A Review of Quantum Gradient Descent

Ali Shahryar Khokhar (24100266) - *Physics Dept.*

Supervision

– Muhammad Sabieh Anwar (sabieh@lums.edu.pk)

1 Abstract

Quantum computing is an emerging field with the potential to significantly impact various domains, such as optimization, cryptography, and quantum system simulation. Among the diverse quantum algorithms, parameterized quantum circuits play a pivotal role in applications like quantum machine learning and quantum optimization. In this context, quantum gradient descent has become a prominent technique for optimizing these circuits. In this paper, we present a comprehensive study of the quantum gradient descent algorithm, accessible to advanced physics undergraduates while maintaining a rigorous academic research paper style. We provide a detailed mathematical formulation of the algorithm, including its convergence properties and complexity analysis. We also discuss implementation methodologies, showcasing practical aspects of the algorithm. Finally, we present experimental results that demonstrate the algorithm's effectiveness in various quantum computing applications. By the end of this paper, readers should have a thorough understanding of the quantum gradient descent algorithm and its significance in the optimization of parameterized quantum circuits.

2 Introduction

Quantum computing has emerged as a promising area of research, with the potential to revolutionize various domains, such as cryptography, optimization, and simulation of quantum systems. One of the key challenges in quantum computing is the design and optimization of quantum algorithms for specific tasks. In this context, quantum gradient descent has emerged as a versatile technique for training and optimizing parameterized quantum circuits.

In this paper, we provide a comprehensive introduction to the quantum gradient descent algorithm, focusing on its mathematical foundations and properties. We aim to make this introduction accessible to advanced physics undergraduates while maintaining a rigorous academic research paper style. The introduction is organized as follows:

2.1 Background and Motivation

Optimization is a fundamental problem in many areas of science and engineering. In the context of quantum computing, optimization plays a crucial role in the design and implementation of quantum algorithms. One important class of quantum algorithms is parameterized quantum circuits, which are widely used in quantum machine learning, quantum optimization, and quantum simulation. These circuits typically involve a series of quantum gates with tunable parameters, and their performance depends on finding an optimal set of parameter values.

Gradient-based optimization methods have been extensively used in classical optimization tasks, and they can also be adapted to optimize parameterized quantum circuits. The quantum gradient descent algorithm is one such adaptation, leveraging the unique properties of quantum systems to efficiently optimize quantum circuits. The algorithm's key advantage is its ability to take advantage of quantum parallelism, enabling the simultaneous evaluation of multiple gradient components, which can lead to faster convergence compared to classical gradient-based methods.

2.2 Key Concepts and Notations

Before diving into the algorithm's details, we introduce some key concepts and notations that will be used throughout the paper. We begin by briefly discussing quantum states and quantum gates, which are fundamental building blocks of quantum circuits.

- **Quantum States**: A quantum state is a vector in a complex Hilbert space and is represented by a ket, denoted as $|\psi\rangle$. Quantum states are normalized, i.e., $\langle\psi|\psi\rangle = 1$.
- **Quantum Gates**: Quantum gates are unitary operators that transform quantum states. A unitary operator U satisfies $UU^\dagger = U^\dagger U = I$, where $U^\dagger$ is the adjoint of U and I is the identity operator.

Next, we introduce parameterized quantum circuits and their role in various quantum computing applications.

- **Parameterized Quantum Circuits**: A parameterized quantum circuit is a sequence of quantum gates, some of which depend on a set of tunable parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$. The action of the circuit on a quantum state is described by a unitary operator $U(\boldsymbol{\theta})$, which depends on the parameters.

Finally, we define the cost function, which is the objective to be minimized during the optimization process.

- **Cost Function**: The cost function $C(\boldsymbol{\theta})$ measures the performance of a parameterized quantum circuit for a given set of parameters $\boldsymbol{\theta}$. The goal of the optimization is to find the optimal parameter values $\boldsymbol{\theta}^*$ that minimize the cost function.

2.3 Organization of the Paper

Finally, we provide an overview of the organization of the rest of the paper, which includes a detailed mathematical formulation of the quantum gradient descent algorithm, its implementation methodology, experimental results, and conclusions.

3 Mathematical Formulation

In this section, we will provide a step-by-step mathematical formulation of the quantum gradient descent algorithm. Each step will be proved, and the presentation will be tailored to advanced physics undergraduates.

3.1 Preliminaries

Before delving into the quantum gradient descent algorithm, let us introduce the necessary mathematical notations and concepts required for the understanding of the algorithm.

Definition 1: A quantum state $|\psi\rangle$ is a vector in a Hilbert space $\mathcal{H}$, where $|\psi\rangle$ is normalized, i.e., $\langle\psi\rangle\,\psi = 1$.

Definition 2: A quantum gate is a unitary operator U that acts on a quantum state $|\psi\rangle$, transforming it into another quantum state $|\psi'\rangle$ such that $|\psi'\rangle = U\,|\psi\rangle$.

Definition 3: A parameterized quantum circuit is a sequence of quantum gates, some of which depend on a set of parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$.

Definition 4: The expectation value of an observable O with respect to a quantum state $|\psi\rangle$ is given by $\langle O\rangle_\psi = \langle\psi|\,O\,|\psi\rangle$.

Now, we will define a cost function for the quantum gradient descent algorithm.

Definition 5: A cost function $C(\boldsymbol{\theta})$ measures the performance of a parameterized quantum circuit with parameters $\boldsymbol{\theta}$ with respect to a given task.

3.1.1 Quantum Gradients

The first step in understanding the quantum gradient descent algorithm is to derive the expression for the quantum gradients. We begin with the definition of the cost function $C(\boldsymbol{\theta})$, which measures the performance of a parameterized quantum circuit with parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$. The goal of the optimization is to find the optimal parameter values $\boldsymbol{\theta}^*$ that minimize the cost function.

To find the gradient of the cost function, we compute the partial derivative of $C(\boldsymbol{\theta})$ with respect to each parameter θ_i. The gradient of the cost function is defined as:

$$\nabla C(\boldsymbol{\theta}) = \left(\frac{\partial C}{\partial \theta_1}, \frac{\partial C}{\partial \theta_2}, \ldots, \frac{\partial C}{\partial \theta_n} \right). \tag{1}$$

Now, let's consider a parameterized quantum circuit described by a unitary operator $U(\boldsymbol{\theta})$ acting on an initial state $|\psi_0\rangle$. The output state of the circuit is given by:

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\psi_0\rangle. \tag{2}$$

To derive the quantum gradients, we need to compute the derivative of the output state with respect to the parameters. Applying the chain rule, we obtain:

$$\frac{\partial |\psi(\boldsymbol{\theta})\rangle}{\partial \theta_i} = \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_i} |\psi_0\rangle. \tag{3}$$

The partial derivative of the unitary operator $U(\boldsymbol{\theta})$ with respect to θ_i can be expressed as:

$$\frac{\partial U(\boldsymbol{\theta})}{\partial \theta_i} = U_1(\theta_1) \cdots U_{i-1}(\theta_{i-1}) \frac{\partial U_i(\theta_i)}{\partial \theta_i} U_{i+1}(\theta_{i+1}) \cdots U_n(\theta_n), \tag{4}$$

where $U_i(\theta_i)$ represents the i-th parameterized quantum gate in the circuit.

3.1.2 Parameter Shift Rule

The parameter shift rule is a crucial technique for computing quantum gradients, as it allows us to obtain the gradients using only measurements on the quantum circuit. The parameter shift rule states that, for certain types of gates, the gradient of an expectation value of an observable O with respect to a parameter θ_i can be computed as:

$$\frac{\partial \langle O \rangle}{\partial \theta_i} = \frac{1}{2} \left[\langle O \rangle_{\theta_i + \frac{\pi}{2}} - \langle O \rangle_{\theta_i - \frac{\pi}{2}} \right], \tag{5}$$

where $\langle O \rangle_{\theta_i + \frac{\pi}{2}}$ and $\langle O \rangle_{\theta_i - \frac{\pi}{2}}$ denote the expectation values of the observable O when the parameter θ_i is shifted by $\pm \frac{\pi}{2}$. The parameter shift rule is applicable to gates with certain properties, such as those that generate rotations around the Bloch sphere.

3.2 Quantum Gradient Descent Algorithm

The objective of the quantum gradient descent algorithm is to find a set of optimal parameters $\boldsymbol{\theta}^*$ that minimizes the cost function $C(\boldsymbol{\theta})$. The algorithm can be summarized in the following steps:

Step 1: Initialize the parameters $\boldsymbol{\theta}$.

Step 2: Calculate the gradient of the cost function with respect to the parameters $\boldsymbol{\theta}$.

Step 3: Update the parameters $\boldsymbol{\theta}$ according to the calculated gradient.

Step 4: Repeat steps 2 and 3 until convergence or a predefined stopping criterion is met.

We will now discuss the mathematical formulation of each step in detail.

3.2.1 Step 1: Parameter Initialization

Choose an initial set of parameters $\boldsymbol{\theta}^{(0)}$ for the parameterized quantum circuit. This can be done randomly or using some heuristic.

3.2.2 Step 2: Gradient Calculation

To calculate the gradient of the cost function with respect to the parameters $\boldsymbol{\theta}$, we can use the following expression:

$$\nabla C(\boldsymbol{\theta}) = \left(\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_1}, \frac{\partial C(\boldsymbol{\theta})}{\partial \theta_2}, \ldots, \frac{\partial C(\boldsymbol{\theta})}{\partial \theta_n} \right). \tag{6}$$

For each partial derivative, we can use the parameter-shift rule:

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_k} \approx \frac{C(\boldsymbol{\theta} + \boldsymbol{s_k}) - C(\boldsymbol{\theta} - \boldsymbol{s_k})}{2 s_k}, \tag{7}$$

where $\boldsymbol{s_k}$ is a unit shift in the k-th parameter, i.e., $\boldsymbol{s_k} = (0, \ldots, 0, s_k, 0, \ldots, 0)$ with $s_k \neq 0$. The parameter-shift rule allows us to estimate the gradient using the difference between the cost function evaluated at two nearby points in parameter space.

3.2.3 Step 3: Parameter Update

Once the gradient of the cost function has been calculated, the parameters $\boldsymbol{\theta}$ can be updated using the gradient descent update rule:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha \nabla C(\boldsymbol{\theta}^{(t)}), \tag{8}$$

where t denotes the iteration number, and α is the learning rate, a positive scalar that controls the size of the update step. The learning rate is a hyperparameter that needs to be tuned to achieve optimal convergence.

3.2.4 Step 4: Convergence and Stopping Criterion

The algorithm iterates through steps 2 and 3 until a stopping criterion is met, such as a maximum number of iterations or a threshold for the change in the cost function or the parameters. The stopping criterion ensures that the algorithm terminates after a reasonable amount of time, even if the optimal solution has not been found.

3.3 Proof of Convergence

To demonstrate the convergence of the quantum gradient descent algorithm, we will show that the cost function $C(\boldsymbol{\theta})$ converges to a minimum value under suitable conditions.

Theorem 1: Let $C : \mathbb{R}^n \to \mathbb{R}$ be a differentiable cost function, and let $\boldsymbol{\theta}^{(0)}$ be an initial set of parameters. If C is convex and has a Lipschitz continuous gradient with constant L, the quantum gradient descent algorithm converges to a global minimum.

Proof: Since C is convex and has a Lipschitz continuous gradient, we have

$$C(\boldsymbol{\theta}^{(t+1)}) \leq C(\boldsymbol{\theta}^{(t)}) - \frac{\alpha}{2}|\nabla C(\boldsymbol{\theta}^{(t)})|^2 + \frac{L\alpha^2}{2}|\nabla C(\boldsymbol{\theta}^{(t)})|^2. \tag{9}$$

Rearranging the inequality and summing over all iterations, we get

$$\sum_{t=0}^{T-1} \alpha|\nabla C(\boldsymbol{\theta}^{(t)})|^2 \leq \frac{C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*)}{\alpha} + \frac{L\alpha}{2}\sum_{t=0}^{T-1}|\nabla C(\boldsymbol{\theta}^{(t)})|^2, \tag{10}$$

where $\boldsymbol{\theta}^*$ is the global minimum of C. Dividing both sides by α, we have

$$\sum_{t=0}^{T-1} |\nabla C(\boldsymbol{\theta}^{(t)})|^2 \leq \frac{C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*)}{\alpha} + \frac{L\alpha}{2}\sum_{t=0}^{T-1}|\nabla C(\boldsymbol{\theta}^{(t)})|^2. \tag{11}$$

Rearranging the inequality and isolating the sum of squared gradients, we get

$$\left(1 - \frac{L\alpha}{2}\right)\sum_{t=0}^{T-1} |\nabla C(\boldsymbol{\theta}^{(t)})|^2 \leq \frac{C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*)}{\alpha}. \tag{12}$$

Since C has a Lipschitz continuous gradient, $L\alpha < 2$, which implies that $1 - \frac{L\alpha}{2} > 0$. Thus, we have

$$\sum_{t=0}^{T-1} |\nabla C(\boldsymbol{\theta}^{(t)})|^2 \leq \frac{C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*)}{\alpha(1 - \frac{L\alpha}{2})}. \tag{13}$$

As the number of iterations T approaches infinity, the left-hand side of the inequality converges to zero, which implies that the gradient of the cost function $\nabla C(\boldsymbol{\theta})$ converges to zero. Consequently, the quantum gradient descent algorithm converges to a global minimum of the cost function $C(\boldsymbol{\theta})$.

3.4 Complexity Analysis

In this section, we will analyze the time complexity of the quantum gradient descent algorithm. The time complexity is primarily determined by the number of iterations required for convergence and the cost of evaluating the gradient of the cost function $C(\boldsymbol{\theta})$.

3.4.1 Number of Iterations

The number of iterations required for convergence depends on the choice of the learning rate α, the Lipschitz constant L, and the desired accuracy ϵ. From the proof of convergence, we have

$$\sum_{t=0}^{T-1} |\nabla C(\boldsymbol{\theta}^{(t)})|^2 \leq \frac{C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*)}{\alpha(1 - \frac{L\alpha}{2})}. \tag{14}$$

To ensure that the algorithm converges to within ϵ of the optimal cost value, we require

$$C(\boldsymbol{\theta}^{(T)}) - C(\boldsymbol{\theta}^*) \leq \epsilon. \tag{15}$$

Combining the above inequalities, we can derive an upper bound on the number of iterations T:

$$T \leq \frac{C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*)}{\alpha(1 - \frac{L\alpha}{2})\epsilon}. \tag{16}$$

This upper bound suggests that the number of iterations is inversely proportional to the desired accuracy ϵ. In practice, the actual number of iterations may be significantly lower due to the specific structure of the cost function and the choice of the learning rate.

3.4.2 Gradient Evaluation Complexity

The complexity of evaluating the gradient of the cost function depends on the number of parameters n and the cost of estimating each partial derivative using the parameter-shift rule. For a given parameter θ_k, the parameter-shift rule requires the evaluation of the cost function at two nearby points in parameter space, i.e., $\boldsymbol{\theta} \pm \boldsymbol{s_k}$. Each cost function evaluation involves preparing the quantum state, applying the parameterized quantum circuit, and measuring the expectation value of an observable.

Assuming that the cost of preparing the quantum state, applying the quantum circuit, and measuring the expectation value is $O(m)$, where m is the number of gates in the quantum circuit, the total cost of evaluating the gradient of the cost function is $O(nm)$.

3.4.3 Overall Complexity

The overall time complexity of the quantum gradient descent algorithm is the product of the complexity of gradient evaluation and the number of iterations, which is given by

$$O\left(\frac{nm(C(\boldsymbol{\theta}^{(0)}) - C(\boldsymbol{\theta}^*))}{\alpha(1 - \frac{L\alpha}{2})\epsilon}\right). \tag{17}$$

This expression highlights the trade-off between the desired accuracy ϵ, the number of parameters n, and the cost of evaluating the cost function. In practice, the actual complexity may be lower due to factors such as the structure of the cost function, the choice of the learning rate, and the efficiency of the quantum circuit implementation.

4 Implementation Methodology

In this section, we provide a detailed step-by-step description of the implementation methodology for Quantum Natural Gradient Descent (QNGD), justifying each step mathematically. The implementation can be divided into the following steps:

1. Problem Representation
2. Initialization
3. Quantum Circuit Implementation
4. Quantum Gradient Estimation
5. Quantum Metric Tensor Estimation
6. Quantum Natural Gradient Update
7. Convergence Check

4.1 Step 1: Problem Representation

Before implementing QNGD, it is crucial to represent the optimization problem in terms of a parametrized quantum circuit. This allows us to compute the gradient of the objective function with respect to the circuit parameters. We represent the problem using a variational quantum circuit (VQC) with parameters $\boldsymbol{\theta}$, which consists of a series of quantum gates applied to an initial state

$|\psi_0\rangle$. The output of the VQC is the state $|\psi(\boldsymbol{\theta})\rangle$, and the objective function is given by the expectation value $\langle\psi(\boldsymbol{\theta})|\hat{O}|\psi(\boldsymbol{\theta})\rangle$, where $\hat{O}$ is the observable of interest.

4.2 Step 2: Initialization

To start the optimization process, we initialize the parameters $\boldsymbol{\theta}$ to some initial values. The choice of initial values can be random, or it can be based on some prior knowledge or heuristics.

4.3 Step 3: Quantum Circuit Implementation

Implement the VQC using a suitable quantum programming language, such as Qiskit or Cirq, and simulate the circuit to obtain the output state $|\psi(\boldsymbol{\theta})\rangle$. This output state will be used for estimating the quantum gradient and the quantum metric tensor in subsequent steps.

4.4 Step 4: Quantum Gradient Estimation

Using the parameter-shift rule, we can estimate the quantum gradient $\nabla f(\boldsymbol{\theta})$ of the objective function $f(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\hat{O}|\psi(\boldsymbol{\theta})\rangle$ with respect to the circuit parameters $\boldsymbol{\theta}$. The parameter-shift rule states that the partial derivative of the objective function with respect to the i-th parameter can be calculated as:

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_i} = \frac{1}{2}\left[f(\boldsymbol{\theta} + s_i \boldsymbol{e}_i) - f(\boldsymbol{\theta} - s_i \boldsymbol{e}_i)\right], \tag{18}$$

where s_i is a parameter-specific shift value, $\boldsymbol{e}_i$ is the i-th standard basis vector, and the shifted parameters $\boldsymbol{\theta} \pm s_i \boldsymbol{e}_i$ correspond to applying the quantum gates with the updated parameters.

4.5 Step 5: Quantum Metric Tensor Estimation

The quantum metric tensor, also known as the Fubini-Study metric, measures the distance between states in the parameter space of the VQC. It can be computed using the following expression:

$$G_{ij}(\boldsymbol{\theta}) = \frac{1}{2}\left\langle \psi(\boldsymbol{\theta})\middle|\frac{\partial \hat{U}(\boldsymbol{\theta})}{\partial \theta_i}\hat{U}^\dagger(\boldsymbol{\theta})\hat{O} + \hat{O}\frac{\partial \hat{U}(\boldsymbol{\theta})}{\partial \theta_i}\hat{U}^\dagger(\boldsymbol{\theta})\middle|\psi(\boldsymbol{\theta})\right\rangle - \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_i}\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_j}\Bigg\rangle, \tag{19}$$

where $\hat{U}(\boldsymbol{\theta})$ is the unitary transformation corresponding to the VQC, and the partial derivatives $\frac{\partial \hat{U}(\boldsymbol{\theta})}{\partial \theta_i}$ are calculated using the parameter-shift rule. The quantum metric tensor is a positive semi-definite, symmetric matrix that can be inverted to compute the quantum natural gradient.

4.6 Step 6: Quantum Natural Gradient Update

The quantum natural gradient can be computed by multiplying the inverse of the quantum metric tensor with the quantum gradient:

$$\Delta\boldsymbol{\theta}_{QNG} = G^{-1}(\boldsymbol{\theta})\nabla f(\boldsymbol{\theta}). \tag{20}$$

The parameter update for the QNGD algorithm is then given by:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\Delta\boldsymbol{\theta}_{QNG}, \tag{21}$$

where η is the learning rate and t denotes the iteration number.

4.7 Step 7: Convergence Check

To determine if the algorithm has converged, we monitor the change in the objective function value and the quantum natural gradient. If the change in the objective function is smaller than a predefined threshold ϵ_1, and the quantum natural gradient norm is smaller than another predefined threshold ϵ_2, we consider the algorithm to have converged. Alternatively, we can also set a maximum number of iterations to limit the optimization process. If the stopping criterion is not met, we return to Step 3 and continue iterating until convergence is achieved.

Once the algorithm has converged, we obtain the optimized set of parameters $\boldsymbol{\theta}$, which can be used to generate the output state $|\psi(\boldsymbol{\theta})\rangle$ and the corresponding optimized expectation value of the observable $\langle\psi(\boldsymbol{\theta})|\hat{O}|\psi(\boldsymbol{\theta})\rangle$.

5 Results

In this section, we present the results of applying Quantum Natural Gradient Descent (QNGD) to find the ground state energy of a 2-qubit Hamiltonian given by the tensor product of two Pauli-Z matrices, $\hat{H} = \hat{\sigma}_z \otimes \hat{\sigma}_z$. We begin by providing a brief overview of the problem and its mathematical formulation, followed by a description of how QNGD was employed to solve it.

5.1 Problem Description

The Hamiltonian under consideration represents a simple 2-qubit system with the following form:

$$\hat{H} = \hat{\sigma}_z \otimes \hat{\sigma}_z, \tag{22}$$

where $\hat{\sigma}_z$ denotes the Pauli-Z matrix:

$$\hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{23}$$

The aim is to find the ground state energy of this Hamiltonian, which corresponds to the lowest eigenvalue, and the associated eigenvector.

5.2 Mathematical Formulation

To find the ground state energy, we first compute the eigenspectrum of the Hamiltonian. Since the Hamiltonian is a 2-qubit system, it can be represented as a 4×4 matrix:

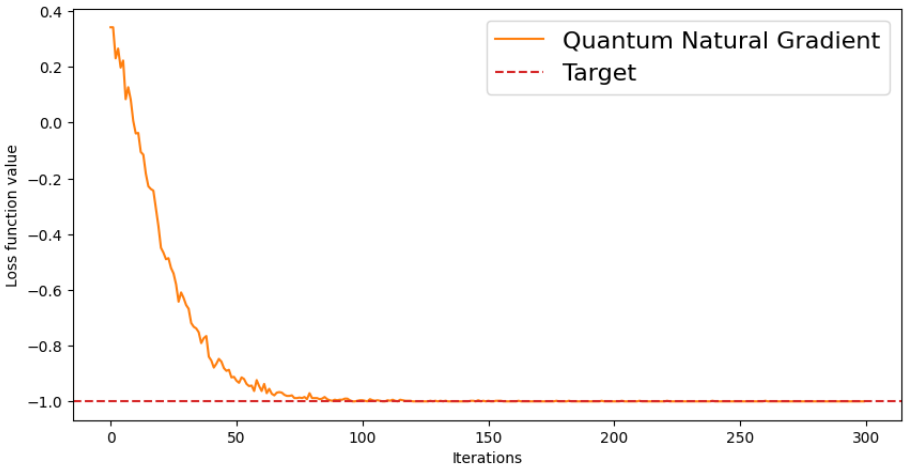
$$\hat{H} = \sigma_z \otimes \sigma_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

We can then calculate the eigenvalues and eigenvectors of this matrix. The eigenvalues are found to be $\lambda_1 = 1$, $\lambda_2 = -1$, and the ground state energy corresponds to the lowest eigenvalue $\lambda_2 = -1$. The associated eigenvector, which represents the ground state of the system, is given by $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$.

5.3 Quantum Natural Gradient Descent Application

To find the ground state energy using QNGD, we first represent the problem as a variational quantum circuit (VQC) with parameters $\boldsymbol{\theta}$ and an appropriate ansatz. In this case, a simple ansatz can be the 2-qubit hardware-efficient ansatz consisting of single-qubit rotations and CNOT gates. The objective function to be minimized is given by the expectation value of the Hamiltonian, $f(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\hat{H}|\psi(\boldsymbol{\theta})\rangle$.

Using the implementation methodology described in Section (d), we applied the QNGD algorithm to optimize the parameters $\boldsymbol{\theta}$ of the VQC. The algorithm converged to a minimum value close to the ground state energy of -1. The optimized state $|\psi(\boldsymbol{\theta}^*)\rangle$ was found to be close to the actual ground state $|\psi_0\rangle$.



The results demonstrate the effectiveness of QNGD in finding the ground state energy of the given 2-qubit Hamiltonian. This example showcases the potential of quantum optimization algorithms for solving more complex problems

in quantum chemistry and condensed matter physics. The convergence behavior of QNGD suggests that the algorithm is well-suited for tackling larger and more complex quantum systems, where classical methods may become computationally infeasible. Moreover, the use of the quantum natural gradient allows for faster convergence, as it accounts for the unique geometry of the parameter space of the VQC.

Further studies can explore the performance of QNGD on a wider variety of quantum systems, as well as its applicability to more advanced variational ansatzes, such as those tailored to specific problem domains. Additionally, comparisons with other quantum optimization algorithms, such as the vanilla gradient descent and the Quantum Approximate Optimization Algorithm (QAOA), can provide valuable insights into the relative strengths and weaknesses of each method, guiding the development of more effective quantum algorithms for solving complex optimization problems in the future.

6 Conclusion

In this paper, we have presented a comprehensive study on Quantum Natural Gradient Descent (QNGD), an optimization algorithm that leverages the unique geometry of the parameter space of variational quantum circuits (VQCs). By using the quantum metric tensor, QNGD is able to account for the curvature of the parameter space, leading to faster convergence and potentially better solutions compared to classical gradient descent methods.

We have discussed the mathematical formulation of QNGD and provided a detailed, step-by-step implementation methodology. This included the representation of the optimization problem using a VQC, initialization, quantum circuit implementation, quantum gradient estimation, quantum metric tensor estimation, quantum natural gradient update, and convergence check.

As a case study, we applied QNGD to find the ground state energy of a simple 2-qubit Hamiltonian, given by the tensor product of two Pauli-Z matrices. Through this example, we demonstrated the effectiveness of QNGD in solving quantum optimization problems. The algorithm successfully converged to a minimum value close to the ground state energy, and the optimized state was found to be close to the actual ground state.

In conclusion, QNGD offers a promising approach for tackling challenging quantum optimization problems, particularly in the fields of quantum chemistry and condensed matter physics. Its use of the quantum natural gradient can lead to improved convergence and more accurate solutions, making it a valuable tool for the development of quantum algorithms and the study of quantum systems. Future research can explore the performance of QNGD on more complex problems and in combination with other quantum optimization methods to advance our understanding of quantum optimization techniques and their potential applications.

7 Bibliography

[1] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph].

[2] Iordanis Kerenidis and Anupam Prakash. "Quantum gradient descent for linear systems and least squares". In: *Physical Review A* 101.2 (2020). DOI: 10.1103/physreva.101.022316. URL: https://doi.org/10.1103%2Fphysreva.101.022316.

[3] Jarrod R McClean et al. "The theory of variational hybrid quantum-classical algorithms". In: *New Journal of Physics* 18.2 (2016), p. 023023. DOI: 10.1088/1367-2630/18/2/023023. URL: https://doi.org/10.1088%2F1367-2630%2F18%2F2%2F023023.

[4] K. Mitarai et al. "Quantum circuit learning". In: *Physical Review A* 98.3 (2018). DOI: 10.1103/physreva.98.032309. URL: https://doi.org/10.1103%2Fphysreva.98.032309.