SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Influence of Noise in Quantum Machine Learning

Erick Ruben Quintanar Salas

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Influence of Noise in Quantum Machine Learning

Einfluss von Rauschen in Quantenmaschinen Lernen

Author: Erick Ruben Quintanar Salas
Supervisor: Prof. Dr. Claudia Eckert
Advisors: M. Sc. Pascal Debus / M. Sc. Kilian Tscharke
Submission Date: Submission date

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, Submission date Erick Ruben Quintanar Salas

Acknowledgments

Abstract

Contents

Acknowledgments **iii**

Abstract **iv**

1 Introduction **1**
1.1 Motivation 1
1.2 Research Goals 2
1.3 Outline 3

2 Theoretical Background **4**
2.1 Fundamentals of Quantum Computing 4
2.1.1 Qubit 4
2.1.2 Bloch Sphere 5
2.1.3 Multiple qubits 6
2.1.4 Quantum Gates 7
2.1.5 Entanglement 11
2.1.6 Quantum Measurement 12
2.1.7 Quantum Circuit 15
2.1.8 Density Operator 15
2.2 Quantum Noise 15
2.3 Quantum Machine Learning 15
2.4 Adversarial Machine Learning 16

3 Implementation **17**
3.1 Section 17

4 Style **18**
4.1 Section 18
4.1.1 Subsection 18

Abbreviations **20**

List of Figures **21**

List of Tables **22**

Bibliography **23**

1 Introduction

In recent years the interest on techniques to utilize quantum mechanics has been rising. One of the many applications is quantum computing, where devices based on the laws of quantum theory are exploited to process information [1]. Although current classical computers have become very powerful, they still struggle to process many applications that quantum computers can in theory easily solve.

Many quantum algorithms for quantum computers have been proposed that highly outperform a classical computer with the best known algorithms [2, 3, 4]. These quantum algorithms solve in polynomial time problems that quickly become intractable to solve in a classical computer, as they normally grow exponentially. The most famous algorithm is Shor's algorithm [2], which can find the prime factors of an integer. It is of special interest because if quantum devices were able to execute it, the current confidentiality and integrity guarantees that the RSA [5] cryptographic mechanism offers would be violated.

1.1 Motivation

Even though the previously mentioned quantum algorithms would surpass the performance of the best classical ones, they still can not be executed in current quantum computers due to noise [6]. This noise occurs because current quantum devices are not completely isolated from the environment and every time we perform an operation on them we introduce a disturbance. This type of device is known as Noisy Intermediate-Scale Quantum (NISQ), meaning that there will be significant noise when operating the quantum device.

In order to reduce the influence of noise in NISQ devices, either the precision in which quantum computers can be manipulated has to improve or error-correcting codes have to be implemented [7]. Currently both techniques are being heavily researched and in conjunction will lead to the next generation of quantum devices, namely fault-tolerant quantum computers.

A technology that right now has gained a bigger presence in our society is Machine Learning (ML). There have been many important breakthroughs for ML in the past few

years, with uses in natural language processing, computer vision, anomaly detection, and many more fields [8]. Nowadays ML has a big impact in society, and even though it depicts big opportunities for improvement in society it also represents significant risks.

Quantum computing and ML are information processing techniques that have improved significantly in recent years. This lead to the natural desire of harnessing the advantages of both and to the emergence of a new field of study denominated Quantum Machine Learning (QML) [9]. QML explores several ideas like whether quantum devices are better at ML than classical computers or if quantum information adds new data that affects how machines recognize patterns.

Returning to the possible risks that ML might encounter, several attacks have been developed to force a ML model to missclassify an input [10]. These attacks are denominated adversarial attacks and are based on crafting specific input data that has been slightly modified to cause the model to erroneously classify the input. At the beginning, when the first adversarial attacks were developed, they needed to know the architecture of the model to be able to fool it. Nevertheless, it was proved that adversarial attacks are transferable between models with the same use case, without knowing the architecture of the model or the dataset it was trained on [11].

Adversarial training was developed in order to defend ML models against adversarial attacks [12, 10]. Adversarial training consists of including adversarial samples into the training of the ML model to better generalize its classification. This mechanism has a tradeoff, in which the accuracy of the model lowers, while increasing the resilience to adversarial attacks [13].

In classical ML noise in training has been shown to improve generalization performance and local optima avoidance [14]. This property from noise is particularly interesting in NISQ devices, as their inherent noise might be able to improve QML performance, accuracy and resilience against adversarial attacks.

TODO: - Could we claim that noise in QML might also be a defense against data poisoning attacks? (AML at scale) [13]

1.2 Research Goals

1. Test the effect of different types of noise in QML regarding robustness. 2. Test the effect of noise in different parts of the QML circuits. 3. Test the effect of noise between VQA and Kernel methods. 4. Test the models with different types of adversarial attacks (FGSM, CaW, PGD)

1.3 Outline

Write here what is the general structure of the thesis.

2 Theoretical Background

In Chapter 2 we will introduce the background information that is required to understand the main ideas of this thesis. First we introduce the basic concepts of quantum computing. Then we will describe advanced concepts regarding quantum noise. We assume some baseline ML knowledge, however, we will provide an outlook into QML. Finally, we present several types of adversarial machine learning and adversarial training as a defense mechanism.

2.1 Fundamentals of Quantum Computing

2.1.1 Qubit

The basic computing unit in quantum computing is the *qubit* [15]. Similar to the classical bit, a qubit also has a state. While a bit has either a *0* or *1* state, the qubit can have many more states. The quantum equivalent to the classical bit states would be $|0\rangle$ and $|1\rangle$ (spoken as *ket*) in Dirac notation [16] and they represent the orthonormal computational basis states in Equation 2.1.

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.1}$$

The complementing element to Dirac's notation is the *bra*: $\langle 0|$ and $\langle 1|$. The *bra* is the transposed vector from a *ket*. The outer product of two vectors u and v in dirac notation is written as $|u\rangle\langle v|$. While the inner product of the same vectors is expressed as $\langle u|v\rangle$.

What makes the qubit different and more capable than the classical bit is that it can also have different states created by a linear combination or *superposition* from its basis states. The linear combination in Equation 2.2 is the complete representation of a qubit, where α and β are two complex numbers that are denominated *probability*

amplitudes. The values α and β represent a distribution, in which with probability $|\alpha|^2$ we will observe a *0* value and with probability $|\beta|^2$ we will observe a *1* value. This distribution is determined by the Born rule [17] and states that $|\alpha|^2 + |\beta|^2 \stackrel{!}{=} 1$. The Born rule thus implies that a qubit state is a unitary vector in a two-dimensional complex vector space. Although the probability amplitudes can take on any complex value as long as they fulfill the Born rule, when we perform a measurement on the qubit it *collapses* to one of the two basis states.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.2}$$

In the real physical world qubits can be implemented by several different small particles. However, the mathematical qubit abstraction helps establish a baseline computing unit for quantum computing independent of which particle it is being represented by [18]. While in this perfect mathematical description noise does not occur, there are different mechanisms to represent the noise that quantum computers suffer from, namely the density operator that will be introduced in Subsection 2.1.8.

2.1.2 Bloch Sphere

The qubit state from Equation 2.2 can be rewritten into Equation 2.3, where e is the Euler number, i is the imaginary number, and γ, φ, and θ are real numbers.

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\beta\,|1\rangle\right) \tag{2.3}$$

Because for a single qubit the global phase $e^{i\gamma}$ has no observable effects, we can omit it and write the state of a qubit as:

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\beta\,|1\rangle \tag{2.4}$$

where θ and φ determine a point in the Bloch sphere [19]. The Bloch sphere (Fig. 2.1) is a helpful visual representation for understanding the state of a qubit. In Section 2.2 this representation will be utilized to show the effects of quantum noise on a quantum state. It can also be used to visualize the effect of the operations performed on quantum states, these operations are called *gates* and they will be introduced in Subsection 2.1.4.

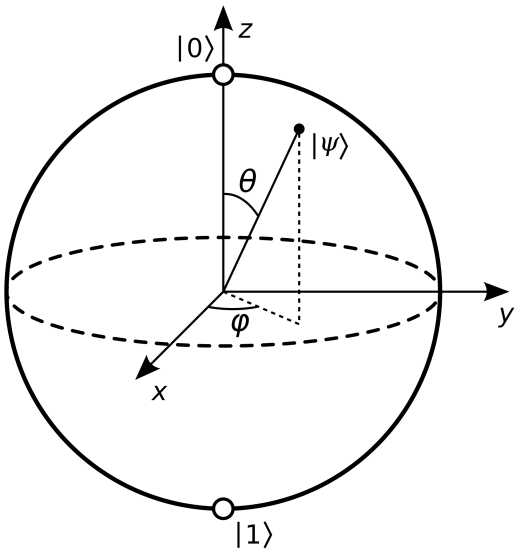

Figure 2.1: Bloch sphere representation of a qubit. Image taken from Wikimedia Commons under the Creative Commons Attribution-Share Alike 3.0 Unported license.

2.1.3 Multiple qubits

To describe multiple qubits we utilize the fundamentals presented in Subsection 2.1.1 and expand them. For two qubits $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$ are the computational basis. A general representation for a two qubit system can be found in Equation 2.5, where all the probability amplitudes must follow the Born rule.

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{2.5}$$

Due to the Born rule the measurement results for $x = \{00, 01, 10, 11\}$ follow the probability distribution determined by $|\alpha_x|^2$. Similar to a single qubit, once a measurement on both qubits is performed, the state of the qubits will collapse to the measured computational basis. Nevertheless, with multiple qubits we are able to perform measurements on a subset of qubits. In the case of a two-qubit system, measuring the first qubit will collapse its value. However, the second's qubit state will remain. In the case of Eq. 2.5, if *0* was measured in the first qubit, the amplitudes α_{10} and α_{11} would disappear from the state as they are no longer possible. Furthermore, the

remaining amplitudes must be normalized, such as in Eq. 2.6, to fulfill the normalization restriction.

$$\left|\psi'\right\rangle = \frac{\alpha_{00}\left|00\right\rangle + \alpha_{01}\left|01\right\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \tag{2.6}$$

In Equation 2.7 we can find a general representation of a set of n qubits. For a system composed by n qubits there are 2^n amplitudes. If we tried to simulate a quantum system with $n = 50$, assuming that complex numbers require 8 bytes to be stored [20], a classical computer would need approximately 9000 terabytes to store the generated quantum state. This simple calculation shows the reason why quantum computers are so promising and also why classical computers are not able to process quantum information efficiently.

$$\left|\psi\right\rangle = \alpha_{00}\left|0\cdots0\right\rangle + \cdots + \alpha_{2^{n-1}}\left|1\cdots1\right\rangle \tag{2.7}$$

2.1.4 Quantum Gates

Once that quantum states have been defined, performing operations on them is the next step to understand how quantum computing works. These operations are denominated *quantum gates* and they modify the quantum state according to its properties. A quantum gate can be represented as a matrix that fulfills one single property, namely that it is a unitary matrix. In Equation 2.8 we can observe that a unitary matrix is one which when multiplied by its own transpose conjugate is equal to the identity matrix.

$$UU^{\dagger} = I \tag{2.8}$$

This property is required because when a quantum gate is used on a quantum state, the resulting quantum state has to be a valid normalized quantum state. By being a unitary matrix, this effect is achieved. There are infinitely many unitary matrices, however, there are some of specific importance. The most important single-qubit quantum gates will be introduced in Subsection 2.1.4.1, while the most significant multiple-qubits gates will be presented in Subsection 2.1.4.2

2.1.4.1 Single-Qubit Gates

Single-Qubit gates can be described by a two by two unitary matrix. The first three quantum gates introduced are described by the Pauli matrices. They are defined as the X, Y and Z gates because each matrix represents a π rotation around the Bloch sphere in their respective axis.

X Gate

The X gate's matrix and its gate representations can be found in Equation 2.9. In classical computing, the X gate is conceptually equivalent to the NOT gate, thus it is also known as a quantum NOT gate.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad -\boxed{X}- \qquad -\oplus- \tag{2.9}$$

In Equation 2.10 we can see the effect of the X gate, where the computational basis states are flipped. This phenomena is known as a *bit flip*.

$$X\left|0\right\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \left|1\right\rangle, \qquad X\left|1\right\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left|0\right\rangle \tag{2.10}$$

Z Gate

Unlike the X gate, there is no conceptually equivalent gate for the Z gate in classical computing. In Equation 2.11 we can observe the matrix and symbol representation of the Z gate.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad -\boxed{Z}- \tag{2.11}$$

The effects of the Z gate can be found in Equation 2.12. We can observe that the $\left|0\right\rangle$ state remains unmodified while the $\left|1\right\rangle$ state is negative after the operation. This phenomena is known as a *phase flip*.

$$Z\left|0\right\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \left|0\right\rangle, \qquad Z\left|1\right\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -\left|1\right\rangle \tag{2.12}$$

Y Gate

The Y gate also doesn't have a conceptually equivalent gate in classical computing. More interestingly the Y gate can be described by the product of the X and Z matrix up to a global phase. The matrix and symbol representation of the Y gate can be found in Equation 2.13.

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad -\boxed{Y}- \tag{2.13}$$

In Equation 2.14 we can see how the Y gate modifies the computational basis states. In it we can observe that a bit flip and a phase flip have been executed, while i has been added as a global phase.

$$Y\left|0\right\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = i\left|1\right\rangle, \qquad Y\left|1\right\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -i\left|0\right\rangle \tag{2.14}$$

As a complement to the Pauli gates, there are gates that perform a specific θ rotation around each axis of the Bloch sphere. They are known as $RX(\theta)$, $RY(\theta)$ and $RZ(\theta)$.

Hadamard Gate

The Hadamard gate is one of the most useful single-qubit gates. It will be later used in Subsection 2.1.5 to create *entanglement* between two different qubits.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad -\boxed{H}- \tag{2.15}$$

The effects of the Hadamard gate can be observed in Equation 2.16. The resulting states from applying the Hadamard gate to the computational basis are denominated $\left|+\right\rangle$ and $\left|-\right\rangle$ respectively and they represent a superposition with equal probability for both computational basis to be measured.

$$H\left|0\right\rangle = \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \left|1\right\rangle\right) = \left|+\right\rangle \qquad H\left|1\right\rangle = \frac{1}{\sqrt{2}}\left(\left|0\right\rangle - \left|1\right\rangle\right) = \left|-\right\rangle \tag{2.16}$$

Another interesting property of the previously mentioned single-qubit gates is that all of them are involutory. An involutory matrix is a square matrix that is its own inverse. This implies that performing twice the same gate will not modify the quantum state as the identity transformation would have been applied.

2.1.4.2 Multiple-Qubits Gates

Multiple-Qubits gates are not that different from single-qubit gates, they still have to be unitary. Nevertheless, depending on the qubits that the gate is trying to modify, the size of the matrix will change. Formally, if a gate is applying a transformation to n qubits, then the dimensions of the gate's matrix $M \in \mathbb{C}^{2^n \times 2^n}$. Therefore, if $n = 2$ then M would have the size $\mathbb{C}^{4\times 4}$.

Controlled NOT Gate

The Controlled NOT (CNOT) gate is a two-qubit quantum gate. It has a *control* qubit and a *target* qubit. In Equation 2.17 we can observe the matrix and symbol representation where the first qubit is the control qubit and the second is the target qubit.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{aligned} CNOT\,|00\rangle &= |00\rangle \\ CNOT\,|01\rangle &= |01\rangle \\ CNOT\,|10\rangle &= |11\rangle \\ CNOT\,|11\rangle &= |10\rangle \end{aligned} \tag{2.17}$$

The effects of the CNOT gate can be seen in Equation 2.17, where depending on the value of the control qubit, a NOT gate will be applied on the target wire. This can also be represented as $|A, B\rangle \rightarrow |A, B \oplus A\rangle$ performing an XOR between the control and target qubits and storing the value in the second qubit.

Regarding the CNOT matrix in Equation 2.17, an intuiton can be gained regarding the columns of the matrix, the computational basis, and the effects of the gate. The ordering of the columns each represent the input state with regards to the computational basis. The first column represents $|00\rangle$, the second column $|01\rangle$, the third column $|10\rangle$ and so on. Moreover, the values that the vectors represent in each column are associated with the output values after the gate has been applied, e.g. $\begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^\top = |00\rangle$, $\begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}^\top = |01\rangle$, $\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}^\top = |10\rangle$, etc.

We can observe this phenomena between the third and fourth column in Eq. 2.17, where the columns and values swap. Also this intuiton helps us construct the matrix from the CNOT gate when the control and the target qubits have been inverted. In Equation 2.18 we construct the matrix from the effects the CNOT gate would have in

the different computation basis states.

$$\begin{aligned} CNOT\,|00\rangle &= |00\rangle \\ CNOT\,|01\rangle &= |11\rangle \\ CNOT\,|10\rangle &= |10\rangle \\ CNOT\,|11\rangle &= |01\rangle \end{aligned} \qquad \qquad CNOT_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{2.18}$$

SWAP gate

The SWAP gate's effect are introduced in Equation 2.19. It shows that the value of the first qubit is swapped with the value of the second value. Therefore, the gate will only have an effect if the qubits' values are different.

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \qquad \begin{aligned} SWAP\,|00\rangle &= |00\rangle \\ SWAP\,|01\rangle &= |10\rangle \\ SWAP\,|10\rangle &= |01\rangle \\ SWAP\,|11\rangle &= |11\rangle \end{aligned} \tag{2.19}$$

The introduced two-qubit gates are also involutory. Therefore, applying them twice would not modify the original quantum state.

2.1.5 Entanglement

One of quantum mechanics most important concepts is entanglement, it is also what fundamentally enables quantum computing to perform better than classical computing. Entanglement occurs when two or more particles' quantum states are correlated and cannot be described or measured independently of each other. In quantum computing the simplest and most common sequence of gates to entangle two qubits can be found in Equation 2.20, where the Hadamard gate puts the first qubit in a superposition and the CNOT gate creates a dependent relationship between both qubits.

$$|0\rangle \quad |0\rangle \qquad \qquad |\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \tag{2.20}$$

Reminiscing the fact that measuring a qubit will collapse its quantum state to one of the computational basis. In the case of $|\psi\rangle$, according to the effects of Equation 2.6, measuring the first qubit will eliminate one of the two possible states. If the measured value of the first qubit is *0*, then $|\psi'\rangle = |00\rangle$. If *1* is measured, then $|\psi'\rangle = |11\rangle$. In the

case where we measured *0*, the second qubit will be *0* when it is measured, else the second qubit would be *1*. Therefore, without measuring the second qubit we can know the value it will take if it was measured.

$$\begin{aligned} |00\rangle &\implies |\Phi^+\rangle = \tfrac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) & |01\rangle &\implies |\Psi^+\rangle = \tfrac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \\ |10\rangle &\implies |\Phi^-\rangle = \tfrac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right) & |11\rangle &\implies |\Psi^-\rangle = \tfrac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right) \end{aligned} \tag{2.21}$$

The quantum states created by the circuit in Equation 2.20 with differring basis inputs are called Bell's states or EPR states [21]. The Bell's states represent the maximally entangled state between two qubits. In Equation 2.21 we can observe the four resulting states, where in the Φ states the value of the measured qubit will be the same as the non-measured qubit. For the Ψ states, the value of the measured qubit will be the opposite of the non-measured qubit. This relationship holds independent on whether the first or the second qubit in the system is measured.

Entanglement is one of the most powerful tools in quantum mechanics. Some of the documented uses of entanglement (mainly using EPR states) are superdense coding [22] and quantum teleportation [23]. Quantum teleportation describes transmitting quantum information from a sender to a receiver in a different location. Superdense coding refers to transferring a certain amount of classical bits of information using a lesser number of qubits, therefore, sending more information with less resources.

2.1.6 Quantum Measurement

So far performing a measurement will collapse the quantum state to one of the two computational basis. Nonetheless, a quantum state can be measured in several different basis depending on the measurement operator that is being used. Quantum measurements are defined by a set M_m of measurement operators. This set must fulfill the *completeness equation* property (Eq. 2.22) where all the operators must add to the identity matrix. This property can be interpreted as all the probability amplitudes from the quantum state must add up to one.

$$\sum_m M_m^\dagger M_m = I \implies \sum_m p\left(m\right) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle = 1 \tag{2.22}$$

Given a quantum state $|\psi\rangle$ right before a measurement, the probability of measuring m is defined as $p\left(m\right) = \langle\psi| M_m^\dagger M_m |\psi\rangle$. The quantum state after the measurement

$|\psi'\rangle$ is described in Equation 2.23.

$$|\psi'\rangle = \frac{M_m |\psi\rangle}{\sqrt{\langle\psi| |M_m^\dagger M_m| |\psi\rangle}} \tag{2.23}$$

We can apply this framework to the computational basis. The measurement operators of the computational basis are $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. Both of the operators are Hermitian, thus they are equal to their conjugate transpose and $M_m^\dagger M_m = M_m^2$. Solving the square matrix of both measurement operators yields the original matrix $M_m^2 = M_m$. We can observe now that the completeness property is fulfilled in Equation 2.24.

$$M_0^\dagger M_0 + M_1^\dagger M_1 = M_0 + M_1 = I \tag{2.24}$$

For a state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ we can utilize this new framework to explain what the probability of measuring *0* or *1* is. In Equation 2.25 we can observe that the probability of measuring *0* is $|\alpha|^2$, equal to the magnitude mentioned in Subsection 2.1.1.

$$p(0) = \langle\psi| M_0^\dagger M_0 |\psi\rangle = \langle\psi| M_0 |\psi\rangle = |\alpha|^2 \tag{2.25}$$

With Equation 2.25 we can also easily derive that the probability for measuring the value *1* is $p(1) = |\beta|^2$. In Equation 2.26 we calculate how measuring affects the quantum state, we use Equation 2.23 with the derived probability values. Recalling that a global phase has no effect in a qubit (Subsection 2.1.2), we can assume that $\frac{\alpha}{|\alpha|} |0\rangle = |0\rangle$ and $\frac{\beta}{|\beta|} |1\rangle = |1\rangle$.

$$|\psi_0'\rangle = \frac{M_0 |\psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|} |0\rangle \qquad |\psi_1'\rangle = \frac{M_1 |\psi\rangle}{|\beta|} = \frac{\beta}{|\beta|} |1\rangle \tag{2.26}$$

One would assume that knowing the magnitude of the probability amplitudes is enough to reconstruct a quantum state, however, this is not the case. For the states $|+\rangle$ and $|-\rangle$ the probability of measuring the value *0* is $p(0) = \frac{1}{2}$ and the probability of measuring the value *1* is $p(1) = \frac{1}{2}$. Nevertheless, the quantum states $|+\rangle$ and $|-\rangle$ are not equivalent. This occurs because there isn't any quantum measurement capable of discerning between quantum states unless they are orthonormal.

To be able to differentiate $|+\rangle$ and $|-\rangle$ we need to measure in a different basis composed by orthonormal vectors. The normalization property assures that they will be valid quantum states. The orthogonality property guarantees linear independence between vectors, thus, they can describe any quantum state in terms of the basis. In

this case we can use $|+\rangle$ and $|-\rangle$ as the new vectors for the basis. We ensure that the vectors are normalized by calculating the inner product of the vectors with themselves $\langle +|+\rangle = \langle -|-\rangle = 1$. We guarantee that the vectors are orthogonal by calculating the inner product of the vectors with each other $\langle +|-\rangle = \langle -|+\rangle = 0$.

$$M_+ = |+\rangle\langle +| = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \qquad M_- = |-\rangle\langle -| = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \tag{2.27}$$

The measurement operators derived from the new basis can be found in Equation 2.27. These operators can now be used to calculate the probability to measure either $|+\rangle$ or $|-\rangle$. Similar to the computational basis case, both measurement projectors are Hermitian. Thus, we can derive the expression for the probability to measure $x \in \{+, -\}$ where $p(x) = \langle \psi | M_x | \psi \rangle$. With this new expression we can clearly differentiate in Equation 2.28 the $|+\rangle$ and $|-\rangle$ quantum states.

$$\begin{aligned} p(+) &= \langle +| M_+ |+\rangle = 1 \qquad & p(-) &= \langle +| M_- |+\rangle = 0 \\ p(+) &= \langle -| M_+ |-\rangle = 0 \qquad & p(-) &= \langle -| M_- |-\rangle = 1 \end{aligned} \tag{2.28}$$

2.1.6.1 Projective Measurement

Projective measurements are a special case of quantum measurements. They are characterized by an *observable*, denoted as M, which is a Hermitian operator acting on the state space of the observed system. The observable's spectral decomposition is in Equation 2.29, where we can see that M is composed by the sum of the projector P_m multiplied by the corresponding eigenvalue m. The observable's eigenvalues are the possible values for the measurement.

$$M = \sum_m m P_m \tag{2.29}$$

From the observable we can derive an expression in Equation 2.30 for the expectation value. This expression is often denoted as $\langle M \rangle$. The expectation value denotes the value of the average measurement with respect to the observable. More importantly, the expectation value can be interpreted as the weighted average of what would be measured over many experiments, thus, analytically calculating the result of several

trials.

$$\mathbb{E}(M) = \sum_m m * p(m) = \sum_m m \langle\psi| P_m |\psi\rangle = \langle\psi| \left(\sum_m m P_m\right) |\psi\rangle = \langle\psi| M |\psi\rangle \tag{2.30}$$

The Pauli Z matrix is of special importance as an observable. In Equation 2.31 we can find the spectral decomposition of the Pauli Z matrix. It has the eigenvalues *1* and *-1*, that correspond to its eigenstates $|0\rangle$ and $|1\rangle$. The eigenstates match the computational basis and form its measurement operators. Hence, the Pauli Z matrix is widely used as an observable for the computational basis.

$$Z = 1 * |0\rangle\langle 0| + (-1) * |1\rangle\langle 1| \tag{2.31}$$

2.1.7 Quantum Circuit

2.1.8 Density Operator

- Introduce the density operator / schuld 87, nielsen 98
- Introduce quantum algorithms*
Todo: - Citation for hadamard gate, pauli gate.

2.2 Quantum Noise

i. Describe the types of noise that can occur. ii. Explain where can noise occur. iii. State how noise can be simulated.

2.3 Quantum Machine Learning

i. Present the difference between QML and classical ML. ii. Introduce variational quantum circuits. iii. Explain quantum kernel methods.

2.4 Adversarial Machine Learning

i. State generalization problems. ii. Present different attacks such as FGSM, C&W, and PGD. iii. Introduce adversarial training as defence mechanism against adversarial attacks. iv. Explain the relationship between general accuracy and adversarial resilience.

3 Implementation

3.1 Section

a. Introduce the used datasets and why they were chosen. b. Describe how the experiments have been chosen and created: i. Possible mixes with different types of noise, in different places, with different datasets and QML mechanisms.

Possible experiments:

a. Change the type of encoding

b. Variational quantum circuits and kernels

1. Amplitude damping (open quantum systems / how the quantum system is affected by its environment)

2. Phase damping (loss of information but not energy)

3. Simpler phase and bit flips

4 Style

4.1 Section

Citation test [1]. Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ ac{tum}, \ ac{tum}` ⇒ Technical University of Munich (TUM), TUM

For more details, see the documentation of the `acronym` package[1].

4.1.1 Subsection

See Table 4.1, Figure 4.1, Figure 4.2, Figure 4.3.

Table 4.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

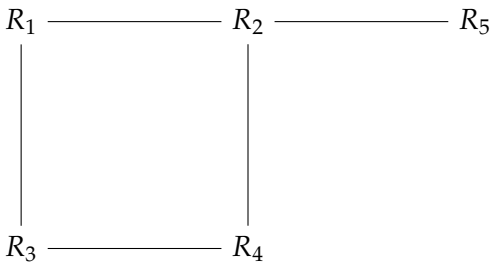

Figure 4.1: An example for a simple drawing.

[1] `https://ctan.org/pkg/acronym`

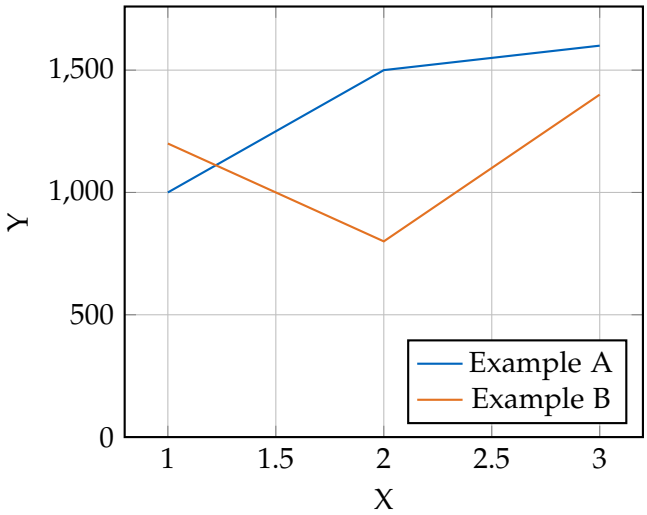

Figure 4.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 4.3: An example for a source code listing.

Abbreviations

TUM Technical University of Munich

NISQ Noisy Intermediate-Scale Quantum

ML Machine Learning

QML Quantum Machine Learning

CNOT Controlled NOT

List of Figures

2.1 Bloch sphere representation of a qubit. Image taken from Wikimedia Commons under the Creative Commons Attribution-Share Alike 3.0 Unported license. 6

4.1 Example drawing 18
4.2 Example plot 19
4.3 Example listing 19

List of Tables

4.1 Example table 18

Bibliography

[1] bibinitperiod M. National Academies of Sciences Engineering. *Quantum Computing: Progress and Prospects*. Google-Books-ID: ATH3DwAAQBAJ. National Academies Press, Mar. 27, 2019. 273 pp. ISBN: 978-0-309-47972-1.

[2] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." In: *SIAM Journal on Computing* 26.5 (Oct. 1997). Publisher: Society for Industrial and Applied Mathematics, pp. 1484–1509. ISSN: 0097-5397. DOI: 10.1137/S0097539795293172.

[3] W. Van Dam, S. Hallgren, and L. Ip. "Quantum Algorithms for Some Hidden Shift Problems." In: *SIAM Journal on Computing* 36.3 (Jan. 2006), pp. 763–778. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/S009753970343141X.

[4] S. Hallgren. "Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem." In: *Journal of the ACM* 54.1 (Mar. 2007), pp. 1–19. ISSN: 0004-5411, 1557-735X. DOI: 10.1145/1206035.1206039.

[5] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." In: *Communications of the ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/359340.359342.

[6] J. Preskill. "Quantum Computing in the NISQ era and beyond." In: *Quantum* 2 (Aug. 6, 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79. arXiv: 1801.00862[cond-mat,physics:quant-ph].

[7] P. W. Shor. "Quantum Computing." In: *Documenta Mathematica - Extra Volume ICM 1998*, pp. 467–486.

[8] R. Bommasani, D. A. Hudson, E. Adeli, et al. *On the Opportunities and Risks of Foundation Models*. July 12, 2022. arXiv: 2108.07258[cs].

[9] M. Schuld and F. Petruccione. *Machine Learning with Quantum Computers*. Quantum Science and Technology. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-83097-7 978-3-030-83098-4. DOI: 10.1007/978-3-030-83098-4.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. *Intriguing properties of neural networks*. Feb. 19, 2014. arXiv: 1312.6199[cs].

[11] N. Papernot, P. McDaniel, and I. Goodfellow. *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples*. May 23, 2016. arXiv: 1605.07277[cs].

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. *Explaining and Harnessing Adversarial Examples*. Mar. 20, 2015. arXiv: 1412.6572[cs,stat].

[13] A. Kurakin, I. Goodfellow, and S. Bengio. *Adversarial Machine Learning at Scale*. Feb. 10, 2017. arXiv: 1611.01236[cs,stat].

[14] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig. "Quantum machine learning: a classical perspective." In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2209 (Jan. 2018), p. 20170551. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2017.0551.

[15] B. Schumacher. "Quantum Coding." In: *Physical Review A* 51.4 (Apr. 1, 1995), pp. 2738–2747. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.51.2738.

[16] P. a. M. Dirac. "A New Notation for Quantum Mechanics." In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (July 1939), pp. 416–418. ISSN: 1469-8064, 0305-0041. DOI: 10.1017/S0305004100021162.

[17] M. Born. "Quantenmechanik der Stoßvorgänge." In: *Zeitschrift für Physik* 38.11 (Nov. 1, 1926), pp. 803–827. ISSN: 0044-3328. DOI: 10.1007/BF01397184.

[18] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Higher Education from Cambridge University Press. ISBN: 9780511976667 Publisher: Cambridge University Press. Dec. 9, 2010. DOI: 10.1017/CBO9780511976667. URL: https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE (visited on 03/28/2024).

[19] F. Bloch. "Nuclear Induction." In: *Physical Review* 70.7 (Oct. 1, 1946), pp. 460–474. ISSN: 0031-899X. DOI: 10.1103/PhysRev.70.460.

[20] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. "Array programming with NumPy." In: *Nature* 585.7825 (Sept. 2020). Publisher: Nature Publishing Group, pp. 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2.

[21] A. Einstein, B. Podolsky, and N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Physical Review* 47.10 (May 15, 1935), pp. 777–780. ISSN: 0031-899X. DOI: 10.1103/PhysRev.47.777.

[22] C. H. Bennett and S. J. Wiesner. "Communication via One- and Two-Particle Operators on Einstein-Podolsky-Rosen States." In: *Physical Review Letters* 69.20 (Nov. 16, 1992), pp. 2881–2884. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.69.2881.

[23] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. "Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels." In: *Physical Review Letters* 70.13 (Mar. 29, 1993), pp. 1895–1899. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.70.1895.