Machine learning for quantum many-body problems

Pavlo Panasiuk, Jonas Gabriel Jonsson

(Dated: June 9, 2023)

We implemented a Markov Chain Monte Carlo (MCMC) with importance sampling based on Metropolis-Hastings algorithm. The goal is to evaluate ground state of a system of nonrelativistic spinless electrons trapped within harmonic oscillator using network quantum state ansatz (NQS, [1]). We developed a 1-hidden-layer dense neural network with gradient descent as optimizer. Energy evaluations are consistent within 4-35% when compared against [2] and [3]. We further discuss possible extensions to the ansatz.

I. INTRODUCTION

With increasing interest to artificial neural networks [1, 4, 5], quantum many-body problems undergo reformulation, in which the main quantum mechanics quantity, namely system's wave function, is replaced with a neural-network-like functional dependence. For simplicity, we introduce a Gaussian-Binary Restricted Boltzmann Machine (RBM, [1]). This means we use continuous particle coordinates - describing a given quantum system - as inputs to the visible layer, while the hidden layer consists of binary nodes. The corresponding wave function is usually called neural network quantum state(NQS).

In the current document, we aim to study and benchmark 3D bosonic systems with hard-core repulsion (against [3]), as well as 2D boson-like (disregarding Grassman structure to avoid complexity) electrons featuring Coulomb repulsion (against [2]).

Monte Carlo simulation approach is taken as a base. We extend the codebase provided in [3] to support bosonic system with Coulomb repulsion. In particular we compare against analytic result of 2 2D electrons energy, equal to 3 a.u. [2].

Reproducible code can be found in our GitHub repository [6]. Credits for source code also goes to [7] from which we forked the initial code base, and [8] from which we reused scripts for blocking method data analysis.

We comment on the features of RBM approach to the quantum system in Sec. II. The outcomes of the benchmark, as well as other evaluations, are to be found in Sec. III. The conclusion is given in Sec. IV. Technical details are provided in Sec. V.

II. THEORY AND METHODOLOGY

For our numerical simulation of quantum systems we invoke Markov chain Monte Carlo (MCMC) with importance sampling (Metropolis-Hastings algorithm). While most relevant details, including statistical analysis and parallelisation, are outlined in [3], we concentrate on neural network related theory and on new features.

A. Restricted Boltzmann machines

Central to the work here is the concept of the Restricted Boltzmann machine, for which we will outline some details.

A Restricted Boltzmann machine (RBM) is a type of generative neural network, that can learn a probability distribution from a series of inputs. A Boltzmann machine consists of one visible layer of nodes serving as input layer, and one hidden layer. We denote M the number of nodes in the visible, and N the number of nodes in the hidden layer [9].

The parameters of the RBM consists of a bias for each node, and a weight for each pair of nodes, one from the visible and one from the hidden layer. The visible biases associated with the visible nodes are denoted by a vector a, and for the hidden biases respectively a vector b. The weights, denoted by a matrix W, describe the connections between the nodes, i.e. how the values of the hidden nodes depend on the visible nodes. For the sake of illustration a figure is borrowed from [9], see Fig. 1.

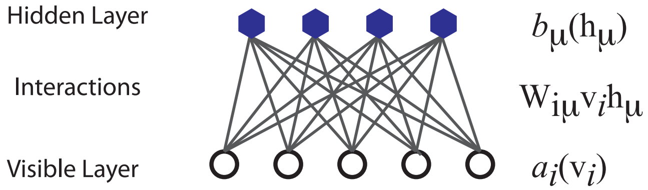

FIG. 1. Illustration of the RBM and the meaning of the parameters. Figure taken from [9].

The term "Restricted" means that there are no connections between pairs of visible or pairs of hidden nodes. How these parameters look and work will become clearer in later sections when we described the NQS, which in turn is based on the RBM concept [9].

The number of visible nodes is the same as the number of degrees of freedom in the system, in our case the coordinates of the particles simulated. The number of hidden nodes is arbitrary and something we can experiment with and find the number which gives the best results compared to for example analytical results [1], [9].

The RBM we use is of type Gaussian-Binary, meaning the input layer node take on continuous variables while the hidden nodes take on binary values. This is done in order to take the particle coordinates as input parameters. In other applications the Binary-Binary variant where both layers take on binary values only, are the most common RBM [9]

As usual when it comes to neural networks there is a cost function associated, which in our case is the calculated energy of the system. By using optimization algorithms, in our case gradient descent, the best choice of parameters for the RBM can be found. The details of both the energy calculation and optimization algorithm will be further detailed in later sections. As this is a stochastic method it's also important to consider the statistical error, which we will also come back to [1].

Because the ground state has the lowest energy, this is a minimization problem well suited for this setup. Since we can produce sets of representative inputs using Monte Carlo methods, this gives us a system where we can use reinforced learning to find an approximation to the ground state [1].

B. Setup

Per this research we mostly aim to study the system of spinless electrons, with renormalized Hamiltonian following

$$\hat{H} = \sum_{i}^{N} \left[-\frac{1}{2}\triangle_i + \frac{\omega^2}{2} r_i^2 \right] + \sum_{i<j}^{N\times N} \frac{1}{r_{ij}}, \tag{1}$$

where $\triangle_i$ is i-th Laplasian operator and $r_{ij} = |\vec{r}_i - \vec{r}_j|$. This is suitable for the description of low density bosonic systems and magnetic field-free fermionic systems, and, to some applicability extent, for very low density fermionic systems. We specifically aim for 2 electrons description, both to be able to compare the calculations in 2D, as well as to limit possible error related to unnatural symmetricity of the considered wave function.

For the wave function ansatz, required by variational MC method, we use the NQS ansatz as anticipated. To make the ansatz physically grounded, we explicitly connect the statistical mechanics (canonical ensemble in particular) probability with wave function

$$\Psi(X,H) = \frac{1}{Z} e^{-\frac{E(X,H)}{T}}, \tag{2}$$

where X, H denote a microstate with X being coordinates and H being hidden parameters. We ignore here the fact that in fact $P \sim \Psi^2 \neq \Psi$, which does not matter for the introduction of an fit ansatz. It is also natural to ignore the constant temperature factor, since we may introduce it as a fit parameter within $E(X,H)$ itself.

Z is usually referred to as statistical sum, and its definition follows from probability normalization

$$Z = \int dX dH e^{-\frac{E(X,H)}{T}}. \tag{3}$$

In what follows we put $T = 1$.

The Gaussian-Binary ansatz reads

$$E(X,H) = \sum_{i}^{M} \frac{(X_i - a_i)^2}{2\sigma^2} - \sum_{j}^{N} b_j H_j - \sum_{i,j}^{M,N} \frac{X_i W_{ij} H_j}{\sigma^2}, \tag{4}$$

where σ is the chosen Gaussian spread.

Finally, the coordinate wave function in this setup is given by marginal distribution

$$\Psi(X) = \frac{1}{Z}\sum_{H} \Psi(x,H) = \\ \frac{1}{Z} e^{-\sum\limits_{i}^{M} \frac{(X_i - a_i)^2}{2\sigma^2}} \prod_{j}^{N} \left(1 + e^{b_j + \sum\limits_{i}^{M} \frac{X_i W_{ij}}{\sigma^2}} \right), \tag{5}$$

where summation is performed over binary Hs. By definition, this wave function is properly normalized.

The local energy for the setup is calculated in Sec. V.

To match performance of the NQS against the reqular harmonic oscillator ansatz (as done in [3]), we also test it on hard-core repulsive system of bosons.

C. Optimization

For optimization purposes, we reuse plain gradient descent from [3] with slight improvements. This is a very basic method, but it serves well for small number of fit parameters. The largest system we plan to run for is 10 particles in 3D, whereas with a choice of number of hidden nodes of 12 this results in 402 parameters, which starts to become demanding, but should still be manageable. However one can see that for larger systems of say 100 particles this may soon become intractable. Implementing stochastic gradient descent would be a nice extensions to this project.

The improvement, first of all, relate to the choice of learning rate, which we now fix instead of automatic scale deduction. Looking at other work done on this [9], we desided to experiment with learning rate of range $\{0.01, 0.05\}$.

A success measure we employ to see the efficiency of the network is comprised of natural quantities in gradient descent method

$$S = \sum_{i} |l_i \cdot \nabla p_i|, \tag{6}$$

where l_i and ∇p_i are learning rate and gradient associated to the i-th parameter respectively. Convergence of the algorithm is relied on S being smaller than a fixed predefined number.

Adaptive learning rate (e.g. ADAM optimizer) would serve a notable extension for running with many parameters. We however limit ourselves to the fixed learning rate.

D. Statistical error estimation and Blocking method

The error analysis is central, because without any confidence interval the result does not tell much. With a stochastic approach like this, we need to be aware of the possibility of both systematic and statistical errors.

We will not directly try to calculate systematic errors are those are associated with how well the model represents the physical system. We will however draw some general conclusion about the model later when comparing with analytical results.

For the statistical errors, we reuse the method from our previous report [3]. It builds on work from [10] and [8], and we also a script from the latter to calculate an estimated true standard deviation.

In summary the background is that since we use MCMC methods to generate samples, and we calculate standard deviation from this sampled set, due to correlation between samples this error analysis becomes too optimistic. By using resampling methods, in this case the Blocking method, we use statistical methods on subsets of the data and as an end results obtains a standard deviation closer to the true standard deviation.

III. RESULTS AND DISCUSSION

In this chapter we present the simulated energies and related observables for Coulomb-like repulsive systems and for hard-core bosons, both systems trapped in harmonic oscillator potential. Whenever applicable, we benchmark the evaluations, as anticipated. We have started by doing some initial investation and tweaking for 1 particle 1 dimension case just to conclude a decent learning rate and number of hidden nodes. It turn out 0.05 was the best learning rate given the choice of max 100 iterations of optimization. Also in this case 6 hidden nodes was optimal.

A. Bosonic system with hard-core repulsion

In this subchapter we check the performance of RBM against a VMC featured in [3]. The check consists of calculating energies for 1 particle in 1D and for 5/10 particles in 3D with $a_{hard-core} = \frac{0.0043}{\sqrt{\omega}}$ (c.f. [3]). Following this report, for $\omega = 1, \beta \approx 2.82$, one expects

$$\begin{aligned} &E_1^{1D} = 0.5 \\ &E_N^{3D} \approx 2.41N, N \ll 1. \end{aligned}$$

Following Table I, we acknowledge a notable divergence from the expected result

$$\delta \approx \frac{|3.3P - 2.41P|}{2.41P} \approx 35\%. \tag{7}$$

SYS	P = 1, 1D	P = 5, 1D	P = 5, 3D	P = 10, 3D
$\langle E_L \rangle$	0.500095	2.5035	16.5048	33.2022
time	~ 4 s	~ 4 s	~ 27 s	~ 41 s
σ_{MC}	0.00003143	0.000172071	0.0151486	0.0224033

TABLE I. Table showing results for MCMC runs with Metropolis algorithm for a cylindrical harmonic trap, with various number P of particles, with 10^6 Monte Carlo cycles, and particle hard-core interaction active. The table shows energy $\langle E_L \rangle$, runtime and standard deviation from MCMC calculation σ_{MC}. For the simulation with NQS here, we use an amount of 10 hidden nodes (other proved to be less efficient in finding minima here, or more prone to diverging).

This error is likely related to the fact the RBM is not well-suited for this exact system. We would expect the MCMC to perform more efficiently in case we explicitly define a factor within ansatz, specifically related to a free trap solution (i.e. harmonic oscillator ground state as a multiplier).

Possible way to improve could be enlarging the network for it to able to catch more features of the energy surface.

These questions however are left for another research.

Even though the implementation in its current stage does not prove reliable for 3D, we indicate a significant speedup in the energy surface optimization, reaching as high as 6x times speedup. The 1D simulation provides a reliable estimate, comparable with

$$E_{P,D} \approx \frac{P \cdot D}{2} \tag{8}$$

B. Repulsive interaction

Here we naturally move to a system prescribed in 12, with interaction potential enabled. To check how expected does it function, we compare against analytic result for 2 electrons with this same Hamiltonian, calculated in [2], where they claim

$$E_{gr} = 3 \text{ a.u.}, \tag{9}$$

where the units are scaled with $\hbar\omega, \omega = 1$.

For this we compare Metropolis (Table II) with Metropolis-Hastings (Table III) as well as different number of hidden nodes, to evaluate what performs best based on the real answer.

The lowest estimate we could evaluate for the problem is, following Tables II, III,

$$E_{gr}^{est} \approx 3.1133, \tag{10}$$

which only overestimates the analytic result by

$$\delta = \frac{3.1133 - 3}{3} \approx 4\%. \tag{11}$$

This proves that NQS approach is more robust in generic case on the contrary to classical VMC formulation, which is indeed expected from a neural network.

SYS	10 nodes	50 nodes	75 nodes
$\langle E_L \rangle$	3.1936	3.1248	3.13546
σ_{MC}	0.005214	0.01415	0.00475037
$\sigma_{Blocking}$	0.03827	0.004922	0.0150221

TABLE II. Table showing results for MCMC runs with Metropolis algorithm for a cylindrical harmonic trap, with 2 particles, with 10^6 Monte Carlo cycles, and learning rate of 0.05 for all parameters. The table shows energy $\langle E_L \rangle$, runtime and standard deviation from MCMC calculation σ_{MC} and $\sigma_{Blocking}$ via blocking algorithm. For the simulation with NQS here, we use different number of nodes in each column as indicated.

SYS	10 nodes	50 nodes	75 nodes
$\langle E_L \rangle$	3.1916	3.1133	3.11523
σ_{MC}	0.005118	0.004707	0.00554925
$\sigma_{Blocking}$	0.02038	0.01438	0.0155593

TABLE III. Table showing results for MCMC runs with Metropolis-Hastings (Importance sampling) algorithm for a cylindrical harmonic trap, with 2 particles, with 10^6 Monte Carlo cycles, and learning rate of 0.05 for all parameters. The table shows energy $\langle E_L \rangle$, runtime and standard deviation from MCMC calculation σ_{MC} and $\sigma_{Blocking}$ via blocking algorithm. For the simulation with NQS here, we use different number of nodes in each column as indicated.

Following Table II, we observe that the energy grows with the number of nodes between 50 and 75, which is a clear sign of undertraining in place. We also tried tweaking the learning rate value, however 0.05 shows the best tradeoff between diverging and seriously undertraining.

The sampling algorithms do not show significant difference between them, which we expect to be a sign of reaching the minima rapidly with both algorithms.

IV. CONCLUSION

In this document we studied an extension to classical variational MC in quantum mechanics, based on replacing the wave function with NQS.

For feebly interaction particles in 1D we reach a notable agreement with analytical result and previous results from our previous project [3].

With particle interaction enabled in 3D the results disagree significantly with expected values (at about 35% rel. error), indicating at Boltzmann machines are not a sufficient wave function representation here. More reliable results might be achieved, if we replace Metropolis-Hasting algorithm with Gibbs sampling. As stated in [5] the most common approach when it comes to machine learning is actually Gibbs sampling. We choose however not to take this route for simplicity and for a more direct comparison with our previous work [3], but it would nevertheless be an interesting option for continued work.

We did not identify a definitive impact of importance sampling introduced into our results, concluding they may be used interchangeably for the purposes presented in the report. This conclusion may be affected in more involved NQSs.

The simulation of two-electron repulsive system agrees within 4% with analytic simulation and therefore proves NQS might be useful for generic simulation. This error, alongside with imprecision of the NQS ansatz, may be enabled by the fact we neglected replacement antisymmetry of the actual wave function. Due to this complexity, we did not probe the algorithm on more electrons, the support for Grassman variables must be added first.

The impact of the node count in hidden layer is studied on two electron system. The simulation shows clear signs of notable undertraining for the hidden layer with ~ 75 nodes. The minimized value $E \approx 3.1133$ was derived with Metropolis-Hastings sampling and 50 hidden nodes.

Beyond mentioned, there are other ways to improve the performance, e.g. by changing the spread on initial guess of parameter values. Some experimentation with these values shows that changing these values too much may cause the simulation to diverge. The computation is thus apparently sensitive to these values, though we did not go into depth of finding more optimal ones.

But in all likelihood to reach a sub-percent precision, we would need to switch to a real neural network representation, like described in [4]. This would be another interesting option, however a much larger project.

REFERENCES

[1] G. Carleo and M. Troyer, “”solving the quantum many-body problem with artificial neural networks”,” *”PHYSICAL REVIEW A 48, 3561”*, November 1993.

[2] M. Taut, “”two electrons in an external oscillator potential: Particular analytic solutions of a coulomb correlation problem”,” *PHYSICAL REVIEW A 48, 3561*, vol. 48, November 1993.

[3] G. Jonsson and P. Panasiuk, “”variational monte carlo simulation on bose-einstein condensation in magnetic trap”,” Apr 2023.

[4] H. Saito, “Method to solve quantum few-body problems with artificial neural networks,” *J. Phys. Soc. Jpn.*, vol. 87, Jun 2018. [Online]. Available: https://journals.jps.jp/doi/10.7566/JPSJ.87.074002

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[6] P. Panasiuk and J. G. Jonsson. (2023) Github repository. [Online]. Available: https://github.com/jgjonsson/jgj-variational-monte-carlo

[7] (2023) Github repository. [Online]. Available: https://github.com/mortele/variational-monte-carlo-fys4411

[8] M. Jonsson, “Standard error estimation by an automated blocking method,” *Phys. Rev. E*, vol. 98, p. 043304, Oct 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.98.043304

[9] M. Hjorth-Jensen. Neural networks and project 2. [Online]. Available: https://github.com/CompPhysics/ComputationalPhysics2/blob/gh-pages/doc/pub/week13/ipynb/week13.ipynb

[10] H. G. P. H. Flyvbjerg, “Error estimates on averages of correlated data,” *The Journal of Chemical Physics, vol 91, issue 1*, vol. 91, pp. 1–17, 1989.

V. LOCAL ENERGY DERIVATION FOR NQS

Further we consider a (normalized) hamiltonian system of N feebly interacting nonrelativistic electrons within oscillator potential. The system is given by

$$\hat{H} = \sum_{i}^{N} \left[-\frac{1}{2}\triangle_i + \frac{\omega^2}{2} r_i^2 \right] + \sum_{i<j}^{N\times N} V_{int}(r_{ij}), \tag{12}$$

whereas

$$V_{int}(r_{ij}) = \frac{1}{r_{ij}}$$

represents a coulomb-like repulsion.

To calculate local energy we introduce neural network quantum state (a.k.a. NQS) ansatz [1]

$$\Psi(\mathbf{X}) = \frac{1}{Z} \prod_{i}^{N\cdot D} e^{-\frac{(X_i-a_i)^2}{2\sigma^2}} \times \prod_{j}^{N} \left(1 + e^{b_j + \sum\limits_{i}^{N\cdot D} \frac{X_i W_{ij}}{\sigma^2}} \right), \tag{13}$$

where $\vec{a}, \vec{b}, \hat{W}$ represent NQS parameters, and Z is partition function, independent of X. We denote $M = N \cdot D$, where D is number of spatial dimentions.

Local energy for the system reads

$$E_L = \frac{\hat{H}\Psi}{\Psi} = -\frac{1}{2} \sum_i \frac{\triangle_i \Psi}{\Psi} + \sum_{i}^{N} \frac{\omega^2}{2} r_i^2 + \sum_{i<j}^{N\times N} V_{int}(r_{ij}), \tag{14}$$

and local laplasian is given by

$$\frac{\triangle_i \Psi_T}{\Psi_T} = \frac{\vec{\nabla}_i \cdot \vec{\nabla}_i \Psi_T}{\Psi_T} = \frac{\vec{\nabla}_i \cdot \frac{\vec{\nabla}_i \Psi_T}{\Psi_T} \Psi_T}{\Psi_T} = \left(\frac{\vec{\nabla}_i \Psi_T}{\Psi_T} \right)^2 + \vec{\nabla}_i \frac{\vec{\nabla}_i \Psi_T}{\Psi_T} = (\vec{\nabla}_i \ln \Psi_T)^2 + \vec{\nabla}_i \vec{\nabla}_i \ln \Psi_T. \tag{15}$$

With

$$\ln \Psi = \ln \frac{1}{Z} - \sum_{i}^{M} \frac{(X_i - a_i)^2}{2\sigma^2} + \sum_{j}^{N} \ln \left(1 + e^{b_j + \sum\limits_{i}^{M} \frac{X_i W_{ij}}{\sigma^2}} \right), \tag{16}$$

we immediately obtain

$$\frac{\partial \ln \Psi}{\partial X_i} = -\frac{X_i - a_i}{\sigma^2} + \sum_{j}^{N} \frac{W_{ij}}{\sigma^2} \frac{1}{1 + e^{-b_j - \sum\limits_{i}^{M} \frac{X_i W_{ij}}{\sigma^2}}} = -\frac{X_i - a_i}{\sigma^2} + \sum_{j}^{N} \frac{W_{ij}}{\sigma^2} S\left(b_j + \sum_{i}^{M} \frac{X_i W_{ij}}{\sigma^2} \right), \tag{17}$$

where we introduced a notation for sigmoid logistic function

$$S(\xi) = \frac{1}{1 + e^{-\xi}}. \tag{18}$$

Finally, we need

$$\frac{\partial^2 \ln \Psi}{\partial X_i^2} = [S'(x) = S(x)S(-x)] = -\frac{1}{\sigma^2} + \sum_{j}^{N} \left(\frac{W_{ij}}{\sigma^2} \right)^2 S\left(b_j + \sum_{i}^{M} \frac{X_i W_{ij}}{\sigma^2} \right) S\left(-b_j - \sum_{i}^{M} \frac{X_i W_{ij}}{\sigma^2} \right). \tag{19}$$

With notation

$$B_j = b_j + \sum_{i}^{M} \frac{X_i W_{ij}}{\sigma^2}, \tag{20}$$

we finally obtain

$$E_L = -\frac{1}{2}\sum_{i}^{M}\left[-\frac{1}{\sigma^2} + \sum_{j}^{N}\left(\frac{W_{ij}}{\sigma^2}\right)^2 S\left(B_j\right) S\left(-B_j\right) + \left(-\frac{X_i - a_i}{\sigma^2} + \sum_{j}^{N}\frac{W_{ij}}{\sigma^2} S\left(B_j\right)\right)^2\right] + \sum_{i}^{N}\frac{\omega^2}{2} r_i^2 + \sum_{i<j}^{N\times N} V_{int}(r_{ij}) \tag{21}$$