MDS572B: QUANTUM MACHINE LEARNING

OBSERVATION NOTEBOOK

Name: Neelanjan Dutta

Roll No: 2448040

Class: 5MDS

LAB No. 1: Installing Qiskit and Setting Up the Environment

Question:

Install Qiskit and do a small sample program of your choice.

Objective:

The objective of this lab was to set up a functional quantum computing development environment using **Qiskit**. This involved installing the necessary Python libraries, verifying the installation by checking the version, and confirming the environment's readiness by successfully creating, executing, and visualizing a fundamental one-qubit quantum circuit that demonstrates the action of a Pauli-X (NOT) gate.

Draft Plan:

Program description:

This program is designed to set up and verify the **Qiskit** development environment. The main objective is to ensure the **Qiskit library** is correctly installed and to run a simple introductory example: creating a circuit, applying a NOT gate (X-gate), and performing a measurement.

Program logic:

- **Installation**: Used the pip package manager to install the qiskit library.
- **Verification**: Verified the installation by importing the Qiskit library in Python. A successful import without errors indicated a correct setup.
- **Version Check**: Printed the installed version of Qiskit using **qiskit.__version__** to confirm the specific release.
- **Circuit Construction**: A QuantumCircuit was built with one quantum bit (qubit) and one classical bit for storing the measurement result.
- **Gate Application**: An X-gate was applied to the qubit at index 0. This flips the qubit's state from its initial state of **|0>** to **|1>**.
- **Measurement**: A measurement operation was added to map the final state of the qubit to the classical bit.
- **Visualization**: Used the display() function with qc.draw('mpl') to generate and explicitly render a visual representation of the circuit.

Program:

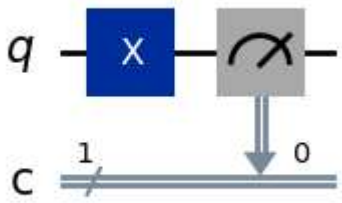
Installing Qiskit and checking its version:

```
!pip install qiskit
import qiskit
v= qiskit.__version__
print("The current version of qiskit is:" , v)
```

Sample Program:

```
from IPython.display import display
from qiskit import QuantumCircuit
# Create a simple quantum circuit
qc = QuantumCircuit(1, 1)   # 1 qubit, 1 classical bit
qc.x(0)                     # Apply NOT (X) gate
qc.measure(0, 0)            # Measure qubit
display(qc.draw('mpl')) # Explicitly display the drawing
```

Output:



Interpretation:

1. **Quantum Register (q):** This is the top line, representing your single quantum bit (qubit).
    - **X Gate**: The blue box labeled **'X'** on this line is the **Pauli-X gate**. It acts like a classical NOT gate, flipping the qubit's state from its initial state of **(spin up)** to **(spin down)**.

- **Measurement**: The grey box with the meter symbol is the **measurement operation**. It measures the final state of the qubit and sends that information down to a classical bit.

2. **Classical Register (c):** This is the bottom double line, representing a standard classical bit that can store either a **0** or a **1**. The '1' next to it indicates it's a classical register of size 1.
    - **Arrow**: The arrow from the measurement on the quantum register to the classical register shows that the result of the qubit measurement (will be **'1'** in this case, because of the **X gate**) is stored in this classical bit.

Test Cases:

Test Case 1: Environment Setup

- **Input:** Ran the installation and import commands.
- **Expected Output:** Successful installation message and no errors upon import. The correct version number is printed.

Test Case 2: Circuit Execution

- **Input:** Ran the sample program code.
- **Expected Output:** A matplotlib diagram showing a single qubit line (q) and a single classical bit line (c). An 'X' gate is shown on the qubit line, followed by a measurement operation connecting the qubit to the classical bit.

Conclusion:

In conclusion, the Qiskit environment was successfully installed and configured, as validated by the error-free execution of the setup commands and a sample quantum program. The lab successfully demonstrated the basic workflow of building a quantum circuit, applying a gate, and visualizing the architecture. This foundational exercise confirms that the system is prepared for developing and running more complex quantum algorithms in future experiments.

****