Challenges and Future Prospects of Quantum Linear System Algorithm in Quantum Machine Learning

Jingwen Yu

College of Business, Lehigh University, Lehigh, 18015, USA

jiy922@lehigh.edu

Abstract. Algorithms designed for quantum computing, such as Harrow-Hassidim-Lloyd (HHL), have shown significant potential in solving linear equations, with the possibility of achieving exponential acceleration. However, there are still several areas that need to be further developed and improved. One of the critical challenges is converting classical data into quantum data, a process that is integral to the functioning of quantum computing algorithms. This transformation is often a computationally expensive process, which, if not addressed, could significantly limit the efficiency of quantum computing. Another important area requiring improvement involves the precision of quantum phase estimation (QPE) and amplitude amplification. These processes are vital for the successful execution of quantum algorithms, yet maintaining a high level of accuracy in a quantum environment remains a complex issue. Also, the concept of Quantum Random Access Memory (QRAM) for temporary data storage poses another challenge. While QRAM can theoretically enable efficient access to quantum data, it may lead to the failure of exponential acceleration in certain circumstances, posing a significant limitation to quantum computing algorithms. It is worth noting that the Harrow-Hassidim-Lloyd algorithm's output does not correspond to the solutions found through classical computations. This discrepancy represents another hurdle in utilizing the HHL algorithm for practical applications. Despite these challenges, Quantum Linear Systems Algorithms (QLSA), an emerging field in quantum computing, shows promise. It's still in the developmental stages but is already being tested and implemented in certain fields like machine learning and optimization. Despite the limitations of the current quantum computing systems, their potential to revolutionize these fields is indeed exciting.

Keywords: Quantum Computing Algorithms, Harrow-Hassidim-Lloyd, Quantum Random Access Memory, Quantum Linear Systems Algorithms

1 Introduction

Solving systems of linear equations is a standard problem across mathematics, science, and engineering. The classical methods used to solve these systems fall broadly into two categories: direct and iterative methods [1]. The computational complexity of

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_33

these methods often depends on the types of problems they address and the dimensionality of the variables involved. A comparison of these algorithms provides insight into their complexity. As shown in Table 1.

However, many modern computational tasks involve linear systems of extraordinary scale, with complexities within the order of 10^6. Classic algorithms such as lower-upper (LU) decomposition and Cholesky decomposition can be employed to tackle these problems on classical computers, but the computational burden remains substantial, leading to significant costs [2].

Table 1. Computational Complexity of Various Direct and Iterative Methods for Solving Linear Equations

Direct Methods	Gaussian Elimination	$O(n^3)$
	LU Decomposition	$O(n^2)$
	Cholesky Decomposition	$O(n^2)$
Iterative Methods	Conjugate Gradient	O(n)
	Jacobi Method	O(n)

Physicist Richard Feynman hypothesized that classical computers would struggle to simulate the quantum states encountered in physics, which include properties such as superposition and entanglement. To further the development of physics and enable the simulation of genuine superposition, entanglement, and their inherent randomness, Feynman proposed the development of a machine grounded in quantum mechanics theory. This proposition gave rise to new computational paradigms and injected new life into the field of computing [3]. In 1994, Peter Shor of Bell Laboratories published a quantum algorithm for factoring integers which could run in time $O((\log N)^3)$ [3]. However, its counterpart in classic, the general number field sieve, runs in time $exp\,(O((\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}))$, the significant improvement in computing speed would invalid the security of Rivest–Shamir–Adleman (RSA) cryptography, one of the widely used public-key cryptosystem for secure data transmission. In 2009, Harrow, Hassidim and Lloyd published their HHL algorithm, for solving the linear system problem on quantum machine, the problem could generally represented by $Ax = b$.

It provides a more efficient mode for many traditional algorithms that involve solving linear equations and require a lot of computation, such as optimization, linear differential equations and machine learning. The superposition state and entanglement of quantum make them react immediately no matter how far away they are. Using this property, the qubits can represent 0 or 1 or 0 and 1 at the same time, and more complex superposition states can be created through the entanglement properties, by then the qubits would become interlinked, and no matter where they are, if one qubit change the state, the correspond qubit would be affected at once. the calculation speed of Quantum computer will greatly surpass that of classical computer under specific rules [4].

For quantum computing, the key points are quantum gates, quantum circuits, and algorithms. Quantum gates are basic operations that act on a small number of quantum bits, usually represented by a 2n * 2n Unitary matrix. For example, the Hadamard gate is used to create superposition states, and controlled-NOT (CNOT) gate can be used to create quantum entanglement. Quantum computing is to use the circuit composed of these quantum gates to implement specific algorithms. For example, the Shor algorithm can be used for factorization of large prime numbers, while the Grover algorithm can be used for fast database retrieval, and the HHL algorithm for solving linear systems of equations [5].

2 Relevant Theories

To solve quantum linear system equation $A|x\rangle = |b\rangle$, where A is Hermitian matrix and b is a known vector. HHL is the core algorithm. Although the practice is not yet perfect, it has been tried in many algorithms [6]. The HHL algorithm consists of the following steps: Quantum state transformation of data, controlled rotation, and result output, where A is Hermitian matrix and $|b\rangle$ is a known vector. A must be a Hermitian matrix in order to convert it into a unitary operator. If A is not a Hermitian matrix, the original problem can be transformed into:

$$\begin{bmatrix} 0 & A \\ A^{\dagger} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \quad (1)$$

It uses QPE to estimate the eigenvalues of A, and finally uses quantum Fourier transform to extract the required eigenvalues. It looks like performing eigenvalue decomposition on A. When A is not a Hermitian matrix, a Hermitian matrix can be constructed.

2.1 Overview of HHL algorithm

To solve the system of linear equations is given by $A|x\rangle = |b\rangle$, $|b\rangle = {}_{i=1}^{N} b_i|u_i\rangle$, apply matrix exponential e^{iAt} to A, where t = , the Hermitian matrix $A = {}_{i=1}^{N} \lambda_i|u_i\rangle\langle u_i|$, where $|u_i\rangle$ is the i^{th} eigenvector of A with eigenvalue λ_i, and would create unitary operator e^{iAt}, whose eigenvalue and eigenvector are the same as matrix A. Then $|b\rangle$ should be projected to A: $|b\rangle = {}_{i=1}^{N} \beta_j|u_j\rangle$.

However, at this moment, the eigenvalue and eigenvector of matrix A is entangled state: ${}_{i=1}^{N} \beta_i|\lambda_i\rangle|u_i\rangle$, Apply inverse quantum Fourier transform after this to extract eigenvalue to the amplitude. The goal is to get $|x\rangle = A^{-1}|b\rangle = {}_{1}^{N} \lambda_i^{-1} b_i|u_i\rangle$.

After QPE, a set of controlled rotations are applied on an ancillary qubit, the angel is $\theta = arccos\left(\frac{1}{C}\right)$, and it effectively creates a state proportional to the solution vector |x⟩, but only if the right phase in the following step be measured [7]. a controlled rotation would be applied to extract the eigenvalue λ_i: $\theta_i \approx 2\,arcsin\left(\frac{C}{\lambda_j}\right)$, the operation gate is $\begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix}$,where $\theta = arccos\left(\frac{C}{\lambda_i}\right)$, then it come out with $\sqrt{1 - \frac{C^2}{\lambda_i^2}}|0\rangle + \frac{C}{\lambda_i}|1\rangle, \lambda_i \geq C$, where C is a certain constant. Now perform the inverse operation on all the above steps, including the inverse unitary operator and quantum Fourier transform to obtain an approximate value of $x \approx A^{-1}|b\rangle$.

2.2 Overview of Quantum Random Access Memory (QRAM)

Classic Random Access Memory (RAM) is divided into DRAM (Dynamic RAM) and SRAM (Static RAM). In DRAM, giving capacitor a high power frequency represents 1, while giving a capacitor a low power frequency represents 0. However, capacitors are prone to leakage and require constant refresh and charging to maintain information integrity. Therefore, DRAM is a type of memory that is prone to information loss, which means it requires a constant current to maintain stored information. Its structure is relatively simple and is usually used in computer memory modules. SRAM, on the other hand, has no capacitors and a more complex structure than DRAM. It can only store data when powered on, and disappears when powered off. It is usually used for central processing unit (CPU) caching [8].

Its storage principle can be compared to DRAM address allocation with a Binary tree structure. All data will be stored in the deepest layer of the Binary tree, that is, at the leaf node. The deeper the depth, the larger the storage space, and the more data can be stored. For example, for a Binary tree with a depth of 3, by drawing a Binary tree diagram, the left side is 0, the right side is 1, and its third address should be expressed as 010, and the seventh address should be expressed as 110.

QRAM is a device that has not yet been manufactured. Its purpose is to realize the function of temporary storage of data in classical RAM. However, in quantum computing, QRAM is used to store Quantum state. Enter the Quantum state of an address first, and RAM will return the address and the entangled state of the data stored at the address. The address can be the superposition state of multiple addresses. The significance of QRAM is that it is necessary for many quantum algorithms. For example, Quantum state need to be prepared in HHL, and the final results need to be stored in QRAM.

2.3 State Preparation in Quantum Systems

In the analysis of classical data, data preprocessing is a very important step. It is necessary to transform the data into a form that can be adapted by the model. Similarly, on a Quantum computer, classical data also needs to be converted into a Quantum state before calculation can be carried out. Moreover, whether the step of preparing quantum is efficient determines whether the acceleration of quantum computing is meaningful. It includes initial Quantum state preparation, Hamiltonian simulation, and Quantum Fourier transform.

The first step is to initialize the circuit with certain quantum states. There are two registers, the first register is the control register, prepare n $|0\rangle$ states as the initial configuration, and the second register is the target register, prepare the Unitary which already got, whose eigenvector is $|\varphi\rangle$, the $|\varphi\rangle$ here is just a mark, the QPE is to estimate it. After this, perform n Hadamard gates on each of the $|0\rangle$ qubits on the first register, convert them into superposition states. The next step is to perform controlled

Unit operations, the unitary matrix is the unitary operator, the QPE step is going to find its eigenvalue and its power's eigenvalue The power depends on the position of the control register. Afterwards, applying inverse Fourier transform, quantum Fourier transform separates information related to eigenvalues and makes them readable. Finally, the obtained phase is measured to obtain an estimate of the eigenvalues of the eigenvector $|\varphi\rangle$ of the unitary operator U. Quantum Fourier transform is an essential part of QPE, which simulates classical Fourier transform into quantum computation and acts on qubits. The quantum Fourier transform maps $|x\rangle = \sum_{i=0}^{N-1} x_i|i\rangle$ to $\sum_{i=0}^{N-1} y_i|i\rangle$ according to $y_k = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_n \omega_N^{nk}, k = 0,1,2,\ldots,N-1$, and its inverse is $x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_n \omega_N^{-nk}, k = 0,1,2,\ldots,N-1$, it also can be expressed as: $|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \omega_N^{xk}|k\rangle$.

3 Applications

3.1 Role of QLSA in Optimization Problems

Optimization problems are very common in today's society, such as finding the maximum return or the minimum error of return in financial investment Combinatorial optimization, or finding the optimal warehouse location and the optimal transportation route in commercial supply chain management. Their goal is to find the maximum or minimum value under given constraints. With the increasing demand and constraints, the calculation time is greatly extended, and the calculation cost also increases. However, their essence or the core part of calculation is to solve System of linear equations, but under many constraints, the amount and cost of calculation will be huge, and the calculation time will become long. However, with the help of Quantum computer, the computing speed will increase exponentially in theory, making it possible to solve the problem of large dimensions [9].

3.2 Application of QLSA in Quantum Machine Learning (QML)

In QSVM's classical counterpart, the problem could be mathematically formulated by [10]:

$$min\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t. y_i(w^T x_i - b) \geq 1 - \xi_i \xi_i \geq 0 \tag{2}$$
$$i = 1,2,\cdots,n$$

It is essentially a quadratic programming, so after preparing the Quantum state, you can use a Quantum computer to solve it. Quantum Principal Component Analysis (QPCA): In essence, classical principal component analysis is the calculation of matrix, calculating the covariance matrix of the matrix, and finding its eigenvalue. These are all Linear algebra problems, which can be theoretically done with the quantum linear system equation [11]. Quantum Linear Regression (QRegression):

Classical linear regression can be transformed into solving System of linear equations. Assuming that the original problem is that the factor data is A and the target data is b, then the problem can be transformed into

$$A^T A x = A^T b \tag{3}$$

4 Conclusion

QRAM is essential to quantum computing, and is still in the theoretical stage, regardless of the architectural model in use. For instance, the Bucket brigade architecture, while theoretically sound, is practically challenging to implement due to its deep circuitry. This complexity implies that the actual circuit would be intricate and possibly unmanageable [12].

Quantum states are inherently delicate and prone to instability, making them susceptible to errors and demanding in terms of environmental requirements. During actual computation, the register needs to be entangled with the newly generated temporary value. Complex computations entail a high degree of superposition entanglement, rendering the quantum state increasingly unstable. Consequently, extracting desired results and effectively accessing data becomes increasingly challenging.

Moreover, quantum computing inherently exhibits noise, further exacerbating the challenge of noise reduction. Although quantum computing can theoretically offer exponential acceleration, it hinges on the data being efficiently prepared into a quantum state. Practically, the preparation of a quantum state could demand substantial computational resources. If the quantum state preparation process is sluggish, the algorithm's exponential acceleration becomes insignificant. The HHL algorithm demands that matrix A be a sparse matrix, which is not always feasible in reality, even though a sparse matrix can be constructed. The HHL algorithm's complexity relates to the condition number of matrix A, used to determine whether the matrix is ill-conditioned. If the condition number is large, exponential acceleration is ineffective. The HHL algorithm's final output is not $|x\rangle$, but a quantum state of $\langle x|M|x\rangle$, where M is some operator. Therefore, additional steps are needed to extract and analyze the information in it. Furthermore, the accuracy of actual calculations is compromised by noise, which implies that accurate results cannot be guaranteed and extensive efforts may be needed to calibrate errors. In machine learning, nonlinear activation functions such as Rectified Linear Unit and sigmoid are crucial to neural network training. However, the quantum linear system algorithm solves linear equations, making the classical nonlinear activation functions unsuitable for quantum computing. Therefore, these functions necessitate further research to discover a version that fits quantum computing.

References

1. Duan, B., Yuan, J., Yu, C. H., et al. (2020). A survey on HHL algorithm: From theory to application in quantum machine learning. Physics Letters A, 384(24).
2. Heimann, D., Hohenfeld, H., Wiebe, F., & Kirchner, F. (2022). Quantum deep reinforcement learning for robot navigation tasks. arXiv preprint arXiv:2202.12180.
3. Duan, C. Y. (2020). A survey on HHL algorithm: From theory to application in quantum machine learning. Physics Letters, A, 384(24).
4. Changpeng, S., Yang, L. I., & Hongbo, L. I. (2019). Quantum Algorithm Design: Techniques and Applications. Journal of Systems Science & Complexity.
5. Dervovic, D., Herbster, M., Mountney, P., Severini, S., Usher, N., & Wossnig, L. (2018). Quantum linear systems algorithms: a primer. arXiv preprint arXiv:1802.08227.
6. Shao, C. (2018). Reconsider HHL algorithm and its related quantum machine learning algorithms.
7. Hou, X., He, X., Lv, C., et al. (2019). Quantum Linear Regression With Regularization. APS March Meeting 2019. American Physical Society.
8. Sajjan, M., Sureshbabu, S. H., & Kais, S. (2021). Quantum Machine-Learning for Eigenstate Filtration in Two-Dimensional Materials.
9. Rahimi, S. A., Kolahdoozi, M., Mitra, A., et al. (2022). Quantum-Inspired Interpretable AI-Empowered Decision Support System for Detection of Early-Stage Rheumatoid Arthritis in Primary Care Using Scarce Dataset. Mathematics, 10.
10. Yonghae, Lee, Jaewoo, et al. (2019). Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience. Scientific Reports.
11. Alata, R., Pauwels, J., Haelterman, M., et al. (2019). Phase Noise Robustness of a Coherent Spatially Parallel Optical Reservoir. IEEE Journal of Selected Topics in Quantum Electronics, PP(99), 1-1.
12. Giovannetti, V., Lloyd, S., & Maccone, L. (2008). Architectures for a quantum random access memory. Physical Review A, 78(5), 052310.