Chem560

Homework #2: Project - Teleportation and Noise

Submit everything as a single Jupyter notebook *unless otherwise noted* through Canvas. You may submit your solution as a group of up to three members. In each submission, state all group members names and departments. Collaboration between groups is encouraged.

1. *Teleportation*

   (a) Prepare the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, perform state tomography, visualize the density matrix using both `plot_state_city` and `plot_bloch_multivector` and measure its fidelity.

      i. on the IonQ simulator (1000 shots)

      ii. on the IonQ qpu (1000 shots)

   (b) Prepare the entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ from the inital $|00\rangle$ state. Perform state tomography, visualize the density matrix using `plot_state_city`, and measure its fidelity

      i. on the IonQ simulator

      ii. Check the cost estimate on the IonQ qpu. Since there are many circuits being sent as past of two-qubit tomography, be sure to use:

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

      iii. on the IonQ qpu

   (c) Prepare a three qubit circuit to prepare and teleport the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ by preparing and using the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Use the deferred measurement method (Fig. 4.15 in *N&C*). Perform state tomography on the final teleported state, visualize the density matrix using both `plot_state_city` and `plot_bloch_multivector`, and measure its fidelity[1] In this case, you cannot use `StateTomography`. Modify the circuit to run the three measurements on the teleported qubit needed for single qubit tomography.

i. on the Aer or Qasm simulator (this will work for the standard, measurement-based teleportation circuit as well).

ii. on the IonQ simulator

iii. on the IonQ qpu

Do your results in (c) make sense based on your results in parts (a) and (b)?

2. *Modeling Noise*

(a) As discussed in class, we can gain a better understanding of noise when we visualize what happens to Bloch sphere under the noise operation. Plot the Bloch sphere transformation for the two most important noise operations: phase damping and generalized amplitude damping. Start with the general expression for the single qubit ρ in terms of the Bloch wavevector components, calculate $\varepsilon(\rho) = \sum_k E_k \rho E_k^\dagger$, and calculate the resultant Bloch vector. You can turn in this calculation in a separate file from your Jupyter notebook. Plot the resulting Bloch "sphere".

i. Phase damping. Use 8.127 and 8.128 in *N&C*.

ii. Generalized amplitude damping. Use 8.116-8.119 in *N&C*.

(b) Qiskit allows you to create a generalized noise model. (Qiskit also has noise models for several of their qpus (but beware, real noise drifts!) In this problem we create a noise model for the IonQ machine based on their published metrics and then compare our simulated model to the IonQ performance for a four qubit GHZ state. Follow this tutorial for aid in coding.

i. Create the circuit to make a four qubit GHZ state and measure the outcome in the measurement basis in the noiseless case. (We are not going to do full state tomography.)

ii. Create the noise model.You can assume all gate (single and two qubit) errors are dephasing errors (`phase_damping_error`) or use the full relaxation model (`thermal_relaxation_error`) in which you can pick T_2 and the gate times to give the desired error rate. Do not take into account native gates. Assume all qubits have the same error rate (which is not so realistic) with a single qubit state fidelity 99.5% after a single qubit gate and a two-qubit state fidelity of 97.5% after a two-qubit gate. Assume a measurement error is 0.7%. Because we are not doing full gate set tomography (or randomized benchmarking) you can justify your noise parameters using a single gate (*e.g.* H for a single qubit and create a Bell state for the two-qubit.).

iii. Measure the outcome for the four qubit GHZ state with your noise model. Use 1,000 shots and 10,000 shots.

iv. Measure the outcome for the four qubit GHZ state on the IonQ simulator (1,000 and 10,000 shots). Estimate the qpu cost for 1000 shot. Check that the 1000 shot cost is reasonable (i.e. $<$ \$10) before moving on.

v. Measure the outcome for the four qubit GHZ state on the IonQ qpu for 1000 shots. Optional: if the cost for 1000 shots is $<$ \$2, you can repeat with 10,000 shots.

vi. Compare your results for the noiseless, simulated noise and real qpu. Discuss how you might improve your noise model.

---

[1] You can use `state_fidelity` (include `from qiskit.quantum_info import state_fidelity`) or compute using the definition of fidelity.