PC3130 Quantum Mechanics II Project
From Bits to Qubits: A high-level overview of Quantum Machine Learning

Qian Jun Beh* and Zhu Ming Hor†

National University of Singapore

(Dated: October 20, 2023)

Quantum Machine Learning (QML) marries machine learning (ML) with quantum computing, offering a transformative approach to solving complex, and classically intractable problems. These challenges encompass simulating quantum systems and handling high-dimensional data, where quantum speedups can be brought about by QML to outperform classical approaches.

This report will embark on the historical exploration of quantum computing and QML, before delving into a high-level overview of QML's protocols. From there, beginning with the introduction to data encoding, followed by discussions on constructing quantum circuits, several methods for loss minimisation will be featured. At the same time, various applications and advantages that QML offers will also be highlighted at the end of this report.

I. INTRODUCTION

A. Background & Motivation

One of the earliest theoretical formulations of quantum computers could be dated back to the quantum circuit model proposed by David Deutsch in 1985 [1]. Subsequently, there was the flourish in the field of quantum computing, culminating in the emergence of the HHL (Harrow-Hassidim-Lloyd) algorithm [2], which is often regarded to be the onset for QML [3].

Concurrently in the field of computer science, the term "machine learning" was popularised in 1950s, by Arthur Samuel's work on developing a checkers-playing programme [4]. Today, the demarcation between the realms of quantum mechanics and computer science has blurred, and mankind has started to harness the power of QML.

QML offers a myriad of potentials, from superpolynomial speedups in prime factoring by Shor's algorithm [5], to quantum parallelism. For the great potential that QML harbour, there is a pressing need to adapt to the paradigm shift that QML brings about, and this is also one of the motivations of embarking on this topic.

This report aims to provide a high-level overview of QML, encompassing the fundamental concepts and knowledge necessary for a comprehensive understanding.

B. QML Approaches

QML approaches can be sectioned into four groups: CC, CQ, QC, and QQ [6][7], this report primarily delves into the discussion of CQ.

* e0735447@u.nus.edu

† e0735464@u.nus.edu

II. DATA ENCODING

Data encoding is a crucial step in QML, analogous to the data preprocessing step in classical ML. It involves embedding numerical data points into quantum states using a quantum circuit called a feature map. In this report, some common data encoding methods will be introduced.

A. Angle Encoding

To encode N features [8] using angle encoding, each qubit is rotated by an angle determined by the feature's value:

$$|x\rangle = \bigotimes_{i=1}^{N} cos(x_i)\,|0\rangle + sin(x_i)\,|1\rangle \tag{1}$$

and the state preparation unitary is:

$$S_{x_j} = \bigotimes_{i=1}^{N} U(x_j^{(i)}) \tag{2}$$

where

$$U(x_j^{(i)}) = \begin{bmatrix} cos(x_j^{(i)}) & -sin(x_j^{(i)}) \\ sin(x_j^{(i)}) & cos(x_j^{(i)}) \end{bmatrix} \tag{3}$$

Though only one data point can be encoded at one time, it only needs at most N qubits, which makes it tractable on NISQ devices [9]. However, simple encoding methods cannot present a quantum advantage over classical classifier models, and a more complicated feature map that is difficult to simulate classically is needed [10], which will be illustrated in the next section.

B. Pauli Feature Map

A feature map encodes classical vectors $\mathbf{x}$ to quantum states $|\psi\rangle$. The choice of feature maps varies according to the specific problem that we are trying to solve. In general, the mapping is done by applying some unitary operations $\mathcal{U}$ to an initial state $|0\rangle^n$, in which n denotes the number of qubits, or in this context, represents the number of encoding features.

In a Pauli feature map, a Hadamard gate is applied to each initial state, followed by a Pauli matrix, in each layer. The general formulation of a Pauli feature map is as follows:

$$\mathcal{U}_{\Phi(\mathbf{x})} = \prod_d U_{\Phi(\mathbf{x})} H^{\otimes n} \tag{4}$$

In [10], the second-order Pauli Z-evolution (or `ZZFeatureMap` as defined in Qiskit) is said to be having the quantum advantage, due to the presence of entanglement between qubits.

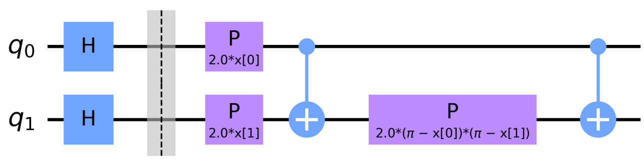

FIG. 1. A one-layer two-qubit `ZZFeatureMap`. H denotes the Hadamard gate, while the P-gate in the figure represents the Pauli-Z gate. [11]

III. ANSATZES

Given a reference state $|\rho\rangle$ mapped from an initial state $|0\rangle^n$ using a feature map $\mathcal{U}_\mathcal{R}$, a variational form (layers of parameterised gates) $\mathcal{U}_\mathcal{V}$ is then acting on $|\rho\rangle$. This ensemble is called an **ansatz**.

A. Two-Local Circuit

A widely used ansatz is the N-local circuit, which generally consists of at least two layers: the rotation layer and the entanglement layer. For the circuit to be parameterised, at least one of the layers has to be set with parameters θ, in order to be optimised for the minimisation of the cost function, which will be discussed further in the next section. A common example of an N-local circuit is the 2-local circuit (refer Appendix B 4), in which the entanglement is created between two qubits, hence the name.

IV. LOSS MINIMISATION

Similar to classical ML, loss minimisation is often regarded as the primary goal in QML, where a cost function [12] is minimised to obtain an optimal parameter $\vec{\theta^*}$ for the quantum variational circuit, via some optimisation algorithms. In most of the current QML implementations, classical optimisation strategies (gradient-based or gradient-free) were used. However, in some algorithms, optimisation techniques that harness quantum properties, like Quantum Natural Gradient Descent and Quantum Approximate Optimization Algorithms (QAOA) were also used.

A. Quantum Variational Circuits and Parameter Optimization

Quantum variational circuits are the essence of many QML models. To illustrate, the 2-local circuit in Section III A is one of them, with adjustable parameters that govern its behaviour.

The optimisation commences by initialising the initial state, $|\Psi_0\rangle$ and an initial set of free parameters, $\vec{\theta}_0$. The initial state is then passed to the quantum circuit, $\mathcal{U}_\mathcal{V}(\vec{\theta})$ to generate quantum predictions, $\mathcal{U}_\mathcal{V}(\vec{\theta}_0)|\Psi_0\rangle$. The design of the circuit is often based on the ML task at hand.

Subsequently, measurements are made on the quantum states produced by the circuit using arbitrary observable, $\hat{H}$. This operation transforms the quantum states, $\mathcal{U}_\mathcal{V}(\vec{\theta})|\Psi\rangle$ back to classical data $\langle\Psi|\mathcal{U}_\mathcal{V}{}^\dagger(\vec{\theta})\hat{H}\mathcal{U}_\mathcal{V}(\vec{\theta})|\Psi\rangle$, yielding outcomes that can be directly compared with the actual target values [13]. This is essentially the assessment of the cost, which is a measure of differences between the model's predictions from the variational circuit, and the target values.

Then, the optimisation algorithm updates the free parameter, $\vec{\theta}$ by minimising the cost function

$$\min_{\vec{\theta}} \left\langle \Psi(\vec{\theta}) \middle| \hat{H} \middle| \Psi(\vec{\theta}) \right\rangle \tag{5}$$

, where $\left|\Psi(\vec{\theta})\right\rangle = \mathcal{U}_\mathcal{V}(\vec{\theta})|\Psi\rangle$. This iterative process persists until a predefined convergence criterion [14] is met.

B. Classical Gradient Descent

Gradient descent as one of the most common optimisation methods in ML, leverages the gradient of the loss function to update parameters. The basic idea is to iteratively adjust the parameters in the direction of the steepest descent of the loss function [15]

While classical gradient descent is effective in many ML scenarios, it faces unique challenges in QML. The non-convex cost functions [16] in QML problems have multiple local minima, and traditional optimisation algorithms might be trapped in suboptimal solutions. Moreover, the classical optimisation algorithms are inherently designed for Euclidean parameter spaces. The choice of optimisation method is intimately linked to the choice of geometry on the parameter space [17]. In some cases, the geometry of the parameter space may not be well-suited for the classical optimization algorithms, leading to suboptimal performance.

C. Quantum Natural Gradient Descent

To address the limitations of classical gradient descent in QML, Quantum Natural Gradient Descent (QNGD) [18] emerges as a quantum-tailored optimisation method. At the core of QNGD is a transformation of the parameter space, by introducing the Quantum Fisher Information Matrix, $g^{-1}(\vec{\theta}_n)$ into the iterative equation.

$$\vec{\theta}_{n+1} = \vec{\theta}_n - \eta\, g^{-1}(\vec{\theta}_n)\boldsymbol{\nabla} f(\vec{\theta}_n) \tag{6}$$

where $g_{ij}(\vec{\theta}) = \mathrm{Re}\Big\{\Big\langle \frac{\partial\psi}{\partial\theta_i}\Big|\frac{\partial\psi}{\partial\theta_j}\Big\rangle - \Big\langle \frac{\partial\psi}{\partial\theta_i}\Big|\psi\Big\rangle\Big\langle\psi\Big|\frac{\partial\psi}{\partial\theta_j}\Big\rangle\Big\}$.

This transformation allows the optimisation process to shift from the steepest descent in the Euclidean parameter space to the steepest descent direction in Quantum Information Geometry [17] [18]. Therefore, QNG allows for faster and more reliable convergence compared to classical gradient descent, especially in high-dimensional spaces [17] [18].

D. Gradient-Free Optimisation

While gradient-based optimisation is common in QML, the computational cost associated with computing exact gradients grows linearly with respect to the dimension of parameter space [17]. Henceforth, Simultaneous Perturbation Stochastic Approximation (SPSA) stands out as a notable gradient-free approach.

SPSA replaces the computationally expensive gradient computation with stochastic approximations of the gradient. It achieves this by leveraging unbiased random sampling to simultaneously perturb all parameters in a random direction to estimate the gradient [17](refer to Appendix for more details).

$$\boldsymbol{\nabla} f(\vec{\theta}_n) = \begin{pmatrix} \frac{\partial f}{\partial\theta_1} \\ \vdots \\ \frac{\partial f}{\partial\theta_n} \end{pmatrix} \approx \frac{f(\vec{\theta}+\epsilon\vec{\Delta}) - f(\vec{\theta}-\epsilon\vec{\Delta})}{2\epsilon}\vec{\Delta} \tag{7}$$

where n-dimensional discrete uniform distribution, $\vec{\Delta} = \mathcal{U}\{1,-1\}^n$ and perturbation step size, ϵ.

With SPSA, the computational cost of evaluating the gradient is reduced significantly. Additionally, the unbiased estimator for the gradient when the random perturbations are sampled appropriately allows SPSA to effectively navigate the noisy landscape of the cost function and converge to the global minimum.

E. Quantum Natural SPSA

To further improve SPSA, Quantum Natural SPSA (QN-SPSA) stands out as a quantum counterpart of the classical SPSA [17] [19]. QN-SPSA integrates the principles of the Quantum Fisher Information Matrix to enhance the optimisation process. The formula is exactly the same as Eq (B7) with gradient approximation using Eq (B6). This enhancement allows QN-SPSA to provide more informed updates for quantum variational circuits, allow for more effective navigating of the complex parameter space and improve the convergence of quantum models [20].

V. APPLICATION

Traditional ML models often require vast computational resources for training. With QML, the ability to process and learn from data at a much faster rate can lead to substantial economic benefits in terms of reduced computational costs[i will add some footnote later] and accelerated decision-making [21].

Scalability is another pivotal advantage of QML. Although it is now possible to reduce dimensionality of data using autoencoders, the training process is still expensive. As quantum computers are inherently well-suited to handle high-dimensional models, QML is able to process high-dimensional data. This scalability allows us to tackle complex problems that were previously infeasible, which is particularly valuable in fields such as quantum chemistry [22], where the complexity of the data and models is inherently high.

The Quantum Fisher Information Matrix (QFIM) adds another layer of efficiency to the optimisation process in QML. By leveraging the QFIM, optimisation algorithms can make more informed and faster decisions and, therefore converge rapidly. This is of paramount importance in applications like quantum chemistry, where the accurate determination of molecular properties is resource-intensive. The QFIM accelerates the optimisation process, making it not only faster but also more reliable.

VI. CONCLUSION

In summary, this report has delivered a comprehensive outline of QML, encompassing three fundamental facets: data encoding, ansatzes, and loss minimisation. The discussion initiated with an introduction to angle encoding, a widely-used data encoding method, and an illustrative implementation using the Qiskit library. Subsequently, the 2-local circuit, a common ansatz, was scrutinised in detail. This report also presented a comparative view of classical and quantum optimisation strategies, followed by the potential computational advantages that could be brought about by QML.

Appendix A: Critques on ChatGPT

Question:
What is the difference between quantum natural gradient descent and classical gradient descent?

ChatGPT:
Quantum Natural Gradient Descent: QNGD is designed to navigate the optimisation landscape in the context of quantum algorithms. It takes into account the curvature of the quantum cost function, providing a more efficient optimisation process for certain quantum machine learning tasks.
Classical Gradient Descent: Classical gradient descent optimises functions in the classical computing domain, considering the gradients of the cost function with respect to classical parameters.

ChatGPT has presented a thorough comparison between Quantum Natural Gradient Descent (QNGD) and Classical Gradient Descent (CGD). Though what it output was not incorrect, nevertheless, someone who is unfamiliar with the topic might find the explanation lacking clarity, as it was mostly just a blanket of generalisation. More details would, of course, be needed for a more comprehensive understanding.

To elaborate, CGD operates within the Euclidean parameter space, with the goal of progressing toward the global minimum of the objective function. In contrast, QNGD delves into the Model Parameters space, also recognised as Quantum Information Geometry.

Further clarification on the spatial transformation process would involve introducing the Quantum Fisher Information Matrix into the iterative equation. Briefly explaining, QFIM plays a crucial role in providing information on how to navigate the model space efficiently.

Appendix B: More Details

1. Pauli Feature Map

In Equation (4), the U_ϕ in the product is given by:

$$U_{\Phi(\mathbf{x})} = \exp\left\{ i \sum_{S\subset[n]} \phi(\mathbf{x}) \prod_{k\in S} P_i \right\} \tag{B1}$$

where $P_i \in \{I, X, Y, Z\}$, which are Pauli matrices. The S in the index denotes the connectivity between qubits, with $S \in \binom{n}{k}, k = 1, 2, ..., n$, and the default mapping function $\phi_S(\mathbf{x})$ is:

$$\phi_S : \mathbf{x} \mapsto \begin{cases} x_i & \text{if } S = \{i\} \\ (\pi - x_i)(\pi - x_j) & \text{if } S = \{i, j\} \end{cases} \tag{B2}$$

2. Hadamard Gate

From all the aforementioned quantum circuits, there will always be Hadamard gates at the beginning of each layer. **Hadamard gate**, often denoted as **H**, can be simply understood as the creator of superposition states, as it maps $|0\rangle$ and $|1\rangle$ to $|+\rangle$ and $|-\rangle$ respectively. More precisely,

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{B3}$$

3. 2-local Circuit

A 2-local circuit can be visualised using the Qiskit's `TwoLocal()` class, as illustrated in the figure below:

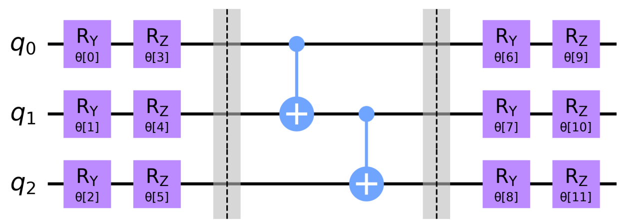

FIG. 2. A three-layer 2-local circuit with three qubits. The gates in blue are CNOT gates, which create entanglement between two adjacent qubits in this circuit. [11]

4. CNOT Gates

A CNOT gate works by flipping $|+\rangle$ to $|-\rangle$ and vice versa when the second qubit is $|-\rangle$ in the Hadamard basis:

$$C_{NOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{B4}$$

$$\begin{aligned} C_{NOT}|+-\rangle &= |--\rangle \\ C_{NOT}|--\rangle &= |+-\rangle \\ C_{NOT}|++\rangle &= |++\rangle \\ C_{NOT}|-+\rangle &= |-+\rangle \end{aligned} \tag{B5}$$

5. SPSA

The SPSA algorithm starts with an initial point in the N-dimensional parameter space, denoted as $\vec{\theta}_0$. It then iteratively updates the parameters based on stochastic approximation. At each iteration k, SPSA samples a random direction $\vec{\Delta}_k$ from a suitable distribution, such as uniformly from $\{1, -1\}$ for each parameter. It then approximates the gradient of the objective function f with respect to the parameters, denoted as $\boldsymbol{\nabla} f(\vec{\theta}_n)$, using the following formula:

$$\boldsymbol{\nabla} f(\vec{\theta}_n) = \begin{pmatrix} \frac{\partial f}{\partial \theta_1} \\ \vdots \\ \frac{\partial f}{\partial \theta_n} \end{pmatrix} \approx \frac{f(\vec{\theta} + \epsilon\vec{\Delta}) - f(\vec{\theta} - \epsilon\vec{\Delta})}{2\epsilon}\vec{\Delta} \quad \text{(B6)}$$

.

This formula is derived from the idea of the Finite-Different method. However, SPSA replaced the computation of gradient by using unbiased random sampling.

The SPSA algorithm uses this stochastic approximation of the gradient to update the parameters at each iteration. The update rule is given by the standard gradient descent formula:

$$\vec{\theta}_{n+1} = \vec{\theta}_n - \eta \boldsymbol{\nabla} f(\vec{\theta}_n) \quad \text{(B7)}$$

where η is the learning rate, which determines the step size in the parameter space.

[1] D. Deutsch, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **400**, 97 (1985).

[2] A quantum algorithm that solves a system of linear equations on a quantum computer, specifically a sparse linear system.

[3] K. Najafi, S. F. Yelin, and X. Gao, **4** (2022).

[4] A. L. Samuel, Some studies in machine learning using the game of checkers, in *Computer Games I* (Springer New York, New York, NY, 1988) pp. 335–365.

[5] H. Yamasaki *et al.*, Learning with optimized random features: Exponential speedup by quantum machine learning without sparsity and low-rank assumptions, arXiv.org (2020).

[6] V. Dunjko, J. M. Taylor, and H. J. Briegel, Quantum-enhanced machine learning, arXiv.org (2016).

[7] Classical data using classical algorithm (CC), classical data using quantum algorithm (CQ), quantum data using classical algorithm (QC), quantum data using quantum algorithm (QQ).

[8] Features are often regarded as the predictor variables in machine learning.

[9] NISQ devices, short for Noisy-Intermediate-Scale-Quantum devices, describe the near-term quantum devices that are currently available or will be available in the near future.

[10] V. Havlicek *et al.*, Supervised learning with quantum enhanced feature spaces, arXiv.org (2018).

[11] Q. J. Beh and Z. M. Hor, QML (2023), GitHub repository.

[12] Cost function, as known as a loss function, is used to describe how well a model is performing with respect to a goal. In the case of classical ML, a cost fucntion could just be as simple as mean-squared error (MSE).

[13] Also known as the target variable, or the response variable.

[14] Depending on the specific problem, it could be a predefined number of iterations, or a tolerance value ϵ. If the current gradient of the objective function (for gradient-based method) is smaller than it, shall the iterations terminate.

[15] The parameters in Gradient descent are updated by:

$$\vec{\theta}_{n+1} = \vec{\theta}_n - \eta \boldsymbol{\nabla} f(\vec{\theta}_n) \quad \text{(B8)}$$

where η is the learning rate and f is the cost function.

[16] Convexity plays an important role in classical ML. Similar to how the Weierstrass Theorem can lead to the fact that a convex function guarantees a global minimum, the convexity of the objective function can also imply the convergence of Gradient Descent, which the mathematical proof is omitted here as the margin is too small to contain.

[17] J. Gacon, C. Zoufal, G. Carleo, and S. Woerner, Simultaneous perturbation stochastic approximation of the quantum fisher information, arXiv.org (2021).

[18] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, arXiv.org (2020).

[19] J. Gidi, B. Candia, A. D. Muñoz-Moller, A. Rojas, L. Pereira, M. Muñoz, L. Zambrano, , and A. Delgado, Stochastic optimization algorithms for quantum applications, arXiv.org (2023).

[20] There exist other approaches to improve SPSA, such as Power-Law SPSA and Autocalibrated SPSA, Complex SPSA.

[21] S. Adhikary, S. Dangwal, and D. Bhowmik, Supervised learning with a quantum classifier using a multi-level system, arXiv.org (2019).

[22] A. Peruzzo *et al.*, A variational eigenvalue solver on a photonic quantum processor, Nature Communications **5**, 1 (2014).