Rogues Gallery - Quantum Computing

Alexander George Kennedy, Jack Sullivan, Dylan Yost, Andrew Asper, Dequontay Stodghill, Saahil Sanganeria, Ashwin Srivastava, Stefan Talpa, Srikar Vanavasam

Project Goals

This semester we formulated a new neural net architecture for cross-checking our previously designed LSTM-driven circuit optimization algorithm. The crucial component of its policy refinements is the analytic noise function we previously formulated determining the benefit of a given gate-swap by the LSTM. To this end, with the new neural net architecture, our work encompassed developing a random circuit generator to run large batches of circuits on various IBMQ hardware backends, new encoding for these to pass into both neural networks, and the new sequential network architecture itself, for predicting hellinger fidelity values for validation of the analytic noise model.

Dataset Construction: Random Circuits on IBMQ Architectures

- Created a script to generate the data by creating a random circuit, computing the corresponding input vector, running the circuit on the quantum backends.
- Idealized return compared with actual distribution of outputs from backend to compute fidelity.
- Automated data generation to run on CRNCH clusters to allow for faster data generation.
- Optimized the data generation script to run 1200 circuits at a time as opposed to 1. This greatly improved our data generation capabilities.
- Optimized our script to find the least busy IBMQ backends, in order to allow our circuits to be entered into the shortest queues to save time.

Reformatting Neural Net Input Vector

- Using one-hot encoding to match categorical input (categories are gate types)
  - Input vector must be elongated to make room for new binary encoding of the circuit geometry
  - Optimizes LSTM, which optimizes gate swaps using the noise function.
  - Optimizes DNN for predicting fidelity values given a vector per random circuit.

Fidelity Predictive Sequential Neural Net

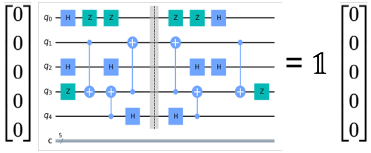
- Fidelity prediction as a regression task (via mean absolute error)
  - Input vectors as described: embedding of circuit + benchmarked noise values of backend run on.
  - Labels as empirical fidelities returned by IBMQ runs of a given vector (circuit).
- Several dense, fully connected layers (ReLU activation).
- Output layer - one neuron (Linear activation).
- Preliminary results shown below for training on one dataset of 1200 circuits. More data needed over **more backends** - **<u>generalizability crucial</u>**.

Keep & Change: Next Steps and Future Challenges

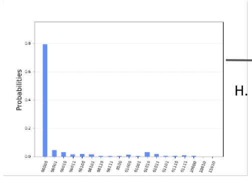
- Explore independent applications of the fidelity-predictive DNN.
- Implement scheduled data generation on CRNCH cluster for training on all available IBMQ backends
  - Training and testing network on as many architectures as possible (especially higher qubit count ones) - minimize overfitting.
- Testing different fidelity predicting DNN architectures to increase generalizability, as well as adjustments to current DNN.
- Devise additional theoretical noise models for use with policy networks akin to our LSTM circuit optimization.
- Publication of dataset.
- Implement gate swaps directly on input vector for use with LSTM.

Circuit attached to its mirror:

No Noise (Ideal Simulation)



Reality of Outputs:



{Circ. vector, H. Fidelity (counts, null)}

1000 epochs of training,

1200 circuits (vector, fidelity)