Influence of Noise on Quantum Variational Algorithms

Philip Hierhager

München 2022

Influence of Noise on Quantum Variational Algorithms

Philip Hierhager

Bachelor's Thesis
at the Faculty of Physics
at Ludwig-Maximilians-Universität
Munich

submitted by
Philip Hierhager
from Munich

München, den 08. April 2022

Abstract

The goal of this work is the characterization of the influence of noise on quantum variational algorithms. The latter is used e.g. in quantum machine learning. Quantum machine learning tries to combine the promises of quantum computing and the benefits of machine learning. At Fraunhofer Institute of Cognitive Systems, this approach is for example used to investigate how to improve tumor detection. In this work, the impact of noise on the training process of quantum machine learning is evaluated. A superconducting quantum computer is used. This quantum device uses electronic circuits to realize the concept of quantum computers in practice and suffers from two kinds of noise. Systematic noise results from miscalibration of operations. Stochastic noise occurs through random errors. Since quantum computers are open quantum systems, they can randomly couple to the environment due to temperature. Also, the electronic circuit elements themselves possess voltage and current fluctuations. Because quantum computers are not yet fault-tolerant, noise is influencing the results. To characterize this influence and be able to mitigate it is therefore crucial for the success of the method. General barren plateaus - which affect the training negatively - could not be shown. However, experiments indicate the arise of noise-induced barren plateaus with more complex structures. But noise seems to have a positive effect as well. In the experiments conducted it was shown, that it can help to speed up the convergence to high accuracy in the training process.

Einfluss von Noise auf Quantum Variational Algorithmen

Philip Hierhager

Bachelorarbeit
der Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Philip Hierhager
aus München

München, den 08. April 2022

Abstract

Das Ziel dieser Arbeit ist die Charakterisierung des Einflusses von Noise auf Quantum Variational Algorithmen. Letztere werden z.B. beim Quantum Machine Learning eingesetzt. Dieses versucht, die Vorteile des Quantencomputings und die des maschinellen Lernens zu kombinieren. Am Fraunhofer Institut für Kognitive Systeme wird dieser Ansatz zum Beispiel genutzt, um Tumorerkennung in medizinischen Bildern zu verbessern. In dieser Arbeit wird der Einfluss des Noise auf den Trainingsprozess des Quantum Machine Learning untersucht. Es wird ein supraleitender Quantencomputer verwendet. Dieser nutzt elektronische Schaltungen zur Realisierung des Konzeptes und leidet unter zwei Arten von Noise. Systematic Noise entsteht durch die Fehlkalibrierung von Operationen. Stochastic Noise entsteht durch zufällige Fehler. Da es sich bei Quantencomputern um offene Quantensysteme handelt, können sie beispielsweise über die Temperatur mit der Umgebung interagieren. Auch die elektronischen Schaltkreiselemente selbst weisen Spannungs- und Stromschwankungen auf. Da Quantencomputer noch nicht fehlertolerant sind, beeinflusst Noise die Ergebnisse. Diesen Einfluss zu charakterisieren und möglicherweise abschwächen zu können ist daher entscheidend für den Erfolg der Methode. Allgemeine Barren Plateaus - die das Training negativ beeinflussen - konnten nicht gezeigt werden. Allerdings deuten Experimente auf das Auftreten von noise-induced Barren Plateaus bei komplexeren Strukturen hin. Noise scheint aber auch eine positive Wirkung zu haben. In den durchgeführten Experimenten wurde gezeigt, dass der Trainingsprozess des verwendeten Designs dadurch beschleunigt werden kann.

Contents

List of Figures

Introduction

Quantum computing is a new technology based on quantum mechanics and promises to be far more efficient in special tasks than classical computers ([1]). It uses the fact that quantum states are a superposition of basis states. Thus, they possess a solution space exponentially larger than classical states. On top of that, so-called quantum entanglement introduces correlations between quantum bits - the basic unit of information in quantum computers. Both features enable higher computational speed. However, quantum computing is currently of limited use due to noise and hardware limitations. Superconducting quantum computers use electronic circuits and are a promising implementation of the theory. They have several advantages over different approaches but still struggle with noise. This includes systematic noise like miscalibration but especially random noise introduced by the coupling to the environment and fluctuations of the current or voltage of the circuit parts themselves. An approach to solve this is to introduce so-called hybrid variational algorithms, e.g. within quantum machine learning. Shallow quantum circuits and classical algorithms are iteratively alternately executed. This reduces the influence of quantum noise on the whole process and could still have advantages like better accuracy. Machine Learning in general is a method to build statistical methods through the training of data to classify data correctly. E.g. the Fraunhofer Institute for Cognitive Systems tries to improve image detection of tumor diagnosis with a hybrid algorithm called quantum-classical convolutional neural network. To evaluate and improve its performance it is crucial to characterize the influence of quantum noise on the algorithm. Since the hybrid process is expensive in resources, this work focuses only on the quantum part. It especially analyses the influence of noise on the training process of the algorithm. After describing the important background knowledge in chapter 1, superconducting quantum computers are described in section 2. After that, chapter 3 introduces quantum noise and section 4 tools to characterize it. The influence of noise on quantum machine learning parts is then evaluated in chapter 5. In chapter 6, the results are interpreted. The work concludes with section 7 by summarizing the outcomes and proposing further research.

Chapter 1

Background

In this chapter, the mathematical tools and terminologies for the next sections are introduced.

1.1 Quantum States and the Hilbert Space

Quantum states can be expressed as complex vectors in a Hilbert Space, which is a normalized space with scalar products defined. To express them, the Dirac - or Bra-Ket - notation is shortly introduced. A quantum state vector v can be described by the Ket $|v\rangle$. An operator A acts on this Ket from the left. The eigenkets $|e_i\rangle$ of A with eigenvalues α_i can be described by $A\,|e_i\rangle = \alpha\,|e_i\rangle$. A covector w from the dual space V^* of V is written as the Bra $\langle w|$. The expectation value of an operator A in the state $|v\rangle$ is calculated by $\langle A\rangle_v = \langle v|A|v\rangle$. This can be also rewritten to $\langle A\rangle_v = \sum_i \alpha_i |\langle v|e_i\rangle|^2$. After the Copenhagen Interpretation $|\langle v|e_i\rangle|^2$ is hereby the probability to get the eigenvalue α_i. A hermitian operator is also called observable, because it directly relates to a possible measurement. This is because its eigenvalues and thus its expectation values are real. Unitary operators are used to transform the quantum state and have to be unitary to preserve $1 = \sum_i |\langle v|e_i\rangle|^2$. A quantum bit or qubit $|b\rangle$ is a quantum state which is a linear combination of two basis states $|e_0\rangle$ and $|e_1\rangle$. With the factors α and β it can be presented as $|b\rangle = \alpha\,|e_0\rangle + \beta\,|e_1\rangle$. As $1 = \langle b|b\rangle$ should always hold, $|\alpha^2| + |\beta^2| = 1$ is required. In this work it is assumed and defined without loss of generality that $|e_0\rangle = (1,0)^T = |0\rangle$ and $|e_1\rangle = (0,1)^T = |1\rangle$. To illustrate quantum states, it makes sense to parametrize α and β by circular coordinates:

$$|\Psi\rangle = exp(i\gamma)(cos\frac{\theta}{2}\,|0\rangle + exp(i\phi)sin\frac{\theta}{2}\,|1\rangle) \tag{1.1}$$

Since the measurement of an expectation value cannot extract the information of a global phase, the exponential term at the beginning can be neglected. Now, it is easily seen, that all of the valid quantum states lie on a spherical surface with a radius of 1. This fact can be visualized by the Bloch Sphere (cf. Figure 1.1). Actually, this is a simplification, because in reality the Bloch vectors precess around the z-axis with frequency $(E_1 - E_0)/\hbar$, if $|1\rangle$

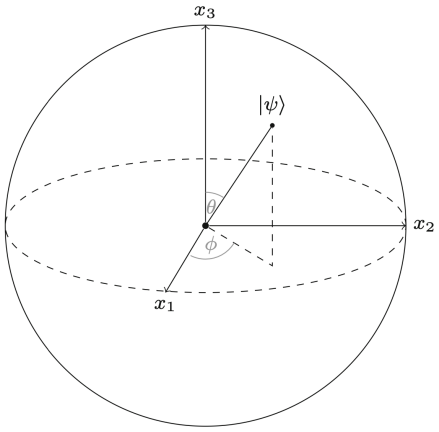

Figure 1.1: A quantum state can be presented as a point on the surface of a sphere by $|\Psi\rangle = cos\frac{\theta}{2}|0\rangle + exp(i\phi)sin\frac{\theta}{2}|1\rangle$. The sphere is called bloch sphere. Taken from [2].

has a higher energy $|0\rangle$. This is usually the case, so to better visualize quantum states a *rotating frame* is used without loss of generality. This makes the Bloch vectors stationary and visualization with the Bloch sphere is possible. Qubits serve as basic information units in a quantum computer being the quantum counterpart for bits in classical computers. One can also create a qudit - a linear combination with more than 2 basis states, but after [3] they can be always efficiently modeled by qubits. In a quantum system V with n qubits, they interact through the tensor product. The common shortcut for tensor products is used. It holds, that $V = |0\rangle_{n-1} \otimes \ldots \otimes |0\rangle_1 \doteq |00...00\rangle$. For more in-depth explanation to quantum bits and tensor products [4] and [3] is recommended. In classical systems, one bit can be represented in one dimension. Accordingly, n bits can span a space of $d = n$ dimensions. Using qubits instead of classical systems, however one qubit can only be presented in a two-dimensional complex vector space. Because of this, n qubits can represent data with dimensionality $d = 2^n$. This fact initiated the idea that computers based on quantum bits could be faster than classical computers (see [3]). Richard Feynman presented the first concept of a quantum computer in [5].

1.2 Quantum Circuits

Quantum circuits consist of the three parts: state preparation, transformation, and measurement of n qubits. Since most quantum computers only have one native preparation and measurement of states, these two steps include transformations as well. Unitary operations applied to one or two qubits are called gates. A list of common elementary gates

is:

$$
\begin{array}{lll}
I: & |0\rangle\langle 0| + |1\rangle\langle 1| & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
X: & |1\rangle\langle 0| + |0\rangle\langle 1| & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
Y: & -|1\rangle\langle 0| + |0\rangle\langle 1| & \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \\
Z: & |0\rangle\langle 0| - |1\rangle\langle 1| & \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
H: & \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |1\rangle\langle 0| \\ & + |0\rangle\langle 1| + |1\rangle\langle 1|) & \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\
CNOT: & |00\rangle\langle 00| + |01\rangle\langle 01| \\ & + |11\rangle\langle 10| + |10\rangle\langle 11| & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}
\end{array}
$$

Here, I describes the identity operator, X the negation transformation. Z changes the relative phase of a superposition of states, and $Y = ZX$. X, Y, and Z are equivalent to the Pauli matrices. H is called Hadamard gate, and it is often used to produce a superposition. All of these gates can be thought of as rotations of the quantum state vector on the Bloch Sphere. These operations do not have to be fixed. Instead, also parametrized rotational operators exist. In the following, the most important class of them is introduced. They are called *Pauli rotations* because they rotate around the same axis the associated Pauli operator does.

R_X	$e^{-i\frac{\theta}{2}X}$	$\begin{pmatrix} cos(\frac{\theta}{2}) & -isin(\frac{\theta}{2}) \\ -isin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix}$
R_Y	$e^{-i\frac{\theta}{2}Y}$	$\begin{pmatrix} cos(\frac{\theta}{2}) & -sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix}$
$R_Z(P)$	$e^{-i\frac{\theta}{2}Z}$	$\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$

An example illustration of the X-Gate (or R_X-Gate with $\theta = \pi$) can be seen in figure 1.2. Controlled Gates such as the CX-Gate are used to entangle qubits. Entanglement

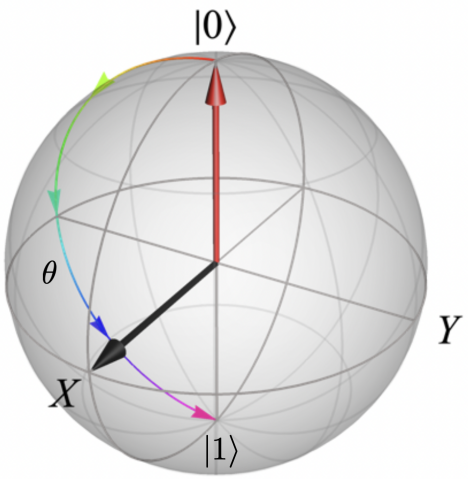

Figure 1.2: One-qubit gates can be represented as trajectories on the Bloch sphere. Taken from [6].

produces correlations of measurement of the two qubits and can thus be used to exploit mutual information. For example, if state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$ would be transformed by a CX-Gate, the qubit would end up in a so-called bell state which is entangled:

$$|\Psi\rangle = CX\,|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Now if the first qubit is measured, the second one is fixed to the same value. This is meant by correlation. Entanglement is a crucial point in designing quantum algorithms, because the whole Hilbert Space cannot be reached without it. Therefore the full potential of quantum computing could not be reaped. Figure 1.3 shows an example of a quantum circuit. The qubits are initialized in their groundstate $|0\rangle$, are then transformed using RY gates and CX gates, and measured in the Z basis. The transformation is called Real Amplitudes Ansatz (RA). In the RA Ansatz, RY-Gates are applied once or multiple times on each qubit. Between the application of them, CX-Gates are applied to every pair of qubits. Another example is shown in figure 1.4. Here, after initializing the qubits again in their groundstate $|0\rangle$, first a Hadamard gate and an RZ gate with an angle depending on x_i is applied to every qubit q_i. After this, a full entanglement of the qubits is achieved by RZZ-Gates on each pair of qubits q_i and q_j. The RZZ-Gate decomposes to two CX-Gates - with the first qubit as the control - and on RZ-Gate in their middle. The RZ-Gate is applied to the second qubit and has an angle of $\theta_i j = (\pi - x_i)(\pi - x_{i+j})$ (see [7] for more information).

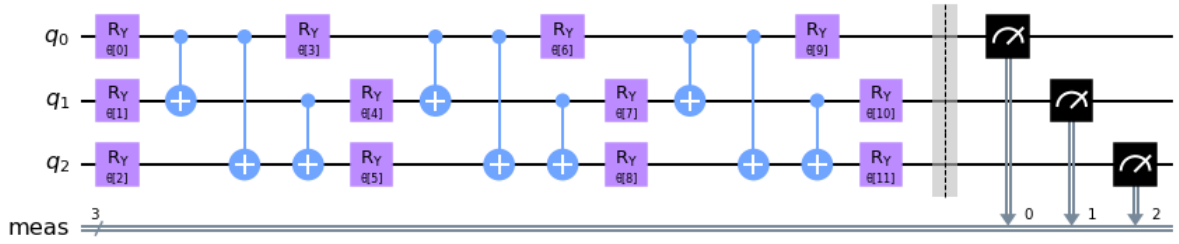

Figure 1.3: Quantum Circuit with a Real Amplitudes Ansatz.

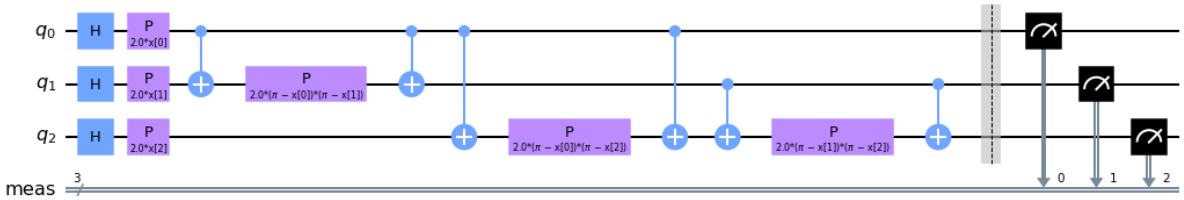

Figure 1.4: Quantum Circuit with a ZZ Feature Map.

1.2.1 Variational Quantum Algorithms

VQAs are quantum-classically hybrid algorithms. Parametrized rotation gates are used to create a circle of evaluation and adjustment. First, an initial state is created. Then, rotation gates with free parameters θ transform it and the observable is measured. The expectation value of this circuit is used to calculate new parameters for the transformation, which optimize a scalar cost function. Graphic 1.5 shows this exemplarily.

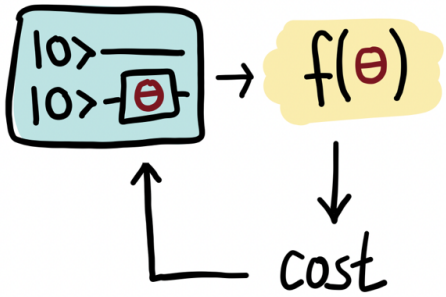

Figure 1.5: The basic principle of a variational algorithm is shown. Taken from [8].

1.3 Noisy intermediate-scale quantum era

At the time of writing, the development of quantum computers is still in the noisy intermediate-scale quantum era (NISQ). This is a time when intermediate-scale quantum processors ranging from 50 to a few hundred qubits are available. It was introduced in [9] by John Preskill. The term *noisy* refers to the sensitivity of quantum processors to noise. In the NISQ era, quantum devices are still not advanced enough to reach natural fault tolerance. The term *intermediate-scale* refers to the medium number of qubits of the quantum computer. Because of this, fault-tolerance through quantum error correction is not achievable. Latter is a way to completely eliminate errors of quantum computation by spreading the information of one logical qubit onto an entangled state of many physical qubits. Because of these two reasons, in the NISQ era, quantum computing is probably not able to reach supremacy over classical algorithms for real problems. To still use quantum processors properly, several approaches were developed. One of them is quantum mitigation. Instead of fully correcting errors, they are only mitigated. This works by approximating the error, a gate, measurement, or state preparation introduced into the circuit through additional measurements of specifically designed circuits. After this, the error of the original quantum circuit is statistically mitigated by post-processing its measurement outcome. Another one is quantum machine learning which is described in the following section.

1.4 Quantum Machine Learning

Machine Learning is a classical computation method where experiences influence a statistical model describing given data. Since this approach is very successful, quantum machine learning tries to project these benefits onto quantum computing. Recent research shows a great potential especially for NISQ devices (see [7]). Therefore, quantum machine learning could be a great step in the direction of the practicability of quantum computers even without them being fault-tolerant. This chapter is based on [2], [10] and [11]. They are also recommended for deeper studying of classical and quantum machine learning.

1.4.1 Classical Neural Networks

Neural networks are a category of machine learning and inspired by biological neurons of the nervous system. They are instrumental in image and speech recognition. To construct an artificial neural network, some *neurons* - just like in biology - are needed, but here they are organized in layers with one input and one output layer. Among themselves, one neuron is connected to neurons in the next layer and units in the previous layer if existing. This structure can be seen in fig. 1.6 with the green dots representing the inputs and the yellow ones the outputs. Blue dots are a middle layer connected to adjacent neurons called a *hidden layer*. The network classifies inputs based on the weights of the links between neurons. The aim is to find a method to determine all the weights properly to classify inputs correctly. First, the weights have a random number. After the initialization, the

training data is forward-propagated through the neural network. The difference between the generated output and the desired result is the error of the network. The goal is to make this error as small as possible by modifying the weights of the neuron links. However, not every neuron is equally to blame for the created error. Neurons producing a significant error need more adjustment than inconsequential ones. At each optimization step, all the influences of the weights on the error from the output to the input layer are evaluated and changed accordingly. The procedure is called *backpropagation*. By applying this method to the set of training data multiple times, the network should label inputs with the correct outputs. As image recognition is concerned, it is desirable to be able to categorize given patterns in images to give a better and more reliable output. Patterns like shapes or edges could help classify for example handwriting like in figure 1.7. To give an instance, if the neural network had a neuron in a layer that detects circles that would strongly point to digits with round patterns like a zero, six, eight, or nine. On the other hand, one neuron could recognize straight vertical edges, which in turn would suggest a one, four, or nine. This concept of detecting patterns in the input data is the idea of *Convolutional Neural Networks*. To recognize patterns, so-called *kernels* or *filters* are used, which are just a matrix representing the desired pattern, like in 1.2, a vertical line tilted to the right at the top.

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.2}$$

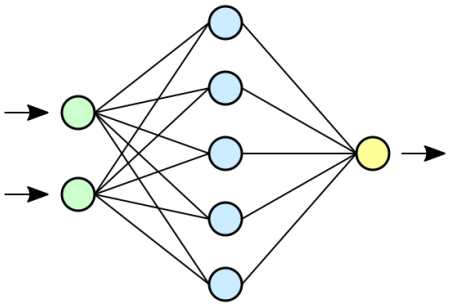

Figure 1.6: Simple illustration for neural networks. Taken from [12].

This kernel consisting of entries $k_{i,j}$ can now be applied on the input matrix called pixel matrix with its elements $m_{i,j}$. A measure $F_{x,y}$ of how well the kernel and pixel matrix match can be calculated by

$$F_{x,y} = \sum_{i,j} m_{x+i,y+j} \cdot k_{i,j}. \tag{1.3}$$

Suppose a pixel matrix 1.4 with the kernel 1.2 the output matrix would be 1.5.

$$\begin{pmatrix} 23 & 94 & 180 & 210 & 127 & 90 & 17 \\ 32 & 154 & 200 & 234 & 186 & 87 & 21 \\ 95 & 181 & 240 & 190 & 209 & 110 & 32 \\ 176 & 187 & 207 & 118 & 182 & 172 & 90 \end{pmatrix} \tag{1.4}$$

Figure 1.7: An example of training data for CNNs. Taken from ([10]).

$$\begin{pmatrix} 534 & 604 & 605 & 324 & 143 \\ 601 & 508 & 625 & 468 & 209 \end{pmatrix} \tag{1.5}$$

This is computed by using the equation 1.3 as following:

$$\begin{aligned} F_{0,0} &= 94 \cdot 1 + 200 \cdot 1 + 240 \cdot 1 = 534 \\ F_{0,1} &= 180 \cdot 1 + 234 \cdot 1 + 190 \cdot 1 = 604 \\ &\vdots \\ F_{1,5} &= 87 \cdot 1 + 32 \cdot 1 + 90 \cdot 1 = 209 \end{aligned}$$

Of course, in actual use cases, the filter can be different sizes and contain any float to represent a given pattern. Moreover, there can be multiple patterns on one layer for several patterns. The way to train a CNN is similar to the previously discussed method of backpropagation of error for conventional neural networks.

1.4.2 A deterministic quantum machine learning model

To interpret a variational algorithm as a determinisic machine learning model, a quantum circuit $U(x, \theta)$ - that depends on the input x and θ - is applied on the initial state $|0\rangle$ and the measurement M interpreted as the output of the model. If $|\psi(x, \theta)\rangle = U(x, \theta)\,|0\rangle$ the function $f_\theta(x) = \langle\psi(x, \theta)|\, M\, |\psi(x, \theta)\rangle$ defines the variational model. Often, the quantum circuit $U(x, \theta)$ is split into the data embedding block $S(x)$ and the parametrized block $W(\theta)$:

$$U(x, \theta) = W(\theta)S(x)$$

Figure 1.8 shows how a quantum machine learning model is built. An example of a machine learning circuit is the Two Layer QNN. It uses a ZZ Feature Map as the encoding block $S(x)$ and the Real Amplitudes Ansatz for the parametrized circuit $W(\theta)$. A second example will be called Fraunhofer QNN in this work. It also encodes with the ZZ Feature Map. This time, however the Hadamard gate and the RZ-Gate at the end are not included. The main variational part $W(\theta)$ is a Real-Amplitudes ansatz, but with RX-Gates, depth 1, and a full

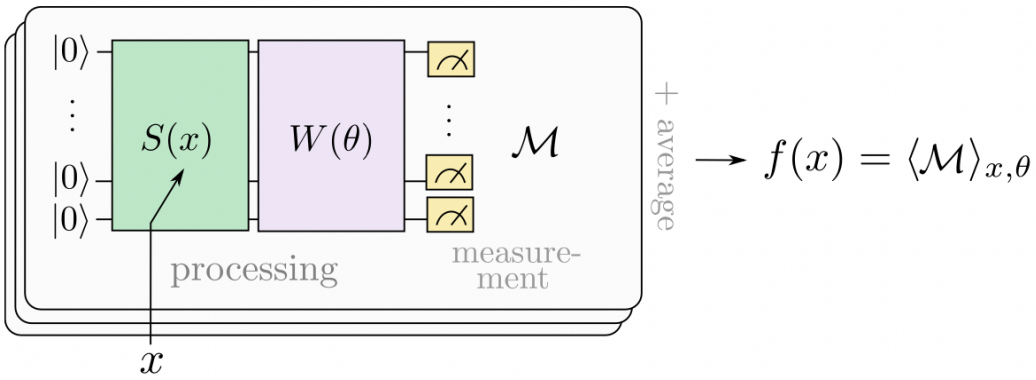

Figure 1.8: To evaluate a model $f_\theta(x)$, the expectation value of the circuit $U(x, \theta) = W(\theta)S(x)$ is experimentally determined.

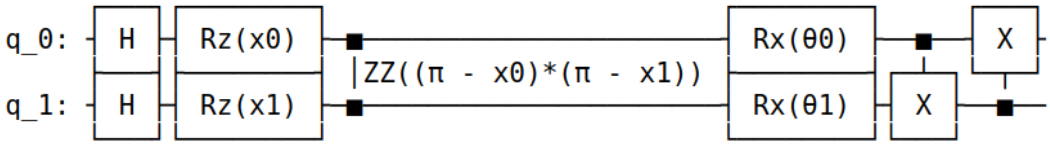

Figure 1.9: After a ZZ-Feature-Map with H-, RZ- and RZZ-Gates, RX-Gates and CX-Gates are applied in the Fraunhofer QNN implementation.

entanglement CNOT gate layer at the end. This model is shown in figure 1.9. There are four different definitions of quantum machine learning regarding its data and its algorithm. It could mean processing quantum or classical data with quantum or classical procedures. This work focuses on quantum machine learning with classical data which is encoded and then processed by a variational quantum algorithm. Figure 1.10 shows the four categories.

1.4.3 Quantum-Classical Convolutional Neural Network

The main motivation for the characterization of noise in variational algorithms is to improve tumor image recognition with quantum machine learning. For this, a quantum-classical convolutional neural network is used at the Fraunhofer IKS group. The QCCNN model combines a classical CNN with quantum convolutional layers. A convolutional neural network is chosen, because after [13], they could have an advantage in generalization over classical counterparts. In medical imaging, there is not much data available to train on, so it is very important to get a good accuracy even then. Because evaluating the whole hybrid algorithm would mix the effects of classical and quantum layers and is expensive in ressources, this work focuses on the characterization of the quantum model only. There are many different approaches on which quantum part to use in the QCNN. In this paper, only two of them are considered: Two-Layer QNN and Fraunhofer QNN.

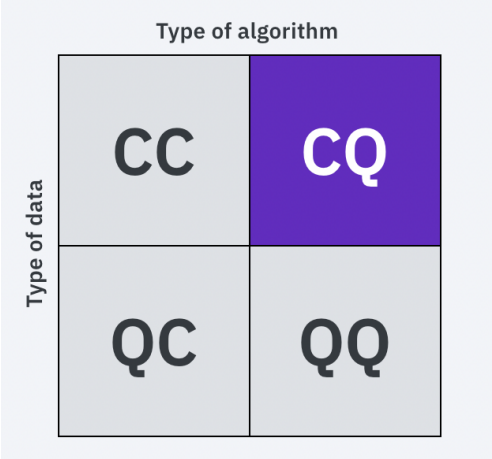

Figure 1.10: There are four different definitions, but in this work, processing encoded classical data by quantum machine models is defined as quantum machine learning. Taken from [6].

Chapter 2

Superconducting Quantum Computer

Leaving the theory behind, one can wonder now how to build a quantum computer. There are several architectures, which include silicon([14]), ion traps([15], [16]) and photons ([17]). One of the most promising candidates for a scalable quantum computer however is superconducting qubits([18], [19]). While the other three naturally do have discrete intrinsic energy levels or other attributes, which make a 2-level system (needed for qubits) possible, superconducting qubits do not own them naturally. It is an LC circuit with a so-called Josephson-Junction to introduce anharmonicity. An LC circuit is an electrical circuit, where an inductor and a capacitor are connected. Its dynamic behavior is described as an electrical oscillator with a resonant frequency. The Josephson-Junction is a device that sandwiches a nonsuperconducting material with superconducting material. The nonlinear behavior occurs due to the tunneling of pairs of electrons from one superconducting material to the other. The junctions are needed to distinguish different level transitions. Compared to other architectures, superconducting qubits have key advantages. They possess a high designability (a lot of different designs for qubits), are probably well scalable (manufacturing based on classical semiconductor production), are easy to couple (by capacitance or inductance) and easy to control (with commercial microwave devices) (see [18]). It was first introduced in [20] and resulted in great research interest over the last two decades. The software package which was used in this work is called qiskit [21] and is connected with the IBM quantum computers [22].

2.1 Anharmonic oscillator and transmon qubit

In any circuit, the time-dependent energy can be calculated with

$$E(t) = \int_{-\infty}^{t} V(t')I(t')dt', \tag{2.1}$$

while $V(t')$ denotes the voltage and $I(t')$ the current at time t'. The magnetic flux is defined by

$$\Phi(t) = \int_{-\infty}^{t} V(t')dt'. \tag{2.2}$$

The Hamiltonian of a LC circuit (circuit with inductor and capactitor) can then be described by

$$H = \frac{Q^2}{2C} + \frac{\Phi^2}{2L} \equiv \frac{1}{2}CV^2 + \frac{1}{2}LI^2 \tag{2.3}$$

with its magnetic flux Φ and charge Q, the inductance L and the capacitance C. The Hamiltonian described above is classical. To transform it into a quantum-mechanical one, magnetic flux and charge are transformed to quantum mechanical operators. For simplicity the same notation for the operators are used. By taking the relationship between classical Poisson brackets and quantum mechanical commutators into account, a commutation relation between both operators is obtained:

$$[\hat{\Phi}, \hat{Q}] = \hat{\Phi}\hat{Q} - \hat{Q}\hat{\Phi} = i\hbar, \tag{2.4}$$

By introducing the reduced flux $\phi \equiv 2\pi\Phi/\Phi_0$ and reduced charge $n = Q/2e$ the quantum-mechanical Hamiltonian is described by

$$H = 4E_C n^2 + \frac{1}{2}E_L\phi^2. \tag{2.5}$$

Here, $E_C = e^2/(2C)$ is the charging energy required to add each electron of the Cooper-pair to the island, $E_L = (\Phi_0/2\pi)^2/L$ is the inductive energy. $\Phi_0 = h/(2e)$ is the superconducting magnetic flux quantum. The operator n represents the excess number of Cooper-pairs on the island and ϕ is denominated the gauge-invariant phase of the inductor. This system is called a quantum harmonic oscillator (QHO). As the desired quantum computer works with qubits, it is necessary, to be able to define a two-level system, even if it is a subspace of the computational space. Since the energy levels of the Hamiltonian in Eq. 2.3 are equidistant, it is not possible to do so. Therefore, it is crucial to introduce a nonlinearity to the circuit to make the levels distinguishable. A *Josephson-Junction* ([23]) is used to introduce this anharmonicity. The linear inductor will be substituted by the Josephson-Junction as a kind of nonlinear inductor. Its energy can be calculated with 2.1 and the two equations

$$I = I_c \sin(\phi), \qquad V = \frac{\hbar}{2e}\frac{d\phi}{dt}, \tag{2.6}$$

which results in the new Hamiltonian

$$H = 4E_C n^2 - E_J \cos(\phi), \tag{2.7}$$

while $E_C = e^2/(2C_\Sigma)$ with $C_\Sigma = C_s + C_J$ as the total capacitance with the shunted capacitance C_s and the one of the Josephson-Junction itself. $E_J = I_c\Phi_0/2\pi$ is the Josephson energy, where I_c is the critical current of the junction. Since a cosinus is introduced in 2.7, the energy function is not longer parabolically (see Figure 2.1). Instead, it introduces the anharmonicity wanted and transforms the energy spectrum to a non-degenerate one. Thus, a Josephson-Junction allows for building a two-level system with regular electric circuits.

There are two important regimes expressed by the E_J/E_C ratio. The state-of-the-art approach is designed with $E_J \gg E_C$, because if $E_J \leq E_C$, the qubit becomes very sensitive

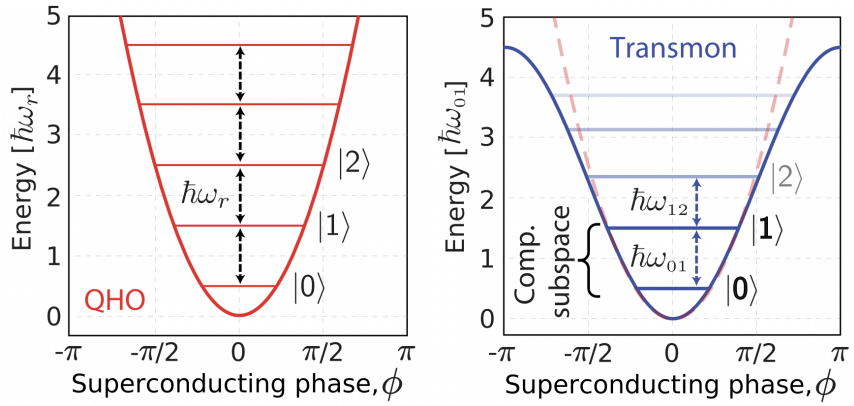

Figure 2.1: Through the introduction of a Josephson-Junction, the harmonic oscillator becomes an anharmonic one. This is the requirement for defining a two-level subsystem for the superconducting transmon qubit. Taken from [19].

to charge noise. This is more challenging for achieving high coherence than flux noise. Qubits with the first design are called flux qubits, the other type is denominated charge qubits. The so-called *transmon qubit* design is a flux qubit, which is achieved by shunting the junction with a large capacitor $C_s \gg C_J$. This decreases E_C and makes the qubit less sensitive to charge noise. As mentioned at the beginning of this chapter, operations can be performed by applying a microwave pulse to the superconducting qubit. An example of a quantum processor using transmon qubits can be seen in figure 2.2.

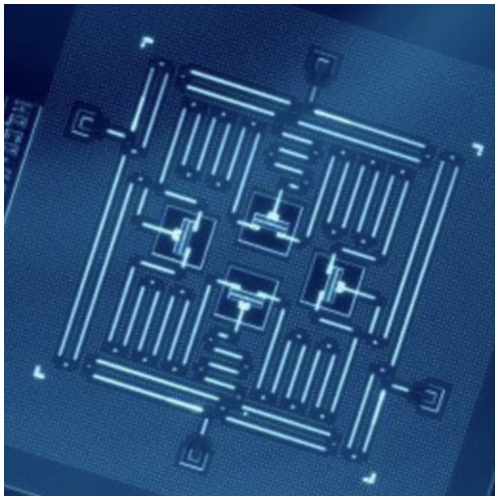

Figure 2.2: Here, a four-qubit quantum processor of IBM is depicted. The four squares in the middle are the Transmon Qubits with the Josephson-Junctions and the lines are quantum buses for control and readout. Taken from [24].

2.2 State of the art

In this section, the theory is set into perspective by describing the state of the art in quantum computing. The most advanced superconducting quantum computer *Eagle* produced by IBM has 127 qubits right now. However, they are planning to present a quantum computer with over 1000 qubits in as soon as 2023. Figure 2.3 shows the roadmap of IBM over the next years. IBM Lima is a quantum processor of IBM and has got 5 qubits.

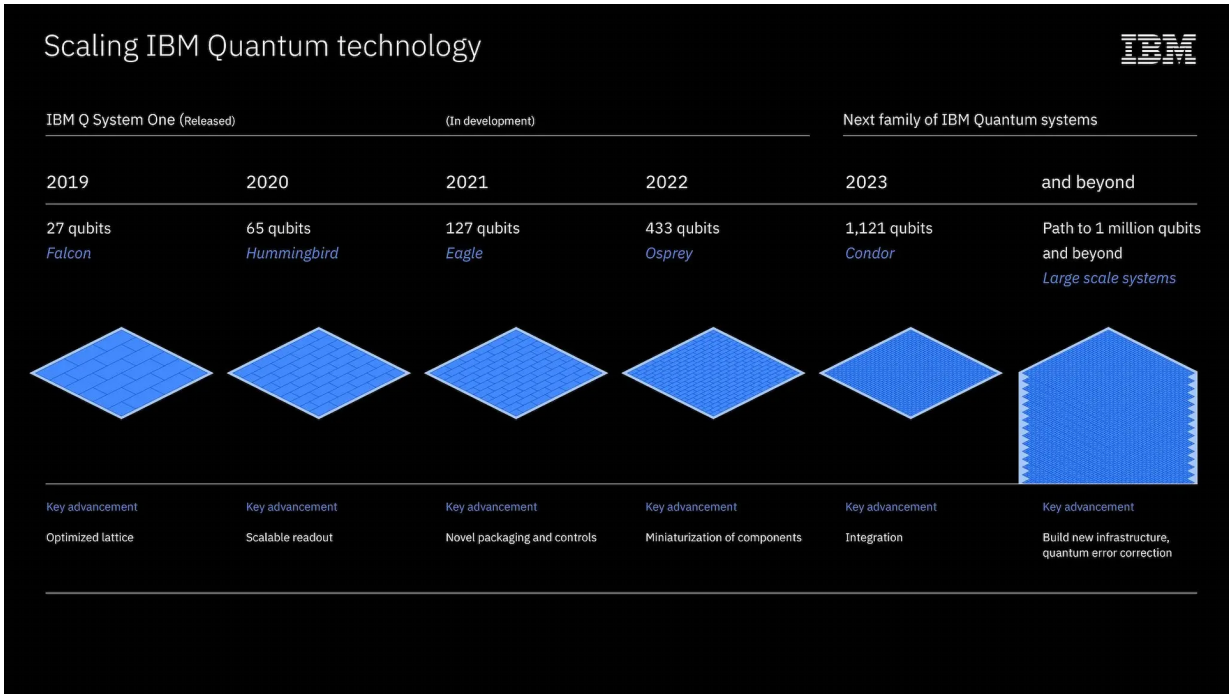

Figure 2.3: IBM aims for over 1000 qubit quantum processors until 2023 ([25])

Chapter 3

Noise

The chapter is oriented at [26] and the resources [6], [27]. It is focused on superconducting quantum computing devices and introduces different types of noise and approaches to how to model them. The final section describes how to approximate the noise of real quantum devices to use them in simulations. This is specified with the quantum processor Lima of IBM.

3.1 Types of noise

In an open quantum system, the qubit is confronted with noise. Instead of a closed on, an open quantum systems interacts with the environment. Those interactions change the behavior of the system. Noise can be structured into two parts, systematic noise, and the more difficult stochastic noise.

3.1.1 Systematic Noise

Systematic errors are fixed control or state preparation and readout errors. First is also called coherent noise and arises from miscalibrated elementary transformations. This produces a slight error that gets bigger with every circuit layer. If one takes the R_X gate for example, it is observed, that $\theta = \pi + \epsilon$ with ϵ as the miscalibration of the angle. So

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = e^{-i\frac{\pi+\epsilon}{2}X} = R_X(\pi)R_X(\epsilon) \tag{3.1}$$

is the real rotation operation. That means, an additional rotation is introduced. This leads to a cosinus term in the expectation value of the quantum state after applying this

operation d-times:

$$\begin{aligned} R_X(\theta)^d &= R_X(d\pi)R_X(d\epsilon) \\ \Psi_{ideal} &= (R_X(d\pi))\,|0\rangle \\ \Psi_{noise} &= (R_X(d\pi)R_X(d\epsilon))\,|0\rangle \\ E_{ideal} &= \langle\Psi_{ideal}|\,Z\,|\Psi_{ideal}\rangle = cos(d\pi) = (-1)^d \\ E_{noise} &= \langle\Psi_{noise}|\,Z\,|\Psi_{noise}\rangle = cos(d\pi + d\epsilon) \end{aligned}$$

To illustrate this behaviour, figure 3.1 shows the noisy rotation on the bloch sphere, and figure 3.2 the behaviour of a circuit with noisy X-Gates and Z-measurement. If a system-

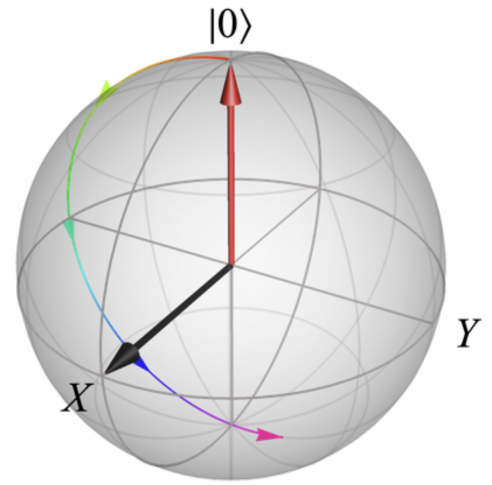

Figure 3.1: Coherent, systematic gate noise can be visualized as a trajectory on the Bloch sphere with a slightly false length or direction. Taken from [6].

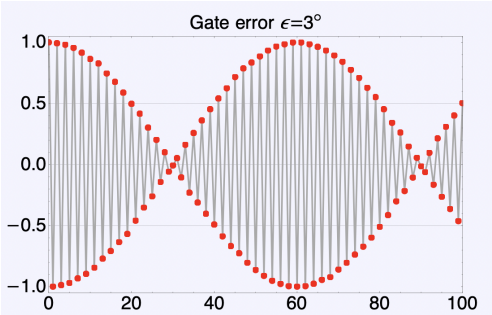

Figure 3.2: If a state is prepared in its ground state and a noisy X-Gate is applied d-times and the expectation value is measured in the Z basis, one gets a cosine trend. Taken from [6].

atic gate error is identified it can be corrected by calibration, that is why it is focused on

gate errors of the second type. The second part of systematic noise are state preparation and measurement (SPAM) errors. Here, one assumes that even without any gates transforming the quantum state, it already possesses an error to the ideal theoretical value. This happens during the state preparation and at the measurement part due to errors in the physical implementations. Since both can not be distinguished very well - if at all - they are combined.

3.1.2 Stochastic Noise

Stochastic Noise occurs if the qubit is randomly coupled to the environment due to temperature. Also, even the circuit elements themselves possess certain voltage and current fluctuations. Both together lead to qubit decoherence. This type of noise is more difficult than the first one, because due to its randomness it is more difficult to characterize.

3.1.3 Sample Error

Since the real probability of states are only approached through measurements, the maximum precision has an upper bound given by the law of large numbers. (see [28]) If the variances of n independent random variables with the same expectation value are observed, the variance of them altogether would decrease with $\frac{1}{n}$. So only with a high number of samples a small variance or small statistical noise is achieved.

3.2 Noise Modeling

A way is to search how to properly model noise to be able to characterize it and in the end, hopefully, mitigate it. Since it is very much easier to visualize, it is once more relied on the Bloch Sphere in this chapter. Two important values have to be mentioned beforehand. One is the *relaxation time* T_1 of the qubit. This is a decay constant describing with which probability state $|1\rangle$ is not yet decayed into state $|0\rangle$: $P(|1\rangle) = e^{-\frac{t}{T_1}}$. If one applies a Hadamard Gate to the groundstate, wait time t and apply a Hadamard gate again. The propability of getting $|0\rangle$ is described by $P(|0\rangle) = \frac{1}{2}e^{-\frac{t}{T_2}} + \frac{1}{2}$. T_2 is the *phase coherence time*. (cf. [29]). There are two important noise rates which are further explained below:

$$\text{longitudinal relaxation rate:} \qquad \Gamma_1 \equiv \frac{1}{T_1} \tag{3.2}$$

$$\text{transverse relaxation rate:} \qquad \Gamma_2 \equiv \frac{1}{T_2} = \frac{\Gamma_1}{2} + \Gamma_\varphi \tag{3.3}$$

3.2.1 Longitudinal relaxation or depolarizing

As visualized in figure 3.3, depolarization or longitudinal relaxation forces the qubit from excited state to the groundstate or the other way round. It occurs through energy exchange

with the environment of the qubit and leads to an *up transition rate* $\Gamma_{1\uparrow}$ and *down transition rate* $\Gamma_{1\downarrow}$. The *longitudinal relaxation rate* is then described by

$$\Gamma_1 \equiv \frac{1}{T_1} = \Gamma_{1\downarrow} + \Gamma_{1\uparrow}. \tag{3.4}$$

With Boltzmann statistics, both transition rates can be set in relation to each other by $\Gamma_{1\uparrow} = \exp(-\hbar\omega/k_B T)\Gamma_{1\downarrow}$, T is the temperature and k_B is the Boltzmann constant. Since most superconducting computers operate at temperatures around $T = 20$mK and qubit frequency of $\omega/2\pi \approx 5$ GHz ([19]), $\Gamma_{1\uparrow} \ll \Gamma_{1\downarrow}$ holds. So only the relaxation rate - through energy emission to the environment - significantly contributes to depolarizing noise. Therefore, every state gets back down to the groundstate if t goes to infinity.

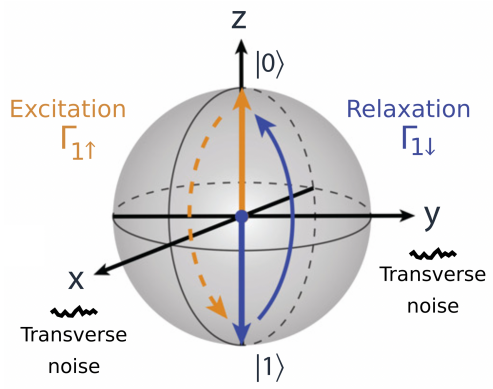

Figure 3.3: Depolarization is shown on the Bloch sphere. Taken from [19].

3.2.2 Pure dephasing

The pure dephasing rate Γ_ϕ is induced by longitudinal noise (see figure 3.4). It causes a precession of the Bloch vector around the z-axis and eventually leads to a complete depolarization of ϕ.

3.2.3 Transverse relaxation

In figure 3.5, the transverse relaxation is graphically represented. It shows how coherent superposition states are affected. Transverse relaxation can be described by

$$\Gamma_2 = \Gamma_1/2 + \Gamma_\phi. \tag{3.5}$$

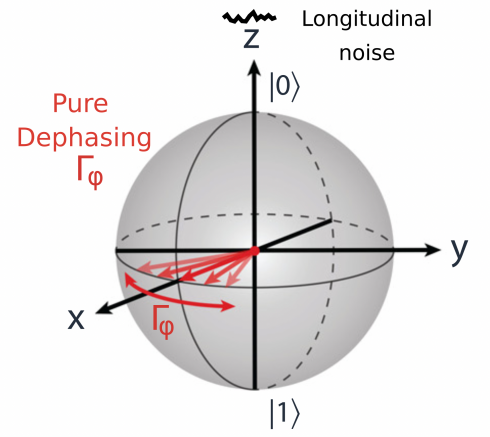

Figure 3.4: Pure dephasing is represented on the Bloch sphere. Taken from [19].

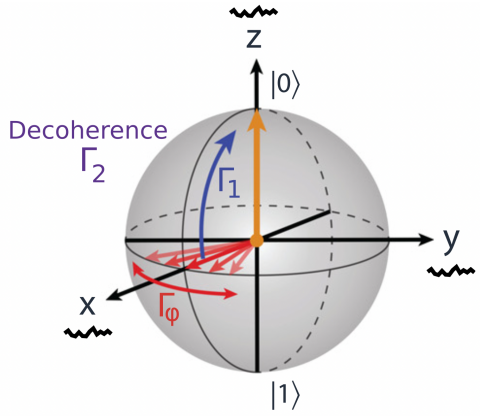

Figure 3.5: Visualization of transverse relaxation. Taken from [19].

3.3 Simulation of Noise

For the simulation of noise, a correct noise model is needed. In the qiskit ([21]) framework, one can load noise models directly from the specifications of IBM backends. In the following work, these provided noise models are used. It is the approximated noise of the quantum device IBM Lima with depolarization noise or/and readout noise channels. Also, noise models can easily be constructed and implemented in the qiskit framework.

Chapter 4

Characterization of Noise

In this chapter, the basic principles of characterization are introduced. Since the goal is to analyse the exact noise model it will be focused on quantum tomography and not other general approaches like Randomized Benchmarking (see [30]). It is based on ideas from [26]. A deeper practical introduction can be found in [31]. Four kinds of tomography are introduced here. They are called Process, State and Measurement Tomography, and Gate Set Tomography. Process, State, and Measurement Tomography cannot make realistic assumptions as a result of their interdependence. This problem is shown in figure 4.1. For example to successfully characterize the error of preparation, one has to assume no errors at the measurement. This is obviously not correct. If the error is tried to be mitigated by linear inversion, measurement tomography is needed. However, this assumes no error in the preparation process, which is contradictory to the starting position. To avoid this issue, Gate Set Tomography determines processes up to a similarity transform. This results in a self-calibrated characterization of the process. However, as a disadvantage, it is even more inefficient than standard tomography, because it needs far more circuits to be evaluated.

4.1 Superoperators and gate sets

This chapter is oriented at [26]. To make modelling noise easier, mathematical descriptions of section 1.1 is deepened. Instead of describing a quantum state by its complex vector in the Hilbert space $\mathcal{H} = \mathbb{C}^d$, it is represented by a $d \times d$ *density matrix* ρ that acts on its Hilbert space ($\rho : \mathcal{H} \to \mathcal{H}$). It must be semidefinite and have trace 1. A trace of a matrix is defined as the sum over its eigenvalues. The matrix can be embedded in a vector space with dimension d^2. This space is called *Hilbert-Schmidt space* $\mathcal{B}(\mathcal{H})$. In the following only the real subspace of it spanned by the Hermitian operators will be used. The *Hilbert-Schmidt space* is equipped with an inner product $\langle A, B\rangle \equiv \mathrm{Tr}(A^\dagger B)$. To distinguish it from the states of the Hilbert space, a vector is denoted with $|B\rangle\rangle$ and its covector with $\langle\langle A|$. Operations on $\mathcal{B}(\mathcal{H})$ are called *superoperators*, so $|B\rangle\rangle$ and $\langle\langle A|$ are called *superket* and *superbra*. Measuring a system yields a result from k possible outcomes. The probability of that outcome is only determined by its density matrix, so it can be presented by the dual

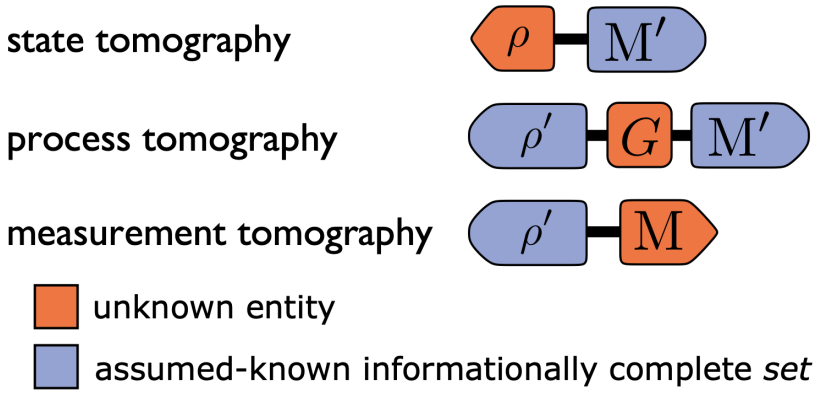

Figure 4.1: Here, different kinds of tomography are shown. State and measurement tomography assumes the other one to be informationally complete and without error. Process tomography assumes, that both of the latter are not perturbed by any error. Taken from [26].

vector $\langle\langle E_i|$ with the probability of getting the i-th result $\Pr(i|\rho) = \langle\langle E_i|\rho\rangle\rangle = \mathrm{Tr}(E_i\rho)$. The set $\{E_i\}$ completely describes the measurement and is called *positive operator-valued measure (POVM)*. $E_i \geq 0$ and $\sum_i E_i = \mathbb{1}$ holds. E_i can be called an *effect*. For convenience, a basis for $\mathcal{B}(\mathcal{H})$ is introduced. It should be hermitian, orthonormal and traceless for $i > 0$: $B_0 = \mathbb{1}/\sqrt{d}$ and $\mathrm{Tr}(B_i) = 0 \;\forall\; i > 0$. Such a basis is given for example by the normalized Pauli matrices for one qubit and for n qubits, the n-fold tensor product of the Pauli operators. It will be the standard basis used in this work. So for one qubit, it is $\{B_i\} = \left\{\mathbb{1}/\sqrt{2}, \sigma_x/\sqrt{2}, \sigma_y/\sqrt{2}, \sigma_z/\sqrt{2}\right\}$ with σ_i representing the Pauli matrices. To describe the set of circuit layers or gates, it is necessary to introduce *superoperators*. As described in section 1.1, a quantum gate corresponds to a $d \times d$ unitary transformation U of the quantum state. So if this gate would be applied, it would convert the density matrix: $\rho \to U\rho U^\dagger$. Since it maps $\mathcal{B}(\mathcal{H})$ linearly to itself, it is called a superoperator. If U is orthogonal if it is ideal. However, real transformations are noisy. This results in them not being orthogonal and thus contractive. These superoperators are called *quantum channels* or *quantum processes*. Since a superoperator Λ linearly transforms a superket or -bra, it can be represented as a $d^2 \times d^2$ matrix - the so-called transfer matrix S_Λ. The probability p_i of the outcome i with density matrix ρ, unitary superoperator $\Lambda : |\rho\rangle\rangle \mapsto \mathsf{S}_\Lambda|\rho\rangle\rangle$ and by measuring the effect E_i can then be calculated with

$$p_i = \langle\langle E_i|\mathsf{S}_\Lambda|\rho\rangle\rangle = \mathrm{Tr}(E_i\mathsf{S}_\Lambda\rho). \tag{4.1}$$

Since probabilistic properties should hold for models of physical systems, a superoperator Λ must be *completely positive and trace-preserving* (CPTP).

The mathematical tools can now be used for describing all three parts of a quantum circuit more conveniently. For modeling state preparation, N_ρ superkets of the density matrices of the natively prepared states are introduced. The measurement process is

represented by N_M superbras of the effects of the native measurements with $N_\mathrm{E}^{(m)}$ different outcomes. And finally, n-qubit operation are modelled using N_G superoperators:

$$\begin{aligned} G_i : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H}) \quad &\text{for} \quad i = 1 \ldots N_\mathrm{G}, \\ |\rho^{(i)}\rangle\!\rangle \in \mathcal{B}(\mathcal{H}) \quad &\text{for} \quad i = 1 \ldots N_\rho, \text{ and} \\ \langle\!\langle E_i^{(m)}| \in \mathcal{B}(\mathcal{H})^* \quad &\text{for} \quad m = 1 \ldots N_\mathrm{M},\, i = 1 \ldots N_\mathrm{E}^{(m)}. \end{aligned} \tag{4.2}$$

Sets of all three parts are called *Gate sets*. They specify the behaviour of a quantum processor. N-qubit operations are circuit layers of 1- or 2-qubit gates. A visualisation of them can be found in figure 4.2. A Gate set can be formally described by:

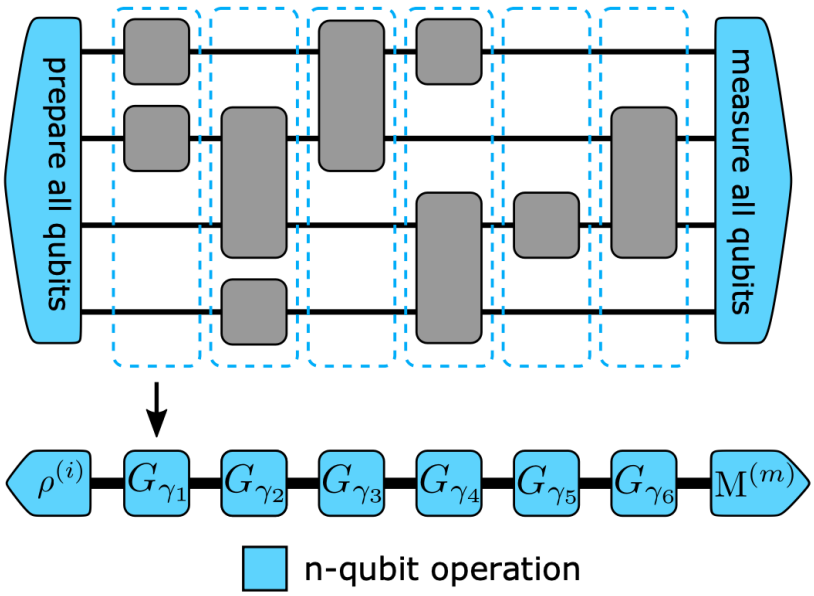

Figure 4.2: Multi-qubit operations can be represented as parallel executions of elementary one- or two-qubit gates. Taken from [26].

$$\mathcal{G} = \left\{ \left\{ |\rho^{(i)}\rangle\!\rangle \right\}_{i=1}^{N_\rho} ; \{G_i\}_{i=1}^{N_\mathrm{G}} ; \left\{ \langle\!\langle E_i^{(m)}| \right\}_{m=1,i=1}^{N_\mathrm{M},N_\mathrm{E}^{(m)}} \right\} \tag{4.3}$$

Normally a quantum computer has only one native state preparation and measurement process and therefore $|N_\rho| = |N_\mathrm{M}| = 1$. An important attribute of gate sets is the *over-specification* of it. Gate sets are equivalent up to a *gauge freedom*, because an applied invertible superoperator M would change any observable probability (rf. 4.1):

$$\begin{aligned} \langle\!\langle E_i^{(m)}| &\to \langle\!\langle E_i^{(m)}| M^{-1} \\ |\rho^{(i)}\rangle\!\rangle &\to M|\rho^{(i)}\rangle\!\rangle \\ G_i &\to M G_i M^{-1}, \end{aligned} \tag{4.4}$$

4.2 Process and Measurement Tomography

There are three different parts to be looked at when trying to characterize noise: state preparation, the process or quantum circuit and the measurement at the end. The basic idea of tomography is to assume two of them to be informationally complete ([26]). Thus, the third entity can be estimated only by measuring the outcomes of the circuit. In figure 4.1 this concept is visualised. After process tomography, measurement tomography is introduced. Since state tomography is basically the same as measurement tomography, it is not described in its own chapter.

4.2.1 Process Tomography

In process tomography they are assumed to be informationally complete and to not produce any error. To explain the algorithm, a few simplifications are needed. Definition 4.2 is used and summarized by

$$A = \begin{pmatrix} \langle\langle E_1| \\ \langle\langle E_2| \\ \vdots \\ \langle\langle E_{N_{f1}}| \end{pmatrix}, \quad B = \begin{pmatrix} |\rho_1\rangle\rangle & |\rho_2\rangle\rangle & \cdots |\rho_{N_{f2}}\rangle\rangle \end{pmatrix}. \tag{4.5}$$

Here, N_{f1} and N_{f2} is the total number of possible measurement effects or state preparations, it does not matter if two different effects get the same result. Because of the probability of one outcome is

$$\begin{aligned} P_{j,i} &= \mathrm{Tr}\left(E_j G[\rho_i]\right) && (4.6) \\ &= \langle\langle E_j|G|\rho_i\rangle\rangle && (4.7) \end{aligned}$$

with G being a superoperator of the observed process, this can just be rewritten with equation 4.5 to

$$P = AGB. \tag{4.8}$$

A and B should be informationally complete, which means they have full row rank. If they are also square, they are invertible and the value of G is determined by

$$G = A^{-1}PB^{-1}. \tag{4.9}$$

In theory, this sounds very good. However, perfect behavior of measurement and state preparation processes cannot be assumed in real devices. Therfore process tomography is very sensitive to measurement error. If this issue is tried to be resolved by adding measurement and state tomography it leads to an "endless loop of self-referentiality" ([26], page 11).

4.2.2 Measurement Tomography

Measurement tomography uses fiducial states and measurement outcomes to reconstruct a POVM operator. If $\{\rho_i\}$ are the states resulting from preparation and transformation and $\{E\}$ is the unknown POVM, one can define the probability vector of this measurement by

$$[p_j]_i = \mathrm{Tr}[\rho_i E_j] \tag{4.10}$$
$$= \langle\langle E_j|\rho_i\rangle\rangle. \tag{4.11}$$

With the defintion of B (see 4.5), this can be transformed to

$$\vec{p_j}^T = \langle\langle E_j|. \tag{4.12}$$

Then, the POVM can be reconstructed by

$$\langle\langle E_j| = \vec{p_j}^T B^{-1}, \tag{4.13}$$

with B^{-1} being the matrix inverse. As described in 4.2.1 and in the introduction of this chapter, measurement tomography suffers self-referentiality problems with process and state tomography. However, it can be used in a straightforward way to mitigate readout errors in quantum circuits. This is presented in the following section.

4.3 Quantum Error Mitigation

Quantum Error Mitigation (QEM) tries to estimate errors and mitigate them. Instead of error correction, it suppresses errors by repeated experiments and postprocessing of the outcomes. In this work, only measurement error mitigation (MEM) is evaluated. Beside this, there are multiple different more complex approaches like Zero Noise Extrapolation ([32]), Probabilistic-Error-Correction ([33]) or Clifford-Data-Regression ([34]). Measurement Error Mitigation (see [35]) tries to mitigate the measurement errors. In its simplest form, one prepares every 2^n basis state and measures the probability of its correct measure. This is essentially measurement tomography and gives a measurement matrix M with rows representing the basis states and the columns the outcomes. If enough shots are used, the normalized matrix approximates then the probabilities of the measurement error. An outcome can be just post-processed by applying the inverse measurement matrix M^{-1}. Further methods and efficient implementation for multiple qubits can be found in [35].

To show the influence of measurement error mitigation, a probability distribution is observed. In the figure 4.3 one can see the outcome of two X-gate acting on a 0 state. Here, the X-gates were implemented as an H-RZ-H circuit with an π angle. The ideal distribution after 1000 shots would be 1000 times the outcome 0. As it can be seen, as a result of the noise, the distribution of the noise model has also 1 states. A great part of these errors are resulting due to readout errors, therefore measurement error mitigation (chapter 4.3) can be used to correct parts of the results.

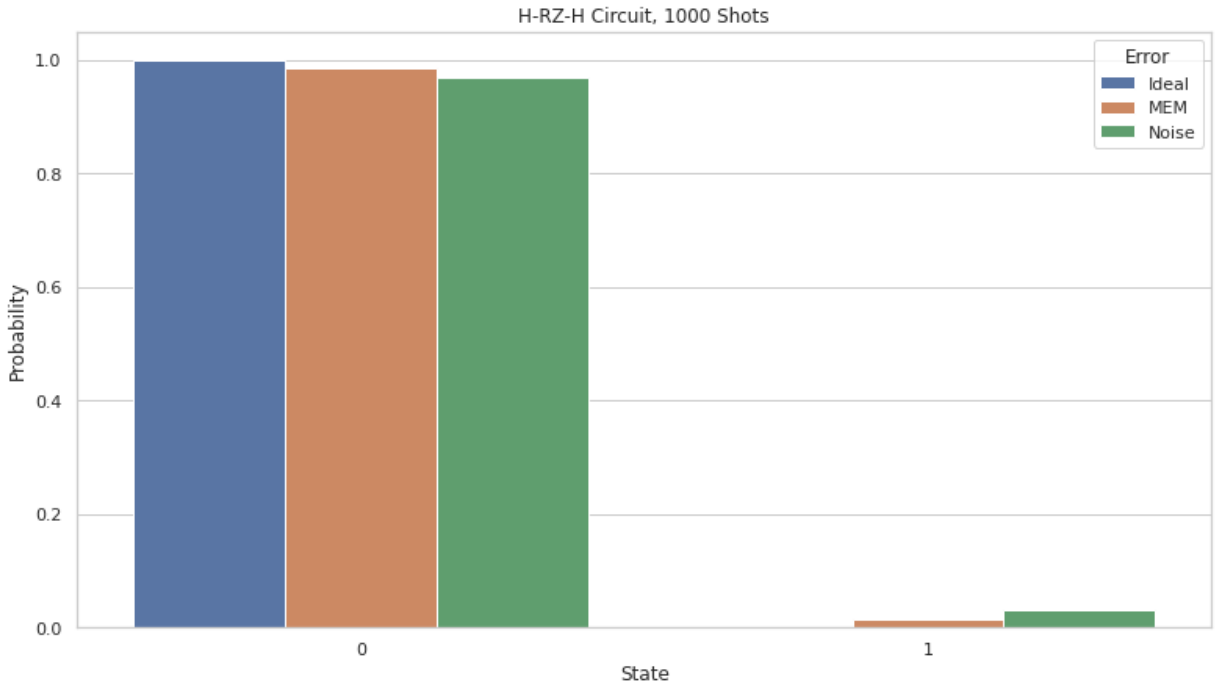

Figure 4.3: Noise influences the outcome distribution. Quantum error mitigation can mitigate this by a certain degree. Here, MEM means measurement error mitigation. By conducting measurement tomography, measurement errors are tried to be mitigated.

4.4 Kullback-Leibler divergence or relative entropy

Kullback-Leibler (KL) divergence is a measure on how two statistical distributions differ. It was introduced in ([36]) and defined by

$$d_{KL}(P|Q) = \sum_{x \in X} P(x) \log(\frac{P(x)}{Q(x)}) \tag{4.14}$$

with P and Q being a probability distribution over a space X. Since this is not yet a distance, it is squared, so that

$$D_{KL}(P|Q) = d_{KL}(P|Q)^2 = (\sum_{x \in X} P(x) \log(\frac{P(x)}{Q(x)}))^2 \tag{4.15}$$

describes the KL distance.

4.5 Influence of noise on quantum circuits

In this chapter, the practical influence of noise on quantum circuits is discussed. For experiments, the quantum simulator *AER* from Qiskit is used. For noise models, the approximation as described in chapter 3.3 for the quantum processor IBM Lima is used.

4.5.1 Virtual Z-Rotation

R_Z transformations inherit a special property in superconducting quantum computers of IBM, which influences the noise of circuits as well. As $R_Z(\pi)R_X(\pi) = iR_Y(\pi)$ and $R_Z(\pi)R_Y(\pi) = -iR_X(\pi)$, R_Z transformations can be implemented by just changing the phase of the other unitaries that are applied. This can be shown formally by looking at an example (cf. [37]). If two R_X microwave pulses are applied, one having a global phase difference ϕ_0, so

$$R_X^{\phi_0}(\theta)R_X(\theta) = e^{-i\frac{\theta}{2}(\cos(\phi_0)\sigma_x + \sin(\phi_0)\sigma_y)}R_X(\theta) \tag{4.16}$$

$$= e^{i\frac{\phi_0}{2}\sigma_z}e^{-i\frac{\theta}{2}\sigma_x}e^{-i\frac{\phi_0}{2}\sigma_z}R_X(\theta) \tag{4.17}$$

$$= R_Z(-\phi_0)R_X(\theta)R_Z(\phi_0)R_X(\theta) \tag{4.18}$$

holds. It means, that is possible to substitute R_Z-gates by applying special global phases to the other unitary transformations. This implies, that the Z rotation gate does neither have error nor does it require any time. It is perfect in that sense. Since any $SU(2)$ gate can be written in the form $U(\phi, \theta, \lambda) = R_Z(\phi - \pi/2)R_X(\pi/2)R_Z(\pi - \theta)R_X(\pi/2)R_Z(\lambda - \pi/2)$, any single-qubit elementary transformation can be efficiently implemented just by using a physical $R_X(\pi/2)$ and using virtual Z-gates. An example are Hadamard gates, which can be rewritten with $H = R_Z(\pi/2)R_X(\pi/2)R_Z(\pi/2)$. Because Z rotation gates can be implemented without any error, Hadamard gates do only inherit the error of the physical

R_X gate in a superconducting quantum computer. Figure 4.4 shows this experimentally by implementing a circuit with different numbers of RZ gates in it on a simulation with a noise model approximating that of the IBM quantum computer IBM Lima. It can be seen, that there is no clear rise of the errors with gates. Additionally, the error vanishes, if measurement error mitigation is applied. That indicates the gate error to be zero and verifies the expectation about the implementation of RZ-gates. Rotations around the Z-axis can be applied without error.

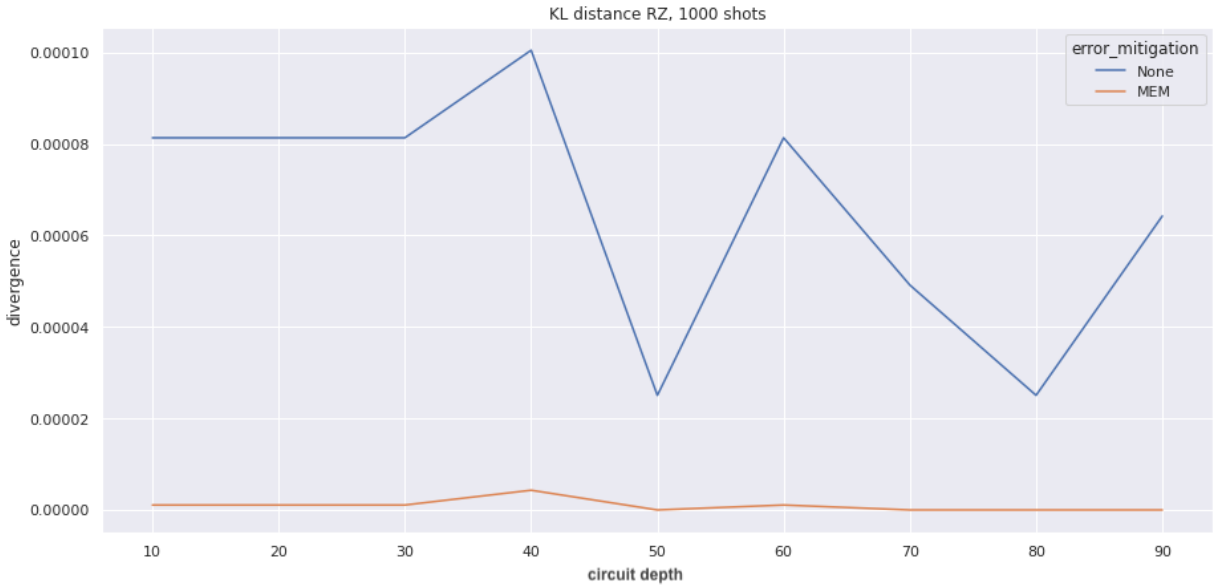

Figure 4.4: In the figure, the KL distance between ideal and noisy simulations of an RZ circuit is shown. This value is plotted against the number of repetitions of RZ gates in the circuit.

4.5.2 Influence of noise and Measurement Error Mitigation on different circuits

Measurement Error Mitigation (MEM) can be also applied to different circuits. To show the effect of noise and its mitigation on quantum circuits, a few experiments are chosen. In the following, comparisons are drawn between ideal and noisy distribution by using the KL distance defined in chapter 4. The *AER* simulator of qiskit in combination with the noise model from the quantum computer IBM Lima is used. In figure 4.5 outcomes are presented of circuits with different repetitions of the SX gate. As expected, the more often the gate is applied, the higher is the error rate of the circuit. Also, measurement error correction can again help to reduce the difference between noisy and ideal distribution. The graphic shows that the error rises also with error mitigation. Since gate errors are accountable for the rise of errors with gates, this is expected. Figure 4.6 and 4.7 verify the observations of figure 4.5. The difference seen in the second diagram however is, that the

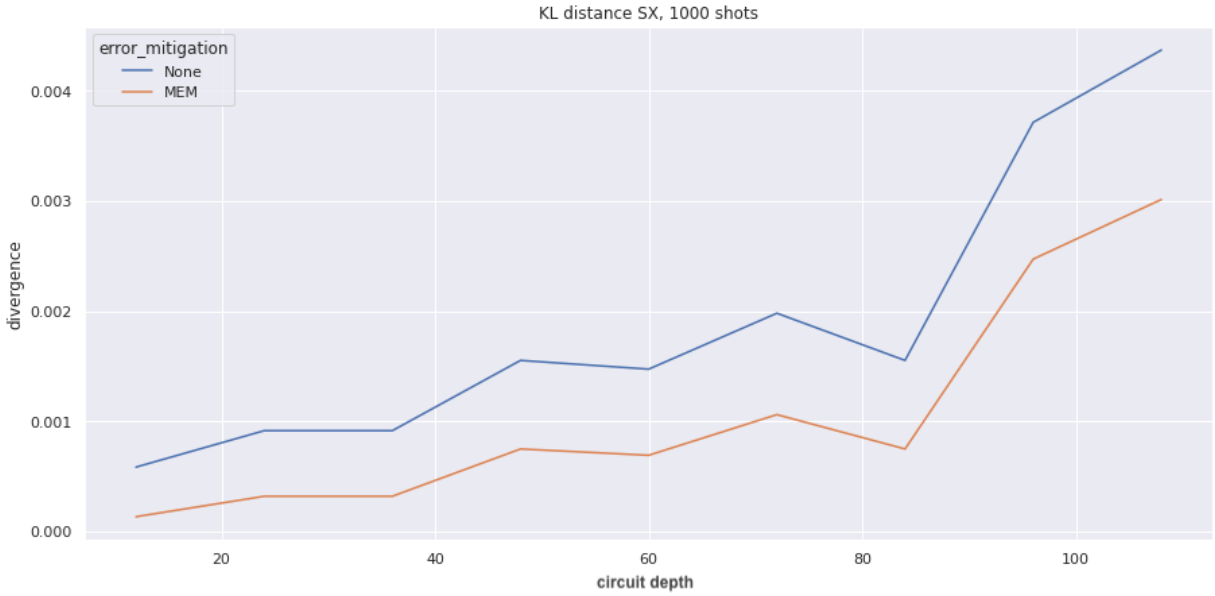

Figure 4.5: The KL distance between ideal and noisy simulations of an SX circuit is shown. It is plotted against the repetition often SX gates in the circuit.

two-qubit operation CX-gate has a far greater error rate as those gates acting only on one qubit.

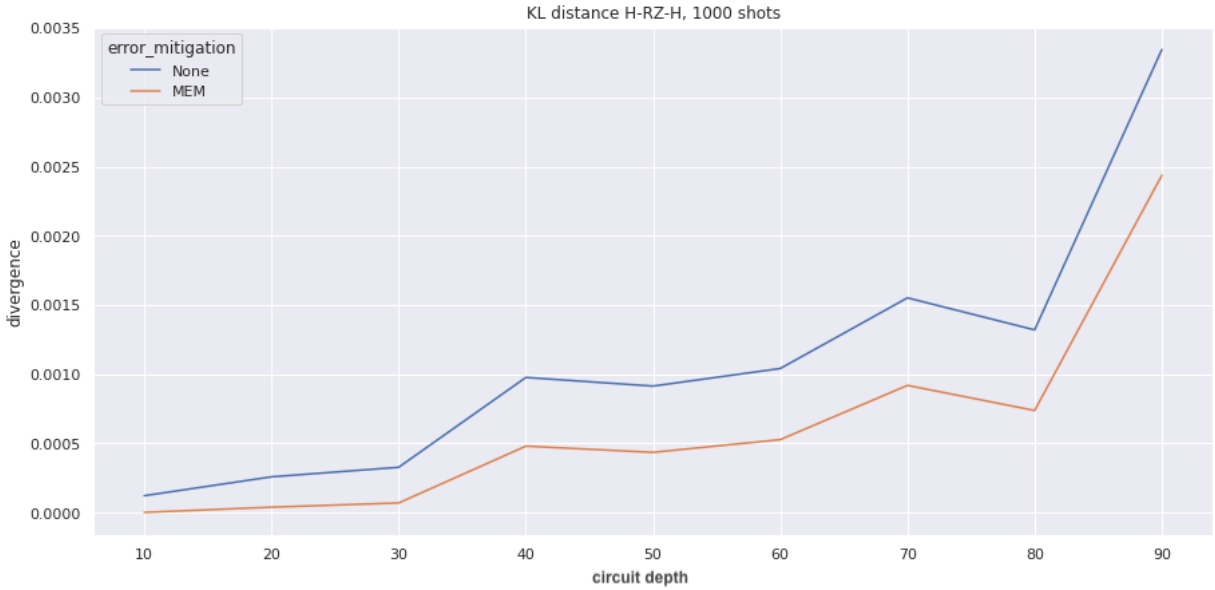

Figure 4.6: In the figure, the KL distance between ideal and noisy simulations of an H-RZ-H circuit is shown. Thereby the value is plotted against the number of repetitions of H-RZ-H gate groups in the circuit.

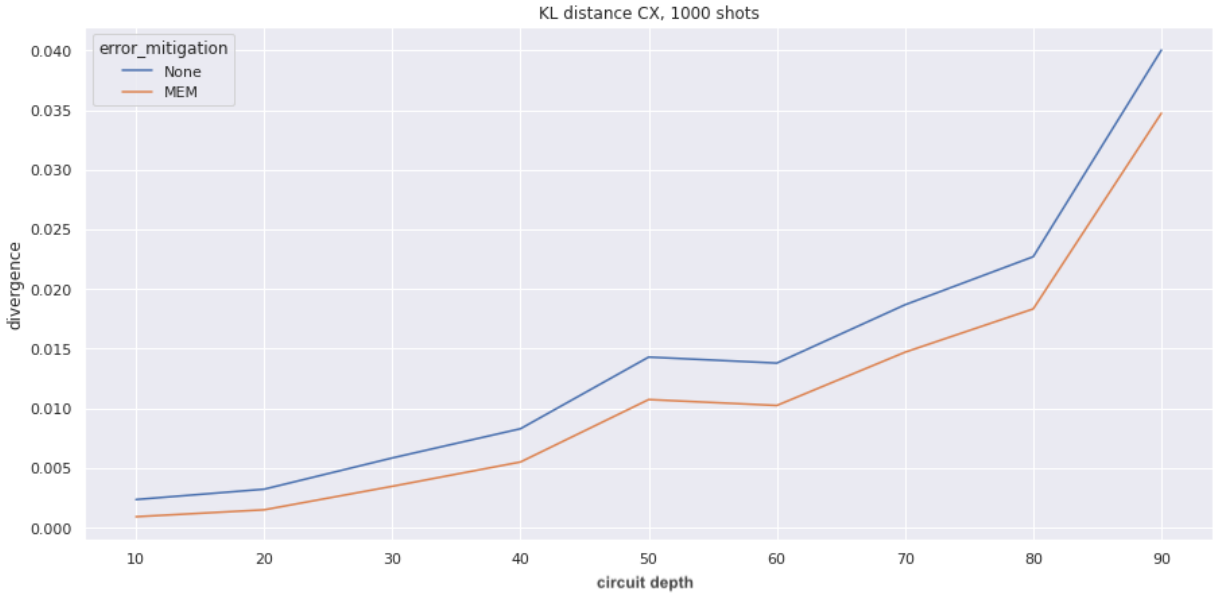

Figure 4.7: In the figure the KL distance between ideal and noisy simulations of a CX circuit is shown. This value is plotted against the number of repetition of CX gates in the circuit.

4.6 Gate Set Tomography

The idea of Gate Set Tomography (GST) is to characterize a gate set up to similarity transformation. It has two major advantages over *normal* Tomography. First, it fixes the circularity, and second, it achieves a higher accuracy with a lower experimental cost. It works by leaving an absolute reference frame and only describing gates relative to each other. This is due to the so-called *gauge freedom* described at the end of chapter 4.1. Though, this degeneracy is also an obstacle to the characterization of quantum computers with GST. Although it fills a significant part of the theory of this work, Gateset Tomography could not be conducted within the given time frame for this thesis. However, it can be easily integrated into the following experiments as gate errors. By performing the gateset circuits on a quantum computer, their gate sets can be evaluated and error channels extracted. Thereafter, readout and state preparation errors can be added to get a complete noise model of the quantum circuit.

4.6.1 Linear Gate Set Tomography

The algorithm of LGST is introduced under the assumption, that it is possible to create informationally complete states and measurement effects. On top of that, sample errors are ignored and it is presumed that matrices A and B from equation 4.8 are square. Definition 4.3 is used to describe a gateset. Gate G_k from the gateset of interest is then represented

by probability matrix P_k:

$$[P_k]_{i,j} = \langle\langle E_i'|G_k|\rho_j'\rangle\rangle \tag{4.19}$$
$$= AG_kB \tag{4.20}$$
$$\tag{4.21}$$

Those probabilities can be measured. Since $A = (\ \langle\langle E_1|\ \ \langle\langle E_2|\ \cdots\ \langle\langle E_{N_{f1}}|\ )$ and $B = (\ |\rho_1\rangle\rangle\ \ |\rho_2\rangle\rangle\ \ \cdots|\rho_{N_{f2}}\rangle\rangle\ )$ are unknown, a new measurement of $\tilde{\mathbb{1}} = AB$ is introduced. $\tilde{\mathbb{1}}$ is called Gram matrix. By linear transformations, this matrix can be used to eliminate A from equation 4.21. Since A and B are invertible and square, the gram matrix is also invertible and it is possible to write 4.21 with $\tilde{\mathbb{1}}^{-1} = B^{-1}A^{-1}$ as

$$\tilde{\mathbb{1}}^{-1}P_k = B^{-1}G_kB, \tag{4.22}$$

and eventually as

$$G_k = B\tilde{\mathbb{1}}^{-1}P_kB^{-1}. \tag{4.23}$$

That means the gate G_k is obtained up to a specific similarity transformation B. If this matrix is the same for all gates of the gatesets, their characteristics are determined relative to each other. It is important to note that $\langle\langle E_i'|$ and $|\rho_j'\rangle\rangle$ do not have to be native. The process is represented in figure 4.8.

4.6.2 Long-sequence GST

Long-sequence GST is more efficient than LGST. It introduces deeper circuits to be more precise and tries to amplify every error. For more in-depth information [26] is recommended.

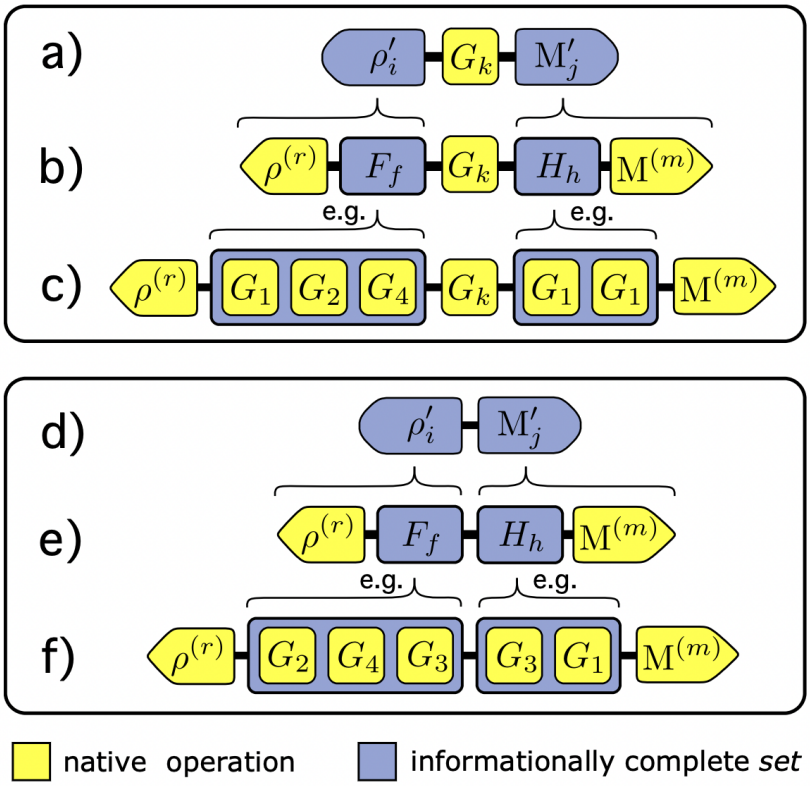

Figure 4.8: Gateset tomography is described here. a)-c): The effective measurements and state preparations contain gates of the gateset as well. d)-f): Because gate set tomography does not assume knowledge of state preparation and measurement, it needs circuits without transformation gates in between to self-calibrate. Taken from [26].

Chapter 5

Characterization of noise in variational algorithms

The goal of this work is to characterize the influence of noise in quantum neural networks. This is a task necessary for the vision of the Fraunhofer Institute for Cognitive Systems to improve tumor diagnostics with hybrid quantum convolutional neural networks. Since these hybrid networks are very computationally intensive, this bachelor thesis focuses solely on its quantum building blocks. The experiments compare ideal quantum simulations with ones with noise. Since the experiments are simulated, it needs a model on which it can be built analytically. This process is described in chapter 3.3. All experiments are conducted using the *Aer* simulator of qiskit. Because the Two-Layer QNN is very similar to the Fraunhofer QNN, not every experiment is executed on both designs. How noise and quantum error mitigation influences the outcome of quantum algorithms is shown in section 4.5. After that, the influence of noise on quantum neural networks and its loss landscape is presented in section 5.3.

5.1 Barren plateaus in the loss landscape of QNN

Barren plateaus are areas in the loss landscape were the gradient is approaching zero. They can be shown numerically by plotting the variance of the gradient of one parameter against the number of qubits of the circuit. An exponential decrease of the variance shows a concentration of measure. This means that the gradient can only deviate from the average exponentially in the qubit number by variable ϵ. The outcomes of paper [38] cannot be recreated with gradient calculation over the trajectory of the autograd feature of PyTorch. In this paper, the mean expectation value of the gradient of the whole loss landscape of the random quantum circuit is argued to be 0. The exponential decrease of the variance of the gradient and thus barren plateaus are showed. However, in figure 5.1, not only the variance of the gradient of the loss landscape with the simulated noise decreases, but the one without does as well.

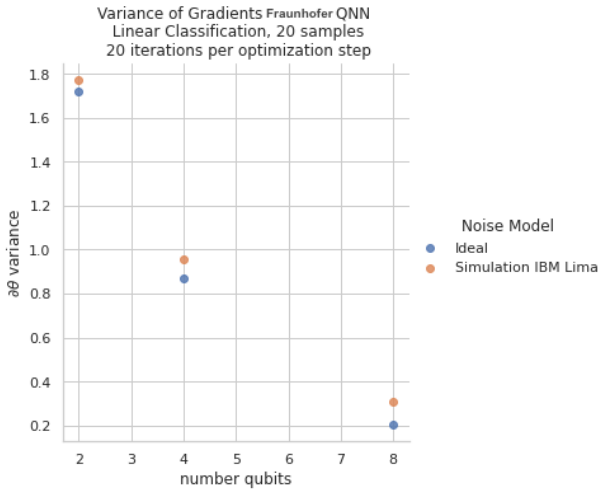

Figure 5.1: Here, the mean variance of the gradient over 20 samples is plotted against number of qubits.

5.2 Influence of noise on the optimization process

Another approach is to look at the loss values in the training process of the neural network. The loss function is the mean-squared error function. The optimizer used is called LBFGS, which is a gradient-based algorithm for searching minimums. Here, figure 5.2 shows different levels of gate errors by plotting their loss value against the optimization step. The initial starting points were chosen randomly and it accounts for very different outcomes. To avoid this incomparability, fixed random starting points were used for all quantum neural networks for the right diagram in 5.2. Noise is modeled here as simple depolarization noise, while its number indicates its strength on 1-qubit gates. 2-qubit gates are assumed to possess a error rate exactly one order higher. As it can be seen, it does not make a difference for the final minimum loss value how strong the gate noise is. To investigate how different types of noise account for the training outcomes of the neural network, graphic 5.3 shows the comparison between the approximated readout and gate errors of IBM Lima (see chapter 3.3). Again, the observation indicates no difference for the training outcome in the used Two Layer QNN structure with 2 qubits. To investigate further, a Two-Layer QNN with a different number of repetitions in its Real Amplitudes Ansatz is used to observe its impact on training results. The minimum loss of the training is plotted against the number of repetitions for readout error and gate error noise model and an ideal simulation in figure 5.4. Again, there is no clear trend observable. The first two experiments were repeated, but now with 100 Samples and 10 repetitions. All loss values are normalized to the first element of the ideal simulation of the repetition, to make different initial starting points comparable. Figure 5.5 shows the outcomes of the Two Layer QNN with approximated noise models and the results with depolarization noise models. As concluded in the first experiments, it can be seen, that the final result does not depend on the error strength. However, it is observed, that quantum neural networks with

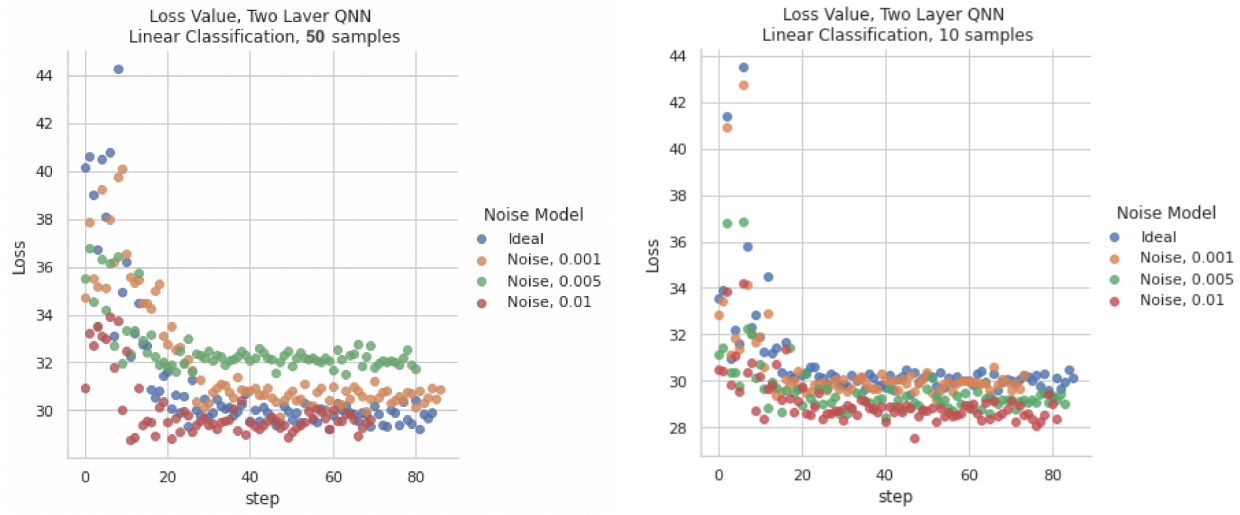

Figure 5.2: In both figures, the loss value of quantum neural networks with different noise models is plotted against the optimization steps. In the graphic on the right, the initial points are different for every noise model. In the one on the right the initial parameters are also random but identical for every neural network with different noise models.

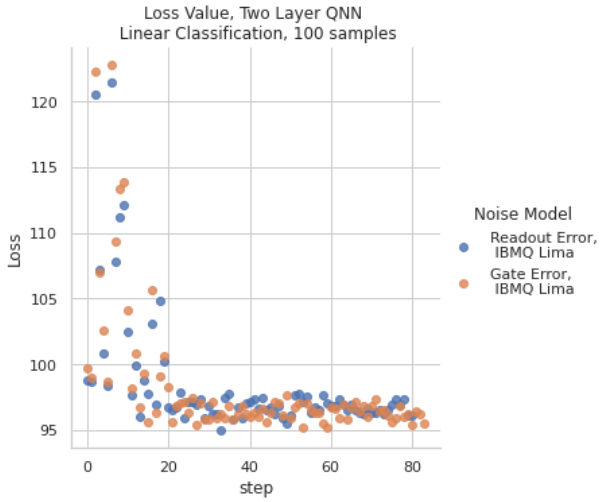

Figure 5.3: In this figure, the loss value of quantum neural networks with different noise models is plotted against the optimization steps.

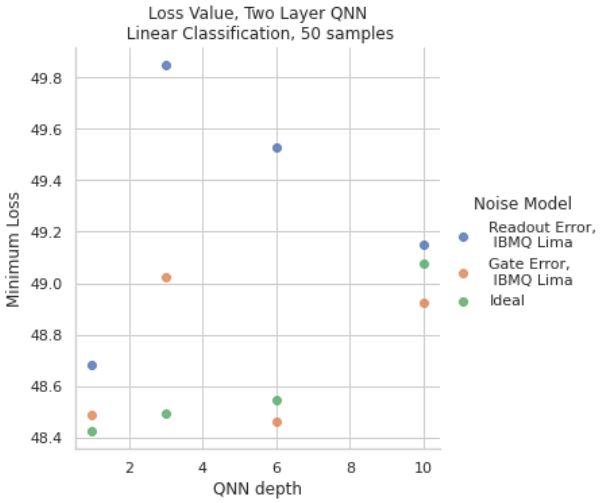

Figure 5.4: The minimum loss value of an optimization process of the Two-Layer QNN with different noise models on a linear classification is depicted.

noise converge quicker than those without. To visualize it better, figures 5.6,5.6,5.7, 5.8, 5.9, 5.10, 5.11 and 5.12, 5.13, 5.14 , 5.15, 5.16, 5.17 show the density of the occurrences of loss values at different optimization steps of the noise models associated with IBM Lima and the different depolarization noises.

5.3 Loss Landscapes of quantum neural networks

In this chapter, the two quantum neural network designs described in chapter are evaluated. To visualize the loss landscape of quantum neural networks, a minimum is searched by the autograd function of PyTorch ([39]) with 20 iteration steps. After this, the model parameters are perturbed in two random directions to reduce the dimensionality. Figure 5.18 and 5.19 shows the outcomes. It can be seen, that noise shrinks the loss landscape in its loss dimension. This is consistent with paper [40].

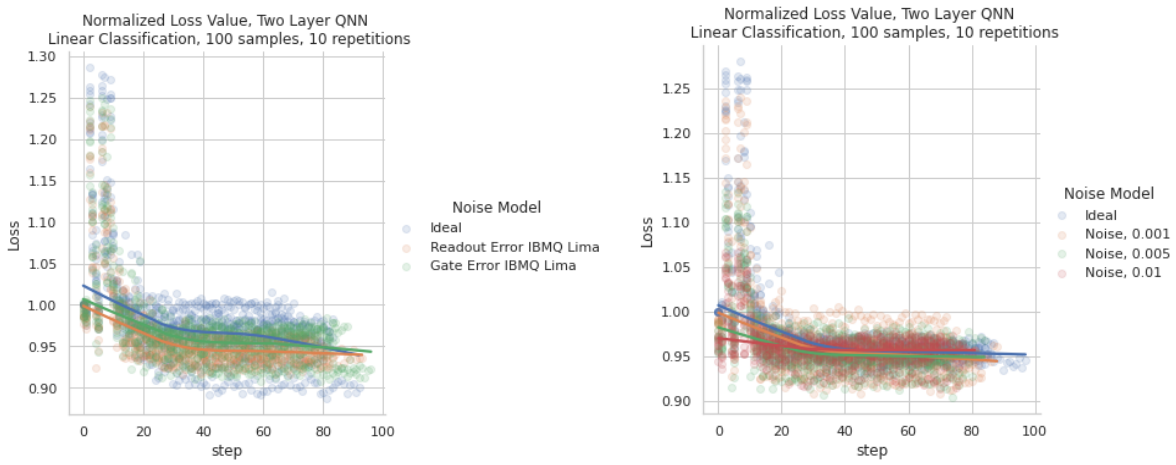

Figure 5.5: The loss value trends of 10 optimization process of the Two-Layer QNN with different noise models on a linear classification with 100 samples is showed.

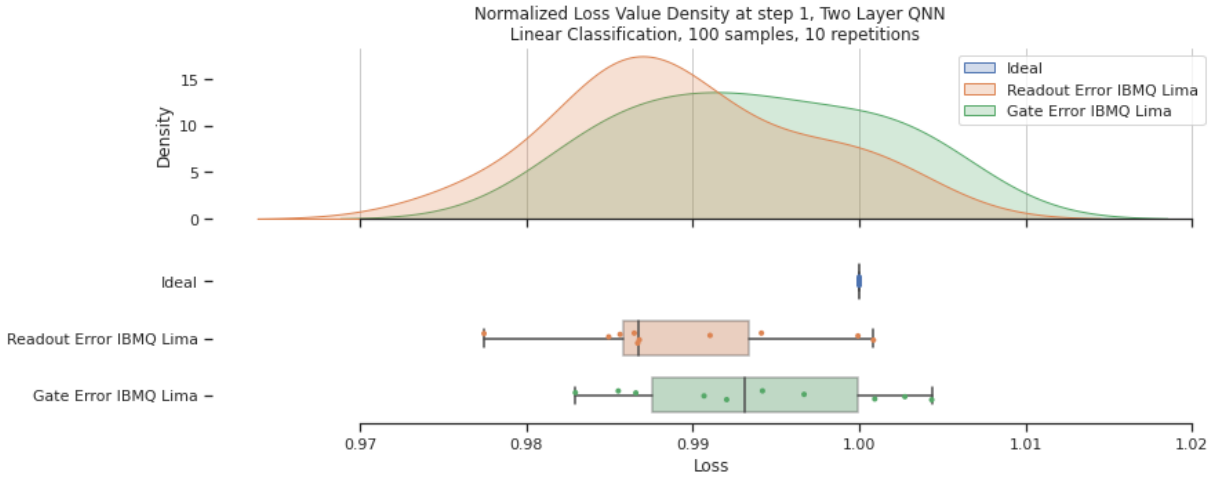

Figure 5.6: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 1. The compared noise models are the ones approximated to the real ones of IBM Lima.

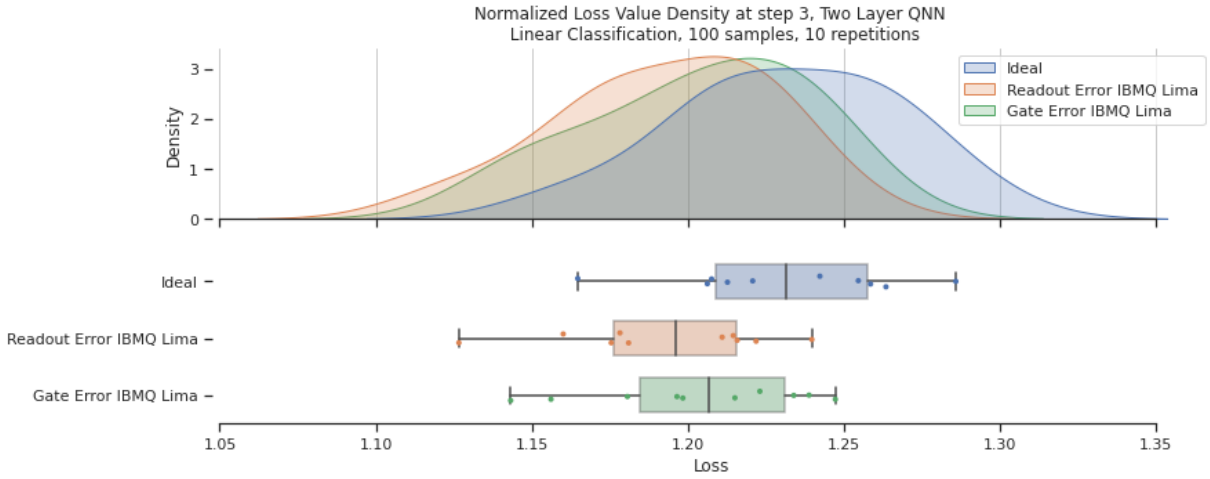

Figure 5.7: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 3. The compared noise models are the ones approximated to the real ones of IBM Lima.

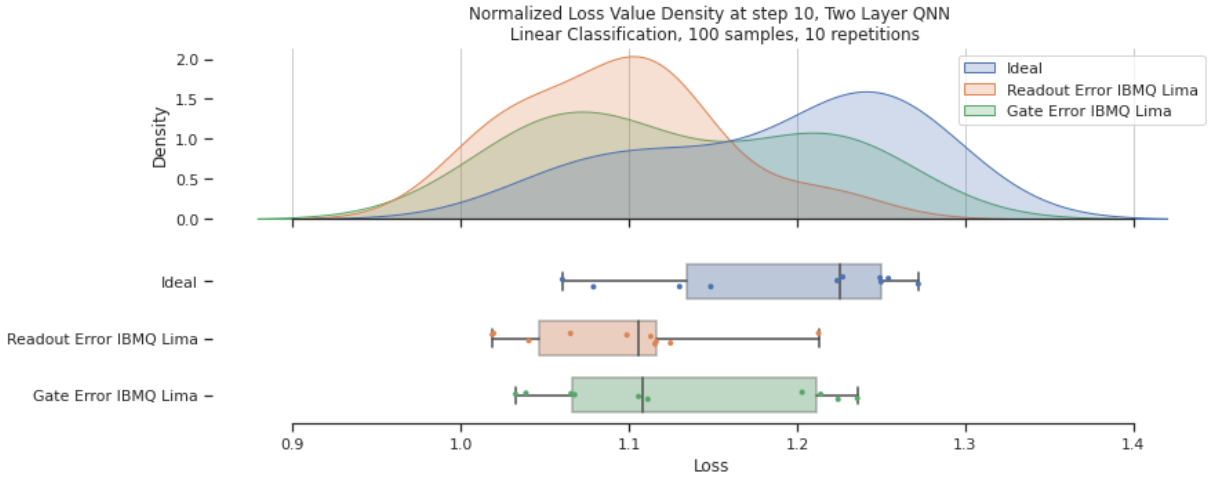

Figure 5.8: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 10. The compared noise models are the ones approximated to the real ones of IBM Lima.

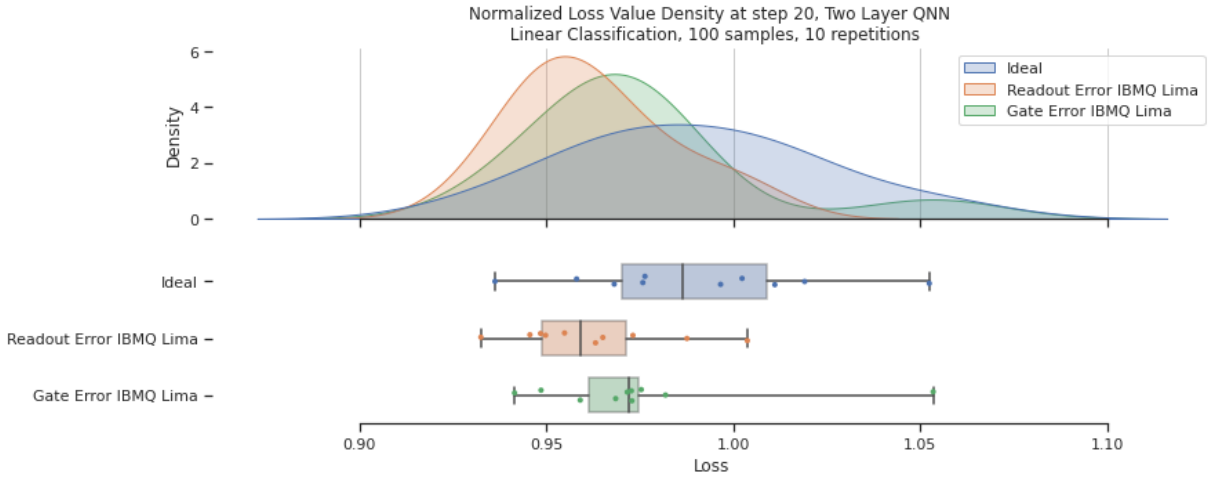

Figure 5.9: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 20. The compared noise models are the ones approximated to the real ones of IBM Lima.

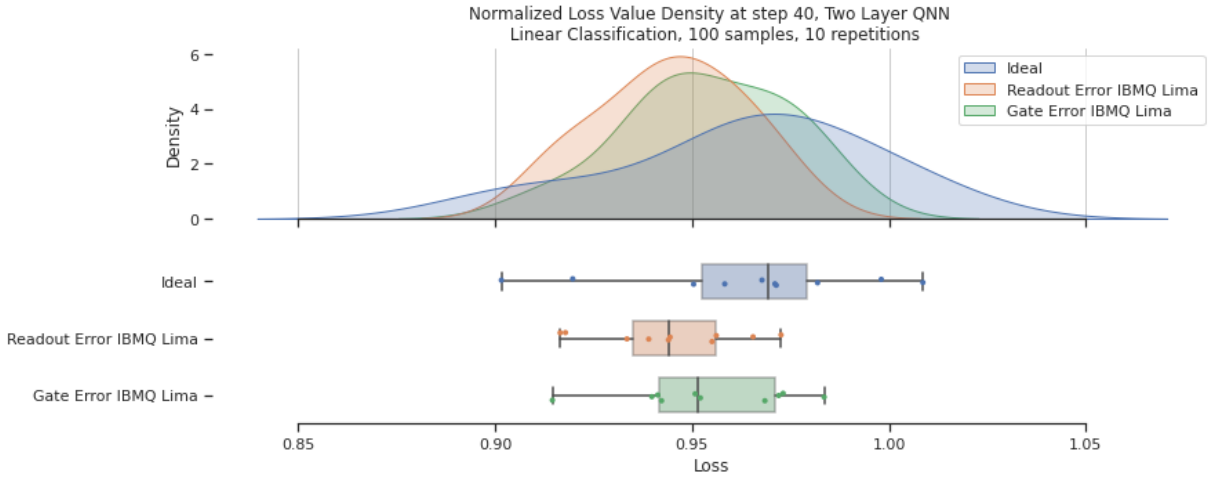

Figure 5.10: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 40. The compared noise models are the ones approximated to the real ones of IBM Lima.

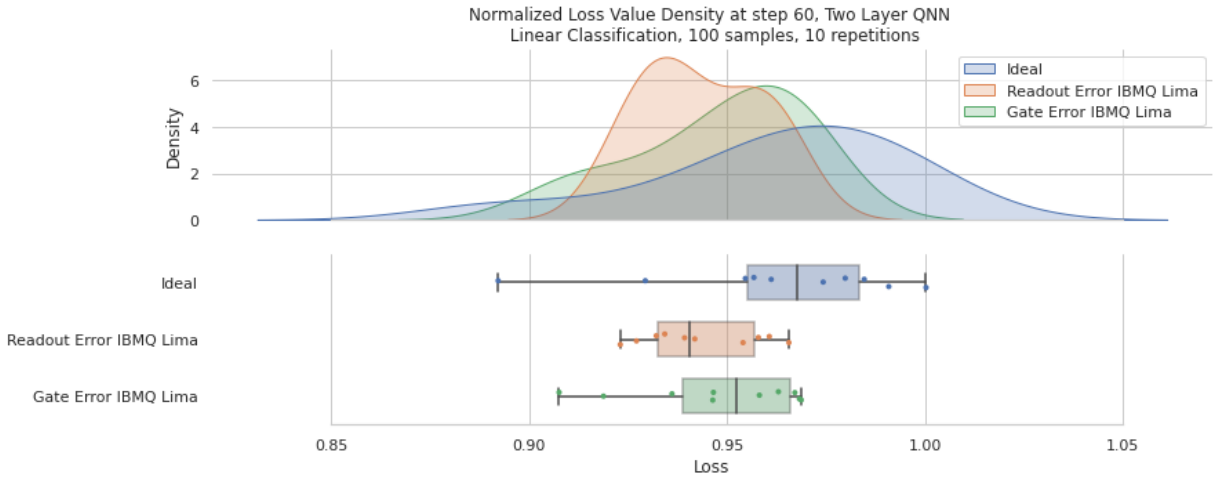

Figure 5.11: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 60. The compared noise models are the ones approximated to the real ones of IBM Lima.

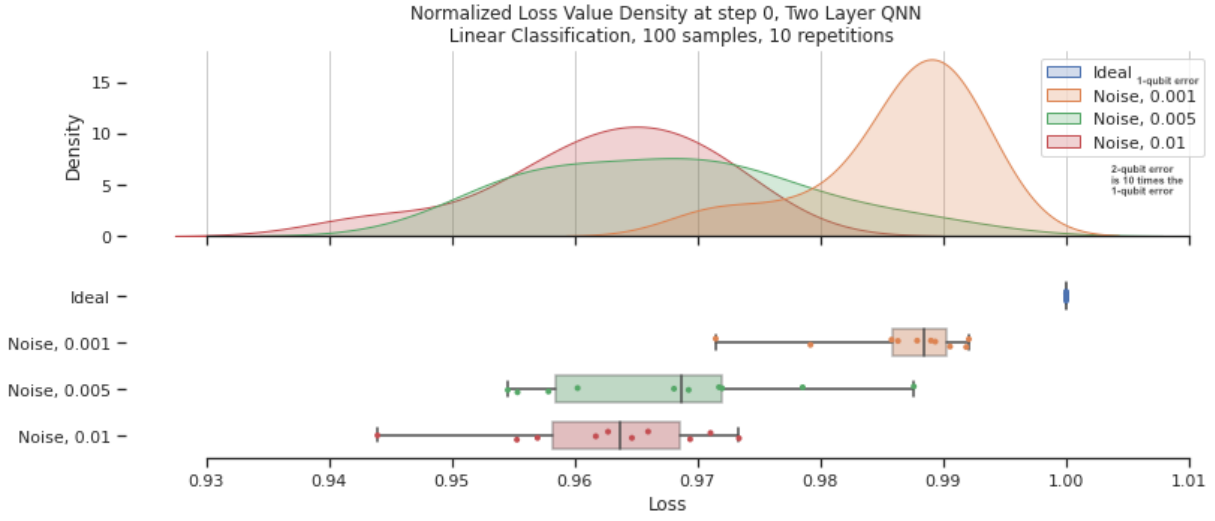

Figure 5.12: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 1. The compared noise models are depolarization noises with different strength. Hereby, a 2-qubit gate has a error rate 10 times bigger than the 1-qubit gate (the number in the legend).

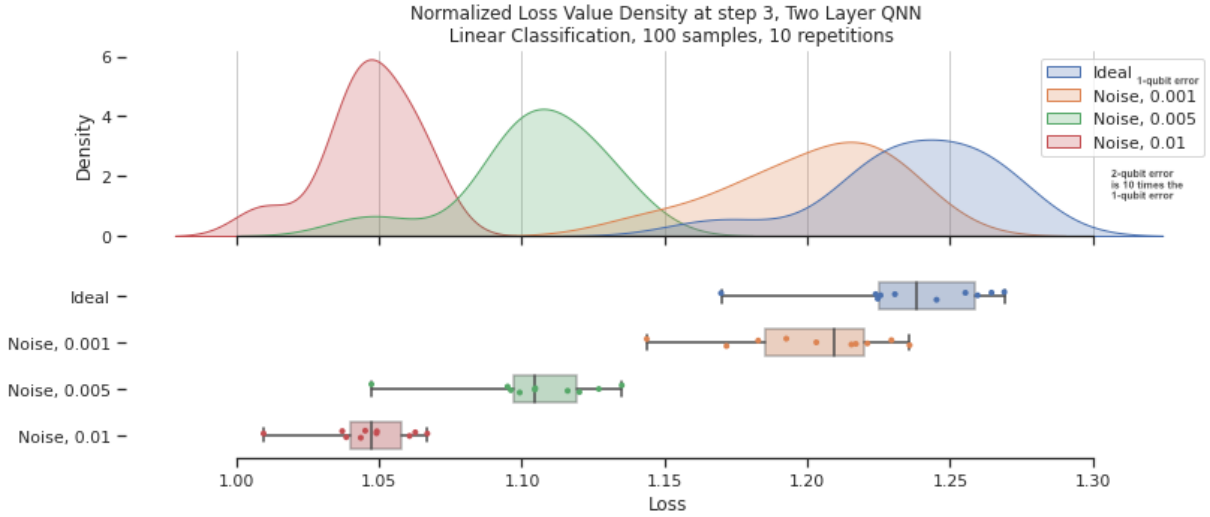

Figure 5.13: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 3. The compared noise models are depolarization noises with different strength. Hereby, a 2-qubit gate has a error rate 10 times bigger than the 1-qubit gate (the number in the legend).

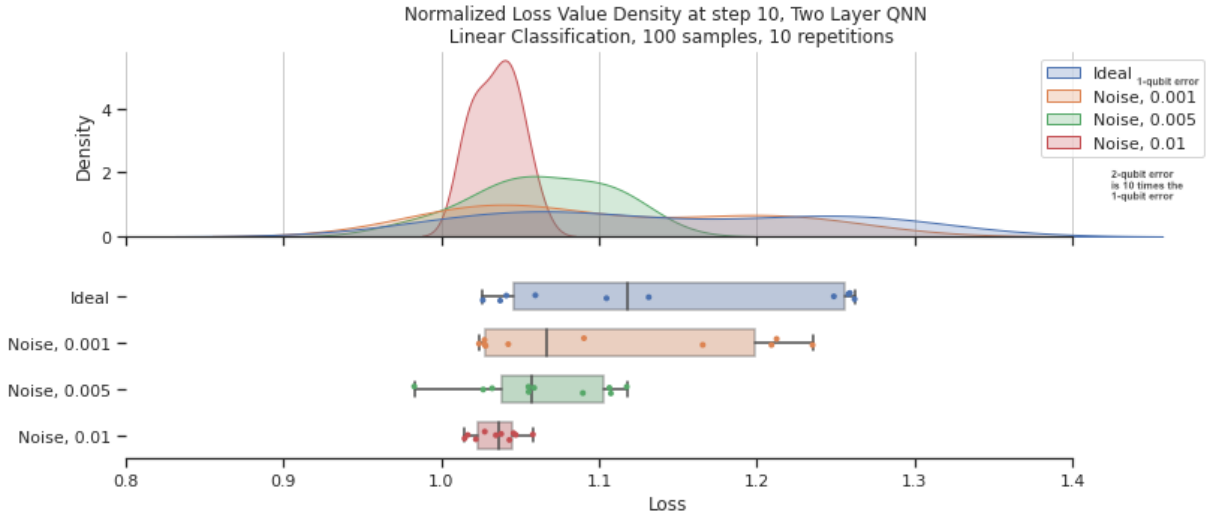

Figure 5.14: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 10. The compared noise models are depolarization noises with different strength. Hereby, a 2-qubit gate has a error rate 10 times bigger than the 1-qubit gate (the number in the legend).

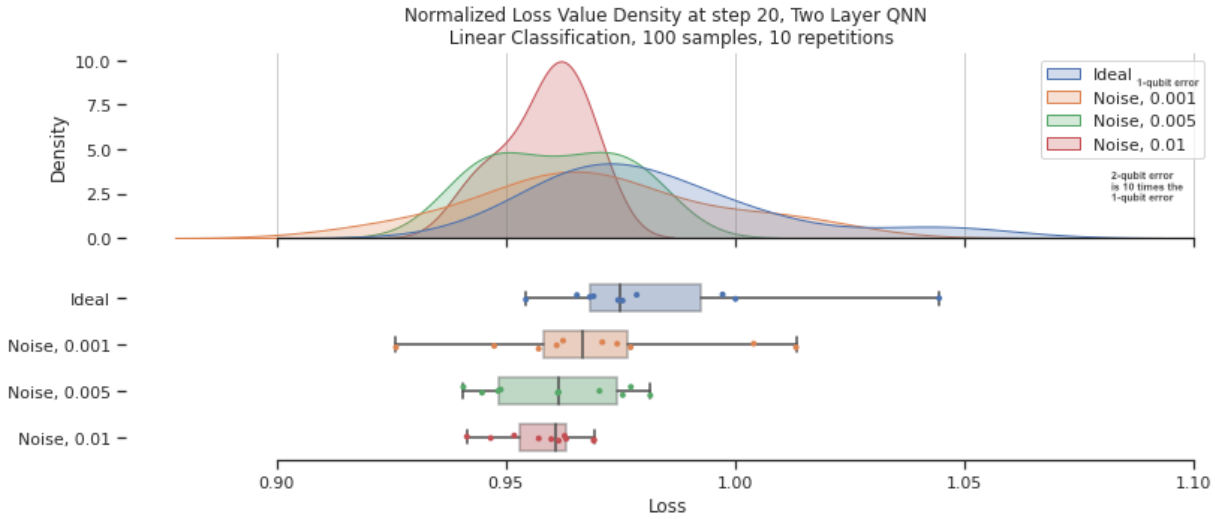

Figure 5.15: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 20. The compared noise models are depolarization noises with different strength. Hereby, a 2-qubit gate has a error rate 10 times bigger than the 1-qubit gate (the number in the legend).

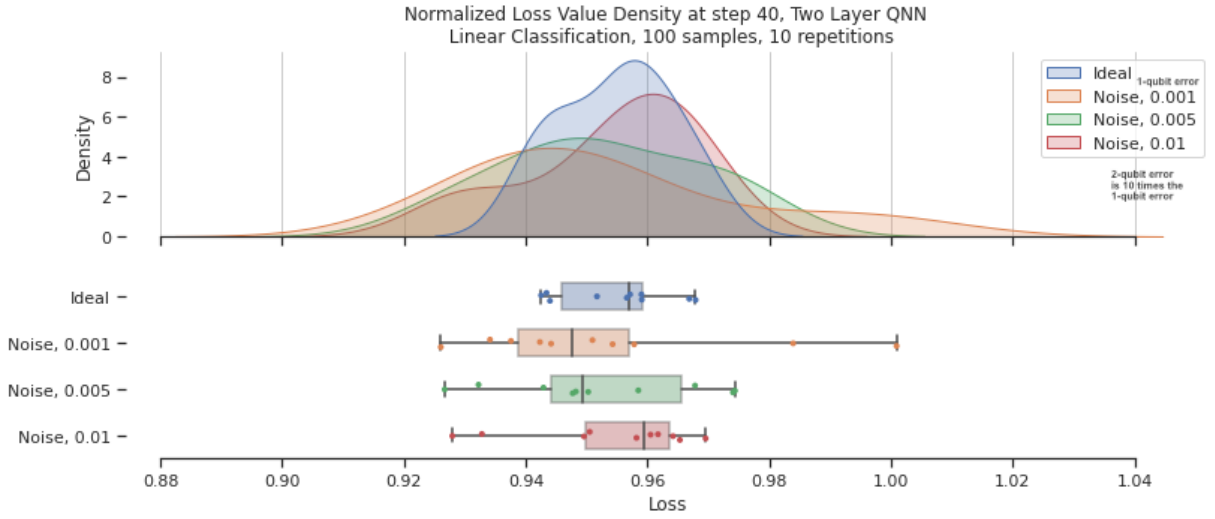

Figure 5.16: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 40. The compared noise models are depolarization noises with different strength. Hereby, a 2-qubit gate has a error rate 10 times bigger than the 1-qubit gate (the number in the legend).

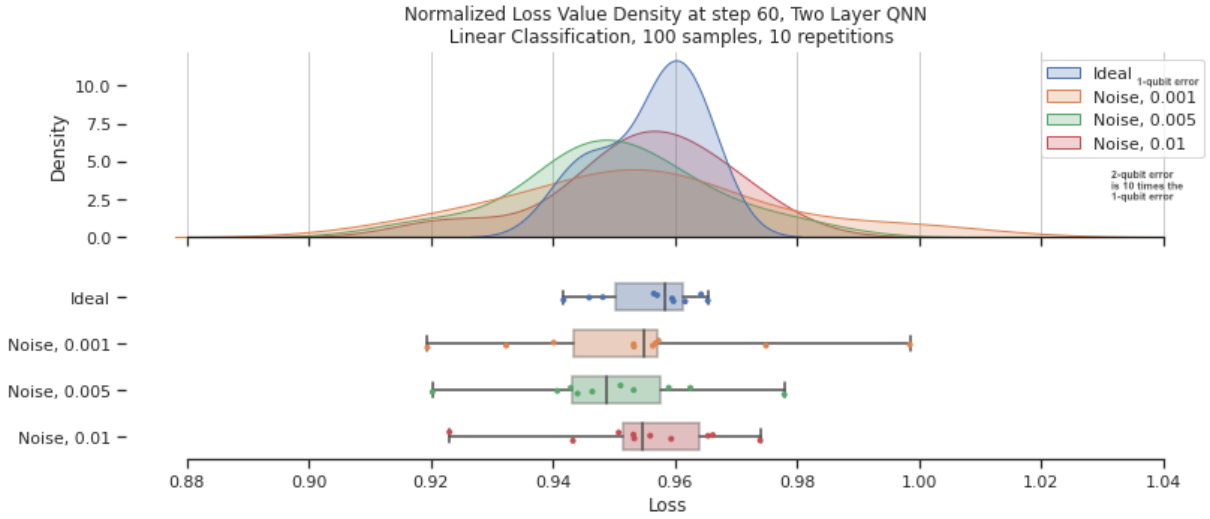

Figure 5.17: The loss values of 10 optimization processes of the Two-Layer QNN with different noise models on a linear classification with 100 samples are depicted at step 60. The compared noise models are depolarization noises with different strength. Hereby, a 2-qubit gate has a error rate 10 times bigger than the 1-qubit gate (the number in the legend).

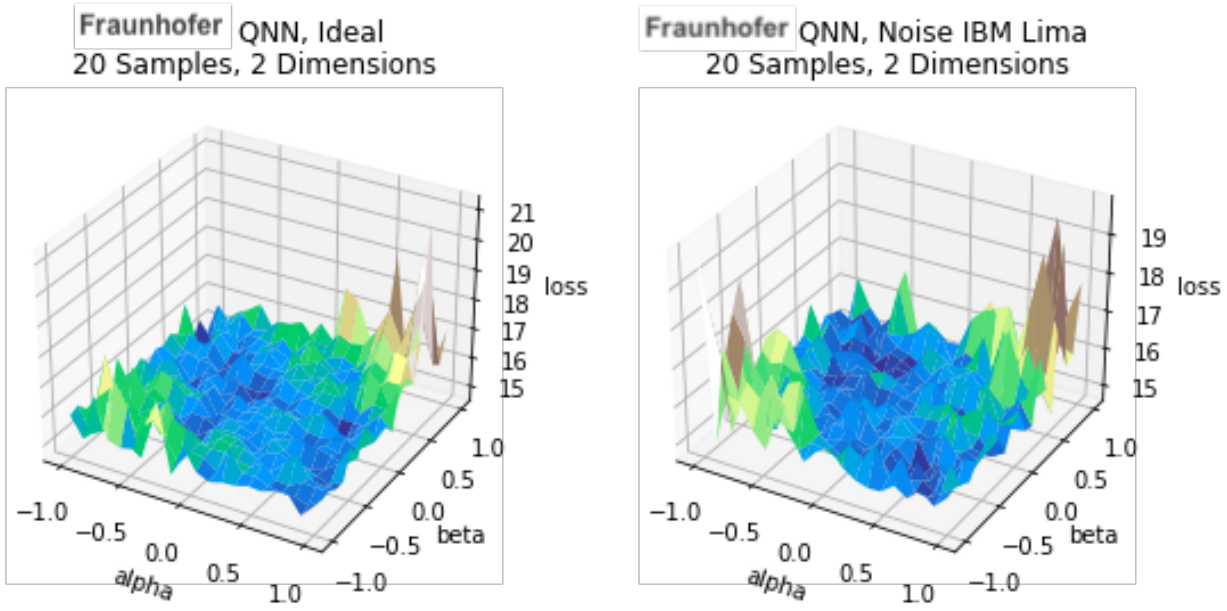

Figure 5.18: Here the loss landscape of the Fraunhofer QNN with and without noise is depicted.

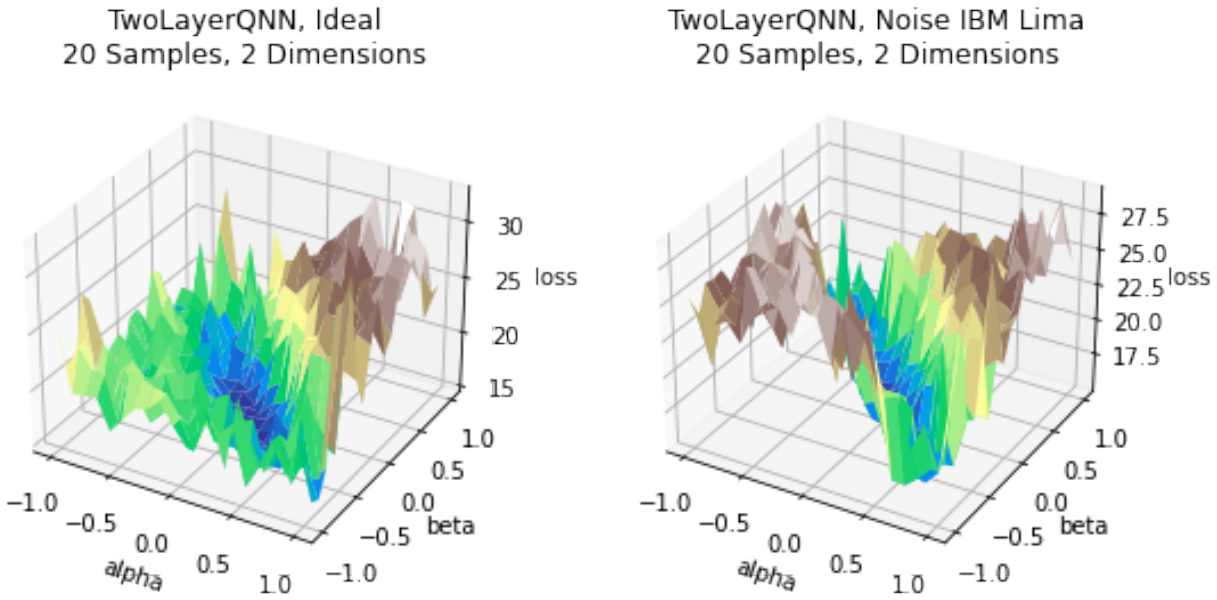

Figure 5.19: In this figure, the loss landscape of the Two-Layer QNN with and without noise is depicted.

Chapter 6

Interpretation

There are two different outcomes of the experiments. On the one hand, the variance of the gradient did not clearly show the existence of general barren plateaus like those shown in [38]. The decrease in height of the loss landscape, however, is an indicator of the development of noise- induced barren plateaus with more qubits and is consistent with the paper of Samson Wang and others (see [40]). On the other hand, the observations show, that up to a specific optimization step linear classification with Two-Layer QNN with noise might converge more rapidly than without. It does possess statistical robustness because it is repeated multiple times. This is surprising, since noise-induced barren plateaus were expected, which have exactly the opposite effect on the training process. The advantage of noise could be because a neural network with more noise has more ability to search the loss landscape for good loss values than those without. If the ideal quantum neural network is stuck in a local minimum, for example, it cannot escape it through normal gradient descent. It needs some kind of random perturbation. This is introduced either by the optimization algorithm itself or occurs naturally by noise. To make the loss function decrease quicker, the experiment could need perturbation to converge to a better local or the global minimum of the problem like shown in figure 6.1. It seems that random perturbation introduced by quantum noise is helpful for the optimization with the LBFGS optimizer. As a result, it could be favorable to use a quantum neural network with noise instead of mitigating the noise. This, however, needs further testing to be confirmed or even generalized.

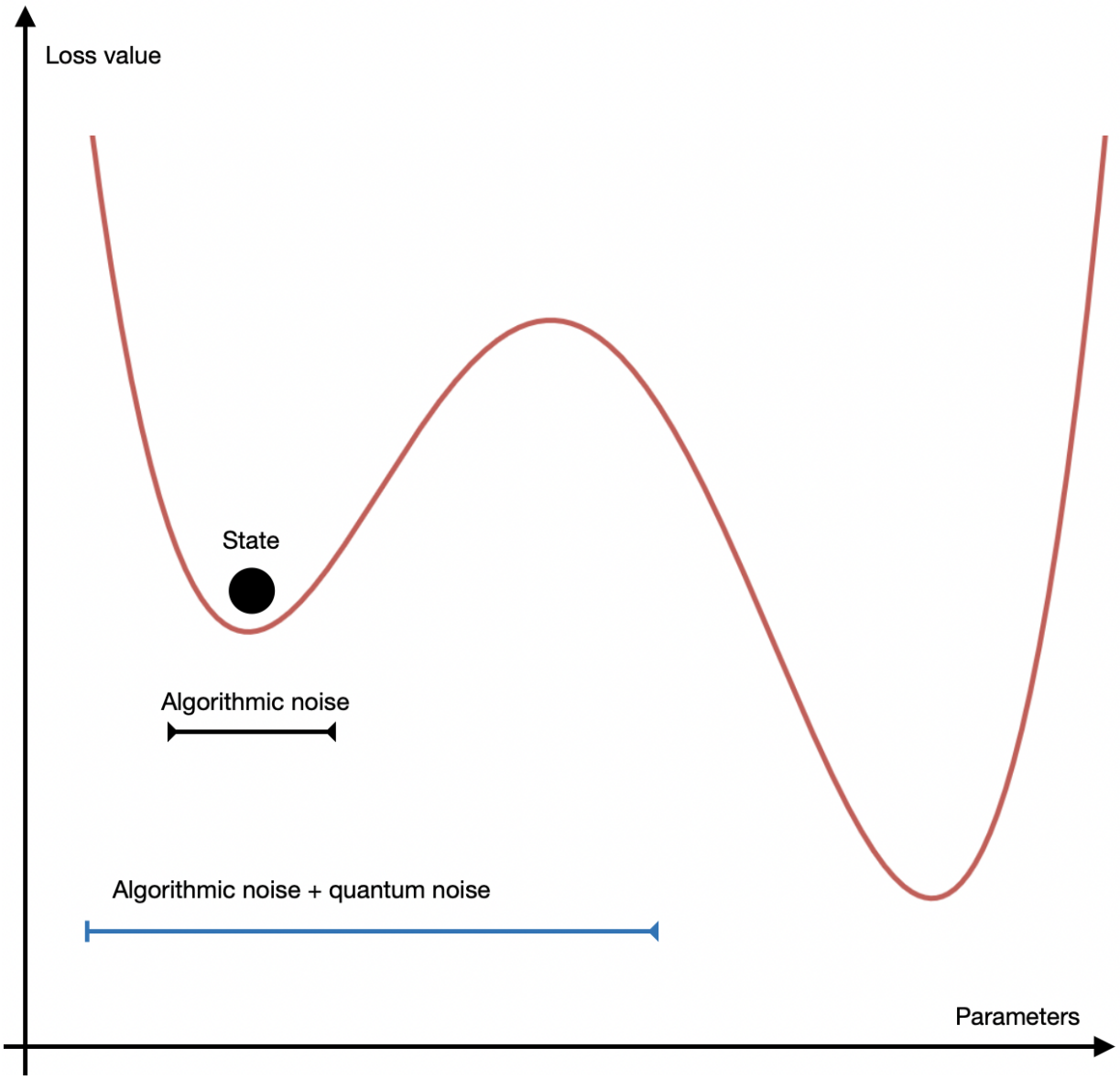

Figure 6.1: If the optimizer is stuck in a local minimum, quantum noise could introduce the perturbation needed to find the global minimum more quickly.

Chapter 7

Conclusion

This thesis aimed to evaluate and analyze the influence of noise on quantum neural networks. Although the experimental setting limits the generality, it was shown, that noise could improve the training procedure in some cases. The shrinking height of the loss landscape indicates the emergence of noise-induced barren plateaus with more complex architectures. However, this did not affect the training of a quantum neural network with 2 qubits in a negative way. Using the characterized influence of noise on quantum neural networks either by mitigating it or using it for perturbation could help approaching better image detection of tumors through quantum-classical convolutional neural networks. To characterize the influence of noise on quantum-classical convolutional neural networks further research should be conducted with the hybrid quantum algorithm. Additionally, the phenomena of beneficial quantum noise could be analyzed more deeply and theoretically proven.

Bibliography

[1] Sukhpal Singh Gill, Adarsh Kumar, Harvinder Singh, Manmeet Singh, Kamalpreet Kaur, Muhammad Usman, and Rajkumar Buyya. Quantum computing: A taxonomy, systematic review and future directions. *Software: Practice and Experience, Wiley Press, USA, Sept. 22, 2021*, September 2020.

[2] Maria Schuld and Francesco Petruccione. *Machine Learning with Quantum Computers*. Springer International Publishing, 2021.

[3] Eleanor G. Rieffel Wolfgang H. Polak. *Quantum Computing*. MIT Press Ltd, August 2014.

[4] David Griffiths. *Introduction to quantum mechanics*. Pearson Prentice Hall, Upper Saddle River, NJ, 2005.

[5] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, jun 1982.

[6] Qiskit Team. Qiskit summerschool, 2021.

[7] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nat Comput Sci 1, 403-409 (2021)*, October 2020.

[8] Pennylane. Key concepts qml, 2022´.

[9] John Preskill. Quantum computing in the nisq era and beyond. *Quantum 2, 79 (2018)*, January 2018.

[10] Michael Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, 2015.

[11] Miroslav Kubat. *An introduction to machine learning*. Springer, 2017.

[12] Picture neural network wikipedia.

[13] Matthias C. Caro, Hsin-Yuan Huang, M. Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J. Coles. Generalization in quantum machine learning from few training data. November 2021.

[14] B. E. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133–137, may 1998.

[15] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland. Quantum dynamics of single trapped ions. *Reviews of Modern Physics*, 75(1):281–324, mar 2003.

[16] R. Blatt and C. F. Roos. Quantum simulations with trapped ions. *Nature Physics*, 8(4):277–284, apr 2012.

[17] Xi-Lin Wang, Yi-Han Luo, He-Liang Huang, Ming-Cheng Chen, Zu-En Su, Chang Liu, Chao Chen, Wei Li, Yu-Qiang Fang, Xiao Jiang, Jun Zhang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. 18-qubit entanglement with six photons' three degrees of freedom. *Physical Review Letters*, 120(26):260502, jun 2018.

[18] He-Liang Huang, Dachao Wu, Daojin Fan, and Xiaobo Zhu. Superconducting quantum computing: A review. *Science China Information Sciences 63 (8), 1-32 (2020)*, June 2020.

[19] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, jun 2019.

[20] J. Q. You and Franco Nori. Superconducting circuits and quantum information. *Phys. Today 58(11), 42 (2005)*, January 2006.

[21] Qiskit, open-source sdk for working with quantum computers.

[22] Ibm quantum computing, 2021.

[23] B.D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251–253, jul 1962.

[24] Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen. Building logical qubits in a superconducting quantum computing system. *npj Quantum Information*, 3(1), jan 2017.

[25] Jay Gambetta. Ibm roadmap quantum computer, 2020.

[26] Erik Nielsen, John King Gamble, Kenneth Rudinger, Travis Scholten, Kevin Young, and Robin Blume-Kohout. Gate set tomography. *Quantum*, 5:557, oct 2021.

[27] Qiskit Team. Textbook, 2022.

[28] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. SPRINGER NATURE, February 2007.

[29] Fen Zhao Isaac Chuang. How to build your own quantum computer. *MIT lecture*, 2003.

[30] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. Randomized benchmarking of quantum gates. *arXiv*, July 2018.

[31] Robin Blume-Kohout, John King Gamble, Erik Nielsen, Kenneth Rudinger, Jonathan Mizrahi, Kevin Fortier, and Peter Maunz. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nature Communications*, 8(1), feb 2017.

[32] Tudor Giurgica-Tiron, Yousef Hindy, Ryan LaRose, Andrea Mari, and William J. Zeng. Digital zero noise extrapolation for quantum error mitigation. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 2020*, May 2020.

[33] Ewout van den Berg, Zlatko K. Minev, Abhinav Kandala, and Kristan Temme. Probabilistic error cancellation with sparse pauli-lindblad models on noisy quantum processors. January 2022.

[34] Piotr Czarnik, Andrew Arrasmith, Patrick J. Coles, and Lukasz Cincio. Error mitigation with clifford quantum-circuit data. *Quantum 5, 592 (2021)*, May 2020.

[35] Sergey Bravyi, Sarah Sheldon, Abhinav Kandala, David C. Mckay, and Jay M. Gambetta. Mitigating measurement errors in multi-qubit experiments. *Phys. Rev. A 103, 042605 (2021)*, June 2020.

[36] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, mar 1951.

[37] David C. McKay, Christopher J. Wood, Sarah Sheldon, Jerry M. Chow, and Jay M. Gambetta. Efficient z gates for quantum computing. *Physical Review A*, 96(2):022330, aug 2017.

[38] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications, Volume 9, Article Number: 4812 (2018)*, March 2018.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[40] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12(1), nov 2021.

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

München, 08. April 2022
Philip Hierhager