International Journal of Mathematics and Statistics Invention (IJMSI)
e-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 6 Issue 3 || March, 2018 || PP-75-80

Linear Algebra in Quantum Machine Learning

Jag Pratap Singh
Professor, Govt. Degree College Nadha Bhoor Sahaswan, Badaun

Abstract:
Linear algebra forms the mathematical backbone of quantum machine learning (QML). Quantum states, operators, tensor-product Hilbert spaces, eigen¬value decompositions, singular value transformations, and matrix exponentials are all linear-algebraic constructs that underlie quantum algorithms for data analysis, optimisation and learning. This paper presents a rigorous mathematical perspective on how key linear algebra concepts—vector spaces over C, Hermitian and unitary matrices, spectral decompositions, tensor products, and block-encodings—enable quantum machine learning models and algorithms. We discuss how quantum linear-algebra subroutines such as the HHL algorithm for solving linear systems, the Quantum singular value transformation (QSVT) framework, and amplitude-encoding of classical data rely fundamentally on linear algebra. We then examine applications in supervised and unsupervised QML—such as kernel methods, clustering and generative models—where linear algebraic tools accelerate or enable novel algorithms. Finally, we identify the mathematical and practical challenges—such as condition numbers, encoding of classical data, and scalability—and propose future directions at the interface of linear algebra, quantum computing and machine learning.
Keywords: *linear algebra; quantum machine learning; Hermitian operators; unitary matrices; singular value decomposition; amplitude encoding; tensor products; HHL algorithm; QSVT.*

I. Introduction

Quantum Machine Learning (QML) stands at the intersection of quantum computation and artificial intelligence, aiming to leverage the principles of quantum mechanics to enhance computational learning capabilities. The mathematical framework unifying these domains is linear algebra, which provides the language for describing quantum states, transformations, and measurements. Every quantum algorithm, from quantum Fourier transform to variational circuits, fundamentally operates on vectors and matrices in complex vector spaces known as Hilbert spaces.

In classical machine learning, linear algebra enables operations such as matrix factorization, least-squares regression, singular value decomposition (SVD), and eigenvalue analysis. Similarly, in the quantum domain, states of qubits are represented as complex vectors in a 2^n-dimensional Hilbert space, and quantum gates correspond to unitary matrices that preserve vector norms and probabilities. Thus, quantum computation is, at its core, a sequence of linear algebraic transformations acting on complex vector spaces.

A quantum state $|\psi\rangle$of a single qubit is expressed as a linear combination (superposition) of basis states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \text{where } \alpha, \beta \in \mathbb{C}, \ |\alpha|^2 + |\beta|^2 = 1$$

For a system of $n$qubits, the joint state space is described by the tensor product:

$$\mathcal{H} = \mathcal{H}_2^{\otimes n},$$

yielding a 2^n-dimensional vector space. Linear algebra governs all operations on this space, including entanglement, measurement, and unitary evolution.

2.1 Linear Algebraic View of Quantum Computation

The state evolution of a closed quantum system is determined by a unitary operator U, such that:

$$|\psi'\rangle = U|\psi\rangle, \text{ where } U^\dagger U = I$$

This unitary evolution ensures the preservation of total probability (norm of the state vector). Quantum measurement, on the other hand, corresponds to projecting the state vector onto orthogonal subspaces using projection operatorsP_i, satisfying:

$$P_i P_j = \delta_{ij} P_i, \sum_i P_i = I$$

The probability of observing outcome $i$is then given by:

$$p_i = \langle\psi|P_i|\psi\rangle.$$

Each of these concepts—unitarity, orthogonality, projection, and normalization—arises directly from linear algebraic principles applied to complex inner product spaces.

2.2 Motivation: Why Linear Algebra is Central to QML

Quantum Machine Learning leverages quantum linear algebra to accelerate or reformulate classical algorithms. Classical data often live in real vector spaces $\mathbb{R}^n$; quantum algorithms map them into complex Hilbert spaces $\mathbb{C}^{2^n}$ via amplitude encoding:

$$|x\rangle = \frac{1}{\|x\|}\sum_{i=1}^{n} x_i \, |\mathrm{i}\rangle.$$

Once encoded, linear operations such as matrix-vector multiplication or inner products can be executed exponentially faster through quantum parallelism. For instance, the **Harrow–**Hassidim–Lloyd (HHL) algorithm solves a system $A\mathbf{x} = \mathbf{b}$ in time $O(\log\ n)$ under certain conditions, exploiting spectral decomposition:

$$A = \sum_j \lambda_j \, |u_j\rangle\langle u_j|$$

By applying controlled rotations based on eigenvalues λ_j, the quantum computer effectively computes the inverse A^{-1} in the eigenbasis, an operation that is classically $O(n^3)$ using Gaussian elimination.

Thus, linear algebra not only describes the structure of quantum systems but also enables algorithmic speedups in core machine learning tasks such as regression, classification, clustering, and principal component analysis.

2.3 Relationship between Quantum Mechanics and Linear Algebra

Quantum theory itself is a linear algebraic formalism. The superposition principle asserts that any linear combination of valid states is also a valid state. The measurement postulate defines probabilities using the Hermitian inner product, and the time evolution of quantum states is governed by the Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H|\psi(t)\rangle,$$

where H is a Hermitian operator representing the Hamiltonian. The exponential of H,

$$U(t) = e^{-iHt/\hbar},$$

is a unitary operator obtained via the matrix exponential, another core concept of linear algebra.

In Quantum Machine Learning, this same linear algebraic machinery is used to represent optimization landscapes (through Hermitian Hamiltonians), cost functions (as expectation values of operators), and gradient evolution (via parameterized unitaries).

2.4 Aim and Scope of This Paper

The purpose of this paper is to mathematically articulate how linear algebra supports the formulation and implementation of Quantum Machine Learning algorithms. The discussion will focus on:

1. the role of matrix algebra and vector spaces in quantum state representation,
2. linear transformations and eigendecomposition in quantum algorithms,
3. tensor algebra and entanglement in multi-qubit systems,
4. matrix inversion, SVD, and Hermitian operators in quantum linear system solvers, and
5. the limitations and open problems linking linear algebra to quantum learning scalability.

The next section presents the mathematical foundations of linear algebra most relevant to quantum computation, forming the base for the QML applications that follow.

Mathematical Foundations of Linear Algebra for Quantum Computation

Quantum computation operates entirely within the framework of linear algebra over complex vector spaces. Every concept in quantum mechanics—state representation, time evolution, entanglement, and measurement—can be expressed using vectors, matrices, and linear transformations. To understand Quantum Machine Learning (QML), one must first grasp how linear algebra defines the algebraic structure of quantum systems.

3.1 Vector Spaces and Inner Products

A quantum state is a vector in a complex vector space called a Hilbert space. For a single qubit, the space is $\mathcal{H}_2 = \mathrm{span}\{|0\rangle, |1\rangle\}$, where the basis vectors satisfy orthonormality:

$$\langle i \mid j\rangle = \delta_{ij}, i, j \in \{0,1\}.$$

A general qubit state can be expressed as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle, \text{with } \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1.$$

The inner product $\langle\phi \mid \psi\rangle$ defines the overlap between two states and determines the probability amplitude of transitioning from $|\phi\rangle$ to $|\psi\rangle$.

The squared magnitude of this inner product gives a measurable probability:

$$P_{\phi\to\psi} = |\langle\phi \mid \psi\rangle|^2.$$

Thus, all probabilistic behavior in quantum systems emerges from inner products—one of the cornerstones of linear algebra.

3.2 Linear Operators, Matrices, and Unitarity

Transformations on quantum states are represented by linear operators, typically expressed as matrices acting on vectors.

If $U$is a quantum gate, then

$$|\psi'\rangle = U|\psi\rangle.$$

To preserve the total probability, quantum operators must be unitary:

$$U^\dagger U = UU^\dagger = I,$$

where $U^\dagger$is the conjugate transpose (Hermitian adjoint) of U.

Unitary matrices preserve inner products and vector norms:

$$\langle U\phi | U\psi\rangle = \langle \phi | \psi\rangle.$$

Examples include the Hadamard gate $H$and Pauli matrices:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \sigma_y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}, \sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}.$$

Each of these operators represents a specific rotation or reflection in complex vector space and forms the building blocks of quantum circuits.

3.3 Hermitian Operators and Measurement

In quantum mechanics, observables—quantities that can be measured—are represented by Hermitian matrices, i.e.,

$$A = A^\dagger.$$

Hermitian operators have real eigenvalues and orthogonal eigenvectors:

$$A|a_i\rangle = \lambda_i|a_i\rangle, \lambda_i \in \mathbb{R}.$$

Measurement collapses a state $|\psi\rangle$into one of these eigenstates $|a_i\rangle$with probability

$$P(a_i) = |\langle a_i|\psi\rangle|^2,$$

and the expectation value of the observable is

$$\langle A\rangle = \langle\psi|A|\psi\rangle.$$

This linear algebraic formulation allows machine learning algorithms on quantum computers to encode loss functions, kernels, and cost observables as Hermitian matrices, linking learning objectives directly to measurable quantities.

3.4 Tensor Products and Multi-Qubit Systems

For systems with multiple qubits, the joint Hilbert space is constructed via the tensor product:

$$\mathcal{H} = \mathcal{H}_2^{\otimes n}.$$

For example, for two qubits:

$$|\psi_{12}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle.$$

If $|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$and $|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, then

$$|\psi_{12}\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle.$$

This tensor structure enables entanglement, where the combined state cannot be written as a product of individual states. Entanglement, represented algebraically through tensor correlations, is a key source of quantum parallelism and the foundation of many QML models.

3.5 Eigenvalue Decomposition and Spectral Theorems

Many quantum algorithms, including those for machine learning, depend on eigenvalue decomposition of Hermitian matrices.

For a Hermitian matrix H:

$$H = \sum_j \lambda_j |u_j\rangle\langle u_j|,$$

where $\{|u_j\rangle\}$form an orthonormal basis of eigenvectors.

The spectral theorem ensures that any Hermitian operator is diagonalizable by a unitary transformation:

$$H = U\Lambda U^\dagger,$$

where $\Lambda$is a diagonal matrix of real eigenvalues.

This property allows quantum computers to simulate the time evolution operator $U(t) = e^{-iHt}$and enables subroutines such as quantum phase estimation and the HHL algorithm, both of which form the mathematical basis of QML linear-algebraic computations.

3.6 Singular Value Decomposition (SVD) and Quantum Analogues

The singular value decomposition (SVD) is another central tool in both classical and quantum linear algebra. For any matrix $A \in \mathbb{C}^{m\times n}$:

$$A = U\Sigma V^{\dagger},$$

where $U$and $V$are unitary and $\Sigma$is a diagonal matrix with non-negative real entries (singular values).

Quantum algorithms leverage SVD implicitly via Quantum Singular Value Transformation (QSVT), which enables the application of polynomial functions to singular values encoded in a block-unitary representation. This is foundational for quantum principal component analysis (qPCA) and quantum recommendation systems, where eigenvalue spectra encode correlation or covariance structures of datasets.

3.7 Linear System Solving and Matrix Inversion

The Harrow–Hassidim–Lloyd (HHL) algorithm exemplifies how quantum computers exploit linear algebra for exponential speedups. Given a system $A\mathbf{x} = \mathbf{b}$, where $A$is Hermitian and well-conditioned, the solution vector is encoded in a quantum state:

$$|x\rangle \propto A^{-1}|\mathbf{b}\rangle = \sum_j \frac{\langle u_j|\boldsymbol{b}\rangle}{\lambda_j}|u_j\rangle,$$

where $A|u_j\rangle = \lambda_j|u_j\rangle$.

This operation effectively computes the inverse of Ain the eigenbasis, a task exponentially faster than classical matrix inversion under appropriate conditions.

Linear Algebra Applications in Quantum Machine Learning

Linear algebra serves not only as the theoretical foundation of quantum mechanics but also as the computational machinery driving quantum machine learning (QML). Most QML algorithms reformulate classical linear-algebraic problems—matrix inversion, eigenvalue decomposition, inner-product computation—into quantum-parallel linear transformations. This section highlights the main applications of linear algebra in QML: quantum data encoding, kernel estimation, principal-component analysis, and variational optimization.

A central step in QML is data encoding, where classical data vectors are mapped to quantum states in a Hilbert space. Using amplitude encoding, a normalized feature vector $x \in \mathbb{R}^n$is represented as

$$|\, x\rangle = \frac{1}{\|\mathrm{x}\|}\sum_{i=1}^{n} x_i \,|\, i\rangle,$$

thereby embedding classical information into a quantum vector space. This allows linear-algebraic operations such as dot products and norm computation to be performed through quantum interference: the inner product between two states $|\, x\rangle, |\, y\rangle$equals

$$\langle x|y\rangle = \frac{1}{\|x\|\|y\|}\sum_i x_i^* y_i\,,$$

which quantum hardware can evaluate exponentially faster through amplitude estimation.

In quantum kernel methods, linear algebra defines the similarity between data points via Hermitian positive-semidefinite matrices. A kernel element is expressed as $K_{ij} = \langle\phi(x_i)|\phi(x_j)\rangle$, where $\phi(x)$denotes the quantum feature map. The resulting kernel matrix Kcaptures the geometry of data in the quantum feature space, and training a support-vector classifier becomes an optimization over a quadratic form

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\alpha^{\top}K\alpha,$$

with Kcomputed from quantum circuits. Here, matrix operations—multiplication, eigenvalue computation, and inversion—directly emerge from linear-algebraic principles.

Quantum principal-component analysis (qPCA) is another clear example of linear algebra in QML. Given a density matrix $\rho = \frac{1}{N}\sum_{i=1}^{N}|x_i\rangle\langle x_i|$, qPCA estimates its eigenvalues and eigenvectors through quantum phase estimation applied to $e^{-i\rho t}$. The spectral decomposition

$$\rho = \sum_j \lambda_j|u_j\rangle\langle u_j|$$

yields principal components $|u_j\rangle$and variances λ_j, analogous to classical PCA but obtained exponentially faster in the data dimension.

Variational quantum circuits used for supervised and unsupervised learning rely on matrix optimization in parameterized unitaries$U(\theta)$. Expectation values

$$\langle O\rangle_{\theta} = \langle\psi_0|U^{\dagger}(\theta)OU(\theta)|\psi_0\rangle$$

define differentiable cost functions whose gradients are computed using the parameter-shift rule, itself a consequence of linear-algebraic commutation relations between Hermitian generators. These constructs link linear algebra, optimization, and probabilistic inference within a unified quantum framework.

Overall, every stage of QML—from encoding and kernel construction to dimensionality reduction and variational learning—relies on linear algebraic formulations of quantum states, operators, and measurements. In practice, quantum linear-algebra subroutines such as the HHL solver and quantum singular-value transformation act as universal primitives that implement core machine-learning computations on quantum hardware.

Challenges, Open Problems, and Future Directions

Despite its mathematical elegance, integrating linear algebra into quantum machine learning (QML) faces significant theoretical and practical challenges. A major difficulty lies in data loading and state preparation: encoding a classical vector $x \in \mathbb{R}^n$into a quantum state $| x\rangle$generally requires $O(n)$operations, offsetting potential exponential speedups. Moreover, quantum linear-algebra algorithms such as HHL depend on matrices that are Hermitian, sparse, and well-conditioned—assumptions rarely satisfied by real-world data. The condition number $\kappa(A)$of the system matrix directly influences the runtime $O(\kappa^2 \log n)$, revealing the sensitivity of quantum solvers to numerical stability. Another challenge concerns noise and decoherence in near-term quantum devices, which corrupt matrix transformations $U | \psi\rangle$and degrade the precision of inner-product and eigenvalue estimation. From a mathematical standpoint, developing robust quantum analogues of classical numerical linear-algebra methods—such as iterative solvers and low-rank approximations—remains an open problem.

Future research aims to create hybrid quantum-classical frameworks where linear-algebraic computations are distributed between quantum processors (handling large-scale linear transformations) and classical optimizers (refining parameters). The emergence of tensor-network simulations, quantum-inspired matrix factorizations, and block-encoding techniques indicates progress toward this goal. Additionally, new formulations of quantum singular-value transformation and quantum gradient descent promise general linear-operator learning frameworks. In the long term, the fusion of linear algebra, quantum information theory, and optimization geometry will determine how efficiently quantum computers can learn and represent high-dimensional data—transforming both the mathematical and computational foundations of machine learning.

Conclusion

Linear algebra constitutes the mathematical core of quantum computation and, consequently, of Quantum Machine Learning (QML). Every operation in a quantum algorithm—from state preparation and entanglement to measurement and optimization—can be expressed as a linear transformation on complex vector spaces. The superposition and interference principles arise from vector addition, unitary operators describe reversible linear transformations, and Hermitian matrices encode observable quantities with real eigenvalues.

Within QML, these linear algebraic constructs provide the basis for encoding classical data into quantum states, computing kernel functions, decomposing density matrices in quantum principal-component analysis, and optimizing parameterized quantum circuits. Algorithms such as HHL and Quantum Singular Value Transformation show how linear algebra not only structures quantum mechanics but also drives algorithmic acceleration in solving large-scale systems.

However, challenges remain in realizing these theoretical advantages due to hardware noise, data-loading bottlenecks, and limitations on matrix sparsity and conditioning. Despite these constraints, ongoing research into hybrid algorithms, block-encoding, and tensor-network methods promises to merge classical numerical stability with quantum parallelism. Ultimately, the marriage of linear algebra and quantum mechanics will continue to shape the mathematical foundation for the next generation of intelligent quantum systems—transforming how machines represent, process, and learn from information.

References

[1]. Aaronson, S. (2015). *Quantum computing since Democritus.* Cambridge University Press.
[2]. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature, 549*(7671), 195–202.
[3]. Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters, 103*(15), 150502.
[4]. Lloyd, S., Mohseni, M., & Rebentrost, P. (2014). Quantum principal component analysis. *Nature Physics, 10*(9), 631–633.
[5]. Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information* (10th anniversary ed.). Cambridge University Press.
[6]. Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum, 2*, 79.
[7]. Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical Review Letters, 113*(13), 130503.
[8]. Tang, E. (2019). A quantum-inspired classical algorithm for recommendation systems. *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*, 217–228.

[9]. Wang, G., Li, Y., Song, Z., & Wang, X. (2017). Quantum singular value transformation and its applications. *Quantum Information Processing, 16*(9), 223.

[10]. Wiebe, N., Kapoor, A., &Svore, K. M. (2014). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information & Computation, 15*(3–4), 316–356.