Quantum machine learning: on the design, trainability and noise-robustness of near-term algorithms

Skolik, A.

CHAPTER 1

Introduction

> *"The history of classical computing teaches us that when hardware becomes available that stimulates and accelerates the development of new algorithms. There are many examples of heuristics that were discovered experimentally, which worked better than theorists could initially explain. We can anticipate that the same thing will happen with quantum computers."*
>
> – John Preskill

Since the first ideas for a quantum computer, a device that uses quantum mechanical effects to process information, were proposed in the 1980s, remarkable progress has been made in the quest to build these types of machines and to find tasks where they outperform their classical counterparts. While the first algorithm with a proven speed-up in a practically relevant task, namely Shor's factoring algorithm that can break a widely used encryption method [1], was already proposed in 1994, the first experimental demonstration of a quantum computer outperforming its classical counterpart was only due in late 2019. In this experiment, a team of researchers at Google and NASA showed that their superconducting quantum hardware could perform a sampling task much faster than any classical supercomputer existing to that date [2]. Just one year later, a similar result on another sampling task was shown on the Chinese photonic quantum device Jiuzhang [3]. In parallel to these experimental advances, theoretical investigation has lead to a multitude of quantum algorithms with guaranteed speed-ups. These algorithms operate under the assumption that one can execute them on a perfect quantum computer, also known as fault-tolerant quantum computing. Among these algorithms are for example Grover's search algorithm, which yields a quadratic speed-up for

unstructured search [4], and an algorithm for solving linear systems of equations that provides an exponential speed-up under certain conditions and in certain cases [5].

The latter has inspired investigation of using quantum computers in the context of machine learning, as the algorithm from [5] can be used to exponentially speed up some of the matrix operations underlying many classical machine learning algorithms [6, 7]. However, these speed-ups come with a set of caveats: there are specific constraints on the structure of the matrices, as well as the assumption that the classical data is stored in a so-called quantum random access memory (qRAM) [8], a memory that gives the algorithm access to data in quantum superposition. As there are numerous technical challenges in building a qRAM, as well as subtleties in when exactly these types of speed-ups exist [9, 10, 11], it is not believed that algorithms based on speeding up solving linear systems of equations will lead to a practical advantage over classical computers in the near future.

In recent years, an alternative to fault-tolerant algorithms has emerged in the so-called noisy intermediate-scale quantum (NISQ) setting, where algorithms for near-term quantum computers are studied [12, 13]. These algorithms are designed with the expectation that they are run on noisy quantum computers with a limited number of qubits and no error correction. One of the most popular approaches in this regime are variational quantum algorithms [14]. In a variational quantum algorithm (VQA), a classical and quantum part work in tandem, where the quantum part of the algorithms is defined in terms of a parametrized quantum circuit (PQC). The parameters of this circuit are then optimized by a classical subroutine in order to solve a certain task. These types of algorithms have been applied to various tasks such as optimization [15, 16, 17], chemistry and simulation [18, 19, 20, 21, 22], and machine learning [23, 24, 25, 26, 27, 28].

While variational algorithms are widely applicable in principle, theoretical statements about their performance and potential speed-ups are difficult to obtain. Especially in a machine learning context, giving rigorous statements in a variational setting is hard, as the performance of the algorithm strongly depends on several factors, like the classical optimizer and the learning task at hand. This resembles the situation in classical machine learning, where first theories of neural networks were developed in the 1940ies, however, the true capabilities of these models only became clear around seventy years later when the increase in computational resources enabled training of large-scale neural networks to perform practically

relevant tasks. While recent breakthroughs in classical machine learning, like beating a grandmaster in the game of Go [29], predicting the structure of proteins [30, 31], or turning natural language prompts into stunning artwork [32], were only possible because current hardware makes it feasible to train models with billions of parameters, these works were built on the basis of a firm understanding on how to design trainable and performant neural networks.

Variational quantum machine learning models are often described as the quantum analog of classical neural networks due to the similarity in their training procedure, and are therefore also referred to as quantum neural networks. Unlike for their classical counterparts however, there are still numerous open questions about how to design trainable and performant quantum neural networks. Examples of this include the questions of how to encode classical data into a quantum model, how to structure the operations that are used to implement models, and how to avoid pitfalls in the trainability of these models that are unique to the quantum setting. Assuming that similarly to the history of classical machine learning, the development of more performant quantum hardware will facilitate large-scale empirical studies on the usefulness of variational quantum machine learning, it is of key importance to build an understanding of how these models can be trained successfully. This thesis aims to contribute to this understanding by studying various aspects of training variational quantum machine learning models.

We start by giving a basic introduction to the topics of quantum computing, machine learning, and their intersection in Chapters 2 and 3, respectively. In Chapter 4, we study how a fundamental issue in the training of variational quantum circuits, namely barren plateaus in the training landscapes, can be addressed by the classical training algorithm to aid scaling up the size of quantum models. To this end, we provide a training scheme that alleviates the problem of barren plateaus for specific cases and compare it to standard training procedures in the existing literature. While this type of training procedure can in principle be used for arbitrary types of machine learning, we focus our attention on a specific type of learning in subsequent chapters, namely on reinforcement learning (RL). First, we study in Chapter 5 how the architectural choices made for a PQC-based quantum agent influence its performance on two classical benchmark tasks from RL literature, where we specifically consider the question of encoding data into, and reading information out of the quantum model. In addition, we establish a theoretical separation between classical and quantum models for the specific type

of RL algorithm that we use, and also perform an in-depth empirical comparison of the quantum model developed in our work to a classical neural network that performs the same task. In addition to the questions of how to encode data and read out information from a PQC, the third key question in the performance of a variational quantum machine learning model is how to design the structure of the circuit itself, also referred to as the ansatz. For this reason, we move on to study this question in Chapter 6 and introduce an ansatz that is tailored to a specific type of input data, namely to weighted graphs. To do this, we take inspiration from the classical field of geometric deep learning, and design a PQC that preserves an important symmetry in graph-based input data. We analytically study the expressivity of this type of circuit, and then go on to numerically compare it to ansatzes that are not tailored to the specific training data at hand. Finally, another important consideration in the study of algorithms for the NISQ era is how the given learning algorithms and models are influenced by quantum hardware-induced noise. In Chapter 7, we study this for two of the variational RL paradigms from recent literature. We investigate analytically and numerically how various types of errors, namely coherent, incoherent, and measurement-based errors, affect the training performance of variational RL algorithms and the robustness of the learned policies. In particular, this study includes an evaluation of the performance of the models we introduced in Chapter 5 and Chapter 6 under various types of noise that are expected to be present on near-term hardware.

With the above, this thesis aims to contribute to building a foundation of knowledge about how to successfully train variational quantum machine learning models, in the hope that similarly to classical machine learning, this knowledge will one day, when quantum hardware has sufficiently matured, aid demonstrations of the practical usefulness of these types of algorithms.