Quantum Mechanical Many-body Problems with Machine Learning Algorithms

Vilde M. Flugsrud,[1] Jane Kim,[2] Even M. Nordhagen,[1] Morten Hjorth-Jensen,[2,3] and Francesco Pederiva[4,5]
[1)] *Department of Physics, University of Oslo, N-0316 Oslo, Norway*
[2)] *National Superconducting Cyclotron Laboratory and Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA*
[3)] *Department of Physics and Center for Computing in Science Education, University of Oslo, N-0316 Oslo, Norway*
[4)] *Dipartimento di Fisica, University of Trento, via Sommarive 14, I-38050 Povo, Trento, Italy*
[5)] *INFN-TIFPA, Trento Institute for Fundamental Physics and Applications, Trento, Italy*

In this work we discuss differences and similarities between traditional many-body methods like variational and diffusion Monte Carlo approaches and novel unsupervised machine learning methods. The latter are based on so-called Boltzmann machines. In particular we point to similarities between Monte Carlo approaches with shadow wave functions and restricted Boltzmann machines. We discuss their pros and cons and apply the various algorithms to studies of quantum dots. These are systems of interacting electrons confined to move in low-dimensional regions. For two interacting electrons only we have analytic solutions for the eigenpairs, providing us the with ways to benchmark the quality of our numerical simulations.

We show also that the results obtained with restricted Boltzmann machines are of the same quality as those obtained with traditional variational Monte Carlo approaches.

PACS numbers: 32.80.Rm, 31.15.A-, 42.65.Ky

I. INTRODUCTION AND MOTIVATION

Traditional many-body methods like full configuration interaction theory (FCI)[?] , coupled-cluster (CC) theory[?] , many-body perturbation theory (MBPT)[?] , in-medium similarity renormalization group (IMSRG)[?] , various Monte Carlo methods[?] and many other many-body approaches, have been rather successful in describing properties of interacting many-body systems. These methods have been widely used in condensed matter physics, nuclear physics, quantum chemistry and materials science, just to mention a few of the areas of applicability.

However, essentially all of these methods face what is normally called the curse of dimensionality. For wave function based method like FCI or CC theories where the original continuous problem is discretized in terms selected basis functions and/or many-body excitations, the dimensionality of the systems under study grows almost exponentially when larger basis sets and/or number of particles are included. As an example, nuclear physics systems close to the limits of stability[?] pose in particular a tough problem to many-body practitioners in terms of a dramaticn increase of the number of relevant degrees of freedom. To describe say weakly bound nuclei within the framework of a wave function based approach, requires often a single-particle basis which includes bound, weakly bound and unbound states, increasing thereby considerably the number of many-body basis states. This renders often a standard FCI calculation infeasible. Within the above mentioned many-body methods there are however several interesting theoretical approachs which attempt at circumventing the dimensionality curse. Smarter basis sets is one of these approaches, as well as the resummation of specific correlations and the recently proposed stochastic sampling of many-body states in both FCI and CC calculations[? ?] .

Recently, several authors have pointed to possibilities within the broad fields of Machine Learning and Quantum Computing as approaches that hold great promise in tackling the ever increasing dimensionalities. There are several groups worldwide which now focus on Machine Learning and/or Quantum Computing applied to many-body problems[?] . Machine learning (ML) is an extremely rich field, in spite of its young age. The increases we have seen during the last three decades in computational capabilities have been followed by developments of methods and techniques for analyzing and handling large date sets, relying heavily on statistics, computer science and mathematics. The field is rather new and developing rapidly.

Within this setting, the aim of this work is to explore similarities and differences between widely used many-body methods like the Variational Monte Carlo or the diffusion Monte Carlo approaches and unsupervised Machine Learning methods using so-called Boltzmann machines. We will also point to similarities between Monte Carlo approaches with shadow wave functions and Machine Learning algorithms that employ Boltzmann machines. Since the field of Machine Learning is evolving rather rapidly and physicists may not be fully familiar with the plethora of Machine Learning algorithms, we will try in this work to keep a more pedestrian approach. In particular, we will try to outline some of the similarities between Monte Carlo methods for solving quantum mechanical many-body problems and Machine Learning

algorithms.

We start thus with a brief review of central aspects of Monte Carlo methods in the next section. Our model system is that of electrons confined to move in two or three dimensions, so-called quantum dots. We will in particular focus on idealized quantum dots with electrons moving in a harmonic oscillator potential. There are several reasons for focusing on quantum dots. For two interacting electrons there are analytical eigenpairs, allowing us thereby to benchmark various many-body methods[? ?] . For more than two electrons, there are extensive reference calculations based on a variety of many-body methods. These calculations, in addition to new ones presented here, will allow us to assess the relevance of our results obtained with Machine Learning algorithms.

Thereafter, in Section **??**, we present and discuss the implementation of Machine Learning algorithms. In that section we present also a discussion of possible similarities between Boltzmann machines and so-called Shadow wave functions.

II. WHAT IS MACHINE LEARNING

Almost every problem in ML and data science starts with the same ingredients:

- The dataset $\mathbf{x}$ (could be some observable quantity of the system we are studying)
- A model which is a function of a set of parameters α that relates to the dataset, say a likelihood function $p(\mathbf{x}|\alpha)$ or just a simple model $f(\alpha)$
- A so-called **cost** function $\mathcal{C}(\mathbf{x}, f(\alpha))$ which allows us to decide how well our model represents the dataset.

We seek to minimize the function $\mathcal{C}(\mathbf{x}, f(\alpha))$ by finding the parameter values which minimize $\mathcal{C}$. This leads to various minimization algorithms.

Machine learning is the science of giving computers the ability to learn without being explicitly programmed. The idea is that there exist generic algorithms which can be used to find patterns in a broad class of data sets without having to write code specifically for each problem. The algorithm will build its own logic based on the data.

Machine learning is a subfield of computer science, and is closely related to computational statistics. It evolved from the study of pattern recognition in artificial intelligence (AI) research, and has made contributions to AI tasks like computer vision, natural language processing and speech recognition. It has also, especially in later years, found applications in a wide variety of other areas, including bioinformatics, economy, physics, finance and marketing.

You will notice however that many of the basic ideas discussed do come from Physics!

The approaches to machine learning are many, but are often split into two main categories. In *supervised learning* we know the answer to a problem, and let the computer deduce the logic behind it. On the other hand, *unsupervised learning* is a method for finding patterns and relationship in data sets without any prior knowledge of the system. Some authours also operate with a third category, namely *reinforcement learning*. This is a paradigm of learning inspired by behavioural psychology, where learning is achieved by trial-and-error, solely from rewards and punishment.

Another way to categorize machine learning tasks is to consider the desired output of a system. Some of the most common tasks are:

- Classification: Outputs are divided into two or more classes. The goal is to produce a model that assigns inputs into one of these classes. An example is to identify digits based on pictures of handwritten ones. Classification is typically supervised learning.
- Regression: Finding a functional relationship between an input data set and a reference data set. The goal is to construct a function that maps input data to continuous output values.
- Clustering: Data are divided into groups with certain common traits, without knowing the different groups beforehand. It is thus a form of unsupervised learning.

An excellent reference, Mehta *et al.*, arXiv:1803.08823 and Physics Reports in press (2018)

Here we will use so-called **reduced Boltzmann Machines** to simulate quantum many-body problems. For Monte Carlo aficionados, there is a very close similarity with what are called **shadow wave functions**, see the work of Pederiva and Kalos and collaborators, Phys Rev. E 90, 053304 (2014).

III. THE SYSTEM: TWO OR MORE ELECTRONS IN A HARMONIC OSCILLATOR TRAP IN TWO DIMENSIONS

The Hamiltonian of the quantum dot is given by

$$\hat{H} = \hat{H}_0 + \hat{V},$$

where $\hat{H}_0$ is the many-body HO Hamiltonian, and $\hat{V}$ is the inter-electron Coulomb interactions. In dimensionless units,

$$\hat{V} = \sum_{i<j}^{N} \frac{1}{r_{ij}},$$

with $r_{ij} = \sqrt{\mathbf{r}_i^2 - \mathbf{r}_j^2}$.

This leads to the separable Hamiltonian, with the relative motion part given by ($r_{ij} = r$)

$$\hat{H}_r = -\nabla_r^2 + \frac{1}{4}\omega^2 r^2 + \frac{1}{r},$$

plus a standard Harmonic Oscillator problem for the center-of-mass motion. This system has analytical solutions in two and three dimensions (M. Taut 1993 and 1994).

A. Quantum Monte Carlo Motivation

Given a hamiltonian H and a trial wave function Ψ_T, the variational principle states that the expectation value of $\langle H \rangle$, defined through

$$\langle E \rangle = \frac{\int d\Psi_T^*(R) H(R) \Psi_T(R)}{\int dR \Psi_T^*(R)\Psi_T(R)},$$

is an upper bound to the ground state energy E_0 of the hamiltonian H, that is

$$E_0 \leq \langle H \rangle.$$

In general, the integrals involved in the calculation of various expectation values are multi-dimensional ones. Traditional integration methods such as the Gauss-Legendre will not be adequate for say the computation of the energy of a many-body system.

$$E_L(R,\alpha) = \frac{1}{\psi_T(R,\alpha)} H\psi_T(R,\alpha),$$

called the local energy, which, together with our trial PDF yields

$$E[\alpha] = \int P(R) E_L(R,\alpha) dR \approx \frac{1}{N}\sum_{i=1}^{N} E_L(R_i,\alpha)$$

with N being the number of Monte Carlo samples.

We want to perform a Variational Monte Carlo calculation of the ground state of two electrons in a quantum dot well with different oscillator energies, assuming total spin $S = 0$. Our trial wave function has the following form

$$\psi_T(r_1, r_2) = C \exp\left(-\alpha_1 \omega (r_1^2 + r_2^2)/2\right) \exp\left(\frac{r_{12}}{(1+\alpha_2 r_{12})}\right), \tag{1}$$

where the αs represent our variational parameters, two in this case.

To find an ansatz for the correlated part of the wave function, it is useful to rewrite the two-particle local energy in terms of the relative and center-of-mass motion. Let us denote the distance between the two electrons as r_{12}. We omit the center-of-mass motion since we are only interested in the case when $r_{12} \to 0$. The contribution from the center-of-mass (CoM) variable $R_{\rm CoM}$ gives only a finite contribution. We focus only on the terms that are relevant for r_{12} and for three dimensions. The relevant local energy becomes then

$$\lim_{r_{12}\to 0} E_L(R) = \frac{1}{\mathcal{R}_T(r_{12})}\left(2\frac{d^2}{dr_{ij}^2} + \frac{4}{r_{ij}}\frac{d}{dr_{ij}} + \frac{2}{r_{ij}} - \frac{l(l+1)}{r_{ij}^2} + 2E\right)\mathcal{R}$$

Set $l = 0$ and we have the so-called **cusp** condition

$$\frac{d\mathcal{R}_T(r_{12})}{dr_{12}} = -\frac{1}{2(l+1)}\mathcal{R}_T(r_{12}) \qquad r_{12} \to 0$$

The above results in

$$\mathcal{R}_T \propto \exp(r_{ij}/2),$$

for anti-parallel spins and

$$\mathcal{R}_T \propto \exp(r_{ij}/4),$$

for anti-parallel spins.

This is the so-called cusp condition for the relative motion, resulting in a minimal requirement for the correlation part of the wave fuction. For general systems containing more than say two electrons, we have this condition for each electron pair ij.

The elements of the gradient of the local energy are then (using the chain rule and the hermiticity of the Hamiltonian)

$$\bar{E}_i = 2\left(\langle \frac{\bar{\Psi}_i}{\Psi} E_L\rangle - \langle \frac{\bar{\Psi}_i}{\Psi}\rangle\langle E_L\rangle\right).$$

From a computational point of view it means that you need to compute the expectation values of

$$\langle \frac{\bar{\Psi}_i}{\Psi} E_L\rangle,$$

and

$$\langle \frac{\bar{\Psi}_i}{\Psi}\rangle\langle E_L\rangle$$

These integrals are evaluted using MC intergration (with all its possible error sources). We can then use methods like stochastic gradient or other minimization methods to find the optimal variational parameters (I don't discuss this topic here, but these methods are very important in ML).

We have a model, our likelihood function.

How should we define the cost function?

IV. WHY BOLTZMANN MACHINES?

What is known as restricted Boltzmann Machines (RMB) have received a lot of attention lately. One of the major reasons is that they can be stacked layer-wise

to build deep neural networks that capture complicated statistics.

The original RBMs had just one visible layer and a hidden layer, but recently so-called Gaussian-binary RBMs have gained quite some popularity in imaging since they are capable of modeling continuous data that are common to natural images.

Furthermore, they have been used to solve complicated quantum mechanical many-particle problems or classical statistical physics problems like the Ising and Potts classes of models.

Why use a generative model rather than the more well known discriminative deep neural networks (DNN)?

- Discriminitave methods have several limitations: They are mainly supervised learning methods, thus requiring labeled data. And there are tasks they cannot accomplish, like drawing new examples from an unknown probability distribution.
- A generative model can learn to represent and sample from a probability distribution. The core idea is to learn a parametric model of the probability distribution from which the training data was drawn. As an example
    1. A model for images could learn to draw new examples of cats and dogs, given a training dataset of images of cats and dogs.
    2. Generate a sample of an ordered or disordered Ising model phase, having been given samples of such phases.
    3. Model the trial function for Monte Carlo calculations

1. Both use gradient-descent based learning procedures for minimizing cost functions
2. Energy based models don't use backpropagation and automatic differentiation for computing gradients, instead turning to Markov Chain Monte Carlo methods.
3. DNNs often have several hidden layers. A restricted Boltzmann machine has only one hidden layer, however several RBMs can be stacked to make up Deep Belief Networks, of which they constitute the building blocks.

A standard BM network is divided into a set of observable and visible units $\hat{x}$ and a set of unknown hidden units/nodes $\hat{h}$.

Additionally there can be bias nodes for the hidden and visible layers. These biases are normally set to 1.

BMs are stackable, meaning they cwe can train a BM which serves as input to another BM. We can construct deep networks for learning complex PDFs. The layers can be trained one after another, a feature which makes them popular in deep learning

However, they are often hard to train. This leads to the introduction of so-called restricted BMs, or RBMS. Here we take away all lateral connections between nodes in the visible layer as well as connections between nodes in the hidden layer. The network is illustrated in the figure below.

figures/RBM.pdf

A. The network

1. A function $\mathbf{x}$ that represents the visible layer, a vector of M elements (nodes). This layer represents both what the RBM might be given as training input, and what we want it to be able to reconstruct. This might for example be the pixels of an image, the spin values of the Ising model, or coefficients representing speech.
2. The function $\mathbf{h}$ represents the hidden, or latent, layer. A vector of N elements (nodes). Also called "feature detectors".

The goal of the hidden layer is to increase the model's expressive power. We encode complex interactions between visible variables by introducing additional, hidden variables that interact with visible degrees of freedom in a simple manner, yet still reproduce the complex correlations between visible degrees in the data once marginalized over (integrated out).

Examples of this trick being employed in physics:

1. The Hubbard-Stratonovich transformation
2. The introduction of ghost fields in gauge theory
3. Shadow wave functions in Quantum Monte Carlo simulations

The network parameters, to be optimized/learned:

1. **a** represents the visible bias, a vector of same length as **x**.
2. **b** represents the hidden bias, a vector of same lenght as **h**.
3. W represents the interaction weights, a matrix of size $M \times N$.

B. Joint distribution

The restricted Boltzmann machine is described by a Boltzmann distribution

$$P_{rbm}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} e^{-\frac{1}{T_0} E(\mathbf{x}, \mathbf{h})}, \tag{2}$$

where Z is the normalization constant or partition function, defined as

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

It is common to ignore T_0 by setting it to one.

The function $E(\mathbf{x}, \mathbf{h})$ gives the **energy** of a configuration (pair of vectors) $(\mathbf{x}, \mathbf{h})$. The lower the energy of a configuration, the higher the probability of it. This function also depends on the parameters **a**, **b** and W. Thus, when we adjust them during the learning procedure, we are adjusting the energy function to best fit our problem.

An expression for the energy function is

$$E(\hat{x}, \hat{h}) = -\sum_{ia}^{NA} b_i^a \alpha_i^a(x_i) - \sum_{jd}^{MD} c_j^d \beta_j^d(h_j) - \sum_{ijad}^{NAMD} b_i^a \alpha_i^a(x_i) c_j^d \beta_j^d(h_j) w_{ij}^{ad}.$$

Here $\beta_j^d(h_j)$ and $\alpha_i^a(x_j)$ are so-called transfer functions that map a given input value to a desired feature value. The labels a and d denote that there can be multiple transfer functions per variable. The first sum depends only on the visible units. The second on the hidden ones. **Note** that there is no connection between nodes in a layer.

The quantities b and c can be interpreted as the visible and hidden biases, respectively.

The connection between the nodes in the two layers is given by the weights w_{ij}.

There are different variants of RBMs, and the differences lie in the types of visible and hidden units we choose as well as in the implementation of the energy function $E(\mathbf{x}, \mathbf{h})$.

a. Binary-Binary RBM: RBMs were first developed using binary units in both the visible and hidden layer. The corresponding energy function is defined as follows:

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{i}^{M} x_i a_i - \sum_{j}^{N} b_j h_j - \sum_{i,j}^{M,N} x_i w_{ij} h_j, \tag{4}$$

where the binary values taken on by the nodes are most commonly 0 and 1.

b. Gaussian-Binary RBM: Another varient is the RBM where the visible units are Gaussian while the hidden units remain binary:

$$E(\mathbf{x}, \mathbf{h}) = \sum_{i}^{M} \frac{(x_i - a_i)^2}{2\sigma_i^2} - \sum_{j}^{N} b_j h_j - \sum_{i,j}^{M,N} \frac{x_i w_{ij} h_j}{\sigma_i^2}. \tag{5}$$

C. Cost function

When working with a training dataset, the most common training approach is maximizing the log-likelihood of the training data. The log likelihood characterizes the log-probability of generating the observed data using our generative model. Using this method our cost function is chosen as the negative log-likelihood. The learning then consists of trying to find parameters that maximize the probability of the dataset, and is known as Maximum Likelihood Estimation (MLE). Denoting the parameters as $\theta = a_1, ..., a_M, b_1, ..., b_N, w_{11}, ..., w_{MN}$, the log-likelihood is given by

$$\mathcal{L}(\{\theta_i\}) = \langle \log P_\theta(x) \rangle_{data} \tag{6}$$

$$= -\langle E(x; \{\theta_i\}) \rangle_{data} - \log Z(\{\theta_i\}), \tag{7}$$

where we used that the normalization constant does not depend on the data, $\langle \log Z(\{\theta_i\}) \rangle = \log Z(\{\theta_i\})$ Our cost function is the negative log-likelihood, $\mathcal{C}(\{\theta_i\}) = -\mathcal{L}(\{\theta_i\})$

D. RBMs for the quantum many body problem

The idea of applying RBMs to quantum many body problems was presented by G. Carleo and M. Troyer, working with ETH Zurich and Microsoft Research.

Some of their motivation included

- "The wave function Ψ is a monolithic mathematical quantity that contains all the information on a quantum state, be it a single particle or a complex molecule. In principle, an exponential amount of information is needed to fully encode a generic many-body quantum state."
- There are still interesting open problems, including fundamental questions ranging from the dynamical properties of high-dimensional systems to the exact ground-state properties of strongly interacting fermions.
- The difficulty lies in finding a general strategy to reduce the exponential complexity of the full many-body wave function down to its most essential features. That is
  1. → Dimensional reduction
  2. → Feature extraction

- Among the most successful techniques to attack these challenges, artifical neural networks play a prominent role.

- Want to understand whether an artifical neural network may adapt to describe a quantum system.

E. Choose the right RBM

Carleo and Troyer applied the RBM to the quantum mechanical spin lattice systems of the Ising model and Heisenberg model, with encouraging results. Our goal is to test the method on systems of moving particles. For the spin lattice systems it was natural to use a binary-binary RBM, with the nodes taking values of 1 and -1. For moving particles, on the other hand, we want the visible nodes to be continuous, representing position coordinates. Thus, we start by choosing a Gaussian-binary RBM, where the visible nodes are continuous and hidden nodes take on values of 0 or 1. If eventually we would like the hidden nodes to be continuous as well the rectified linear units seem like the most relevant choice.

F. Representing the wave function

The wavefunction should be a probability amplitude depending on x. The RBM model is given by the joint distribution of x and h

$$F_{rbm}(\mathbf{x},\mathbf{h}) = \frac{1}{Z} e^{-\frac{1}{T_0}E(\mathbf{x},\mathbf{h})}. \tag{8}$$

To find the marginal distribution of x we set:

$$F_{rbm}(\mathbf{x}) = \sum_{\mathbf{h}} F_{rbm}(\mathbf{x},\mathbf{h}) \tag{9}$$

$$= \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}. \tag{10}$$

Now this is what we use to represent the wave function, calling it a neural-network quantum state (NQS)

$$\Psi(\mathbf{X}) = F_{rbm}(\mathbf{x}) \tag{11}$$

$$= \frac{1}{Z}\sum_{h} e^{-E(\mathbf{x},\mathbf{h})} \tag{12}$$

$$= \frac{1}{Z}\sum_{\{h_j\}} e^{-\sum_i^M \frac{(x_i-a_i)^2}{2\sigma^2} + \sum_j^N b_j h_j + \sum_{i,j}^{M,N} \frac{x_i w_{ij} h_j}{\sigma^2}} \tag{13}$$

$$= \frac{1}{Z} e^{-\sum_i^M \frac{(x_i-a_i)^2}{2\sigma^2}} \prod_j^N (1 + e^{b_j + \sum_i^M \frac{x_i w_{ij}}{\sigma^2}}). \tag{14}$$

$$\tag{15}$$

G. Choose the cost function

Now we don't necessarily have training data (unless we generate it by using some other method). However, what we do have is the variational principle which allows us to obtain the ground state wave function by minimizing the expectation value of the energy of a trial wavefunction (corresponding to the untrained NQS). Similarly to the traditional variational Monte Carlo method then, it is the local energy we wish to minimize. The gradient to use for the stochastic gradient descent procedure is

$$G_i = \frac{\partial \langle E_L\rangle}{\partial \theta_i} = 2(\langle E_L \frac{1}{\Psi}\frac{\partial \Psi}{\partial \theta_i}\rangle - \langle E_L\rangle\langle \frac{1}{\Psi}\frac{\partial \Psi}{\partial \theta_i}\rangle), \tag{16}$$

where the local energy is given by

$$E_L = \frac{1}{\Psi}\hat{\mathbf{H}}\Psi. \tag{17}$$

The trial wave function is based on the product of a Slater determinant with Gaussian orbitals, a simple Jastrow factor $\exp(r_{ij})$ and the reduced Boltzmann machines.

figures/figN2.pdf

V. CONCLUSIONS AND WHERE DO WE STAND

- A simple extension of the work of G. Carleo and M. Troyer, Science **355**, Issue 6325, pp. 602-606 (2017) gives excellent results for two-electron systems as well as good agreement with standard VMC calculations for $N = 6$ and $N = 12$ electrons.
- Minimization problem can be tricky.
- Anti-symmetry dealt with multiplying the trail wave function with an optimized Slater determinant.
- To come: Analysis of wave function from ML and compare with diffusion and Variational Monte Carlo calculations as well as the analytical results of Taut for the two-electron case.
- Extend to more fermions. How do we deal with the antisymmetry of the multi-fermion wave function?
    1. Here we used standard Hartree-Fock theory to define an optimal Slater determinant. Takes care of the antisymmetry. What about constructing an anti-symmetrized network function?
    2. Use thereafter ML to determine the correlated part of the wafe function (including a standard Jastrow factor).
    3. Test this for multi-fermion systems and compare with other many-body methods.
- Can we use ML to find out which correlations are relevant and thereby diminish the dimensionality problem in say CC or SRG theories?