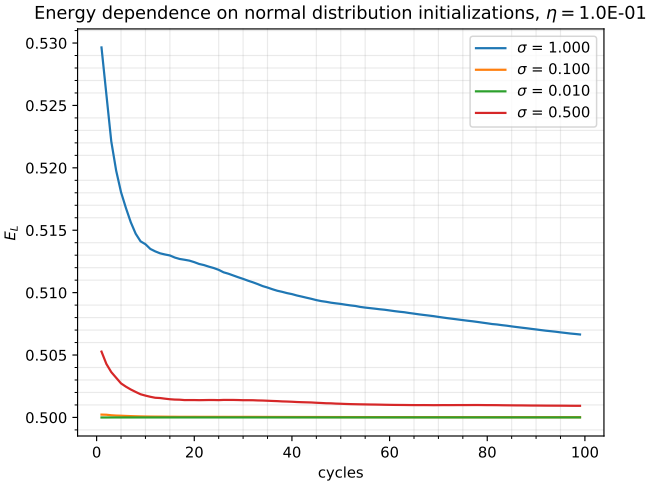

FIG. 5. Energy vs. RBM cycles, with weights initialized according to a normal distribution for different values of σ. Performed using the brute force method.

The energy dependence on the number of hidden nodes can be seen in table I. The energies shown have all been computed for one particle in one dimension, with 2^{19} Metropolis steps, and 100 RBM cycles. Here, the blocking error increases accordingly with the number of hidden nodes. Thus, among the hidden nodes listed in table I, it appears that 4 hidden nodes is the optimum choice for 1 particle in 1D when we have a normal initialization of the weights. However, if we use a uniform distribution of the weights then it appears that 10 hidden nodes result in the lowest blocking error by almost an order lower than that of 4 hidden nodes. Figures 4 and 5 are computed using 2 hidden nodes.

The energy dependence on the learning rates can be seen in table II for weights initialized both using a uniform distribution and normal distribution. Here, the energy estimated is the local energy measured during the last RBM cycle, with corresponding blocking error σ_b.

TABLE II. Local energies for one particle in 1D as a function of learning rates, with corresponding blocking error, for both a uniform and normal distribution of weights. Uniform weights are initialized between (-0.01, 0.01), while normal weights are initialized with $\sigma = 0.01$ and $\mu = 0$. This was run using 2 hidden nodes.

	Uniform		Normal	
η	E_L[a.u.]	σ_b	E_L[a.u.]	σ_b
1.0e-07	0.5000	2.87254e-05	0.5000	1.05893e-05
1.0e-06	0.5000	2.85790e-05	0.5000	1.02797e-05
1.0e-05	0.5000	2.81894e-05	0.5000	1.00948e-05
1.0e-04	0.5000	2.85388e-05	0.5000	1.03552e-05
1.0e-03	0.5000	2.52666e-05	0.5000	8.79470e-06
1.0e-02	0.5000	5.80964e-07	0.5000	2.22527e-07
1.0e-01	0.5000	7.25645e-06	0.5000	2.63677e-06
1.0e+00	0.5000	5.99223e-07	0.5000	2.46922e-07
1.1e+00	0.5000	9.99585e-06	0.5000	3.52038e-06
1.2e+00	0.5000	7.41994e-06	0.5000	2.58580e-06

Figures 6 and 7 show the convergence of the local energy according to the learning rates used. In this case, one can see that a learning rate between 0.01 and 1.2 results in the most approximate values of E_L. These values also appear to converge the fastest with a relatively stable average value of E_L after about 50 cycles. For values lower than 0.01, the fluctuations become more distinct, and the energies have still not reached a stable value of $E_L = 0.5$ after 100 RBM cycles. From table II we can see the effect of the learning rates through their blocking errors. Here, the error appears greater for $\eta \in$ (1e-7, 1e-3) than for $\eta \in (0.01, 1.2)$ for both a uniform and normal distribution. However, in all cases the blocking error produced by the uniform distribution appears greater than those produced from weights with a normal initialization. We also see that a learning rate of $\eta = 0.01$ gives the lowest blocking error, for both a normal and uniform initialization. We will therefore utilise a learning rate of 0.01 in most cases beyond this section, if not stated otherwise.

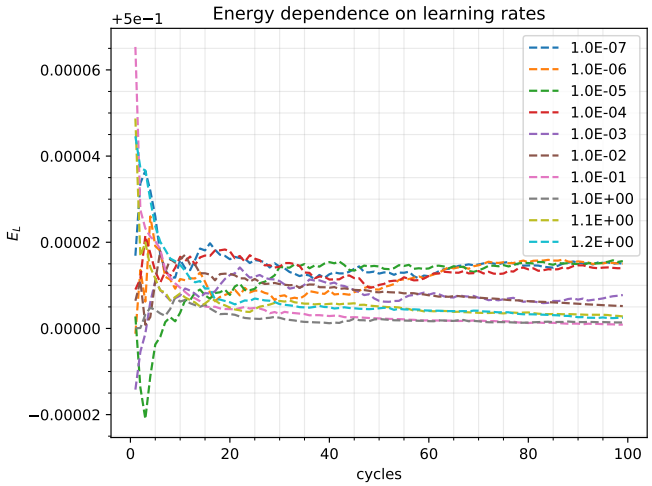

FIG. 6. Energy vs. RBM cycles, with a dependence on learning rates, for weights initialized according to a uniform distribution in the range (-0.01, 0.01). Performed using the brute force method.

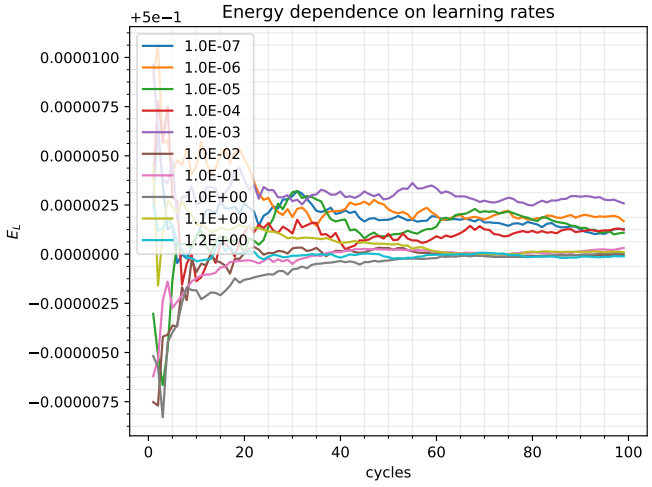

FIG. 7. Energy vs. RBM cycles, with a dependence on learning rates, for weights initialized according to a normal distribution with $\sigma = 0.01$ and $\mu = 0$. Performed using the brute force method.

B. Importance sampling

The local energies as produced by importance sampling can be seen in table IV. This was done using 2^{19} Monte Carlo cycles, with a learning rate of 0.01 and a time step $\Delta t = 1.35$ in order to achieve an acceptance ratio of approximately 0.5. Here, the energies compare well to the analytic energies, and includes also a smaller blocking error in all dimensions compared to that of the brute force method. The performance of importance sampling

TABLE III. Local energies as measured per last RBM cycle using the brute force method. First column corresponds to the learning rate used, with the remaining columns shows the biased error, blocking error and the ratio of blocking error over biased error. This was run using 2^{19} Metropolis steps for 1 particle in multiple dimensions with a step length of 2.0, and using a normal initialization of weights.

η	d	E_L[a.u.]	σ	σ_b	σ_b/σ
0.01	1	0.5000	1.03700e-07	1.13462e-07	1.09413e+00
	2	1.0000	1.13127e-07	1.83396e-07	1.62115e+00
	3	1.5000	9.64888e-07	1.68151e-06	1.74270e+00
0.1	1	0.5000	6.41582e-08	1.29744e-06	2.02226e+01
	2	1.0000	6.97965e-08	2.11074e-06	3.02413e+01
	3	1.5000	5.94960e-07	2.19610e-05	3.69117e+01

TABLE IV. Local energies as measured per last RBM cycle using importance sampling. This was done using 2^{19} Monte Carlo cycles and 100 RBM cycles, with a time step of 1.35 and a learning rate $\eta = 0.01$.

d	E_L[a.u.]	σ	σ_b	σ_b/σ
1	0.5000	9.89124e-08	1.02575e-06	1.03703e+01
2	1.0000	1.07774e-07	1.74102e-06	1.61544e+01
3	1.5000	9.19789e-07	1.86157e-05	2.02391e+01

compared to that of the brute force method can also be seen in figure 8. Here, the effect of importance sampling becomes quite clear as the variance in the local energies are much less distinct than the variances from the brute force method.

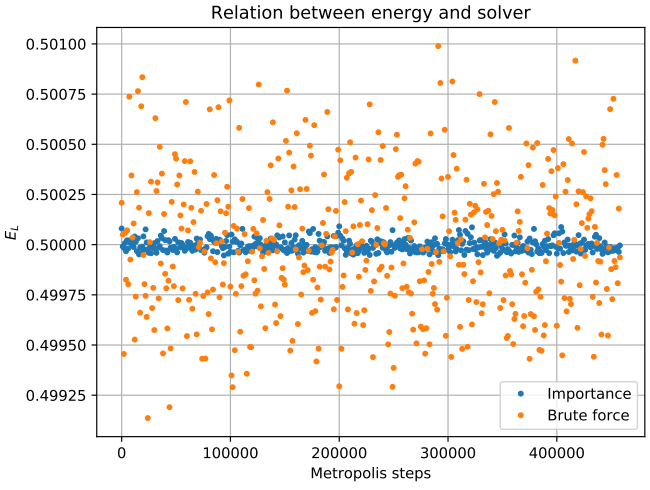

FIG. 8. Energy vs. number of Metropolis steps, for a learning rate of $\eta = 0.1$. This is performed for one particle in one dimension.

TABLE V. Local energies as measured per last RBM cycle using Gibbs sampling. This was done using 2^{19} Monte Carlo cycles and 100 RBM cycles, with a learning rate $\eta = 0.01$.

d	E_L[a.u.]	σ	σ_b	σ_b/σ
1	0.5000	1.16867e-07	1.42357e-06	1.21811e+01
2	1.0000	1.27225e-07	1.54806e-06	1.21678e+01
3	1.5001	1.08569e-06	1.30847e-05	1.20519e+01

C. Gibbs sampling

The local energies produced by Gibbs sampling can be seen in table V for a particle in 1, 2, and 3 dimensions. This was done using 2^{19} Metropolis steps for 100 RBM cycles, and a learning rate $\eta = 0.01$ with 2 hidden nodes. Here, the energies correspond well with the analytical values in all dimensions, although the blocking error for a particle in 3 dimensions is an order higher than in 1 and 2 dimensions. Furthermore, the relation between the energies and the number of hidden nodes can be seen in figure 9.

From figure 9 we see that as the number of hidden nodes increases so too does the energy and the error. We can from the plot conclude that 2 hidden nodes result in the best approximation of the energy to its analytical value with the least error. As we reach 20 hidden nodes, the energy still lies close to 0.5, signifying that our RBM is working as desired, but with the most significant error of the nodes tested.

The relation between the learning rate and the energy is shown in figure 10. Here, the approximation to the local energy is quite exact, with a differing only present in the 10^{-7} power. We can see that a learning rate η =0.01 results in the best fit for the analytical energy, but not necessarily with the optimum value in the error, where σ_b >1e-6. The optimal error is given by $\eta = 0.1$ with σ_b <1e-6.

The energies for different number of particles in the range (4, 10) is shown in table VII for different number of dimensions. In this case, we have chosen to neglect the interaction between the particles, and thus only tested the approximations of the energies produced by our RBM to the analytical, expected values. The relation between σ used for Gibbs sampling is shown in table VI. Here, we see that $\sigma = 1.0$ results in the most precise measurement of the local energy, which is also reflected in the blocking error. As we decrease σ the local energy decreases, while the blocking error increases.

D. Interaction between particles

The energies for the interaction between 2 particles up to 3 dimensions can be seen in table VIII. Here, we have compared a learning rate of η =1e-7 to a learning rate of $\eta = 1.0$. The learning rates used were obtained from figure 11. Here, η =1e-7 gave the highest blocking error

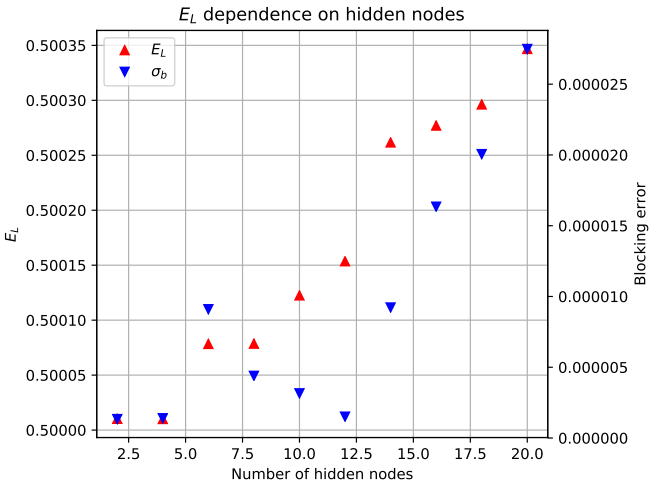

FIG. 9. Local energy dependence on number of hidden nodes using Gibbs sampling. This was run using 2^{19} Metropolis steps, for 100 RBM cycles, and with a learning rate $\eta = 0.01$ for 1 particle in 1D.

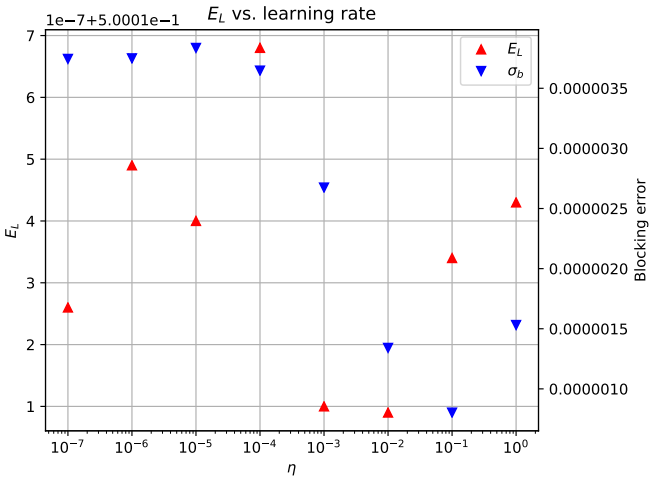

FIG. 10. Local energy dependence on learning rates using Gibbs sampling. This was run using 2^{19} Metropolis steps, for 100 RBM cycles, with 2 hidden nodes for 1 particle in 1D.

TABLE VI. Relation between local energy and σ used in Gibbs sampling. This was run for 1 particle in 1D, with 2^{19} Metropolis steps for 100 RBM cycles, and with a learning rate $\eta = 0.01$ and 2 hidden nodes.

σ	E_L[a.u.]	σ_{biased}	$\sigma_{blocking}$	$\sigma_{blocking}/\sigma_{biased}$
1.0	0.5000	1.16883e-07	1.40240e-06	1.19983e+01
0.75	0.2813	1.27483e-04	1.67872e-03	1.31682e+01
0.50	0.0783	3.96558e-04	5.30722e-03	1.33832e+01
$\sqrt{0.5}$	0.2500	1.57387e-04	2.07649e-03	1.31935e+01

TABLE VII. Local energy as a function of the number of particles, p,and number of dimensions, d, using Gibbs sampling. This was run with 2 hidden nodes and a learning rate $\eta = 0.01$, with 2^{19} Metropolis steps for 100 RBM cycles. The results shown are that of the final RBM cycle and does not include interaction between the particles.

d	p	E_L[a.u.]	σ	σ_b	σ_b/σ
1	4	2.0001	1.71394e-06	2.10424e-05	1.22772e+01
1	6	3.0002	1.71810e-06	2.06602e-05	1.20250e+01
1	8	4.0002	2.04293e-06	2.46488e-05	1.20654e+01
1	10	5.0003	2.48047e-06	3.00556e-05	1.21169e+01
2	4	4.0003	2.04296e-06	2.46492e-05	1.20654e+01
2	6	6.0004	2.49126e-06	3.02323e-05	1.21353e+01
2	8	8.0006	3.25816e-06	3.98699e-05	1.22369e+01
2	10	10.0007	3.65423e-06	4.47637e-05	1.22498e+01
3	4	6.0004	2.49220e-06	3.00048e-05	1.20395e+01
3	6	9.0006	3.48780e-06	4.18399e-05	1.19961e+01
3	8	12.0009	4.35052e-06	5.27878e-05	1.21337e+01
3	10	15.0011	4.94124e-06	6.01605e-05	1.21752e+01

TABLE VIII. Local energy according to the number of dimensions for 2 particles with interaction using Gibbs sampling. The energies shown have been produced using respectively η =1e-7 and $\eta = 1.0$ with 4 hidden nodes.

η	d	E_L[a.u.]	σ	σ_b	σ_b/σ
1e-7	1	10.1804	3.79821e-01	1.04829e+00	2.75995e+00
	2	2.8871	4.48613e-04	2.92050e-03	6.51007e+00
	3	3.5650	6.34314e-05	8.07111e-04	1.27242e+01
1.0	1	-	-	-	-
	2	2.9803	2.43789e-04	2.91744e-03	1.19671e+01
	3	3.5951	5.99514e-05	6.42739e-04	1.07210e+01

while $\eta = 1.0$ resulted in the best fit for the analytical value of the local energy with the smallest blocking error. We have also used 4 hidden nodes. These were obtained from figure 12, where it is quite clear that 4 hidden nodes results in the best approximation to the local energy for 2 particles in 2 dimensions, with a small blocking error. We can see that the expected energy in 2 dimensions is roughly equal to 3 a.u. with a blocking error $\sigma_b \sim$ 2.92e-3 for η =1e-7. Comparing this to $\eta = 1.0$, where the local energy is nearly equal to 3 a.u., the ratio between the blocking error and the biased error appears smaller for η =1e-7. We were however not able to obtain a reasonable, convergent energy for two particles interacting in one dimension with $\eta = 1.0$. Since the distance between the particles is only defined in one dimension, then the energy is solely dependent on the separation in this one-dimensional space. If that separation becomes small enough then the interaction energy has the potential of divergence.

The comparison of the local energies provided by the three solvers for two interacting particles can be seen in table IX. This was run using 4 hidden nodes and $\eta = 0.01$, the learning rate which was proven best for the brute force method. Here, Gibbs sampling appears stronger in its approximation to E_L for 2 particles in 2 dimensions

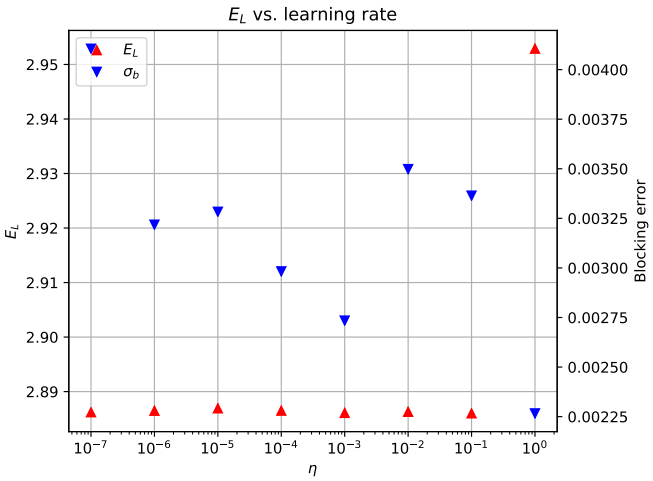

FIG. 11. Local energy dependence on learning rates using Gibbs sampling with interaction between 2 particles in 2D. This was run using 2^{19} Metropolis steps, for 100 RBM cycles, with 2 hidden nodes and $\sigma = 1.0$.

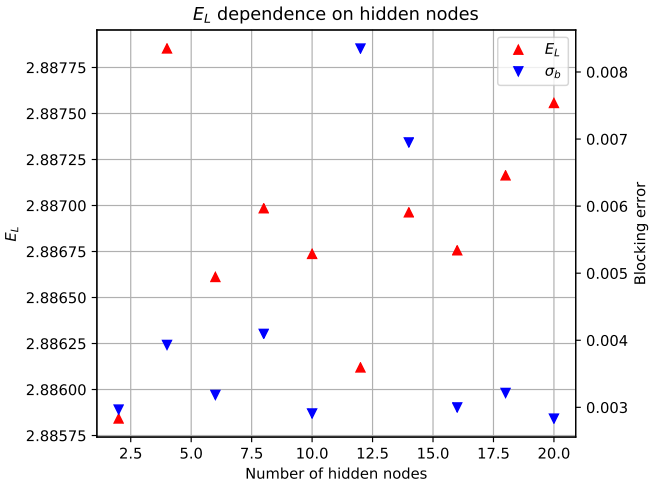

FIG. 12. Local energy dependence on number of hidden nodes using Gibbs sampling with interaction between 2 particles in 2D. This was run using 2^{19} Metropolis steps, for 100 RBM cycles, with 2 hidden nodes and $\sigma = 1.0$.

compared to that of the brute force method and importance sampling. We also see that the energies given in one dimension are all different. Since we do not know the exact analytical value in one dimension we don't have a way of approximating which solver is best in one dimension. Nor do we have an exact analytical value in 3 dimensions, but here the solvers appear a bit more consistent and lie closer to each other compared to the energies in one dimension.

TABLE IX. Local energy for 2 particles in multiple dimensions, given by brute force, importance sampling and Gibbs sampling. This was run with 4 hidden nodes and with $\eta = 0.01$ for 2^{19} Metropolis steps and 100 RBM cycles.

	d	E_L[a.u.]	σ	σ_b	σ_b/σ
Brute force	1	25.9846	1.09282e+00	2.52776e+00	2.31307e+00
	2	3.2533	3.87140e-04	1.10446e-02	2.85287e+01
	3	3.7974	8.85196e-05	3.63086e-03	4.10176e+01
Importance	1	17.9950	1.26751e-04	5.04460e-03	1.32750e-02
	2	3.7816	5.22845e-04	1.69175e-02	3.23565e+01
	3	4.1357	1.26751e-04	5.04460e-03	3.97992e+01
Gibbs	1	12.6715	1.09282e+00	6.13071e-01	4.10135e-01
	2	2.8863	2.84045e-04	3.58587e-03	1.26243e+01
	3	3.5647	6.40747e-05	8.26195e-04	1.28942e+01

IV. Discussion

The initialization of the weights clearly has an effect on the local energy produced. We have here tested two initializations of the weights, either a uniform or normal distribution. The uniform distribution allows us to initialize weights that are symmetric about zero and within a certain interval $(-a, a)$, while the normal distribution allows us to initialize weights with a mean about zero and a standard deviation equal to a. We found that the distribution which leads to the least blocking error is that of the normal distribution with $\sigma = 0.01$. The local energies given by a uniform distribution also result in decent approximations to the analytical local energy, but with a bit greater blocking error as can be seen in table II. The optimal learning rate for the brute force method, when looking at 1 particle in 1D, was found to be $\eta = 0.01$, while the optimal number of hidden nodes was found to be 2.

Looking at Gibbs sampling, the approximations of the local energies to the analytical ones also appear reasonable in all dimensions, although the blocking errors are a bit, about an order or so, higher than the blocking errors produced from the brute force method. Compared to importance, the blocking errors are about the same order. From figure 12 it was found that the optimal number of hidden nodes for a particle in one dimension was 2, while the optimal learning rate was found to be $\eta = 0.01$ from figure 11, same as for the brute force method.

Now, the hidden nodes might be thought of as corresponding to the different spin values the particles can take [1], with two hidden nodes for every particle. For two particles, there are 4 possible spin combinations and thus it is not surprising that the optimal number of hidden nodes in this case was found to be 4.

Furthermore, importance sampling was found to yield more precise energy measurements compared to brute force for the same number of metropolis steps, as seen in Figure 8. This is to be expected because, as discussed in section II E, the importance sampling algorithm is more efficient at mapping out the wave function, given the same number of steps (due to rejecting fewer proposals).

As mentioned section III D, the local energy measurements diverge for 2 particles in one dimension for $\eta = 1.0$. To see why this might be the case, consider the expression for local energy given in Eq (A.4). Specifically, consider the interacting term

$$E_{L_I} = \frac{1}{\Psi_T}\left[\sum_{p<q}\frac{1}{r_{pq}}\right]\Psi_T \tag{33}$$

Where

$$r_{pq} \equiv |\mathbf{r_p} - \mathbf{r_q}| \tag{34}$$

$$= \sqrt{(r_{1_p} - r_{1q})^2 + (r_{2_p} - r_{2q})^2 + ... + (r_{D_p} - r_{Dq})^2} \tag{35}$$

Here r_{1_p} denotes the first coordinate of particle p, and D is the number of dimensions of the system. We see that as r_{pq} goes to zero, E_{L_I} goes to infinity. We therefore have a pole at $r_{pq} = 0$ on the hypersurface defined by $E_L(\mathbf{X}, \mathbf{h}, \alpha)$, regardless of the dimensionality of the system.

Now, when we are doing stochastic gradient descent, we do so while holding $\mathbf{X}$ and $\mathbf{h}$ fixed. However, the starting point on the this hypersurface still depends on the initial value of $\mathbf{h}$, but more importantly, also on the initial values of $\mathbf{X}$. Therefore, if the distance between the particles is very small, we will start our descent close to the pole and with a correspondingly high value of E_L. We might then get a NaN as our energy measurement, and therefore not be able to perform the gradient descent.

The number of dimensions determine how likely it is that the starting position is close to the pole. This is most easily demonstrated in the case where we initialize the visible nodes with a uniform distribution, but also holds for the normal distribution. Say we set the uniform distribution to $(-a, a)$.

Now, consider the probability of the particles being separated by a distance of e.g. $r_{pq} = 2(a/10)$ or less. In one, two and three dimensions, this probability is

$$P\left(r_{pq} < \frac{2a}{10}\middle|1D\right) = \frac{2a/10}{2a} = \frac{1}{10} \tag{36}$$

$$P\left(r_{pq} < \frac{2a}{10}\middle|2D\right) = \frac{\pi\left[2a/10\right]^2}{\pi\left[2a\right]^2} = \frac{1}{100} \tag{37}$$

$$P\left(r_{pq} < \frac{2a}{10}\middle|3D\right) = \frac{(4\pi/3)\left[2a/10\right]^3}{(4\pi/3)\left[2a\right]^3} = \frac{1}{1000} \tag{38}$$

Which shows that the probability of the particles being initialized within a certain distance of each other decreases with the number of dimensions. This could also have been demonstrated using the normal distribution.

Therefore, a possible explanation for the values of E_L given in table III D for $\eta = 1.0$ is that our normal distribution, which is highly localized around $r_{pq} = 0$, gives a larger probability for the starting position of the gradient descent to be located close to the pole in a one dimensional system as compared to two and three dimensions.

However, choosing a different learning rate no longer gives a divergence in E_L in the one dimensional interacting case with two particles. We currently have no explanation for this.

V. Conclusion

We have throughout this report studied the quantum many-body problem by the use of a restricted Boltzmann machine (RBM). We have here represented the ground state of a system through a neural quantum state wave function, once the local energy has been minimized. The minimization has been achieved by reinforcement learning and optimisation of weights and biases. We have included three different solvers, i.e. the brute force method, importance sampling and Gibbs sampling, to minimize the energy. It was found that all three solvers gave reasonable approximations to the analytical local energy in both one, two and three dimensions. This also applies when we are looking at multiple particles. When including interaction between the particles it was shown that Gibbs sampling gave the best approximation of the local energy for two particles in two dimensions. However, neither of the solvers gave close-lying energies to each other. It was also found that the energy becomes divergent when including interaction between two particles in one dimension, for certain learning rates.

The reason for this is not clear to us, and further improvements could include an investigation of this phenomenon.

References

[1] G. Carleo and M. Troyer. "Solving the quantum many-body problem with artificial neural networks." eng. In: *Science (New York, N.Y.)* 355.6325 (2017), pp. 602–606. ISSN: 00368075. URL: http://search.proquest.com/docview/1867546250/.

[2] E. Christensen and H. D. Fossheim. *Project 1: Simulation of hard-sphere Bose gas using Variational Monte Carlo.* Apr. 2020. URL: https://github.com/elisabethchr/FYS4411/blob/master/Project1/report/Variational_Monte_Carlo_report.pdf.

[3] D. J. Griffiths and D. Schroeter. *Introduction to Quantum Mechanics.* 3rd ed. Cambridge University Press, 2018.

[4] M. Hjorth-Jensen. *Computational Physics Lecture Notes Fall 2015.* Aug. 2015. URL: https://github.com/CompPhysics/ComputationalPhysics2/blob/gh-pages/doc/Literature/lectures2015.pdf.

[5] M. Hjorth-Jensen. *From Variational Monte Carlo to Boltzmann Machines and Machine Learning. Notebook 2: Boltzmann Machines.* July 2019. URL: https://github.com/CompPhysics/ComputationalPhysics2/blob/gh-pages/doc/pub/notebook2/pdf/notebook2-print.pdf.

[6] J. Melchior, N. Wang, and L. Wiskott. "Gaussian-binary restricted Boltzmann machines for modeling natural image statistics". In: *PLOS ONE* 12.2 (Feb. 2017), pp. 1–24. DOI: 10.1371/journal.pone.0171015. URL: https://doi.org/10.1371/journal.pone.0171015.

[7] M. Taut. "Two electrons in a homogeneous magnetic field: particular analytical solutions". In: *Journal of Physics A: Mathematical and General* 27.3 (Feb. 1994), pp. 1045–1055. DOI: 10.1088/0305-4470/27/3/040. URL: https://doi.org/10.1088%2F0305-4470%2F27%2F3%2F040.

Appendices

A. Local Energy

The local energy can be expressed as

$$E_L = \frac{1}{\Psi_T}\hat{H}\Psi_T \tag{A.1}$$

where $\hat{H}$ is the Hamiltonian for a harmonic oscillator involving interaction between particles, and is expressed as

$$\hat{H} = \sum_{p}^{P}\left(-\frac{1}{2}\nabla_p^2 + \frac{1}{2}\omega^2 r_p^2\right) + \sum_{i<j}\frac{1}{r_{pq}} \tag{A.2}$$

As mentioned in [ref theory, Boltzmann machine], we use the marginal distribution of the visible nodes to model the trial wave function. In the case of brute-force and importance sampling, this is done by setting the wave function directly equal to said distribution:

$$\begin{aligned}
\Psi_T(\mathbf{X}) = P(\mathbf{X}) &= \frac{1}{Z}\sum_{\{\mathbf{h}\}} e^{-E(\mathbf{X},\mathbf{h})} \\
&= \sum_{h_1=0}^{1}\cdots\sum_{h_N=0}^{1}\frac{1}{Z}e^{-\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{\sigma^2}+\sum_{j=1}^{N}b_jh_j+\sum_{j,i}^{N,M}\frac{X_i\omega_{ij}h_j}{\sigma^2}} \\
&= \frac{1}{Z}e^{-\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{\sigma^2}}\sum_{h_1=0}^{1}\cdots\sum_{h_N=0}^{1}\prod_{j=1}^{N}e^{b_jh_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}h_j}{\sigma^2}} \\
&= \frac{1}{Z}e^{-\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{\sigma^2}}\sum_{h_1=0}^{1}\cdots\sum_{h_{N-1}=0}^{1}\prod_{j=1}^{N-1}e^{b_jh_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}h_j}{\sigma^2}} \\
&\cdot\left(1+e^{b_N+\sum_{i=1}^{M}\frac{X_i\omega_{iN}}{\sigma^2}}\right) \\
&= \frac{1}{Z}e^{-\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{\sigma^2}}\sum_{h_1=0}^{1}\cdots\sum_{h_{N-2}=0}^{1}\prod_{j=1}^{N-2}e^{b_jh_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}h_j}{\sigma^2}} \\
&\cdot\left(1+e^{b_N+\sum_{i=1}^{M}\frac{X_i\omega_{iN}}{\sigma^2}}\right)\left(1+e^{b_{N-1}+\sum_{i=1}^{M}\frac{X_i\omega_{iN-1}}{\sigma^2}}\right) \\
&= \frac{1}{Z}e^{-\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2}}\prod_{j=1}^{N}(1+e^{b_j+\sum_{i=1}^{M}\frac{X_iw_{ij}}{\sigma^2}})
\end{aligned} \tag{A.3}$$

Here, the RBM parameters $= \mathbf{a}, \mathbf{b}$ and $\mathbf{W}$ correspond to the M visible biases, the N hidden biases and the $M \times N$ weight matrix, respectively.

We may split the local energy in eq. (A.1) into a kinetic, potential and an interacting term:

$$\begin{aligned}
E_L &= \frac{1}{\Psi_T}\left[\sum_{p=1}^{P}(-\frac{1}{2}\nabla_p^2 + \frac{1}{2}\omega^2 r_p^2) + \sum_{p<q}\frac{1}{r_{pq}}\right]\Psi_T \\
&= \underbrace{\frac{1}{\Psi_T}\left[\sum_{k=1}^{M}-\frac{1}{2}\frac{\partial^2}{\partial X_k^2}\right]\Psi_T}_{E_{L_K}} + \underbrace{\frac{1}{\Psi_T}\left[\sum_{p=1}^{P}\frac{1}{2}\omega^2 r_p^2\right]\Psi_T}_{E_{L_P}} \\
&+ \underbrace{\frac{1}{\Psi_T}\left[\sum_{p<q}\frac{1}{r_{pq}}\right]\Psi_T}_{E_{L_I}}
\end{aligned} \tag{A.4}$$

As often is the case, finding a closed-form for the kinetic term is the main difficulty. To do this, we find an expression for $\frac{1}{\Psi_T}\frac{\partial^2\Psi_T}{\partial X_k^2}$ in terms of logarithms so as to simplify the calculations:

$$\begin{aligned}
\frac{\partial^2}{\partial X_k^2}\ln\Psi_T &= \frac{\partial}{\partial X_k}\left(\frac{1}{\Psi_T}\frac{\partial\Psi_T}{\partial X_k}\right) \\
&= -\frac{1}{\Psi_T^2}\left(\frac{\partial\Psi_T}{\partial X_k}\right)^2 + \frac{1}{\Psi_T}\frac{\partial^2\Psi_T}{\partial X_k^2} \\
&= -\left(\frac{\partial}{\partial X_k}\ln\Psi_T\right)^2 + \frac{1}{\Psi_T}\frac{\partial^2\Psi_T}{\partial X_k^2}
\end{aligned} \tag{A.5}$$

And so

$$\frac{1}{\Psi_T}\frac{\partial^2\Psi_T}{\partial X_k^2} = \underbrace{\left(\frac{\partial}{\partial X_k}\ln\Psi_T\right)^2}_{(i)} + \underbrace{\frac{\partial^2}{\partial X_k^2}\ln\Psi_T}_{(ii)} \tag{A.6}$$

We therefore need to find $\frac{\partial}{\partial X_k}\ln\Psi_T$ and $\frac{\partial^2}{\partial X_k^2}\ln\Psi_T$. From eq. A.3, we have

$$\begin{aligned}
\ln\Psi_T &= -\ln Z - \sum_{i=1}^{M}\frac{(X_i-a_i)}{2\sigma^2} \\
&+ \ln\left[\prod_{j=1}^{N}\left(1+e^{b_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}}{\sigma^2}}\right)\right] \\
&= -\ln Z - \sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2} + \sum_{j=1}^{N}\ln\left(1+e^{b_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}}{\sigma^2}}\right)
\end{aligned} \tag{A.7}$$

Because the partition function Z is a constant, we get

$$\begin{aligned}
\frac{\partial}{\partial X_k}\ln\Psi_T &= -\frac{(X_k-a_k)}{\sigma^2} \\
&+ \sum_{j=1}^{N}\left(\frac{1}{1+e^{b_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}}{\sigma^2}}}e^{b_j+\sum_{i=1}^{M}\frac{X_i\omega_{ij}}{\sigma^2}}\cdot\frac{\omega_{kj}}{\sigma^2}\right)
\end{aligned} \tag{A.8}$$

The logistic function (a sigmoid function) is given by

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x}. \tag{A.9}$$

Now let

$$v_j = b_j + \sum_{i=1}^{M} \frac{X_i \omega_{ij}}{\sigma^2} \tag{A.10}$$

We then get

$$(i) = \left(\frac{\partial}{\partial X_k} \ln \Psi_T \right)^2 = \left(-\frac{(X_k - a_k)}{\sigma^2} + \sum_{j=1}^{N} \frac{\omega_{kj}}{\sigma^2} S(v_j) \right)^2 \tag{A.11}$$

From this, we realize that we need to differentiate the logistic function in order to calculate (ii) in eq. (A.6):

$$\begin{aligned} \frac{\partial}{\partial X_k} S(v_j) &= \frac{\partial}{\partial X_k} (1 + e^{-v_j})^{-1} \\ &= \frac{\omega_{kj}}{\sigma^2} \frac{e^{-v_j}}{(1 + e^{-v_j})^2} \\ &= \frac{\omega_{kj}}{\sigma^2} S(v_j) S(-v_j) \end{aligned} \tag{A.12}$$

Using this yields

$$(ii) = \frac{\partial^2}{\partial X_k^2} \ln \Psi_T = -\frac{1}{\sigma^2} + \sum_{j=1}^{N} \frac{\omega_{kj}^2}{\sigma^4} S(v_j) S(-v_j) \tag{A.13}$$

Summing over all particles gives us the kinetic term

$$\begin{aligned} E_{L_K} &= \frac{1}{\Psi_T} \left(\sum_{k=1}^{M} -\frac{1}{2} \frac{\partial^2}{\partial X_k^2} \right) \Psi_T = -\frac{1}{2} \sum_{k=1}^{M} \frac{1}{\Psi_T} \frac{\partial^2}{\partial X_k^2} \Psi_T \\ &= -\frac{1}{2} \sum_{k=1}^{M} \left[\underbrace{\left(\frac{\partial}{\partial X_k} \ln \Psi_T \right)^2}_{(i)} + \underbrace{\frac{\partial^2}{\partial X_k^2} \ln \Psi_T}_{(ii)} \right] \end{aligned} \tag{A.14}$$

Next, we insert for equations (A.11) and (A.13):

$$\begin{aligned} E_{L_K} = &-\frac{1}{2} \sum_{k=1}^{M} \left\{ -\frac{(X_k - a_k)}{\sigma^2} + \sum_{j=1}^{N} \frac{\omega_{kj}}{\sigma^2} S(v_j) \right\}^2 \\ &- \frac{1}{2} \sum_{k=1}^{M} \left\{ -\frac{1}{\sigma^2} + \sum_{j=1}^{N} \frac{\omega_{kj}^2}{\sigma^4} S(v_j) S(-v_j) \right\} \end{aligned} \tag{A.15}$$

By expanding this and including the potential- and the interaction term, we get the following closed-form expression for local energy in brute-force and importance sampling:

$$\begin{aligned} E_L = &-\frac{1}{2} \sum_{k=1}^{M} \left\{ \frac{(X_k - a_k)^2}{\sigma^4} - \frac{2(X_k - a_k)}{\sigma^2} \sum_{j=1}^{N} \frac{\omega_{kj}}{\sigma^2} S(v_j) \right. \\ &\left. + \left[\sum_{j=1}^{N} \frac{\omega_{kj}}{\sigma^2} S(v_j) \right]^2 \right\} - \frac{1}{2} \sum_{k=1}^{M} \left\{ \sum_{j=1}^{N} \frac{\omega_{kj}^2}{\sigma^4} S(v_j) S(-v_j) \right\} \\ &+ \frac{M}{2\sigma^2} + E_{L_P} + E_{L_I} \end{aligned} \tag{A.16}$$

A. Local energy in Gibbs sampling

Contrary to the brute-force and metropolis sampling algorithms, which use the marginal distribution of $\mathbf{X}$ as the trial wave function,

$$\Psi_T(\mathbf{X}) = P(\mathbf{X}) \tag{A.17}$$

, the Gibbs sampling algorithm sets the trial wave function equal to the square root of this distribution:

$$\Psi_T(\mathbf{X}) = \sqrt{P(\mathbf{X})}. \tag{A.18}$$

We have to account for this in our derivation of the closed-form expression for E_L. From eq. (A.4), we see that the wave function is cancelled in both the potential and interaction terms. We therefore only have to adjust the kinetic term. Specifically, we need to adjust the terms (i) and (ii) in eq. (A.6):

$$\begin{aligned} (i) = \left(\frac{\partial}{\partial X_k} \ln \Psi_T \right)^2 &\rightarrow \left(\frac{\partial}{\partial X_k} \ln \sqrt{\Psi_T} \right)^2 \\ &= \frac{1}{4} \left(\frac{\partial}{\partial X_k} \ln \Psi_T \right)^2 = \frac{1}{4} \cdot (i) \end{aligned} \tag{A.19}$$

$$\begin{aligned} (ii) = \frac{\partial^2}{\partial X_k^2} \ln \Psi_T \rightarrow \frac{\partial^2}{\partial X_k^2} \ln \sqrt{\Psi_T} &= \frac{1}{2} \frac{\partial^2}{\partial X_k^2} \ln \Psi_T \\ &= \frac{1}{2} \cdot (ii) \end{aligned} \tag{A.20}$$

We therefore need to trace which of the terms in eq. (A.16) stem from (i) and which stem from (ii) and multiply them by $\frac{1}{4}$ and $\frac{1}{2}$ respectively. Doing this gives us the following closed-form expression for the local energy in Gibbs sampling:

$$\begin{aligned} E_L = &-\frac{1}{8} \sum_{k=1}^{M} \left\{ \frac{(X_k - a_k)^2}{\sigma^4} - \frac{2(X_k - a_k)}{\sigma^2} \sum_{j=1}^{N} \frac{\omega_{kj}}{\sigma^2} S(v_j) \right. \\ &\left. + \left[\sum_{j=1}^{N} \frac{\omega_{kj}}{\sigma^2} S(v_j) \right]^2 \right\} - \frac{1}{4} \sum_{k=1}^{M} \left\{ \sum_{j=1}^{N} \frac{\omega_{kj}^2}{\sigma^4} S(v_j) S(-v_j) \right\} \\ &+ \frac{M}{4\sigma^2} + E_{L_P} + E_{L_I} \end{aligned} \tag{A.21}$$

B. Gradient Descent

As is shown in [2], the partial derivative of the local energy with respect to one of the RBM parameters is given by

$$G_i = \frac{\partial \langle E_L \rangle}{\partial \alpha_i} = 2\langle E_L \frac{1}{\Psi_T}\frac{\partial \Psi_T}{\partial \alpha_i}\rangle - 2\langle E_L\rangle\langle \frac{1}{\Psi_T}\frac{\partial \Psi_T}{\partial \alpha_i}\rangle. \tag{B.1}$$

Thus we need an analytical expression for $\partial\Psi_T/\partial\alpha_i$ where $\alpha_i = a_1, ..., a_M, b_1, ..., b_N, \omega_{11}, ..., \omega_{MN}$. Recalling the expression for the wave function:

$$\Psi_T(\mathbf{X}) = e^{-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}} \prod_{j=1}^{N}\left(1 + e^{b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}}\right) \tag{B.2}$$

We get

$$\boxed{\frac{\partial \Psi_T}{\partial a_k} = \frac{X_k - a_k}{\sigma^2}\Psi_T} \tag{B.3}$$

Differentiating with respect to b_l gives

$$\begin{aligned}\frac{\partial \Psi_T}{\partial b_l} &= e^{-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}} \\ &\cdot \frac{\partial}{\partial b_l}\left\{(1 + e^{v_1}) \cdot ... \cdot (1 + e^{v_l}) \cdot ... \cdot (1 + e^{v_N})\right\}\end{aligned} \tag{B.4}$$

Only $(1 + e^{v_l})$ depends on b_l and so after differentiation, we are left with

$$\begin{aligned}\frac{\partial \Psi_T}{\partial b_l} &= e^{-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}} \\ &\cdot \frac{\partial}{\partial b_l}\left\{(1 + e^{v_1}) \cdot ... \cdot (e^{v_l}) \cdot ... \cdot (1 + e^{v_N})\right\} \\ &= e^{-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}} \prod_{j=1}^{N}\left(1 + e^{b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}}\right)\frac{e^{v_l}}{1 + e^{v_l}} \\ &= \Psi_T \frac{e^{v_l}}{1 + e^{v_l}}\end{aligned} \tag{B.5}$$

and so

$$\boxed{\frac{\partial \Psi_T}{\partial b_l} = S(v_l)\Psi_T.} \tag{B.6}$$

Next, we calculate $\frac{\partial \Psi_T}{\partial \omega_{kl}}$:

$$\frac{\partial \Psi_T}{\partial \omega_{kl}} = e^{-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}} \frac{\partial}{\partial \omega_{kl}} \prod_{j=1}^{N}\left(1 + e^{b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}}\right) \tag{B.7}$$

We can apply the same reasoning as when calculating the derivative w.r.t. b_l, but we need to account for the extra factor in the exponent when differentiating:

$$\begin{aligned}\frac{\partial \Psi_T}{\partial \omega_{kl}} &= e^{-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}} \\ &\cdot \prod_{j=1}^{N}\left(1 + e^{b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}}\right)\frac{\frac{X_k}{\sigma^2} e^{b_l + \sum_i^M \frac{X_i \omega_{il}}{\sigma^2}}}{1 + e^{v_l}}\end{aligned}$$

And thus

$$\boxed{\frac{\partial \Psi_T}{\partial \omega_{kl}} = \frac{X_k}{\sigma^2} S(v_l)\Psi_T} \tag{B.8}$$

A. Gradient descent in Gibbs sampling

As was the case for local energy, we need to replace Ψ_T with $\sqrt{\Psi_T}$. The replacements in this case are however a lot easier to make:

$$\frac{\partial \Psi_T}{\partial a_k} \to \frac{\partial \sqrt{\Psi_T}}{\partial a_k} = \frac{1}{2}\frac{X_k - a_k}{\sigma^2}\Psi_T \tag{B.9}$$

$$\frac{\partial \Psi_T}{\partial b_l} \to \frac{\partial \sqrt{\Psi_T}}{\partial b_l} = \frac{1}{2} S(v_l)\Psi_T \tag{B.10}$$

$$\frac{\partial \Psi_T}{\partial \omega_{kl}} \to \frac{\partial \sqrt{\Psi_T}}{\partial \omega_{kl}} = \frac{1}{2}\frac{X_k}{\sigma^2} S(v_l)\Psi_T \tag{B.11}$$

C. The Variational Principle

When we are dealing with systems for which we cannot find the exact wave function, the variational principle states that

$$E_{gs} \leq \langle \Psi_T | \hat{H} | \Psi_T \rangle = \langle E_L \rangle \tag{C.1}$$

To see this, consider first the Hamiltonian $\hat{H}$. Since $\hat{H}$ is always Hermitian, the unknown eigenfunctions of $\hat{H}$ form a complete set [3]. The trial wave function can therefore be expressed as a linear combination of them:

$$\Psi_T = \sum_n c_n \psi_n \tag{C.2}$$

Assuming that these eigenfunctions have been orthonormalized, we may rewrite the expected local energy as

$$\begin{aligned}\langle E_L \rangle &= \left\langle \sum_n c_n \psi_n \middle| \hat{H} \middle| \sum_m c_m \psi_m \right\rangle \\ &= \sum_n \sum_m c_n^* c_m \langle \psi_n | \hat{H} | \psi_m \rangle \\ &= \sum_n \sum_m c_n^* c_m E_m \langle \psi_n \,|\, \psi_m \rangle = \sum_n |c_n|^2 E_n.\end{aligned} \tag{C.3}$$

By definition, the ground state is the lowest energy the system can take and so

$$E_{gs} = \sum_n |c_n|^2 E_{gs} \leq \sum_n |c_n|^2 E_n = \langle \Psi_T | \hat{H} | \Psi_T \rangle = \langle E_L \rangle \tag{C.4}$$