Problem-2:Classification of Valid and Invalid Quantum States Using Neural Networks

Matreyee Kandpal

IISER Mohali

email: ph23022@iisermohali.ac.in

Introduction to Matrices

What is a Matrix?

A matrix is a rectangular array of numbers or symbols arranged in rows and columns. It is widely used in physics, engineering, and machine learning.

Properties of a 2×2 Matrix

Consider a general 2×2 matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{1}$$

- **Trace**: The trace of a matrix is the sum of its diagonal elements:

$$\mathrm{Tr}(A) = a + d \tag{2}$$

- **Transpose**: The transpose of A is obtained by swapping rows and columns:

$$A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \tag{3}$$

Properties of a 2×2 Matrix (contd.)

- **Complex Conjugate**: The complex conjugate of a matrix is obtained by taking the conjugate of each element:

$$A^* = \begin{bmatrix} a^* & b^* \\ c^* & d^* \end{bmatrix} \tag{4}$$

If A contains imaginary components, i is replaced with $-i$ in each element.

- **Determinant of a Matrix**: The determinant of A is given by:

$$\det(A) = ad - bc \tag{5}$$

The determinant gives important information about a matrix, such as whether it is invertible ($\det(A) \neq 0$) or singular ($\det(A) = 0$).

Properties of a 2×2 Matrix (contd.)

- **Hermitian**: A matrix is Hermitian if it is equal to its conjugate transpose:

$$A^\dagger = (A^*)^T = A \tag{6}$$

This means a, d must be real and $b = c^*$.

- **Eigenvalues of a Matrix:** The eigenvalues of A satisfy:

$$\det(A - \lambda I) = 0 \tag{7}$$

Expanding the determinant:

$$\begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = 0 \tag{8}$$

Which simplifies to the characteristic equation:

$$\lambda^2 - (a + d)\lambda + (ad - bc) = 0 \tag{9}$$

The solutions λ_1, λ_2 are the eigenvalues of A.

Density Matrix and Its Properties

Definition of Density Matrix

A **density matrix** ρ is a mathematical representation of a quantum state used to describe both pure and mixed states. It satisfies three key properties:

- **Hermitian:** $\rho = \rho^\dagger$
- **Positive Semi-definite:** $\lambda_i \geq 0$ (eigenvalues must be non-negative)
- **Trace Condition:** $\mathrm{Tr}(\rho) = 1$

These properties ensure a physically valid quantum state representation.

Examples of Valid Density Matrices

Pure State Example:

$$\rho = |\psi\rangle\langle\psi|$$

For $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$:

$$\rho = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Mixed State Example:

$$\rho = p_1|\psi_1\rangle\langle\psi_1| + p_2|\psi_2\rangle\langle\psi_2|$$

For a system in $|0\rangle$ with probability 0.7 and in $|1\rangle$ with probability 0.3:

$$\rho = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.3 \end{bmatrix}$$

Examples of Invalid Density Matrices

1. Trace Not Equal to 1:

$$\rho = \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}$$

Here, $\mathrm{Tr}(\rho) = 0.8 \neq 1$, making it invalid.

2. Negative Eigenvalues:

$$\rho = \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$$

One eigenvalue is negative, so it is not positive semi-definite.

3. Non-Hermitian Matrix:

$$\rho = \begin{bmatrix} 0.5 & i \\ -i & 0.5 \end{bmatrix}$$

Off-diagonal elements are not complex conjugates, violating Hermitian property.

Why is a Density Matrix representing a Quantum State Always Valid ?

A density matrix ρ representing a quantum state is always valid because:

- It is derived from a well-defined quantum state $|\psi\rangle$, ensuring Hermitian property.
- The trace is normalized by construction ($\mathrm{Tr}(\rho) = 1$).
- It is obtained from a probability distribution over valid quantum states, ensuring positive semi-definiteness.

Problem: Classification of Valid and Invalid Quantum States Using Neural Networks

Objective:

Train a neural network to classify density matrices as valid or invalid based on their properties.

- **Step 1: Data Generation:**
  Generate 10,000 random matrices of dimension 2×2 (5,000 Valid and 5,000 Invalid).
- **Step 2: Model Training:**
  Train the Fully Connected Neural Network using the generated data and evaluate performance using accuracy, precision, recall, and F1-score.
- **Step 3: Model Evaluation:**
  Generate an independent dataset of 1,000 random 2×2 matrices without labels and use the trained model to predict their validity.

Step 1: Data Generation

Generating Training Data

To train a machine learning model, we generate a balanced dataset of valid and invalid density matrices.

- **Valid Density Matrices:** Generate random Hermitian matrices of dimension 2×2 using NumPy.
- Normalize valid matrices to satisfy $\rho = \frac{AA^\dagger}{\mathrm{Tr}(AA^\dagger)}$.
- **Invalid Density Matrices:**
  - Introduce violations (It will work with any one, but you can choose to include one, two, or all three as per your preference):
    - Negative eigenvalues
    - Incorrect trace
    - Non-Hermitian structure
- Generate 10,000 matrices (5,000 valid, 5,000 invalid) and label them as 1 for valid and 0 for invalid.

Step 2: Model Training

Neural Network Architecture an Training

- Fully connected feedforward neural network (FCNN)
- **Input:** Flattened density matrix (vectorized representation)
- **Output:** Binary classification (valid or invalid).
- Split data into training and test sets.
- Train the model using labeled data.
- Evaluate performance using accuracy, precision, recall, and F1-score
- Adjust hyperparameters to optimize results

Practice Task: Dealing with Complex Entries

Handling Complex Density Matrices

- Quantum states often involve complex numbers: $a + ib$ where $a, b \in \mathbb{R}$
- Example:

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

- Conditions for validity:
  - Hermitian: $\rho = \rho^\dagger$
  - Positive semi-definite
  - Unit trace

Extending the Model for Complex Matrices

- Modify input representation to handle real and imaginary parts separately
- Adjust neural network layers to process complex-valued data
- Train and evaluate using newly generated complex density matrices

Feel free to contact if you have any doubts.