

Application of restricted Boltzmann machines to trapped interacting electrons

Adrian Martinsen Kleven

Autumn 2020

Contents

1 Abstract	3
2 Introduction	3
3 Theory	3
3.1 The wave function	4
3.2 The Hamiltonian	4
3.3 The Local energy	4
3.4 Analytic energies	5
3.4.1 Non- interacting	5
3.4.2 Interacting	5
4 Implementation	5
4.1 The Wave function	5
4.2 Optimization	5
4.2.1 Cost function gradient	5
4.2.2 Stochastic Gradient Descent (SGD)	6
4.2.3 Adaptive Momentum estimation (ADAM)	6
4.3 The Metropolis Algorithm	6
4.3.1 Metropolis- Hastings	6
4.4 Initialization	7
4.5 Post Analysis and Error Estimation	7
4.6 Parallelization	7
5 Results	7
5.1 Brute force v. Importance sampling	7
5.1.1 Adam v. SGD	8
5.2 Non- interacting electrons	9
5.3 Interacting electrons	10
5.4 Electron Locations	11
6 Discussion	13
6.1 Loss of significance	13
6.2 Divide by zero	14
6.3 The nature of an upper bound	14
6.4 Bug	14
7 Conclusion	15
8 Appendix A	16
9 Appendix B	16

List of Figures

1 Brute force v. Importance sampling	8
2 SGD v. ADAM	8
3 SGD v. ADAM high learning rate	9
4 Parameter sweep for two non- interacting electrons	10
5 Parameter sweep for two interacting electrons	11
6 Normalized radial distribution of singlet and non- interacting doublet	12

7	Spatial distribution of electrons	13
8	numerically unstable run- electron positions	15

List of Tables

1	Reproducing analytical results for trivial cases	10
---	--	----

1 Abstract

In this paper we examine systems of one and two trapped electrons in two dimensions using a restricted Boltzmann machine to represent the trial wave function. The ground state is approximated using variational Monte- Carlo methods and compared to analytical expressions for both the interacting and the non- interacting case provided by M. Taut. [5]. We examine the spatial distribution of electrons for the two- dimensional systems and the convergence rates for stochastic gradient descent (SGD) and adaptive momentum gradient descent (ADAM). ADAM was shown to have far greater convergence rates and so was the optimizer of choice for numerical experiments. The convergence rates and errors for importance sampling and brute force sampling were compared for the same number of steps, and importance sampling, being far superior was used for every simulation. The local energy expectation values for single- electron systems were found to be in agreement with analytical results with standard deviations of order $\mathcal{O}(10^{-6}a.u.)$. The energy for two electrons in one dimension was calculated to be 0.97796656 with a standard deviation of the order $\mathcal{O}(10^{-6}a.u.)$. A large discrepancy from the analytic value of 1.0. Finally the energy for two interacting electrons was calculated to be 3.04450717a.u. with a standard deviation of 0.01146434a.u., slightly above the analytic value of 3.0a.u..

2 Introduction

The mathematical complexity of quantum mechanical many- body systems has brought on the necessity for numerical solutions. The quantum wave function has been successfully modelled using a restricted Boltzmann machine (RBM) in many- body lattice spin- systems [1]. The purpose of this paper is to examine its applicability in estimating the upper bound for the ground state energy of interacting two- electron systems confined in a harmonic oscillator potential. To do this we perform numerical experiments of systems with known solutions and make an estimation of its accuracy.

First we cover the bare minimum of theory for transforming the RBM joint probability distribution to something representing a wave function. We then cover the relevant expressions for computations in the model and the analytical values for the ground state energies. Because the exclusion principle is not taken into account, we only examine systems for at most two electrons. We then cover the specifics for this particular implementation including the choice of sampler and optimization scheme. Ending with the results of the numerical experiments and a discussion of the results, the numerical stability, and the potential of the program.

All energies are given in atomic units (a.u.) and length scales in fractions of the Bohr radius ($2 \cdot a_0$).

The program was implemented using c++ with post analysis, blocking, plotting being done via Python. All relevant files, including selected runs for testing can be found at [Github](#).

3 Theory

We can approximate the upper bound for the ground state of a quantum mechanical system by using the variational principle. We propose a trial wave function with some variational parameters, then optimize the energy- expectation value with respect to those parameters. The Expectation value for the energy is given by

$$\langle E \rangle = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}. \quad (1)$$

The probability distribution for the locations of the electrons are estimated using the Metropolis and Metropolis- Hastings algorithms. A discussion of these can be found here [3].

3.1 The wave function

The quantum mechanical wave function will be represented by a RBM. The joint probability distribution for a RBM is defined by the Boltzmann distribution

$$F_{rbm}(\mathbf{X}, \mathbf{H}) = \frac{1}{Z} e^{-\frac{1}{T_0} E(\mathbf{X}, \mathbf{H})} \quad (2)$$

where Z is the partition function

$$Z = \iint \frac{1}{Z} e^{-\frac{1}{T_0} E(\mathbf{x}, \mathbf{h})} d\mathbf{x} d\mathbf{h}. \quad (3)$$

We set $T_0 = 1$ for simplicity. Here, E is the configuration energy of the network. The configuration energy expression is unique to different problems. The one of interest to us is the Gaussian- Binary RBM whose joint probability distribution is given by equations (2), (3) and

$$E(\mathbf{X}, \mathbf{H}) = \frac{\|\mathbf{X} - \mathbf{a}\|^2}{2\sigma^2} - \mathbf{b}^T \mathbf{H} - \frac{\mathbf{X}^T \mathbf{W} \mathbf{H}}{\sigma^2} \quad (4)$$

where \mathbf{X} are the visible nodes corresponding to the coordinates of each particle, \mathbf{H} are the hidden nodes. \mathbf{a} and \mathbf{b} are the visible and hidden biases respectively. \mathbf{W} is the matrix of weights connecting nodes between layers. To elucidate the connection between the joint probability distribution and the wave function, the hidden nodes are summed over, giving a distribution solely dependent on \mathbf{X} .

$$\begin{aligned} \Psi(\mathbf{X}) &= \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{X}, \mathbf{h})} \\ &= \frac{1}{Z} \sum_{\{h_j\}} e^{-\sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} + \sum_j^N b_j h_j + \sum_{i,j}^{M,N} \frac{x_i w_{ij} h_j}{\sigma^2}} \\ &= \frac{1}{Z} e^{-\sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2}} \prod_j^N \left(1 + e^{b_j + \sum_i^M \frac{x_i w_{ij}}{\sigma^2}} \right) \end{aligned} \quad (5)$$

3.2 The Hamiltonian

The Hamiltonian of the system is given by a standard harmonic oscillator Hamiltonian with an included, idealized interaction term

$$\hat{H} = \sum_{i=1}^N \left(-\frac{1}{2} \nabla_i^2 + \frac{1}{2} \omega^2 r_i^2 \right) + \sum_{i < j} \frac{1}{r_{ij}} \quad (6)$$

using natural units ($\hbar = c = e = m_e = 1$).

3.3 The Local energy

The local energy

$$E_L = \frac{1}{\Psi} \hat{\mathbf{H}} \Psi \quad (7)$$

can be determined analytically (see Appendix A), giving the expression (19). The local energy has to be minimized with respect to the network parameters. The gradient for the expectation value of the local energy is given by equation (21) (see Appendix B).

3.4 Analytic energies

The wave function for one electron in a harmonic oscillator potential in two dimensions is given by

$$\phi_{n_x, n_y}(x, y) = A H_{n_x}(\sqrt{\omega}x) H_{n_y}(\sqrt{\omega}y) \exp(-\omega(x^2 + y^2)/2) \quad (8)$$

where A is the normalization and H_{n_x} are the Hermite polynomials corresponding to the principal quantum number n_x . In its ground state, the unperturbed wave function has a local energy of ω .

3.4.1 Non- interacting

The two- electron wave function is given by

$$\Phi(\mathbf{r}_1, \mathbf{r}_2) = \frac{C}{\sqrt{2}} \phi_{0,0}(\mathbf{r}_1) \phi_{0,0}(\mathbf{r}_2) \{ \xi_+(\mathbf{r}_1) \xi_-(\mathbf{r}_2) - \xi_+(\mathbf{r}_2) \xi_-(\mathbf{r}_1) \} \quad (9)$$

where we have ensured anti- symmetry under interchange of electrons by taking the product of the symmetric spatial part and the anti- symmetric spin part. Because the Hamiltonian doesn't couple to the electron spins, the spin- expression will vanish in the local energy calculation.

$$\frac{1}{\Phi(\mathbf{r}_1, \mathbf{r}_2)} \hat{\mathbf{H}}_0 \Phi(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{\phi_{0,0}(\mathbf{r}_1) \phi_{0,0}(\mathbf{r}_2)} \left(\phi_{0,0}(\mathbf{r}_1) \hat{\mathbf{H}}_0 \phi_{0,0}(\mathbf{r}_2) + \phi_{0,0}(\mathbf{r}_2) \hat{\mathbf{H}}_0 \phi_{0,0}(\mathbf{r}_1) \right) \quad (10)$$

$$E_L^0 = 2\omega \quad (11)$$

3.4.2 Interacting

The case for two interacting electrons in two dimensions has been provided by Taut [5] and is given by 3 *a.u* for the Hamiltonian (6).

4 Implementation

4.1 The Wave function

The wave function (5) is partly separable, meaning we can avoid a few calculations by factoring out the parts of the wave function that don't change when the position of one particle is changed, meaning the evaluation in (13) and (14) becomes somewhat faster.

The value of σ in the wave function is set to 1.0 for all simulations.

4.2 Optimization

The optimization of the network parameters was done using gradient descent methods with the local energy as the cost function. The learning rate is denoted by η and the parameters by $\alpha_i = a_1, \dots, a_M, b_1, \dots, b_N, w_{11}, \dots, w_{MN}$.

4.2.1 Cost function gradient

To calculate the Cost function Gradient in equation(21), the program needs to sample three different quantities to obtain their expectation values. These are

$$\left\langle E_L \frac{1}{\Psi} \frac{\partial \Psi}{\partial \alpha_i} \right\rangle, \quad \langle E_L \rangle \quad \text{and} \quad \left\langle \frac{1}{\Psi} \frac{\partial \Psi}{\partial \alpha_i} \right\rangle. \quad (12)$$

The relevant values can be found in Appendix A and Appendix B.

4.2.2 Stochastic Gradient Descent (SGD)

SGD uses a fixed learning rate and the cost function gradient to update the current parameters according to

$$\alpha_i^{(t+1)} \leftarrow \alpha_i^{(t)} - \eta G^{(t)}.$$

4.2.3 Adaptive Momentum estimation (ADAM)

ADAM is a method of optimization that utilizes a running average of previous gradients to inform the current learning step. the parameters $\alpha_i^{(t+1)}$ at optimization step $t + 1$ is updated according to

$$\begin{aligned} m_\alpha^{(t+1)} &\leftarrow \beta_1 m_\alpha^{(t)} + (1 - \beta_1) G^{(t)} \\ v_\alpha^{(t+1)} &\leftarrow \beta_2 v_\alpha^{(t)} + (1 - \beta_2) (G^{(t)})^2 \\ \hat{m}_\alpha &= \frac{m_\alpha^{(t+1)}}{1 - \beta_1^{t+1}} \\ \hat{v}_\alpha &= \frac{v_\alpha^{(t+1)}}{1 - \beta_2^{t+1}} \\ \alpha_i^{(t+1)} &\leftarrow \alpha_i^{(t)} - \eta \frac{\hat{m}_\alpha}{\sqrt{\hat{v}_\alpha} + \epsilon} \end{aligned}$$

with the values $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ chosen for this implementation. ADAM incurs a marginally greater overhead than SGD, so any improvement in convergence will be much more use than the cumulative overhead from just a few hundred cycles.

4.3 The Metropolis Algorithm

The Metropolis algorithm works by choosing one particle at random, adjusting its position then calculating the new value of the wave function that arises. The probability of accepting that move is then calculated from equation (13) where we assume that $T(x \rightarrow x') = T(x' \rightarrow x)$

$$A(x \rightarrow x') = \min \left(1, \frac{|\Psi(x')|^2 T(x \rightarrow x')}{|\Psi(x)|^2 T(x' \rightarrow x)} \right). \quad (13)$$

In this way, the Metropolis algorithm samples from a probability distribution using a biased random walker[3].

4.3.1 Metropolis- Hastings

The Metropolis- Hastings algorithm uses the Greens function solution to the Fokker- Planck equation (15) as the transition probability which gives the acceptance probability

$$A(x \rightarrow x') = \min \left(1, \frac{|\Psi(x')|^2 G(x, x', \delta t)}{|\Psi(x)|^2 G(x', x, \delta t)} \right) \quad (14)$$

$$\frac{\partial P(x, t)}{\partial t} = D \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} - F \right) P(x, t) \quad (15)$$

The Greens function in this case is

$$G(x, x', \delta t) = \frac{1}{(4\pi D \delta t)^{\frac{3}{2}N}} \exp \left(-\frac{(x' - x - D\delta t F(x))^2}{4D\delta t} \right). \quad (16)$$

Particle positions are updated according to

$$x' = x + DF(x)\delta t + \eta\sqrt{\delta t}, \quad (17)$$

the discretized version of the Langevin equation the D in these equations is the diffusion coefficient which the implementation sets to 0.5. η number sampled from a uniform distribution between 0 and 1. $F(x)$ is the quantum force

$$\mathbf{F} = \frac{2\nabla\Psi}{\Psi} \quad (18)$$

[3] whose relevant terms are calculated in Appendix A.

The time- step δt was chosen so as to make the acceptance rate of the proposed moves on average higher than 95%. This meant that $\delta t = 0.01$ for most runs, except for runs very close to the minimum for two interacting electrons, in which case $\delta t = 0.001$.

4.4 Initialization

The parameters a_i were initialized according to a Gaussian distribution with a standard deviation σ_0 usually of the order $10^{-3}a.u..$ The particle positions were initialized with a uniform distribution between -0.5 and 0.5 Bohr radii.

4.5 Post Analysis and Error Estimation

The post analysis involved printing the energy at each sampling stage to file, then analysing it using Python. The error estimates for the expectation value of the local energy were evaluated using Marius Jonsson's Blocking code [2].

4.6 Parallelization

Building a solution for c++ in Visual Studio in x- 64 release mode, automatically vectorizes and parallelizes many loops that don't have dependencies that can't be parallelized. The only loop to be explicitly parallelized in the program was a grid search for the number of hidden nodes and the learning rate for ADAM. This loop was parallelized using OpenMP [4].

5 Results

To get the best results possible, we compare the different methods against each other and chose the ones with the most promising results.

5.1 Brute force v. Importance sampling

In figure 1 we compare brute force sampling and importance sampling with the same number of Metropolis steps, with step sizes and time steps respectively such that the acceptance rate exceeded 95%.

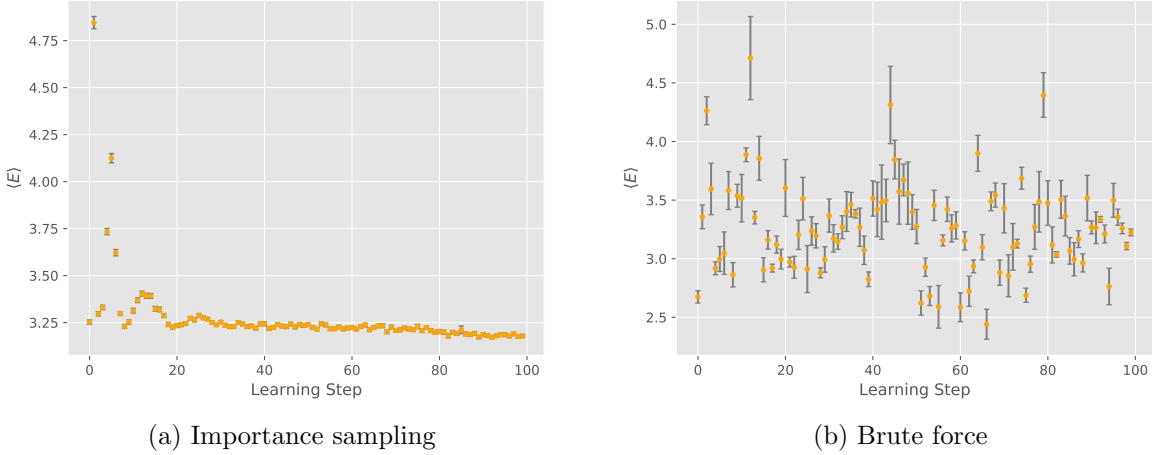


Figure 1: Two Interacting electrons in two dimensions with 6 hidden nodes. Computed using 2^{20} equilibration steps, 2^{20} sampling steps. Expectation values for the local energy (a.u.) as a function of learning step upto 100 ADAM learning steps with a learning rate of 0.2. The expectation values and standard deviations were calculated using blocking.

We see that brute force sampling gives much higher errors and a much lower rate of convergence, including giving much lower estimates than the theoretical minimum of $3a.u.$ for some runs. This is indicative of the system undergoing too few Monte- Carlo cycles. Because of this, all remaining results are calculated using importance sampling.

5.1.1 Adam v. SGD

In figures 2 and 3 we compare the two different implemented methods for optimizing the network parameters. First with a learning rate of 0.01 then 0.1.

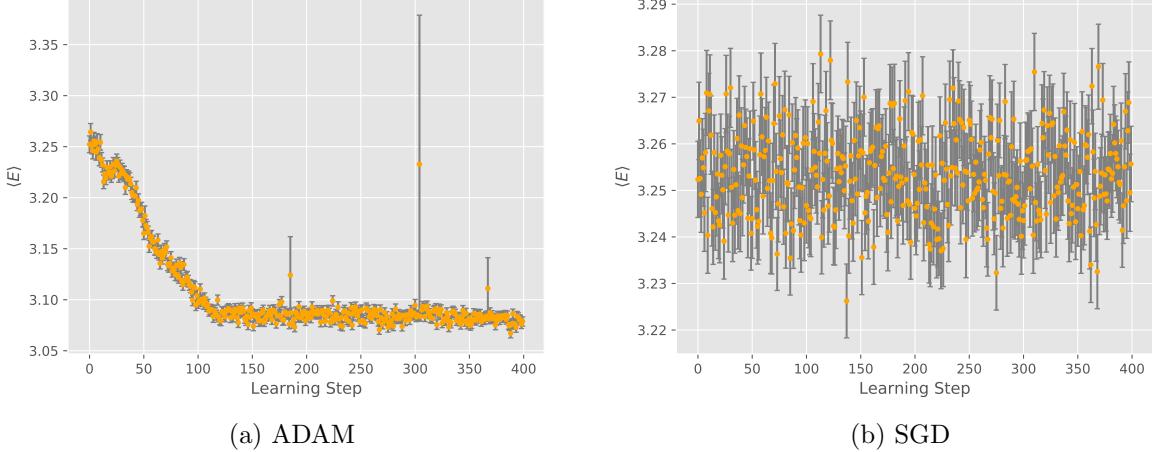


Figure 2: 2 interacting particles in 2 dimensions. Expectation values for the local energy (a.u.) as a function of learning step using a learning rate of 0.01. 3 hidden nodes, 2^{21} metropolis-Hastings steps and 2^{21} equilibration steps.

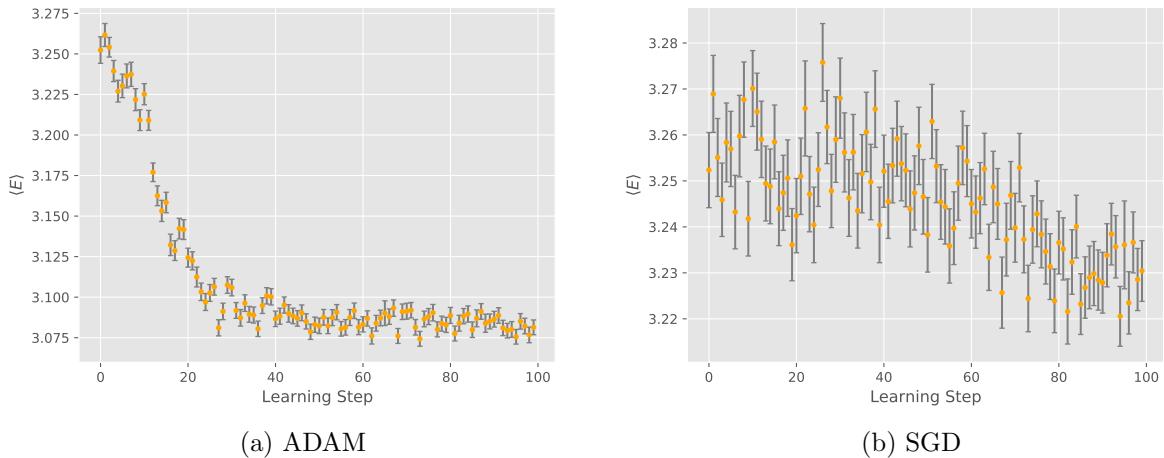


Figure 3: 2 interacting particles in 2 dimensions. Expectation values for the local energy (a.u.) as a function of learning step using a learning rate of 0.1. 3 hidden nodes, 2^{21} metropolis-Hastings steps and 2^{21} equilibration steps.

This isn't an entirely fair comparison since the learning rate plays a much more subtle role with ADAM compared to SGD. It does however highlight the central issue with fixed learning rate schemes. You want a large learning rate for fast convergence, and a small learning rate when you're close to the minimum. Both figures illustrates the point, which is to say that ADAM converges much more rapidly for both learning rates.

There are notable occurrences in figure 2a, namely three runs that had much higher energies and standard deviations compared to surrounding values. The surrounding values are still within one standard deviation of those results and so might be a statistical fluke, but they still stand out as perhaps being warning signs of unaccounted sources of systematic error.

5.2 Non-interacting electrons

The first set of numerical experiments relate to ground state energies for the trivial systems of one or two non-interacting electrons, whose analytic expressions are detailed in section 3.4. Figure 4 shows the calculated ground state energies of two non-interacting electrons for different ADAM learning rates and number of hidden nodes. This is only meant to be illustrative of the general trend, and instructive towards more in depth runs.

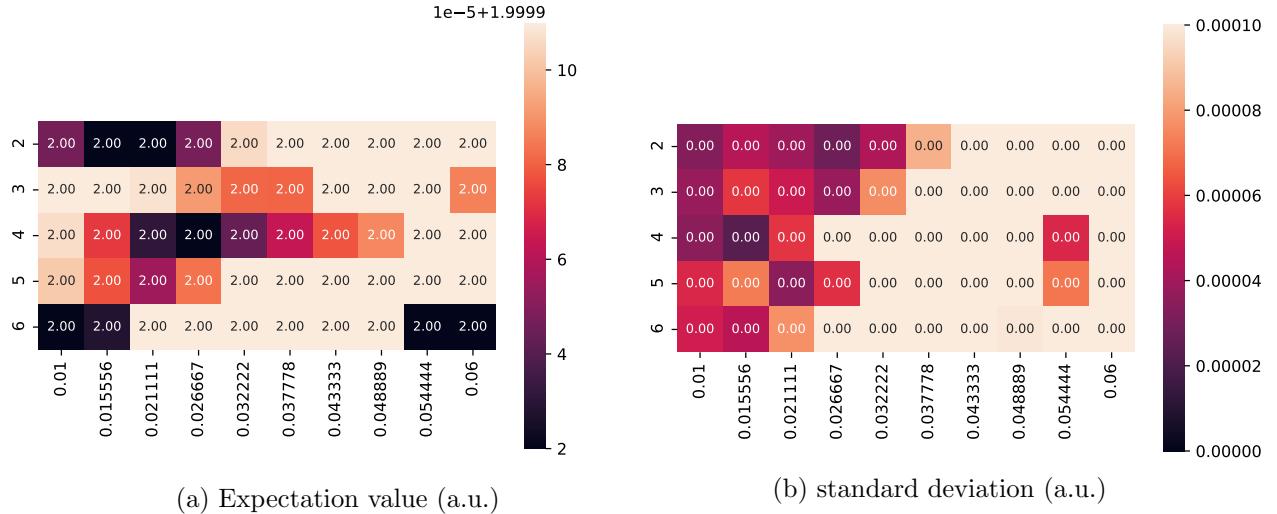


Figure 4: Two non-interacting electrons in two dimensions. Computed using importance sampling with 2^{20} equilibration steps, 2^{20} sampling steps and 50 ADAM learning steps. The number of hidden nodes on the y-axis and the learning rate on the x-axis. Expectation values and standard deviations were computed using blocking on the energies from the final learning step.

The energies don't provide any unambiguous patterns except that lower learning rates seem to perform somewhat better. We see this again in the standard deviations, which have a clear trend towards lower learning rates. Initially the sectors with energies lower than $2a.u.$ are somewhat worrisome, but this seems to be accounted for, with standard deviations being somewhat higher for sectors with dark blue hues, and all standard deviations being above the threshold for bringing energies above $2a.u..$ It may of course also be the case that the energies are lower than $2(a.u.)$ due to some systematic error.

Without extensive searching, the energies for the trivial cases are provided in table 1 below.

Table 1: Expectation value for the local energy and standard deviation for the trivial systems. Using 20 ADAM learning steps with a learning rate of 0.01, 2^{20} equilibration and sampling steps with importance sampling.

Nr. Electrons	Nr. dimensions	$\hat{E}(a.u.)$	$\sigma(a.u.) \times 10^5$	Analytic a.u.
1	1	0.49999599	0.311658852	0.5
1	2	0.99999918	0.568598342	1.0
2	1	0.97796656	0.489544015	1.0
2	2	2.00002012	1.060556	2.0

The 1-electron systems are in agreement with the analytic results to within one standard deviation. So is the 2-electron system in 2 dimensions. Worryingly, the 2-electron system in 1 dimension has a substantial discrepancy with the analytic result.

5.3 Interacting electrons

For the interacting case, we again make a sweep of learning rates and number of hidden nodes, the result is shown in figure 5.

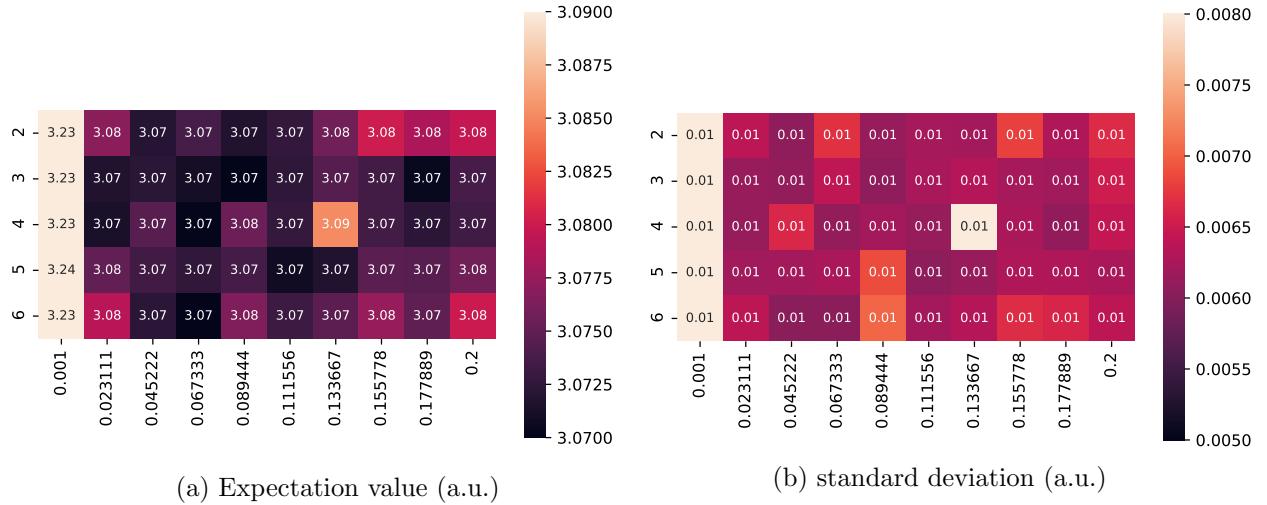


Figure 5: Two interacting electrons in two dimensions. Computed using importance sampling with 2^{20} equilibration steps, 2^{20} sampling steps and 200 ADAM learning steps. The number of hidden nodes on the y-axis and the ADAM- learning rate on the x- axis. Expectation values and standard deviations were computed using blocking on the energies from the final learning step.

We see from figure 5a that a learning rate of 0.001 initially, is not sufficient for the program to converge in the course of 200 ADAM steps. Using this parameter sweep as a starting point and using the network parameters acquired at the final learning step for the learning rate 0.067 and hidden node number 4, successive runs with incrementally smaller learning rates yielded the current best estimate of $3.04450717a.u.$ with a standard deviation of $0.01146434a.u.$ using a final learning rate of 0.00001. Smaller learning rates and more computations could of course be attempted.

5.4 Electron Locations

Another measure of interest is how the electrons are distributed in space, and how that distribution changes from system to system. In figure 6 we see the radial distribution of electrons from the center of the HO potential.

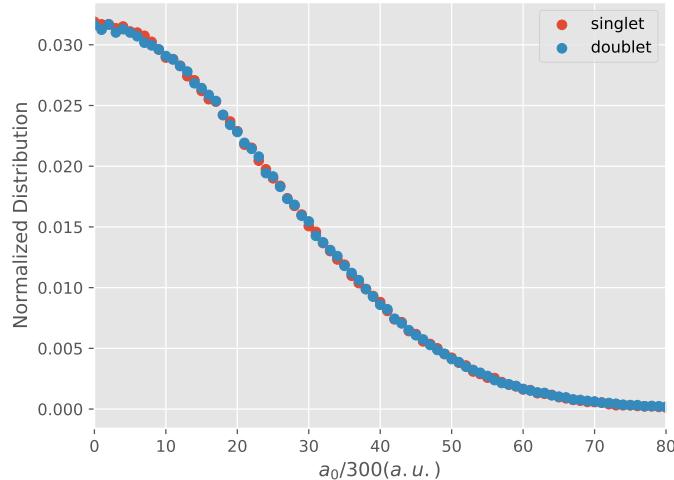


Figure 6: Normalized radial distribution for the singlet and non- interacting doublet states in $1/600$ of a Bohr radius, the natural length scale of the system. The energies are $1.0000a.u.$ and $1.99997a.u.$ for the singlet and doublet respectively.

Because the energy of two electrons is exactly twice that of one electron, and because the Hamiltonian doesn't couple to spin, this is exactly the result we expect to see. The normalized radial distribution for the singlet and doublet states are the same barring a very slight discrepancy toward the peak of the distribution, with very slightly fewer electrons inhabiting the near center of the potential.

The spatial distributions of all 2- dimensional systems can be seen in figure 7.

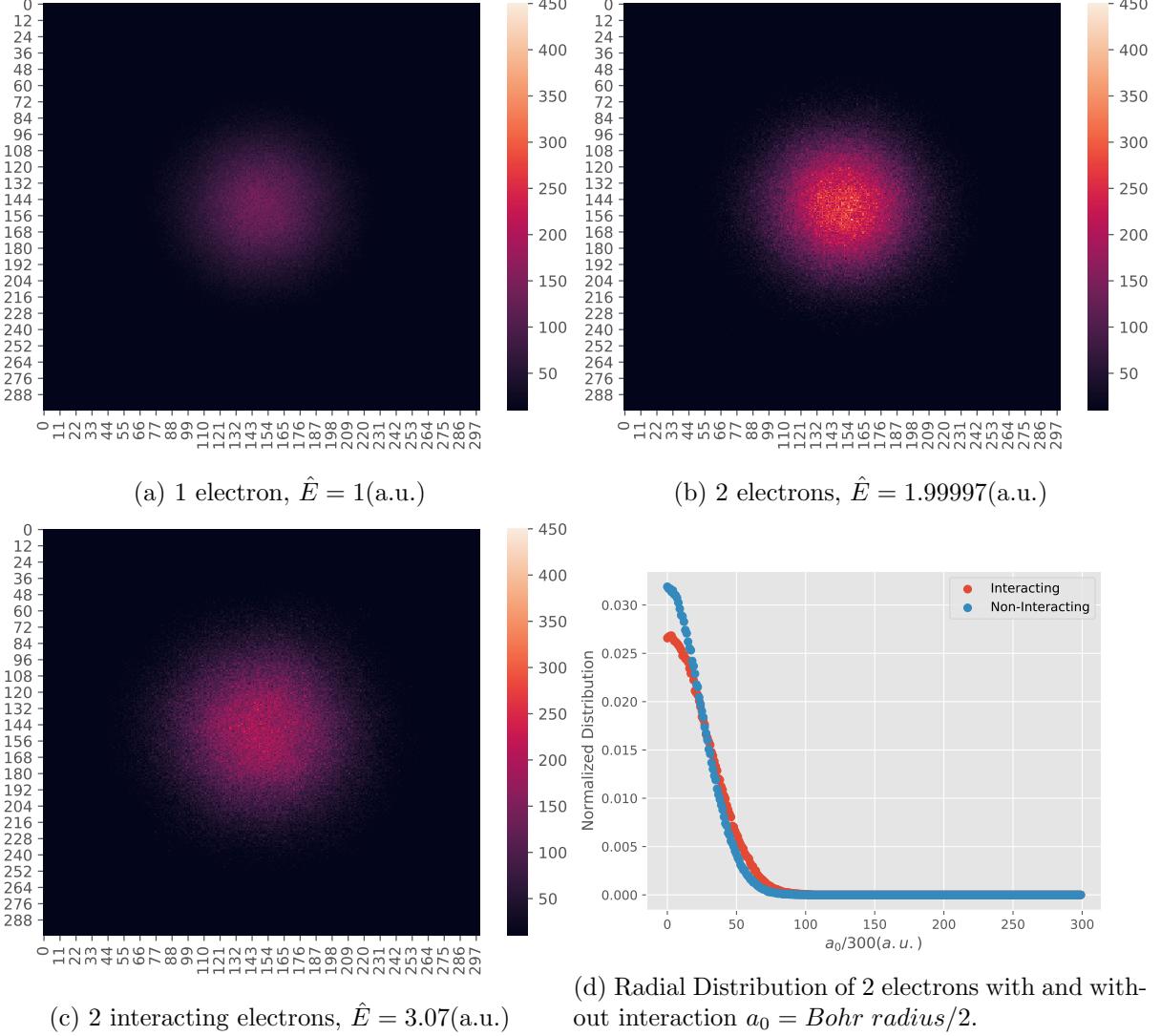


Figure 7: Electrons in two dimensions. Computed using importance sampling with 2^{20} equilibration steps and 2^{20} sampling steps. The entire interval in the x and y positions are six times the natural length scale of the system, centered at $x, y = 0$. Those intervals are split into 300 zones that record each time an electron is located in that zone in the course of the sampling steps. Note that the x- and y- axes in figures 7a, 7b and 7c are given in 1/300th's of the Bohr radius, not a_0 as I've annoyingly managed to call half the Bohr radius in the radial distribution diagrams.

Here we see exactly what we would expect. When the interaction is included, the electrons distribute themselves farther from the center of the HO potential to reduce the interaction potential. We see this both in the spatial distribution plot where the bright central spot is notably missing with the interaction, and in the radial distribution plot where electrons are shifting to between 40 and 80 $\frac{1}{600}th's$ of a Bohr radius from the center.

6 Discussion

6.1 Loss of significance

In the computation of equation (20), the term $(1 - \zeta(u_j))$ will inevitably lead to loss of precision as $\zeta(u_j)$ takes on values $\{f \mid 0 < f < 1\}$ and can therefore come very close to one. A solution

would be to compute $\zeta(-u_j)$ instead.

Loss of significance can happen many places in this program, but the unique shape of the Sigmoid makes it much more likely in this case.

6.2 Divide by zero

The interaction term of the Hamiltonian involves dividing by the absolute value of the distance between particles which may at some point during the simulation become zero. The implementation accounts for this by adding an epsilon of order 10^{-10} to denominator, which since we're taking the absolute value, should prevent any division by zero. This is not ideal, as it artificially lowers the computed energy. This method could potentially be improved by branching if the particle distance would lead to overflow, replacing the output with a value sufficiently below the overflow threshold.

6.3 The nature of an upper bound

Ideally, we would want for our program to come as absolutely close to the analytic value, from above, and never below it. Because of the inevitable shortcomings of floating point arithmetic, we would want to extensively catalogue the systematic sources of error and bias them if possible to over estimate (rather than under estimate) the energy of the system. This would be one of many necessary steps to making this program predictive in problems, not blessed by analytic solutions.

6.4 Bug

A bug in an earlier implementation of this program would sometimes cause the energy of the system to explode. Now, I wouldn't normally include details about a bug, except that this one was quite interesting. Because of an indexing error, the y- coordinate of the second particle would never be updated in a metropolis step. The same indexing error also meant that if the move was rejected, the program would do something like this

```
set  $P_1(y) = P_2(x);$ 
set  $P_2(x) = P_2(y) = \text{Constant};$ 
```

effectively constraining the second particle to move along a small vertical line. This meant that every time two moves were rejected in a row, we would get

Result: $P_1(y) = P_2(x) = P_2(y) = \text{Constant}$

```
set  $P_1(y) = P_2(x);$ 
set  $P_2(x) = P_2(y);$ 
set  $P_1(y) = P_2(x);$ 
set  $P_2(x) = P_2(y);$ 
```

This would (less strictly) constrain the first particle to align with the second, horizontally. The only (actual) degree of freedom for the system then became the x- position of the first particle. This created an obvious failure mode where every time the first particle crossed a diagonal ($P_1(x) = P_1(y)$), the interaction term would explode. The chance then, that the only relevant parameter ($P_1(x)$) would be updated in the next cycle was less than 1 in 3. The energy would then aggregate over several cycles and likely cause a memory leak. This problem would become worse, the closer the initial value of $P_2(y)$ was to zero, as the first particle would intercept the diagonal more frequently near the origin.

One example of this failure mode is seen in figure 8 below. The diagonal is marked in red. The x- and y- axes are flipped.

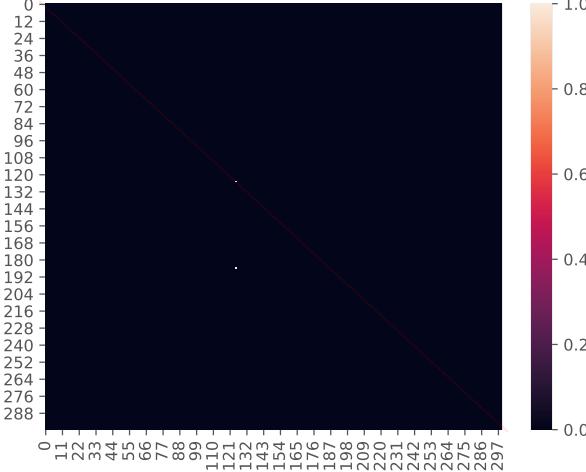


Figure 8: Position sampling for 2 interacting electrons in 2 dimensions using importance sampling. In this case, the energy was calculated to be -1374 a.u. The electrons remained at those locations for every sampling step. The second particle (most likely) is located on a diagonal, and the first (most likely) is horizontally locked to it.

7 Conclusion

The application of RBMs to systems of interacting electrons has been shown to be worthwhile, and largely successful. Using good optimization schemes and reducing statistical errors have been the most fruitful endeavours in getting good estimates. The optimization can only work well as long as the random fluctuations in expectation values between optimization runs don't exceed the incremental improvements made by tweaking the parameters. For this, importance sampling and the ADAM optimizer proved to be best options. The number of hidden nodes in the wave function made close to no difference in its predictions. Using very small ADAM learning rates (0.00001) close to the minimum for the energy was very effective in further minimizing the energy.

The program seems to be largely in agreement with analytical results except for two electrons in one dimension, where it underestimated the energy by over 0.023 a.u. . This is a very big disparity that seriously undermines the predictive power of the program. Either more Monte-Carlo cycles are needed or there are systematic sources of error that would need to be addressed. The program gave a quite accurate upper bound for the interacting system with 3.04450717 a.u. compared to 3.0 a.u. for the analytic solution with a standard deviation of 0.01146434 a.u.

This implementation is not currently in a good state for making predictions. It should be in much greater agreement with theory and provide thorough details about the sources of error in this program. It would therefore benefit greatly from a host of unit tests to pinpoint the main contributors.

To expand the breadth of this program one would need to implement new `waveFunction` and `Hamiltonian` daughter classes to incorporate the Pauli exclusion principle, spin coupling and potentially external magnetic fields. This would in turn require a method of numerical differentiation of the wave function, unless analytic derivatives for such wave functions exist.

8 Appendix A

Using the hamiltonian in (6) we get that the local energy is given by

$$\begin{aligned} E_L &= \frac{1}{\Psi} \mathbf{H} \Psi \\ &= -\frac{1}{2} \frac{1}{\Psi} \sum_i^M \frac{\partial^2}{\partial X_i^2} \Psi + \frac{1}{2} \omega^2 \sum_i^M X_i^2 + \sum_{j < k} \frac{1}{r_{jk}} \end{aligned}$$

with M being the number of visible nodes in the network. The only non-trivial contribution comes from the Laplacian. We can simplify the differential term like this

$$\frac{\partial^2}{\partial X_i^2} \ln \Psi = \frac{\partial}{\partial X_i} \left(\frac{1}{\Psi} \frac{\partial}{\partial X_i} \Psi \right) = \frac{1}{\Psi} \frac{\partial^2}{\partial X_i^2} \Psi - \left(\frac{\partial}{\partial X_i} \ln \Psi \right)^2$$

giving us this expression

$$\begin{aligned} E_L &= \frac{1}{\Psi} \mathbf{H} \Psi \\ &= -\frac{1}{2} \sum_i^M \left(\frac{\partial^2}{\partial X_i^2} \ln \Psi + \left(\frac{\partial}{\partial X_i} \ln \Psi \right)^2 \right) + \frac{1}{2} \omega^2 \sum_i^M X_i^2 + \sum_{j < k} \frac{1}{r_{jk}}. \end{aligned} \quad (19)$$

The natural log of the wave function is computed to be

$$\ln \Psi = -\ln Z - \sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2} + \sum_j^N \ln \left(1 + e^{b_j + \sum_i^M \frac{X_i W_{ij}}{\sigma^2}} \right).$$

The differentials can then be solved to get

$$\frac{\partial}{\partial X_i} \ln \Psi = \frac{1}{\sigma^2} \left(a_i - X_i + \sum_j^N W_{ij} \zeta(u_j) \right)$$

and

$$\begin{aligned} \frac{\partial^2}{\partial X_i^2} \ln \Psi &= -\frac{1}{\sigma^2} + \frac{1}{\sigma^4} \sum_j^N W_{ij}^2 \zeta(u_j) \zeta(-u_j) \\ \frac{\partial^2}{\partial X_i^2} \ln \Psi &= -\frac{1}{\sigma^2} + \frac{1}{\sigma^4} \sum_j^N W_{ij}^2 \zeta(u_j) (1 - \zeta(u_j)) \end{aligned} \quad (20)$$

where $\zeta_j = \frac{1}{e^{-u_j} + 1}$ is the familiar sigmoid function with $u_j = b_j + \frac{1}{\sigma^2} \sum_k^M X_k W_{kj}$. This concludes the calculation.

9 Appendix B

The gradient of the local energy with respect to the network parameters is given by

$$G_i = \frac{\partial \langle E_L \rangle}{\partial \alpha_i} = 2 \left(\left\langle E_L \frac{1}{\Psi} \frac{\partial \Psi}{\partial \alpha_i} \right\rangle - \langle E_L \rangle \left\langle \frac{1}{\Psi} \frac{\partial \Psi}{\partial \alpha_i} \right\rangle \right). \quad (21)$$

From the expression for the wave function in 8 the derivative of the wave function with respect to the weights and biases are calculated to be

$$\frac{\partial}{\partial a_i} \ln \Psi = \frac{X_i - a_i}{\sigma^2},$$

$$\frac{\partial}{\partial b_i} \ln \Psi = \zeta_i$$

and

$$\frac{\partial}{\partial W_{ij}} \ln \Psi = \frac{X_i}{\sigma^2} \zeta_j.$$

this concludes the calculation.

References

- [1] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.
- [2] Marius Jonsson. Standard error estimation by an automated blocking method. *Phys. Rev. E*, 98:043304, Oct 2018.
- [3] A. Kleven. Ground state energy of hard sphere bose gas in elliptical ho potential by vmc methods. Feb 2020.
- [4] OpenMP Architecture Review Board. OpenMP application program interface version 3.0, May 2008.
- [5] M. Taut. Two electrons in an external oscillator potential: Particular analytic solutions of a coulomb correlation problem. *Phys. Rev. A*, 48:3561–3566, Nov 1993.