Learning basis gates for one qubit with microwave pulses

Jaden Nola, Elizabeth Behrman[1]

[1] Dept. of Mathematics, Statistics and Physics, Wichita State University

Introduction

What is a qubit?

Qubits are the building blocks of quantum computing. Unlike the classical computer bit whose value is always known, we have a *probability* of finding a value of zero *or* one when we *measure* a qubit. To find the probability of measuring one or the other, we can look at the quantum **state** of a qubit, which can be represented by:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \qquad \begin{matrix} 0 \le \theta \le \pi \\ 0 \le \varphi \le 2\pi \end{matrix}$$

Just like with bits on classical computers, we interact with qubits using operations known as **gates**.

One way of physically realizing a qubit is the **superconducting qubit**. To implement a gate, we send **microwave pulses** (represented with a time-varying voltage $V_d(t)$) with specific parameters that result in our desired transformation [1].

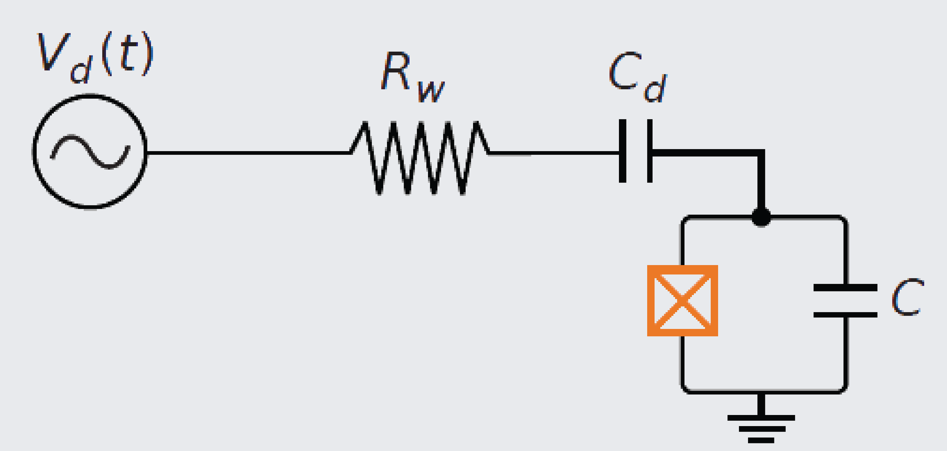

Figure 1. Depiction of a transmon qubit — a type of superconducting qubit — with a microwave drive line attached [1]

Quantum Neural Network

A neural network consists of layers of nodes, with a typical send-forward structure being one input layer, one or more hidden layers, and one output layer. Each node connects to every node of the next layer, each connection with an associated weight and bias, which are numbers that determine whether the node "activates". Neural networks work to minimize a **cost function**, which determines how *wrong* the output of the neural network is compared to what we want. We adjust the weights and biases based on what would minimize the cost function. One pass-through of a training dataset of inputs and expected outputs is called an **epoch**.

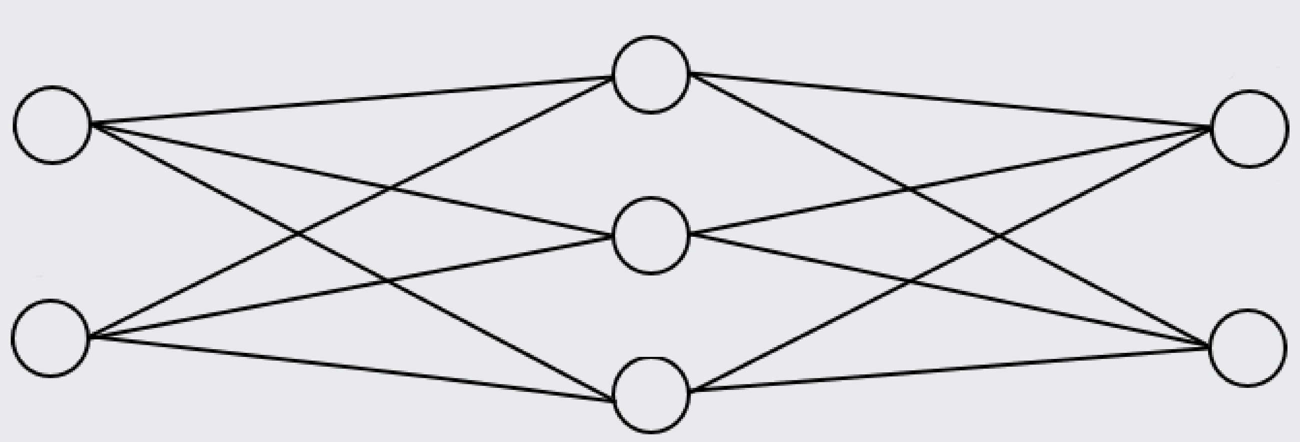

Figure 2. A neural network with one hidden layer

To carry out this work, we employ a **Quantum Neural Network (QNN)**, which is similar to a classical neural network, but the hidden layers incorporate operations on quantum devices. It has been shown that models of this sort have advantages such as scaling better to larger systems with more qubits and being robust against noise and decoherence [2].

Difficulties of Quantum Computing

Despite the developments of the past few decades that show the promise of quantum computing, useful quantum algorithms are quite hard to come by. To be useful, quantum algorithms must provide a **quantum advantage** over classical algorithms — they must be faster than any current and future classical approach. To combat these difficulties, our group has proposed the usage of machine learning techniques to instead train quantum algorithms.

Method

We implement our QNN using the Python libraries Qiskit Dynamics [3] and PyTorch [4], where our single hidden layer is a simulator from Qiskit Dynamics.

Our cost function is the **infidelity** — one minus the fidelity. Fidelity is defined by:

$$F = \left(\mathrm{Tr}\sqrt{\sqrt{\rho}\,\sigma\sqrt{\rho}}\right)^2$$

Training datasets were obtained by generating 10 random θ and φ pairs as our dataset for each QNN. From these, we were also able to generate the desired transformations for each of the gates through matrix multiplication.

The pulse we train is called a **Derivative Removal by Adiabatic Gaussian (DRAG)** pulse. This means that the we needed to train the following parameters: duration, amplitude, variance, and correction amplitude.

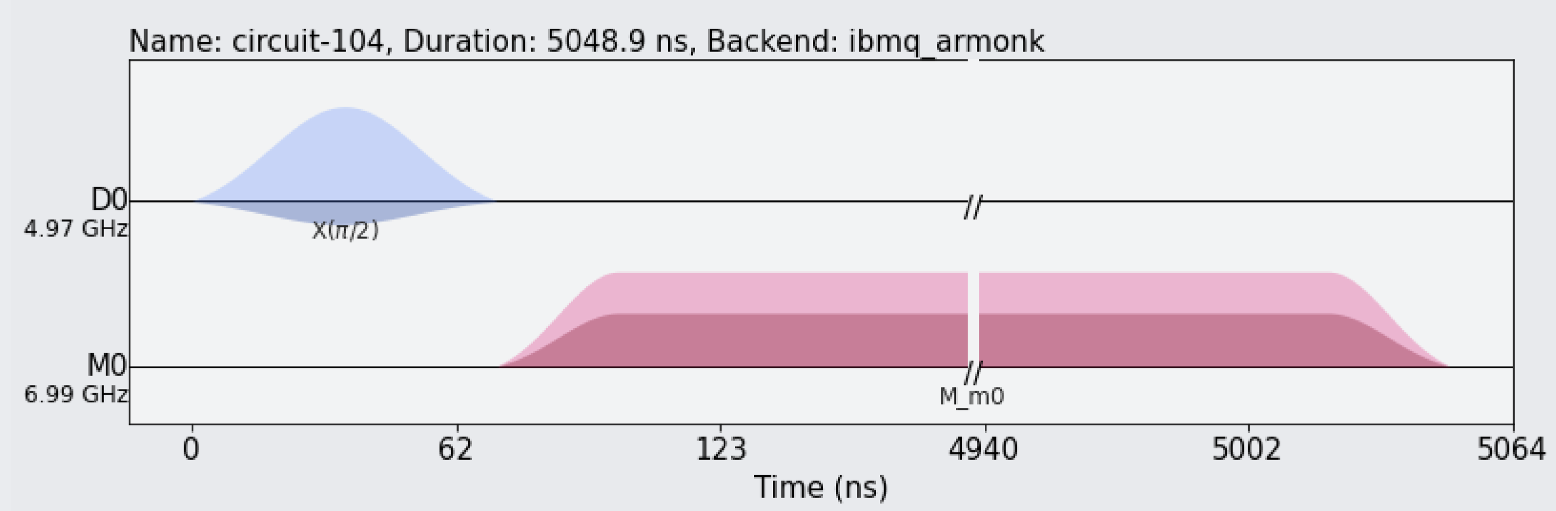

Figure 3. Pulses that correspond to an SX-gate and measurment. This was plotted using Qiskit and matplotlib.

Future work

Next we hope to train a Hadamard gate for a single qubit. We also hope to apply this work to systems of multiple qubits, specifically with the intent of improving the fidelity of the model described in [2].

Acknowledgements

We thank the entire research group whose discussions have been valuable to the development of this project. We are thankful to the various contributors of the Qiskit and PyTorch library. We also acknowledge the use of the IBM Quantum services for this work.

References

[1] P. Krantz *et al.*, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, vol. 6, p. 021318, jun 2019.

[2] N. H. Nguyen, E. C. Behrman, and J. E. Steck, "Quantum learning with noise and decoherence: a robust quantum neural network," *Quantum Machine Intelligence*, vol. 2, p. 1, Jan 2020.

[3] H. Abraham *et al.*, "Qiskit: An open-source framework for quantum computing," 2021.

[4] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

Preliminary Results

We found that we are able to train pulses corresponding to the quantum X and SX gates successfully on our simulator. These pulses have an infidelity with the desired gate state with an order of magnitude between 10^{-4} to 10^{-8}. The infidelity of an X-gate model and an SX-gate on a set of initial states are shown below:

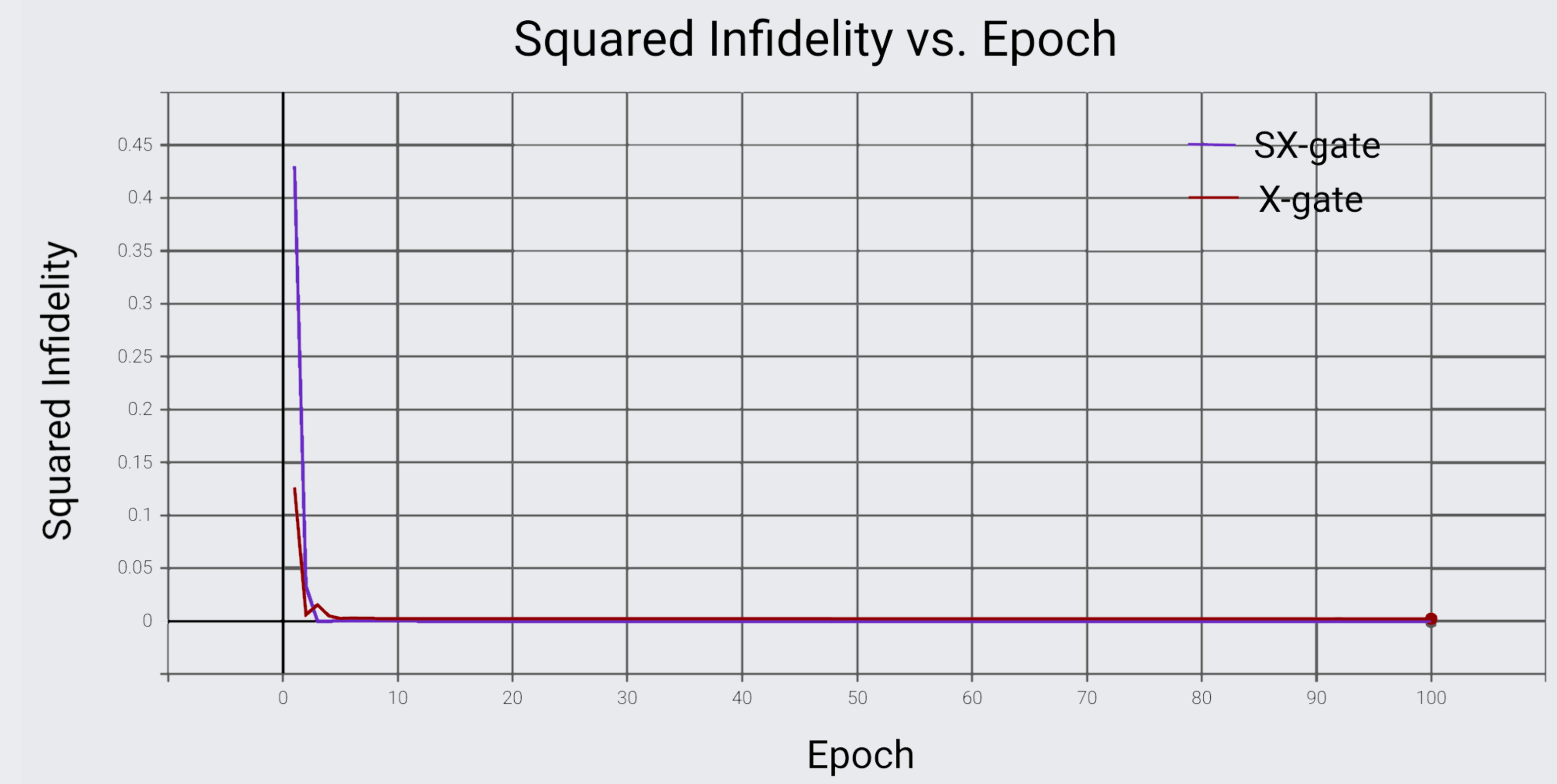

Figure 4. A graph of squared infidelity vs. epochs trained. This was plotted using Tensorboard, which is supported by PyTorch.

From this graph, we can see that the model trained very quickly in less than 10 epochs. In general, the X-gate was easier and faster to train than the SX-gate.

In addition to training on our simulator, we tested gate identities on a physical quantum computer at IBM: *ibmq_armonk*. Probabilities from a Hadamard gate decomposed into an S-SX-S composition with our trained SX-gate pulse are shown below.

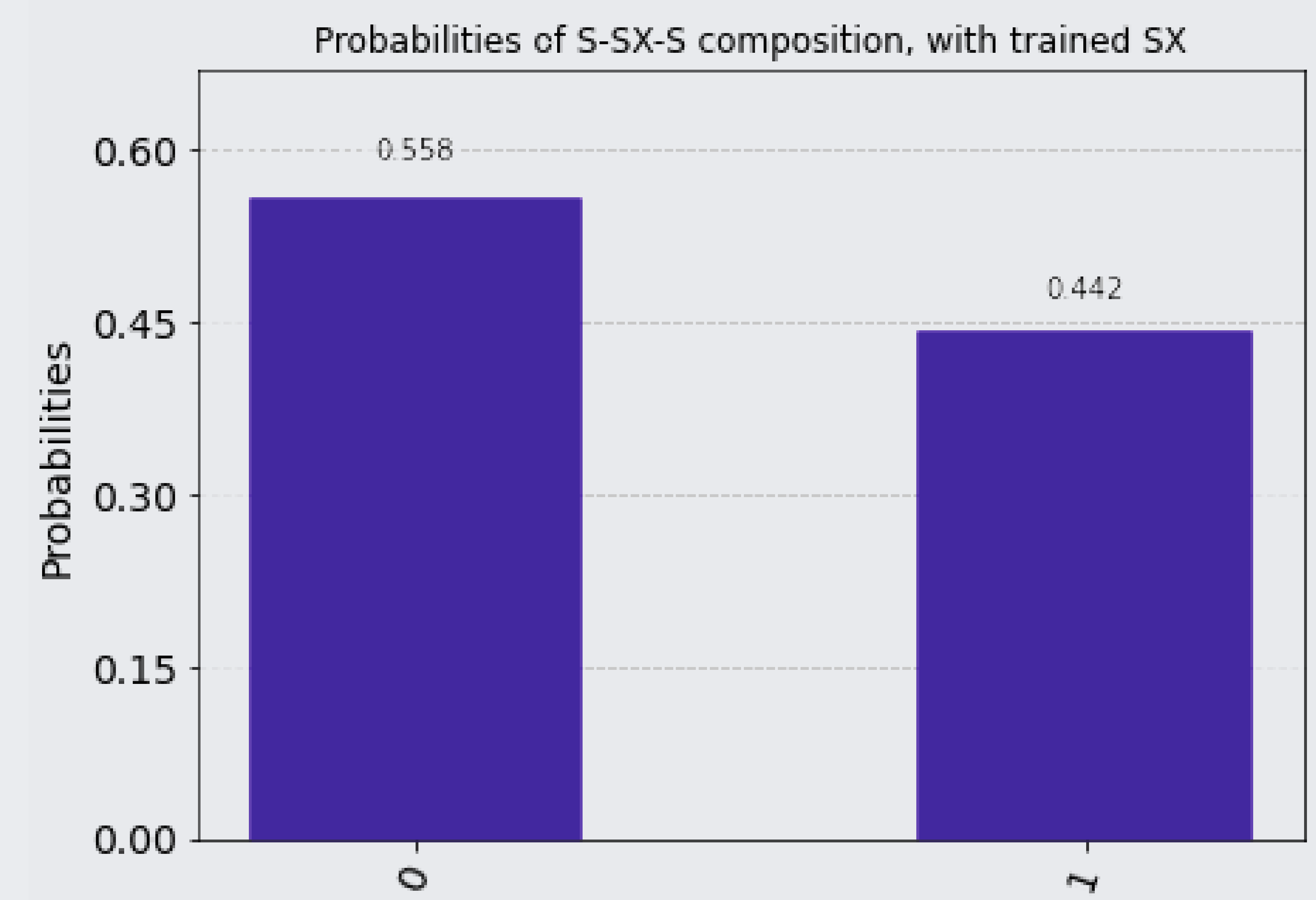

Figure 5. A bar graph of probabilities from an S-SX-S composition, with S being native gates on the computer and SX being our trained pulse. This was plotted using Qiskit's *plot_histogram* function.

These probabilities are nearly identical to the same gate composition but with a native SX gate on the computer, which gives us confidence in having trained a pulse with at most a difference in the global phase.