Johannes Mellak, BSc

Solving the Quantum Many-Body Problem Using Artificial Neural Networks

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Technical Physics

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Phys. Dr.rer.nat.
Wolfgang von der Linden
Institute of Theoretical and Computational Physics

Graz, July 2020

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date, Signature

Abstract

Solving strongly correlated quantum many-body problems requires numerical approximations due to the exponential size of the Hilbert space. Variational Monte-Carlo methods are limited by the expressive power and flexibility of the variational forms. To describe the correlations in many-body wavefunctions efficiently, reducing them to the most important features is necessary. This is also a typical problem encountered in machine learning applications. A new approach has been recently put forward, showing that it is possible to encode correlated quantum states with artificial neural networks. Particularly, variational optimization of a restricted Boltzmann machine was shown to achieve competitive results for strongly correlated quantum spin models.

In this thesis the theoretical foundation of variational quantum Monte-Carlo is laid out and placed in the terminology of machine learning. Based on this, the proposed method of *neural network quantum states* was investigated. The early results for the Heisenberg model were quickly reproduced and the main focus was put towards implementing the fermionic Hubbard model within this framework. The results suggest that sign structures are more difficult to learn efficiently in agreement with recent literature. A reference state implementation, restricting the variational ansatz to describe only correlations, achieves better results and reproduces the so far only published research applying NQS to the Hubbard model.

Kurzfassung

Die Lösung von stark korrelierten quantenmechanischen Vielteilchenproblemen erfordert numerische Näherungsmethoden aufgrund des exponentiell anwachsenden Hilbertraums. Die variationelle Monte-Carlo Methode ist durch die oft geringe Flexibilität der Testfunktionen limitiert. Um Korrelationen in den Vielteilchen-Wellenfunktionen effizient zu beschreiben, müssen diese auf die wichtigsten Eigenschaften reduziert werden – ein ebenfalls für Anwendungen maschinellen Lernens typisches Problem. Mit einem neuen Verfahren konnte gezeigt werden, dass künstliche neuronale Netze in der Lage sind, korrelierte Quantenzustände abzubilden. Insbesondere konnten durch variationelles Optimieren einer Restricted Boltzmann Machine vielversprechende Ergebnisse für Quanten-Spin Modelle erzielt werden.

In dieser Masterarbeit werden die theoretischen Grundlagen der variationellen Quanten Monte-Carlo Methode dargelegt und in Beziehung zur Terminologie des Maschinellen Lernens gesetzt. Darauf aufbauend wurde die vorgeschlagene Methode der *Neural Network Quantum States* behandelt und die ersten Ergebnisse für das Heisenberg Modell reproduziert. Der Hauptteil der Arbeit bestand aus der Implementierung der Lösung des Fermi-Hubbard Modells. Die erhaltenen Ergebnisse legen nahe, dass das Lernen der Vorzeichenstruktur für dieses Verfahren ein Problem darstellt, was auch durch neueste Veröffentlichungen bestätigt scheint. Weiters wurde ein Referenzzustand-Ansatz implementiert, um das neuronale Netzwerk auf das Lernen der Korrelationen zu beschränken. Damit konnte ein besseres Ergebnis erzielt und die bisher einzige publizierte Anwendung dieser Methode auf das Hubbard Model bestätigt werden.

Acknowledgements

I was fortunate to have Professor Wolfgang von der Linden as my supervisor who brought this interesting new research topic to my attention. Thank you for always being there to offer your expertise and support and making it a pleasure to work with you. I also want to thank Professor Enrico Arrigoni, both he and Professor von der Linden took the time on a weekly basis to look at new results, discuss technicalities and always provided insightful answers to my questions.

I want to thank Sabrina for keeping up my good spirits during the time of self-isolation caused by the COVID-19 pandemic that started to keep the world in suspense when I was in the midst of working on my thesis.

To all of my friends and family I also want to express my gratitude for helping to keep me sane during these troubling times – after it was possible to meet again.

I also consider myself quite fortunate to have been able to attend the ESI Winter School Machine Learning in Physics, organized by the Vienna Doctoral School in Physics, right before the pandemic shut down all of Austria for many weeks. Meeting inspiring people and being introduced to fascinating projects fueled my enthusiasm for this kind of research.

Contents

List of Figures

Chapter 1

Introduction

Ab initio simulations of condensed matter plays an important role in modern physics. Solving the quantum mechanical many-body problem of interacting electrons to find the discrete energy levels of the system comes down to solving the time-independent Schrödinger equation

$$\hat{H}|\psi_n\rangle = E_n|\psi_n\rangle \,. \tag{1.1}$$

After the Born-Oppenheimer approximation, decoupling the movements of the atomic nuclei from the electrons, the resulting many-body problem for the electron system is still too complicated to solve. The many-body Hamiltonian for the electronic system in second quantization can be written as

$$\begin{aligned}\hat{H} &= \hat{H}_0 + \hat{H}_1 \\ &= -\sum_{i,j,\sigma} t_{ij}\hat{c}^{\dagger}_{i\sigma}\hat{c}_{j\sigma} + \sum_{i,j,k,l,\sigma,\sigma'} V_{ijkl}\hat{c}^{\dagger}_{i\sigma}\hat{c}_{j\sigma}\hat{c}^{\dagger}_{k\sigma'}\hat{c}_{l\sigma'}\end{aligned} \tag{1.2}$$

with σ, σ' denoting the spin and each of the remaining indices defining the other quantum numbers, indexing an orbital or a specific site in a lattice model. The electrons will minimize their kinetic energy by *hopping* between lattice sites, which is described by the first part $\hat{H}_0$. Whenever these itinerant electrons come close to one another, the electrostatic repulsion of their charges raises the Coulomb energy, which is described by $\hat{H}_1$. If the considered orbitals ϕ are strongly localized, the hopping which is defined by the parameter

$$t_{ij} = \int \phi_i^*(\vec{x})\,\hat{H}_0\,\phi_j(\vec{x})\,\mathrm{d}V_x \,, \tag{1.3}$$

can be limited to nearest neighbors $<ij>$, leading to the Tight Binding Hamiltonian

$$\hat{H}_0 = -t\sum_{<ij>,\sigma} \hat{c}^{\dagger}_{i\sigma}\hat{c}_{j\sigma} \,. \tag{1.4}$$

Furthermore, the Coulomb parameter

$$V_{ijkl} = \int\int \phi_i^*(\vec{x})\phi_j(\vec{x})\,\frac{1}{|\vec{x}-\vec{x}'|}\,\phi_k^*(\vec{x}')\phi_l(\vec{x}')\,\mathrm{d}V_x\mathrm{d}V_{x'} \tag{1.5}$$

can be simplified by assuming the orbitals to be strongly localized $i = j$ and $k = l$, and considering the screening by the other electrons leading to an exponential decay of the interaction and $i = k$. Finally, demanding translational invariance, the so calld *Hubbard-U* is introduced as $U = V_{iiii}$. Because of the Pauli exclusion principle, only electrons with different spin quantum numbers can occupy the same site, hence H_1 counts the doubly occupied sites and increases their energy

$$\hat{H}_1 = U \sum_i \hat{c}^\dagger_{i\sigma} \hat{c}_{i\sigma} \hat{c}^\dagger_{i\sigma'} \hat{c}_{i\sigma'} = U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \,. \tag{1.6}$$

The resulting total Hamiltonian is called the single-band Hubbard Model

$$\hat{H} = -t \sum_{<ij>,\sigma} \hat{c}^\dagger_{i\sigma} \hat{c}_{j\sigma} + U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \,. \tag{1.7}$$

The behavior of this class of model systems is defined by the competition between itinerancy and localization of the electrons, which as a result are usually highly correlated.

Models such as the Hubbard model cannot generally be solved analytically and numerical approximations are very computationally expensive because the Hilbert space of all possible configurations grows exponentially with the number of sites. Especially fermionic and frustrated systems escape an efficient treatment with existing Monte-Carlo approximations, as they evoke the infamous fermionic sign problem [27]. Better numerical tools to compute properties of strongly correlated electron systems in condensed matter or quantum chemistry could help understand novel physical phases such as high-temperature superconductivity and would bring advancements in the design of new materials.

Simplified lattice models also offer a good benchmark to compare new algorithms to established solutions and to test their performance. One such new algorithm was published by Carleo and Troyer [3] in 2017 and entailed numerous different proposals to solve quantum many-body problems with a new set of tools. They essentially showed that artificial neural networks are capable of efficiently encoding strongly correlated quantum states and can be optimized as variational wavefunctions in a reinforcement learning scheme, calling their ansatz *neural network quantum states* (NQS). In recent years machine learning frameworks and especially deep multi-layered neural networks have set new benchmarks in problems usually involving dimensional reduction of very large feature-spaces and pattern recognition. Modern networks applied to image recognition problems or playing games make use of deep learning architectures that are able to optimize functions with millions of parameters. Many aspects of these problems are common between computational quantum physics and what is sometimes phrased *extreme* data science [9], and there has been an active search for possibilities to exchange knowledge between both fields.

In this thesis the proposed combination of the variational quantum Monte-Carlo method and artificial neural networks, in particular the so called restricted Boltzmann machine is investigated. After this short introduction, some more physical considerations about quantum many-body models are discussed to understand the following numerical treatments. In Chapter 3 the variational Monte-Carlo technique as the starting point for optimizing neural network quantum states is presented before a short overview of some fundamental concepts from the field of machine learning is given in Chapter 4. The newly proposed variational ansatz is also finally explained there. After reproducing some results for the Heisenberg model, the Hubbard model is tackled, writing extensions to a dedicated software library for quantum spin models to allow for simulating fermions. While attempting to solve the Hubbard model, introducing a reference state and thereby putting in more physical prior knowledge seems necessary to improve the obtained results. Finally some problems and opportunities facing future research are outlined.

Chapter 2

Strongly Correlated Quantum Many-Body Problems

In this Chapter selected definitions and concepts of quantum many-body physics are presented, which are later used for treating the discussed models numerically with the already mentioned neural network quantum states.

2.1 Quantum Many-Body Problems

The eigenstates and associated energy levels of quantum mechanical many-body systems are obtained by solving the time independent Schrödinger equation eigenvalue problem

$$\hat{H}|\psi_n\rangle = E_n|\psi_n\rangle \,, \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian of the system, describing the interactions and energies, and E_n is the energy of the state $|\psi_n\rangle$. Since only the theoretical concepts and their numerical evaluation are discussed, fundamental constants are set to one for the sake of simplicity. Throughout this thesis Dirac notation and second quantization is used. In this way a specific state $|\psi\rangle$ with a particle with spin σ at site i can be written as

$$|\psi\rangle = \hat{c}^\dagger_{i\sigma}|0\rangle \tag{2.2}$$

using the creation operator $\hat{c}^\dagger_{i\sigma}$, which *creates* a fermionic particle at site i with spin σ when acting on the vacuum state $|0\rangle$. The annihilation operator $\hat{c}_{i\sigma}$ on the other hand *destroys* a particle at site i, if there is one. For these fermionic operators the anticommutation relations apply

$$\begin{aligned} \{\hat{c}^\dagger_{i\sigma}, \hat{c}_{j\sigma'}\} &= \hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma'} + \hat{c}_{j\sigma'}\hat{c}^\dagger_{i\sigma} = \delta_{ij}\delta_{\sigma\sigma'} \\ \{\hat{c}_{i\sigma}, \hat{c}_{j\sigma'}\} &= \{\hat{c}^\dagger_{i\sigma}, \hat{c}^\dagger_{j\sigma'}\} = 0 \,. \end{aligned} \tag{2.3}$$

The basis states in real-space can be written as

$$|x\rangle = |n_{1\uparrow}, n_{2\uparrow}, \ldots n_{L\uparrow}, n_{1\downarrow}, n_{2\downarrow}, \ldots n_{L\downarrow}\rangle = \prod_{\sigma=\uparrow,\downarrow}\prod_{i=1}^{L}\left(\hat{c}^\dagger_{i\sigma}\right)^{n_{i\sigma}}|0\rangle \tag{2.4}$$

where $x = (n_{1\uparrow}, \ldots n_{L\uparrow}, n_{1\downarrow}, \ldots n_{L\downarrow})$ is used as a shorthand notation for all the fermionic occupation numbers $n_{i\sigma} \in \{0, 1\}$ for the i-th of the L lattice sites with spin σ. In numerical implementations x can also be identified as an integer-valued index of a specific configuration. A general many-body state $|\psi\rangle$ expanded in this basis

$$|\psi\rangle = \sum_x |x\rangle\langle x|\psi\rangle = \sum_x \psi(x)|x\rangle \tag{2.5}$$

assigns each possible configuration $|x\rangle$ the complex weight $\psi(x) = \langle x|\psi\rangle$. Assuming a single orbital per lattice site, there are four possible occupations per site, $|0\rangle$, $|\uparrow\rangle$, $|\downarrow\rangle$ and $|\uparrow\downarrow\rangle$ leading to 4^L different basis states. The dimension of the Hilbert space grows exponentially, which is why even supercomputers can usually only handle a few tens of lattice sites.

2.2 Hubbard Model

The Hamiltonian of the single-band Hubbard Model that was heuristically derived in the introduction

$$\hat{H} = -t \sum_{<ij>,\sigma} \hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma} + U \sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow} \tag{2.6}$$

consists of a kinetic energy term, that lowers the energy if hopping of nearest-neighbor electrons is possible, and the so called *Hubbard* or Coulomb term which raises the energy for double occupations on a lattice site due to Coulomb repulsion. Terms for the chemical potential or an external magnetic field are not included here. The parameter t is set to 1 in the remainder of this work, as only the fraction U/t determines the physical properties, but for the numerical treatment the individual values make a difference. Different physical properties appear when varying the number of electron and usually the half-filled case with $N_e = L$ is discussed. The Hubbard model can describe an effect called *Mott insulation*, which arises from the Coulomb repulsion of the electrons at large U/t, unlike the common band insulating state. This happens only at half-filling, because in any other case there is necessarily either a doubly occupied site or an empty one [8]. The kinetic energy and Coulomb repulsion parts of the operator are diagonal in momentum space and real space respectively, but the combination of both can't be solved trivially in either basis. Consequently two limits of the model parameters U/t can be identified. If the Coulomb interactions are completely screened $U \ll t$, the electrons can move freely and optimize the kinetic energy, leading to the tight binding model. If on the other hand $U \gg t$, the electrons will hinder each other from moving as double occupations are prohibited. At half-filling there will be exactly one electron per site and the system can be identified as a Heisenberg model of spins.

2.2.1 Particle-Hole Transformation

At half-filling the Hubbard model is particle-hole symmetric, meaning that the same results are achieved if particles and holes are swapped. On a bipartite lattice a staggered particle-hole transformation

$$\begin{aligned} \hat{c}_{i\downarrow} &\to (-1)^i\, \hat{c}^\dagger_{i\downarrow} \\ \hat{c}_{i\uparrow} &\to \hat{c}_{i\uparrow} \end{aligned} \tag{2.7}$$

maps to an attractive model, which is easier to solve for some numerical methods [16]. The kinetic energy term remains unchanged, while for the ↓ spins the occupation number operator becomes

$$\hat{n}_{i\downarrow} = \hat{c}^{\dagger}_{i\downarrow}\hat{c}_{i\downarrow} \to (-1)^i(-1)^i \hat{c}_{i\downarrow}\hat{c}^{\dagger}_{i\downarrow} = 1 - \hat{c}^{\dagger}_{i\downarrow}\hat{c}_{i\downarrow} = 1 - \hat{n}_{i\downarrow} \,. \tag{2.8}$$

Inserting this into Equation (2.6) the resulting Hamiltonian now has a negative interaction term $U \to -U$ and an additional external field for one spin channel $U\sum_i \hat{n}_{i\uparrow} = UN_{\uparrow}$, which can be cancelled at half-filling when considering a chemical potential.

2.3 Tight Binding Model

If the electrons can move freely only the kinetic energy term remains

$$\hat{H} = -t \sum_{l,\sigma} \left(\hat{c}^{\dagger}_{l\sigma}\hat{c}_{(l+1)\sigma} + \hat{c}^{\dagger}_{(l+1)\sigma}\hat{c}_{l\sigma} \right) \,. \tag{2.9}$$

Only hopping in one dimension is considered in this section, as each additional dimension can be added independently in the sum. The ground state of the tight binding (TB) model, called the Fermi-sea because the lattice is filled with freely moving electrons up to the Fermi energy, can be derived exactly in momentum space using the Fourier transformation. Inserting the Fourier transformed operators

$$\hat{c}^{\dagger}_{l\sigma} = \frac{1}{\sqrt{L}} \sum_k e^{ikl} \hat{c}^{\dagger}_{k\sigma} \tag{2.10}$$

and

$$\hat{c}_{l\sigma} = \frac{1}{\sqrt{L}} \sum_p e^{-ipl} \hat{c}_{p\sigma} \tag{2.11}$$

the Hamiltonian transforms into

$$\begin{aligned} \hat{H} &= -t \sum_{l,\sigma} \frac{1}{L} \sum_{k,p} e^{ikl} \hat{c}^{\dagger}_{k\sigma} e^{-ip(l+1)} \hat{c}_{p\sigma} + \text{c.c.} \\ &= -t \sum_{k,p,\sigma} \underbrace{\frac{1}{L} \sum_{l} e^{i(k-p)l}}_{\delta_{kp}} e^{-ik} \hat{c}^{\dagger}_{k\sigma}\hat{c}_{p\sigma} + \text{c.c.} \\ &= -t \sum_{k,\sigma} \hat{c}^{\dagger}_{k\sigma}\hat{c}_{k\sigma} \left(e^{-ik} + e^{ik}\right) \\ &= -2t \sum_{k,\sigma} \cos(k)\, \hat{c}^{\dagger}_{k\sigma}\hat{c}_{k\sigma} = \sum_{k,\sigma} \underbrace{(-2t)\cos(k)}_{\epsilon(k)} \hat{n}_{k\sigma} \,. \end{aligned} \tag{2.12}$$

This operator is diagonal in k with the eigenvalues $\epsilon(k)$. Adding hopping terms in another dimension to the Hamiltonian, the energy levels for a two-dimensional lattice simply become

$$\epsilon(\vec{k}) = -2t\cos(k_x) - 2t\cos(k_y) \,. \tag{2.13}$$

The Fermi-sea state is obtained by filling the lowest k-levels with the N_e electrons. The Fermi-energy ϵ_F at zero temperature is equal to the largest occupied energy level.

2.3.1 Periodic and Anti-Periodic Boundary Conditions

Applying periodic boundary conditions (PBC) of the lattice to the wavefunction e^{ikl}, the allowed $k \in (-\pi, \pi]$, using

$$e^{ikl} = e^{ik(l+L)} = e^{ikl} e^{ikL - i2\pi n},\ n \in \mathbb{Z}\,,$$

are determined by

$$k = \frac{n2\pi}{L},\ n \in \mathbb{Z}\,. \tag{2.14}$$

With antiperiodic boundary conditions (APBC) the waves get a phase jump of $i\pi$ when crossing over a boundary

$$e^{ikl} = e^{ik(l+L)} e^{i\pi} = e^{ikl} e^{ikL - i\pi(2n-1)},\ n \in \mathbb{Z}\,,$$

so now the k-values are

$$k = \frac{(2n-1)\pi}{L},\ n \in \mathbb{Z}\,. \tag{2.15}$$

This is used in Chapter 5 to lift the degeneracy of the TB ground state. The allowed k-values and the corresponding energies are plotted in Fig. 2.1.

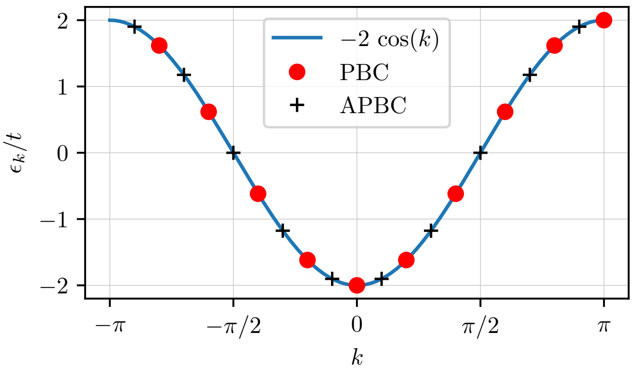


Figure 2.1: Allowed k-values and their energies for the 1D Tight Binding model with $L = 10$ sites using PBC or APBC.

2.3.2 Projection of the Fermi-Sea State to Real Space

To be able to count double occupations starting from the Fermi-sea state

$$|\mathrm{FS}\rangle = \prod_{k}^{\epsilon(k)<E_F} \hat{c}^\dagger_{k\uparrow} \prod_{k'}^{\epsilon(k')<E_F} \hat{c}^\dagger_{k'\downarrow} |0\rangle \tag{2.16}$$

with the lowest k-levels up to the Fermi energy E_F filled, the projection to real-space $\langle x|\mathrm{FS}\rangle$ is needed. Similarly to the reformulation of the Hamiltoninan in Eq. (2.12) in terms of momentum-space operators, the inverse Fourier-transforms are inserted

$$\begin{aligned} |\mathrm{FS}\rangle &= \prod_{k}^{\epsilon(k)<E_F} \hat{c}^\dagger_{k\uparrow} \prod_{k'}^{\epsilon(k')<E_F} \hat{c}^\dagger_{k'\downarrow} |0\rangle \\ &= \prod_{k}^{\epsilon(k)<E_F} \left(\frac{1}{\sqrt{L}} \sum_l e^{-ikl} \hat{c}^\dagger_{l\uparrow} \right) \prod_{k'}^{\epsilon(k')<E_F} \left(\frac{1}{\sqrt{L}} \sum_l e^{-ik'l} \hat{c}^\dagger_{l\downarrow} \right) |0\rangle\,. \end{aligned} \tag{2.17}$$

To write this state in the real-space basis, the creation operators $\hat{c}^{\dagger}_{l\sigma}$ have to be rearranged to fulfill the standard ordering convention defined in Equation (2.4). Applying the anticommutation relation, a minus sign is collected each time two operators are exchanged. This can be demonstrated for a simple two-electron state, discarding the spin momentarily,

$$\begin{aligned}
|k_1 k_2\rangle &= \hat{c}^{\dagger}_{k_1}\hat{c}^{\dagger}_{k_2}|0\rangle \\
&= \frac{1}{L}\sum_{l_1} e^{-ik_1 l_1}\hat{c}^{\dagger}_{l_1}\sum_{l_2} e^{-ik_2 l_2}\hat{c}^{\dagger}_{l_2}\,|0\rangle \\
&= \frac{1}{L}\left(e^{-ik_1 1}\hat{c}^{\dagger}_{1} + \cdots + e^{-ik_1 L}\hat{c}^{\dagger}_{L}\right)\left(e^{-ik_2 1}\hat{c}^{\dagger}_{1} + \cdots + e^{-ik_2 L}\hat{c}^{\dagger}_{L}\right)|0\rangle \\
&= \frac{1}{L}\sum_{l_1<l_2}\left(e^{-ik_1 l_1}e^{-ik_2 l_2} - e^{-ik_2 l_1}e^{-ik_1 l_2}\right)\hat{c}^{\dagger}_{l_1}\hat{c}^{\dagger}_{l_2}\,|0\rangle\,.
\end{aligned} \tag{2.18}$$

After performing the multiplication, each configuration in $\{l_1, l_2\}$, with an electron at lattice sites l_1 and l_2, appears exactly twice. Once where the order is correct, $l_1 < l_2$, and a second time where the operators have to be exchanged to the correct order, collecting a minus sign. The last line can be rewritten as a determinant of the Bloch factors

$$|k_1 k_2\rangle = \frac{1}{L}\sum_{l_1<l_2}\begin{vmatrix} e^{-ik_1 l_1} & e^{-ik_1 l_2} \\ e^{-ik_2 l_1} & e^{-ik_2 l_2}\end{vmatrix}\hat{c}^{\dagger}_{l_1}\hat{c}^{\dagger}_{l_2}\,|0\rangle\,. \tag{2.19}$$

This can then be systematically expanded [8] to the full system of $N_e = N_\uparrow + N_\downarrow$ electrons from Eq. (2.17) to obtain

$$|\mathrm{FS}\rangle = L^{-\frac{N_e}{2}}\sum_x \begin{vmatrix} e^{-ik_1 l_1^{\uparrow}} & \dots & e^{-ik_1 l_{N_\uparrow}^{\uparrow}} \\ \vdots & & \vdots \\ e^{-ik_{N_\uparrow} l_1^{\uparrow}} & \dots & e^{-ik_{N_\uparrow} l_{N_\uparrow}^{\uparrow}}\end{vmatrix}\begin{vmatrix} e^{-ik_1 l_1^{\downarrow}} & \dots & e^{-ik_1 l_{N_\downarrow}^{\downarrow}} \\ \vdots & & \vdots \\ e^{-ik_{N_\downarrow} l_1^{\downarrow}} & \dots & e^{-ik_{N_\downarrow} l_{N_\downarrow}^{\downarrow}}\end{vmatrix}|x\rangle\,. \tag{2.20}$$

Here the configuration x, introduced in Eq. (2.4), and the notation l_n^σ for the site-index of the n-th electron with spin σ are used.

2.4 The Heisenberg Model of Spins

In the other limit of large U/t the half-filled Hubbard model reduces to the antiferromagnetic ($J = 4t^2/U > 0$) Heisenberg model of interacting spins with the Hamilton operator [8]

$$\hat{H} = J\sum_{<ij>}\hat{\boldsymbol{S}}_i \cdot \hat{\boldsymbol{S}}_j\,. \tag{2.21}$$

For a spin-$\frac{1}{2}$ model of L spins there are 2^L different configurations

$$|x\rangle = |s_1^z, \dots s_L^z\rangle \tag{2.22}$$

with $s_i^z \in \{-\frac{1}{2}, +\frac{1}{2}\}$. The local states in the z-basis $|\downarrow\rangle$ and $|\uparrow\rangle$ are represented by $\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$, and have the eigenvalues $s^z = -\frac{1}{2}$ and $s^z = \frac{1}{2}$ respectively. The local spin-operators $\hat{S}_i$ are given by the Pauli matrices. The Hamiltonian can be rewritten to

$$\begin{aligned}\hat{H} &= J \sum_{<ij>} \left(\hat{S}_i^{\ x} \hat{S}_j^{\ x} + \hat{S}_i^{\ y} \hat{S}_j^{\ y} + \hat{S}_i^{\ z} \hat{S}_j^{\ z} \right) \\ &= J \sum_{<ij>} \left(\frac{1}{2} (\hat{S}_i^{\ +} \hat{S}_j^{\ -} + \hat{S}_i^{\ -} \hat{S}_j^{\ +}) + \hat{S}_i^{\ z} \hat{S}_j^{\ z} \right)\end{aligned} \tag{2.23}$$

using the ladder operators $\hat{S}_j^{\pm} = \hat{S}_j^x \pm i \hat{S}_j^y$, revealing *hopping* terms between neighboring spins. This first part of $\hat{H}$ is called the exchange energy. The operators $\hat{S}$ now have the form

$$\hat{S}^z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \hat{S}^+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \hat{S}^- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{2.24}$$

such that $\hat{S}^z$ retrieves the z component of the spin $\hat{S}^z |s_i^z\rangle = s_i^z |s_i^z\rangle$, and the ladder operators flip the spin if possible, $\hat{S}^+ |\downarrow\rangle = |\uparrow\rangle$, $\hat{S}^+ |\uparrow\rangle = 0$ and analogically for $\hat{S}^-$. The model of spins can be mapped to a model of spinless fermions with the s^z values corresponding to the occupation numbers, but only in one dimension since particles would catch a fermionic minus sign if they were able to switch places.

2.5 Numerical Solutions

There are different approaches to solve quantum many-body problems numerically, performing better or worse in different scenarios. Exact diagonalization schemes such as Lanczos work reliably with general Hamiltonians but are limited to very small system sizes as the dimension of the Hilbert-space increases exponentially. While density functional theory is very accurate for weakly correlated systems and is often used for calculating real materials, effectively treating a noninteracting system, it fails when correlations become too strong [9].

Other approaches try to overcome the exponential complexity either by compression of the wavefunction or by evaluating the high-dimensional integrations stochastically in quantum Monte-Carlo (QMC) schemes. The density-matrix renormalization group (DMRG) has established itself as a very powerful method to solve strongly correlated quantum lattice problems in one dimension [20]. Matrix product states (MPS) or the more general tensor networks offer an efficient representation of the full wavefunction by truncating the tensors to keep only the most important features. However, the entanglement entropy as the limiting factor that usually follows the area law scales unfavorably when looking at higher dimensions [21]. QMC approaches offer a direct stochastic solution to the Schrödinger equation but for a lot of interesting fermionic or frustrated models suffer from the sign problem where the appearance of negative Boltzmann weights results in an exponential growth of the statistical error with the number of particles [27].

Variational methods typically rely on rigid forms, for example fixed node approximations, and their expressive power cannot be systematically extended in an efficient way [6]. The general idea of variational Monte-Carlo (VMC) approaches is described in detail in the following Chapter. It promises not to suffer from the typical negative sign problem [15] and based on this approach the proposed solution incorporating notions from machine learning will be discussed. This algorithm, in different terms, can be regarded as a *reinforcement learning* scheme and using an artificial neural network as the variational form in Chapter 4 ultimately brings this method into the realm of machine learning.

Chapter 3

Variational Monte-Carlo

The theoretical foundation for this family of algorithms is described in the following sections, as it is used for the neural network quantum states presented in Chapter 4.

3.1 Variational Monte-Carlo

In variational Monte-Carlo (VMC) a trial wavefunction is optimized to minimize the energy expectation value. Using the Rayleigh-Ritz variational principle

$$E[\psi] = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0 \tag{3.1}$$

the exact groundstate energy E_0 presents a lower boundary and the time-independent Schrödinger equation can be reformalized as a stochastic optimization problem. The energy functional $E[\psi]$ can be minimized by varying the wavefunction ψ to find the groundstate

$$\psi_0 = \underset{\psi}{\operatorname{argmin}}\, E[\psi]\,. \tag{3.2}$$

The expectation value of the operator can be evaluated stochastically using Markov-Chain Monte-Carlo integration which gives this method its name. A variational optimization is needed that offers stability when being exposed to statistically noisy Monte-Carlo samples.

3.2 Statistical Expectation Value of a Local Operator

Inserting full basis sets $\mathbb{1} = \sum_x |x\rangle\langle x|$ and using the definitions $\psi(x) = \langle x|\psi\rangle$ for the wavefunction amplitude and phase and the matrix element $H_{xx'} = \langle x|\hat{H}|x'\rangle$, Eq. (3.1) can be rewritten as

$$
\begin{aligned}
E = \langle \hat{H} \rangle = \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} &= \frac{\sum_{x,x'} \langle \psi | x \rangle \langle x | \hat{H} | x' \rangle \langle x' | \psi \rangle}{\sum_x \langle \psi | x \rangle \langle x | \psi \rangle} \\
&= \frac{\sum_{x,x'} \psi^*(x) H_{xx'} \psi(x')}{\sum_x \psi^*(x) \psi(x)} \\
&= \frac{\sum_x \psi^*(x) \psi(x) \sum_{x'} H_{xx'} \frac{\psi(x')}{\psi(x)}}{\sum_x |\psi(x)|^2} \\
&= \sum_x p(x) E_{\text{loc}}(x) \\
&= \langle\langle E_{\text{loc}} \rangle\rangle
\end{aligned}
\tag{3.3}
$$

with the definition of the probability distribution

$$
p(x) = \frac{|\psi(x)|^2}{\sum_{x'} |\psi(x')|^2} \, . \tag{3.4}
$$

$\langle\langle\, . \,\rangle\rangle$ denotes the statistical expectation value over the probability distribution $p(x)$ and the final result is the expectation value of the so called local energy

$$
E_{\text{loc}}(x) = \sum_{x'} H_{xx'} \frac{\psi(x')}{\psi(x)} \, . \tag{3.5}
$$

Naturally, in place of the Hamilton operator, for any quantum mechanical observable $\hat{O}$ the expectation value can be calculated in this way. If $\hat{O}$ is diagonal in the computational basis, it can be evaluated right away as $\langle\langle \hat{O} \rangle\rangle$, whereas in the general case $\hat{O}_{\text{loc}}$ is calculated as it is written in Eq. (3.5) for the Hamiltonian. In this formulation, the sum over exponentially many basis vectors can then be approximated with N_S Monte-Carlo samples

$$
\begin{aligned}
\langle \hat{H} \rangle &= \sum_x p(x) E_{\text{loc}}(x) \\
&\approx \frac{1}{N_S} \sum_{x \sim p(x)} E_{\text{loc}}(x)
\end{aligned}
\tag{3.6}
$$

while the sum in $E_{\text{loc}}(x)$ (Eq. (3.5)) is carried out exactly over the small subspace of connected states x' that are reached from x, that means for which $H_{xx'} \neq 0$.

3.2.1 Markov Chain Monte-Carlo Sampling

To evaluate Eq. (3.6), N_S configurations are drawn according to the probability distribution $p(x)$ using the Metropolis algorithm. Belonging to the family of Markov-chain Monte-Carlo (MCMC) methods, it creates a sequence of configurations where each sample is only dependent on the previous one. Starting from an initial configurations x, a new configuration x' is proposed and accepted as the next sample with the probability

$$
p(x'|x) = \min\left(1, \frac{|\psi(x')|^2}{|\psi(x)|^2}\right) \, , \tag{3.7}
$$

if it gets rejected, the previous sample is taken again. Because only the quotient of the wavefunctions is used, no normalization constant is needed. A number of samples are discarded in a so called thermalization period in the beginning. Two possible ways to get an update step x' are to either perform a *hopping* step to a configuration connected by the Hamiltonian, or to exchange the occupations of two sites [15]. Both conserve the number of particles and the total spin S^z. If total spin conservation is not needed, additional local spin flips can be used, and a lot more advanced update procedures have been developed in the framework of quantum Monte-Carlo.

3.3 Trial Wavefunctions

The variational trial wavefunctions should be able to capture as much complexity and be as accurate as possible, but still be fast to calculate. If more physics is already built into the form of the wavefunction, a good representation will usually be found quicker. But this also limits the expressiveness and prohibits finding new and unknown representations, so there is an ongoing search for efficient representations of quantum states. The simplest example of a trial state for solving the Hubbard model is the idea of Gutzwiller [11], to start with an uncorrelated metal state $|\Phi_0\rangle$ represented by the Fermi-sea, and then to start suppressing double occupations. A variational parameter is introduced to find the optimum number of doubly occupied and empty sites when U increases from 0. By including the Gutzwiller factor

$$|\psi\rangle = \prod_j \left[1 - (1-\eta)\hat{n}_{j\uparrow}\hat{n}_{j\downarrow}\right] |\Phi_0\rangle \tag{3.8}$$

with the variational parameter $\eta \in [0, 1]$, double occupations are either ignored at $\eta = 1$ or, for $\eta < 1$, the probabilities of states where many of them occur are decreased [8].

Based on this idea, an often used variational wavefunction parameterized by a set of parameters θ, is a product

$$\psi(x|\theta) = P_G(x|\theta)P_J(x|\theta)\,\Phi_0(x|\theta) \tag{3.9}$$

of an antisymmetric part Φ_0 and so called Gutzwiller and Jastrow correlation factors P_G and P_J that encode the correlations induced by interactions. The antisymmetric part ensures the fermionic properties and sets the nodal structure of the wavefunction and can consist of for example a single Slater determinant or the pair-product or *geminal* wavefunction [15]. General Gutzwiller and Jastrow factors are positive many-body correlation functions, and can be written as

$$P_G = \exp\left[\sum_i g_i n_{i\uparrow} n_{i\downarrow}\right] \tag{3.10}$$

and

$$P_J = \exp\left[\frac{1}{2}\sum_{i\neq j} v_{ij}(n_{i\uparrow}-1)(n_{j\downarrow}-1)\right] \tag{3.11}$$

with the variational parameters g_i and v_{ij} [15].

3.4 Gradient Optimization

Using an ansatz for the wavefunction $\psi(x) = \psi(x|\theta)$ that is parameterized by a set of parameters θ and the definition

$$D_k(x) = \frac{1}{\psi(x|\theta)} \frac{\partial}{\partial \theta_k} \psi(x|\theta) = \frac{\partial}{\partial \theta_k} \log \psi(x|\theta) \tag{3.12}$$

the gradient of the energy expectation value with respect to the parameters θ can be written as

$$\begin{aligned}
\frac{\partial}{\partial \theta_k} \langle \hat{H} \rangle &= \frac{\partial}{\partial \theta_k} \frac{\sum_{x,x'} \psi^*(x) H_{xx'} \psi(x')}{\sum_x \psi^*(x)\psi(x)} \\
&= \frac{\partial}{\partial \theta_k} \frac{1}{Z} \sum_{x,x'} \psi^*(x) H_{xx'} \psi(x') \\
&= \frac{1}{Z} \frac{\partial}{\partial \theta_k} \sum_{x,x'} \psi^*(x) H_{xx'} \psi(x') + \sum_{x,x'} \psi^*(x) H_{xx'} \psi(x') \frac{\partial}{\partial \theta_k} \frac{1}{Z} \\
&= \frac{1}{Z} \left[\sum_{x,x'} \psi^*(x) H_{xx'} \psi(x') D_k^*(x) + \sum_{x,x'} \psi^*(x) H_{xx'} \psi(x') D_k(x') \right] - \langle \hat{H} \rangle \frac{1}{Z} \frac{\partial Z}{\partial \theta_k} \\
&= \left[\sum_x p(x) E_{\text{loc}}(x) D_k^*(x) + \sum_{x'} p(x') E_{\text{loc}}^*(x') D_k(x') \right] - \langle \hat{H} \rangle \frac{1}{Z} \frac{\partial Z}{\partial \theta_k} \\
&= \langle\langle E_{\text{loc}} D_k^* \rangle\rangle + \langle\langle E_{\text{loc}}^* D_k \rangle\rangle - \langle\langle E_{\text{loc}} \rangle\rangle \sum_x p(x) \left[D_k^*(x) + D_k(x) \right] \\
&= \langle\langle E_{\text{loc}} D_k^* \rangle\rangle + \langle\langle E_{\text{loc}}^* D_k \rangle\rangle - \langle\langle E_{\text{loc}} \rangle\rangle \langle\langle D_k^* \rangle\rangle - \langle\langle E_{\text{loc}} \rangle\rangle \langle\langle D_k \rangle\rangle \\
&= \langle\langle E_{\text{loc}} D_k^* \rangle\rangle + \langle\langle E_{\text{loc}}^* D_k \rangle\rangle - 2 \langle\langle E_{\text{loc}} \rangle\rangle \langle\langle \text{Re}\{D_k\} \rangle\rangle
\end{aligned} \tag{3.13}$$

If we assume the local energy to be real-valued for all configurations x

$$E_{\text{loc}}(x) = E_{\text{loc}}^*(x) \tag{3.14}$$

the gradient can be written in a simpler form

$$\begin{aligned}
\frac{\partial}{\partial \theta_k} \langle \hat{H} \rangle &= 2\text{Re} \, \langle\langle E_{\text{loc}} D_k \rangle\rangle - 2\text{Re} \left[\langle\langle E_{\text{loc}} \rangle\rangle \langle\langle D_k \rangle\rangle \right] \\
&= \langle\langle 2\text{Re} \left[\left(E_{\text{loc}} - \langle\langle E_{\text{loc}} \rangle\rangle \right) D_k \right] \rangle\rangle \\
&= \langle\langle 2\text{Re} \left[E_{\text{loc}} \left(D_k - \langle\langle D_k \rangle\rangle \right) \right] \rangle\rangle \\
&= \langle\langle G_k \rangle\rangle
\end{aligned} \tag{3.15}$$

and is again a statistical expectation value of the gradient estimator

$$G_k(x) = 2\text{Re} \left[E_{\text{loc}}(x) \left(D_k(x) - \langle\langle D_k \rangle\rangle \right) \right] . \tag{3.16}$$

Even if Eq. (3.14) is not fulfilled, this is often a good enough approximation of the gradient for the optimization procedure to work.

3.4.1 Vanishing Gradient for an Eigenstate of the Hamiltonian

If $|\psi\rangle$ is an eigenstate of $\hat{H}$ with an eigenvalue E, it is easy to see that the gradient is zero

$$\begin{aligned}\frac{\partial}{\partial\theta_k}\langle\hat{H}\rangle &= \frac{1}{\langle\psi|\psi\rangle}\frac{\partial}{\partial\theta_k}\langle\psi|\hat{H}|\psi\rangle + \langle\psi|\hat{H}|\psi\rangle\frac{\partial}{\partial\theta_k}\frac{1}{\langle\psi|\psi\rangle} \\ &= \frac{E}{\langle\psi|\psi\rangle}\frac{\partial}{\partial\theta_k}\langle\psi|\psi\rangle - E\langle\psi|\psi\rangle\frac{1}{\langle\psi|\psi\rangle^2}\frac{\partial}{\partial\theta_k}\langle\psi|\psi\rangle \\ &= \frac{E}{\langle\psi|\psi\rangle}\frac{\partial}{\partial\theta_k}\langle\psi|\psi\rangle - \frac{E}{\langle\psi|\psi\rangle}\frac{\partial}{\partial\theta_k}\langle\psi|\psi\rangle \\ &= 0\,.\end{aligned} \tag{3.17}$$

This is true not only for the groundstate with the lowest eigenvalue, but for any eigenstate. A simple gradient-descent scheme might therefore end up in a local minimum of the energy landscape at an excited state of the system.

3.4.2 The Zero-Variance Principle

The variance of the energy expectation value becomes zero if an exact eigenstate ψ with the energy E is reached

$$\mathrm{Var}(H) = \langle\hat{H}^2\rangle - \langle\hat{H}\rangle^2 = \langle\psi|\hat{H}^2|\psi\rangle - \langle\psi|\hat{H}|\psi\rangle^2 = \langle\psi|\psi\rangle^2E^2 - (\langle\psi|\psi\rangle E)^2 = 0\,. \tag{3.18}$$

The local energy

$$E_{\mathrm{loc}}(x) = \sum_{x'}\langle x|\hat{H}|x'\rangle\frac{\langle x'|\psi\rangle}{\langle x|\psi\rangle} = \frac{\langle x|\hat{H}|\psi\rangle}{\langle x|\psi\rangle} = E \tag{3.19}$$

then becomes independent of x and its variance is also zero. This means that stochastically evaluating the expectation value for an exact eigenstate is not affected by fluctuations of the local energy as it is constant. As a result, also a minimization of the variance instead of the energy should lead to an eigenstate, and this is done as well. Indeed this was sometimes believed to be superior to optimizing the energy expectation value, however, Snajdr et al. [23] have shown that consistently better estimates of non-energy-related properties are achieved with energy-optimized wave functions.

3.4.3 Stochastic Gradient Descent and the Stochastic Reconfiguration Method

To optimize the energy, gradient descent algorithms can be used to iteratively update the set of parameters θ until a convergence criteria is met. In the simplest case of stochastic gradient descent, at each iteration t a number of configuration samples are drawn to evaluate the gradient Eq. (3.16) stochastically and update the parameters

$$\theta_k^{(t+1)} = \theta_k^{(t)} - \eta\,\partial_k\langle\hat{H}\rangle\,. \tag{3.20}$$

The step-size η is a so called *hyperparameter* and also called the learning rate. It can be set to a constant value or be decreased continuously after some time to further improve a converging result. Stochastic gradient descent is also widely used in machine learning applications where the data is split up into smaller randomly chosen batches and the gradients are estimated with only such subsets of the data. This approximates the true gradient with some additional Gaussian noise that can prevent overfitting. Stochastic gradient descent can be linked to Langevin dynamics and there is a lot of theoretical knowledge how it helps with large datasets to allow for parameter uncertainty [28]. If an exact eigenstate is reached the variance goes to zero, effectively turning off the noise term.

Second-order methods like the Newton method improve the convergence behavior, especially if the gradient is much larger in certain directions of the energy hyperplane, at the cost of inverting the Hessian matrix. Closely related is the stochastic reconfiguration (SR) method proposed by Sorella [25] that allows for a more stable convergence of the parameter optimization and enables VMC simulations with an order of a hundred thousand variational parameters [15]. This makes use of the fact that the imaginary-time evolution of a quantum state becomes the exact groundstate in the limit of large times, if the two states are not orthogonal to each other. Some steps of the derivation are shown here to convey the physical motivation for the parameter update equation. The imaginary-time-dependent Schrödinger equation is

$$\frac{\partial}{\partial \tau}|\psi^{(\tau)}\rangle = -\hat{H}|\psi^{(\tau)}\rangle\,. \tag{3.21}$$

Substituting the normalized wavefunction $|\tilde{\psi}^{(\tau)}\rangle = |\psi^{(\tau)}\rangle/\sqrt{\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle}$ into this equation yields

$$\begin{aligned}\frac{\partial}{\partial \tau}|\tilde{\psi}^{(\tau)}\rangle &= \frac{1}{\sqrt{\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle}}\frac{\partial}{\partial \tau}|\psi^{(\tau)}\rangle - \frac{1}{2}\frac{|\psi^{(\tau)}\rangle}{\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle^{\frac{3}{2}}}\frac{\partial}{\partial \tau}\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle \\ &= -\frac{1}{\sqrt{\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle}}\hat{H}|\psi^{(\tau)}\rangle + \frac{\langle\psi^{(\tau)}|\hat{H}|\psi^{(\tau)}\rangle}{\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle}\frac{|\psi^{(\tau)}\rangle}{\sqrt{\langle\psi^{(\tau)}|\psi^{(\tau)}\rangle}} \\ &= -\left(\hat{H} - \langle\hat{H}\rangle\right)|\tilde{\psi}^{(\tau)}\rangle\,.\end{aligned} \tag{3.22}$$

After writing the derivatives in terms of the variational parameters θ_k

$$\frac{\partial}{\partial \tau}|\tilde{\psi}^{(\tau)}\rangle = \sum_k \frac{\partial \theta_k}{\partial \tau}\frac{\partial}{\partial \theta_k}|\tilde{\psi}^{(\tau)}\rangle = -\left(\hat{H} - \langle\hat{H}\rangle\right)|\tilde{\psi}^{(\tau)}\rangle \tag{3.23}$$

the so called time-dependent variational principle (TDVP) is obtained by minimizing the L_2-norm of this equation with respect to the derivatives of the parameters $\dot{\theta}_k = \frac{\partial \theta_k}{\partial \tau}$ [15]. After introducing discrete time steps $\dot{\theta}_k = \frac{\Delta\theta_k}{\Delta\tau}$, a formulation for the updated parameters is reached

$$\begin{aligned}\theta_k^{(t+1)} &= \theta_k^{(t)} + \Delta\theta_k^{(t)} \\ &= \theta_k^{(t)} - \Delta\tau \sum_m S_{km}^{-1} g_m\end{aligned} \tag{3.24}$$

with

$$S_{km} = \mathrm{Re}\langle D_k^* D_m\rangle - \mathrm{Re}\langle D_k\rangle\langle D_m\rangle \tag{3.25}$$

and

$$g_m = \mathrm{Re}\langle HD_m \rangle - \langle H \rangle \mathrm{Re}\langle D_m \rangle \, . \tag{3.26}$$

Having gone trough these steps, the stochastic reconfiguration method approximates the imaginary-time evolution starting from the time-dependent Schrödinger equation. It involves the diagonalization of the covariance matrix S which can be achieved using a Cholesky decomposition or accelerated with the conjugate gradient (CG) method [15]. g_m resembles the gradient estimate in Eq. (3.15) that was derived in terms of the local energy, and $\Delta\tau$ takes the place of the step size or learning rate.

Chapter 4

Neural Network Quantum States

In order to solve strongly correlated quantum many-body problems numerically, compact representations of the quantum states must be found that are able to capture the most essential features while reducing the degrees of freedom down from the exponential complexity of the Hilbert space. This boils down to dimensional reduction and feature extraction, for which artificial neural networks have proven to be a very capable tool in recent years. Machine Learning algorithms and especially artificial neural networks are able to approach complex problems in the realm of image and speech recognition, optimizing hundreds of thousands of parameters. The proposed variational ansatz for the VMC method by Carleo and Troyer [3] merges ideas from the machine learning community with the longstanding experience of statistical physics. In this Chapter, a short overview over some concepts in the field of machine learning is provided first, to help place the ideas coming from the physics domain within the field and terminology of machine learning. Then the particular ansatz as well as some further solutions are discussed.

4.1 Machine Learning Rudiments

4.1.1 Supervised, Unsupervised and Reinforcement Learning

The field of *machine learning* is usually divided into three categories, namely supervised, unsupervised and reinforcement learning. *Learning* in all cases means an optimization of parameterized mathematical functions to improve at a certain task, measured by a cost function.

Problems assigned to supervised learning deal with sets of labeled data $\{(x_i, t_i)\}$ containing feature vectors x and target values t [1]. The task is then either regression, optimizing a parameterized function $y(x)$ to best fit the target values, or classification, assigning probabilities for predefined classes, minimizing a distance function to the true class or distribution t. After an optimization period using the training data, the model is used to predict the targets or classes of new, unlabeled data. Best-in-class results in image recognition problems have been achieved using deep, multi-layered neural networks, especially convolutional neural networks (CNN), due to an increase of computational power in recent years mainly relying on GPUs, and the availability of large datasets.

Unsupervised learning schemes are used on unlabeled data sets for data compression, discovering patterns and features in the data and clustering of similar features [1]. The most basic example is principal component analysis. Another example to mention here, as this model is used in the following sections, is the restricted Boltzmann machine (RBM) which is mostly used for learning probability distributions. The RBM was able to achieve very good results in recommender systems (collecting the so called netflix price). Recent advances were made with generative adversarial networks (GAN) that use a pair of a generative and a discriminative network trying to trick each other and are also able to produce new samples from a learned probability distribution, for example realistically looking human faces of people who don't actually exist.

The third learning scheme is reinforcement learning. In the machine learning terminology this is usually formulated as an agent-environment scenario. Inspired by how a human baby would learn to walk, a so-called agent is placed in an environment in which it receives rewards if it achieves or gets closer to a predefined goal. The *agent*, having a set of rules on how to react to measurements in the *environment*, takes an *action* and gets a *reward* based on its location and the state of the environment. The rewards, positive or negative, are then used to modify its parameterized rule-set. Reinforcement learning schemes have come to fame thanks to Google Deepmind's AlphaZero, an algorithm beating the best human Go player after training only by playing against itself with no expert input.

The variational Monte-Carlo algorithm can be identified as reinforcement learning and placed in this last category [3]. In an attempt to make a vague connection to the agent-environment formulation, the goal of minimizing the energy expectation value provides feedback, a reward, for a change in the parameters of the wavefunction, the rule-set. The environment is the energy surface that is defined by the variational wavefunction and is essentially defined by the *behavior* of the agent. In this case the parameterization of the agent is deeply interwoven with the shape of the environment, so at every step, the actions, being an update of the variational parameters, change not only the agents rule-set, but also the environment.

4.1.2 Feed Forward Neural Networks

To follow the discussion in the following sections, the concept of a feed forward neural network (FFNN) is briefly summarized here. An artificial neuron, also called a perceptron, receives multiple inputs, assigns them an importance according to some parameters, and outputs an activation that usually tries to implement a certain threshold for the signals coming in. The output of a single neuron i

$$y_i(\boldsymbol{x}|W, \boldsymbol{b}) = \sigma\left(\sum_j W_{ij} x_j + b_i\right) \tag{4.1}$$

is the weighted sum over all the inputs x_j and an additional bias b_i, fed into a nonlinear activation function σ. For the activation function, besides $\tanh(x)$ and the sigmoid-function $\frac{1}{1+e^{-x}}$, the rectified linear unit (ReLU) function $\max(x, 0)$ is most often used. Many of these neurons are combined to a layer, and multiple layers can be stacked to form a deep neural network, the final layer being the output of the computational graph, see Fig. 4.1. Intermediate layers are also called hidden layers and the final-layer's softmax activation function usually limits the outputs to $[0, 1]$ in classification applications. An important fact is that the activation function must be nonlinear, as otherwise the whole network would collapse into a linear function and would not gain anything from adding additional layers [1]. The parameters to be optimized, $\{W_{ij}\}$ and $\{b_j\}$ for each layer, are called the *weights and biases*.

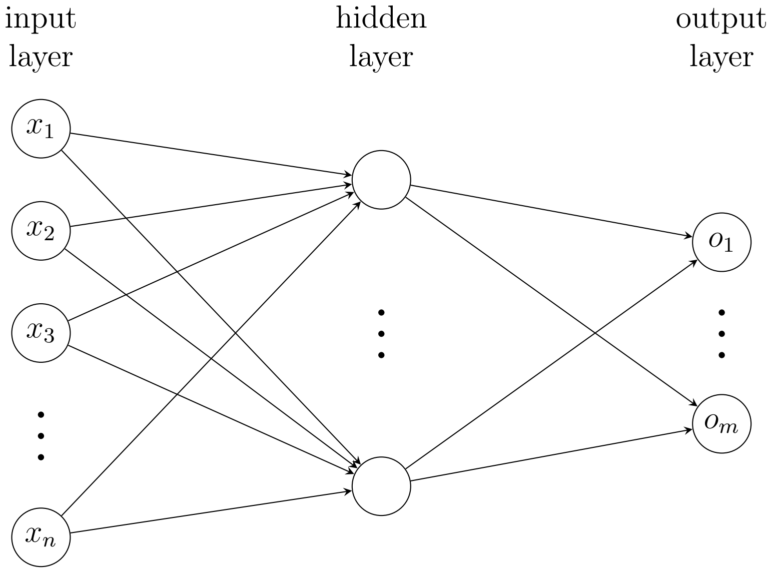


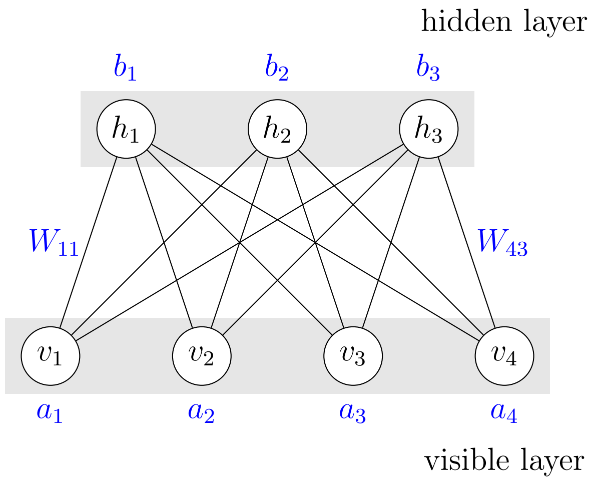
Figure 4.1: Neural network (NN) computational graph with an input layer representing the data vector $\boldsymbol{x}$, hidden intermediate layers and an output layer $\boldsymbol{o}$, for example probabilities for some predefined classes.

Such a network can be trained with gradient descent algorithms to lower a predefined cost function. Using the chain rule to compute the derivatives with respect to the parameters of all the layers is called *backpropagation* of the error. In such a network, the output of one layer becomes the input of the next layer, and the final computation equates to evaluating functions of functions of some input vector.

4.1.3 The Restricted Boltzmann Machine

A Boltzmann machine is a probabilistic model on an undirected graph of binary nodes, that can learn latent representations of data. It can learn arbitrary probability distributions through an unsupervised learning scheme and is then able to compute new data samples from it, however, there are also different ways to use it with supervised learning methods. Boltzmann machines descend from Hopfield networks which are identical to the fully connected Ising model in statistical physics. Each node, also called stochastic neuron, has a binary state and is connected to all other neurons. Boltzmann machines introduce an additional type of nodes, so-called hidden neurons. N visible neurons, written as a vector $\boldsymbol{v}$ represent a data sample or feature vector while M hidden neurons $\boldsymbol{h}$ are used as latent variables to capture the correlations between the visible neurons. Each edge in the graph, representing a connection between two neurons i and j, has a weight W_{ij} with $W_{ii} = 0$. A large positive weight indicates a positive correlation between two units, while a large negative weight means a negative correlation. Each neuron also has an additive bias.

In a *restricted* Boltzmann machine (RBM) there are edges only between the visible and the hidden nodes, but not among them, see Fig. 4.2, which makes the computational effort a lot more manageable.



[ht]

Figure 4.2: Undirected graph of a restricted Boltzmann machine (RBM) with $N = 4$ visible nodes v_i representing the data and $M = 3$ hidden nodes h_j encoding correlations.

It can be thought of as consisting of a visible and a hidden layer with connections between the layers. An RBM can also be seen as a special case of a Markov random field [1]. An Ising-type energy is defined as

$$E(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{a}, \boldsymbol{b}, W) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} W_{ij} v_i h_j \,. \tag{4.2}$$

with the bias parameters a_i and b_j for the visible and hidden neurons respectively. A symbol $\theta = (\boldsymbol{a}, \boldsymbol{b}, W)$ will be used from now on as a shorthand notation for all the variational parameters. The probabilistic nature of the model comes from the definition of the probability for a specific configuration

$$p(\boldsymbol{v}, \boldsymbol{h}|\theta) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h}|\theta)} \tag{4.3}$$

according to the Boltzmann distribution, which lends the model its name. The partition function Z is usually intractable to compute, but conditional probabilities can be used that don't require its computation. From this the probability of a single neuron having the value 1

$$p(v_i = 1|v_1, \ldots v_{i-1}, v_{i+1}, \ldots v_N, \boldsymbol{h}, \theta) = \frac{1}{1 + e^{-\Delta E_i}} = \sigma\left(\sum_{j \neq i} W_{ij} h_j + b_i\right) \tag{4.4}$$

can be derived, with $\Delta E_i = E(v_i = 1) - E(v_i = -1)$ [1]. This can be interpreted as the firing rate of the neuron. The right-hand side of this expression is equal to the deterministic activation $\text{sig}(y_i)$ of a general perceptron-type neuron, but in case of RBM the output is of stochastic nature. For a fixed set of parameters, Markov-chain Monte-Carlo sampling of the neurons activations is performed until thermal equilibrium is reached, to obtain a new sample from the distribution that was learned from data during training [1].

An exemplary use-case would be repairing a corrupt image by setting most of the visible units to the known valid pixel-values and sampling the missing pixels, obtaining a very likely image according to the learned distribution. The same procedure can be used in a recommender-system of a movie-streaming platform to guess a very likely personalized rating of a movie that a user has not yet watched, based on his ratings of other movies and the correlations of ratings for different movies learned from a large data base. An RBM is a *universal approximator*, meaning that it can approximately represent any probability distribution to arbitrary accuracy given a sufficient number of hidden units [12]. Training an RBM with methods like contrastive divergence is quite different than training networks on directional graphs (feed forward neural networks), but will not be elaborated here because in the application of RBM as a variational wavefunction a different learning scheme is used.

4.2 Neural Network Quantum States (NQS)

A general variational ansatz for the wavefunction is

$$|\psi_\theta\rangle = \sum_x \psi(x|\theta)\,|x\rangle\,, \tag{4.5}$$

where x is short for all the discrete-valued degrees of freedom in the computational basis spanning the Hilbert space $\mathcal{H}$. This could be for example the $\{\sigma_i^z\}$ of a spin-chain or all the occupation numbers $\{n_{i\sigma}\}$ of orbitals or lattice sites in an electron system. The weights $\psi(x|\theta)$ are given by a function that is parameterized by the set of parameters θ. This can be an arbitrary function

$$\begin{aligned} \psi\colon \mathcal{H} &\mapsto \mathbb{C} \\ x &\to \psi(x|\theta) \end{aligned} \tag{4.6}$$

that takes as the input a configuration x from the N-dimensional Hilbert space in the computational basis and maps it to a complex number to represent the amplitude and phase of the wavefunction ψ. In order to be used as a variational wavefunction, $\psi(x|\theta)$ and its partial derivatives with respect to the parameters must be efficiently computable.

In the initial work on this method, Carleo and Troyer [3] proposed to view ψ as a computational black box and apply an artificial neural network to the problem, terming this ansatz *neural network quantum state* (NQS). Using a restricted Boltzmann machine in a reinforcement learning scheme, they were able to achieve competitive results for finding the groundstate energy of the Heisenberg model. An RBM seems to be an efficient representation of quantum states meaning that the necessary number of parameters increases at most polynomially with the dimensions of the system [10]. With NQS it is also possible to incorporate intrinsic symmetries of the model either by design or by sharing parameters between nodes [3].

The basic goal for finding a good variational form is to be able to extract the most relevant features in the correlations and represent them efficiently. Some research has already been done investigating different neural network architectures as trial wavefunctions for VMC calculations and some of it will be mentioned in the next sections. In addition to finding the groundstate energy and a representation for the wavefunction, dynamical properties [3] as well as investigating excited states [5] were also shown to be in the reach of NQS.

4.2.1 Restricted Boltzmann Machines as Variational Wavefunctions

Because of its origins in statistical physics and its proven applicability to learning probability distributions, the restricted Boltzmann machine is a natural candidate for a trial neural network quantum state. The wavefunction ansatz proposed by [3] is the expression for the probability of a configuration of visible nodes $\boldsymbol{v}$ without the normalization

$$\psi(\boldsymbol{v}|\theta) = \sum_{\{\boldsymbol{h}\}} e^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)} \,. \tag{4.7}$$

In physical terms this represents the partition function of a gas of M hidden units connected to the N physical spins [3]. The summation over the hidden units $h_j = \pm 1$ can be carried out to obtain

$$\begin{aligned} \psi(\boldsymbol{v}|\theta) &= \sum_{\{\boldsymbol{h}\}} e^{\sum_i a_i v_i + \sum_j b_j h_j + \sum_{i,j} W_{ij} v_i h_j} \\ &= e^{\sum_i a_i v_i} \sum_{\{\boldsymbol{h}\}} \prod_j e^{\left(b_j + \sum_i W_{ij} v_i\right) h_j} \\ &= e^{\sum_i a_i v_i} \prod_j \left(e^{b_j + \sum_i W_{ij} v_i} + e^{-b_j - \sum_i W_{ij} v_i}\right) \\ &= e^{\sum_{i=1}^N a_i v_i} \prod_{j=1}^{M} 2\cosh\left(b_j + \sum_{i=1}^{N} W_{ij} v_i\right) \\ &= e^{\Phi(\boldsymbol{v}|\theta)} \end{aligned} \tag{4.8}$$

with

$$\Phi(\boldsymbol{v}|\theta) = \sum_{i=1}^{N} a_i v_i + \sum_{j=1}^{M} \log\left[2\cosh\left(b_j + \sum_{i=1}^{N} W_{ij} v_i\right)\right] \,. \tag{4.9}$$

To represent the complex-valued and negative phases of a quantum mechanical wavefunction, the parameters (W, a, b) are now generalized to complex numbers. This expression can be computed fairly easily and can take on arbitrary complex values. Furthermore, the complexity of the model, determined by the number of latent variables M, can be systematically increased by setting a larger fraction $\alpha = \frac{M}{N}$ of hidden nodes, and the network can be shown to be capable of approximating representations of any desired probability distribution to arbitrary accuracy [14]. In the general case there are $\alpha N^2 + \alpha N + N$ variational parameters for the weights and hidden and visible biases. A transformation-invariant restricted Boltzmann machine has been proposed [24] based on which a translation symmetric model can be formulated. By introducing sums in Equation (4.8) over all the permutations of a certain feature following a symmetry operation, the effective number of parameters can be reduced to αN [3]. Finally, the partial derivatives of the logarithmic wavefunctions $D_k = \frac{\partial}{\partial \theta_k} \ln \psi(\boldsymbol{v}|\theta)$, that were defined in Chapter 3 and are necessary for the VMC algorithm, are

$$\begin{aligned} D_{a_i} &= v_i \\ D_{b_j} &= \tanh(y_j) \\ D_{W_{ij}} &= \tanh(y_j)\, v_i \,, \end{aligned} \tag{4.10}$$

with the expressions simplified using the definition $y_j = b_j + \sum_i W_{ij} v_i$.

4.2.2 Representational Power of Restricted Boltzmann Machines as Quantum States

Some efforts have been made to bridge the fields of deep learning and theoretical physics to be able to estimate the representational power of RBM and compare it to better understood theoretical concepts for solving quantum problems. Tensor networks (TN) and RBM are both universal approximators, but given a limited number of parameters, they define two independent yet possibly overlapping sets of functions. Chen et al. [4] have shown theoretical connections of tensor network states (TNS) and RBM and have found that RBM can generally encode quantum many-body states with fewer parameters. In theoretical frameworks such as TN the entanglement entropy is a measure for the efficiency of a many-body wavefunction representation. States fulfilling the area law, where the maximum entanglement between two subspaces is determined by the surface area of the boundary between them, are well described by tensor network states [14]. RBM however were shown to be capable of encoding quantum states that require volume law entanglement, hence they may be able to describe states that are out of reach for MPS [4].

RBM could be capable of encoding ground states of systems with long-range interactions or excited states of local Hamiltonians, both of which require volume law entanglement. In some higher dimensional models area law entanglement would suffice but tensor networks cannot be contracted efficiently, whereas RBM might be applied with no extended effort [14]. Since in an RBM the correlations are modeled by all hidden units, they are intrinsically non-local and therefore not dependent on the dimensionality of the model [3].

Reducing the number of parameters by restricting interlayer connections or introducing parameter sharing, similar to how a convolutional neural network (CNN) works, may increase the efficiency for problems suitable to area-law entanglement [14]. On the other hand, an increase in the complexity can be achieved by stacking RBM on top of each other, creating what is called a deep RBM (DBM). Computational effort generally increases with the model complexity but DBM have been shown to provide more efficient representations of certain many-body wave functions. In a DBM the visible units can principally have correlations over a longer range with the same number of parameters, since the deep hidden units encode correlations between the hidden units, which themselves correlate the visible units, giving DBM exponentially better efficiency compared to RBM [10].

NQS results applying an RBM to finding the groundstates of diatomic molecules systematically improved results obtained by conventional methods and were able to recover almost all of the correlation energy, capturing correlations beyond double excitations, which the other methods could not [6].

4.2.3 Feed Forward Neural Network Quantum States

The focus in the first stage of research on neural network quantum states was often put on the RBM. There are however already signs that other architectures might be more suitable for a lot of models. For different machine learning problems, different architectures have turned out to be favorable. Similarly, for the various Hamiltonians describing physical system, from bosonic spin models to high-dimensional fermionic problems, there will probably not be a single best ansatz.

Convolutional neural networks (CNN), which are the go-to method for image-recognition tasks, apply convolutions with kernel matrices (also called filters) to the input data and hence incorporate translational invariance by design. The output of a such a convolution, called a feature map, is taken as one input channel for the next layer, where other filters are used. An intermediate step of a further reduction of the feature size and an activation function is usually applied. A number of these filters per layer constitute the parameters to be optimized, and usually tens of layers are stacked on top of each other. When compared to a fully connected network (Fig. 4.1) this means between layers some connections are missing, and at the same time some weights are shared between different connections. When applied to image recognition problems, it can be visualized how the first layers learn to represent very coarse patterns and colors, while the higher layers combine the lower level features to achieve more and more complicated attributes [31]. A pre-training on large image data sets can be used, such that the filters take on meaningful shapes for a certain class of images, before the final parameter optimization for a specific classification task is performed [31]. These concepts could all apply just as well for learning and representing correlations in physical systems, as solving these problems also involves discovering patterns in configurations of discrete (quantum mechanical) degrees of freedom.

This re-use of information due to the parameter sharing in the computational graph cannot be achieved with conventional tensor networks. An equivalent TN would have to introduce a lot of duplicate external indices, whose data it cannot simply copy, to achieve the same computation of a CNN's convolutions of the input with smaller kernels [13]. Compared to an RBM it has also been proven that a CNN can encode volume-law entanglement polynomially more efficiently in 2D systems, wherefore Levine et al. [13] conclude to focus on looking at state-of-the-art deep learning principles for highly entangled many-body systems.

Some research on applying NQS using feed forward neural networks and in particular CNN include Yang et al. [30] applying a deep learning architecture to compute the ground state of a one-dimensional spin chain, Choo et al. [7] studying the 2D frustrated $J1 - J2$ model and also making an effort to enforce symmetries, and an applications to quantum chemistry by Schütt et al. [22].

4.2.4 Reinforcement Learning of a NQS using Variational Monte-Carlo

After all these theoretical considerations, an implementation of optimizing an NQS is now outlined in this section, and applied to an actual model in the following. To set up a VMC calculation for an NQS ansatz, applying the reinforcement learning scheme described in Chapter 3, the following have to be defined first:

- The graph and Hilbert space, specifying the local states (f.e. $\sigma^z \in \{-\frac{1}{2}, \frac{1}{2}\}$).
- The Hamilton operator's action to find the matrix elements when acting on a state. The matrix does not need to be stored.
- The variational wavefunction $\psi_\theta(x)$. For an RBM, the fraction of hidden neurons $\alpha = M/N$ must be defined and the parameters initialized.

- The optimization method, whether to use simple gradient descent or the stochastic reconfiguration method (see section 3.4.3). A fixed or adaptive learning-rate optimizer can be set.
- The sampler, using hopping or exchange updates. Parallel tempering may be used to reach a larger portion of the Hilbert space.

The optimization procedure using the stochastic reconfiguration method is summarized in Algorithm 1. The calculations of local energy and gradient are derived in Chapter 3 in Eq. (3.5) and Eq. (3.15). The statistical expectation values $\langle\langle . \rangle\rangle$ at each step are over the Monte-Carlo samples drawn according to $|\psi_\theta|^2$ defined by the current set of parameters θ. To increase efficiency, look-up tables can be set up at each iteration to store objects that are used multiple times.

Algorithm 1 Reinforcement Learning of a NQS using variational Monte-Carlo

```
Initialize parameters $\theta$, let $n = 0$, let $\eta = 0.02$
repeat
  $n \leftarrow n + 1$
  Draw $N_S$ samples $\mathcal{S} = \{x\}$ via MCMC according to $p(x) = |\psi_\theta(x)|^2$
  for all $x \in \mathcal{S}$ do
    Find $\{x'\}$ for which $H_{xx'} \leftarrow \langle x'|H|x\rangle \neq 0$
    $E_{\text{loc}}(x) \leftarrow \sum_{x'} H_{xx'} \frac{\psi_\theta(x')}{\psi_\theta(x)}$
    Compute derivatives $\{D_k(x)\}$
  end for
  $\partial_k\langle H\rangle \leftarrow 2\text{Re}\,\langle\langle E_{\text{loc}} D_k\rangle\rangle - 2\text{Re}\,\langle\langle E_{\text{loc}}\rangle\rangle\langle\langle D_k\rangle\rangle$
  $S_{km} \leftarrow \text{Re}\langle\langle D_k^* D_m\rangle\rangle - \text{Re}\langle\langle D_k\rangle\rangle\langle\langle D_m\rangle\rangle$
  $\theta_k \leftarrow \theta_k - \eta \sum_m S_{km}^{-1}\, \partial_m\langle H\rangle$
  $E_n \leftarrow \langle\langle E_{\text{loc}}\rangle\rangle$
until $E_n$ converges
```

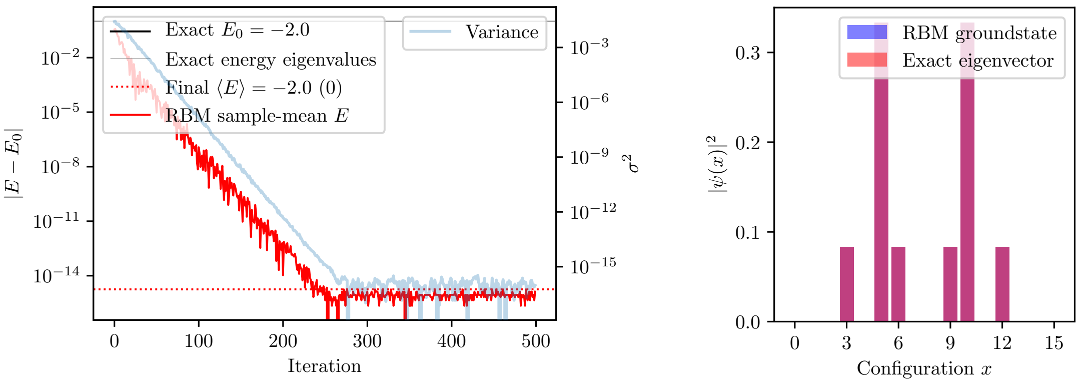
4.2.5 Solving the Heisenberg Model using NQS

To demonstrate NQS, the initial results on the Heisenberg model are reproduced using NetKet [2], an open-source library that implements - for spin systems - the variational Monte-Carlo optimization and different variational functions. In particular, in this section the groundstate of the antiferromagnetic Heisenberg model Eq. (2.21) on a rectangular lattice with $J = 1$ and periodic boundary conditions is investigated. The visible nodes of the RBM are mapped to the s^z values of a spin-1/2 lattice with L sites. Indices for configurations are calculated by interpreting the vectors of zeros and ones, representing spin-down and spin-up respectively, as the binary representation of integers

$$\begin{aligned} &(0,\dots,0,0,0,1) \to 1 \\ &(0,\dots,0,0,1,0) \to 2 \\ &(0,\dots,0,0,1,1) \to 3 \\ &\qquad\qquad\quad \dots \\ &(\underbrace{1,\dots,1,1,1,1}_{L}) \to 2^L - 1\,. \end{aligned} \tag{4.11}$$

An observable can be defined as the sum of local operators acting on specific lattice sites. The two-site operators $S_i^z S_j^z$ and $S_i^\pm S_j^\mp$ in the Heisenberg Hamiltonian Equation (2.23) are created as Kronecker products of the operators in Eq. (2.24) and assigned to pairs of site indices (i, j). Total spin S^z is set to be conserved at the sampling step, restricting the size of the Hilbert space. This way, the network doesn't learn anything about configurations x' where S^z is not conserved, therefore $\psi(x')$ has an arbitrary value.

As a first result, the groundstate of a one-dimensional spin chain with $L = 4$ sites is computed with a translation-symmetric RBM. For better optimization performance, the Marshal sign rule is used, which in the Hamiltonian counts the exchange energy negative but results in the same eigenvalues. The learning progression and the final probability distribution of the configurations is shown in Fig. 4.3. In this *learning curve* also the variance of the local energy E_{loc} calculated from the sampled configurations at each iteration is plotted on the right axis. This can act as an indicator how well the current state corresponds to an eigenstate, as described in section 3.4.2. A sample size of 1000 was used for these calculations, the fraction of hidden units is set to $\alpha = 4$ and the learning rate η to a fixed value of 0.04. This hyperparameter is to be set to a meaningful value, where convergence is fast, but fluctuations are not too strong. Changing the coupling constant in the Hamiltonian scales the update-step in a similar way as changing the learning rate (compare Eq. (3.15)), for example leaving out the factors $\frac{1}{2}$ in the S-operators in Equation (2.21) while also setting $\eta = 0.01$ results in the exact same convergence behavior. In Fig. 4.3a the diminishing variance as the energy approaches the groundstate can be seen. The groundstate eigenvalue and its corresponding eigenvector are retrieved to numerical precision compared to the exact result obtained with the Lanczos algorithm.

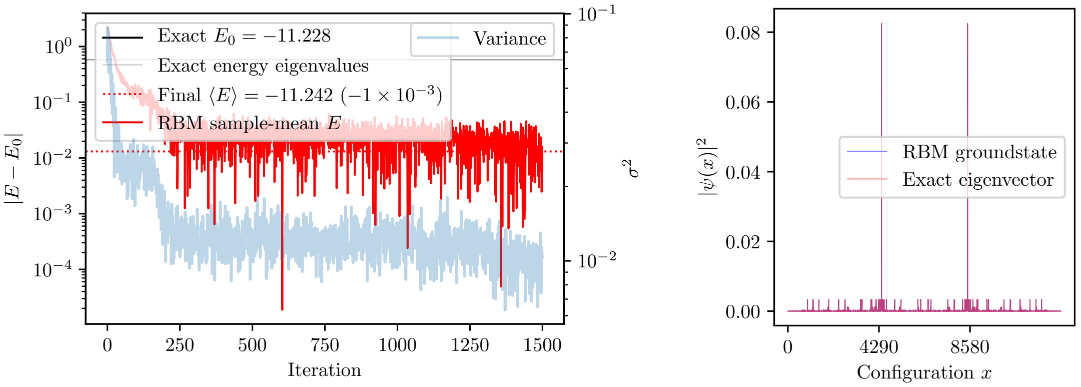


(a)

(b)

Figure 4.3: Heisenberg model of a 4 × 1 spin chain. (a) Groundstate energy learning progression and (b) final probability density using an RBM ansatz compared to exact results. Relative error in parenthesis up to numerical precision.

The same results for the two-dimensional 4 × 4 Heisenberg model are displayed in Fig. 4.4. There remains a relative error 10^{-3} and in (b) one has to look very closely to see some over- or are underrepresented configurations and the overlap with the exact result is $\langle\psi_0|\ \psi_{\text{RBM}}\rangle = 0.984$. In the one-dimensional case the translation-invariant RBM reached numerical accuracy in a quarter of the steps needed for the full RBM, in the two-dimensional system it is able to achieve a lower energy, see Fig. 4.5. This implementation does not improve optimization times however, as the shared parameters are copied to form the full RBM and the inversion of the covariance matrix S is of the same computational effort. A simple convolutional neural network can also be applied to the task and performs similarly to the symmetric RBM.



(a)

(b)

Figure 4.4: Heisenberg model of a 4 × 4 spin lattice. (a) Groundstate energy learning progression and (b) final probability density of the allowed states using an RBM ansatz compared to exact results. Relative error in parenthesis.

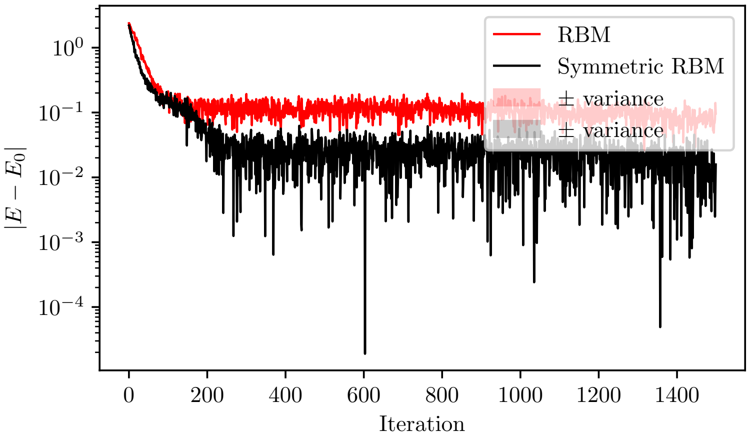


Figure 4.5: Optimization of the 4×4 Heisenberg model using the translation-symmetric RBM with 69 effective parameters compared to the full RBM with 1104.

Finally, the groundstate energy of an 8×8 lattice is calculated for different α. The deviations of the obtained energies from known results [19] are plotted in Fig. 4.6, a result that matches the published results for a similarly sized system [3]. For increasing model complexity, the relative error becomes smaller as the groundstate is approximated with better accuracy. The obtained variances cannot really be used as an uncertainty estimate because they represent simply the statistical deviations over the noisy Monte-Carlo samples at that particular iteration with a fixed set of parameters. In that state, the network represents a quantum state that must not necessarily be the groundstate.

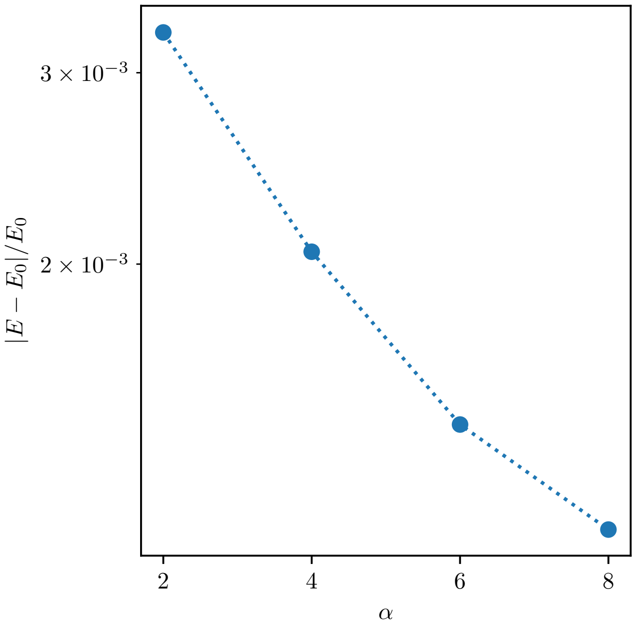


Figure 4.6: Results for the groundstate energy of a 8 × 8 Heisenberg model using an RBM compared to known results [19] with increasing network size determined by the fraction of hidden units α.

Chapter 5

Solving the Fermionic Hubbard Model with Neural Network Quantum States

Solving the Hubbard model with NQS has so far been attempted by Nomura et al. [16] using an ansatz similar to the Jastrow-Slater variational form mentioned in section 3.3, by replacing the Jastrow and Gutzwiller factors with an RBM to encode the correlations. With help of the RBM their program could find better groundstate energies compared to conventional correlation functions in the variational ansatz. In this Chapter it is attempted to apply neural network quantum states to the whole fermionic problem, looking at the groundstate of the Fermi-Hubbard model at half-filling on different one and two-dimensional rectangular lattices. Also the solution by [16] is included introducing a Fermi-sea reference state. While the authors introduced the RBM into a conventional VMC code, the implementation in this thesis is based on NetKet [2], and consists of extensions to the C++ library to allow for fermionic degrees of freedom, reference states and correlation functions as well as Python classes for the solution of the tight binding model and for setting up the graphs and simulations. After the necessary mapping to spin degrees of freedom and some technical aspects of implementing fermionic NQS are explained, some selected results are presented and discussed to finally state a few points of improvements and to give an outlook on further research.

5.1 Implementation of the Hubbard Model for NQS

5.1.1 Mapping Fermionic Occupation Numbers to Spins

In order to encode the fermionic degrees of freedom, the occupation numbers can be mapped to a model of interacting spins. This can be formally done using the Jordan-Wigner Mapping, which maps the fermionic creation and annihilation operators

$$\begin{aligned} \hat{c}_j &\to \left(\prod_{i=0}^{j-1} \hat{\sigma}_i^z\right) \hat{\sigma}_j^- \\ \hat{c}_j^\dagger &\to \left(\prod_{i=0}^{j-1} \hat{\sigma}_i^z\right) \hat{\sigma}_j^+ \end{aligned} \tag{5.1}$$

to the Pauli matrices $\hat{\sigma}$ [6]. Since a fermionic quantum state can be occupied only once, this results in a a one-to-one correspondance to a spin-half chain. Another maybe more illustrative way is to identify the encoding of a spin model, for example $|\downarrow,\downarrow,\uparrow\rangle \rightarrow (0,0,1)$, as the occupation numbers and implement the operators to account for the sign change. In machine learning this representation of data with only binary values is called *one-hot* encoding, and is often better suited as an input to neural networks compared to using fewer channels with values from a larger set.

To use NetKet's representation of spin lattices, in the following discussions the occupation numbers of L sites are mapped to $2L$ spins σ_i with the local values -1 and 1, which directly correspond to the N visible nodes v_i in an NQS ansatz, following the convention

$$\begin{aligned} v_i &= 2\,n_{i\uparrow} - 1 && \text{for} \quad i = 1 \ldots L\,, \\ v_i &= 2\,n_{(i-L)\downarrow} - 1 && \text{for} \quad i = L+1, \ldots 2L\,. \end{aligned} \tag{5.2}$$

For a two-dimensional lattice, the following 2×2 example demonstrates how the sites are indexed:

$$\begin{array}{c} \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} \\ \begin{array}{|c|c|}\hline 5 & 6 \\ \hline 7 & 8 \\ \hline \end{array} \end{array} \rightarrow \begin{array}{c} \begin{array}{|c|c|}\hline n_{1\uparrow} & n_{2\uparrow} \\ \hline n_{3\uparrow} & n_{4\uparrow} \\ \hline \end{array} \\ \begin{array}{|c|c|}\hline n_{1\downarrow} & n_{2\downarrow} \\ \hline n_{3\downarrow} & n_{4\downarrow} \\ \hline \end{array} \end{array}$$

With a fixed particle number and fixed total spin there are now

$$\prod_{\sigma=\uparrow,\downarrow} \frac{L!}{(L-N_\sigma)!N_\sigma!} \tag{5.3}$$

possible configurations.

A custom graph to define the physical lattice is built with edges of different *colors* between the visible nodes of the neural network ansatz. These colors represent the different interactions that are defined in an operator acting on the lattice sites. Three different colors, or interaction indices, are used for the Hubbard Hamiltonian:

- 1: Hubbard terms, connecting nodes representing spin-up and spin-down occupations of the same site.
- 2: Hopping terms inside the lattice, connecting nearest neighbors.
- 3: Hopping terms crossing a boundary, needed for implementing periodic and antiperiodic boundary conditions.

Periodic or Antiperiodic boundary conditions are implemented for the spin-up sites and spin-down sites independently. By encoding the types of interactions in the graph object, the implementation of operators is completely independent of the lattice's geometry.

5.1.2 Fermionic Hubbard Hamiltonian Implementation

When simulating systems of electrons, the anticommutation relations

$$\begin{aligned} \{\hat{c}^\dagger_{i\sigma}, \hat{c}_{j\sigma'}\} &= \hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma'} + \hat{c}_{j\sigma'}\hat{c}^\dagger_{i\sigma} = \delta_{ij}\delta_{\sigma\sigma'} \\ \{\hat{c}_{i\sigma}, \hat{c}_{j\sigma'}\} &= \{\hat{c}^\dagger_{i\sigma}, \hat{c}^\dagger_{j\sigma'}\} = 0 \end{aligned} \tag{5.4}$$

must be taken into account. The Hubbard Hamiltonian defined in Eq. (2.6) acting on a state evolves it if hopping is possible and counts double occupations. As an example, given a particular configuration in a 3-site system

$$|x_1\rangle = |\uparrow,\downarrow,\downarrow\rangle = \hat{c}^\dagger_{1\uparrow}\hat{c}^\dagger_{2\downarrow}\hat{c}^\dagger_{3\downarrow}|0\rangle \tag{5.5}$$

one possible hopping event would be for the electron on site 3 to hop through the periodic boundaries to site 1, connecting to the configuration $|x_2\rangle = |\uparrow\downarrow,\downarrow,0\rangle$. To calculate the matrix element

$$\begin{aligned}\langle x_2|\hat{H}|x_1\rangle &= \langle x_2|(-t\,\hat{c}^\dagger_{1\downarrow}\underbrace{\hat{c}_{3\downarrow})\,\hat{c}^\dagger_{1\uparrow}}_{-\hat{c}^\dagger_{1\uparrow}\hat{c}_{3\downarrow}}\hat{c}^\dagger_{2\downarrow}\hat{c}^\dagger_{3\downarrow}|0\rangle \\ &= +t\,\langle x_2|\hat{c}^\dagger_{1\downarrow}\hat{c}^\dagger_{1\uparrow}\underbrace{\hat{c}_{3\downarrow}\hat{c}^\dagger_{2\downarrow}}_{-\hat{c}^\dagger_{2\downarrow}\hat{c}_{3\downarrow}}\hat{c}^\dagger_{3\downarrow}|0\rangle \\ &= -t\,\langle x_2|\underbrace{\hat{c}^\dagger_{1\downarrow}\hat{c}^\dagger_{1\uparrow}}_{-\hat{c}^\dagger_{1\uparrow}\hat{c}^\dagger_{1\downarrow}}\hat{c}^\dagger_{2\downarrow}\underbrace{\hat{c}_{3\downarrow}\hat{c}^\dagger_{3\downarrow}}_{1-\hat{c}^\dagger_{3\downarrow}\hat{c}_{3\downarrow}}|0\rangle \\ &= +t\,\langle x_2|x_2\rangle\end{aligned} \tag{5.6}$$

the anticommutation relations are used to exchange the operators, each time catching a minus sign, until $\hat{c}_{3\downarrow}$ is moved all the way to the right, where it acts on the vacuum states resulting in a zero $\hat{c}_{3\downarrow}|0\rangle = 0$. After this, the operators have to be brought back into the correct order, so that all spin-up operators come first, which in this case introduces another negative sign. All the operators with an index lower than both the hopping electron's final and start index contribute exactly twice, once when $\hat{c}_{3\downarrow}$ is brought to the right, and once when the correct ordering of $\hat{c}^\dagger_{3\downarrow}$ is restored. Therefore for a hopping from site index i to j only the occupations of sites after the start index i and before the final index j end up determining the sign. A factor of $(-1)^n$ is obtain by n occupied sites between them. In this example there was just one such occupation created by $\hat{c}^\dagger_{2\downarrow}$, resulting in $H_{x_2x_1} = +t$.

The implementation of the Hubbard Hamiltonian needs to find the connected states $\{x'\}$ for a given configuration x of the visible units $\boldsymbol{v}$. Looping over all edges in the graph dedicated to hopping interactions, a vector of possible new configuration is returned together with the numerical values of the matrix elements ($\pm t$). For the Coulomb energy, the double occupations are counted, looping over the dedicated edges connecting corresponding spin-up and spin-down nodes. If antiperiodic boundary conditions are chosen, the hopping terms across a boundary get another factor of -1.

5.1.3 Order Convention of the Creation Operators

Changing the order of the fermionic operators in the mapping leads to a different sign structure and therefore a different behavior in reaching the ground state. The two usual conventions are

$$|\psi_n\rangle = \left(\prod_{i=1}^{L}\left(\hat{c}^\dagger_{i,\uparrow}\right)^{n_{i,\uparrow}}\right)\left(\prod_{j=1}^{L}\left(\hat{c}^\dagger_{j,\downarrow}\right)^{n_{j,\downarrow}}\right)|0\rangle \tag{5.7}$$

and

$$|\psi_n\rangle = \prod_{i=1}^{L} \left(\hat{c}^{\dagger}_{i,\uparrow}\right)^{n_{i,\uparrow}} \left(\hat{c}^{\dagger}_{i,\downarrow}\right)^{n_{i,\downarrow}} |0\rangle \tag{5.8}$$

where $n_{i,\sigma} \in \{0,1\}$ is the occupation number of the site i with spin σ, and the products are written in ascending order. The first line corresponds to the convention introduced in Equation (5.2). One advantage of this ordering can be that nearest-neighbor hopping events within the same row of a 2-D lattice don't affect the sign structure, as they transfer to neighboring indices as well. An advantage of ordering the sites first and then the spins, as in Eq. (5.8), is the locality of same-site occupations in the feature vectors, which could be taken advantage of by a network performing local operations like a CNN. The first ordering convention $|n_{1_\uparrow}, n_{2_\uparrow}, \dots n_{L_\uparrow}, n_{1_\downarrow}, n_{2_\downarrow}, \dots n_{L_\downarrow}\rangle$ is necessary for the projection of the Fermi-sea state in the form that was derived in Eq. (2.20), therefore this is what was used in the implementation. The difference this makes to the sign structure has some effects on the optimization process as mentioned. As an example, for a 4×4 tight binding model the best achievable error of the groundstate energy was twice as high with the second convention, which was initially implemented.

The updates of configurations during the Monte-Carlo sampling need to conserve the number of particles and the total spin, hence conserve the number of occupations on the two sub-lattices for spin-up and spin-down. The implementation of this is dependent on the order convention for the mapping of occupation numbers to spin-indices. Using hopping-updates, applying the Hamiltonian to a configuration, automatically produces states in the correct subspace. Exchange updates, to affect larger regions of the lattice at a single step, have to be restricted to the sub-lattices of equal spins.

5.2 Approaching the Fermi-Hubbard Groundstate using a Pure RBM Ansatz

In the following sections different results of computations for the tight binding and Hubbard models at half-filling are discussed. Unless noted differently, the fraction of hidden units α is set to 4 and a sample size of 1000 is used.

5.2.1 Solving the Tight Binding Model with an RBM Ansatz

Before solving the Hubbard model, the tight binding model of uncorrelated electrons for which an exact solution is known and described in section 2.3, is investigated. This has the benefit of being able to view analytically calculated excited states, which are possible local minima, during the optimization. The most basic setup is comprised of two sites and two electrons with opposite spin, resulting in four possible configurations $|0, \uparrow\downarrow\rangle$, $|\downarrow, \uparrow\rangle$, $|\uparrow, \downarrow\rangle$ and $|0, \uparrow\downarrow\rangle$. With $U = 0$ the movement of the electrons is not limited, therefore all of these states should occur with equal probability. 100 samples were used for each iteration. Using a fairly large learning rate $\eta = 0.5$ was possible, speeding up convergence, to reach numerical precision and learn the correct uniform distribution $|\psi(x)|^2 = 0.25$. The learning progression is shown in Fig. 5.1.

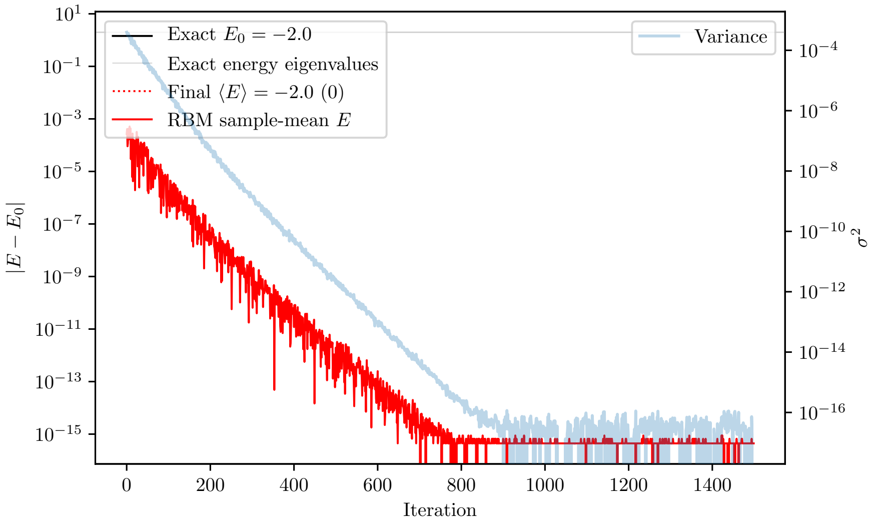


Figure 5.1: Minimal 2-site tight binding model. Optimization progress of finding the groundstate using an RBM.

For more complex systems a smaller η must be chosen, as otherwise the system frequently makes big jumps to higher energies. In the following calculations values between 0.01 and 0.05 proved to work best. The groundstate of two-dimensional lattices is investigated next. An RBM trial wavefunction on a 2×2 lattice reaches the degenerate exact groundstate energy up to a relative error in order of 10^{-5} after about 100 iterations and doesn't improve much further, as depicted in Fig. 5.2. As a rough orientation, running this 2×2 model for 1000 iterations with a sample size of 1000 and $\alpha = 4$ took about 90 seconds on a single core of a 2018 notebook computer. The program is well parallelizable and takes only 28 seconds when running on 4 cores, even though thermal throttling of the CPU happens a lot quicker when more cores are active.

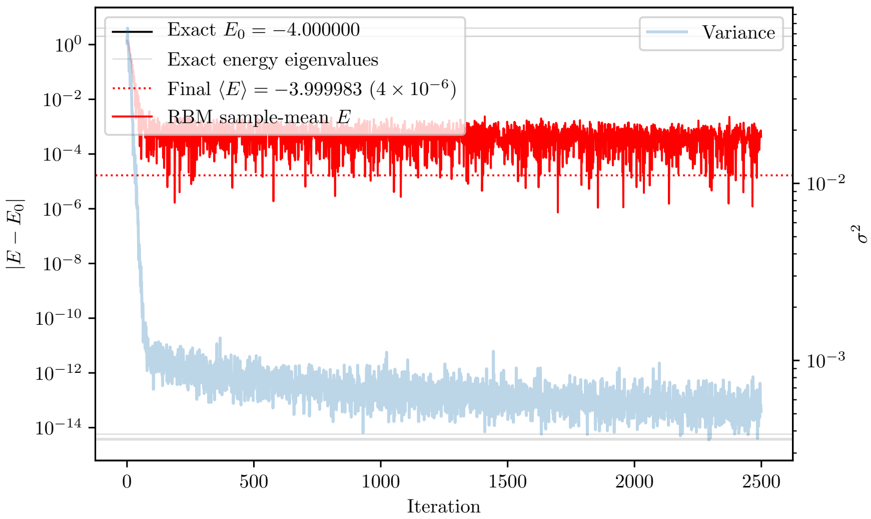


Figure 5.2: 2×2 tight binding model. Optimization progress of finding the groundstate using an RBM.

Finally, the groundstate of a 4 × 4 model is attempted with an RBM state. In Figure 5.3 it becomes clear, that this ansatz has some trouble navigating through the energy surface, and eventually seems to be stuck at no exact eigenvalue with a relative error of 6%. Using parrallel tempering, where independent samplers with different acceptance criteria for new configurations are used and whole configurations are swapped from time to time, did not yield a significantly better result. Neither did the use of a momentum optimization scheme, where at each step a fraction of the previous gradient is taken along.

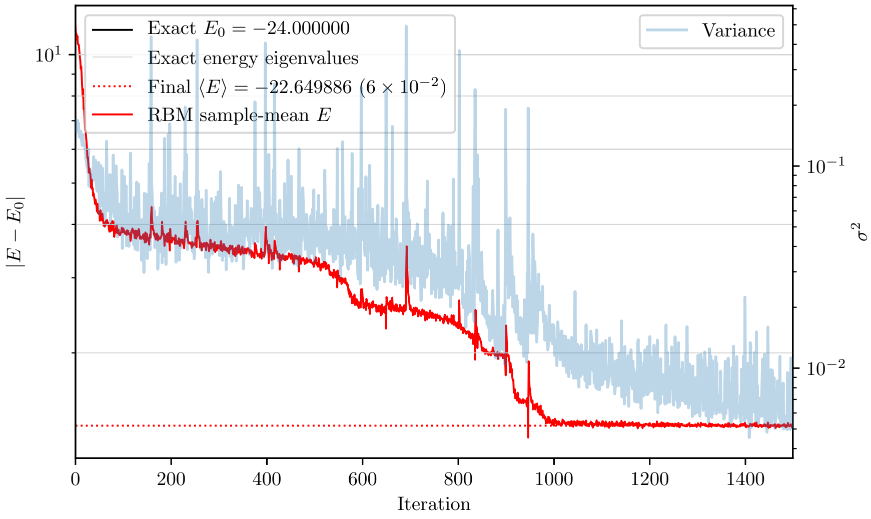
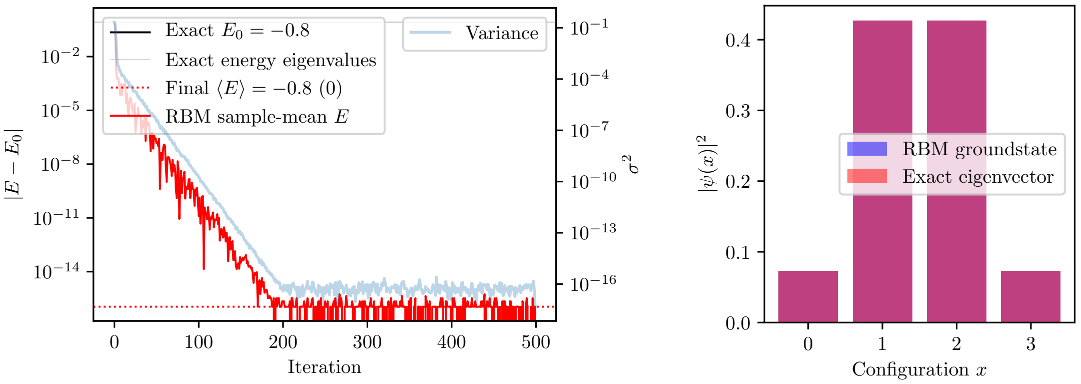


Figure 5.3: 4 × 4 tight binding model. Optimization progress of finding the groundstate using an RBM.

5.2.2 Solving the Hubbard Model with an RBM Ansatz

Interestingly, the same optimization that was done for the simplest 2-site tight binding model in Fig. 5.1, but now with $U = 4$, reaches numerical precision 4 times faster, see Fig. 5.4. With the Coulomb interaction included, the configurations with double occupancies become less probable (Fig. 5.4b).



(a)

(b)

Figure 5.4: 2-site Hubbard model with $U = 4$. (a) Groundstate energy learning progression and (b) final probability density using an RBM ansatz compared to exact results. Faster convergence than in the case of $U = 0$.

For a 2×2 lattice, getting stuck in a local energy minimum at an excited state already seems to be a problem. In Fig. 5.5 the stochastic reconfiguration method spends a long time apparently being stuck in an excited eigenstate, until it finally manages to escape it. It seems as though the RBM is capable of representing the quantum state, the difficulty lies in finding good optimization steps. An example of running the same simulation but with basic stochastic gradient descent instead of the stochastic reconfiguration method is shown in Fig. 5.6. Compared to stochastic reconfiguration, the plain stochastic gradient descent algorithm usually gets stuck a lot quicker and takes more steps to reach good results if it can.

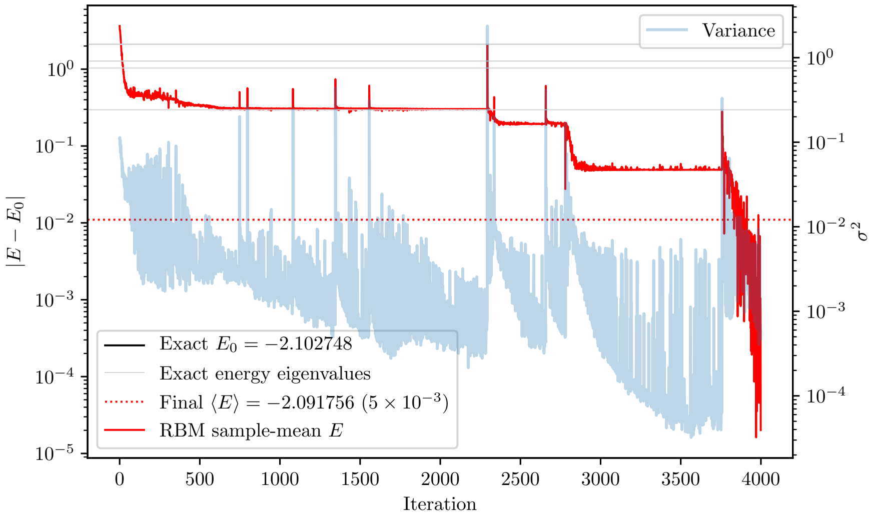


Figure 5.5: 2 × 2 Hubbard model groundstate optimization progress of an RBM intermittently getting stuck at exact eigenvalues.

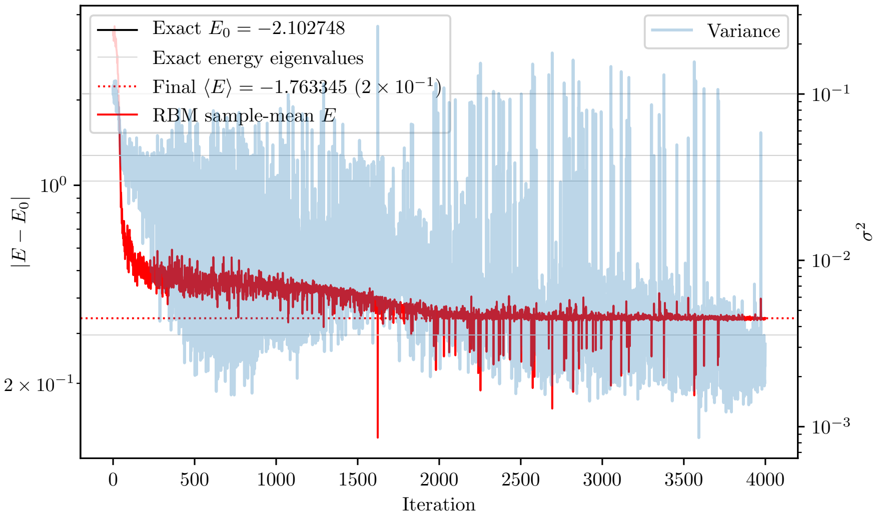


Figure 5.6: 2 × 2 Hubbard model groundstate optimization progress of an RBM using basic stochastic gradient descent instead of the stochastic reconfiguration method.

As an experiment, modifying the sampler to start from a random configuration at every iteration and using a momentum optimization scheme to smooth out the jumps in the parameters that now occur, seems to help these small systems to overcome such barriers. In Fig. 5.7 both energy optimization curves are compared. For larger lattices this technique seems not to be able to improve the final energy and can lead to divisions by very small $\psi(x)$ when computing the local energy for a very unlikely configuration.

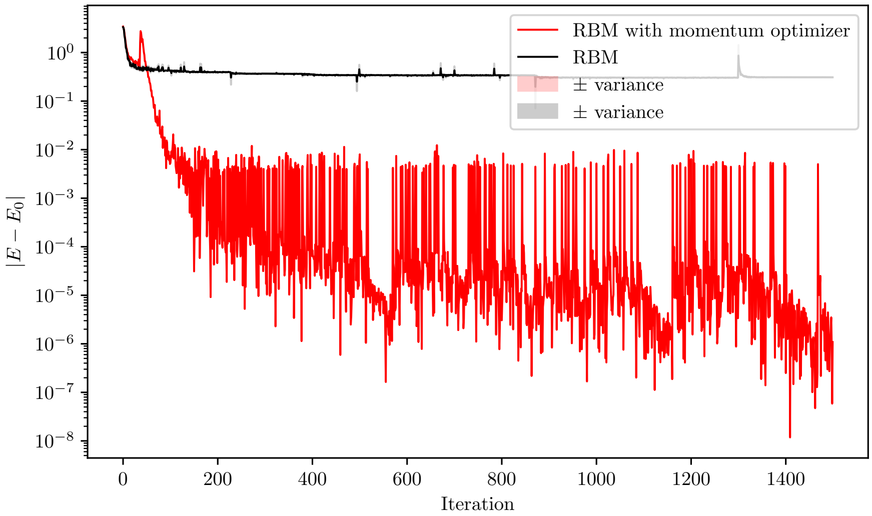


Figure 5.7: 2×2 Hubbard model groundstate optimization progress comparison of an RBM using the stochastic reconfiguration method and in addition using a momentum scheme for the parameter update and resetting the sampler to random configurations at the start of every iteration.

While the tight binding groundstate was degenerate, now there is no degeneracy and the final groundstate of the momentum-optimized RBM can be compared to the exact eigenvector in Fig. 5.8. Compared to the Heisenberg model, the symmetric picture is gone now, as the wavefunction is negative for some configurations. Therefore the requirement for a complex-valued RBM becomes clear and the sign structure seems to be learned correctly. The probabilities are a little asymmetrical in the RBM result, so the most likely configurations, at 24 % each,

$$\begin{pmatrix} -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix} \Longleftrightarrow \begin{array}{|c|c|}\hline \downarrow & \uparrow \\ \hline \uparrow & \downarrow \\ \hline \end{array}$$

and

$$\begin{pmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix} \Longleftrightarrow \begin{array}{|c|c|}\hline \uparrow & \downarrow \\ \hline \downarrow & \uparrow \\ \hline \end{array}$$

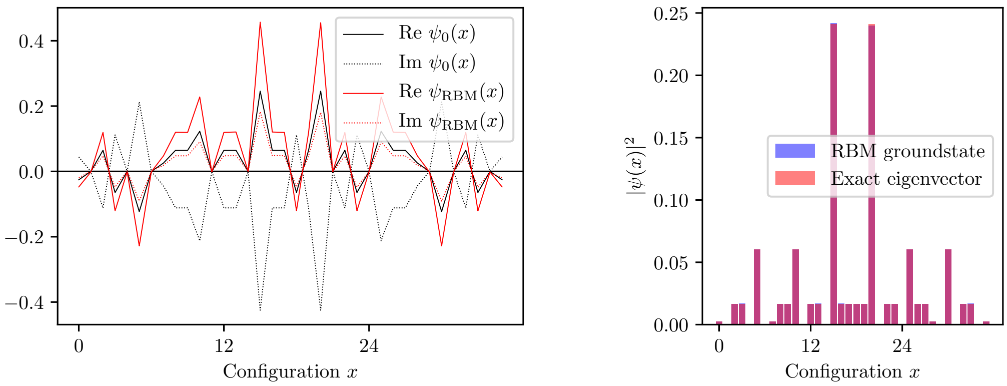
are not sampled equally. In the worse optimization case, these probabilities are even more asymmetric, and for some configurations a wrong sign is assigned. Another such pair is

$$\begin{pmatrix} -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 \end{pmatrix} \Longleftrightarrow \begin{array}{|c|c|}\hline \downarrow & \uparrow\downarrow \\ \hline \uparrow & 0 \\ \hline \end{array}$$

and

$$\begin{pmatrix} 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 \end{pmatrix} \Longleftrightarrow \begin{array}{|c|c|}\hline \uparrow & 0 \\ \hline \downarrow & \uparrow\downarrow \\ \hline \end{array}.$$

These correspond to a particle-hole transformation. Unfortunately, while the probabilities have this symmetry, the sign structure does not and this affects the values of the local energies. A fundamentally particle-hole invariant formulation of the wavefunction ansatz could solve these inaccuracies.



(a)

(b)

Figure 5.8: 2 × 2 Hubbard model. (a) Real and imaginary part of the wavefunction and (b) final probability density function of an RBM ansatz compared to exact results ψ_0.

When looking at computing larger systems, like a 4 × 4 lattice (Fig. 5.9), it becomes clear that this ansatz in its basic form seems to be unable to efficiently learn a good approximation of the ground state of the Hubbard model. The best results achieved reached a relative error of about 9 % but converged very slowly. Looking at the exact evaluation of the expectation values discussed in the next section, it might simply take a lot more update steps to eventually reach some abrupt change in energy, but this could not be observed with the accessible computing time.

To summarize, using the stochastic reconfiguration method speeds up convergence considerably compared to basic stochastic gradient descent. Either way, the complex-valued RBM seems to get stuck at local minima that are not always at excited eigen-energies. Abrupt jumps to better energy values appear with longer phases of a constant error between them. For small systems introducing a momentum scheme and more randomization during the sampling helped to overcome these barriers.

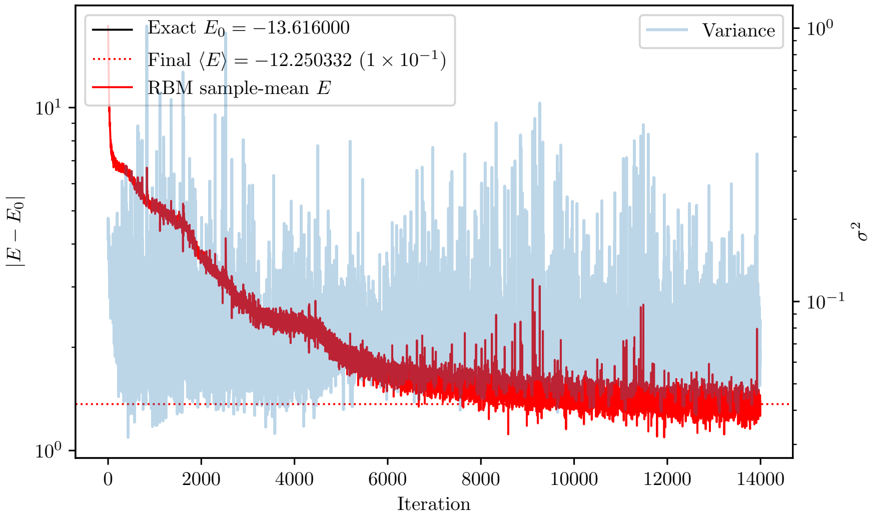


Figure 5.9: 4 × 4 Hubbard model groundstate optimization progress using an RBM ansatz.

5.2.3 Exact Evaluation of the Expectation Value

To further investigate how an RBM learns the groundstate of the 2 × 2 Hubbard model, modifications to the library code were made to compute the expectation value Eq. (3.3) exactly, instead of evaluating it using Monte-Carlo samples. This way, the true gradient rather than a stochastic estimate is found. For this to work, in the implementation of the VMC algorithm the gradient Eq. (3.16), assuming a real-valued local energy, had to be replaced with the full complex form written in Equation (3.13).

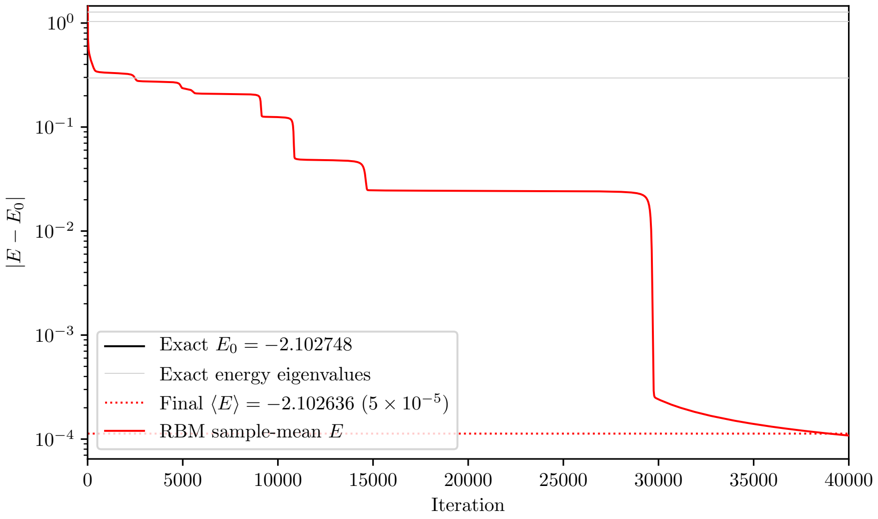
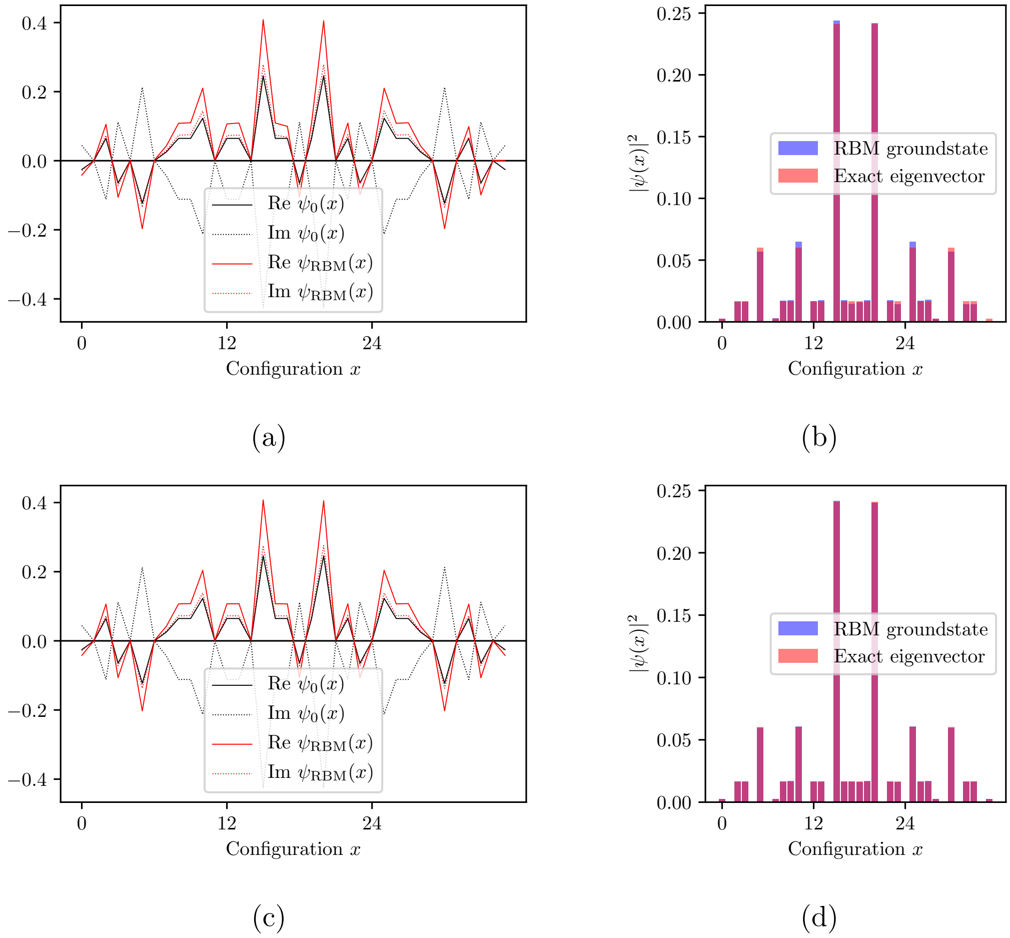


Figure 5.10: Learning the 2 × 2 Hubbard model groundstate using exact expectation values rather than Monte-Carlo samples for updating the RBM parameters.

In Fig. 5.10 the optimization progress shows long episodes of constant energy followed by a larger jump. Compared to the stochastic evaluation it takes a large number of update steps to get to a good result. Adding a Gaussian noise term $\mathcal{N}(0, 0.01)$ to $\psi(x)$ accelerated the convergence. In Fig. 5.11 the wavefunction after 20000 and after 40000 iterations is plotted, showing the difference before and after the last jump in energy from a relative error 10^{-2} to 5×10^{-5}. The main difference is that the probabilities are more symmetric and that the very last configuration in the plot is not considered at all before the jump, while afterwards the network learned the correct negative value. What might be of impact here, is that if the network somehow learns a value 0 for a particular configuration x, it will not sample x in the following step. Also, the $\frac{1}{\psi(x)}$ term in the local energy might be problematic in this case should it appear in the sampled configurations. The output value can change again if the parameters are updated. Although interesting to see, it is hard to derive any kind of direct insight to get to an improved optimization scheme from this.



(a) (b)

(c) (d)

Figure 5.11: 2 × 2 Hubbard model ground state using exact expectation values rather than Monte-Carlo samples. The top images are after 20000 iterations, the bottom images after 40000 iterations, before and after a jump to a better energy. (a), (c) Real and imaginary part of the wavefunction and (b), (d) final probability density function of an RBM ansatz compared to exact results ψ_0.

5.3 Solution to the Hubbard Model Using a Reference Sign Structure and an RBM for Correlations

In the following sections the results for the groundstate of the half-filled $U = 4$ Hubbard model are improved with a variational wavefunction that makes use of a physically motivated reference state.

5.3.1 Introducing a Reference State for the Fermionic Sign Structure

Looking at insights obtained by research so far, there seems to be a difference in difficulty between learning the phases determining the sign structure of a quantum mechanical wavefunction and learning the amplitudes [26]. One way around this is to introduce some physical knowledge about the sign structure of the groundstate. Nomura et al. [16] have proposed to use a reference state, similar to the Jastrow-Slater wavefunction ansatz in VMC calculations described in section 3.3, and a neural network quantum state to model the correlations. The variational wavefunction

$$|\psi\rangle = \sum_x |x\rangle F(x) \langle x|\Phi_{\text{ref}}\rangle \tag{5.9}$$

is split into an antisymmetric function Φ_{ref} encoding the fermionic nodal structure and a RBM $F(x)$. The authors replaced the Gutzwiller and Jastrow correlation functions in conventional VMC calculations with a real-valued RBM and with this they were able to improve the results in finding the Hubbard model's groundstate energy. One disadvantage is that they used a *real-valued* Restricted Boltzmann Machine which can only produce positive numbers resulting in a fixed node approximation. With a sub-optimal sign structure, even in the theoretical case of an infinitely sized neural network, the exact groundstate can then never be achieved. They started with the Fermi-sea, being the groundstate of an uncorrelated tight binding model, as the most basic reference state.

5.3.2 Improving the Sign Structure With a More Advanced Reference State

While the Fermi-sea sign structure improves the result considerably compared to the plain RBM ansatz, to get competitive results it took including a better sign structure into the reference state by using the pair-product state

$$|\Phi_{\text{pair}}\rangle = \left(\sum_{i,j=1}^{L} \sum_{\sigma,\sigma'=\uparrow,\downarrow} f_{ij}^{\sigma\sigma'} c_{i\sigma}^{\dagger} c_{j\sigma'}^{\dagger} \right)^{\frac{N_e}{2}} |0\rangle \tag{5.10}$$

with more variational parameters $f_{ij}^{\sigma\sigma'}$ [16]. This state, also called the Geminal or also Pfaffian wavefunction, is an extension of the Slater-determinant [15] but also introduces more variational parameters, being able to adapt during the optimization. This ansatz therefore represents a more advanced variational function, combining a neural network with prior physical knowledge. Due to limited time and the focus on the neural network part of the variational ansatz, this function was not used in this thesis.

5.3.3 Solving the Hubbard Model Using a Fermi-Sea Reference State

To implement a Fermi-sea reference state the NetKet library was extended with a couple of C++ classes. A *Reference-state machine* class was created that can be used with an arbitrary pair of neural network and reference state. A Fermi-sea state class that fills the system with a list of occupied $\boldsymbol{k}$ vectors and implements the projection to real space according to Equation (2.20) was paired with an RBM.

At first a relatively small 3×2 Hubbard model with periodic boundary conditions is investigated. For the half-filled model a relative error of 7×10^{-2} is reached with some excited energies still below the result. After taking out an electron, at $N_e = 4$ the relative error becomes 1×10^{-2}. Known results of the 4×4 half-filled Hubbard model [18] can be approximated only up to about 20 %. When looking at the lowest $\boldsymbol{k}$ states in the Fermi-sea the groundstate turns out to be degenerate for periodic boundary conditions. With an electron missing in the 3×2 simulation, this degeneracy is lifted and a much better result was reached. From this and similar simulations, it can be concluded that the degeneracy of the Fermi-sea hinders the variational function to reach a good energy result.

Table 5.1: Energy levels of the 3×2 Tight Binding Model.

(PBC in $X = 3$ and open boundary condition in $Y = 2$.)

n	k_x	k_y	$\epsilon_{\boldsymbol{k}}$
0	0	0	-3
1	0	3.142	-1
2	2.094	0	0
3	-2.094	0	0
4	2.094	3.142	2
5	-2.094	3.142	2

A way of lifting the degeneracy is to introduce antiperiodic boundary conditions as described in section 2.3.1 along one axis, termed APBC-PBC. This is done by adding a phase when an electron jumps across the border in the implementation of the Hamiltonian and by using new conditions for the allowed k values in the Fermi-sea. In Figure 5.12 the learning progression of the 4×4 half-filled Hubbard model with PBC and APBC-PBC is compared, where the latter achieves a much better result of about 2.7 %. Finally, the results of [16] using the Fermi-sea are reproduced and the relative errors compared to known results [18] for different model complexities α are plotted in Fig. 5.13.

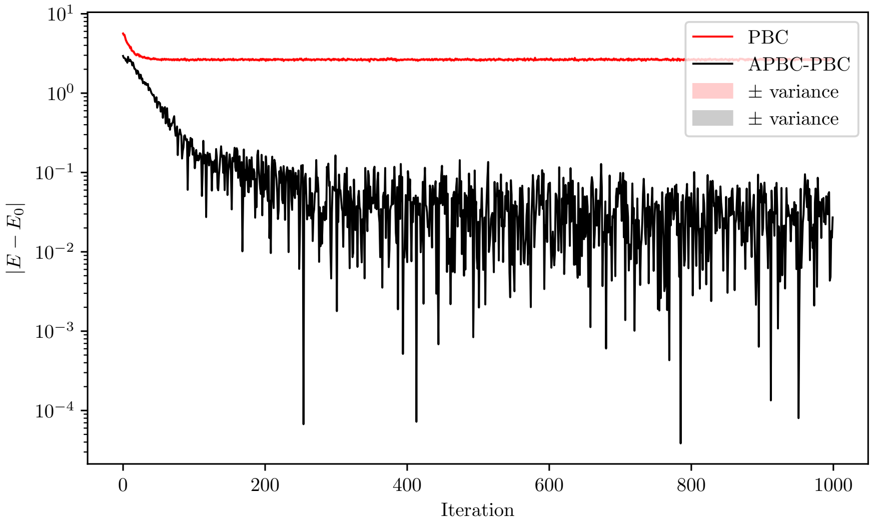


Figure 5.12: 4 × 4 Hubbard model groundstate learning progression of an RBM ansatz with a Fermi-sea reference state which is degenerate using PBC compared to using APBC-PBC.

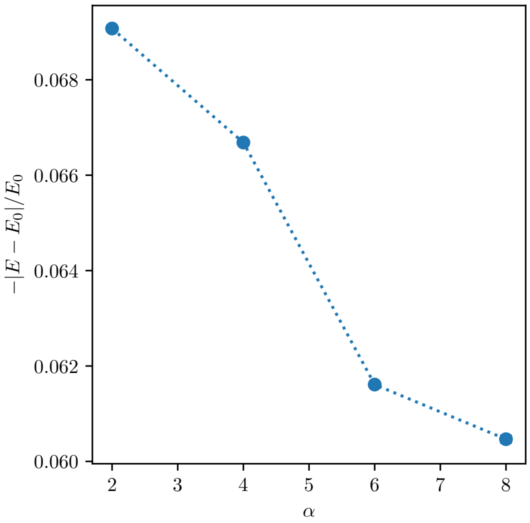


Figure 5.13: Relative errors to known results [18] of a RBM with a Fermi-sea reference state for the 8 × 8 Hubbard model.

Nomura et al. [16] used the mapping to an attractive model by applying the staggered particle-hole transformation described in section 2.2.1, claiming it improved the convergence when using the pair-product reference state. For the Fermi-sea approach no difference could be observed when using a positive or negative U. Also simply lifting the restriction of real parameters with a complex RBM did not improve the obtained energy for large systems.

5.3.4 Considering Only the Complex Phases of the Reference State

In general an issue arises if certain configurations x don't appear in the reference state, $\langle x|\Phi_{\text{ref}}\rangle = 0$, because as a result the variational function always vanishes at these x. This is the case with the Fermi-sea, as can be seen in Figure 5.14, where the RBM is zero for a range of configurations that should have a small but nonzero contribution to the exact groundstate. A plausible solution to this, since the reference state is there mainly for determining the sign structure, is to only use the phases. So in order to avoid reducing the representational power of the wavefunction, only the phase ϕ_{FS} of the Fermi-sea state

$$\psi(x) = F(x)\Phi_{\text{FS}}(x) \to F(x)e^{i\phi_{\text{FS}}} \tag{5.11}$$

could be taken where it is calculated as

$$\phi_{\text{FS}} = \text{atan}\left(\frac{\text{Im}\,\Phi_{\text{FS}}(x)}{\text{Re}\,\Phi_{\text{FS}}(x)}\right) .$$

The phase equates to zero in the case that $\langle x|\Phi_{\text{ref}}\rangle = 0$, so the reference state value is just 1. This enables a variational wavefunction to take nonzero values at these points in configuration space while also starting from a physically motivated sign structure. The relative errors for the 3 × 2 computations discussed above improve by an order of magnitude in the case where the Fermi-sea is not degenerate, when using only the phases. For larger models sadly no improvement could be achieved with this modification.

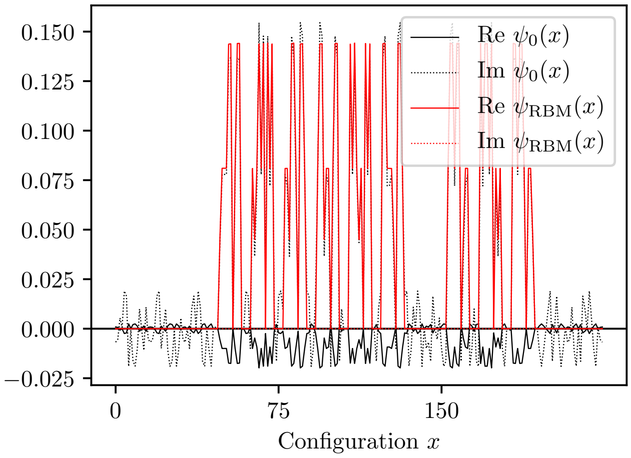


Figure 5.14: 3 × 2 Hubbard model with $N_e = 4$. Groundstate variational wavefunction compared to the exact eigenvector ψ_0. Because the Fermi-sea reference state is zero for certain configurations, the output of the entire wavefunction is zero as well.

5.3.5 Density Correlation Functions

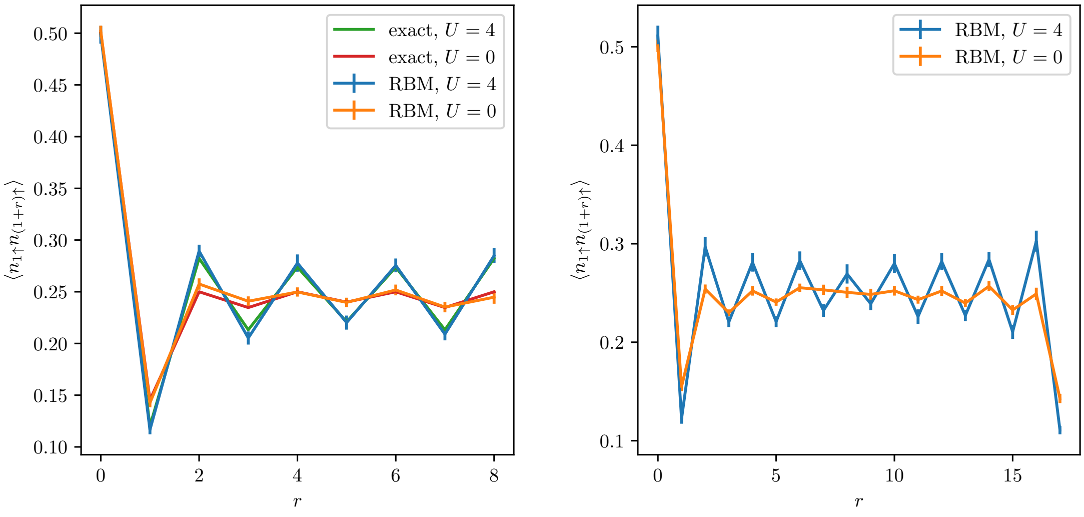
To determine whether basic known physical properties are actually captured by this variational form, the density-density correlation function is calculated for a one-dimensional Hubbard model with periodic boundary conditions. The parallel-spin density-density correlation function is

$$\begin{aligned}\langle \hat{n}_{i\sigma}\hat{n}_{(i+r)\sigma}\rangle &= \langle\psi_0|\hat{n}_{i\sigma}\hat{n}_{(i+r)\sigma}|\psi_0\rangle \\ &= \sum_{x,x'}\psi(x')^*\psi(x)\langle x'|\hat{n}_{i\sigma}\hat{n}_{(i+r)\sigma}|x\rangle \\ &= \sum_{x}|\psi(x)|^2\, n_{x_{i,\sigma}}n_{x_{i+r,\sigma}} \\ &\approx \frac{1}{N_{\text{Samples}}}\sum_{x\sim|\psi_0(x)|^2} n_{x_{i,\sigma}}n_{x_{i+r,\sigma}}\end{aligned} \tag{5.12}$$

with the occupation numbers $n_{x_{i,\sigma}}$ of site i with spin σ for a specific configuration x. Since the fermionic occupations are mapped to a spin chain where an empty site has a value -1, for the calculation of Eq. (5.12) the density operators in the first two lines can be replaced by local spin operators

$$\hat{S}_\alpha = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

that retrieve the σ^z value only for $|\uparrow\rangle$. As an exact evaluation to compare the results, the third line in Eq. (5.12) is used. This correlation function is plotted in Fig. 5.15 for both $U = 0$ and $U = 4$. Both the electron density $\langle n_1 n_1\rangle = \frac{1}{2}$ at $r = 0$ and the value for uncorrelated electrons $\frac{1}{4}$ approaching large distances are correctly reproduced. Also the exchange hole at $r = 1$ due to the Pauli exclusion principle is obtained in the tight binding model, and when switching on the interaction correlations become stronger.



(a)

(b)

Figure 5.15: Parallel spin density-density correlations for Hubbard and tight binding models calculated with an RBM on a (a) 10-site chain compared to exact results and (b) on an 18-site chain. Both are symmetric about $L/2$.

The expectation value of the occupation of a k-state is obtained by Fourier-transforming the operators

$$\begin{aligned}
\langle \hat{n}_k \rangle = \langle \psi | \hat{c}_k^\dagger \hat{c}_k | \psi \rangle &= \langle \psi | \left(\frac{1}{\sqrt{L}} \sum_{l=1}^{L} e^{-ikl} \hat{c}_l^\dagger \right) \left(\frac{1}{\sqrt{L}} \sum_{l'=1}^{L} e^{ikl'} \hat{c}_{l'} \right) | \psi \rangle \\
&= \frac{1}{L} \sum_{l,l'=1}^{L} e^{-ik(l-l')} \langle \psi | \hat{c}_l^\dagger \hat{c}_{l'} | \psi \rangle \\
&= \sum_{r=0}^{L-1} e^{ikr} \langle \psi | \hat{c}_{1+r}^\dagger \hat{c}_1 | \psi \rangle \\
&= \frac{1}{2} + \sum_{r=1}^{L-1} e^{ikr} \langle \psi | \hat{c}_{1+r}^\dagger \hat{c}_1 | \psi \rangle \, .
\end{aligned} \tag{5.13}$$

Translational symmetry and the fact that the summands only depend on $r = |l - l'|$ lead to L terms with $l' = 1$. In the last step the electron density $\langle \hat{c}_1^\dagger \hat{c}_1 \rangle = \frac{1}{2}$ is inserted for the $r = 0$ term. In the Fermi-sea at zero temperature all states up to the Fermi-energy are expected to be filled and none of the higher ones, resulting in a step function. Correlations due to interactions however reduce this sharp Fermi step [8].

To evaluate this, the correlation functions $\langle \hat{c}_{(1+r)\sigma}^\dagger \hat{c}_{1\sigma} \rangle$ need to be calculated and then Fourier-transformed. In the NetKet framework, two-body operators can be written as Kronecker products $\hat{S}_i^+ \otimes \hat{S}_j^-$ where $\hat{c}_{i\sigma}^\dagger$ and $\hat{c}_{j\sigma}$ should become $\hat{S}^+$ and $\hat{S}^-$ for the corresponding sites i and j. However, while this mapping to bosonic spin operators works with the local operators used for the density-density correlations, for these nonlocal fermionic operators obeying the anticommutation relations is nontrivial. The operator $\hat{c}_{(1+r)\sigma}^\dagger \hat{c}_{1\sigma}$ must therefore be implemented along the same lines as the Hamiltonian described in section 5.1.2.

This is calculated for the tight binding and Hubbard model with $U = 4$ on a 18×1 chain with PBC. In Fig. 5.16 the correlation function and its discrete Fourier-transforms are plotted. The Fermi-sea seems to be well represented, but there are some unphysical fluctuations in the unoccupied states. For the Hubbard model the increased localization due to the Coulomb interactions results in occupations of higher k-levels. The correlations lead to a reduced Fermi-step which is clearly visible. On larger systems the step at the Fermi-level also for the Hubbard model becomes sharper while with an increased model complexity α the fluctuations in the density are further reduced.

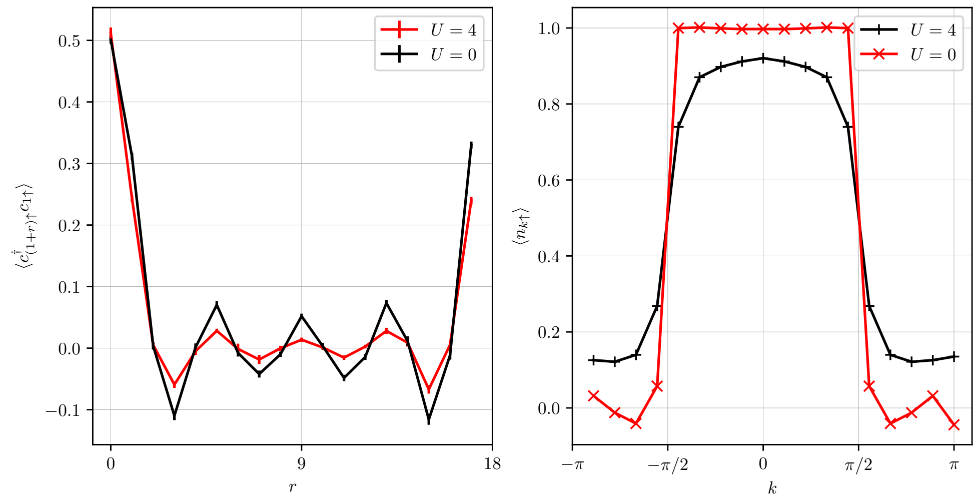


Figure 5.16: Hubbard and tight binding k density expectation value (right) as the discrete Fourier transform of the correlations functions (left) calculated with an RBM on an 18-site chain. The reduced Fermi-step for the Hubbard model due to correlations is clearly visible.

5.4 Possible Improvements and Outlook

As already mentioned, a better reference state is needed for good results, in the case of learning only correlations with the NQS ansatz. The pair product state for example is what has been used in conventional VMC packages as the nodal part of the wavefunction [15] and was also used by [16] for their results.

When trying to learn the complete groundstate, it may be of help to consider that the nodal structure and the modulus of the wavefunction do not share the same symmetries. While the probabilities are symmetric under, for example, a particle-hole transformation, the signs may change, as seen in Figure 5.11. This among other things motivates trying to separate the phase and the modulus of the complex wavefunction during optimization. An ansatz

$$\psi(x|\,\theta,\gamma) = S_\theta(x)A_\gamma(x) \tag{5.14}$$

similar to the idea of the reference state for the nodal structure could be used, where $S_\theta(x)$ in the simplest case just maps to a sign from $\{-1,1\}$ and $A_\gamma(x)$ is positive and real. A reference state could be added as yet another factor. In a machine learning workflow one of the first tasks is the creation of features from the data, considering symmetries of the given problem, that are then taken as the input of the neural network. Now, following such a procedure, certain invariances could be enforced just for the amplitude part A while the nodal part S can be parameterized by a different kind of network and with a higher complexity if needed. An implementation of this was attempted in this work as well, but it turned out to require extensive modifications to the code base of the software library beyond what could be achieved in the available time.

There also seems to be some consensus in the active community that the sign-structure is the more difficult thing to learn also for neural network quantum states. Recent research [29] has shown, in a supervised learning scheme, that fixing the amplitudes to exact values and learning the sign structure is considerably more difficult than learning the amplitudes with fixed signs. Another very recent work [26] already applied this idea of separately learning modulus and phase of the wavefunction and the authors also noted that this effect is apparently more dramatic with CNN architectures than RBM. For frustrated systems they achieved much better results compared to other variational functions by improving the initialization, first only learning the signs with fixed amplitudes, and then optimizing both concurrently.

Up to now most of the work involving NQS was focused on quantum spin systems. One new idea that has been put forward to tackle fermions is optimizing so-called backflow wavefunctions consisting of slater determinants of many-body orbitals that are each parameterized by a neural network [17]. For optimizing deep learning architectures as quantum states, that generally help reducing parameters and often offer certain invariances by design, some progress has also been made [30]. While convolutional neural networks can readily be applied to configurations of spin models like the Heisenberg model, for fermionic systems there are more local states per lattice site than just up or down. Mapping the occupation numbers to spins ordering the spins first and the sites last removes the locality of a particular site's up and down indices. In this case a typically sized convolutional kernel would not have both the up and down occupation number of a that site in its input frame at the same time. A Different mapping is necessary, for example introducing another dimension for up and down spins like in an RGB image there are different color channels. Another area of research could be using generative neural networks, for which a lot of progress has been made in recent years. Generative adversarial networks (GAN) can produce new samples of a learned distribution and could replace the Monte-Carlo sampling part of the calculations by directly sampling from the probability distribution.

Chapter 6

Conclusion

The focus of this work was on solving the two-dimensional Hubbard model with the method of neural network quantum states (NQS). This new method based on variational Monte-Carlo promises to find better representations for quantum many-body states and opens opportunities to benefit from recent advancements of machine learning frameworks [3]. From a quick review of the restricted Boltzmann machine (RBM) it can be argued that this approach seems superior to conventional tensor network approaches for encoding strongly correlated states. Moreover, convolutional neural networks should possess an even greater expressive power. While there are already a lot of results for quantum spin systems, the way to properly treat fermionic problems with NQS is still an open research question and only a few published results exist so far.

After understanding the main ideas and concepts, some early results for the Heisenberg model were reproduced. The main effort of this thesis was to implement a fermionic neural network quantum state and the Hubbard Hamiltonian in the framework of variational Monte-Carlo. Mapping the fermionic degrees of freedom to a spin chain, the implementation could be based on a software library available for quantum spin models. For small lattices the resulting probability distributions and complex wavefunctions were investigated to try to find a way for the RBM to learn a good groundstate approximation more efficiently. Different optimization methods and hyperparameters were tested. As another tool to verify that the stochastic nature of the optimization algorithm indeed provides a good gradient estimate, an exact evaluation of the expectation values was implemented. It turned out that noisy gradients in practice lead to a faster convergence. Though it seems that the RBM is capable of representing the strongly correlated groundstate, finding good optimization steps to get there is difficult, as in most cases during the optimization infrequent jumps in energy followed by long slow-moving periods appear.

Recent literature suggests that the sign structure, determined by the complex phases, is more difficult to learn than the modulus of the wavefunctions. Fermionic models or frustrated systems that suffer from the negative sign problem, at first glance seem to also be more difficult to solve with the approaches tested so far. This difficulty of learning sign structures was also realized in this thesis. A reference state implementation, restricting the variational neural network ansatz to describe only correlations, reaches better results. A Fermi-sea reference state was used in this thesis, but for an actual application better representations for the fermionic sign structure are needed and available. This reproduces the so far only published research for applying NQS to the Hubbard model [16], where the authors could improve upon results by other variational forms. In order for this method to work, degeneracy of the reference state must be taken into account and eventually convergence to a fairly good approximation was reached quickly. Some efforts were made to handle the special case where the reference state wavefunction is zero for a certain configuration, but except for very small systems, no improvements could be achieved. The density-density correlation function and the momentum space occupation expectation value were calculated and display the expected Fermi-sea occupations and the reduced Fermi-step due to correlations. An effort to implement the idea to separate the parameterization of the sign and amplitude of the wavefunctions, for which in the meantime an attempt already appeared in the literature and showed promising results, could not be finished in time.

A further outlook on research opportunities outlines multiple points of attack, as for example significant improvements in performance can be expected by including inherent symmetries and invariances into the variational architecture.

Bibliography

[1] C. Aggarwal. *Neural Networks and Deep Learning: A Textbook.* Springer International Publishing, 2018.

[2] G. Carleo, K. Choo, D. Hofmann, J. E. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin, M. Mauri, G. Mazzola, C. B. Mendl, E. [van Nieuwenburg], O. O'Reilly, H. Théveniaut, G. Torlai, F. Vicentini, and A. Wietek. Netket: A machine learning toolkit for many-body quantum systems. *SoftwareX*, 10:100311, 2019.

[3] G. Carleo and M. Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602–606, Feb. 2017. arXiv: 1606.02318.

[4] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang. Equivalence of restricted Boltzmann machines and tensor network states. *Physical Review B*, 97(8):085104, Feb. 2018. arXiv: 1701.04831.

[5] K. Choo, G. Carleo, N. Regnault, and T. Neupert. Symmetries and Many-Body Excitations with Neural-Network Quantum States. *Physical Review Letters*, 121(16):167204, Oct. 2018.

[6] K. Choo, A. Mezzacapo, and G. Carleo. Fermionic neural-network states for ab-initio electronic structure. *arXiv:1909.12852 [cond-mat, physics:physics, physics:quant-ph]*, Sept. 2019. arXiv: 1909.12852.

[7] K. Choo, T. Neupert, and G. Carleo. Study of the Two-Dimensional Frustrated J1-J2 Model with Neural Network Quantum States. *Physical Review B*, 100(12):125124, Sept. 2019. arXiv: 1903.06713.

[8] P. Fazekas. *Lecture notes on electron correlation and magnetism.* World Scientific, Singapore River Edge, N.J, 1999.

[9] J. K. Freericks, B. K. Nikolic, and O. Frieder. The Nonequilibrium Many-Body Problem as a paradigm for extreme data science. *International Journal of Modern Physics B*, 28(31):1430021, Dec. 2014. arXiv: 1410.6121.

[10] X. Gao and L.-M. Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature Communications*, 8(1):662, Dec. 2017.

[11] M. C. Gutzwiller. Effect of Correlation on the Ferromagnetism of Transition Metals. *Physical Review Letters*, 10(5):159–162, Mar. 1963.

[12] N. Le Roux and Y. Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural Computation*, 20(6):1631–1649, 2008.

[13] Y. Levine, O. Sharir, N. Cohen, and A. Shashua. Quantum Entanglement in Deep Learning Architectures. *Physical Review Letters*, 122(6):065301, Feb. 2019. arXiv: 1803.09780.

[14] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac. Restricted Boltzmann machines in quantum physics. *Nature Physics*, 15(9):887–892, Sept. 2019.

[15] T. Misawa, S. Morita, K. Yoshimi, M. Kawamura, Y. Motoyama, K. Ido, T. Ohgoe, M. Imada, and T. Kato. mVMC—Open-source software for many-variable variational Monte Carlo method. *Computer Physics Communications*, 235:447–462, Feb. 2019.

[16] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada. Restricted Boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20):205152, Nov. 2017.

[17] D. Pfau, J. S. Spencer, A. G. d. G. Matthews, and W. M. C. Foulkes. Ab-Initio Solution of the Many-Electron Schr\"odinger Equation with Deep Neural Networks. *arXiv:1909.02487 [physics]*, Mar. 2020. arXiv: 1909.02487.

[18] M. Qin, H. Shi, and S. Zhang. Benchmark study of the two-dimensional Hubbard model with auxiliary-field quantum Monte Carlo method. *Physical Review B*, 94(8):085103, Aug. 2016.

[19] A. W. Sandvik. Finite-size scaling of the ground-state parameters of the two-dimensional heisenberg model. *Phys. Rev. B*, 56:11678–11690, Nov 1997.

[20] U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259–315, Apr 2005.

[21] U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, Jan. 2011.

[22] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature Communications*, 10(1):5024, Dec. 2019.

[23] M. Snajdr and S. M. Rothstein. Are properties derived from variance-optimized wave functions generally more accurate? Monte Carlo study of non-energy-related properties of H2, He, and LiH. *The Journal of Chemical Physics*, 112(11):4935–4941, Mar. 2000.

[24] K. Sohn and H. Lee. Learning invariant representations with local transformations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1339–1346, Madison, WI, USA, 2012. Omnipress.

[25] S. Sorella. Generalized Lanczos algorithm for variational quantum Monte Carlo. *Physical Review B*, 64(2):024512, June 2001.

[26] A. Szabó and C. Castelnovo. Neural network wave functions and the sign problem. *arXiv:2002.04613 [cond-mat, physics:physics, physics:quant-ph]*, Feb. 2020. arXiv: 2002.04613.

[27] M. Troyer and U.-J. Wiese. Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Physical Review Letters*, 94(17):170201, May 2005.

[28] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[29] T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov. Generalization properties of neural network approximations to frustrated magnet ground states. *Nature Communications*, 11(1):1593, Mar. 2020.

[30] L. Yang, Z. Leng, G. Yu, A. Patel, W.-J. Hu, and H. Pu. Deep Learning-Enhanced Variational Monte Carlo Method for Quantum Many-Body Physics. *arXiv:1905.10730 [cond-mat]*, Aug. 2019. arXiv: 1905.10730.

[31] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.