Chem560

Homework #1: State preparation, measurement and tomography

In this first problem set we will go over some of the important concepts involved with state preparation, measurement and tomography.

Submit everything as a single Jupyter notebook (even including non-computational questions).

1. *Native gates*

   (a) In class we introduced the native gates on the IonQ simulator. What are the native gates for another (non trapped ion) cloud-based quantum simulator (describe the gates and name the simulator)? If you have some understanding of the differences in the hardware, explain why the native gates are different.

   (b) Decompose a CNOT circuit into the IonQ's native gates (https://ionq.com/best-practices). Demonstrate that the two are equivalent. (Hint: We can decompose a CNOT circuit into RX, RY and Mølmer-Sørenson gates.)

   Currently, due to the high cost of quantum computing, users typically have little or no control over any of the transpilation/compilation that will occur. Compilation of quantum algorithms is an active area of development and research (even including its complexity!).

2. *Qiskit measurement warm up*: Prepare each qubit state either using `initialize`, or `reset` followed by the appropriate single qubit gate function. Use the QasmSimulator as the backend as in the in-class tutorial.

   (a) Prepare $|0\rangle$ and measure in the computational basis.

   (b) Prepare $|1\rangle$ and measure in the computational basis.

   (c) Prepare $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and measure in the computational basis. Comment on the discrepancy from the ideal result. What is the number of shots needed to obtain an (average) accuracy of 1%?

   (d) An end of a quantum a quantum computation always ends in measurement. Comment on any implications of your results in (c) on quantum algorithm implementations.

(e) Design a circuit which will always give a measurement outcome of +1 for an input state defined by the Bloch vector $\vec{v} = \frac{1}{\sqrt{2}}(0, 1, 1)$.

3. *Visualizing density matrices:* There are several ways to visualize the final state. For the states below, plot both the Bloch vector and also the "cityscape" of the density matrix. Be sure to add to your code `from qiskit.visualization import plot_state_city, plot_bloch_multivector`. Both visualizations work for (wave)vectors and density matrices.

   (a) $|0\rangle$

   (b) $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

   (c) $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$

4. *Single qubit tomography:* In class we discussed quantum state tomography. Here we will attempt to use quantum state tomography to detect SPAM errors.

   (a) *Single qubit tomography, ideal*: Prepare the following states and plot both the density matrix representation and Bloch sphere representation of the final state. Use the QasmSimulator as the backend. After preparing your circuit (which should consist of a single qubit), perform tomography using the `StateTomography()` [1]. (e.g. see tutorial.) Plot both the Bloch vector and also the "cityscape" of the density matrix.

      i. $|0\rangle$

      ii. $|1\rangle$

      iii. $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

      iv. $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$

   (b) *Single qubit tomography, qpu:* IonQ cites this paper for their 11-qubit computer performance which demonstrated 0.7% SPAM (state preparation and measurement) errors.

      i. Estimate the number of shots you need to detect SPAM errors with S/N of 10 assuming the 0.7% SPAM error rate.

      ii. For state $|0\rangle$, run the qiskit tomography code with the number of steps from (i) using the ionq.simulator backend to check your results are reasonable. [2]

iii. Run the resource estimation step to check that resources are reasonable. Share your resource estimation on the discussion board for approval to move to the next step. Because Azure can only send one circuit at the time, estimate using the following code

```
circuits = qstexp = StateTomography(qc)
qiskit.transpile(qstexp.circuits(), backend)
for c in circuits:
cost = backend.estimate_cost(c, shots=1000)
print(cost)
```

iv. Run your code in (ii) on the ionq-qpu for the input states $|0\rangle$ and $|1\rangle$. For each state,

   A. `print` the density matrix

   B. Plot both the Bloch vector and also the cityscape visualization of the density matrix.

   C. Calculate the fidelity of the measured state with the ideal state.

v. Discuss your results including any deviations in both the length and angle of the Bloch vector.

vi. *Optional:* Repeat utilizing 2 or 3 input qubits to determine if you can detect differences in SPAM performances between qubits. Discuss your results. We are not sure if Azure supports the qiskit function ParallelExperiment. If it does not, this will be a more challenging problem. [3]

---

[1] Be sure to add to your code `from qiskit_experiments.library import StateTomography` and `from qiskit.quantum_info import DensityMatrix`

[2] Because QuantumTomography creates a list of circuits, and Azure was not set up to send circuit lists, please install the beta package using `!pip install azure-quantum[qiskit]==0.23.201228b1`

[3] For a related work in the literature see Chen, Farahzad, Yoo and Wei, PRA 100, 052315, 2019.