

## Master thesis

Master's Degree in Computational Engineering and  
Intelligent Systems

---

# **Building hybrid classical-quantum classifiers to deal with unbalanced datasets**

---

*Unai Lizarralde Imaz*

### **Advisors**

Jose A. Pascual Saiz

September 2022



# Acknowledgements

First and foremost, I want to show my gratitude towards my master thesis advisor Jose A. Pascual Saiz of the Faculty of Informatics in the Basque Country's University (UPV/EHU). Through the many hardships I have encountered, he offered me the best guidance possible towards how to pursue the objectives of the master thesis, while providing an insightful judgement of the decisions and course of actions taken to improve the content beyond the initial premise and immeasurable help on how to deal with more sophisticated approaches.

I would like to express my sincerest gratitude to my closest friends who have been essential in trudging through the more personal struggle and hesitation. They offered me the most crucial pieces of advice when I needed them the most, even when I didn't know how much I needed them. Their words of companionship and encouragement gave me the proper attitude and mentality to tackle with the glaring issues and difficulties I have faced in the completion of this master thesis.

My final and most heartfelt gratitude is directed towards my family for giving me undivided support despite the longevity of the master thesis and always trusting me across my many years of studies and researches conducted. The final offspring of the entire work and the invested countless hours in would have been impossible to achieve without their consistent involvement, absolute faith and untiring support.

May this accomplishment serve as testament of their invaluable effort provided to back me up. Thank you kindly.

- Unai Lizarralde



# Abstract

Quantum circuits offer a different approach to process data through quantum operations and measurements of quantum states. At the same time, the increasing number of advances in technology has opened a path to turning these quantum circuits into building blocks of a machine learning model. The transition of data from classical to quantum and reverting it from quantum to classical offers a presumably much more nuanced and diverse form of learning the data instead of using two different scenarios separately.

The idea consists on combining these two scenarios meaningfully into a single hybrid classical quantum model and observe how these two settings may offer value instead of only focusing in one or another. Therefore, the challenge consists on dealing with a machine learning problem using three different means; namely, quantum models, classical models and hybrid classical quantum models and make an assessment of the procedure's design, techniques employed and infrastructures built.

In particular, the machine learning task belongs to a binary classification problem of an unbalanced dataset. Given that the drive of the comparison between classical and quantum means stems from evaluating and comparing their performance, the already hard and complex underlying pattern representation from the features, owing to the presence of unbalanced class distribution, is an adequate choice. The unbalanced dataset contains a binary classification problem of transactions being either classified as fraud or as valid, the former class being considerably lower in number.

Apart from the machine learning approach and the difficulty of the task, the quantum model addition is developed by picturing how the quantum tools brought to the table can be used to achieve a possibly better representation and pattern description of such complicate unbalanced data sets. The main focus is exploring data representation in quantum circuits using different types of embeddings, the variational quantum classifier model for classification, the interpretation of an observable as a Hermitian operator in quantum mechanics and the come and forth between classical and quantum communication.



# Contents

<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 The aims of the project</b>	<b>3</b>
<b>3 Background</b>	<b>5</b>
3.1 Machine learning scaling from classical to quantum systems . . .	5
3.1.1 Quantum speedup . . . . .	7
3.2 Machine learning in classical systems . . . . .	7
3.3 The importance of linear algebra . . . . .	8
3.3.1 Fundamental concepts . . . . .	9
3.3.2 Quantum principal component analysis . . . . .	10
3.3.3 Quantum support vector machines . . . . .	13
3.4 The problem of loading classical data into quantum computers . .	17
3.5 Deep quantum learning . . . . .	18
3.5.1 Quantum feedforward neural networks . . . . .	19
3.5.2 Quantum convolutional neural networks . . . . .	20
3.5.3 Quantum Boltzmann machines . . . . .	20
<b>4 Related work</b>	<b>25</b>
4.1 Associated topics . . . . .	25
4.2 Contributions . . . . .	26
<b>5 Variational quantum circuits</b>	<b>27</b>
5.1 Quantum circuit configuration . . . . .	28
5.2 Quantum embedding . . . . .	28
5.2.1 Basis embedding . . . . .	29
5.2.2 Amplitude embedding . . . . .	30
5.2.3 Angle embedding . . . . .	31
5.3 Processing data . . . . .	32

5.4	Measurement . . . . .	32
5.4.1	Measuring an observable in a quantum state . . . . .	33
5.4.2	Probability vector . . . . .	33
<b>6</b>	<b>Quantum operations</b>	<b>35</b>
6.1	Superposition and entanglement . . . . .	35
6.2	Pure states . . . . .	38
6.3	Mixed states . . . . .	40
6.4	Reduced density matrix . . . . .	42
6.5	Expectation value of an observable . . . . .	44
6.5.1	Pauli matrices . . . . .	45
<b>7</b>	<b>Experiment and design</b>	<b>49</b>
7.1	Environment and packages . . . . .	49
7.2	Data management . . . . .	51
7.3	Machine learning models . . . . .	52
7.3.1	Quantum models . . . . .	52
7.3.2	Classical model . . . . .	55
7.3.3	Hybrid models . . . . .	56
7.4	Results . . . . .	60
7.4.1	Metrics' results . . . . .	60
7.4.2	Balanced test subset results . . . . .	60
7.4.3	Interpretation and reasoning of the results . . . . .	62
<b>8</b>	<b>Conclusions</b>	<b>65</b>
	<b>Bibliography</b>	<b>67</b>



# List of Figures

3.1	A Bloch sphere representation of a quantum state from Wikipedia. The Bloch sphere surface corresponds to quantum states, such as the quantum state $ \psi\rangle$ . . . . .	10
3.2	Standard quantum gates transformations with their corresponding quantum circuit representation and matrix representation from Wikipedia. . . . .	11
3.3	SVM illustration in a two dimensional space. Source: Wikipedia. . . . .	14
3.4	Kernel function on a data set for a support vector machine. Source: Wikipedia. . . . .	15
3.5	Quantum swap test. Source: Wikipedia. . . . .	16
3.6	Quantum feedforward neural network from [1]. . . . .	19
3.7	Different approaches to build CNN from [2]. A classical CNN in <b>a</b> , a quantum CNN in <b>b</b> and QCNN comparison in <b>c</b> . . . . .	21
6.1	A quantum circuit with Hadamard gate and CNOT gate that produces the Bell state $ \psi_{q_0q_1}\rangle$ . . . . .	40
7.1	General workflow for different models and specifications . . . . .	50
7.2	The quantum model diagram. . . . .	54
7.3	The quantum circuit diagram using amplitude embedding. . . . .	55
7.4	The quantum circuit diagram using data reuploading. . . . .	55
7.5	Classical model with two dense layers and <i>sigmoid</i> activation function in the output layer. . . . .	56
7.6	Sequential hybrid model using amplitude embedding. . . . .	58
7.7	Functional hybrid model with sequential layer collections and angle embedding. . . . .	59
7.8	Quantum strongly entangling layers using angle embedding. . . . .	59
7.9	Quantum model with amplitude embedding distance plot towards correct class and confusion matrix. . . . .	61
7.10	Quantum model with data reuploading distance plot towards correct class and confusion matrix. . . . .	61
7.11	Hybrid sequential model classification plot and confusion matrix. . . . .	62
7.12	Hybrid functional model classification plot and confusion matrix. . . . .	62
7.13	Classical model classification plot and confusion matrix. . . . .	62

# List of Tables

7.1	Results of each and every model presented for the test set. . . . .	60
-----	---	----

# Introduction

In a more than ever connected world where information of every and any kind is available and accessible, technologies, methodologies and distribution of that knowledge are key parts to build the current information society. Information of all kinds is distributed under demand and communication technologies are the norm causing ripples in education, economy, health, government and, at its very core, social interactions.

However, analyzing raw data in most cases leads to a difficult stage of recognizing the nature of the data and the hidden patterns within itself. Understanding and learning the patterns found in data is the purpose of many machine learning approaches. However, the complexity of the data, its incomplete or unordered presentation and in general the lack of any type of convention is a huge impediment. More often than not the data is presented in unconventional forms that require some processing before even considering learning from it, since the data is not necessarily collected thinking about how different machine learning approaches might handle it or how to benefit the specific architectures using data types that enhance learning processes. This is generally an afterthought unfortunately and, as a result, it is often tied to using preprocessing steps such as sampling of cases or handling noise present in the data.

Amidst the sea of data collections and data sets, some of the hardest types of these data collections include unbalanced distribution of classes, which means that there not the same amount of labeled cases of each type and, therefore, the training process will be fed with more cases of certain classes and few of others unless some prior steps are taken to handle the situation. These kinds of datasets, specially when the class distribution is completely unbalanced, can pose to be a challenging scenario to learn the underlying pattern correctly, often leading to overfitting a class. Moreover, in unbalanced datasets, if it was not enough with the terrible class distribution already, the nuance required to separate cases with the given features does not have enough time to be polished.

## 1. INTRODUCTION

---

Pattern recognition of data to later fulfill a specific task is the very soul of machine learning. Classical computers have achieved excellent performance in learning from data and choosing wisely a machine learning model or a suitable algorithm that performs outstandingly or, at least, can provide meaningful results. Machine learning approaches, specially based on neural networks, are in the eye of the storm right now, thank to their capability to learn nonlinear and complex functions from data vectors to improve how the data is processed with optimization methods (gradient descent methods). In comparison linear or logistic regression methods fall short for complex data sets with nonlinear underlying functions which neural networks can tackle with the nonlinearity present in their activation function for instance.

In this context, the design of a novel approach to the classification task performed on these types of unbalanced datasets has been envisioned. Quantum computing has opened a path to learning patterns and grasping the logic and behaviour of the features through more distinctive methodologies. The achievable computational performance with the more refined and subtle incorporation of quantum mechanics leads to quantum speedups. As a result, the main focus of the project is a joint effort of both classical and quantum algorithms to create hybrid classical-quantum classifiers.

While separately classical algorithms and quantum algorithms already show promise in their own departments (such as quantum algorithms in simulations), the intention of blending them into a single model that includes a classical stage and a quantum stage lies in observing how they can either support one another for a better classification performance or, on the other hand, hinder that same performance or not offer any notable improvement for the time being.

## The aims of the project

The aims of the project are geared towards learning relevant quantum machine learning methods that can, in theory, be more proficient when learning more complex patterns and when reducing the computational cost associated to classical machine learning methods. Despite the current in development and growing quantum industry, some of the already created and constructed quantum algorithms are showing promising future prospects.

Quantum mechanics allow for a different approach to machine learning. At the very core, quantum machine learning is sustained by quantum gates that process quantum data vectors and quantum parameter vectors to learn in conjunction with unitary operations using the aforementioned parameters that best characterize how data is processed in a quantum circuit and result in the measurement of the final quantum states. The study of some prime examples of relevant quantum algorithms such as Boltzmann machines or linear algebra related methods such as quantum principal component analysis is part of the process to assess the current capabilities, benefits and limitations of quantum machine learning and quantum computing in general.

The ideal scenario when processing data is to have a meaningful feature space and well distributed cases over the dataset. However, raw datasets more often than not have large feature spaces adding complexity but not necessarily having good quality representations in return. Moreover, the class distribution of the data can many times be irregular and unbalanced, which leads to difficulties in the training phase of many machine learning approaches and requires preprocessing techniques to compensate the issue (which still doesn't guarantee better performance when generalizing).

Given the properties of quantum mechanics (quantum superposition or quantum coherence, for instance) and the amount of data that can be stored and manipulated thanks to quantum states' probability amplitude representation, quantum machine learning approaches that tackle with this kind of problems are the main focus. The

## 2. THE AIMS OF THE PROJECT

---

particular machine learning task to solve is the classification problem of unbalanced datasets using quantum machine learning algorithms. Among quantum machine learning algorithms that serve this purpose, the quantum variational algorithm is the quantum algorithm to showcase a study on and evaluate the performance of.

The final step involves going a step further and observing both classical machine learning approaches for classification, such as neural networks, and quantum machine learning approaches for classification using the quantum variational algorithm stand against each other. And, to top it off, a hybrid classifier combining both machine learning paradigms for classification purposes is considered. Therefore, the objective is not only a direct comparison between both classical machine learning approaches and quantum machine learning approaches, but also an opportunity to combine and assess the new possibilities the combination of both can deliver.

All in all, the goals of the project can be gathered in the list below:

- The study of notable and relevant quantum algorithms for linear algebra and machine learning.
- The potential quantum machine learning methods hold, and the construction of a specific design of a quantum variational algorithm to fit the classification needs.
- The application of the quantum variational algorithm to solve classification problems as well as including classical machine learning methods.
- The creation of a hybrid classical quantum classifier that can learn patterns combining the best of both worlds.

## Background

Machine learning has produced impressive results when tackling with complex problems such as natural language processing tasks, image processing tasks, time series prediction or various AI related operations. In general, building a model that can learn from data autonomously and solve a specific task with stellar performance is associated usually with machine learning in these times.

The combination of these sophisticated machine learning approaches with the new and wide-spreading quantum algorithms is the foundation of quantum machine learning. The main concepts of quantum machine learning are explored [3] and the immediate application through tensor networks with more classical systems is presented [4] in this chapter.

### 3.1 Machine learning scaling from classical to quantum systems

Statistical patterns recognition in classical systems is the core of the machine learning process when processing data. Classical systems have proved to be incredibly capable of achieving high performance when learning from data the particular scheme behind it, and, actually, generate that same data following the statistical patterns detected.

However, quantum mechanics show a slightly different behaviour in statistical pattern recognition. That is, the statistical patterns are more often than no considered to be irregular and defy the very intuition. Such contrast, in turn, is the main appeal (or vision) quantum mechanics can provide and make the difference against commonly used methods such as deep neural networks of some specific kind depending on the nature of the data.

Therefore, the pursuit of introducing quantum mechanics in machine learning stems from the principle which stands that, if quantum processors hold the

### 3. BACKGROUND

---

possibility to learn statistical patterns labeled computationally difficult for even already well performing classical systems, then it is not far fetched to regard the same statistical patterns as difficult to recognize with a classical system as well.

Even if the statistical patterns are not that difficult to recognize to begin with, whether a classical system or a quantum system is involved, the ease with which the quantum system computational performance can surpass the classical system still remains. The interest is in both senses, one for the computational difficulty reduction, and the other for the difficult statistical pattern recognition.

Nevertheless, behind the main idea, quantum algorithms for machine learning that fit this description of efficiency and performance are required. Quantum systems that process a number of instructions with the purpose of finding a solution to a problem describe a quantum algorithm, which is a fundamental part of the quantum machine learning process.

Depending on the task at hand, there might be a quantum algorithm that can fit the data effectively and learn statistical pattern unknown, which provide never before seen insight, or on the contrary, a fitting quantum algorithm might not exist at all. The potential of quantum algorithm to surpass their classical counterparts is known as quantum speedup and can be used as a measure to know the level of improvement of a quantum system over the classical system.

The quantum speedup can be viewed as a formal theorem with mathematical proof behind to back up from scientific perspective, or as a more grounded one, where a more realistic approach with restricted features in the computer and a solid scaling advantage is introduced for a few problem sizes. Unfortunately, the quantum algorithm used in quantum machine learning does not have always the perfect performing classical algorithm to compare it with. That can be for either the magnitude of the problem or there is no guarantee that the current best algorithm can't actually be improved.

The scaling advantage from a classical system to a quantum system can be given by a benchmark. The improvements given the quantum speedup could be witnessed in the accuracy of the solver and the sampling of quantum systems, for instance. However, in the field of machine learning two scaling advantages stand out: query complexity and gate complexity.

In case of using query complexity to measure the quantum speedup, the amount of accesses to the data source is taken both for the classical version of the algorithm, as well as for the quantum version of the algorithm. If the results show a favorable minor amount of interactions with the source for the quantum algorithm than for the classical algorithm, then it is concluded that a quantum speedup occurred.

On the other hand, the gate complexity refers to the total amount of quantum elementary operation involved in the quantum algorithm and, in other words, the total amount of quantum gates that are included. Ultimately, both methods are idealized systems for the quantification of the resources needed, that is, the time required for the handling of the data and the computational cost of quantum



operations to name a few.

However, in a real world scenario, an idealization is not a viable method without finding a suitable and down to the ground representation or mapping to the particular case at hand. Without it, the amount of resources scaling is unfeasible to grasp and, as a result, they become unreliable methods that are not supported by a reasonable foundation.

### 3.1.1 Quantum speedup

In a quantum computer, quantum mechanics allow for impossible phenomena to occur while processing and manipulating information in a classical counterpart, such as quantum entanglement or quantum coherence. The constant evolution since research about how a quantum computer would operate to the actual birth of ever improving quantum computers have opened up the opportunity to develop quantum algorithms that run in these quantum machines.

In result, quantum machine learning makes use of quantum algorithms to work with data and learn patterns that classical methodologies can't recognise. This leads to obtaining a better performance from the quantum machine learning for certain scenarios and problems, hence the name quantum speedup. Naturally, the quantum algorithms rely on quantum computers and their features to approach quantum machine learning in these systems.

Some tasks that show a significant improvement over the classical approach are the following:

- *Unsorted search*: Given an unordered database with  $N$  entries a quantum computer is capable of finding the desired entry in  $\mathcal{O}(\sqrt{N})$  while a classical computer requires  $\mathcal{O}(N)$ . The resulting quantum speedup is  $\sqrt{N}$ .
- *Invert matrices and transformations*: These operations include Fourier transforms on  $N$  points, sparse  $N \times N$  matrix inversions and finding out the eigenvalues and eigenvectors of such matrices in polynomial  $\mathcal{O}(\log_2 N)$ . In comparison, the best algorithms for these tasks in classical systems are  $\mathcal{O}(N \log_2 N)$  in contrast. The resulting quantum speedup is exponential over the classical counterparts.

## 3.2 Machine learning in classical systems

Data analysis and machine learning algorithms in classical systems present different approaches depending the manner and nature of the data itself. On one hand, classical systems perform data analysis techniques to extract knowledge and make inference on the data. Some of the major techniques fall into data mining, cluster analysis or regression analysis, such as polynomial interpolation, multiple linear regression or hierarchical clustering to name a few.

On the other hand, machine learning operations depend on the type of data and learning that is desired, that is, whether the machine learning process will include a supervised learning or unsupervised learning or a hybrid of both. Apart from these, reinforcement learning methods also prevail in these systems, which are often strongly tied with artificial intelligence tasks.

Supervised learning consists of building a machine learning model where the dataset provides features with specific information and a corresponding label for each entry of the dataset. In the training process, the label is predicted from the features of the entry and compared with the true label of the entry to adjust the parameters of the model accordingly. Therefore, the machine learning makes predictions from different cases about the value of the label.

Unsupervised learning covers machine learning models that learn from datasets whose entries are not broken down in different categories. Instead, each entry includes a set of features, yet unlabeled. Learning the structure of the data and often finding the underlying information about the relationships between data points is the objective of the unsupervised scheme. In case the data is both labeled and unlabeled, the learning paradigm is called semi-supervised learning.

On a last note, reinforcement learning presents a different learning system. These variants of machine learning are based on a reward function for setting the behaviour and decisions followed in the target environment and specific activity. In consequence, how to react to different inputs to favor the reward and reduce the error produced. Generally, these methods require continuous training to hone the wit of the model. For instance, when an agent learns to escape from another agent, the reward and error function should improve whenever the agent is further in the distance from the pursuer.

### 3.3 The importance of linear algebra

Linear algebra is present in a large number of data analysis and machine learning methods, since these methods are quite often full of matrix operations with vectors of high dimensional data. For the quantum mechanics department, matrix operations on vectors from high dimensional vector spaces. Specifically, the vector space in a quantum mechanics based methodology is a complex vector space of dimension  $n$  ( $|\psi\rangle \in \mathbb{C}^n$ ).

A quantum state of a quantum computer with  $n$  qubits. Each quantum bit is a quantum mechanics system including two possible states and, therefore, represented as a two dimensional complex vector. Given a quantum system composed of  $n$  qubits, the quantum state of such quantum system when performing a measurement (collapsing through a projection an instant of the quantum system itself) has the dimension  $2^n$ .

Quantum operations performed in a quantum system rely in these complex vectors to represent the quantum state. In order to do so with many qubits and cap-

ture the quantum state with a measurement, tensor products are used to represent the joint state of many qubits, while defining the quantum state with a projective measurement.

Quantum computer are capable of achieving a better performance in many cases when undergoing basic operations in linear algebra. Evidence can be found with Fourier transforms, searching for the eigenvalues and eigenvectors and finding a solution to linear equations in a complex vector space with dimension  $2^n$ . Particularly, these examples can be calculated in polynomial time, which is exponentially quicker to do compared to classical algorithms.

### 3.3.1 Fundamental concepts

#### 3.3.1.1 Qubits

Qubits are the minimum quantum information storage, similar to bits in classical computation. A single qubit is a quantum system composed of two states, the smallest and most basic quantum system which fulfills the properties found in quantum mechanics. While a classical bit is restricted to only represent one state (0 or 1), a quantum bit can be in both states at the same time in a quantum principle known as quantum superposition. Quantum superposition is one of the key factors that allow the possibility to store many states into a single state and it's necessary to carry out quantum computing processes in many levels.

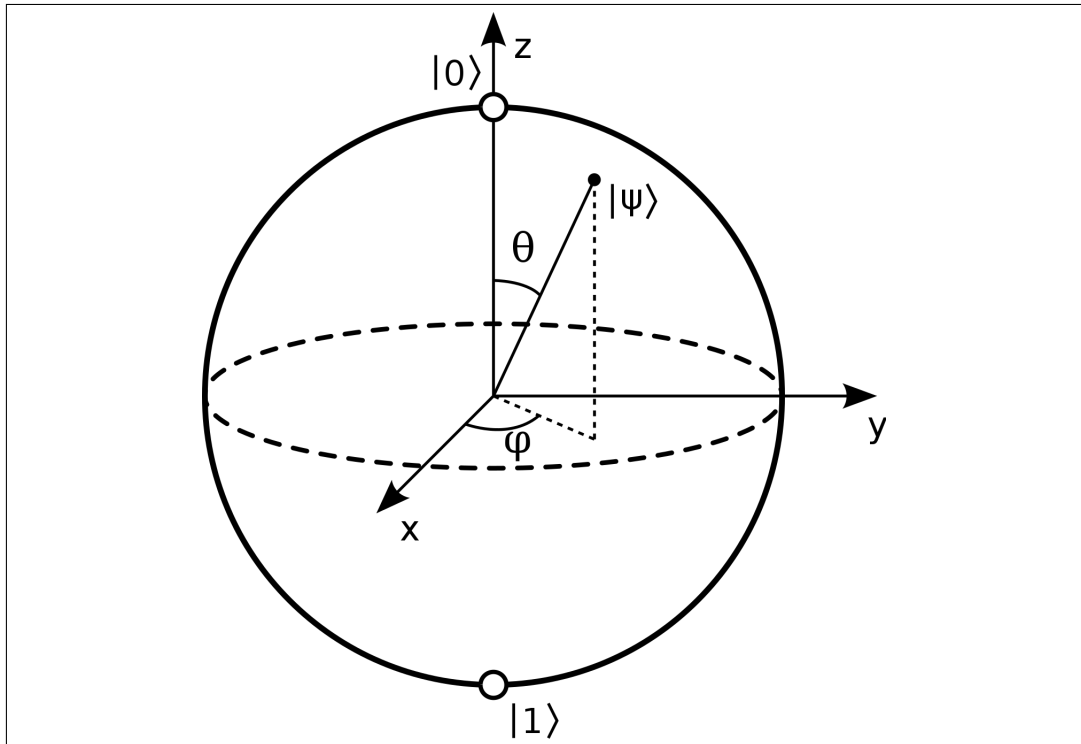
In order to determine the final state of a qubit, that is the outcome, a measurement is performed to collapse the quantum bit into one of the two states (0 or 1). However, a measurement of a qubit irrevocably changes its quantum state, and the superposition of the two states is modified in the process.

The qubit state can be represented as a superposition of the computational basis states or, in other words, a single qubit is defined by a linear combination of the  $|0\rangle$  and  $|1\rangle$  states.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.1}$$

The arbitrary parameters  $\alpha$  and  $\beta$  are complex numbers that correspond to the probability amplitudes of the qubit. In consequence, the measurement of the qubit in the standard basis using the Born rule would yield the outcome of state 0 with a probability of  $|\alpha|^2$ , while it would give the outcome of state 1 with a probability of  $|\beta|^2$ . Therefore, the absolute square of the probability amplitudes are the probabilities for each of the outcomes and satisfy  $|\alpha|^2 + |\beta|^2 = 1$ .

A suitable representation of a qubit graphically is the visualization of the qubit in the Bloch sphere 3.1. Unlike a classical bit which would only be either at one of the ends top or bottom (where the computational basis states  $|0\rangle$  and  $|1\rangle$  are specifically). The surface of the Bloch sphere represents the entire space of quantum states available for a qubit, which are obtained as linear combinations of computational basis states.



**Figure 3.1:** A Bloch sphere representation of a quantum state from Wikipedia. The Bloch sphere surface corresponds to quantum states, such as the quantum state  $|\psi\rangle$ .













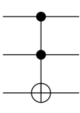
### 3.3.1.2 Quantum gates

In quantum computing, in order to apply different types of operations and transformations on a qubit a specific quantum circuit is devised. These quantum circuits include quantum operators that, similar to how logic gates work in conventional digital circuits in classical machines, manipulate and transform the initial qubit. They are called quantum gates in result.

One of the many properties of the quantum gates is their reversible nature. This occurs owing to the mathematical representation of quantum gates as unitary operators, specifically unitary matrices with a defined basis (generally the computational basis is selected). A number of quantum gates can be found in figure 3.2.

### 3.3.2 Quantum principal component analysis

PCA (Principal Component Analysis) strives to represent a dataset with new uncorrelated variables (or components in this case) that stem from the original set of variables. The objective is to extract information from the original variables, so that the components can bring out the essential information in a much more compact form. It is a common practice to reduce the number of variables or to detect correlation between variables. However, the method to achieve this differs in quantum systems to the one employed in classical systems.

Operator	Gate(s)	Matrix
Pauli-X (X)	 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

**Figure 3.2:** Standard quantum gates transformations with their corresponding quantum circuit representation and matrix representation from Wikipedia.

### 3.3.2.1 Classical methodology

In each variable vector  $\vec{v}_i$ , the value for each case of the dataset is presented. From them, the covariance matrix is obtained by  $M_c = \sum_i \vec{v}_i \vec{v}_i^T$ , which contains the correlations between variables of the dataset. For instance, consider the price and the supply of a product. In case the supply of a product suffers from a supply scarcity, specially if its demand is high, the price is bound to increase as well. Therefore, the correlation between the price of a product and its demand is probably going to reach a considerable amount of correlation.

The PCA is carried out by diagonalizing the correlation matrix and breaking down the matrix into its eigenvalues  $\lambda_i$  and eigenvectors  $\vec{v}_i$ , so that the correlation matrix is represented as  $M_c = \sum_i \lambda_i \vec{v}_i \vec{v}_i^T$ . Given the symmetry of the covariance matrix  $M_c$ , the eigenvectors associated to the eigenvalues form an orthonormal basis (in case the eigenvalues are different across the eigenvectors).

In order to detect which are the principal components among all the eigen-

### 3. BACKGROUND

---

vectors, an observation is made to how large the eigenvalues associated to the eigenvectors are. If a minority of the components present a substantially larger eigenvalues, then the eigenvectors associated to those few eigenvalues constitute the principal components of the data. The principal components provide insight about the inner relationships existing among the variables in an endeavour to capture the largest amount of information in a lower dimension. All in all, classical approaches to PCA take  $\mathcal{O}(n^2)$  where  $n$  is the dimension of the vector space, for instance  $\mathbb{R}^n$ .

#### 3.3.2.2 Quantum methodology

For the case of qPCA (quantum principal component analysis) [5] on a standard dataset, a data vector  $\vec{v}_i$  (with dimension  $n$ ) is randomly selected and mapped into a quantum state  $|\psi_i\rangle$  with the use of qRAM (quantum random access memory) [6] method. The resulting quantum state  $|\psi_i\rangle$  has a total of  $\log n$  qubits, while the qRAM requires  $\mathcal{O}(n)$  operations split into  $\mathcal{O}(\log n)$  steps that work in parallel.

A density matrix of a quantum state provides knowledge about the probability of an arbitrary outcome occurring when performing a projective measurement of the quantum system. Particularly, a density matrix is a positive semi-definite Hermitian operator of trace one of the quantum system's Hilbert space. Considering the quantum state  $|\psi_i\rangle$ , whose probability is  $p_i$ , the probability for an outcome  $m$  with the projective measurement when using projectors  $\Pi_m$  is calculated as

$$p(m) = \sum_i p_i \langle \psi_i | \Pi_m | \psi_i \rangle = \text{Tr} \left[ \Pi_m \left( \sum_i p_i |\psi_i\rangle \langle \psi_i| \right) \right] \quad (3.2)$$

where the density matrix  $\rho$  is defined as follows:

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| \quad (3.3)$$

The data vector  $\vec{v}_i$  has been randomly selected, as such the density matrix associated to the quantum state is  $\rho = (\frac{1}{N}) \sum_i |\vec{v}_i\rangle \langle \vec{v}_i|$  where  $N$  is the total amount of data vectors. This assumes that the quantum state of each of the data vectors is equally probable. The density matrix bears a striking similarity with the covariance matrix employed in the classical methodology, with the only difference being the probability factor.

Multiple iterations of sampling the data with density matrix exponentiation [7] and quantum phase estimation [8] algorithm, which looks for the eigenvalues and the eigenvectors of a matrix, lead to representing the data vector as a quantum vector  $|\vec{v}\rangle$  and break it down into the principal components  $|c_k\rangle$ . That is the largest eigenvalues and eigenvectors of the covariance matrix. This quantum algorithm takes time complexity and query complexity  $\mathcal{O}((\log n)^2)$ , which is significantly

and exponentially better in performance compared to the standard and classical approach of calculating PCA.

### 3.3.3 Quantum support vector machines

The basic methods for supervised machine learning include support vector machine and perceptrons. In these methods, the objective is to draw hyperplanes in the variables' space, so that different classes of data points don't fall into the same subspace restricted by the edges and borders of the hyperplanes. In other words, each subspace should contain mostly (if not only) data points that are labeled with one class in particular. This allows to identify the class of the data depending of the subspace it fall into and make an accurate prediction.

These models learn parameters that set the boundaries of the hyperplanes, that is, the weights are adjusted to separate data points that fall into different categories. The main appeal of using SVM is the capability to search for nonlinear hyperplanes through kernel methods. In particular, classifiers based on SVM are specially powerful in the image segmentation department as well as in biological analysis.

#### 3.3.3.1 Classical methodology

From a data set with data points  $x_i$  and labeled with  $y_i \in \{-1, 1\}$  for each one of them, the objective is to find the hyperplane following the equation  $wx - b = 0$  that optimally separates the space for the best classification of the data points. The weight vector, whose weights are yet to be learned in the training process, correspond to the normal vector of the hyperplane, which when normalized, the nearest points of different sides and different classes are in the hyperplanes  $w^T x - b = \pm 1$ . This is shown in figure 3.3.

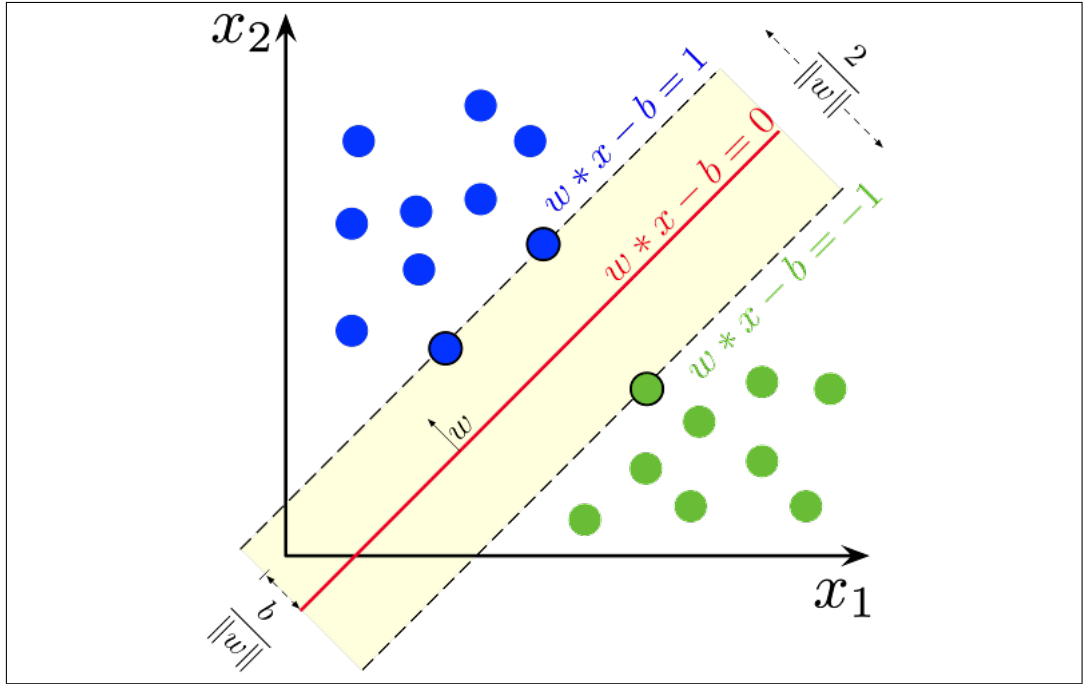
Actually, the distance from the hyperplane to the closest point (or points in some cases) in each of the sides is  $\frac{1}{|w|}$ . Given the fact that the objective is to maximize the distance as much as possible for the sake of splitting the space optimally, the method equivalently is trying to minimize  $|w|$  with the constraint  $\forall i [y_i(w^T x_i - b) \geq 1]$ . The minimization process is carried out with Lagrange multipliers as follows:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}|w|^2 - \sum_i \lambda_i [y_i(w^T x_i - b) - 1] \quad (3.4)$$

When calculating the partial derivatives of  $\mathcal{L}$  with respect to  $\lambda$  equal to 0, end up being the constraints. For the case of the partial derivatives of  $\mathcal{L}$  with respect to  $w$  equal to 0 gives the following result:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_i \lambda_i y_i x_i = 0 \quad (3.5)$$

### 3. BACKGROUND



**Figure 3.3:** SVM illustration in a two dimensional space. Source: Wikipedia.

Therefore,  $w$  will be adjusted according to the support vectors (the closest data points), as noted when isolating the normal vector of the hyperplane  $w$  in the equation above.

$$w = \sum_i \lambda_i y_i x_i \quad (3.6)$$

Lastly, the partial derivative of  $\mathcal{L}$  with respect to  $b$  equal to 0 is obtained:

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_i \lambda_i y_i \quad (3.7)$$

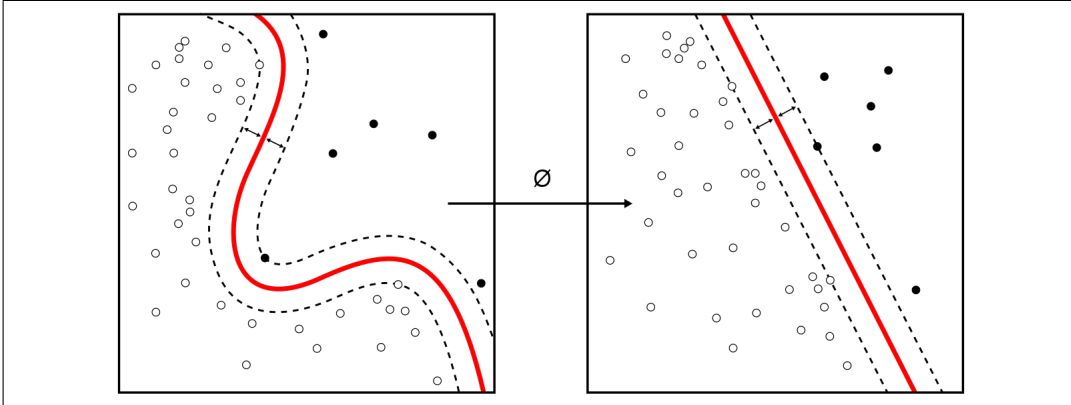
These finding and then placed in the original Lagrange operator, which result in:

$$\sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j (x_i x_j) \quad (3.8)$$

The strategy is to maximize this expression 3.8 by finding the  $\lambda_i$  that does it with the constraint 3.7. The algorithm that follows this strategy is the coordinate descent based one.

In many cases, however, it is not possible to separate the space into subspaces that correspond to a specific class effectively with a hyperplane given how spread is the data as shown in figure 3.4. The solution to that problem is to transform every





**Figure 3.4:** Kernel function on a data set for a support vector machine. Source: Wikipedia.

data point to a new feature space in which a hyperplane can be placed properly. This transformation is the kernel function  $K(x_i, x_j)$  that acts on the data points and replaces the dot product on equation 3.8.

The maximization problem from equation 3.8 is transformed into the following system of linear equations with some tweaks, where every pair of data points kernel function is stored as  $K_{ij} = K(x_i, x_j)$ :

$$\begin{pmatrix} 0 & 1 \\ 1 & K \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ y \end{pmatrix} \quad (3.9)$$

In order to calculate the time complexity running a support vector machine algorithm, consider the datapoints  $x_i \in \mathbb{R}^n$  from a dataset where  $1 \leq i \leq m$ , so that there are a total of  $m$  data points. Each entry  $K_{ij}$  takes  $\mathcal{O}(n)$  and, therefore, all entries from  $K$  take  $\mathcal{O}(m^2n)$  to compute. A system of linear equations takes time  $\mathcal{O}(m^3)$  to be solved in a classical computer, which in total leads to the entire process having a time complexity  $\mathcal{O}(m^2(n + m))$ .

### 3.3.3.2 Quantum methodology

Following the same trend as the classical method, the quantum support vector machine [9] is a staple quantum machine learning algorithm. The most recent methods to perform quantum support vector machine lean on using a least-squares quantum support vector machine that uses the qBLAS subroutines (an open source quantum basic linear algebra and quantum simulation library) effectively.

In the classical methodology, the major computation cost came from both the calculations for every pair of data points' kernel function  $K_{ij}$  and solving a system of linear equations. Both of these problems can be tackled with quantum algorithms to reduce the computational cost.

The dot product of two different data points  $x_i \cdot x_j$ , whose  $|x_i|$  and  $|x_j|$  are

### 3. BACKGROUND

known, can be computed in this form:

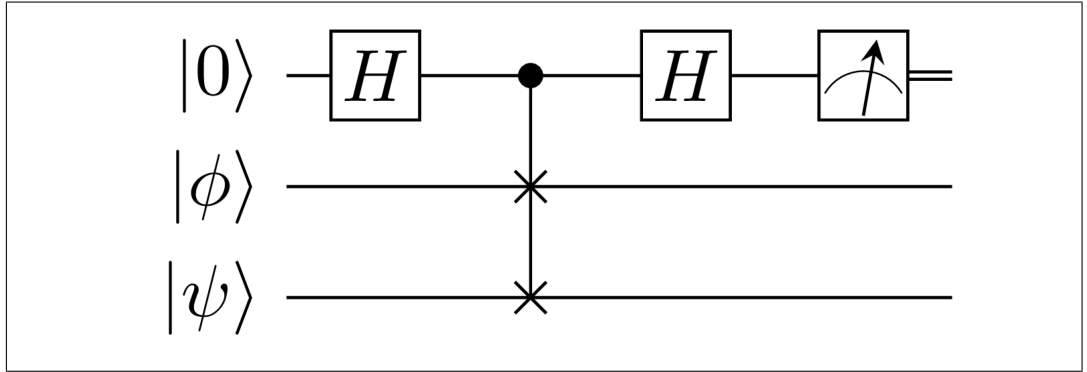
$$x_i \cdot x_j = \frac{|x_i|^2 + |x_j|^2 - |x_i - x_j|^2}{2} \quad (3.10)$$

This trims the problem to calculating the distance  $|x_i - x_j|^2$ . In order to accomplish this, two quantum states are defined  $|\psi\rangle$  and  $|\phi\rangle$ , which require qRAM to be built and stored on demand.

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|x_i\rangle + |1\rangle) \quad (3.11)$$

$$|\phi\rangle = \frac{1}{\sqrt{|x_i|^2 + |x_j|^2}}(|x_i||0\rangle - |x_j||1\rangle) \quad (3.12)$$

The final step is to carry out a swap test on both quantum states  $|\psi\rangle$  and  $|\phi\rangle$  and make a measurement afterwards, as shown in figure 3.5. The quantum state after the swap test is illustrated in equation 3.13, which indicates that the probability of measuring 1 is 0 if both quantum states  $|\psi\rangle$  and  $|\phi\rangle$  are equal (which is essential to fulfill the requirements every distance function must fulfill to be regarded as such). Once the swap test results are obtained, the desired distance is calculated as  $p_{ij} \sqrt{|x_i|^2 + |x_j|^2}$ , where  $p_{ij}$  refers to the probability of measuring 1 in the swap test. All in all, the time complexity of calculating this distance is  $\mathcal{O}(\log n)$ .



**Figure 3.5:** Quantum swap test. Source: Wikipedia.

$$\frac{1}{2}(|0\rangle(|\phi, \psi\rangle + |\psi, \phi\rangle) + |1\rangle(|\phi, \psi\rangle - |\psi, \phi\rangle)) \quad (3.13)$$

In order to solve the system of linear equations, a matrix inversion is required to the equation 3.9, which is achieved by using the HHL algorithm (through quantum phase estimation and matrix inversion). This algorithm is an indispensable subroutine to invert systems of equation and find solutions to systems of equations formulated as  $A\vec{x} = \vec{b}$  in quantum computers. Both vectors are quantified as quantum states  $|x\rangle, |b\rangle \in \mathbb{C}^n$  with  $\log_2 n$  qubits. The matrix A is assumed to be

Hermitian, in case this is not true, the space is expanded to ensure this fact. The presented equation  $A|x\rangle = |b\rangle$  is solved multiplying both sides with the inverse of the matrix  $A^{-1}$ .

The HHL algorithm assumes that  $|b\rangle = \sum_n b_n |E_n\rangle$ .  $|E_n\rangle$  is an eigenvector of  $A$  and its eigenvalue  $\lambda_n \geq \Lambda$ . With quantum phase estimation applied under  $A$ , the eigenvalue  $\lambda_n$  is calculated. Afterwards, the ancillary qubit is rotated with an angle of  $\arcsin \frac{\Lambda}{\lambda_n}$  and the quantum phase estimation is undone, which results in:

$$\sum_n b_n |E_n\rangle \left( \frac{\Lambda}{\lambda_n} |1\rangle + \sqrt{1 - \frac{\Lambda^2}{\lambda_n^2}} |0\rangle \right) \quad (3.14)$$

If the ancillary qubit is measured with observation 1, each of eigenstates is divided by  $\lambda_n$ , affecting the inverse of the matrix  $A^{-1}$ . The state preparation circuit is required to be applied  $\mathcal{O}(\frac{\|A\|}{\Lambda})$  times once amplitude amplification is also applied. The HHL algorithm has computational complexity of  $\mathcal{O}((\log n)^2)$  to find the  $|x\rangle$ , while the best classical algorithm in comparison takes  $\mathcal{O}(n \log n)$  to find the  $\vec{x}$ .

### 3.4 The problem of loading classical data into quantum computers

Before a quantum computer starts working the information and data must be loaded and passed into it. Likewise, when outputting the result, it has to be processed with classical means. All this leads to certain bottleneck and overhead issues that occur in certain algorithms and problem scenarios.

Many quantum algorithms such as HHL algorithm, least squares, qPCA, quantum support vector machines and quantum methodologies that require loading classical data first are faced with this problem, that is, loading a large amount of data beforehand. It can be quite taxing in the before and after algorithm parts in some cases which can lead to requiring an exponential time [10]. Despite the fact that qRAM can elude the problem, the shortcomings of its use are too costly to assume when the amount of data is enormous.

Most of the linear algebra problems, as well as combinatorial optimization problems depend on using qRAM in large scale to circumvent the input data problem. One of the exceptions in the linear algebra set of quantum algorithms is the quantum algorithm to perform a topological analysis of the data [11].

In the output problems side of things, the situation is bleaker. Leaving least squares fitting and quantum support vector machines aside, linear algebra problems suffer from output problems. This is can be observed both in the solution vector  $|x\rangle$  of the HHL algorithm or the principal components of the qPCA, whose classical quantities are exponentially difficult to estimate.

With all this put into consideration, the optimization of quantum algorithms is still far from over to fulfill the necessities the machine learning task requires to solve. Among these problems, effectively estimating the cost of running these algorithms and building quantum computers according to the needs of the machine learning problem are specially crucial for offering a quantum approach to machine learning tasks while still being feasible to manage.

## 3.5 Deep quantum learning

Classically approached neural networks provide unparalleled performance and methods for machine learning. Deep quantum learning networks are also following the same track. These quantum networks are usually built into quantum information processors, among which are included quantum annealers (for combinatorial optimization problems, for instance) and programmable photonic circuits [12, 13, 14].

Mainly, the advantage of deep quantum learning over classical means is the lack of need on vast and general purpose quantum computers. Quantum annealers, as mentioned above, are excellent examples of quantum processors whose construction and scale-up potential are easier than the ones found in general purpose quantum computers. As a result, quantum anneals are candidates to consider when build a deep quantum neural network and are available to the market. One of the providers of quantum annealers service is the D-Wave platform for quantum computing. Their quantum anneals are specially built as trainable transverse Ising models with the possibility to return the thermal states of both classical and some quantum spin systems.

The main incentives for developing quantum approached machine learning algorithms rely on their ability to work with quantum data, which allows a reduced representation of classical data than can be stored in qRAM based methods and retrieved quickly and reliably, and the quantum speedup potential that some methods have already showcased in contrast to their classical counterpart.

Nevertheless, there are some hindrances that difficult the path in some cases. The input and output problem have already been discussed with their costly exponential estimation on quantities that cause bottlenecks and overheads. Moreover, the activation function is linear in quantum mechanics, unlike the nonlinear activation functions that are found usually in classical neural networks.

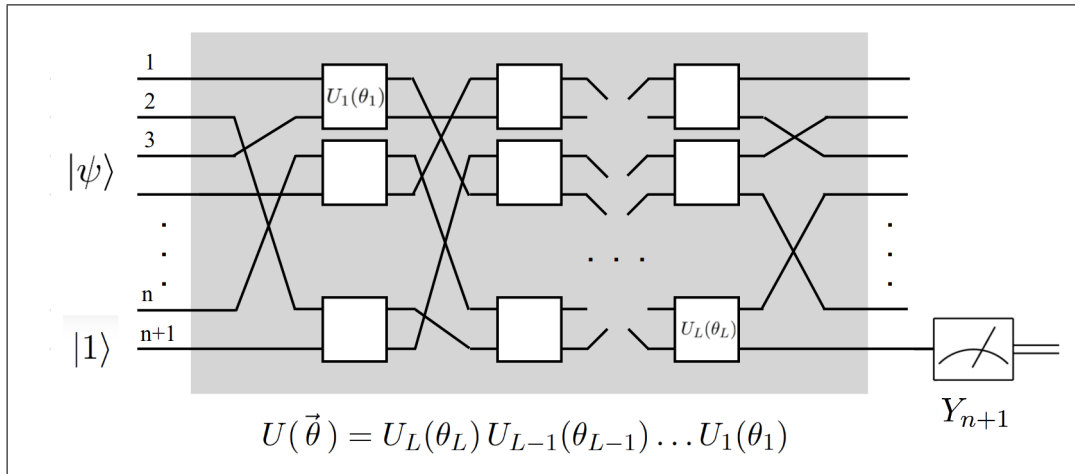
Another thing to consider is the current limitation of quantum computers themselves. In classical neural networks finding deep neural networks with many connections it is not far fetched, however, such difficult and complex connections are sometimes difficult to fit or embed into the quantum computer itself, which effectively makes the current development in quantum computers a constraint for more complex quantum neural networks to come to fruition, not to mention the placement and layout of the qubits themselves in the quantum computer that limits

the embedding process itself.

Finally, backpropagation is another of the most difficult stages to implement in a quantum mechanics based algorithm. In a classical environment, backpropagation is conducted in every output layer of the network during the training phase. This goes against one of the properties of quantum mechanics, which is the quantum superpositions of many qubits being lost in the process.

### 3.5.1 Quantum feedforward neural networks

Since there are many examples on how to propose a quantum feedforward neural network, in order to illustrate a working quantum feedforward neural network the one presented here is used [1] and showed in the figure 3.6.



**Figure 3.6:** Quantum feedforward neural network from [1].

In this example, quantum gates substitute the classical neural network layers and the information processed by the combination of different quantum gates is the input quantum state  $|\psi\rangle$  and the ancillary qubit  $|1\rangle$ . The commonly initialized and trained weights and biases are now changed by the parameters  $\theta_i$  of the quantum gates, which generally refer to how much the quantum state is transformed depending on the particular type of quantum gates utilized. Finally, the output quantum state is measured and the class or label (out of two possibilities) corresponding to the input quantum state is selected. The entire quantum circuit is compressed into an unitary operation  $U(\theta)$ .

In order to train the quantum feedforward neural network the loss function proposed is:

$$C(\theta, z) = 1 - l(z)\langle z, 1|U^\dagger(\theta)Y_{n+1}U(\theta)|z, 1\rangle \quad (3.15)$$

The input to the quantum gate is  $z$  and the label function  $l(z)$  returns the true label of the input  $\pm 1$ . If the network labels the input correctly the cost function return 0 otherwise a larger value is returned. The parameter optimization function

is stochastic gradient descent (SGD) to minimize the loss function as much as possible. The process to calculate the gradient differs to the standard however, since no backpropagation is involved in this quantum neural network. Many iterations are processed to gather enough concluding information to calculate the partial derivatives, that is, the output is evaluated to have a definitive output and, therefore, ensure a precise gradient.

One of the benefits of this particular quantum feedforward neural network lies in the stability of the gradient and its reluctance to explode, which is a common issue in many of the classical machine learning neural networks (often solved by approaching the gradient function more cautiously). Since there is no nonlinear function, learning the patterns that classify different cases accordingly can be put into question. However the label function works, despite the requirement to build a quantum circuit with exponential depth depending on the particular case.

#### 3.5.2 Quantum convolutional neural networks

Convolutional neural networks excel in image recognition tasks and image processing or detection. A quantum convolutional neural network has been proposed that offers an interesting quantum application [2] and is shown in figure 3.7.

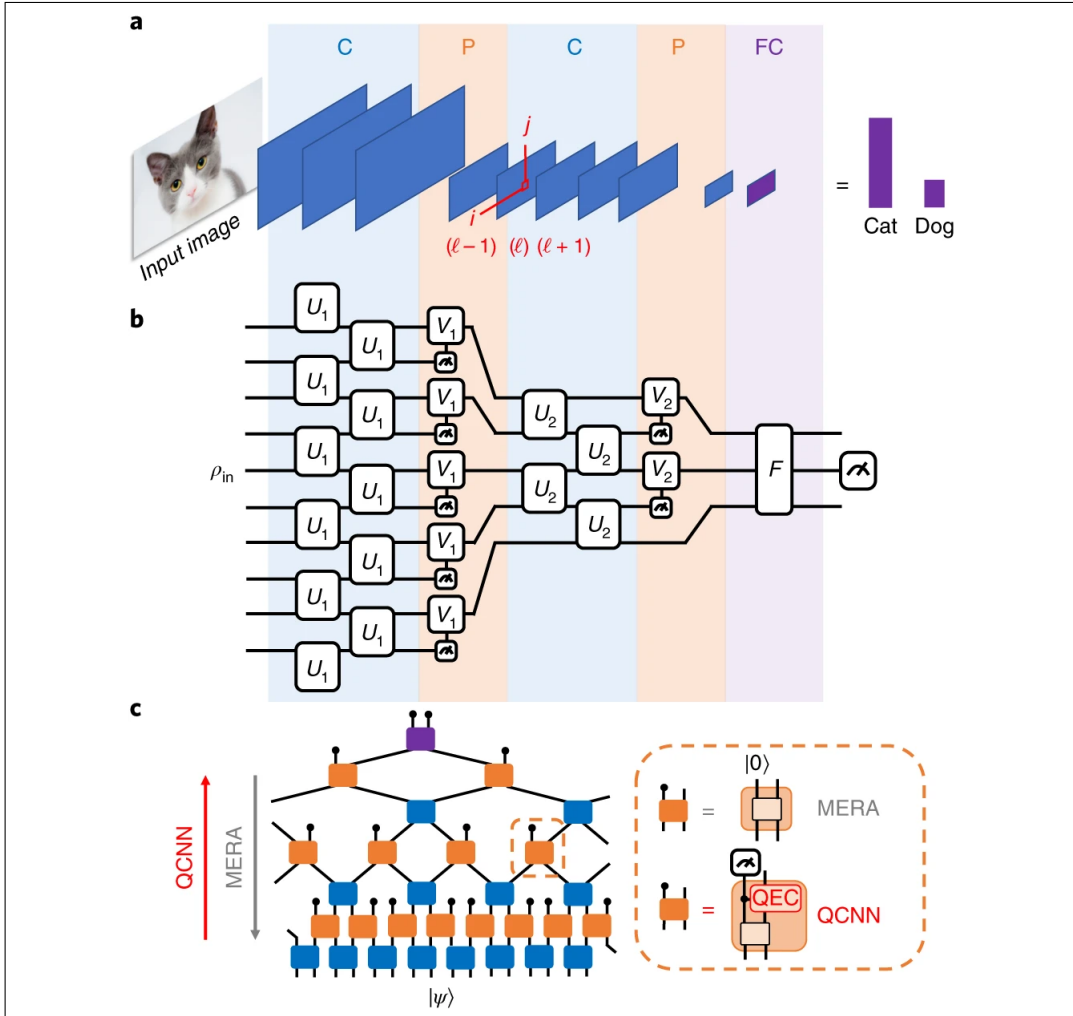
For the classical version of CNNs, three main layers constitute the foundation of CNNs. In the convolutional layers (C) the image is processed and read into feature maps, in the pooling layers (P) the feature maps are reduced according to the pooling technique applied (max-pooling for example) and finally the features obtained from the convolution and pooling layers are processed and classified with fully connected layers (FC).

For the quantum version of CNNs, the filtering, the pooling and the classification are replaced with unitary operations on qubits. In the convolutional layer  $U$ , each filter  $i$  is changed with the unitary operator  $U_i$  acting in two qubits simultaneously across all qubits that are next to each other. In the pooling layer  $V$ , each pooling layer number  $k$  is modified after measuring half of the total qubits and applying a one qubit operation  $V_k$  on the rest of the qubits, which is affected by the outcome of the qubit measured next to it. The pooling layers  $V_k$  operations include nonlinearity, which allows for more complex patterns to be detected.

The final part of the QCNN is similar to the quantum feedforward neural network, that is, quantum gates replace the fully connected layers. As in the previous case, there is no backpropagation and the optimization function is, once again, stochastic gradient descent.

#### 3.5.3 Quantum Boltzmann machines

Quantum Boltzmann machines are a prime example of quantum neural networks. One of the strong points for quantum methods is their capacity to cause the system to have a quadratically faster thermalization [15, 16, 17, 18]. Another improvement



**Figure 3.7:** Different approaches to build CNN from [2]. A classical CNN in **a**, a quantum CNN in **b** and QCNN comparison in **c**.

in the quantum Boltzmann training is the improved method of sampling. The neuron activation event has a stochastic behaviour in Boltzmann machines, which requires multiple iterations to find successful results. This also has an impact on how updating weight of the Boltzmann machine affect the performance achieved. In opposition to the limitation classical approaches posses, quantum Boltzmann machines make use of quantum coherence during the training phase, which drastically (in a quadratic rate) reduces the number of samples required to improve the performance. Furthermore, the addition of qRAM provides faster access both in time and number to the data in contrast to classical means [17].

A classical Boltzmann machine has two type of neurons, either visible (set of visible neurons  $V$ ) or hidden (set of hidden neurons  $H$ ). These neurons have one binary value ( $v_i, h_i \in \{0, 1\}$ ). The purpose of training a Boltzmann machine is to learn the probability distribution of the training data and make the visible neurons

### 3. BACKGROUND

---

act according to that probability distribution  $P(V)$ .

$$P(V, H) = \frac{1}{Z} e^{-E(V, H)} \quad (3.16)$$

The normalization factor  $Z$  is the partition function and the energy function  $E(V, H)$  corresponds to the restricted Boltzmann machine case

$$E(V, H) = \sum_i a_i v_i + \sum_i b_i h_i + \sum_{i,j} v_i W_{ij} h_j \quad (3.17)$$

where  $a_i$  and  $b_i$  are the bias of each neuron and  $W$  is the weight matrix with visible neurons as rows and hidden neurons as columns. This is the case for the restricted Boltzmann machines since there are no connections between neurons of the same type. The objective in the training phase is to approximate the real probability distribution  $P_{data}(V)$  through the probability distribution generated by the Boltzmann machine  $P_{BM}(V)$ . The similarity between both probability distributions is measured using the Kullback–Leibler divergence for every existing states of  $V$ :

$$KL = \sum_V P_{data}(V) \ln \left( \frac{P_{data}(V)}{P_{BM}(V)} \right) \quad (3.18)$$

Since the probability distribution generated by the machine  $P_{BM}(V)$ , the energy function described by the weights affect the probability distribution. An optimization function using a gradient descent algorithm looks for the partial derivative of  $KL$  with respect to each weight of the weight matrix  $W$  and subtracts the value of the partial derivative to the corresponding weight each case.

On the other hand, a quantum Boltzmann machine learns Hamiltonian parameters  $w_i$  where the input state  $\rho_{data}$  for a specific set  $H_i$  can be approximated with according to [19, 20]:

$$\sigma = \frac{e^{-\sum_i w_i H_i}}{Tr(e^{-\sum_i w_i H_i})} \quad (3.19)$$

The quality of the approximation for the visible neurons can be measured using the quantum relative entropy. This measurement's upper boundary is the distance between the two states (when both  $\rho$  and  $\sigma$  are equal). Therefore, minimizing this distance means minimizing the error and therefore improving the performance of the Boltzmann machine.

$$S(\rho_{data} || \sigma) = Tr(\rho_{data} \log(\rho_{data}) - \rho_{data} \log(\sigma)) \quad (3.20)$$



The quantum relative entropy, despite being a fantastic measurement method for two states, it can be hard to learn in practice. The gradient of the relative entropy, on the contrary, is simpler to estimate:

$$\partial_{w_i} S(\rho_{data} || \sigma) = Tr(\sigma H_i) Tr(\rho H_i) \quad (3.21)$$

With gradient descent the weight vector  $\vec{w}$  is updated for  $\eta > 0$ :

$$w_{t+1}^{\vec{}} = \vec{w}_t - \eta \nabla S(\rho || \sigma) \quad (3.22)$$



## Related work

For starters, quantum computing has been a slumbering technology for quite a while until technological advances and progress in the integration of quantum computing as an integral and standardized part of computation have been steadily made. In recent years, different methodologies developed have allowed plenty of experiments and improvements in the application of quantum computing for solving machine learning tasks of diverse natures. In this section some of these methodologies are gathered and briefly discussed to understand the motivation behind the sudden raise of quantum computing and explain the intrinsic intention behind building quantum classifiers.

### 4.1 Associated topics

In many works, the progress of quantum machine learning is discussed [21, 3] and how combining quantum computing with conventional machine learning techniques may yield important benefits in the long run. In the state of the art part, various quantum approaches already have been presented among which are included quantum PCA [5], quantum deep learning methods [22] or quantum support vector machines [23].

These methodologies showcase a leap in attainable performance that might suit the interest going forward into the future as resources for quantum computing are more available. In the same line, the idea of building a quantum machine learning model arises naturally [24]. There is already a number of costly algorithms to handle through classical means that are explored and offer better performance when processed with quantum computing mechanisms reaching exponential advantage [25, 26].

Another aspect in quantum computing is the representation of the data itself within the boundaries of quantum mechanics. These representations include finding a suitable mapping to quantum environments for features in Hilbert spaces

[27]. These mappings include embedding techniques such as quantum embedding kernels [28] or quantum embeddings for machine learning purposes [29]. QUBO (quadratic unconstrained binary optimization) problems (NP-hard problems) are already processed in quantum computers using embedding algorithms [30, 31].

Quantum variational classifiers offer parameterized quantum circuits to build a machine learning model from. Quantum variational classifiers have been used for learning particle physics [32]. Quantum classification for predicting classes through measuring an observable in selected quantum states for supervised learning problems is a particularly robust approach in which quantum variational classifiers play a central role [33].

## 4.2 Contributions

The value of the project, in contrast with the presented work above, is expanding the notion of variational quantum classifiers into hybrid classical quantum classifiers. By challenging the quantum variational classifier with unbalanced data and representing the quantum classifier as a quantum circuit, two main possibilities appear: the quantum variational classifier can be integrated as a separate quantum layer into hybrid functioning machine learning models in abstract thinking, and, apart from that, current existing limitations when dealing with complex and unbalanced data can be determined.

In other words, it is firstly a combination between observing what quantum computing offers in terms of capabilities in the field of machine learning for supervised machine learning problems with the current technological limitations and resource hungry quantum systems and, secondly, whether the discussed improved pattern recognition of quantum systems mixed together with the current performance level of classical systems produces positive results when building a hybrid classical quantum classifier following the quantum variational classifier scheme.

## Variational quantum circuits

In the context of quantum algorithms, one of the most prominent examples of a quantum algorithm that depends on some free and trainable parameters is the variational quantum circuit. Variational circuits, often called quantum circuits with parameters, follow the same common standards of many other quantum circuits, which feature three fundamental characteristics.

The first part consists on the initialization of the quantum state at the beginning of the quantum circuit. This requires an embedding process of the classical information into the quantum information, so that the initial states can be processed by the quantum circuit.

The second part corresponds to the quantum circuit  $U(\theta)$  itself. The set of trainable  $n$  parameters of the quantum circuit  $U(\theta)$  are parameters  $\theta_i$  for  $0 \leq i < n$ , otherwise called the vector of parameters  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_{n-1})$ .

Last, the measurement of the final quantum state in the quantum circuit. The observable  $\hat{B}$  is built around the wires or qubits of the quantum circuit and the selection of the observables is up to the preferred subset of wires without limitation.

The expectation values  $f(\theta)$  of the parameters for each of the quantum circuits they are used on are tied to the specific observable  $\hat{B}$  and express the cost associated to solve the pertinent task at hand. Through these expectation values a cost function is defined to assess the performance of the quantum circuit when solving the task. Afterwards, the free parameters  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_{n-1})$  are submitted for tuning them and, therefore, the parameters can be optimized reducing the cost function. An example of a cost function  $f(\theta)$  with vacuum initial states to draw the expectation values goes as follows:

$$f(\theta) = \langle 0|U^\dagger(\theta)\hat{B}U(\theta)|0\rangle \quad (5.1)$$

The training process of a variational quantum circuit differs not from classical optimization algorithms. As such, the classical system sends requests to the quan-

tum system and receives an output from the quantum side. The quantum reply is transformed into classical data to optimize through an iterative method the best possible parameters  $\theta^*$  to solve the given task.

Variational quantum circuits are currently shining in popularity given the technological constraints of quantum devices (the amount of qubits in physical devices or the lack of fine control in optimal embedding processes). These quantum circuits are only capable of short quantum gate sequences when constructed, given the error of the output when fault tolerance is not included in each of the quantum gates. The core of every quantum algorithm is broken down into standard elementary operations being inherently included in quantum systems.

The underlying value of the variational algorithm resides in having the quantum circuit's parameters optimized in the classical machine which allows, irrevocably, to merging the procedure into a single pipeline. In result, the parameters to be tuned define the quantum algorithm without having to resort to fixed elementary gate sets. The optimization process also supervises the systematic errors and corrects them in the process.

## 5.1 Quantum circuit configuration

The input information into the quantum circuit for the quantum gates that will be following the quantum variational algorithm is divided into trainable parameters  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_{n-1})$  and fixed parameters  $x = (x_1, x_2, x_3, \dots, x_n)$ . The common pattern is to use these fixed parameters as the data information. With this taken into account, the classical information (both  $\theta$  and  $x$ ) is transformed into quantum information given by the quantum state  $U(x; \theta)|0\rangle$ .

The quantum information is transformed the other way around into classical information through the calculation of the expectation value of the observable  $\hat{B}$ .

$$f(x; \theta) = \langle \hat{B} \rangle = \langle 0|U^\dagger(x; \theta)\hat{B}U(x; \theta)|0\rangle \quad (5.2)$$

There is no established rule as to how the quantum gates or the circuit design are adjusted, except for the parameters  $\theta$  being the arguments of the quantum gates. In other words, the variational algorithm allows for great freedom when building a quantum circuit. The quantum circuit also allows the possibility to include quantum gates with fixed parameters beyond only data entries if specified.

## 5.2 Quantum embedding

In order to actually feed data into a quantum circuit a quantum representation of the data input is required. For that end, a quantum feature map is utilized which transforms classical data into quantum states in a Hilbert space (as presented in [34] and [35]). Given a classic data vector  $x$ , the data vector is transformed into

the quantum state  $|\phi_x\rangle$ . This is a crucial step that involves the designing of the quantum algorithm and its computational power.

Given a dataset  $D$  with the corresponding  $K$  data points and  $n$  variables each

$$\begin{aligned} D &= \{x_1, x_2, \dots, x_K\} \\ x_i &= \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\} \end{aligned} \quad (5.3)$$

where  $x_i$  is a data vector and  $x_{i,j}$  a feature of the data vector. There are various methods to embed the data into the quantum system composed of  $m$  qubits or qumodes in case discrete or continuous variables are involved. Some prime examples are included below.

### 5.2.1 Basis embedding

Basis embedding, as noted by the naming of the embedding method, consists on translating the inputs into computational basis states of a qubit system. For this to work, data of classical nature is required to be adopting the form of binary strings. The binary string is translated into a quantum state bit-wisely in order to embed it into the specific computational basis of the quantum subsystem. For instance, if the input data vector  $x = 110$  had to be embedded, then the quantum state  $|110\rangle$  formed with three qubits would be its representation. In this fashion, one bit of classical information is encoded into a quantum subsystem for an equivalent one quantum subsystem.

Let  $D$  be considered the classical dataset being embedded with basis embedding method. The nature of the data points  $x_i$  are required to be binary strings composed of  $n$  bits each.

$$x_i = \{b_1, b_2, b_3, \dots, b_n\} \text{ with } b_i \in \{0, 1\} \quad (5.4)$$

Given the assumption that every feature is represented with only one bit, each of the data inputs  $x_i$  can be embedded into the quantum state  $|x_i\rangle$ . Therefore, the bare minimum of quantum subsystems (qubits for example) described by  $m$  is set to be the number of features (or bits in this case)  $n$  in each binary string. The dataset can be represented in result as the superpositions of the computational basis states:

$$|D\rangle = \frac{1}{\sqrt{K}} \sum_{k=1}^K |x_k\rangle \quad (5.5)$$

Consider the following two binary strings  $x_1 = 10$  and  $x_2 = 01$  as the complete classical dataset. The basis embedding of the classical data is represented using two

qubits (at the very least)  $|x_1\rangle = |10\rangle$  and  $|x_2\rangle = |01\rangle$  and the dataset is represented in the quantum subsystem as

$$|D\rangle = \frac{1}{\sqrt{2}}|10\rangle + \frac{1}{\sqrt{2}}|01\rangle \quad (5.6)$$

As long as, the number of data vector  $K \ll 2^n$ , the basis embedding of the dataset will be sparse. The amount of basis states available for  $n$  bits is  $2^n$ , though. Moreover, integer spaces are also suitable for basis embedding by representing the integer value with binary coding and then applying the basis embedding to their binary representation, which extends the possibilities of the basis embedding beyond only binary data.

## 5.2.2 Amplitude embedding

In quantum mechanics, a probability amplitude establishes a relation between the quantum state vector of a quantum system and the measurements results of observations made into the quantum system. In mathematical terms, a probability amplitude refers to a complex number used to explain and define the behaviour of the quantum systems. In particular, a probability density is defined through applying the squared modulus to the aforementioned probability amplitude, that is, to the complex number.

The amplitude embedding method is a embedding technique to encode the classical data into the amplitudes of a quantum state. Given a datapoint  $x$  conformed by  $M$  dimensions, the quantum representation is achieved with the amplitudes of a quantum state  $|\psi_x\rangle$  with  $n$  qubits as

$$|\psi_x\rangle = \sum_{i=1}^M x_i |i\rangle \quad (5.7)$$

where  $M = 2^n$ ,  $x_i$  corresponds to the  $i$ -th dimension of the datapoint and  $|i\rangle$  is the  $i$ -th computational basis state. Unlike with basis embedding, amplitude embedding allows both integer and floating point data types, they can even appear simultaneously. By way of illustration, consider the following four dimensional data vector with floating point values  $x = (2.4, 0.0, -1.4, 4.5)$  to be embedded via amplitude embedding. The data vector  $x$  must be normalized to  $\hat{x}$ :

$$\begin{aligned} \|x\| &= \sqrt{2.4^2 + 0.0^2 + (-1.4)^2 + 4.5^2} = \sqrt{27.97} \\ \hat{x} &= \frac{1}{\sqrt{27.97}} [2.4|00\rangle - 1.4|10\rangle + 4.5|11\rangle] \end{aligned} \quad (5.8)$$



When embedding the entire dataset  $D$  presented at the beginning using amplitude embedding, a convenient way to visualize it is through merging every data point  $x^K$  into a single vector as pictured below

$$\alpha = C\{x_1^1, \dots, x_M^1, x_1^2, \dots, x_M^2, \dots, x_1^K, \dots, x_M^K\} \quad (5.9)$$

where  $C$  corresponds to the normalization constant being used. The dataset vector is normalized, so that the amplitude vector  $|\alpha|^2 = 1$ . Once normalized, the dataset can be represented in terms of the computational basis in this form

$$|D\rangle = \sum_{i=1}^{2^n} \alpha_i |i\rangle \quad (5.10)$$

where  $\alpha_i$  is the  $i$ -th element of the amplitude vector  $\alpha$  and  $|i\rangle$  is the  $i$ -th computational basis state. In total, the amount of probability amplitudes to be encoded is  $M \times K$ . Therefore, since  $2^n$  amplitudes are administered with a  $n$  qubits system, amplitude embedding needs to meet the requirement

$$n \geq \log_2(MK) \quad (5.11)$$

Nevertheless, commonly the number of amplitudes to embed won't be a exact power of 2. In these cases,  $M \times K$  is less than  $2^n$  and the leftover amplitudes are padded to the amplitude vector  $\alpha$  to fill the gap [36].

In order to illustrate the padding technique, consider a dataset  $D$  with 2 data vectors and 3 variables each, in consequence there are  $2 \times 3 = 6$  probability amplitudes to be embedded. Yet, the minimum amount of qubits to use is  $\lceil \log_2(6) \rceil = 3$  qubits at least. However, with 3 qubits there are  $2^3 = 8$  states for amplitude embedding from which only 6 are required. As a result,  $2^3 - 6 = 2$  constants are added at the end of the amplitude vector to compensate.

### 5.2.3 Angle embedding

A more geometrical embedding for floating point data is angle embedding. The angle embedding consists on encoding floating point values as if they were parameters of a standard elementary operation, such as a rotation. Consider the floating point value  $x$ , the embedding of the value into the quantum state  $|\psi_x\rangle$  is achieved through the following mapping

$$|\psi_x\rangle = R_j(x)|0\rangle = e^{\frac{-ix\sigma_j}{2}}|0\rangle \quad (5.12)$$

where  $j \in \{x, y, z\}$  corresponds to the axis rotation in the Bloch sphere and  $\sigma_j$  is the Pauli matrix associated to the half spin rotation of the axis  $j$ . Generally, either the axis  $j = x$  or  $j = y$  are acceptable choices for rotation with the first

computational basis state. However, the axis  $j = z$  is not a viable choice, since the value  $x$  is mapped to the state  $|0\rangle$  as a result of  $R_z(\theta)$  operation having an eigenvalue of 1 for the eigenvector  $|0\rangle$ . Therefore, the value is lost in the mapping process.

The Pauli rotation operations being used are  $2\pi$  periodic, which encourages normalizing the data values to the range  $[0, \pi) \subset \mathbb{R}$  whenever possible. This prevents encoding two different values as the same quantum state and ensures therefore a distinction between data values after normalization.

### 5.3 Processing data

Once the data is embedded into a quantum state, it is time to transform and process the quantum information. As of now, the current circuit reciprocates the input value, that is, it's the identity function  $f(x) = x$ . Purposefully including and adding gates (either with free parameters  $\theta$  or constant parameters) into the quantum circuit, the complexity of the function may increase adequately.

As a mean to illustrate the growing complexity of the quantum circuit as more gates are added into it, consider taking a one dimension data point  $x_0$ . The data point  $x_0$  is run through a rotation operator with the free parameter  $\theta_0$ . The additional gate has changed the quantum circuit and the quantum transformation is defined as

$$f(x_0; \theta_0) = x_1 \cos(\theta_0) \tag{5.13}$$

The endless possibilities to apply different operations in quantum circuits are a valuable resource. With an embedding process and standard elementary operations, complex functions can be evaluated with the capabilities quantum circuits provide. The key idea behind these operations, beyond the simple function evaluated above that requires no quantum device actually, is that building deeper quantum circuits and steadily increasing the number of qubits yields a progressively more expensive and difficult function to evaluate through classical means. For which quantum systems are more suitable.

### 5.4 Measurement

Once the data has been handled and processed in the quantum circuit, there needs to be a measurement performed on the final quantum state of the system or subsystem that effectively describes a behaviour of the quantum system in classical means. Depending on the problem, qubits, entanglement and choice of operator, a measurement can be performed upon the quantum system. Two measurement procedures will be introduced, both of them possess high scalability and provide great flexibility for interpretation.

### 5.4.1 Measuring an observable in a quantum state

Observables are physical quantities that can be measured in a quantum state and provide a real value. Common observables include position, momentum, angular momentum or time evolution of systems. In quantum physics, an observable appears as a linear operator on a Hilbert space and represents the quantum state space of quantum states. The dynamical variable represented by the observable can be measured having its own eigenvalues which are real numbers (incredibly relevant for measurement). Therefore, an observable in quantum mechanics provides a real number as an outcome when measuring a particular quantum system, which corresponds to the eigenvalue of the operator in regards to the quantum system's measured quantum state.

In order to measure an observable in a quantum state, the expectation value of the operator corresponding to the observable is measured in the final quantum state. There are many possibilities on which operators could be used, which can be any Hermitian operator in a Hilbert space. For instance, Pauli matrices or the density operator can be measured in the final quantum state for classification problems, since the eigenvalues of the Pauli matrices are  $\{+1, -1\}$  and the eigenvalues of the density operator in basis states are  $\{0, +1\}$ . For the particular calculation and a more in depth explanation, a section is included in 6.5.

### 5.4.2 Probability vector

A different approach is using a probability density vector extracting the probability amplitudes of the final quantum state by calculating the square modulus of the probability amplitudes. In particular, given a computational basis  $\{|c_i\rangle\}_{i=1}^{2^n}$  of the n-qubit quantum system's quantum state that has to be measured, the measurement of the quantum state  $|\psi\rangle$  provides the probability of measuring the computation basis state  $|c_i\rangle$  in the current quantum state  $|\psi\rangle$ .

Alternatively, it can be defined as calculating the density operator of each of the computational basis states  $|c_i\rangle\langle c_i|$  and obtaining the expectation value of each of the density operators  $\langle\psi|c_i\rangle\langle c_i|\psi\rangle$  describing one of the computational basis states. Therefore, in a quantum state  $|\psi\rangle$ , the probability vector would be a vector of expectation values of the density operators in that precise quantum state.

$$\{\langle\psi|c_1\rangle\langle c_1|\psi\rangle, \langle\psi|c_2\rangle\langle c_2|\psi\rangle, \dots, \langle\psi|c_{2^n}\rangle\langle c_{2^n}|\psi\rangle\} \quad (5.14)$$

Therefore, the probability vector is actually the set of expectation values of the density operators representing computational basis states in a given quantum state.



## Quantum operations

Qubits are represented in quantum state-vector notation when the knowledge about the quantum state is absolute. This means that the quantum state is defined without uncertainty and the probability of the quantum state is 1. These group of quantum states are known as pure states, since they can be expressed as a linear combination of basis states, where each one has an accompanying probability amplitude associated.

Nevertheless, certain occurrences and practical scenarios impede representing the quantum state of the n-qubit system through a linear combination of basis states. Instead, each of the qubits has many possible quantum states describing its state as a whole, that is, the quantum state is represented as a statistical ensemble of multiple pure quantum state and each one of them occurs with a certain probability.

Generally, uncertainty of a quantum system is associated to the quantum decoherence or natural interference of the ongoing environment, however initial uncertainty about the quantum system can also lead to this situation. In formal terms, a mixed quantum state is represented as a probabilistic ensemble of a number of pure states. In order to tackle having multiple pure quantum states representing a single mixed state, which can't be represented with the quantum state-vector notation used thus far, mixed states are described using a different method called the density matrix or density operator.

### 6.1 Superposition and entanglement

Quantum superposition is one of the principles of quantum mechanics, which states that, in a similar fashion to waves, two or more quantum states can be added together. Therefore, the resulting valid quantum state will be a superposition of the previous quantum states. One of the most common representation of quantum states is in terms of the superposition of the computational basis states. For instance, any pure quantum state can be expressed as a linear combination of a basis.

A qubit is represented as a quantum superposition of the basis states  $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  and  $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ . The measurement of the qubit will collapse the quantum state into one of the possible outcomes, either 0 or 1. A pure one qubit quantum state  $|\psi\rangle$  is therefore represented as a linear combination of basis states with probability amplitudes  $\alpha_0$  and  $\alpha_1$  as follows

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad (6.1)$$

where  $|\alpha_0|^2 + |\alpha_1|^2 = 1$ . Probability amplitudes describe the behaviour of a quantum system in this case and the squared magnitude of the probability amplitude is the value of a classical probability density function for that specific behaviour. In the superposition, the probability of measuring state  $|0\rangle$  is  $|\alpha_0|^2$  and the probability of measuring state  $|1\rangle$  is  $|\alpha_1|^2$ .

Another of the most prominent physical phenomena of quantum systems is entanglement. A number of particles are said to be entangled as a result of spatial closeness or interaction among other possibilities when the quantum state of each of the particles can't be described independent from the other particles.

The qubit  $q_0$  is considered to be entangled with the qubit  $q_1$  if measuring the quantum state of  $q_0$  gives some information about the quantum state  $q_1$ . Two qubits are said to be in maximal entanglement if measuring the quantum state of one qubit completely describes the quantum state of the other qubit. In case only one of the qubits is measured no meaningful information could be measured about the individual systems, only getting randomly state 0 or 1 as a result. The group of two qubit maximally entangled quantum states are called Bell states shown below

$$\begin{aligned} |\Phi^+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ |\Phi^-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\ |\Psi^+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ |\Psi^-\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \quad (6.2)$$

Given that the correlation between the quantum states is absolute, the quantum states described above are said to be maximally entangled. If one of the qubits is measured the state corresponding to the other qubit is determined with complete certainty. A qubit is known to have two complex probability amplitudes, therefore a quantum state of  $n$  qubits has a total of  $2^n$  complex probability amplitudes. Consider the following two qubit quantum system for which the quantum state  $|\psi\rangle$  of the

two qubits has to be described in a normalized quantum state vector notation as a linear superposition of the computational basis states.

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \quad (6.3)$$

$$|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$$

In order to express the quantum state of two qubits, the Kronecker product  $\otimes$  is used, which is a generalization of the outer product from vectors to matrices. Given two one qubit quantum states  $|\psi_1\rangle$  and  $|\psi_2\rangle$ , the Kronecker product  $|\psi_1\psi_2\rangle$  is calculated as follows

$$|\psi_1\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \quad |\psi_2\rangle = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad (6.4)$$

$$|\psi_1\psi_2\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = \begin{bmatrix} \alpha_0 & \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \\ \alpha_1 & \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix}$$

A quantum system is said not to be entangled if the quantum state representing the two qubit system or subsystem presents the following equality

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\psi_2\rangle \quad (6.5)$$

for any arbitrary pair of single qubit quantum states  $|\psi_1\rangle$  and  $|\psi_2\rangle$ . In order to prove if a pure quantum state is entangled or not is via the reduced density matrix 6.4  $\rho$  on one of the qubits and tracing out the other. The quantum state is separable and it is possible therefore to write it in the above form as a Kronecker product if the reduced density matrix  $\rho$  has rank 1. If not, the quantum system is entangled to some degree. In order to do so, the rank condition is tested evaluating  $Tr(\rho^2)$  and checking if the value is 1 (not entangled) or otherwise conclude that the quantum system is entangled.

$$\begin{aligned} \rho_1 &= Tr_2(|\psi\rangle\langle\psi|) = Tr_2(|\psi_1\psi_2\rangle\langle\psi_1\psi_2|) \\ &= Tr_2(|\psi_2\rangle\langle\psi_2|)|\psi_1\rangle\langle\psi_1| = \langle\psi_2|\psi_2\rangle|\psi_1\rangle\langle\psi_1| = |\psi_1\rangle\langle\psi_1| \end{aligned} \quad (6.6)$$

$$Tr(\rho_1^2) = Tr(|\psi_1\rangle\langle\psi_1|\psi_1\rangle\langle\psi_1|) = Tr(|\psi_1\rangle\langle\psi_1|) = \alpha \quad (6.7)$$

If  $\alpha = 1$  the quantum system is not entangled, otherwise the quantum system will be entangled.

## 6.2 Pure states

Pure states are completely defined quantum states at any given point in time. As mentioned, they can be expressed in quantum state-vector notation as a linear combination of the basis states. Consider the single qubit system  $|\psi\rangle$  initialized in the state  $|0\rangle$ . Applying a Hadamard gate to the quantum state  $|\psi\rangle$  the following quantum state is obtained:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |+\rangle \quad (6.8)$$

The measurement of this state would be probabilistic, where state  $|0\rangle$  is measured with 50% probability and state  $|1\rangle$  is measured with 50% probability. The key idea however is not around the measurement itself but around the certainty of the quantum information at our disposal. It's absolutely certain that, without interference of any kind, in an ideal case with exact initialization and no interference in the Hadamard gate the resulting quantum state will always be the quantum state  $|+\rangle$ . Given that no uncertainty exist whatsoever about the quantum state itself, the quantum state  $|\psi\rangle$  is a pure state.

Following the quantum state-vector notation, a n-qubit system's quantum state  $|\psi\rangle$  is described as

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n-1} \end{bmatrix} \quad (6.9)$$

A crucial part is to consider the exponential expansion of the quantum state vector in terms of the amount of qubits of the said system. Since there are n qubits in the quantum system, the total amount of possible outcomes for the quantum system when measured is  $2^n$ . For instance, if there are  $n = 3$  qubits, the total amount of possible outcomes would be  $2^n = 8$ . A different approach to describe the quantum state in the form of a matrix is the density matrix representation. For pure quantum states, the density operator is calculated as

$$\rho = |\psi\rangle\langle\psi| \quad (6.10)$$



In the above calculation, the outer product  $|\psi\rangle\langle\psi|$  is obtained as follows

$$\begin{aligned}
 \rho &= |\psi\rangle\langle\psi| \\
 &= \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n-1} \end{bmatrix} [\alpha_0 \ \alpha_1 \ \alpha_2 \ \dots \ \alpha_{2^n-1}] \\
 &= \begin{bmatrix} |\alpha_0|^2 & \alpha_0\alpha_1 & \alpha_0\alpha_2 & \dots & \alpha_0\alpha_{2^n-1} \\ \alpha_1\alpha_0 & |\alpha_1|^2 & \alpha_1\alpha_2 & \dots & \alpha_1\alpha_{2^n-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \alpha_{2^n-1}\alpha_0 & \alpha_{2^n-1}\alpha_1 & \alpha_{2^n-1}\alpha_2 & \dots & |\alpha_{2^n-1}|^2 \end{bmatrix}
 \end{aligned} \tag{6.11}$$

As a mean to illustrate the density operator calculation, consider the following two qubit system at the maximal entanglement Bell state  $|\psi_1\rangle$ .

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{6.12}$$

The density operator of the Bell state is computed as

$$\begin{aligned}
 \rho_1 &= |\psi_1\rangle\langle\psi_1| \\
 &= \left( \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) \left( \frac{1}{\sqrt{2}} [1 \ 0 \ 0 \ 1] \right) \\
 &= \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{6.13}$$

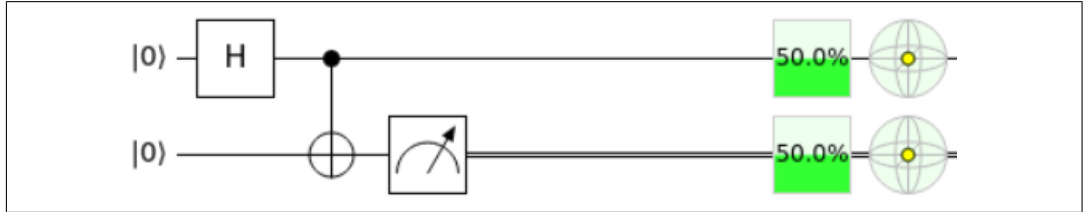
Up to this point, only a different representation for pure quantum states has been provided. However, since they can already be represented in quantum state-vector notation, there seems to be no benefit in using the density matrix representation of a pure quantum state. The prevalence and importance of density matrices is the focus of mixed states.

### 6.3 Mixed states

Mixed quantum states are quantum states represented by statistical ensembles of multiple different pure quantum states. The direct consequence of this fact is that mixed states have no possible representation in terms of linear superpositions of normalized pure quantum state vectors. For a better understanding, consider the previous quantum state  $|\psi_{q_0q_1}\rangle$  of a two qubit quantum system with qubits  $q_0$  and  $q_1$  respectively.

$$|\psi_{q_0q_1}\rangle = \frac{1}{\sqrt{2}} (|0_{q_0}0_{q_1}\rangle + |1_{q_0}1_{q_1}\rangle) \quad (6.14)$$

The corresponding circuit that produces the quantum state  $|\psi_{q_0q_1}\rangle$  is shown in the quantum circuit 6.1. The first wire corresponds to the first qubit  $q_0$  and the second wire corresponds to the second qubit  $q_1$ . At the end of the quantum circuit the second qubit  $q_1$  is measured.



**Figure 6.1:** A quantum circuit with Hadamard gate and CNOT gate that produces the Bell state  $|\psi_{q_0q_1}\rangle$

The qubits  $q_0$  and  $q_1$  are entangled, therefore knowing the measurement of  $q_1$  immediately collapses the other qubit  $q_0$ . If the measurement  $q_1$  yields the value 0 then the quantum state  $q_0$  will be projected into the state  $|0_{q_0}\rangle$ . The purpose is to find a representation of the final quantum state  $\psi_{q_0}$  independent and representative for any arbitrary measurement outcome of  $q_1$ . After performing a measurement on  $q_1$ , the quantum state  $\psi_{q_0}$  will be either in state  $|0_{q_0}\rangle$  with 50% probability and in state  $|1_{q_0}\rangle$  with 50% probability, however not in a superposition of  $\frac{1}{\sqrt{2}}(|0_{q_0}\rangle + |1_{q_0}\rangle)$  in any case.

Therefore, the quantum state  $|\psi_{q_0}\rangle$  has to be represented as a mixed state:

$$|\psi_{q_0}\rangle = \{|0_{q_0}\rangle, |1_{q_0}\rangle\} \quad (6.15)$$

Each one of the possible quantum states for  $|\psi_{q_0}\rangle$  has the following associated probability:

$$p_{q_0} = \{p_0, p_1\} = \{0.5, 0.5\} \quad (6.16)$$

With this convenient representation, it is clear that after measuring  $q_1$  in whichever case it falls, the quantum state  $\psi_{q_0}$  will be either in the state  $|0_{q_0}\rangle$

or in the state  $|1_{q_0}\rangle$  with equal classical probability. The ket notation of  $\psi_{q_0}$  is omitted since it can't be represented in quantum state vector notation using the computational basis in any case.

For a more general definition of a mixed state, a mixed state  $\psi$  is a quantum state which is an statistical ensemble of  $n$  pure quantum states:

$$\{|\psi_i\rangle\}_{i=1}^n = \{|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle, \dots, |\psi_n\rangle\} \quad (6.17)$$

Each of the pure quantum states  $|\psi_i\rangle$  has a corresponding probability of being collapsed into it:

$$\{p_i\}_{i=1}^n = \{p_1, p_2, p_3, \dots, p_n\} \quad (6.18)$$

The probability of occurrence  $p_i$  is the classical probability of the quantum system in quantum state  $\psi_{q_i}$  being projected onto the quantum state  $|\psi_i\rangle$ . The number of possible quantum states in the statistical ensemble  $n$  is not limited by the dimension of the underlying Hilbert space.

Despite the given representation being valid, it is by no means practical when  $n$  grows large and many quantum gates are applied, since keeping track of every possible outcome becomes too complicated. In order to capture the evolution of the quantum system both reliably and with accurate representation, the density matrix representation is utilized.

Previously, the density matrix for a pure quantum state was introduced. By all means, a pure quantum state refers to only one quantum state so it is comparable to the quantum mixed state case when there is only one pure state ( $n = 1$ ). The probability of obtaining that pure state is always 1, since there is no competition. The general density matrix definition for quantum states, both pure or mixed, goes as follows:

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| \quad (6.19)$$

For the presented case of the quantum state  $\psi_{q_0}$ , there are two quantum states in the statistical ensemble  $\{|0_{q_0}\rangle, |1_{q_0}\rangle\}$  with their respective probability  $\{0.5, 0.5\}$ .

The density operator representation of the quantum state  $\psi_{q_0}$  is built:

$$\begin{aligned}
\rho_{q_0} &= \frac{1}{2}|0_{q_0}\rangle\langle 0_{q_0}| + \frac{1}{2}|1_{q_0}\rangle\langle 1_{q_0}| \\
&= \frac{1}{2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{6.20}$$

An important notion to grasp is that the statistical ensemble of a mixed state can have any quantum pure state, it is not exclusive to the basis states (like the ones used in this showcase  $|0\rangle$  and  $|1\rangle$ ).

## 6.4 Reduced density matrix

One of the main advantages of the density matrix description of quantum systems comes into play when dealing with composite systems. With the reduced density matrix, it's possible to extract the state of each of the quantum subsystems including when they are entangled. Given a quantum system composed of the quantum subsystems  $Q_0$  and  $Q_1$ , for which the density operator  $\rho_{Q_0Q_1}$  describes the quantum system entirely, the reduced density matrix of the subsystem  $Q_0$  is calculated as

$$\rho_{Q_0} = Tr_{Q_1}(\rho_{Q_0Q_1}) \tag{6.21}$$

The Tr operation corresponds to the trace operation, which in case of two quantum states  $Tr(|\psi_1\rangle\langle\psi_2|) = \langle\psi_2|\psi_1\rangle$ .

The quantum state describing the composite quantum system entirely for Hilbert spaces  $Q_1$  and  $Q_2$  is given by

$$|\psi_{Q_0Q_1}\rangle = Q_1 \otimes Q_2 \tag{6.22}$$

For instance, the pure entangled quantum state of the system is described as

$$|\psi_{Q_0Q_1}\rangle = \frac{1}{\sqrt{2}}(|0_{Q_0}0_{Q_1}\rangle + |1_{Q_0}1_{Q_1}\rangle) \tag{6.23}$$

The entire quantum system is constructed with two single qubit systems. The first one qubit subsystem  $Q_0$  has basis vectors  $\{|0_{Q_0}\rangle, |1_{Q_0}\rangle\}$  and the second one qubit subsystem  $Q_1$  has basis vectors  $\{|0_{Q_1}\rangle, |1_{Q_1}\rangle\}$ . Through the reduced density matrix, the description of each of the quantum subsystems  $Q_0$  and  $Q_1$  can be

extracted. Therefore, the density matrix  $\rho_{Q_0Q_1}$  of the composite quantum system  $|\psi_{Q_0Q_1}\rangle$  is expressed using the outer products of the basis vectors as

$$\begin{aligned}
 \rho_{Q_0Q_1} &= |\psi_{Q_0Q_1}\rangle\langle\psi_{Q_0Q_1}| \\
 &= \frac{1}{2}[|0_{Q_0}0_{Q_1}\rangle\langle 0_{Q_0}0_{Q_1}| \\
 &\quad + |0_{Q_0}0_{Q_1}\rangle\langle 1_{Q_0}1_{Q_1}| \\
 &\quad + |1_{Q_0}1_{Q_1}\rangle\langle 0_{Q_0}0_{Q_1}| \\
 &\quad + |1_{Q_0}1_{Q_1}\rangle\langle 1_{Q_0}1_{Q_1}|]
 \end{aligned} \tag{6.24}$$

The reduced density matrix for the subsystem  $Q_1$  would be computed as

$$\begin{aligned}
 \rho_{Q_1} &= Tr_{Q_0}(\rho_{Q_0Q_1}) \\
 &= \frac{1}{2}[Tr_{Q_0}(|0_{Q_0}0_{Q_1}\rangle\langle 0_{Q_0}0_{Q_1}|) + Tr_{Q_0}(|0_{Q_0}0_{Q_1}\rangle\langle 1_{Q_0}1_{Q_1}|) \\
 &\quad + Tr_{Q_0}(|1_{Q_0}1_{Q_1}\rangle\langle 0_{Q_0}0_{Q_1}|) + Tr_{Q_0}(|1_{Q_0}1_{Q_1}\rangle\langle 1_{Q_0}1_{Q_1}|)] \\
 &= \frac{1}{2}[Tr(|0_{Q_0}\rangle\langle 0_{Q_0}|)|0_{Q_1}\rangle\langle 0_{Q_1}| + Tr(|0_{Q_0}\rangle\langle 1_{Q_0}|)|0_{Q_1}\rangle\langle 1_{Q_1}| \\
 &\quad + Tr(|1_{Q_0}\rangle\langle 0_{Q_0}|)|1_{Q_1}\rangle\langle 0_{Q_1}| + Tr(|1_{Q_0}\rangle\langle 1_{Q_0}|)|1_{Q_1}\rangle\langle 1_{Q_1}|] \\
 &= \frac{1}{2}[\langle 0_{Q_0}|0_{Q_0}\rangle|0_{Q_1}\rangle\langle 0_{Q_1}| + \langle 1_{Q_0}|0_{Q_0}\rangle|0_{Q_1}\rangle\langle 1_{Q_1}| \\
 &\quad + \langle 0_{Q_0}|1_{Q_0}\rangle|1_{Q_1}\rangle\langle 0_{Q_1}| + \langle 1_{Q_0}|1_{Q_0}\rangle|1_{Q_1}\rangle\langle 1_{Q_1}|] \\
 &= \frac{1}{2}[|0_{Q_1}\rangle\langle 0_{Q_1}| + |1_{Q_1}\rangle\langle 1_{Q_1}|] \\
 &= \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
 \end{aligned} \tag{6.25}$$

The interpretation to the procedure of obtaining the reduced matrix is that the density matrix  $\rho_{Q_1}$  describes the statistical outcomes of the subsystem  $Q_1$  when the measurements results for the subsystem A are taken in average. The concept of tracing out a quantum subsystem from the quantum systems is the formal definition of this technique. Interestingly, the density matrix  $\rho_{Q_0Q_1}$  represents a pure quantum state, while the reduced density matrix  $\rho_{Q_1}$  for the quantum subsystem  $Q_1$  describes the mixed state. This is a precise reason why choosing the density matrix representation is a much more practical and robust representation, specially when dealing with mixed quantum states.

## 6.5 Expectation value of an observable

The expectation value of an observable refers to the average of all the possible outcomes from a measurement of a quantum system weighted by their probabilities. An important aspect is that it does not indicate the most probable outcome by any means, instead outcomes are evaluated according to the probabilities they have to occur and an average of that is produced as the expectation value of the observable.

Consider a Hermitian operator  $\hat{Q}$  for which the states  $\{|q_i\rangle\}_{i=1}^n$  constitute a complete set of eigenstates in regards to an observable  $Q$  with non-degenerate discrete eigenvalues  $q_i$  ( $\hat{Q}|q_i\rangle = q_i|q_i\rangle$ ). The expectation value of the observable  $Q$  in a generic quantum state  $|\psi\rangle$  can be expressed by expanding  $|\psi\rangle$  as a linear superposition of the eigenstates and eigenvalues of the operator  $\hat{Q}$ .

The quantum state  $|\psi\rangle$  is expanded as linear superposition of the operator  $\hat{Q}$  with discrete eigenvalues as

$$|\psi\rangle = \sum_{i=1}^n |q_i\rangle \langle q_i|\psi\rangle = \sum_{i=1}^n c_i |q_i\rangle \quad (6.26)$$

where  $c_i = \langle q_i|\psi\rangle$ . The previous  $c_i = \langle q_i|\psi\rangle$  corresponds to the projection of the quantum state  $|\psi\rangle$  along the eigenstate  $|q_i\rangle$  of the operator  $\hat{Q}$  with eigenvalue  $q_i$ . The probability of a measurement having outcome  $q_i$  for the physical quantity  $Q$  corresponding to the operator  $\hat{Q}$  is given by  $|c_i|^2 = |\langle q_i|\psi\rangle|^2$ . With these expansion and relationships established, the expectation value of the operator  $\hat{Q}$  in the generic state  $|\psi\rangle$  is defined as

$$\begin{aligned} \langle \hat{Q} \rangle_\psi &= \langle \psi|\hat{Q}|\psi\rangle = \langle \psi|\hat{Q} \sum_{i=1}^n c_i |q_i\rangle = \sum_{i=1}^n c_i \langle \psi|\hat{Q}|q_i\rangle \\ &= \sum_{i=1}^n q_i c_i \langle \psi|q_i\rangle = \sum_{i=1}^n q_i c_i c_i^* = \sum_{i=1}^n q_i |c_i|^2 \end{aligned} \quad (6.27)$$

A different approach can be taken to resolve the expectation value of the observable  $Q$  by introducing the identity operator  $\hat{I}$  using the complete set of eigenstates of the operator  $\hat{Q}$  as terms ( $\hat{I} = \sum_{i=1}^n |q_i\rangle \langle q_i|$ ) on the expectation value calculation.

$$\begin{aligned} \langle \hat{Q} \rangle_\psi &= \langle \psi|\hat{Q}|\psi\rangle = \langle \psi|\hat{Q}\hat{I}|\psi\rangle \\ &= \langle \psi|\hat{Q} \sum_{i=1}^n |q_i\rangle \langle q_i||\psi\rangle = \langle \psi|\sum_{i=1}^n c_i \hat{Q}|q_i\rangle \\ &= \sum_{i=1}^n c_i q_i \langle \psi|q_i\rangle = \sum_{i=1}^n q_i c_i c_i^* = \sum_{i=1}^n q_i |c_i|^2 \end{aligned} \quad (6.28)$$

For continuous eigenvalues  $q$ , given that the states  $|q\rangle$  are a complete set of eigenstates of the operator  $\hat{Q}$  and the identity operator  $\hat{I}$  in terms of the eigenstates of the operator  $\hat{Q}$  is  $\hat{I} = \int_{-\infty}^{+\infty} |q\rangle\langle q|dq$ . If the eigenvalue spectrum of  $\hat{Q}$  is discrete, the expectation value of the observable  $Q$  in the quantum state  $|\psi\rangle$  can be computed in terms of eigenstates  $|q\rangle$  and eigenvalues  $q$  as

$$\langle\hat{Q}\rangle_{\psi} = \int_{-\infty}^{+\infty} q|\langle q|\psi\rangle|^2dq \quad (6.29)$$

### 6.5.1 Pauli matrices

In the field of quantum computing, the Pauli matrices correspond to three complex unitary Hermitian matrices of dimension  $2 \times 2$ . They are conventionally named as  $(\sigma_1, \sigma_2, \sigma_3)$  or  $(\sigma_x, \sigma_y, \sigma_z)$  respectively which denotes in which of the axis in the Bloch sphere each of the Pauli matrices rotates by  $180^\circ$ . These are the matrix representation of the Pauli matrices.

$$\begin{aligned} \sigma_1 = \sigma_x &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \sigma_2 = \sigma_y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ \sigma_3 = \sigma_z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned} \quad (6.30)$$

Apart from the Bloch sphere interpretation and matrix representation, another aspect of interest is the expectation value of Pauli matrices as operators  $\hat{H}_1$ ,  $\hat{H}_2$  and  $\hat{H}_3$  in a given quantum state  $|\psi\rangle$ . To do so, firstly it is necessary to calculate the eigenvalues of the Pauli matrices, which can be observed from the value of the determinant and trace.

$$\begin{aligned} \det(\sigma_i) &= -1 \\ \text{Tr}(\sigma_i) &= 0 \end{aligned} \quad (6.31)$$

The eigenvalues for each of the matrices  $\sigma_i$  are eigenvalues  $+1$  and  $-1$ . The

corresponding normalized eigenvectors of the matrices are the following

$$\begin{aligned} |q_{x+}\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ |q_{x-}\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ |q_{y+}\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} \\ |q_{y-}\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} \\ |q_{z+}\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ |q_{z-}\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{6.32}$$

Consider the matrix  $\sigma_z$  as the operator with eigenvalues  $q_z \in \{+1, -1\}$  from which the expected value  $\langle \sigma_z \rangle_\psi$  is desired to be measured in a generic one qubit quantum state  $|\psi\rangle$ . Consider the probability amplitudes of the quantum state  $|\psi\rangle$  to be  $\alpha_0 = \frac{1}{\sqrt{2}}$  and  $\alpha_1 = \frac{1}{\sqrt{2}}$ . The quantum state  $|\psi\rangle$  can be written as a linear superposition of the eigenstates  $\{q_{z+}, q_{z-}\}$  of the operator  $\sigma_z$  as follows

$$\begin{aligned} |\psi\rangle &= \sum_{i \in \{+, -\}} |q_{zi}\rangle \langle q_{zi} | \psi \rangle = \sum_{i \in \{+, -\}} c_i |q_{zi}\rangle \\ &= c_+ |q_{z+}\rangle + c_- |q_{z-}\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{6.33}$$

$$\begin{aligned} c_+ &= \langle q_{z+} | \psi \rangle = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \\ c_- &= \langle q_{z-} | \psi \rangle = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \end{aligned}$$

The expectation value of the operator  $\sigma_z$  in the generic state  $|\psi\rangle$  is obtained as



follows

$$\langle \sigma_z \rangle_\psi = \langle \psi | \sigma_z | \psi \rangle = \sum_{i \in \{+, -\}} q_{zi} |c_i|^2 = 1 \left( \frac{1}{\sqrt{2}} \right)^2 + (-1) \left( \frac{1}{\sqrt{2}} \right)^2 = 0 \quad (6.34)$$

The expected value  $\langle \sigma_z \rangle_\psi$  of the operator  $\sigma_z$  in the generic state  $|\psi\rangle$  has given the value 0. This might seem strange at first, but it is actually a perfectly possible value. The expected value  $\langle \sigma_z \rangle_\psi$  of the operator  $\sigma_z$  returns the average eigenvalue after measuring the operator in the quantum state  $|\psi\rangle$  weighted by their probability. Since the eigenvalues are  $q_z \in \{+1, -1\}$ , and the probabilities for each of the eigenvalues are  $\{|c_+|^2, |c_-|^2\} = \{\frac{1}{2}, \frac{1}{2}\}$ , it means that there is the same probability to either eigenvalue outcome when measuring the operator  $\sigma_z$  in the quantum state  $|\psi\rangle$ .

These matrices are particularly important in the classification task using quantum circuits, since they provide an expected value in the range  $[-1, 1]$  and therefore can be used as operators for binary classification problems. The idea behind using one of these matrices is assigning the expected value  $+1$  to one of the class labels and assign the expected value  $-1$  to the other one. In a Bloch sphere representation, it would be trying to improve the model parameters so that the eigenstate  $|q_i\rangle$  with eigenvalue  $q_i$  is as close in the Bloch sphere to the final quantum state being measured as possible. Mathematically, it would mean that the inner product of the eigenstate  $|q_i\rangle$  (with the corresponding desired eigenvalue  $q_i$  associated to the class label being predicted) with the final quantum state  $|\psi\rangle$  of the system being as close as possible to 1.



# Experiment and design

With the purpose of assessing the performance of quantum systems by themselves or together with other classical components, some workflows have been carried out where models with selected designs and thoroughly specified properties. For a better organization of how each workflow works, each of the stages are separated to explain the steps involved and establish connections with the already divulged information (specially in relation to quantum systems).

A general view of a workflow consists on the following scheme 7.1.

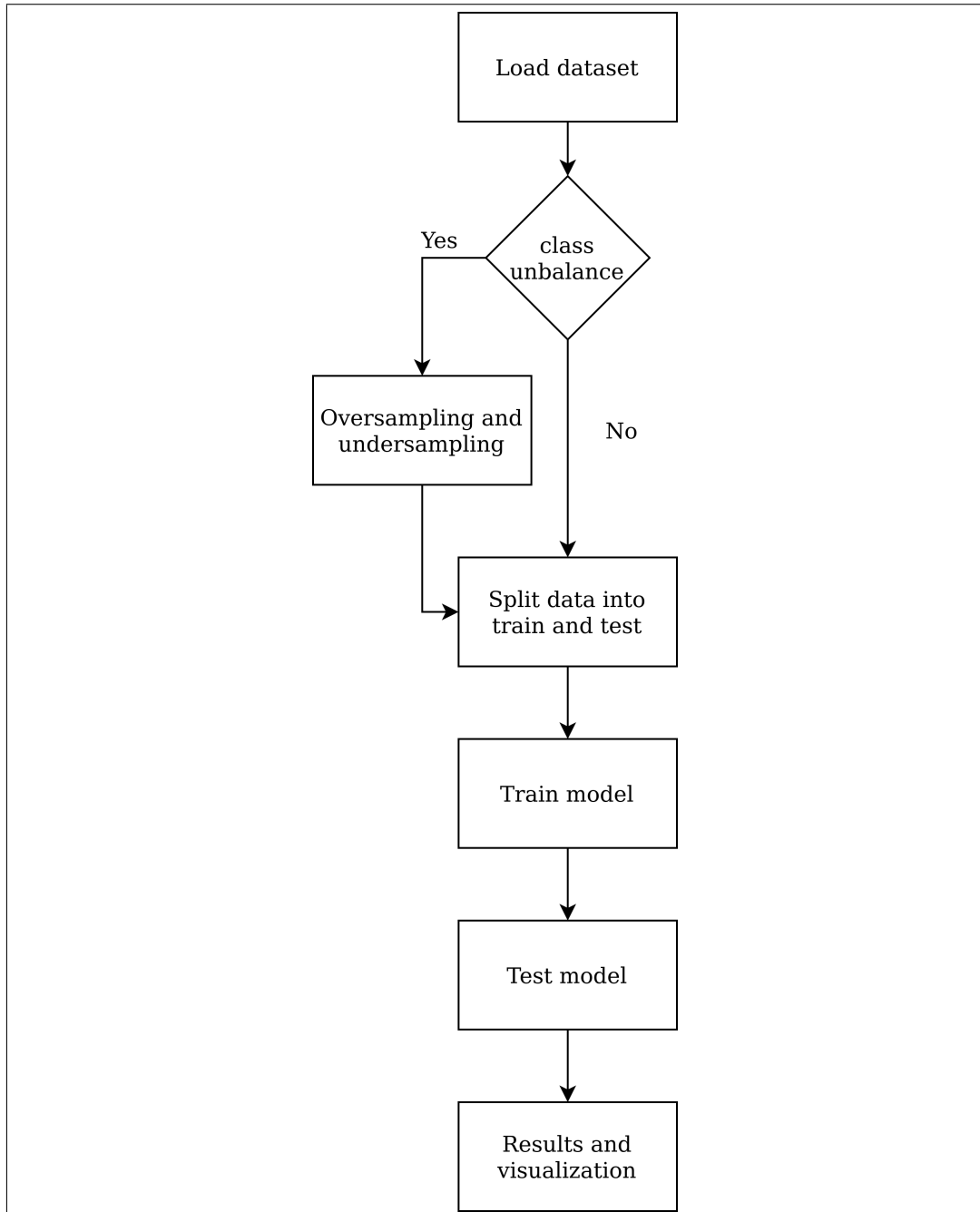
## 7.1 Environment and packages

All the experiments have been carried out using *Python 3.10* version and its powerful libraries for scientific research. These libraries include *numpy*, *sklearn* and *matplotlib* among others. For setting up model architectures and organizing training and testing grounds *tensorflow* and *keras* have been chosen primarily. Nonetheless, the quantum circuits used have been deployed through the *pennylane* library, which offers access to differentiable programming of quantum systems, giving access to training quantum circuits in the same way classical models are approached.

The code of the experiments is accessible from here<sup>1</sup>. The code is organized in a package for the models and other functions that help in dealing with the task. For carrying out experiments however, *Jupyter notebooks* are included to show the step by step procedure used for every model and technique employed. Moreover, in case the experiments are desired to be replicated, some hyperparameters allow this possibility. Among them, the seed for the initialization of weights, the learning rate of the optimizer or the choice to save weights from previous experiments are provided (along with the possibility to load weights from these experiments themselves).

---

<sup>1</sup>[https://github.com/Totx/VQC\\_Hybrid\\_classifiers](https://github.com/Totx/VQC_Hybrid_classifiers)



**Figure 7.1:** General workflow for different models and specifications

One thing to note, however, is that, despite claiming to be using a quantum circuit built in a quantum system, actually the experiments are run using *penny-lane* libraries quantum devices which, as such, are simulating the behaviour of a quantum system not running in quantum hardware whatsoever. Although not the focus of the experiments, the discussion between how close a quantum device simulator approaches quantum hardware is a concerning issue that currently given the resources available can't be salvaged in many cases unfortunately. Having concluded that, the quantum devices simulating quantum hardware still serve as a good reference when attempting different kinds of quantum operations before accessing quantum hardware directly, as they provide a good estimation of how quantum hardware would behave given the circumstances.

## 7.2 Data management

In order to determine how to manage the data, the nature of the data has to be described first. The dataset belongs to a credit card fraud dataset, where transactions are split into two possible cases. Namely, the labels of the dataset are valid cases (label 0) and fraud cases (label 1). The amount of features  $x_i$  of the dataset is 28 and every one of the features is within  $x_i \in [-1, 1]$ . There are some other information regarding the amount or time, however, these are not features themselves and, as such, they won't be accounted in the experiments in any manner.

With the presented nature of the data, the task of the machine learning algorithm is to perform a supervised classification of transactions and decide whether they are potentially fraudulent or not (binary classification). Moreover, the features themselves leave no room for interpretability and, therefore, are quite hard to handle. On that note, the main issue of the dataset is the disparity between the amount of transactions labeled as fraudulent in contrast with how much are labeled as valid though. In particular, there are barely 492 fraudulent cases in comparison with the enormous quantity of 284315 valid cases. However, in practice some of the valid cases could still contain potential fraudulent cases since there could be errors in the labelling process of the dataset.

The unbalanced data is a complicate issue in machine learning and to circumvent that two preprocessing steps have been considered. Nevertheless, with the intention of having a fair testing data collection that has not been produced externally and therefore could give false testimony to the actual performance the models could produce, the data has been divided into a training set and a testing set before any kind of preprocessing. All the preprocessing steps therefore, refer exclusively to the training set data, not test data at all.

The first step is an oversampling of the minority class (fraud cases) to have more cases from the minority class to learn from. The oversampling technique applied is *SMOTE* which synthetically reproduces through k-nearest neighbours algorithm new cases from the minority class. It is a particularly powerful technique that in many cases can reduce the possible overfitting caused by other oversampling

techniques such as random oversampling. However, since the newly created cases are still approximations from the already existing cases and the number of fraud cases is small, a compromise has been set to prevent overfitting issues by only creating up to around 10000 cases of this kind.

The second step involves the opposite operation. Given the existing unbalance in the training set, in order to balance the scales an undersampling technique is used to reduce cases from the majority class (valid cases). The undersampling algorithm used is *Near Miss*, which allows to remove cases from the majority class by checking the distance to the  $n$  minority class cases depending on the heuristic approach chosen and leaving those that closely follow the heuristic specification while leaving out the other cases until only the desired number of cases remain. In this case, the *Near Miss 1* undersampling technique has been used together with computing the average distance of the 3 closest point of the minority class as a result.

Therefore, the train set presents around 20000 cases, which has incorporated class balance after using the presented sampling techniques.

### 7.3 Machine learning models

There are mainly three routes that have been considered and built around in the model section. The first route corresponds to models which are centered around using almost exclusively quantum circuits with little necessary classical additions. The second alternative is building a machine learning model that combines both classical layers with quantum layers without restraints. Finally, there is the standard classical machine learning model relying only in classical approaches when building the machine learning model.

#### 7.3.1 Quantum models

The Quantum models are divided into three layers for specific purposes. As mentioned before, the quantum models haven been reduced to relying as much as possible only in quantum circuits, since the main interest in these models is to measure the performance without the aid of the classical counterpart. However, classical layers help in inputting the data and outputting the data as if it were communicating with the quantum layer to send information and receive it.

The first layer is common to all the quantum models, which is the input layer. Given that there are 28 variables, the input layer is merely receiving the features. The second layer is the quantum layer and the core and soul of the quantum models, for which two different quantum circuits with different data processing approaches have been proposed. And last, the output layer gathers the output from the quantum layer and outputs a single scalar.

Before entering into the specifics of each of the models, the output layer of the

model is of great importance. The information output received from the quantum layer into the dense layer is a collection of one per qubit measurements, where each of the measurements contains the expectation value of the Pauli operator  $\sigma_z$  in the quantum state described by each of the qubits. Therefore the key idea behind this design choice is to train the model so that the output scalar of the quantum layer is as close as possible to the eigenvalue  $q_i$  that represents the label it has been assigned to. The output layer behaves as any other dense layer, so in a sense it is helping to somewhat improve the result of the quantum model, for which reason the activation function has been set to linear.

The  $\sigma_z$  operator has two possible eigenvalues  $-1$  and  $+1$ , which have been assigned to the label of valid cases and the label of fraud cases. In the case of quantum layers, 5 qubits operated quantum devices are being used which means that the operator is measured in each one of the qubits' quantum state and then sent to the output layer. The reasoning behind using the expectation value is that it does not only measure the probability of how close is the quantum system from the desired quantum state (eigenstate of the operator for the desired label assigned to an eigenvalue) but also takes into account the probability of how close it is to other quantum states (eigenstates that don't correspond to the eigenvalue of the label). After all, the expectation value returns the average of all possible outcomes, so in that regard, the model parameters are trained to not only approach the desired eigenstate of the operator associated to an eigenvalue but also distance themselves from the other ones.

One of the shortcomings of using the weighted sum of the expectation values of an operator as the output of a quantum model is that it is a scalar output instead of the more classical approach of using a softmax layer to obtain the probabilities of a case belonging to each one of the classes (in multi class classification particularly). Therefore, the selected loss function of the quantum models is the mean squared error and the performance metric is the mean absolute error. The class labels of the dataset haven been converted from  $\{0, 1\}$  to  $\{-1, 1\}$ . Finally, the optimization algorithm used is *Adam* with custom learning rate of 0.01.

The mean squared error as a loss function might seem a bit out of place in the task of binary classification. The purpose however is to try to measure the performance of quantum circuits without adding nonlinearity a *sigmoid*-like function brings into the table, which would help the model gain complexity to a higher degree from outside the quantum circuit itself.

The quantum circuits are built around two operations primarily, that is to say, the rotation operations with the parameters or the data itself and the *CNOT* gates. The rotation gates are straightforward, they apply the rotation with the given three parameter values in each one of the axes. The *CNOT* gates are the main method of communication between qubits and enable entanglement of the control qubit (blue dot) with the target qubit (empty dot), so that the qubits are not separated from each other and actually have effect on each other.

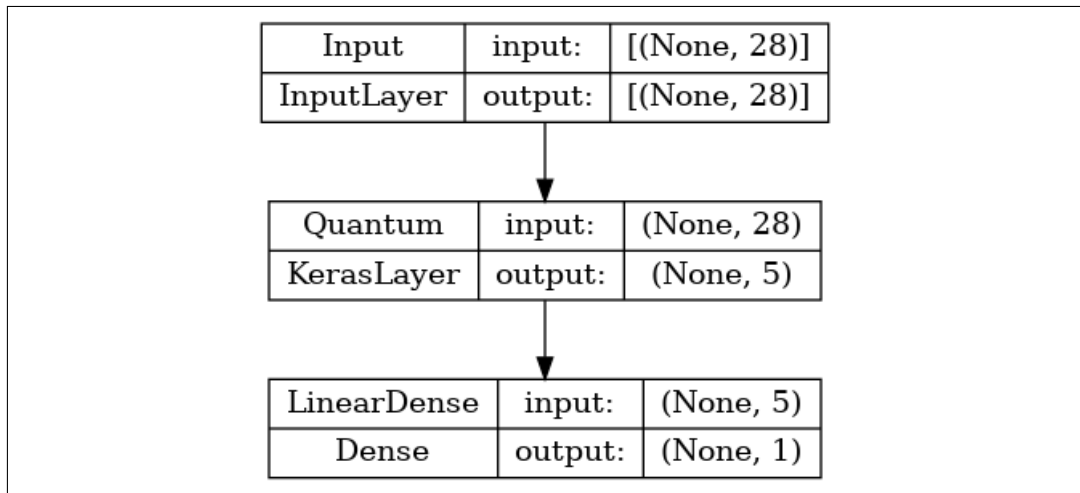


Figure 7.2: The quantum model diagram.

The quantum layers are separated depending on how the features are processed when entering the quantum layer, which has several implications of interest. The entire model diagram can be visualized in the figure 7.2.

### 7.3.1.1 Quantum models based on amplitude embedding

In order to introduce the data into the quantum circuit, the first option given the float type nature of the data is to transform it into the probability amplitudes of the quantum state of the entire quantum system. The features are normalized so that the probability amplitudes of the quantum state of the quantum system are equal to 1. The total number of  $n$  qubits required to use amplitude embedding on  $k$  features is  $\lceil \log_2(k) \rceil = n$  and  $2^n$  probability amplitudes can be embedded in consequence.

In this case, given that  $k = 28$  the amount of qubits required is  $\lceil \log_2(28) \rceil = 5$ , which corresponds to the amount being used. Apart from that, since  $2^5 = 32$  probability amplitudes are available, the remaining probability amplitudes are padded with 0, so that they don't affect in the normalization process of the amplitude embedding.

The quantum layer's underlying quantum circuit itself with the already trained weights after the optimization process is also included in the figure 7.3. The inspiration for this quantum circuit architecture with strongly entangling layers has been drawn from [37].

### 7.3.1.2 Quantum models based on data reuploading

Another way to bring about classical data into quantum circuits is using the data reuploading approach to it. Instead of introducing the data through an embedding process, features are included as arguments to quantum circuits' quantum gates. For this case, the rotation operations in quantum gates receive the input data in sets of three in each of the qubits and apply a rotation. Afterwards, another rotation occurs



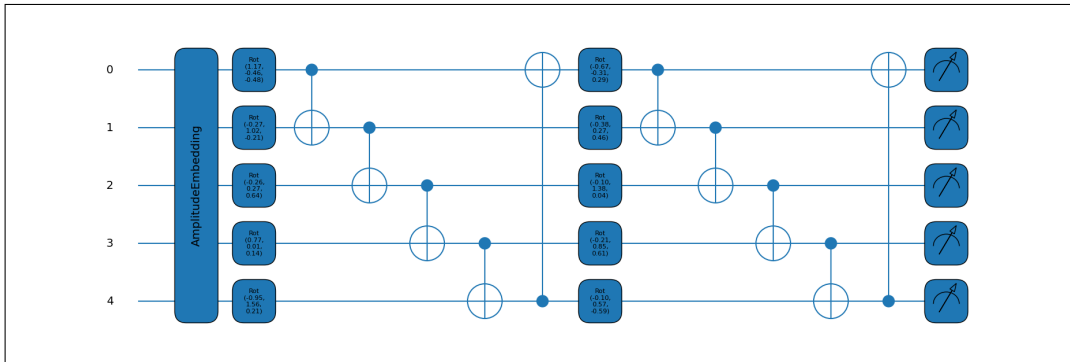


Figure 7.3: The quantum circuit diagram using amplitude embedding.

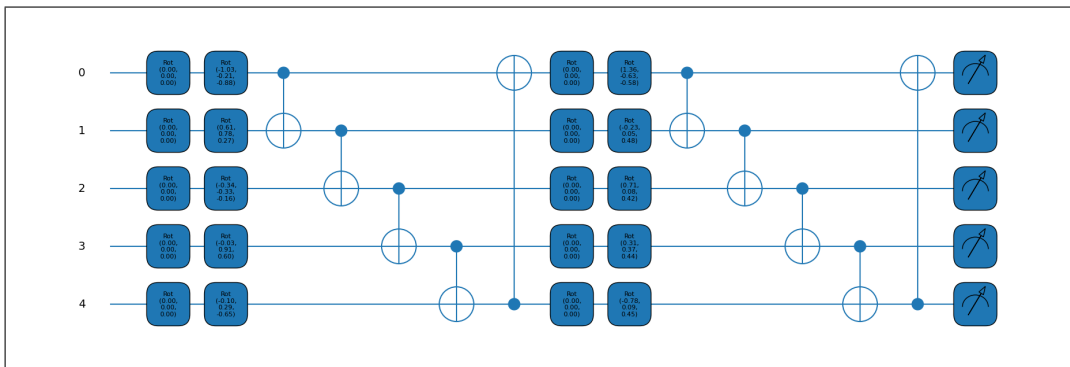


Figure 7.4: The quantum circuit diagram using data reuploading.

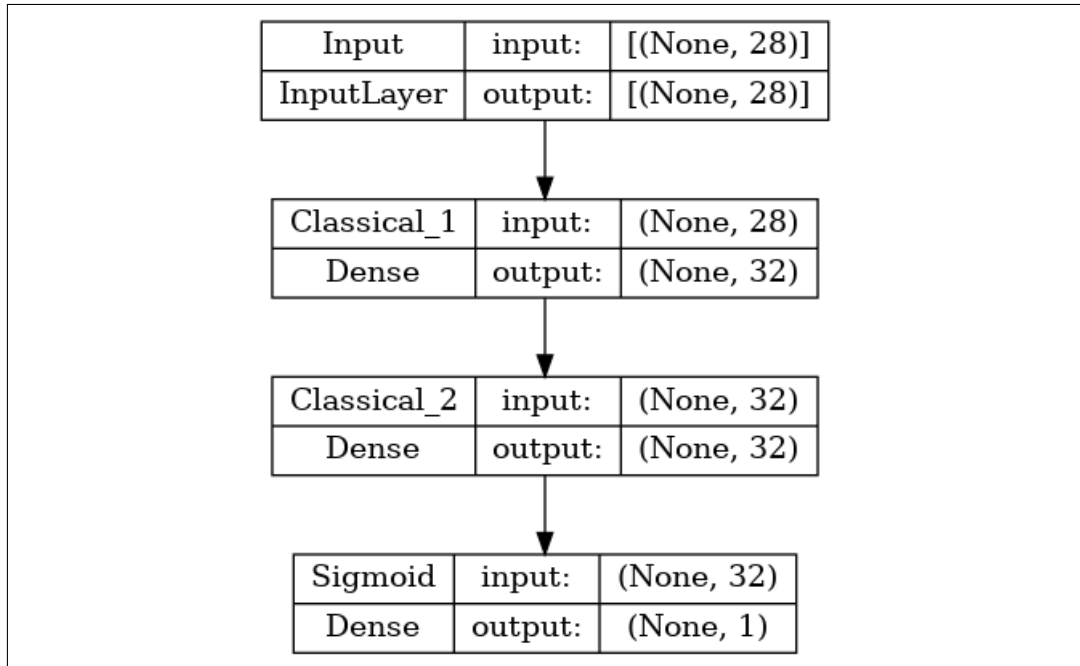
in each of the qubits corresponding to the weights rotations. While embedding separates quantum gates from using data directly, this encourages the opposite.

The amount of rotations per layer and qubit of the quantum circuit is tied to the number of features  $k$ . Since there are 3 parameters required for each of the rotations, the amount of values for rotations required in each qubit of each of the layers is  $k + (3 - (k \bmod 3))$ , which in this case corresponds to  $28 + (3 - (28 \bmod 3)) = 30$ . Therefore, two extra values are padded to compensate for the lack of the two missing parameters for the last rotation. The features are used as rotation parameters in each of the layers.

The quantum layer's underlying quantum circuit itself with the already trained weights after the optimization process is also included in the figure 7.4. Instead of including ten rotations per layer and qubit, for representation reasons in the given space, it is considered as if only three features were entering the quantum circuit with value 0 in each of them.

### 7.3.2 Classical model

The classical model of choice is akin to a multilayer perceptron in the realm of deep neural networks. While there are many algorithms which could fit in the description to be used in binary classification, dense layers presented in these types



**Figure 7.5:** Classical model with two dense layers and *sigmoid* activation function in the output layer.

of neural networks can be joined with quantum layers for binary classification. However, as a matter of fact, one of the best supervised learning machine learning algorithms known to handle the task of binary classification for the dataset is the random forest algorithm.

The distribution of layers in the classical model consists of four interconnected layers. The first layer corresponds to the input layer where 28 features are loaded in batches. The second and third layers are dense layers with 32 units each and *ReLU* activation function. The final layer is a 1 unit layer with the *sigmoid* activation function, a suitable choice for the given binary classification problem. Every classical dense layer's weights are initialized using random sampling.

The optimizer choice for the classical model remains being *Adam* with custom learning rate of 0.01, while the loss function is changed to binary cross entropy. The latter is a common choice for binary data classification and, in this case, the output from the last layer *sigmoid* activation function is used as the probability  $p_i$  for the transaction case  $x_i$  to be from the positive 1 class (fraud class), while  $1 - p_i$  corresponds to the probability for the data vector to be from the negative 0 class (valid class).

The entire model diagram is presented in figure 7.5.

### 7.3.3 Hybrid models

The hybrid models are combinations of the above presented exclusively classical or quantum approaches respectively. The idea is to combine in different settings the

nonlinearity present in activation functions of classical dense layers and quantum circuits' quantum operations and measurement processes. Two hybrid models have been considered in total.

As mentioned before, the classical layers present in the hybrid architecture have the same properties, that is, 32 units and *ReLU* activation function. The output layer is once again a 1 unit *sigmoid* activation function for binary classification. The quantum circuits follow the same idea of strongly entangled layers discussed, however the quantum circuits utilized follow the embedding techniques to incorporate classical data and process it in every case. Therefore, in quantum circuits the rotation and *CNOT* operations remain unchanged.

The optimizer keeps being *Adam* with custom learning rate 0.01 and the loss function remains being the binary cross entropy.

### 7.3.3.1 Sequential model

The first one of the models is a straightforward one. The hybrid model has a total of five layers. The first layer corresponds to the classic input layer already introduced for the 28 features, the second layer is the quantum circuit using the quantum circuit using amplitude embedding presented before and the third and fourth layers correspond to the classical layers with 32 units each and the *ReLU* activation function. The last layer is the output layer of 1 unit with *sigmoid* activation function.

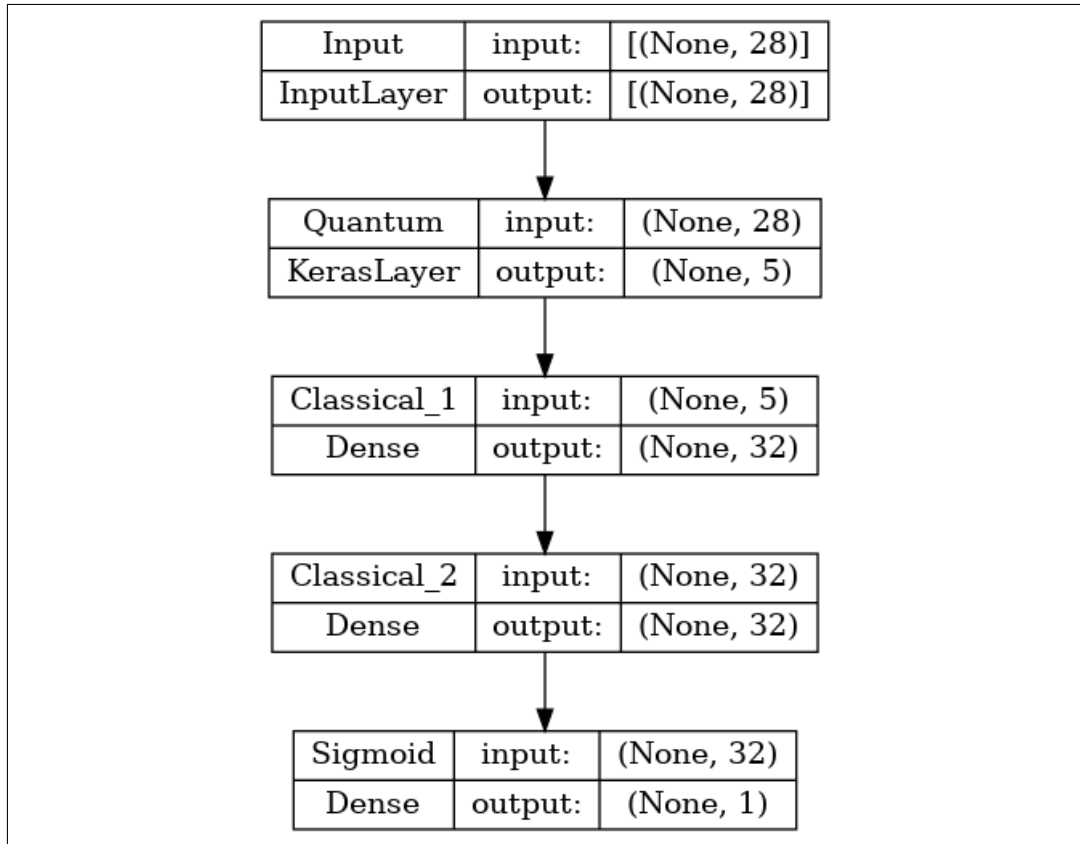
The amplitude embedding present in the quantum layer is a limiting factor when building differentiable models in the sense that features are not currently differentiable when using it. The direct implication is that it does not allow differentiable parameters before the quantum layer. In this case, since it is the first layer with differentiable parameters, it does not cause any problem despite being somewhat a restraint in the design of the model.

The entire model diagram can be found in figure 7.6, while the quantum circuit itself corresponds to the one already included 7.3.

### 7.3.3.2 Functional model

Despite sequential models so far being quite capable themselves, a functional model, that explores a more complex, enriching and hopefully better representation of the problem by splitting data and processing it separately to join them afterwards, has been devised for the binary classification task at hand. The input data is split into two equivalent portions and introduced into two analogous sequential layer collections.

A sequential layer collection contains specifically four layers, however, unlike with the sequential hybrid model a different approach is taken in the order of the classical layers and the quantum layer. The first two layers are the classical layers presented so far with 32 units and *ReLU* activation function. For the quantum layer



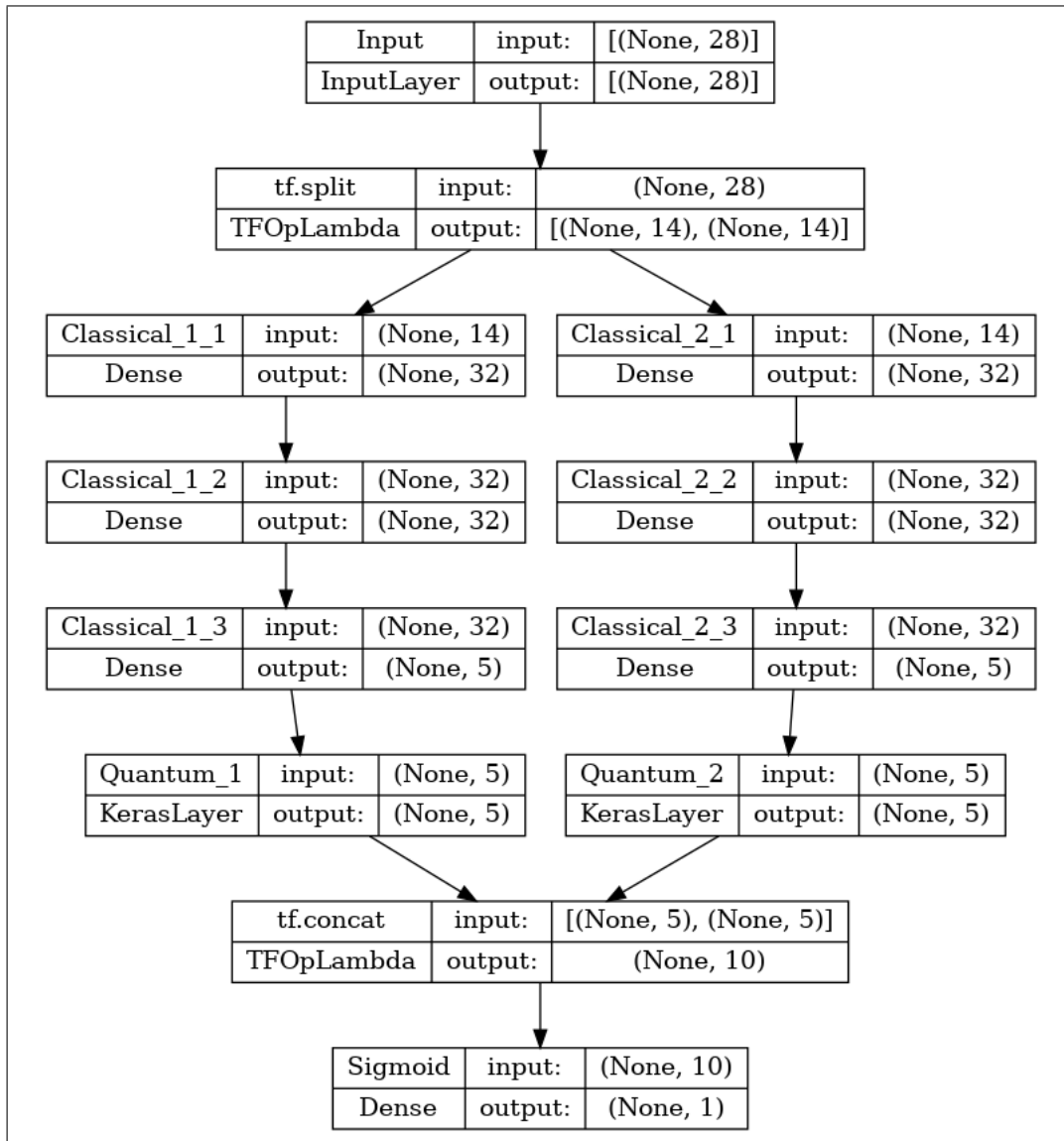
**Figure 7.6:** Sequential hybrid model using amplitude embedding.

part, it is still a strongly entangling layer composed of rotation operations and *CNOT* gates.

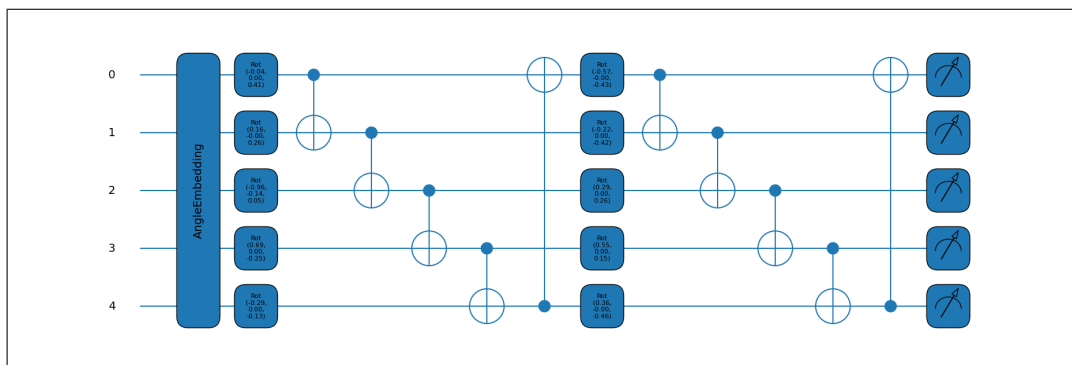
Nevertheless, the amplitude embedding proposed up to this point has been discarded given the problem presented of not allowing differentiable parameters before it. In change, the angle embedding approach is used to solvent this issue. The amount of qubits necessary for angle embedding ascends to one qubit per feature and, since the quantum circuits use quantum devices with 5 qubits, another classical layer is introduced in the sequential layer after the two classical layers and before the quantum layer.

The in between classical and quantum classical layer serves as an intermediary to gather the information into its 5 units from the 32 input values being received from the previous layer. Another aspect to consider is the *ReLU* activation choice, which is particularly helpful for the angle embedding given that it is recommended that all the values being embedded are  $x_i \in [0, \pi)$ .

The final part correspond to joining together the expectation values obtained for the operator in each of the sequential layers and apply the output layer's parameters and *sigmoid* activation function. The functional model diagram is show in 7.7, while the angle embedding quantum circuit is also included in 7.8.



**Figure 7.7:** Functional hybrid model with sequential layer collections and angle embedding.



**Figure 7.8:** Quantum strongly entangling layers using angle embedding.

## 7.4 Results

There are mainly two ways that have been proposed to envision the capacity of the models to classify transaction cases either into valid or fraud ones. The first one is through metrics of the entire test set, which includes a fourth part of the entire dataset. However, while metrics are valuable, the test set is heavily unbalanced since there are only 120 fraud cases in opposite of the staggering 71082 valid cases present. Given the circumstances and in order to ensure the validity of the results for generalization, the test set is processed as such for the calculation of the performance metrics.

The other approach, albeit not as faithful as the former one, is selecting a subset of the test set. The subset of the test set incorporates a balanced quantity of both valid cases and fraud cases in exchange of losing the more accurate representation of the problem's settings. Nevertheless, the purpose of this balanced test set is to visually represent the performance each of the models has in regards to cases particularly from each of the classes. These results are helping to understand the metric values of the standard approach and are not substitutes in any case. With all things considered, the number of class instances is even with 120 from each one which adds up to a total 240 cases for the subset of the test set.

### 7.4.1 Metrics' results

Given the binary classification nature of the problem as well as the unbalanced nature of the dataset, metrics such as accuracy that can't capture the intricacies of the present disparity in class distribution are not suitable. Instead, the chosen metrics are precision, recall and F1-score for each of the classes, which more accurately represent the actual performance of the different models. The results for each of the models are presented in the table 7.1 below.

	Fraud (1 label) cases			Valid (0 label) cases		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Qmodel with amplitude embedding	0.7	0.87	0.77	0.83	0.625	0.71
Qmodel with data reuploading	0.54	0.76	0.63	0.6	0.35	0.45
Hybrid sequential model	0.69	0.83	0.76	0.79	0.64	0.70
Hybrid functional model	0.71	0.93	0.81	0.90	0.63	0.74
Classical dense model	0.62	0.85	0.72	0.99	0.99	0.99

**Table 7.1:** Results of each and every model presented for the test set.

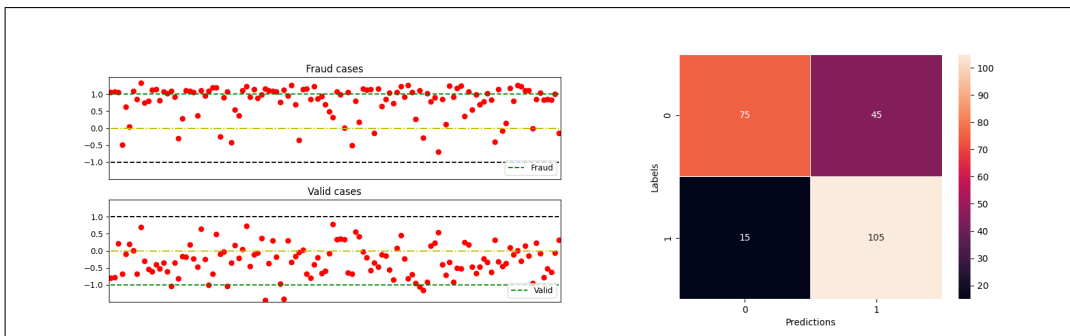
### 7.4.2 Balanced test subset results

While the metrics' results are much more reliable for the assessment of a general case, there are some graphical results that may aid in understanding how well the model classifies data vectors of different classes. Two different approaches have been taken to illustrate this, namely, a distance plot or classification plot of the predictions towards the corresponding label and a confusion matrix of the test

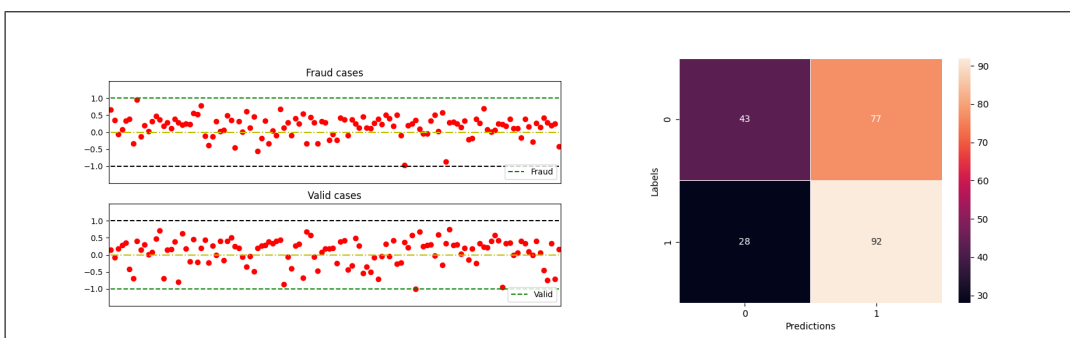
subset to have a glimpse of the meaning and obtain a better grasp of the metrics' interpretation.

The confusion matrix establishes the relation between the predicted class of the inputs with the actual class they belong to. For the plots, there are two distinctions depending on whether the plot corresponds to a quantum model or to a hybrid model or classical model. For quantum models, the output value of the model towards the labels  $+1$  and  $-1$  is considered, that is, the distance to the appropriate label and the opposite as a means to show which eigenvalue follows the closest each data vector. In case of the hybrid or classical plots, the classification plot represents what the data vector was classified as in comparison to what it should have been classified as.

The results for the subset of the test set are gathered in figures 7.9 for the quantum model with amplitude embedding, 7.10 for the quantum model with data reuploading, 7.11 for the hybrid sequential model, 7.12 for the hybrid functional model and, finally, 7.13 for the classical model.



**Figure 7.9:** Quantum model with amplitude embedding distance plot towards correct class and confusion matrix.



**Figure 7.10:** Quantum model with data reuploading distance plot towards correct class and confusion matrix.

## 7. EXPERIMENT AND DESIGN

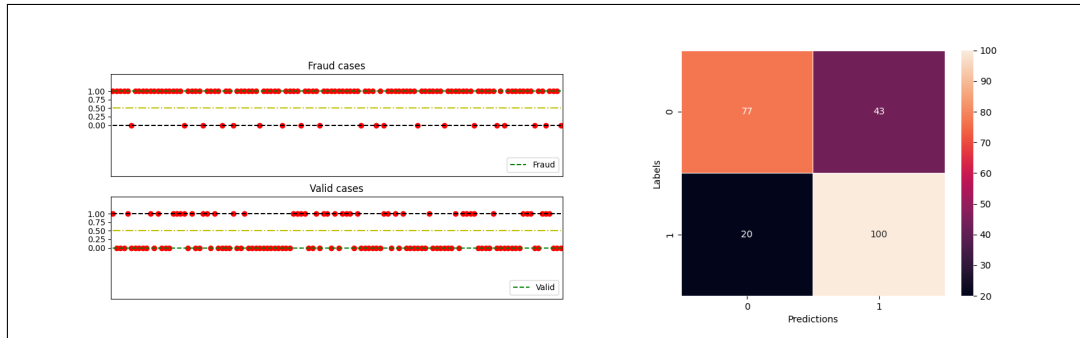


Figure 7.11: Hybrid sequential model classification plot and confusion matrix.

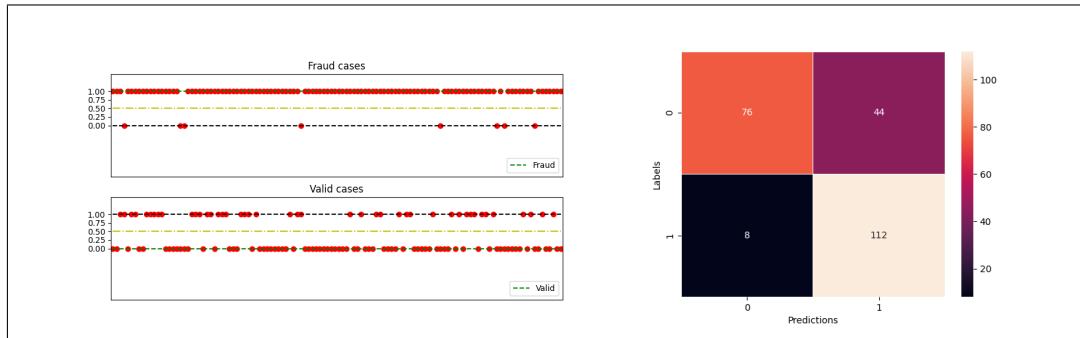


Figure 7.12: Hybrid functional model classification plot and confusion matrix.

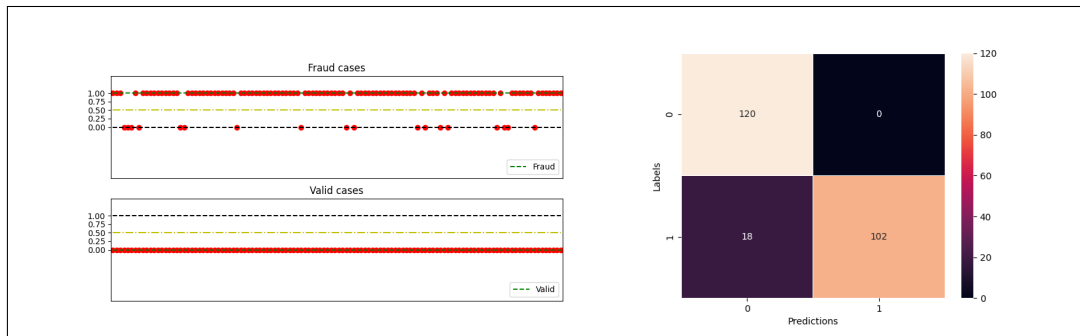


Figure 7.13: Classical model classification plot and confusion matrix.

### 7.4.3 Interpretation and reasoning of the results

The obtained results reflect the performance of the five models with different configurations presented. When designing the characteristics of the models, the key, to use fully the diversity allowed and compress it, was considering as many possible combinations and detecting which ones were essential from the more incidental ones. Therefore, the results represent improvements of taking the better decisions when selecting the many methodologies and techniques presented.

A general view of the classification problem can be given when observing the common issue for the quantum models and hybrid models. There is a particularly



nondeterministic behaviour when dealing with valid cases (0 label) given the difficulty for detecting them accordingly (a bit above 0.6 recall for most quantum models). The root of the problem may be due to the undersampling technique used for reducing considerably the number of valid cases and provide a balanced distribution to the train split, which inevitably has reduced the distinctive data points of valid cases included in the train split following the undersampling technique's criteria (*Near Miss 1* algorithm) and hampered the parameter optimization process in retrospective which prevented a better capture of the patterns in the feature space.

The quantum model using data reuploading has a quite random behaviour (F1-score 0.63 for label 1 and 0.45 for label 0) when compared to using amplitude embedding (F1-score 0.77 for label 1 and 0.71 for label 0) when introducing the data into the quantum circuit. The main issue this time is considering how the data is processed in the quantum circuit. When including the data as standard elementary operations (rotations) in each layer in the data reuploading quantum model, the quantum circuit has a higher difficulty when tuning the parameters. Instead, amplitude embedding allows the data to be represented as probability amplitudes, which greatly improves the performance of the quantum circuit over the alternative. The transition to the sequential part in the hybrid sequential model doesn't provide too much benefit either (F1-score 0.76 for label 1 and 0.70 for label 0) when adding classical layers to the quantum model with amplitude embedding after the quantum circuit.

The next meaningful step can be found in the hybrid functional model (F1-score 0.81 for label 1 and 0.74 for label 0) in contrast with both the hybrid sequential model and the quantum model with amplitude embedding. The main advantage the hybrid functional model presents is the shift from processing the entire data in a single sequence to instead dividing it, which alleviates the burden to adequate the quantum circuits' weights into two even subsets of the feature space. Moreover, the classical layers are processed before the quantum circuits in the hybrid functional model unlike in the hybrid sequential model.

Nevertheless, the main reason, for which the order of the classical layers going before the quantum circuits in the hybrid functional model was selected, has been the type of embeddings the quantum circuits in the hybrid functional model include. The change from amplitude embedding to angle embedding has notable consequences as the results portray, since the data is inserted no longer as a probability amplitude causing a condensed representation of the input.

The angle embedding only includes one rotation in each of the qubit for each feature and, while it requires a larger number of qubits to encapsulate the entire data to one qubit per feature, the classical layer reducing the dimension of input features to only five and the entire input being split into two at the beginning help greatly in compensating for the reduced dimension of the data features. The data input entering the quantum circuit, owing to the angle embedding requiring as many qubits as the input dimension, is represented more broadly than when

amplitude embedding is selected therefore. Even if there are shortcomings to it.

Another aspect that differentiates the hybrid models from the quantum models is the loss function used. While the hybrid models include a sigmoid activation function together with the binary cross entropy loss function, the quantum models provide an expectation value of the operator in each qubit which is used to calculate the distance in the mean squared error loss function. There does not seem to be too large of difference in this aspect and therefore the expectation values used with the mean squared error work as intended without relatively hindering the optimization process in principle.

The last step is to consider whether quantum circuits serve a purpose or provide a more subtle classification in contrast with directly tackling the classification task exclusively with classical means. The results show an inclination in quantum models and hybrid models on classifying more accurately fraud cases in contrast with the classical model (F1-score 0.72 for label 1). On the contrary, the classical model obtains a near perfect score (F1-score 0.99 for label 0) when classifying valid cases while quantum models and hybrid model have more difficulties detecting these cases correctly.

Therefore, in the present circumstances, there is a slight advantage when using quantum models and hybrid models if detecting potential fraud cases is a priority, which, in the scope of this classification task, it is a great priority in many cases.

## Conclusions

The premise of the entire work was to connect both classical and quantum machine learning capabilities, without disregarding one from another and instead combining them. This has proved to be particularly insightful in the sense that has inspired hybrid models that have shown the best of both worlds. On that note, quantum systems are both highly anticipated yet uncharted territory that will be discussed for years to come.

The research work carried out on the quantum circuits and merging them with classical machine learning approaches has provided an important fact, that is, at this moment understanding the underlying representation of the machine learning problems, designing a hybrid model according to the specifications and necessities of the data and thoughtfully organizing quantum circuits' settings is already a thing of the present. Even when dealing with task that present such difficult representations as occurs with unbalanced data and unidentifiable features, a common issue in the machine learning community and data management teams.

The choice of the linear operator for obtaining the expectation value, the use of the convenient loss functions to learn from data, adjusting the learning rate of the right optimizer for the problem, adding nonlinearity with classical and quantum layers and notably using the proper selection of the type of embedding to incorporate data into the quantum circuit are some of the most valuable and enriching qualities of hybrid models.

All in all, with quantum systems already showing promising future, as larger the amount of resources grows in terms of computational capabilities or quantum systems magnitude (more qubits) and complexity (more intricate and richer representations of problems), the more close the current technology will be for even more advances. The current performance of quantum models both hybrid and not together with more quantum approaches to deal with complicate tasks are paving the way to even more diverse classical-quantum solutions.



# Bibliography

- [1] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.
- [2] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.
- [3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, Sep 2017.
- [4] Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell, 2017.
- [5] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.
- [6] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.
- [7] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.
- [8] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [9] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13), Sep 2014.
- [10] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.
- [11] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):1–7, 2016.
- [12] Misha Denil and Nando De Freitas. Toward the implementation of a quantum rbm. 2011.
- [13] Vincent Dumoulin, Ian Goodfellow, Aaron Courville, and Yoshua Bengio. On the challenges of physical implementations of rbms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.
- [14] Steven H Adachi and Maxwell P Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.
- [15] Kristan Temme, Tobias J Osborne, Karl G Vollbrecht, David Poulin, and Frank Verstraete. Quantum metropolis sampling. *Nature*, 471(7336):87–90, 2011.
- [16] Man-Hong Yung and Alán Aspuru-Guzik. A quantum–quantum metropolis algorithm. *Proceedings of the National Academy of Sciences*, 109(3):754–759, 2012.
- [17] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum deep learning. *arXiv preprint arXiv:1412.3489*, 2014.

## BIBLIOGRAPHY

---

- [18] Anirban Narayan Chowdhury and Rolando D Somma. Quantum algorithms for gibbs sampling and hitting-time estimation. *arXiv preprint arXiv:1603.02940*, 2016.
- [19] Mohammad H Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Physical Review X*, 8(2):021050, 2018.
- [20] Maria Kieferova and Nathan Wiebe. Tomography and generative data modeling via quantum boltzmann training. *arXiv preprint arXiv:1612.05204*, 2016.
- [21] Yao Zhang and Qiang Ni. Recent advances in quantum machine learning. *Quantum Engineering*, 2(1):e34, 2020.
- [22] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J Osborne, Robert Salzmann, Daniel Scheiermann, and Ramona Wolf. Training deep quantum neural networks. *Nature communications*, 11(1):1–6, 2020.
- [23] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.
- [24] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):1–9, 2021.
- [25] Ismail Yunus Akhalwaya, Shashanka Ubaru, Kenneth L. Clarkson, Mark S. Squillante, Vishnu Jejjala, Yang-Hui He, Kugendran Naidoo, Vasileios Kalantzis, and Lior Horesh. Exponential advantage on noisy quantum computers, 2022.
- [26] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, 2013.
- [27] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019.
- [28] Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan H. S. Derks, Paul K. Faehrmann, and Johannes Jakob Meyer. Training quantum embedding kernels on near-term quantum computers, 2021.
- [29] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning, 2020.
- [30] Robert Patton, Catherine Schuman, Thomas Potok, et al. Efficiently embedding qubo problems on adiabatic quantum computers. *Quantum Information Processing*, 18(4):1–31, 2019.
- [31] Vicky Choi. Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Information Processing*, 7(5):193–209, 2008.
- [32] Andrew Blance and Michael Spannowsky. Quantum machine learning for particle physics using a variational quantum classifier. *Journal of High Energy Physics*, 2021(2):1–20, 2021.
- [33] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.
- [34] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, 122(4), feb 2019.

- [35] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, mar 2019.
- [36] Maria Schuld and Francesco Petruccione. *Supervised learning with quantum computers*, volume 17. Springer, 2018.
- [37] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3), mar 2020.
- [38] Advanced quantum algorithms. <https://www.chalmers.se/en/centres/wacqt/graduate%20school/aqa/Pages/default.aspx>. Accessed: 2021-09-01.