Department of Physics and Astronomy

University of Heidelberg

Bachelor Thesis in Physics

submitted by

Michael Straub

born in Penarth (United Kingdom)

2019

Neural Network Approach to Open Quantum Systems

This Bachelor Thesis has been carried out by Michael Straub at the
Kirchhoff Institut für Physik in Heidelberg
under the supervision of
Prof. Thomas Gasenzer
and the co-supervisors
Dr. Martin Gärttner and Stefanie Czischek

Zusammenfassung

Für viele hochdimensionale Probleme bietet maschinelles Lernen sehr erfolgreiche Ansätze. Wir wollen das nutzen um die Lindblad-Gleichung für viele Teilchen zu lösen. Die Gleichung berücksichtigt neben dem eigentlichen System noch die Interaktionen mit der Umgebung, weßhalb sie bestens geeignet ist, realistische experimentelle Szenarien zu simulieren. Wir werden in dieser Arbeit einen Variationsansatz mit einem neuronalen Netzwerk präsentieren und untersuchen. Das Netzwerk wird mit maschinellem Lernen optimiert, wobei wir testen werden wie effizient diese Methode ist.

Abstract

For many high dimensionality problems, machine learning has shown great success. We will use this to solve the many-body Lindblad equation. This equation includes coupling with an environment and is therefore well suited to simulate realistic experimental scenarios. We present and investigate a variational neural network ansatz, to represent mixed states. Lastly we show how to use machine learning techniques to train the neural network and test the efficiency of those schemes.

Contents

1 Introduction

It goes without saying, that machine learning is rapidly growing in importance in nearly all fields of science. Machine learning has been used extensively to handle big amounts of data, e.g. for the incredible amount of data in modern particle physics experiments [1]. One of the most prominent topics in machine learning is artificial neural networks (ANN).

Neural networks seem to be omnipresent in fields like image processing [2, 3], speech recognition[4], self-driving cars [5] and so on. All of those problems have in common, that the input is very high dimensional while the output is low dimensional. The neural network is supposed to find low-dimensional patterns in a high dimensional data set[6].

Recently ANN's have been shown to solve many-body problems[7]. In many-body physics, the curse of dimensionality (the exponential growth of the system's Hilbert space with particle number) stops exact computation of solutions even for a small number of particles. Luckily, for most problems, only a small physical portion of the Hilbert space has to be considered. Conceptually, the neural network has to find the low dimensional subspace in the huge Hilbert space.

If we want to simulate realistic experimental scenarios we cannot assume perfect isolation. Therefore involvement of the environment has to be considered. To approach this problem we cannot rely on the system to be a pure state and have to consider mixed states. Compared to a pure state the number of parameters grows quadratically. The problem is therefore conceptually harder.

The dynamics of open quantum systems are often well captured by the Lindblad master equation. The time evolution converges to some stationary solution called steady state.

To represent states with area law entanglement, DMRG and MPS states have been the state of the art since 1992 [8, 9] and remain mostly unchallenged. Tensor networks are also used to solve the Lindblad master equation[10]. Those methods showed great success, but are only feasible for one dimensional problems[11] and collapse with volume law entanglement.

The focus of this work will be the calculation and representation of steady state solutions of the Lindblad master equation. We concrete ourselves with simulating a Rydberg system[12].

Central in this work will be to exceed the small entanglement limit of Tensor networks. The objectives is to find an efficient neural network representation of steady states and study its properties.

As a brief outline of this thesis, we will quickly introduce the necessary theoretical framework. Then the thought process of constructing the variational neural density operator will be presented. After establishing the methods used, we will start testing if the neural density operator is capable of representing the steady state with reasonable precision. We study the representative power of the network and the number of parameters necessary. In the last chapter we will investigate the efficiency of the network's learning capabilities by applying active learning techniques.

The neural network approach to open systems has been of high interest lately [13–16]. Other groups showed that one can use Monte Carlo methods to do time evolution and find steady states. We will instead try to frame the limits of the neural network approach. Additionally, active learning is not used in any of the other publications, which is why we want to test the efficiency of this approach.

2 Theory

Here we want to introduce the theoretical framework necessary for the neural network approach to open systems. We will start with the Lindblad master equation (§2.1) and talk about its steady state solutions (§2.2). Then we continue with the Rydberg model (§2.3) and finish this chapter by introducing the restricted Boltzmann machine and the neural network state ansatz (§2.4).

2.1 Lindblad Master Equation

A correct description of an open system must feature incoherent processes, e.g. the spontaneous emission of a photon [17].

A master equation formalism is used in Quantum optics since 1965 and has since shown great success[18, 19]. The formal derivation includes several approximations[20].

- *Born approximation*: the bath and the system couples weak s.t. the bath is not affected by the systems dynamics.
- *Markovian approximation*: neglect memory effects in the system
- *Markovian environment*: the timescale of the bath is much smaller than the system's

After those approximations, we can trace out the environment's degrees of freedom and end with an equation for a mixed state, describing the systems dynamics. For a full derivation, we refer to *The Theory of Open Quantum Systems* by H. Breuer and F. Petruccione[17].

We will look at small laser driven systems. The bath aka the laser is therefore mostly unaffected by the system, thus we can assume the approximations to hold.

The Lindblad master equation consists of the von Neumann term

$$-i[\mathscr{H}, \rho]$$

describing the system's internal dynamics and the incoherent Lindblad terms

$$L_k(\rho) = \gamma_k(\Gamma_k \rho \Gamma_k^\dagger - \frac{1}{2}\Gamma_k^\dagger \Gamma_k \rho - \frac{1}{2}\rho \Gamma_k^\dagger \Gamma_k),$$

where $\{\Gamma_k\}$ is a diagonal operator basis and γ_k its coefficients. The Lindblad terms will enact the interaction with the environment.

The full Lindblad master equation is:

$$\frac{d\rho}{dt} = -i[\mathscr{H}, \rho] + \sum_k L_k(\rho) = \mathscr{L}(\rho) \tag{2.1}$$

2.2 Steady State

Most dissipative systems will evolve to a *steady state* for $t \to \infty$. In terms of the master equation this means $\frac{d\rho_{\text{steady}}}{dt} = 0$.

Often $\mathscr{L}$ is called superoperator, as it is an operator acting on another operator. Formally the steady state is the eigenstate of the superoperator with eigenvalue $\lambda_0 = 0$. It is shown that for the Lindblad master equation (with time-independent Hamiltonian and finite Hilbert space) there exists at least one steady state ρ_S [21].

The other eigenvalues of the superoperator have negative real parts, as they are responsible for the relaxation to the steady state [22]. We can identify a time scale of the system by calculating $\Delta_L = -\text{Re}(\lambda_1)$, where λ_1 is the highest non-zero eigenvalue. This quantity is called the Liouvillian gap. A small gap results in slow relaxation to the steady state [22].

2.3 Many-body-Rydberg Ensemble

The ensemble we want to describe consists of N Rydberg-atoms able to be in either their ground state $|g\rangle$ or their excited Rydberg state $|r\rangle$. We will couple the two states with a laser.

The Hamiltonian we will look at is the following:

$$\mathscr{H} = \sum_{i=1}^{N} \sigma_x \Omega + \sum_{<ij>} |r_i\rangle \langle r_j| \frac{J}{r_{ij}^6} + \sum_{i=1}^{N} |r_i\rangle \langle r_j| \Delta \tag{2.2}$$

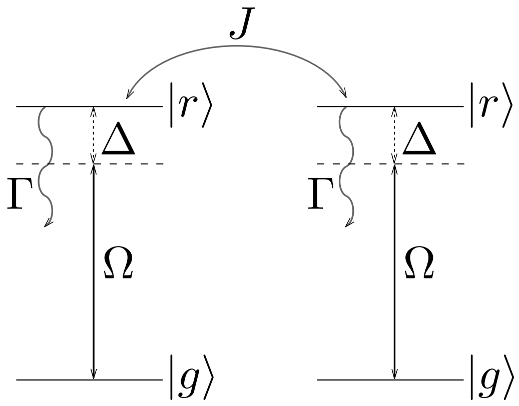

Figure 2.1: The Rydberg system with Interaction strength J, level coupling Ω, laser detuning Δ and spontaneous emission Γ.

The first term describes the laser coupling between the states with strength Ω. The next term describes the strong short ranged *van der Waals* interactions scaling with r^6. One could also think of the longer ranged dipole-dipole interaction with r^3 or ionic coupling with r^1. The interaction term is only active between two excited atoms. Additionally, we add a simple detuning Δ, shifting the energy of the excited state.

Finally, there will be a spontaneous decay to the ground state.

$$L_i(\rho) = \gamma \left|g_i\right\rangle \left\langle r_i\right| \rho \left|r_i\right\rangle \left\langle g_i\right| - 1/2\{\left|g_i\right\rangle \left\langle g_i\right|, \rho\}, \tag{2.3}$$

where γ is the decay rate.

This system model can be realised experimentally by Rydberg atoms in a Bose-Einstein condensate[23].

2.4 Neural Network Quantum State

One of the most studied network structure is the restricted Boltzmann machine architecture. It is a probabilistic model, well known for its application in dimensionality reduction [24] and feature learning [25].

The network consists of two layers which are connected to each other but have no connections in between one layer (this is the reason why the architecture is called restricted). The first layer will consist of N visible nodes and get the input

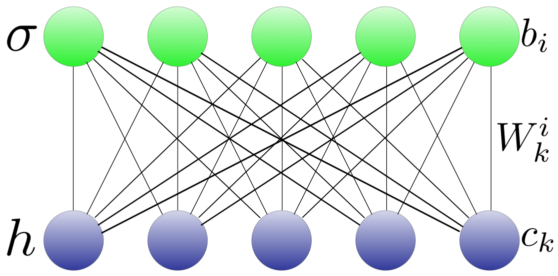

Figure 2.2: Fully connected Restricted Boltzmann machine with 5 visible units σ and 5 hidden units h. Each node has its unique bias (either b_i or c_k) and σ_i is connected to h_k via the matrix element W^k_i.

σ. Each visible node will be connected to the hidden layer h which consists of αN nodes where α is the density of hidden nodes per visible ones. The connection will have a strength $W = (W_{ij})^{i=1...N}_{j=1...\alpha N}$. At the same time, all nodes will have a bias ($b = (b_i)_{i=1...N}$) (for the visible) and $c = (c_j)_{j=1...\alpha N}$ (for hidden nodes) (compare Figure 2.2).

The name *Boltzmann machine* originates from the concept of using the network's "energy"for the configuration (σ, h)

$$E(\sigma, h, \mathscr{W}) = \sum_{i=1}^{N} a^i \sigma_i + \sum_{j=1}^{\alpha N} b^j h_j + \sum_{i,j} W^{ij} \sigma_i h_j \tag{2.4}$$

to calculate a pseudo probability with the Boltzmann distribution for some configuration of σ (integrating over h)

$$p(\sigma) = \sum_h \exp(E(\sigma, h, \mathscr{W})). \tag{2.5}$$

We can now use this distribution to approximate a many-body Quantum state [7] of N particles σ_i

$$|\Psi\rangle = \frac{1}{Z} \sum_\sigma p(\sigma) |\sigma\rangle, \tag{2.6}$$

where Z is the normalisation $Z = \sum |p(\sigma)|^2$

This ansatz is called the *Neural Network Quantum State* (NQS).

In the following, we will look at binary systems $\dim(\sigma_i) = 2$. We will choose the arbitrary values $\sigma_i \in \{-1, 1\}$, $h_j \in \{-1, 1\}$. For this choice we can calculate the sum over all configurations of h explicitly:

$$p(\sigma, \mathscr{W}) = \exp\left(\sum_{i}^{N} a^{i} \sigma_{i} \right) \cdot \prod_{j}^{\alpha N} 2 \cosh \left(b_{j} + \sum_{i}^{N} W_{j}^{i} \sigma_{i} \right). \tag{2.7}$$

The NQS is known to represent the ground states of interacting systems [7] and is used for state tomography [26]. We know that states exist, where the efficient representation collapses [27]. We can extend the power of a Neural Network by stacking additional hidden layers [27] to make the network deep. But, this keeps us from efficiently calculating the sum over the hidden configurations to get $p(\sigma)$.

The RBM architecture features volume law entanglement entropy and even an analytic expression for a fully entangled state [28].

3 Neural Density Operator

For an open system, we will generally have a mixed state, thus we cannot use the pure state ansatz by Carleo [7] anymore. Here we will introduce a neural network to represent a density operator. We will start with a trivial ansatz (§3.2) and show its limits. We then continue with the concept of space purification (§3.3) and how we can use this for an efficient representation (§3.4).

3.1 Density Operator

THe density operator is the quantum version of a statistical ensemble of states. The basic properties are:

- ρ is hermitian
- $\mathrm{tr}(\rho) = 1$ which is essentially a normalizing factor
- ρ is semi-positive which means all eigenvalues $\lambda_i \geq 0$

As an ensemble of quantum states, the most general form is

$$\rho = \sum_i p_i \left|\Psi_i\right\rangle \left\langle\Psi_i\right| \tag{3.1}$$

with the probabilities $p_i \in \mathbb{R}$ and the normalisation $\sum_i p_i = 1$.

Eq.(3.1) represents all possible density operators if and only if $\{\Psi_i\}$ is a basis of the whole Hilbert space $\mathcal{H}$.

3.2 Trivial Ansatz

With the Neural Network Quantum State (NQS) and the general form of (3.1) we can simply represent all Ψ_i as individual networks with the probabilities p_i as additional parameters.

For $n = \dim(\mathcal{H})$, we would need n individual networks to represent ρ over a full basis. If we choose a smaller amount of networks, we essentially limit the

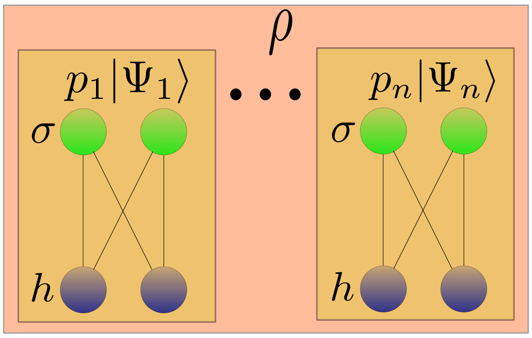

Figure 3.1: n individual networks representing the density matrix ρ as a classical ensemble of NQS states $|\Psi_i\rangle$

Schmidt rank of the density operator by the number of networks. A quantity closely related is the *von Neumann entropy* $S = -\sum_i \mathrm{tr}(\rho \log(\rho))$ of the mixed state. The entropy S is limited by the Schmidt rank S_n

$$S \leq \log(S_n). \tag{3.2}$$

In the derivation of the Lindblad master equation, we see that the mixing of our state originates from tracing out the environment. The entropy of ρ is therefore a measure of entanglement of our system with the environment.

There are cases of steady states being pure (eg. electromagnetically induced transparency [29]). Naturally however, the entanglement with the environment is increasing in time. The steady state being $\rho(t \to \infty)$ means that for most cases large *von Neumann entropy* is expected.

3.3 Space Purification

Purification of a Quantum State is the property that all density operators ρ living in $\mathcal{H}_S$ can be expressed as the partial trace of some pure state $|\Psi\rangle\langle\Psi|$ in an extended Hilbert space $\mathcal{H} = \mathcal{H}_S \otimes \mathcal{H}_E$. The mathematical proof is rather simple [30]:

Consider a general mixed state $\rho = \sum_i p_i |i_A\rangle\langle i_A|$, with $|i_A\rangle$ being a state in $\mathcal{H}_A$. Let $|\Psi_{AB}\rangle$ be a pure state living in $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, where $\mathcal{H}_B$ is a copy of

$\mathcal{H}_A$. By defining $|\Psi_{AB}\rangle = \sum_i \sqrt{p_i}\,|i_A\rangle \otimes |i_B\rangle$ we can calculate:

$$\begin{aligned}\mathrm{tr}_B(|\Psi_{AB}\rangle\langle\Psi_{AB}|) &= \mathrm{tr}_B\Big[\sum_i(\sqrt{p_i}\,|i_A\rangle\otimes|i_B\rangle)\sum_j(\sqrt{p_j}\,\langle j_A|\otimes\langle j_B|)\Big]\\ &= \mathrm{tr}_B\Big[\sum_{ij}\sqrt{p_ip_j}\,|i_A\rangle\langle j_A|\otimes|i_B\rangle\langle j_B|\Big] \qquad (3.3)\\ &= \sum_{ij}\delta_{ij}\sqrt{p_ip_j}\,|i_A\rangle\langle j_A| = \sum_i p_i\,|i_A\rangle\langle i_A|\end{aligned}$$

which is again our general ρ.

We can conclude that for a general mixed state we can trace over a simple copy of the systems Hilbert space. One has to note that neither the pure state nor the additional Hilbert space is unique.

3.4 Constructing the Neural Density Operator

The method of space purification to construct a mixed state with a neural network was first published by Torlai [31]. The network consists of N visible nodes σ representing the particles of the system with connections to αN hidden nodes h. Additionally, we will add another βN nodes for the auxiliary system a with connections only to the visible layer.

The network architecture is still a restricted Boltzmann machine with additional nodes along the hidden layer. The energy of the network is

$$E(\sigma, h, a, \mathscr{W}) = \sum_{i=1}^{N} b^i\sigma_i + \sum_{j=1}^{\beta N} d^j a_j + \sum_{k=1}^{\alpha N} c^k h_k + \sum_{i,j} a^j U^i_j \sigma_i + \sum_{i,k} h^k W^i_k \sigma_i, \quad (3.4)$$

with b^i being the bias for the visible spin σ_i and c^k, d^j being the biases for the auxiliary and hidden spins. W^i_k denotes the weight of the connection from the visible spin σ_i to the hidden spin h_k and U^i_j the connection's weight between the visible spin σ_i and the auxiliary spin a_j. All weights are assumed to be complex.

Similar to the NQS ansatz eq.(2.6) we calculate the pure state by summing over the hidden layer

$$\Psi(\sigma, a) = \langle\sigma, a|\Psi\rangle = \frac{1}{Z}\sum_{\{h\}}\exp(E(\sigma, h, a)). \quad (3.5)$$

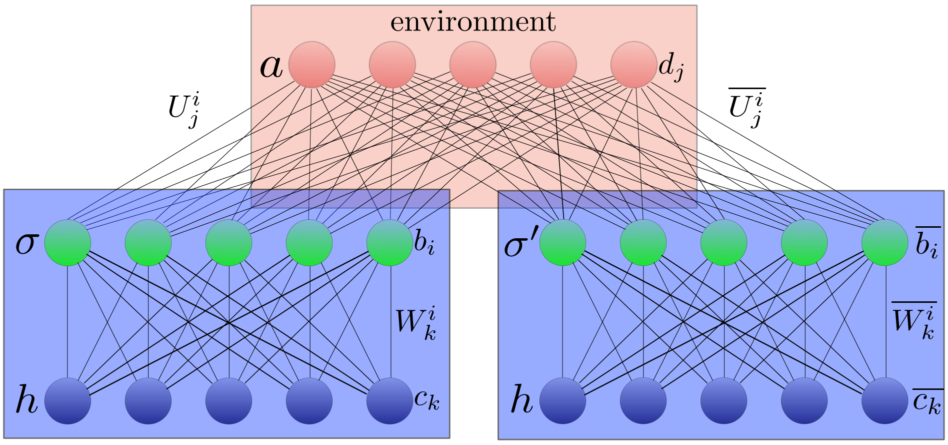

Figure 3.2: Two input layers both connected to the same environment layer. The system's hidden layer gets contracted to build $\Psi(\sigma, a)$. The environment is traced over afterwards.

Following the construction (3.3) we have to trace out the auxiliary system with a sum over all configurations of a,

$$\mathrm{tr}_A(|\Psi\rangle \langle\Psi|) = \sum_{\{a\}} \langle a| |\Psi\rangle \langle\Psi| |a\rangle . \tag{3.6}$$

An element of the density matrix $\rho(\sigma, \sigma')$ is then

$$\rho(\sigma, \sigma') = \sum_{\{a\}} \Psi(\sigma, a)\overline{\Psi(\sigma', a)}. \tag{3.7}$$

If we again concretize the system as binary, $\sigma, h, a \in \{-1, 1\}$, we can calculate the sums over all configurations explicitly

$$\rho(\sigma, \sigma') = \frac{1}{C} \; \tilde{\Psi}(\sigma) \; \overline{\tilde{\Psi}(\sigma')} \;\; \Lambda(\sigma, \sigma'), \tag{3.8}$$

where $\tilde{\Psi}(\sigma)$ is given by the original NQS ansatz

$$\tilde{\Psi}(\sigma) = \exp\left(\sum_i^N a^i \sigma_i\right) \cdot \prod_j^{\alpha N} 2\cosh\left(b_j + \sum_i^N W_j^i \sigma_i\right),$$

Λ is a matrix given by

$$\Lambda(\sigma, \sigma') = \prod_{j=1}^{\beta N} 2\text{cosh}\left(d_j + \bar{d}_j + \sum_i \left(U_j^i \sigma_i + \overline{U}_j^i \sigma'\right)\right)$$

and C is the normalisation factor, which we will not want to calculate explicitly,

$$C = \mathrm{tr}(\rho) = \sum_{\{\sigma\}} \rho(\sigma, \sigma).$$

We want to note some immediate properties of this approach:

- The calculation of a matrix element (dropping the normalisation) takes only products and sums over the number of nodes into account and not their configurations. Therefore the computing time grows polynomially in network size.
- There is a scheme given by Torlai [31] to Monte Carlo sample the expectancy value of some sparse operator $\hat{\mathcal{O}}$.
- The mixing of the system depends solely on the matrix Λ.

The reason to introduce the purification ansatz was the entropy limit (Schmidt rank) of the trivial one. We can make a neural network learn to optimise for maximal entropy. For this result, we only need N auxiliary spins and no additional hidden nodes. We can prove this by constructing an explicit neural density operator:

Choosing all biases to be zero ($b_i = 0, c_j = 0$) and the weight matrix $U = i\,\frac{\pi}{4}\mathbb{1}$, we can calculate:

$$\begin{aligned}\rho(\sigma,\sigma') = \Lambda(\sigma,\sigma') &= \prod_{j=1}^{N_a} 2\text{cosh}\Big(i\;\frac{\pi}{4}(\sigma_j - \sigma_j')\Big)\\ &= \begin{cases} 1 \text{ if } \sigma = \sigma' \\ 0 \text{ if } \exists j \text{ s.t. } \sigma_j \neq \sigma_j' \end{cases}\end{aligned} \tag{3.9}$$

The Schmidt rank of unity being maximal concludes the proof.

4 Implementation

In this chapter we want to specify the implementation and methods for the neural network. We start with the importance of the loss function and specify the one we use (§4.1). After introducing a scheme for the time evolution (§4.2) we close this chapter with the machine learning techniques used (§4.3).

4.1 Loss Function

The loss function is a measure of how good the neural network performs. In unsupervised learning (no labelled training data available) we want to directly minimize the loss function to obtain the optimal weights.

Solving the steady state problem, we want the loss function to satisfy the following properties:

- The loss function has to be a real, bounded function of the network's weights.
- The loss function obtains its global minimum only if the weights represent the desired state.

Some norm of $\frac{d\rho}{dt}$ is certainly capable to satisfy these properties. The norm chosen was the Euclidean norm

$$||A||_F^2 = \mathrm{tr}(A^\dagger A) = \sum_{ij} |A_{ij}|^2. \tag{4.1}$$

The gradient needed for optimisation can then be written down as the sum over the individual gradient of the matrix elements.

$$\nabla_{\mathscr{W}} \left\| \frac{d\rho}{dt} \right\|_F^2 = \sum_{\sigma\sigma'} \nabla_{\mathscr{W}} \left| \frac{d\rho(\sigma,\sigma')}{dt} \right|^2 \tag{4.2}$$

This enables a procedure called mini-batch learning as explained in §4.3.

4.2 Time Evolution

Another way of obtaining the steady state would be a time evolution of some initial state ρ_0 [1]. For a finite Liouvillian gap Δ_L (highest non-zero eigenvalue), every state not orthogonal to the steady states will evolve into a steady state.

Following the Lindblad master equation we know the time evolution of a given initial density operator

$$\frac{d\rho}{dt} = \mathscr{L}(\rho). \tag{4.3}$$

The time evolution for our system corresponds to the time evolution of our weights. Therefore we have to introduce time-dependent weights $\mathscr{W}(t)$ and their time derivatives $\dot{\mathscr{W}}(t)$.

With the chain rule we can calculate

$$\frac{d}{dt}\rho(\mathscr{W}) = \frac{\partial \rho}{\partial \mathscr{W}} \cdot \frac{d\mathscr{W}}{dt}. \tag{4.4}$$

We can evaluate the inner product from eq.(4.4)

$$\frac{\partial \rho}{\partial \mathscr{W}} \cdot \frac{d\mathscr{W}}{dt} = \sum_i \frac{\partial \rho}{\partial \mathscr{W}_i} \dot{\mathscr{W}}_i. \tag{4.5}$$

Tuning the time derivatives of the weights, we want to minimize some distance $\operatorname{dist}(\frac{d\rho}{dt}, [L](\rho))$. In general, $\operatorname{dist}(\cdot,\cdot)$ function needs the property of non-negativity ($\operatorname{dist}(x, y) \geq 0$, $\forall x, y$) and the identity of indiscernibles ($\operatorname{dist}(x, y) = 0 \Leftrightarrow x = y$).

We use the Euclidean norm as the measure of distance,

$$\operatorname{dist}(\rho, \rho') = \|\rho - \sigma\| = \sum_{\sigma\sigma'} \left| \langle \sigma | \rho - \rho' | \sigma' \rangle \right|^2. \tag{4.6}$$

Plugging in eq.(4.3) and eq.(4.5) we can now write the problem down as

$$\delta = \sum_{\sigma\sigma'} \left| \sum_i \frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_i} \dot{\mathscr{W}}_i - \mathscr{L}(\rho)_{\sigma\sigma'} \right|^2. \tag{4.7}$$

To find the minima of such a problem one can simply find the gradient of the function and set it to zero. The function is real, therefore we could calculate

[1] During work in progress, we became aware that Carleo and Hartmann already discussed time evolution of the Lindblad master equation with the same neural network ansatz [14].

either $\nabla_{\mathscr{W}}$ or the complex conjugate $\nabla_{\overline{\mathscr{W}}}$ [32] to find the minimum. To calculate the non-conjugated weight derivatives, we choose $\nabla_{\overline{\mathscr{W}}}$.

For the gradient of eq.(4.7) we can calculate

$$\frac{\partial}{\partial \overline{\dot{\mathscr{W}}}_j} \sum_{\sigma\sigma'} \left| \sum_i \frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_i} \dot{\mathscr{W}}_i - \mathscr{L}(\rho)_{\sigma\sigma'} \right|^2 = \sum_{\sigma\sigma'} \left(\sum_i \frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_i} \dot{\mathscr{W}}_i - \mathscr{L}(\rho)_{\sigma\sigma'} \right) \overline{\left(\frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_j} \right)} \tag{4.8}$$

If we set the gradient to zero we can formulate this as a linear equation,

$$A\dot{\mathscr{W}} = b \tag{4.9}$$

with

$$(A_i^j) = \sum_{\sigma\sigma'} \left(\overline{\left(\frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_i} \right)} \frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_j} \right) \text{ and } (b)_i = \sum_{\sigma\sigma'} \mathscr{L}(\rho)_{\sigma\sigma'} \overline{\left(\frac{\partial \rho_{\sigma\sigma'}}{\partial \mathscr{W}_i} \right)}. \tag{4.10}$$

We can now take the Moore-Penrose inverse [33] (a generalized inverse for matrices) of A to solve the equation and calculate

$$\dot{\mathscr{W}} = A^{-1}b. \tag{4.11}$$

We can integrate this differential equation via some of the usual algorithms like Runge-Kutta method [34].

This scheme tends to be unstable for large time steps, as the high non-linearity of the network induces large deviations for small parameter changes.

4.3 Mini Batch Learning

For large data sets - like the huge basis of our many body Hilbert space - it is more efficient to optimise the neural network with only chunks of the data set at once [35]. In machine learning this procedure is called mini-batch learning.

In general, we would like to choose a mini-batch with the most impact on the learning process. Choosing the samples and not taking them randomly is called active learning. We will sample from the probability distribution given by the diagonal elements of the density matrix $\rho(\sigma, \sigma)$. The intuition behind this method is that small populations should have only small effects in our Liouvillian time evolution. We use the samples to calculate all off-diagonal combinations as well.

Meaning that if we draw the set $\{\sigma\}$, we use it to calculate both sums $\sum_{\sigma,\sigma'}$ in (4.2). With M samples drawn, we will calculate the gradient with M^2 elements.

With this procedure we do not attempt to approximate the gradient. Instead, we approximate the direction of the gradient.

To choose samples we apply the Metropolis-Hastings algorithm[36]. We will use a simple spin flip - alternating through all particles - to choose candidates. To improve the algorithm we include a burn-in phase of $2N^2$ candidates to reduce finite chain size effects and use only every N^2th sample afterwards to reduce the correlations in between the samples (N is the particle number).

With Metropolis-Hastings, we only need the fraction $\frac{\rho(\sigma',\sigma')}{\rho(\sigma,\sigma)}$. The ratio can be calculated very efficiently for a simple spin flip. We only need to consider terms which get changed due to the flip.

4.4 Learning Algorithms

In order to minimize the loss function we need some kind of optimisation algorithm (often called learning).

Complex Gradient Descent

In general we will look at complex weights while our loss function is real. Instead of the usual complex derivative, we have to take the real gradient, considering the real and imaginary part as individual subspaces. This is equivalent to the complex conjugated Wirtinger derivative $\frac{\partial}{\partial\bar{z}} = \frac{1}{2}\big(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\big)$ [32] where x is the real part of z and y the imaginary. This derivative will act on z like a constant $\frac{\partial}{\partial\bar{z}}z = 0$.

With this complex gradient we can use standard gradient descent to reach the local minima. One update step will then be

$$\mathscr{W}^{i+1} = \mathscr{W}^{i} - \eta\ \nabla_{\overline{\mathscr{W}}}, L(\mathscr{W}), \tag{4.12}$$

where η is the step size or learning rate.

Resilient Back Propagation (Rprop) [37]

For highly non-linear functions the gradient may vary widely in magnitudes. Therefore it is hard to find a single learning rate for the global problem.

This method takes only the sign of the gradient into account and acts individually on each weight. Each weight has its own learning rate which increases if the sign of the gradient has the same sign as the one before and decreases otherwise.

Algorithm 1 Rprop Algorithm

```

for all $\mathscr{W}_i$ do

    if $(\nabla_{\mathscr{W}_i} L^{n+1} \cdot \nabla_{\mathscr{W}_i} L^n) > 0$ then        ▷ gradients have the same sign
        $\Delta\mathscr{W}_i^{n+1} = \min\{\Delta\mathscr{W}_i^n \cdot \gamma^+, \Delta\mathscr{W}_{max}\}$
        $\mathscr{W}^{n+1} = \mathscr{W}^n + \mathrm{sign}\{\nabla_{\mathscr{W}_i} L^{n+1}\}\Delta\mathscr{W}^{n+1}$        ▷ update weights

    else if $(\nabla_{\mathscr{W}_i} L^{n+1} \cdot \nabla_{\mathscr{W}_i} L^n) < 0$ then        ▷ gradient changed sign
        $\Delta\mathscr{W}_i^{n+1} = \max\{\Delta\mathscr{W}_i^n \cdot \gamma^-, \Delta\mathscr{W}_{min}\}$
        $\mathscr{W}^{n+1} = \mathscr{W}^n + \mathrm{sign}\{\nabla_{\mathscr{W}_i} L^{n+1}\}\Delta\mathscr{W}^{n+1}$
    else        ▷ if either one of the gradients are zero
        $\Delta\mathscr{W}_i^{n+1} = \Delta\mathscr{W}_i^n$
        $\mathscr{W}^{n+1} = \mathscr{W}^n + \mathrm{sign}\{\nabla_{\mathscr{W}_i} L^{n+1}\}\Delta\mathscr{W}^{n+1}$        ▷ sign{0}=0
```

In the algorithm we have $\Delta\mathscr{W}_{max}$ and $\Delta\mathscr{W}_{min}$ which are the maximal and minimal step. We also have to choose the increase and decrease factors γ^+ and γ^-. Good choices for those parameters are $(\Delta\mathscr{W}_{max} = 10, \Delta\mathscr{W}_{min} = 10^{-6}, \gamma^+ = 1.2, \gamma^- = 0.5)$ [38].

With this learning method, we can cross flat potential landscapes and saddle points quickly. This algorithm can be extended to complex numbers, simply by separating the real and complex update step (split complex BP algorithm [39]).

We will use this method to test the representative power. However, this method does not work well for mini-batches. The statistical fluctuations tend to decrease the learning rate to its minimum which results in the learning process getting stuck.

5 Testing the Representative Power of the Neural Network

As a toy model for the network we use the many-body Rydberg system introduced in §2.3. The Hamiltonian we will consider is

$$\mathscr{H} = \sum_{i=1}^{N} \Omega \sigma_x^i + \sum_{i=1}^{N} |r_i\rangle \langle r_i| \Delta + \sum_{<ij>} |r_i\rangle \langle r_j| \frac{J}{r_{ij}^6} \tag{5.1}$$

together with a spontaneous decay with rate γ, induced by the environment.

For further calculations we will fix $\Omega = 1$ as an energy scale, as well as the distance $r = 1$ between the particles. Further, we will not use sampling in this chapter, but the whole Hilbert space, to probe only the networks capabilities and dismiss the efficiency. We will use Rprop to learn the network and will not limit the learning time of the neural network to find the local minima. The efficiency of the learning process is only considered in the next chapter. If not stated otherwise, we remain with the same number of hidden and auxiliary nodes as particles.

For easier implementation, we will only consider the one-dimensional case with open boundary conditions. In principle, the geometry of the system is not encoded in the structure of the neural network. Therefore the dimensionality of our problem should be arbitrary.

5.1 Fidelity and Loss

While we can be sure to have found a steady state for $L(\mathscr{W}) = 0$ one cannot immediately make a statement for finite loss. Instead one uses the fidelity for mixed states as a measure of how close two states are.

For two pure states the fidelity is $F(|\Psi\rangle, |\Phi\rangle) = \left|\langle\Psi|\Phi\rangle\right|^2$. The generalisation for mixed states becomes

$$F(\rho, \sigma) = \mathrm{tr}\Big(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\Big)^2, \tag{5.2}$$

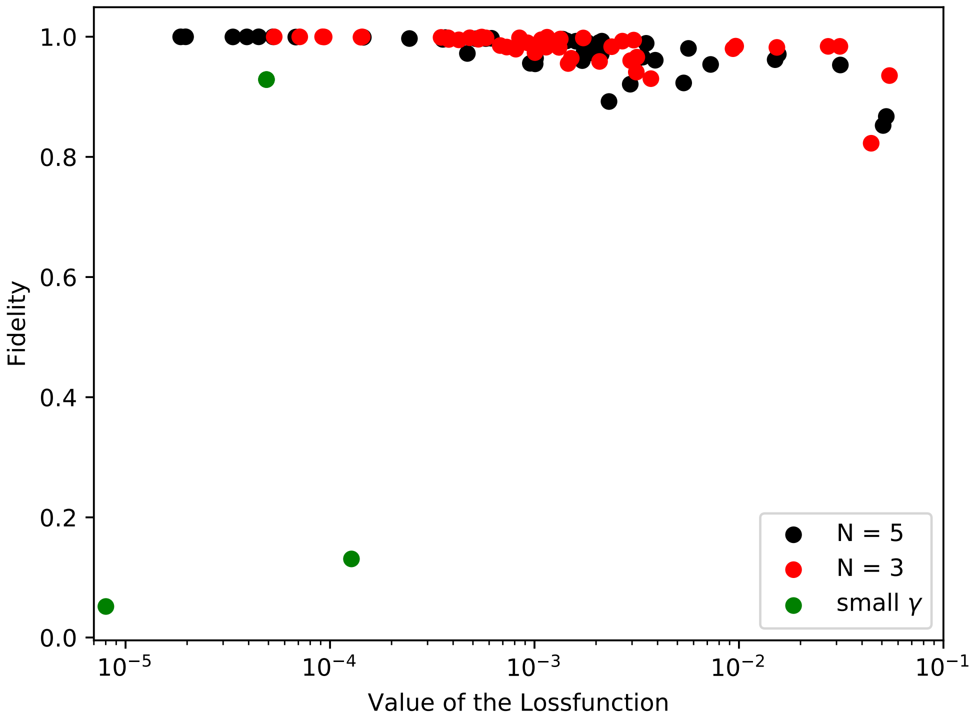

Figure 5.1: The value of our loss function $|L(\rho)|^2$ against the fidelity to the steady state. Various parameter with different learning times were chosen to get an overview.

where $\sqrt{\rho}$ is uniquely defined for hermitian matrices by diagonalisation [42]. For this chapter we will continue to look at the fidelity to benchmark the network. The exact steady state will be calculated with exact diagonalisation (Lanczos method) of the Liouvillian superoperator.

We scan through various values of Δ, J and γ, and compare the neural network results with the exact diagonalisation. We can observe that for most parameters, a small loss results in fidelity close to one. There is an exception for small γ where the fidelity drops. On the other hand, there are many points which show the opposite behaviour. The loss function is not very small while the fidelity is close to one. In §5.2 we want to explain both phenomenon in more detail.

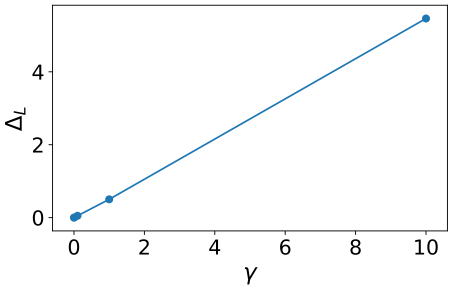

Figure 5.2: There is a linear relation for Δ_L over γ, so we can examine the Liouvillian gap by tweaking the decay rate

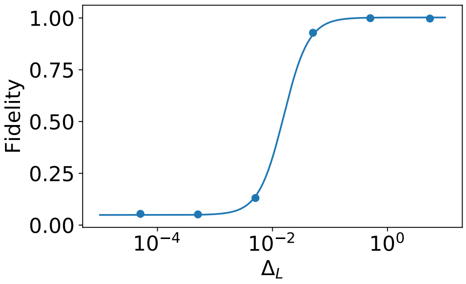

Figure 5.3: The network is not capable to capture the steady state for vanishing Liouvillian gap. Other parameters were $J = -1.0$, $\Delta = 1.0$ and $N = 5$

5.2 Liouvillian gap analysis

The Liouvillian gap Δ_L is the time scale to reach a steady state. By tweaking the decay rate γ, we can choose any gap size Δ_L (Figure 5.2).

For small γ we observed the special case of small loss with low fidelities (Figure 5.1). We can explain this phenomenon by examining our loss function. If we picture ρ as a vector in the operator space $\mathcal{H}_{Op} = \mathcal{H}_S \otimes \mathcal{H}_S$, then the Liouvillian superoperator is simply a matrix. We can write the loss function as a matrix vector product,

$$\|\mathscr{L}\rho\|^2 = \vec{\rho}^{\,\dagger} \left(\mathscr{L}^\dagger \mathscr{L}\right) \vec{\rho}. \tag{5.3}$$

$\mathscr{L}^\dagger\mathscr{L}$ is hermitian and diagonalisable. We can rewrite the loss function the following way

$$\|\mathscr{L}\rho\|^2 = \sum_{i=0}^{4^N} |\lambda_i|^2 \, |\langle \vec{\sigma_i}^\dagger, \vec{\rho}\rangle|^2, \tag{5.4}$$

where $\vec{\sigma_i}$ are the eigenstates of $\mathscr{L}$. We sort the eigenvalues such that $|\lambda_i| < |\lambda_{i+1}|$. Therefore $\lambda_0 = 0$ and σ_0 is the steady state. The first non-zero eigenstate is then σ_L corresponding to the Liouvillian gap eigenvalue.

The learning process tries to reduce the overlap of $\rho(\mathscr{W})$ with the eigenstates belonging to non-zero eigenvalues. For small Δ_L the overlap with σ_L will be

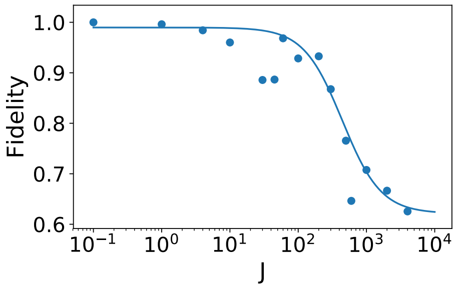

Figure 5.4: Decreasing Fidelity for increasing Interaction strength. Other parameters were $\Delta = 0.0$ and $\gamma = 1.0$

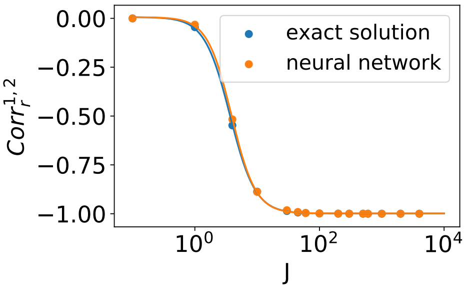

Figure 5.5: The relative correlation function is well captured, even though the fidelity decreases.

ignored in favour of reducing the overlap with other states. This also explains the opposite behaviour, we could observe in Figure 5.1. The loss function can be high even though the fidelity is close to one. The finite overlap then corresponds to very high eigenvalues.

We conclude, that a small Liouvillian gap does not only determines a long relaxation time but also stops the neural network from learning the steady state with our loss function. In our case the fidelity went under 90 % for $\gamma \approx 0.1\Omega$.

5.3 Van Der Waals Interaction

Now we scan through the strong but short ranged *van der Waals* interaction. The next to nearest neighbour interaction will be $\frac{1}{64}$ weaker. Even though the next to nearest neighbour term is often approximated as zero, we want to especially look into regimes, where this approximation does not hold.

We can observe a decline of fidelity for high interaction strength (Figure 5.4). We see, that strong particle coupling seems to exceed the limits of our representation.

First of all, the Liouvillian gap does not close for high interaction. It was also explicitly shown, that for the NQS ansatz, the entanglement entropy is not a limiting factor [28] and we constructed a network, fully entangled with the environment(§3.4).

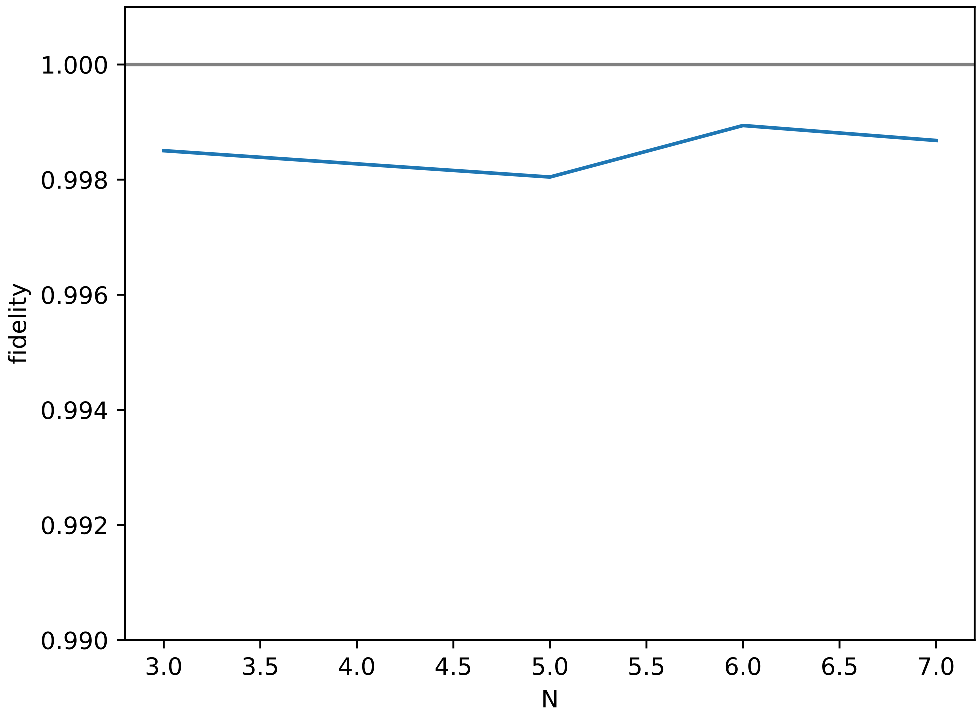

Figure 5.6: The fidelity reached for different particle numbers. The parameters were $J = 1$, $\Delta = -1$ and $\gamma = 1$. The deviations can be explained due to finite precision of the learning process.

For strong interaction, the network starts a blocking mechanism. One excited atom stops its neighbours from getting excited. To test if the network reproduces this behaviour, we calculate the relative correlation functions for the Rydberg populations:

$$\mathrm{Corr}_r^{ij} = \frac{\langle r_i r_j \rangle}{\langle r_i \rangle \langle r_j \rangle} - 1 \tag{5.5}$$

We can observe the effect of the blocking mechanism. Even though the network does not show great fidelity the correlation function is well captured ()Figure 5.5).

5.4 Particle scaling

One of the most interesting questions is the fidelity scaling with the particle number. For the hidden density and the auxiliary density

$$\alpha = \frac{\text{hidden nodes}}{\text{particle}}, \quad \beta = \frac{\text{auxiliary nodes}}{\text{particle}} \tag{5.6}$$

the parameters in the network consists of:

- N biases b_i for the visibles
- αN biases c_k for the hiddens
- αN^2 weights W_k^i for the connections between the hiddens and visibles
- βN biases and βN^2 weights for the biases and connection to the auxiliary system

In total we get

$$\dim(\mathscr{W}) = N(1 + \alpha + \beta + \alpha N + \beta N) = \mathcal{O}(N^2). \tag{5.7}$$

In comparison, the number of matrix elements scales with $\mathcal{O}(4^N)$, e.g. for 6 particles we already have 4096 matrix elements, while for $\alpha = 1$ and $\beta = 1$ we use only 90 weights.

We test the representative power for the non-trivial case of *facilitated excitations* [40]. Figure 5.6 shows the fidelity of the network representation for up to seven particles. The network shows good consistency in fidelity.

5.5 Effect of Network Density

At this point, we only considered a fully connected network with the same number of hidden and auxiliary nodes per particles. We examine the impact of more hidden units by raising the hidden density $\alpha = \frac{\text{hidden nodes}}{\text{particle}}$. We are especially interested in the case of strong interaction, where we showed that the network is struggling to represent the steady state.

For more hidden units, the higher dimensional manifold should be able to capture more features of our Quantum problem. For example, the network should be able to approximate more correlations[43].

The network is trained with the trick published by A.Borin et al. [43], to reduce the risk of getting stuck in a suboptimal minima. We observe that the fidelity is exponentially approaching some maximal value Figure 5.7. Especially for interaction dominated systems, we can improve the fidelity a lot. However, the exponential fit we obtained does not converge to 100%. Reason for this can be either, that the network is not capable of representing certain states at all or that the learning process is not capable of mapping the states to the network. Gao and Duan showed, RBMs are not capable to represent all pure states with a polynomial number of weights [27]. If all states can be represented with an exponential number of weights remains unclear.

On the other hand we looked at the case of too many weights given. We investigated the case of no interaction between the particles. In this trivial case, the problem gets reduced to a one body problem. Given a fully connected network, the network was often not capable to find the steady state. We had too many redundant local minima our network was able to get trapped in. We conclude that we can improve the network's capability by adding more hidden nodes but have to be careful not to overparameterise.

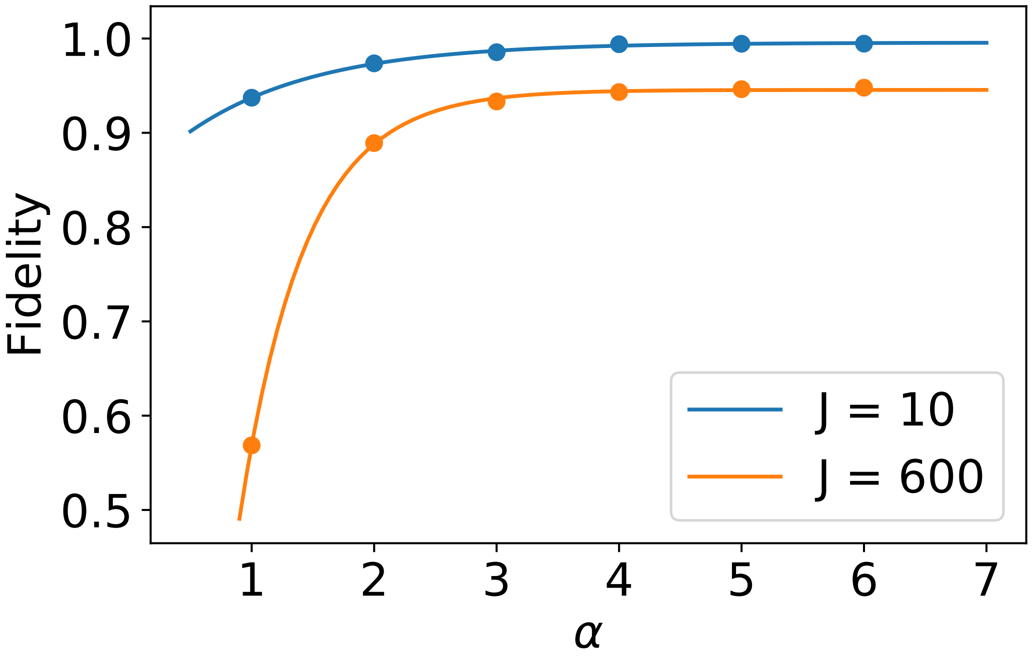

Figure 5.7: The fidelity to the exact steady state over the hidden density α. We fitted an exponential to the data points. We observed convergence to a maximal fidelity $F(\alpha \to \infty) = 99.5\%$ for $J = 10$ and $F(\alpha \to \infty) = 94.5\%$ for the dominating coupling $J = 600$. The calculation was made for five particles and $\Delta = 0$, $\gamma = 1$.

6 Efficiency of the Learning Process

In this chapter, we start to test the efficiency of the learning process. First, we look at the effect of different mini-batch sizes. After choosing an appropriate batch size, we investigate the computing time necessary to approximate steady states.

We will work again with the Rydberg toy model. The energy scale and distance stay fixed to one. We use *van der Waals* coupling with interaction strength $J = 1$, a detuning $\Delta = -1$ and a decay rate $\gamma = 1$. For those parameters, we get non-trivial behaviour but we know from the previous work that our network can express the steady state.

6.1 Choosing the Mini-Batch Size

We will start emphasizing the difference between active learning and the Monte-Carlo approximation. In the approach of Yoshioka et al.[13], they use the sampling to approximate the complete gradient. They use Metropole-Hastings to approximate the distribution

$$p(\sigma, \sigma') = \frac{1}{Z} |\rho(\sigma, \sigma')|^2 \tag{6.1}$$

Instead of calculating the whole gradient, we are only interested in a direction, which brings us closer to the optimal minima. We want the network to focus on minimizing the time derivatives of big populations. Very small sample sizes can be chosen.

If we choose the batch size too small for our problem, stochastic fluctuations will hinder the learning progress until it stops. On the other hand, we want to choose a small batch size to make the process as efficient as possible.

In Figure 6.1 we show an example for loss curves with different batch sizes n_B. We can observe the typical stop in learning for $n_B = 1$. Surprisingly we are able to get good results even for very small batch sizes like $n_B = 2$.

We will continue to look at the very small batch sizes $n_B \in \{2, 4\}$. Keep in

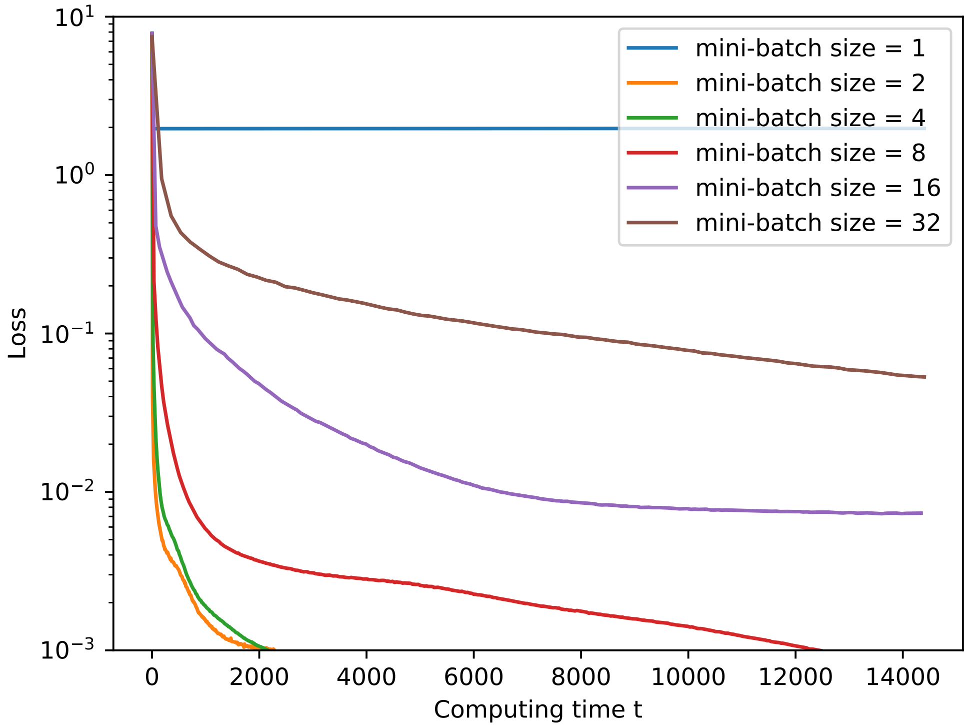

Figure 6.1: Exemplary loss curve over computation time for different batch sizes. The calculation was for seven particles.

mind, that we use the samples to calculate not only the diagonal elements but the corresponding off-diagonal ones as well. Therefore the computation time for each update increases quadratically in batch size.

6.2 Scaling with the particle number

The learning process with the mini-batches is very stochastic. We will continue to present the mean values of the computation time together with their statistical error. We have to mention that the learning process often got stuck and we had to restart the measurement[1]. We exclude those runs from the data but will use them later to estimate the reliability.

We took 25 runs for both batch sizes and the different particle numbers (Figure 6.2,6.3). With the finite amount of data available, we assume an scaling

[1] We restart the calculation after four hours, assuming that the learning process got stuck.

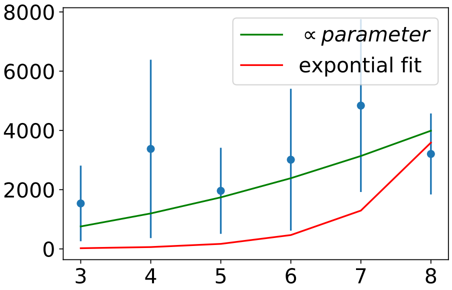

Figure 6.2: $n_B = 2$

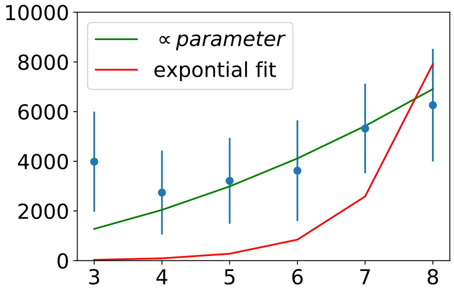

Figure 6.3: $n_B = 4$

with number of parameter. For the densities $\alpha, \beta = 1$, equation (5.7) gives a scaling for the computation time t_C with

$$t_C(N) \propto \frac{3}{2}N + N^2. \tag{6.2}$$

With the stochastic data available, we cannot exclude the possibility of an exponential scaling. We can observe a steeper ascent for a bigger batch size, supporting the argument that the smallest batch size possible will also yield the most efficient learning process.

6.3 Stochastics of the learning process

One could already observe the huge variations in computing time for the individual runs. Here we examine if different batch size have an effect of limiting the stochastic behaviour and produce more reliable results.

First off, for the batch size $n_B = 4$ we had to restart the learning process even more often than for $n_B = 2$, so we could not observe an improvement in this regard. The smaller batch size was more capable to retrieve the process out of a suboptimal learning process. Probably the stochastic fluctuations helped to move the process out of an local minima. Another reason for this behaviour could also be the smaller computation time for the small mini-batch in general.

To quantify the statistical behaviour, we calculate the relative statistical error for the individual particle number.

	Particle number	3	4	5	6	7	8
$n_B = 2$	restarts	0%	12%	28%	16%	12%	8%
	rel. error	0.832	0.892	0.741	0.795	0.603	0.426
$n_B = 4$	restarts	20%	28%	20%	24%	4%	20%
	rel. error	0.505	0.616	0.538	0.560	0.339	0.362

We see that for $n_B = 4$, the statistical error is getting smaller. We also observe smaller relative errors for higher particle numbers. More data points with more particles are needed to verify this behaviour.

7 Summary and Conclusion

We presented a variational neural network ansatz to solve the Lindblad master equation. This ansatz is hermitian and semi-positive. We proved with an explicit network, that the ansatz is capable of representing a maximally entangled state. Therefore the approach is not limited by area law as compared to tensor networks.

We were able to show that the variational neural network ansatz is capable to represent the experimentally interesting weak or medium interacting Rydberg system. Analogue to regular time evolution we found that a small Liouvillian gap limits us in finding the steady state. We observed that strong coupling is hindering the representation from working efficiently. We showed that one can improve the network's performance by adding additional hidden nodes. At the same time a bigger parameter space increases the danger of getting stuck in a suboptimal minimum. In the investigated non-trivial regime parameter space grows only quadratically with the particle number.

The learning process with active learning is very stochastic. Quite often the process gets stuck and has to be abandoned. Raising the mini-batch size does not limit this risk, but helps decreasing the stochastic fluctuations. The sampling process will be harder for more particles, as the Hilbert space still grows exponentially. We could not detect a need to raise the number of samples drawn. Nevertheless, the challenge to find the best-suited samples for learning will get harder which could lead to more samples necessary. The data available points to a scaling with the number of weights. This would mean that as long as the steady state can be represented efficiently we can also find it efficiently.

In this thesis we showed that the machine learning approach is limited. Inside those limits, however, the results are promising.

8 Outlook

The structure of our network is not dependent on the geometry of the system. While tensor networks are only feasible for one dimension, the neural network approach should not be troubled by higher dimensions. At the same time, the fully connected network is highly non-local. The network is therefore not limited to short ranged interactions either. Further work has to be done, to prove these predictions.

This thesis was very limited in particle numbers. Following the reasoning "the more, the better", one wants to observe the behaviour of the network for more particles. At this moment, the data available is not capable of showing if the scaling is as efficient as we hope or not.

One of the biggest problems we see is the reliability of the learning process. Following the idea of adaptive learning rates, changing the batch size during the learning process could increase efficiency and stability.

To reduce the risk of getting stuck in suboptimal minima, we could adaptively increase the parameter space. In this way we would not overparameterize but still get the necessary amount of weights to represent the state.

The active learning approach works best if the samples drawn are the most suitable for reducing the loss. In this thesis we just worked with the basic idea that big populations contribute the most to the time evolution. We do not think that the sampling process is optimal for active learning. In many-body physics, active learning is widely unexplored. The presented approach offers a rich field of study, worth looking into more thoroughly.

Bibliography

[1] Darin Edward Acosta, Bobby Scurlock, Jia Fu Low, Jamal Tildon Rorie, Ivan-Kresimir Furic, Elena Laura Busch, Khristian Kotov, Andrew Wilson Brinkerhoff, Andrew Mathew Carnes, Wei Shi, Sergei Gleyzer, and Alexander Madorsky. Boosted Decision Trees in the Level-1 Muon Endcap Trigger at CMS. In *Proceedings of Science*, page 143, oct 2017.

[2] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks—a review. *Pattern Recognition*, 35(10):2279–2301, oct 2002.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, may 2017.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, nov 2012.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Miguel Pozuelo Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Junbo Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *CoRR*, 1604.07316, 2016.

[6] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313(5786):504–7, jul 2006.

[7] Giuseppe Carleo and Matthias Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602–606, jun 2016.

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

[9] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix Product State Representations. *Quantum information & computation*, 7(5):401–430, aug 2007.

[10] A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero. Positive Tensor Network Approach for Simulating Open Quantum Many-Body Systems. *Physical Review Letters*, 116(23):237201, jun 2016.

[11] Augustine Kshetrimayum, Hendrik Weimer, and Román Orús. A simple tensor network algorithm for two-dimensional steady states. *Nature Communications*, 8(1):1291, dec 2017.

[12] M Marcuzzi, J Schick, B Olmos, and I Lesanovsky. Effective dynamics of strongly dissipative Rydberg gases. *Journal of Physics A: Mathematical and Theoretical*, 47(48):482001, dec 2014.

[13] Nobuyuki Yoshioka and Ryusuke Hamazaki. Constructing neural stationary states for open quantum many-body systems. feb 2019.

[14] Michael J. Hartmann and Giuseppe Carleo. Neural-Network Approach to Dissipative Quantum Many-Body Dynamics. feb 2019.

[15] Alexandra Nagy and Vincenzo Savona. Variational quantum Monte Carlo with neural network ansatz for open quantum systems. 2019.

[16] Filippo Vicentini, Alberto Biella, Nicolas Regnault, and Cristiano Ciuti. Variational neural network ansatz for steady states in open quantum systems. 2019.

[17] Heinz-Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, jan 2007.

[18] W. Weidlich and F. Haake. Coherence-properties of the statistical operator in a laser model. *Zeitschrift fuer Physik*, 185(1):30–47, feb 1965.

[19] Dariusz Chruściński and Saverio Pascazio. A Brief History of the GKLS Equation. *Open Systems & Information Dynamics*, 24(03):1740001, oct 2017.

[20] Elena del Valle Reboul. *Quantum Electrodynamics with Quantum Dots in Microcavities*. PhD thesis, 2009.

[21] Ángel Rivas and Susana F. Huelga. *Open Quantum Systems. An Introduction*. Springer Berlin Heidelberg, apr 2011.

[22] Fabrizio Minganti, Alberto Biella, Nicola Bartolo, and Cristiano Ciuti. Spectral theory of Liouvillians for dissipative phase transitions. *Physical Review A*, 98(4):042118, oct 2018.

[23] Robert Löw, Hendrik Weimer, Johannes Nipper, Jonathan B Balewski, Björn Butscher, Hans Peter Büchler, and Tilman Pfau. An experimental and theoretical guide to strongly interacting Rydberg gases. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 45(11):113001, jun 2012.

[24] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313(5786):504–507, jul 2006.

[25] Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14 th International Con- ference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[26] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, may 2018.

[27] Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature Communications*, 8(1):662, dec 2017.

[28] Dong-Ling Deng, Xiaopeng Li, and S. Das Sarma. Quantum Entanglement in Neural Network States. *Physical Review X*, 7(2):021021, may 2017.

[29] Michael Fleischhauer, Atac Imamoglu, and Jonathan P. Marangos. Electromagnetically induced transparency: Optics in coherent media. *Reviews of Modern Physics*, 77(2):633–673, jul 2005.

[30] Purification of quantum state - Wikipedia, 2018.

[31] Giacomo Torlai and Roger G. Melko. Latent Space Purification via Neural Density Operators. *Physical Review Letters*, 120(24):240503, jun 2018.

[32] D.H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proceedings H Microwaves, Optics and Antennas*, 130(1):11, 1983.

[33] Roger Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[34] J.C. Butcher. Numerical methods for ordinary differential equations in the 20th century. *Journal of Computational and Applied Mathematics*, 125(1-2):1–29, dec 2000.

[35] Sebastian Ruder. An overview of gradient descent optimization algorithms. sep 2016.

[36] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[37] M Riedmiller and H Braun. Rprop - A Fast Adaptive Learning Algorithm. In *Proceedings of the International Symposium on Computer and Information Science VII*, 1992.

[38] Christian Igcl, Michael Hiiskcn, and Christian Igel. Improving the Rprop Learning Algorithm. *Proceedings of the Second International Symposium on Neura l Computation*, pages 115–121, 2000.

[39] A. Kantsila, M. Lehtokangas, and J. Saarinen. Complex RPROP-algorithm for neural network equalization of GSM data bursts. *Neurocomputing*, 61:339–360, oct 2004.

[40] David W. Schönleber, Martin Gärttner, and Jörg Evers. Coherent versus incoherent excitation dynamics in dissipative many-body Rydberg systems. *Physical Review A*, 89(3):033421, mar 2014.

[41] H. Schempp, G. Günter, M. Robert-de Saint-Vincent, C. S. Hofmann, D. Breyel, A. Komnik, D. W. Schönleber, M. Gärttner, J. Evers, S. Whitlock, and M. Weidemüller. Full Counting Statistics of Laser Excited Rydberg Aggregates in a One-Dimensional Geometry. *Physical Review Letters*, 112(1):013002, jan 2014.

[42] Richard Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41(12):2315–2323, dec 1994.

[43] Artem Borin and Dmitry A. Abanin. Approximating power of machine-learning ansatz for quantum many-body states. jan 2019.

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 666.666.666, ...
(Unterschrift)