MODULE 2: MATHEMATICAL FOUNDATIONS

Learning Outcomes

2.1

Linear Algebra Refresher (Vectors, Matrices)

2.2

Complex Numbers and Dirac Notation

2.3

Tensor Products and State Spaces

2.4

Matrix Operations for Quantum Computing

Linear Algebra Refresher (Vectors, Matrices)

Some familiarity with linear algebra is essential to understand quantum computing. This article introduces the basic concepts of linear algebra and how to work with **vectors and matrices** in quantum computing.

Vectors:

- A vector is a mathematical object with magnitude and direction, represented as an ordered list of numbers (components) in a chosen basis.
- In quantum computing, vectors represent quantum states in a complex vector space (Hilbert space).

Notation: A vector $\vec{v}$ in $\mathbb{C}^n$ is written as $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$, where $v_i \in \mathbb{C}$.

Linear Algebra Refresher (Vectors, Matrices)

Orthonormal Basis:

An orthonormal basis is a set of vectors in a vector space that are both orthogonal (mutually perpendicular) and normalized (unit length).

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$\langle 0|1\rangle = 0$ → Orthogonal

$\langle 0|0\rangle = \langle 1|1\rangle = 1$ → Normalized

Vector Operations:

1. Addition: Adds corresponding components of two vectors.

$$\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \quad \Rightarrow \vec{a} + \vec{b} = \begin{bmatrix} 1+3 \\ 2+4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

- In robotics, if two movement commands are represented as vectors (e.g., move forward and turn right), their **vector addition** gives the resulting direction and speed.

Linear Algebra Refresher (Vectors, Matrices)

2. Scalar multiplication: Multiplies each component of a vector by a scalar.

$$k = 2, \quad \vec{a} = \begin{bmatrix} 3 \\ -1 \end{bmatrix} \Rightarrow k\vec{a} = \begin{bmatrix} 2 * 3 \\ 2 * (-1) \end{bmatrix} = \begin{bmatrix} 6 \\ -2 \end{bmatrix}$$

- In graphics rendering, scaling a shape (like zooming in) uses **scalar multiplication** on its position vectors.

3. Scalar Product/Inner product/Dot Product: Sum of the products of corresponding components, with the first vector's components conjugated.

$$\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad \vec{a} \cdot \vec{b} = 1 * 3 + 2 * 4 = 11$$

- In quantum computing, the **dot product** is used to calculate probability amplitudes during qubit state measurements.

Linear Algebra Refresher (Vectors, Matrices)

4. Norm: The length (magnitude) of a vector.

$$\vec{a} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad \|\vec{a}\| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$$

- In quantum mechanics, the norm of a quantum state vector must be 1 to represent a valid probability distribution.

5. Normalization: Adjusting a vector so that its norm is 1.

$$\vec{a} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad \text{Norm} = 5, \quad \text{Normalized } \vec{a} = \begin{bmatrix} 3/5 \\ 4/5 \end{bmatrix}$$

- In machine learning, feature vectors are normalized to ensure that all features contribute equally to distance-based algorithms (like k-NN or SVM).

Linear Algebra Refresher (Vectors, Matrices)

Addition

$$\vec{v} + \vec{w} = \begin{bmatrix} v_1 + w_1 \\ v_2 + w_2 \\ \vdots \\ v_n + w_n \end{bmatrix}$$

Norm

$$\|\vec{v}\| = \sqrt{\langle \vec{v}, \vec{v} \rangle}$$

Scalar Multiplication

$$c\vec{v} = \begin{bmatrix} cv_1 \\ cv_2 \\ \vdots \\ cv_n \end{bmatrix},$$

where $c \in \mathbb{C}$

Scalar Product

$$\langle \vec{v}, \vec{w} \rangle = v_1^* w_1 + v_2^* w_2 + \cdots + v_n^* w_n$$

Normalization

$$\vec{v}_{\text{normalized}} = \frac{\vec{v}}{\|\vec{v}\|}$$

Linear Algebra Refresher (Vectors, Matrices)

Matrices:

A matrix is a 2D array of numbers (often complex) used to represent quantum gates and linear transformations. A matrix of size m×n is a collection of m · n complex numbers arranged in m rows and n columns as shown below:

$$M = \begin{bmatrix} M_{11} M_{12} \cdots M_{1n} \\ M_{21} M_{22} \cdots M_{2n} \\ \ddots \\ M_{m1} M_{m2} \cdots M_{mn} \end{bmatrix}$$

Quantum Gates as Matrices:

Quantum gates are represented by unitary matrices, which act on qubit state vectors.

Example – Pauli-X Gate: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow X|0\rangle = |1\rangle$

Linear Algebra Refresher (Vectors, Matrices)

Matrix Operations:

1. Addition: $A+B$, element-wise addition of matrices of the same size.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \Rightarrow A + B = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

- In image processing, pixel matrices from two images can be added to create blending effects.

2. Scalar multiplication: cA, multiply each element by scalar c.

$$c = 2, \quad A = \begin{bmatrix} 1 & -1 \\ 3 & 4 \end{bmatrix} \Rightarrow cA = \begin{bmatrix} 2 & -2 \\ 6 & 8 \end{bmatrix}$$

- In audio signal processing, scaling amplitude matrices controls volume.

Linear Algebra Refresher (Vectors, Matrices)

3. Matrix multiplication: For an $m \times n$ matrix A and an $n \times p$ matrix B, the product AB is an $m \times p$ matrix with entries.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \Rightarrow AB = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

- In computer graphics, transformation matrices rotate, scale, or translate objects.

4. Transpose: A^T, Swap rows and Columns.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

- In data science, transposing helps align datasets for operations like dot products.

Linear Algebra Refresher (Vectors, Matrices)

5. Conjugate transpose (Hermitian Conjugate): $A^\dagger=(A^*)^T$, where A^* is the element-wise complex conjugate.

$$A = \begin{bmatrix} 1+i & 2 \\ -i & 3 \end{bmatrix} \Rightarrow A^\dagger = \begin{bmatrix} 1-i & i \\ 2 & 3 \end{bmatrix}$$

- Used in quantum computing to reverse gate operations or check unitarity of matrices.

6. Identity matrix: I, Diagonal matrix with 1s on the diagonal.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

In linear algebra, it behaves like 1 in multiplication.

$A.I=I.A=A$ In quantum mechanics, it represents "do nothing" operations on qubits.

Linear Algebra Refresher (Vectors, Matrices)

Inverse: For a square matrix A, A^{-1} satisfies $AA^{-1} = I$, If it exists.

$$A = \begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}, \quad A^{-1} = \frac{1}{10}\begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix}$$

- Used in control systems and machine learning (e.g., solving linear equations, normal equations in regression).

Special Matrices:

In quantum computing, special matrices play a central role in representing quantum gates, quantum states, and operations. These matrices follow specific mathematical rules essential to preserve the nature of quantum systems.

- Hermitian Matrix
- Unitary Matrix

Linear Algebra Refresher (Vectors, Matrices)

1. Hermitian Matrix:

$$A = \begin{bmatrix} 2 & i \\ -i & 3 \end{bmatrix}, \quad A^{\dagger} = A$$

- Represents observable quantities like energy or spin in quantum mechanics. Measurements yield real eigenvalues.

2.Unitary: A † A=I. Represents quantum gates, preserving the norm of quantum states.

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad U^{\dagger}U = I$$

- All quantum gates (e.g., Hadamard, Pauli-X) are unitary. They preserve probability (norm) of quantum states.

Complex Numbers and Dirac Notation

Complex Nmbers:

Complex numbers are the combination of Real part and the imaginary part. These numbers are essential in quantum computing because quantum states have complex amplitudes, allowing for interference and superposition.

- A complex number is of the form of

$$z = a + bi \quad \text{where } a, b \in \mathbb{R} \text{ and } i = \sqrt{-1}$$

Polar Form:

Polar form is a way of expressing a complex number using its magnitude and angle (phase) rather than its rectangular (Cartesian) coordinates.

$$z = re^{i\theta}$$

Where:

- $r = |z| = \sqrt{a^2 + b^2}$ → magnitude (modulus)
- $\theta = \tan^{-1}(b/a)$ → angle (argument or phase)

Complex Numbers and Dirac Notation

Operations:

Addition: Adds two quantum state amplitudes component-wise. Used when combining quantum states.

$$(a+bi)+(c+di)=(a+c)+(b+d)i$$

Multiplication: Used when applying complex coefficients or interference calculations in quantum superposition.

$$(a+bi)(c+di)=(ac-bd)+(ad+bc)i$$

Conjugate: Essential for computing inner products and probabilities.

$$z^*=a-bi$$

Magnitude (Modulus): The square of the magnitude gives the probability of measuring a particular quantum state.

$$|z|=\sqrt{z^*z}=\sqrt{a^2+b^2}$$

Complex Numbers and Dirac Notation

Dirac Notation:

The row and column vectors are represented in a notation system special to quantum mechanics called **Dirac Notation.** These notations are used to describe quantum states and operations in concise way.

Ket: A quantum state vector, denoted as

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix}$$

Bra: The conjugate transpose of a ket

$$\langle\psi| = [\psi_1^* \ \psi_2^* \ \cdots]$$

Inner Product: Gives a complex number representing the overlap between quantum states.

$$\langle\phi|\psi\rangle = \sum \phi_i^* \psi_i$$

Complex Numbers and Dirac Notation

Outer Product: Forms a matrix used to define projection operators.

$$|\psi\rangle\langle\phi|$$

Normalization: A valid quantum state must satisfy:

$$\langle\psi|\psi\rangle = 1$$

Basis States: A Standard qubit basis:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

Superposition: A general qubit state:

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

Tensor Products and State Spaces

Tensor Product:

The tensor product combines vector spaces to describe composite quantum systems (e.g., multiple qubits).

- For vectors $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$ and $|\phi\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$, the Tensor product is $|\psi\rangle \otimes |\phi\rangle = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}$
- For matrices $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and B, the tensor product $A \otimes B$ is:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}$$

Properties:

Dimension of the combined space is the product of individual dimensions (e.g., two qubits: 2×2=4-dimensional space).

Not commutative: $A \otimes B \neq B \otimes A$

Tensor Products and State Spaces

State Spaces:

The state space of a quantum system is a Hilbert space, a complete complex vector space with an inner product.

- For n qubits, the state space $\mathbb{C}^{2^n}$, with basis states like $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, (for 2 qubits).
- A general n-qubit state is

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle, \text{ where } \sum |c_i|^2 = 1.$$

- Entangled states: States like $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ cannot be written as a tensor product of individual qubit states.

Matrix Operations for Quantum Computing

Matrix Operations:

Matrix operations are essential tools in quantum computing because they provide a mathematical framework to describe and manipulate quantum states and gates. Quantum systems are represented using vectors and matrices from linear algebra, particularly involving complex numbers and Hilbert spaces.

Unitary operations:

Quantum gates are represented by unitary matrices **($U^\dagger U = I$).**

Examples:

Pauli Matrices: Represent fundamental single-qubit quantum gate operations.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Matrix Operations for Quantum Computing

Hadamard gate: Creates superposition from a basis state.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

CNOT Gate (Controlled-NOT):

Flips the second qubit (target) if the first qubit (control) is 1.

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Measurement operators:

Measurements are represented by Hermitian operators (observables).

A measurement projects a state onto the eigenstates of the observable, with probabilities given by the Born rule:

For state $|\psi\rangle$ and projector $P_i = |\phi_i\rangle\langle\phi_i|$,

probability is $\langle\psi|P_i|\psi\rangle$.

Matrix Operations for Quantum Computing

Eigenvalues and eigenvectors:

- For a matrix ***A***, an eigenvector $\vec{v}$ satisfies $A\vec{v} = \lambda\vec{v}$, is the eigenvalue.
- In quantum computing, eigenvectors of observables correspond to possible measurement outcomes.

Trace and partial trace:

Trace: Sum of diagonal elements, Used to compute expectation values.

Partial trace: Used to trace out subsystems in a composite system, e.g., to analyze entanglement.

Thank you