TECHNISCHE UNIVERSITÄT DRESDEN

Title of Bachelor thesis

Bachelor-Arbeit
zur Erlangung des Hochschulgrades
Bachelor of Science
im Bachelor-Studiengang Physik

vorgelegt von

Felix Soest
geboren am 16.09.1998 in Düsseldorf

Institut für Theoretische Physik
Fakultät Physik
Bereich Mathematik und Naturwissenschaften
Technische Universität Dresden
2021

Eingereicht am xx. Monat 20xx

1. Gutachter: Prof. Dr. Walter Strunz
2. Gutachter: Prof. Dr. Oscar Dahlsten

Summary

Abstract

English:

Aufmerksamkeit große Fragen? unsere Frage unsere Antwort

Abstract

Deutsch

Contents

1. Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

1.1. Intro

Test 123

2. Background

2.1. Supervised Machine Learning

Machine learning is a subfield of artificial intelligence, ‘concerned with the question of how to construct computer programs that automatically improve with experience.’ [6] Supervised machine learning is one of the three machine learning disciplines, besides unsupervised and reinforcement learning. The goal is to find a mapping between an input and an output, in our case an excitation and its respective optimal harvesting policy. Multiple algorithms to find such a mapping exist, however for high dimensional problems artificial neural networks (ANNs) are usually used. In this section we review ANNs, following the exposition given in [4].

Let $\mathfrak{R}$ be a fully-connected feedforward ANN, meaning there are no loops in the neuron connections and all neurons in a layer are connected to every neuron of the next layer, $\mathfrak{R} : \mathbb{R}^{n_1} \to \mathbb{R}^{n_L}$. n_1 and n_L denote the dimensionality of the input and output respectively. $\mathfrak{R}$ has L layers, or columns of neurons. The network architecture is given by the amount of neurons n_l in each hidden layer $l \in [2, L-1]$ (see figure 2.1). The neurons in layer l are represented by their activations $\vec{a}_l \in \mathbb{R}^{n_l}$, which represent the matrix multiplication output. Additionally each layer includes trainable parameters $W_l \in \mathbb{R}^{n_{l+1} \times n_l}$ and $\vec{b}_l \in \mathbb{R}^{n_l}$ called weights and biases. The activations can then be calculated using the following formulae [10]:

$$\begin{aligned} \vec{a}_2 &= W_1 \vec{a}_1 + \vec{b}_1, \\ \vec{a}_l &= W_{l-1} \xi(\vec{a}_{l-1}) + \vec{b}_{l-1}, \ l \in [3, L], \end{aligned}$$

where $\xi(x)$ is a function called the activation function applied elementwise. Historically, functions such as *tanh* and sigmoid have been used. However, it has been shown [5, 2] that the rectified linear unit $\text{ReLU}(x) = \max(0, x)$ often provides better results and is used here.

To train an ANN a cost function is defined, often the mean squared error

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\vec{a}_{L,i} - \vec{y}_i)^2,$$

where the summation is performed over the training data $\{(\vec{x}_i, \vec{y}_i)\}$ with N samples, where $\{\vec{x}_i\}$ is the input and $\{\vec{y}_i\}$ the output data, and $\vec{a}_{L,i} = \mathfrak{R}(\vec{x}_i)$ is the output of the neural network. The so-called backpropagation algorithm is used to calculate the gradient of the cost function with respect to the trainable parameters and improve the performance of the ANN [8, 7].

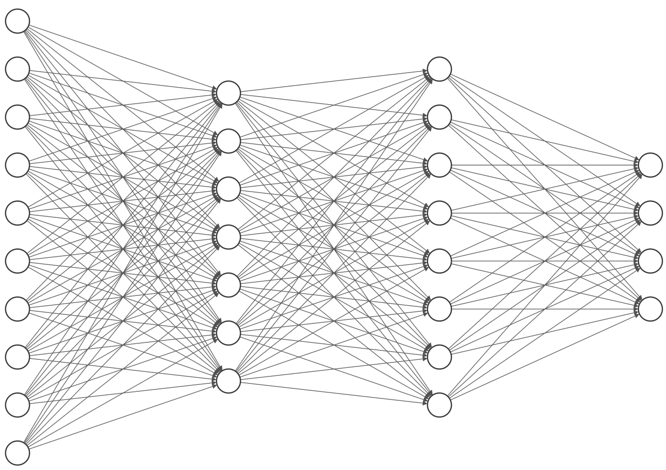

Figure 2.1.: Example fully-connected feedforward ANN with four layers, including input, output and two hidden layers [3]

.

2.2. Collision model dynamics

2.3. Setting

Our setting consists of three qubits: the Drive, System and Transducer qubits. The Drive and Transducer qubits can be set by the experimenter in N discrete steps modelled as piecewise constant functions (PWC) of (θ_D, ϕ_D) and (θ_T, ϕ_T) respectively (see figure 2.2), the system qubit is initialised in a pure state. In general, unitary evolution of a multipartite system will lead to entanglement, meaning Drive and Transducer bits are no longer pure states. This is at odds with the assumption of piecewise constant control functions. We therefore model Drive and Transducer qubits as series of ancilla qubits (figure 2.3) which interact with the system such that the state does not entangle and can afterwards be measured [1]. In the remainder of this work we use the interaction Hamiltonian on the three qubit Hilbert space

$$H_{DST} = H_I \otimes \mathbb{1}_T + \mathbb{1}_D \otimes H_I, \qquad H_I = \sigma_+ \otimes \sigma_- + \sigma_- \otimes \sigma_+$$

unless otherwise noted. The time evolution and work extraction is then calculated as follows, where Δt is time span between qubit switching:

$$H_S^i = \langle \psi_D^i | \langle \psi_T^i | H_{DST} | \psi_D^i \rangle | \psi_T^i \rangle \tag{2.1}$$

$$\rho_S^{i+1} = U^i \rho_S^i U^{i\dagger}, \; U^i = e^{-iH_S^i \Delta t} \tag{2.2}$$

$$W = -\Sigma_i \mathrm{Tr}\; \rho_S^i \; dH_S^i \tag{2.3}$$

$$dH_S^i = \langle \psi_D^i | \langle \psi_T^{i+1} | H_{DST} | \psi_D^i \rangle | \psi_T^{i+1} \rangle - \langle \psi_D^i | \langle \psi_T^i | H_{DST} | \psi_D^i \rangle | \psi_T^i \rangle . \tag{2.4}$$

Here we use the partial Hamiltonian H_S^i on S at time step $i \in [1, N-1]$, as well as corresponding system density matrix ρ_S^i.

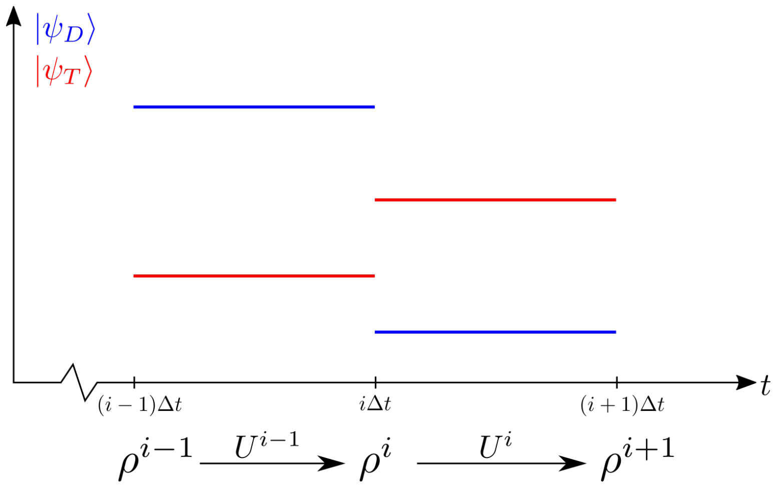

Figure 2.2.: Piecewise constant implementation of Drive and Transducer qubits: the vertical axis shows qubit state in arbitrary units. The qubit states are switched instantaneously and then kept constant for Δt while ρ_S evolves unitarily.

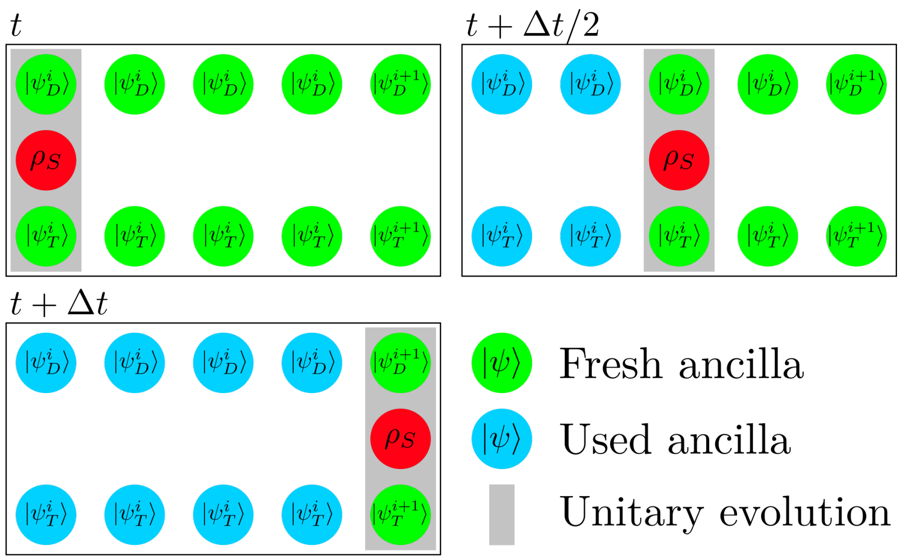

Figure 2.3.: Collision model used in this work: Drive and Transducer are series of qubits interact once with the system and evolve the reduced density operator ρ_S. The qubit configuration can be changed in intervals of Δt.

3. Experimental Results

The training data is created using a minimisation algorithm [9], which finds the optimal Transducer protocol $\{|\psi_T^i\rangle\}$ given a Drive sequence $\{|\psi_D^i\rangle\}$. The networks are trained to learn the mapping $\{|\psi_D^i\rangle\} \rightarrow \{|\psi_T^i\rangle\}$. Both the input (Drive) and output (Transducer) are transformed by the embedding

$$\left\{\begin{pmatrix}\theta^i & \phi^i\end{pmatrix}\right\} \rightarrow \left\{\begin{pmatrix}sin(\theta^i) & cos(\theta^i) & sin(\phi^i) & cos(\phi^i)\end{pmatrix}\right\}.$$

The reasons for this operation are twofold: it normalises the data to the interval $[-1, 1]$. Additionally it adds information regarding the periodicity of the qubit angle representation.

To compare the accuracy of different models a performance indicator is required. Naturally one might use the MSE as introduced in section 2.1. Instead we define the *efficiency* of a model $\mathfrak{R}$ on a dataset $\{(\vec{x}_i, \vec{y}_i)\}$ as

$$\eta = \frac{1}{N}\sum_{i=1}^{N}\frac{W(\vec{x}_i, \mathfrak{R}(\vec{x}_i))}{W(\vec{x}_i, \vec{y}_i)}, \tag{3.1}$$

i.e. the arithmetic mean of the ratios of work output predicted by the model to optimal work output. The function $W(\vec{x}_i, \vec{y}_i) = W(\{|\psi_D\rangle\}_i, \{|\psi_T\rangle\}_i)$ returns the work given a Drive and Transducer protocol.

3.1. Dependence of Δt on Work Output

We start our investigation by determining the work output W when varying the time between qubit switching Δt. If the System qubit is initialised in the pure state $\rho_S = |0\rangle\langle 0|$, the work output for a single jump is given by (see appendix **??** for a derivation)

$$W = \frac{1}{|\alpha|}sin(2|\alpha|\Delta t)\,\mathrm{Im}\{(\tau' - \tau)\alpha^*\} \tag{3.2}$$

$$\alpha = \frac{1}{2}\left[sin(\theta_D^1)e^{i\phi_D^1} + sin(\theta_T^1)e^{i\phi_T^1}\right], \qquad \tau' - \tau = \frac{1}{2}\left[sin(\theta_T^2)e^{i\phi_T^2} - sin(\theta_T^1)e^{i\phi_T^1}\right].$$

From equation 3.2 it becomes evident that for $\Delta t \rightarrow 0, W \rightarrow 0$. This is confirmed by the following where, for multiple values of N, we simulate 500 random Drive functions for each Δt

and find their optimal Transducer policy. The average work output over the 500 runs scaled by the amount of PWC steps $\overline{W}/N$ for 20 values of Δt is shown in figure 3.1.

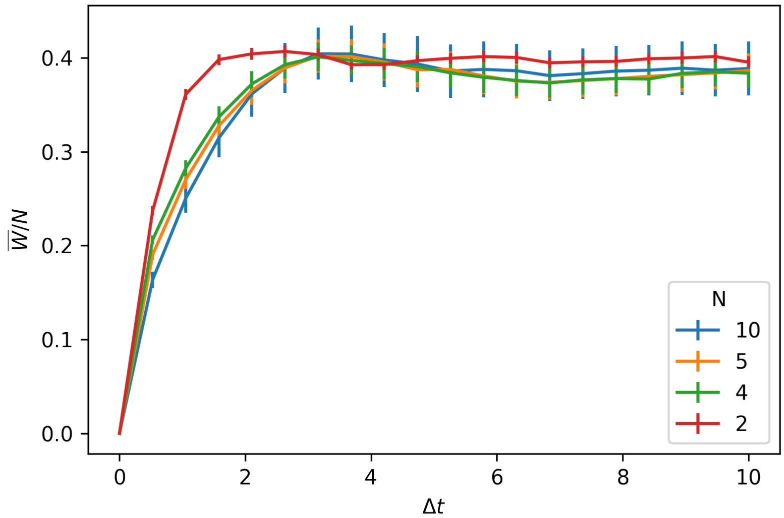

Figure 3.1.: We plot the average work $\overline{W}$ over 500 runs for random excitations divided by amount of PWC steps N, for all of which we use $\rho_0 = |0\rangle\langle 0|$. The error bars correspond to the standard deviation $\sigma_{\overline{W}}$.

3.2. $N = 2$: Learning Single Jump Optimal Control Sequences

For the simplest case of $N = 2$, we generate a data set of size $N_{data} = 50000$ with $\rho_0 = |0\rangle\langle 0|$, which we then separate into train and test data.

4. Summary and Outlook

A. Derivations

A.1. Single jump work output

$$
\begin{aligned}
H_S(\theta_D(t),\phi_D(t),\theta_T(t),\phi_T(t)) &= \frac{1}{2}\left[\sin(\theta_D(t))e^{i\phi_D(t)}+\sin(\theta_T(t))e^{i\phi_T(t)}\right]\sigma_+ + h.c.\\
&= \alpha\sigma_+ + h.c.\\
dH = H_S(\theta_D(t),\phi_D(t),\theta_T(t+1),\phi_T(t+1)) &- H_S(\theta_D(t),\phi_D(t),\theta_T(t),\phi_T(t))\\
= \frac{1}{2}(\sin(\theta_T(t+1))e^{i\phi_T(t+1)} &- \sin(\theta_T(t))e^{i\phi_T(t)})\sigma_+ + h.c. =: (\tau'-\tau)\sigma_+ + h.c.\\
U = e^{-iH_S\Delta t} = exp\begin{pmatrix} 0 & -i\alpha^*\Delta t\\ -i\alpha\Delta t & 0\end{pmatrix} &= \frac{1}{2}\begin{pmatrix}\frac{\alpha^*}{|\alpha|} & -\frac{\alpha^*}{|\alpha|}\\ 1 & 1\end{pmatrix}\begin{pmatrix} e^{-i|\alpha|\Delta t} & 0\\ 0 & e^{i|\alpha|\Delta t}\end{pmatrix}\begin{pmatrix}\frac{|\alpha|}{\alpha^*} & 1\\ -\frac{|\alpha|}{\alpha^*} & 1\end{pmatrix}\\
&= \begin{pmatrix}\cos(|\alpha|\Delta t) & -i\frac{\alpha^*}{|\alpha|}\sin(|\alpha|\Delta t)\\ -i\frac{|\alpha|}{\alpha^*}\sin(|\alpha|\Delta t) & \cos(|\alpha|\Delta t)\end{pmatrix}\\
&\text{With } \left|\psi_0\right\rangle = a\left|0\right\rangle + b\left|1\right\rangle, |a|^2+|b|^2 = 1, \text{ we have}\\
&\rho_0 = \begin{pmatrix}|a|^2 & ab^*\\ a^*b & |b|^2\end{pmatrix},\ \rho = U\rho_0U^\dagger =
\end{aligned}
$$

$$
\begin{pmatrix}
|a|^2\ \cos^2(|\alpha|\Delta t)+|b|^2\ \sin^2(|\alpha|\Delta t)-\frac{1}{|\alpha|}\sin(2|\alpha|\Delta t)\,\mathrm{Im}\{ab^*\alpha\} & \frac{\alpha^*}{2|\alpha|}i\ \sin(2|\alpha|\Delta t)(|a|^2-|b|^2)+\frac{\alpha^*}{\alpha}a^*b\sin^2(|\alpha|\Delta t)+ab^*\ \cos^2(|\alpha|\Delta t)\\
\frac{|\alpha|}{2\alpha^*}i\ \sin(2|\alpha|\Delta t)(|b|^2-|a|^2)+\frac{\alpha}{\alpha^*}ab^*\ \sin^2(|\alpha|\Delta t)+a^*b\ \cos^2(|\alpha|\Delta t) & |a|^2\ \sin^2(|\alpha|\Delta t)+|b|^2\ \cos^2(|\alpha|\Delta t)+\frac{1}{|\alpha|}\sin(2|\alpha|\Delta t)\,\mathrm{Im}\{ab^*\alpha\}
\end{pmatrix}
$$

$$
\begin{aligned}
\mathrm{dW} = -\mathrm{Tr}\ \rho\ \mathrm{dH} &= \frac{|a|^2-|b|^2}{|\alpha|}\sin(2|\alpha|\Delta t)\,\mathrm{Im}\{(\tau'-\tau)\alpha^*\}\\
&+\frac{1}{2}[\cos^2(|\alpha|\Delta t)\,\mathrm{Re}\{(\tau'-\tau)ab^*\}+\sin^2(|\alpha|\Delta t)\,\mathrm{Re}\left\{(\tau'-\tau)a^*b\frac{\alpha^*}{\alpha}\right\}]
\end{aligned}
$$

B. Bibliography

[1] Konstantin Beyer, Kimmo Luoma, and Walter T. Strunz. Work as an external quantum observable and an operational quantum work fluctuation theorem.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[3] Alexander LeNail. Nn-svg: Publication-ready neural network architecture schematics. *Journal of Open Source Software*, 4(33):747, 2019.

[4] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples, 2020.

[5] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.

[6] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[7] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[8] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[9] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[10] Andreas Zell. *Simulation neuronaler Netze*. Oldenbourg, München, 2., unveränd. nachdr. edition, 1997.

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit im Rahmen der Betreuung am Institut für Theoretische Physik ohne unzulässige Hilfe Dritter verfasst und alle Quellen als solche gekennzeichnet habe.

Felix Soest
Dresden, Monat 2021