Higher Energy State Discrimination of IBM Superconducting Qubit Quantum Computer Using Machine Learning

I-Tung Chen

Department of Electrical and Computer Engineering
University of Washington
Seattle, WA
itungc@uw.edu

June 9, 2021

Abstract

Superconducting qubit has been through a rapid development in recent years, while the fidelity has been increasing for each qubit but higher energy states(e.g. the second excited energy level) of superconducting qubit has not been widely explored since typically ground and first excited energy level are used only. The higher energy state can be potentially used in quantum information processing but to correctly classify different energy state within a qubit is still challenging. This report will try to explore the higher energy state of qubits via IBM Qiskit package[1] and present a method to correctly classify each energy state using pytorch neural network architecture.

1 SC Qubit Energy State Introduction

1.1 What Are Qubits?

Quantum Computers, which can access the mystery power of quantum mechanics, use quantum bits, a.k.a qubits, as a basis to process and store information. Unlike classical bits that can only be either 0 or 1 states, qubit can be at a state that is in between 0 *and* 1, and this is a unique resource that we can harness from quantum bits: superposition of states are written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Where α, β are the phase information in which we encode our information. We will present more detail about

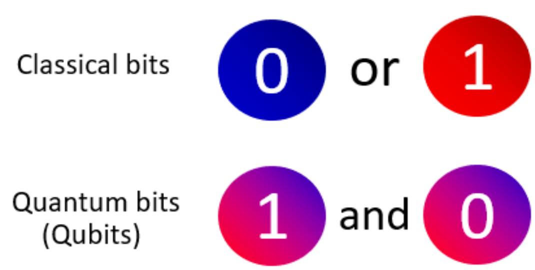

Figure 1: Qubits and classical bits

the physical realization of qubits in microwave circuits in section 1.2. Other than microwave pulses, there many different physical types of qubits: a single photon with two directions of polarizations, or an electron with two directions of spins, even cold atoms traped by lasers can also be a qubit. Each has it strength and weakness, and we will focus only on microwave qubits in this report.

1.2 Anharmonicity of Transmon Qubit

The Hamiltonian of quantum harmonic oscillator (QHO) can be calculated by quantizing Hamiltonian of a linear LC circuit. It's energy levels are equally spaced, shown in Fig.2a. However, by replaciing the inductance in the LC circuit with a Josephson junction, the energy level of such resonator will be modified. That is, in other word, the energy level of the transmon qubits are now unevenly spaced, so we call it the anharmonicity of the qubit energy level,shown in Fig.2b. The spacing between higher energy state will decrease: $|0\rangle \rightarrow |1\rangle$ will need bigger energy than $|1\rangle \rightarrow |2\rangle$. The anharmonicity δ is defined as the energy difference between each energy level transition: $\delta = \hbar\omega_{12} - \hbar\omega_{01}$. By tuning the microwave frequency to ω_{01}, me can effectively address the multi-level transmon qubit as a two-level system, that is, a qubit.

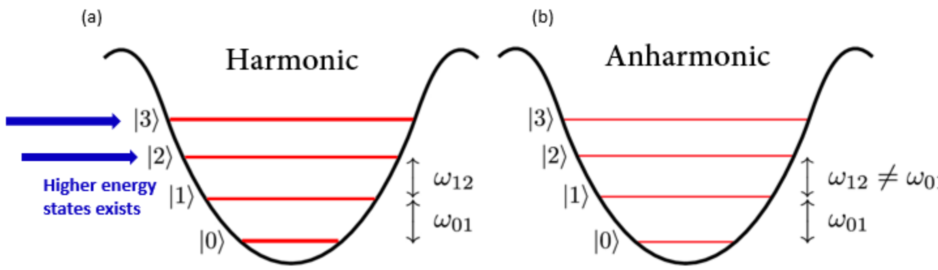

Figure 2: The comparison of Harmonic and Anharmonic energy level

1.3 Higher Energy States of Qubits

In most quantum algorithms/applications, computations are only done in a 2D space spanned by $|0\rangle$ and $|1\rangle$. However, in IBM's hardware, higher energy states exist too but are not typically used.

This report will focus on exciting these higher energy state by tuning the microwave frequency and build a discriminator to classify the $|0\rangle$, $|1\rangle$, and $|2\rangle$ states.

2 Accessing Higher Energy State Using Qiskit Pulse

2.1 Qubit Calibrations and Accessing $|0\rangle \rightarrow |1\rangle$ state

Before we do the classification task, we first need to calibrate the qubit central frequency with Qiskit Pulse. The quantum hardware used in this report is Armonk, a single-qubit quantum processor that supports Qiskit Open Pulse operation, which is accessible via IBM Quantum Experience free of charge. The quantum hardware used in this report is *ibmq_ Armonk*, which has one 1 qubit and an anharmonicity of -0.347GHz. In Qiskit Pulse, we can program different channel to send in the microwave pulse signal to drive the qubit and measure the outcome, as shown in Fig.3. The drive channel sents in a microwave pulse with a gaussian amplitude distribution. The goal of sending in this microwave pulse is to calibrate the central frequency of the qubit in Armonk. Each time we start the measurement we will have slightly different central frequency so it is crucial to find out the true central frequency before we start the experiment.

At first, we know the default qubit frequency in Armonk is 4.9718GHz, so we drive the qubit with a Gaussian microwave pulse and scan the resonance frequency with a span of ±20MHz at default frequency to find the exact frequency for our qubit. As shown in Fig.4(a), we can see there is a peak at 4.9718GHz, meaning our qubit's resonance frequency is locate at this point. Once we update our excitation frequency to the newly calibrated frequency, we are all set and ready to access $|1\rangle$ state! After calibration, now we do something called the Rabi experiment, basically what it does is that it send in a microwave pulse and excite the qubit from a lower energy state($|0\rangle$) to a higher energy state($|1\rangle$). The result is shown in Fig.4(b) and we can see that it is a sine wave output, the reason it look like this is because the qubit is oscillating between the $|0\rangle$ and $|1\rangle$ state. The hill-top of the sine wave corresponding to $|1\rangle$ and valley of which is $|0\rangle$.

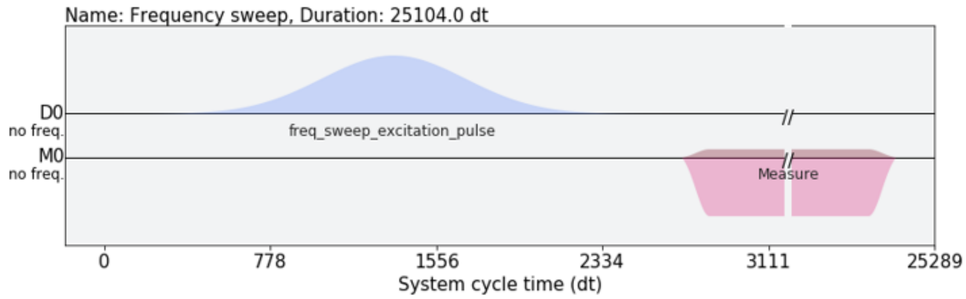

Figure 3: Drive channel and measurement channel in Qiskit Pulse

The drive amplitude between the hill and valley is the *pi-amplitude* of our qubit, which is 0.1414, meaning the amplitude we need to excite the qubit from $|0\rangle \rightarrow |1\rangle$.

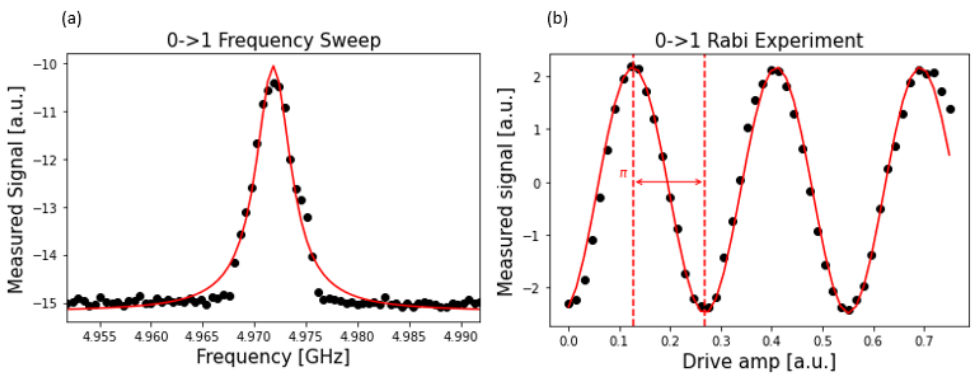

Figure 4: Result of the frequency sweep and Rabi experiment. (a)Output of the excited qubit signal (b)Rabi experiment of the qubit

After knowing the pi-amplitude and our central frequency, we can start measuring the outcome of our qubit states. When we do nothing and measure the qubit, it will be in the $|0\rangle$ state. And when we sent in microwave pulses, 1024 shots to be specific, we can measure the outputs which represent $|1\rangle$ state. The outcome of the 01 shots are fairly straight forward, as shown in Fig.5, each dot is one shot, and the blue(red) dots are the $|0\rangle(|1\rangle)$ state. The classifier of 01 state is simply an linear classifier, which give relatively high scores 0.96.

2.2 Accessing $|1\rangle \rightarrow |2\rangle$ States

While 01 discriminator is easy to build and execute, we now turn our attention in accessing $|1\rangle \rightarrow |2\rangle$ States. To get our hands on $|2\rangle$, we need a special method called the sideband pulse to excite the qubit into $|2\rangle$. The reason we use this method is due to the limitation of the quantum hardware, where the highest energy we can excite cannot allow us to directly access $|2\rangle$. So we need to first pump in a $|0\rangle \rightarrow |1\rangle$ and then send in an additional sideband to open the way for the qubit to jump from $|1\rangle \rightarrow |2\rangle$. The pulse sequence is shown in Fig.6, pi-pulse is a Gaussian pulse and sideband pulse is a sine wave pulse envelope by a Gaussian shape. Before we read out the signal, we need to sweep the frequency and find out what is the resonance frequency of the $|1\rangle \rightarrow |2\rangle$ transition. The 012 frequency sweep is shown in Fig.7, the central frequency is around 4.623GHz and we also do the Rabi experiment to know the 012 pi-amplitude, which is also shown in Fig.7 with the value 0.2498. Finally, we can visualize the $|0\rangle, |1\rangle, |2\rangle$ state in the IQ plot in Fig.8.

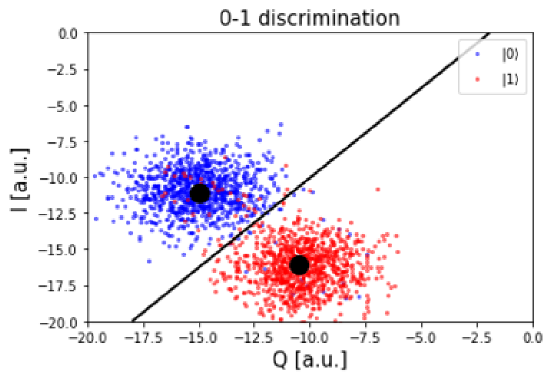

Figure 5: The signal is plotted by taking the real and imaginary part ot the complex number outcome. 01 discriminator can be easily done by a linear classifier.

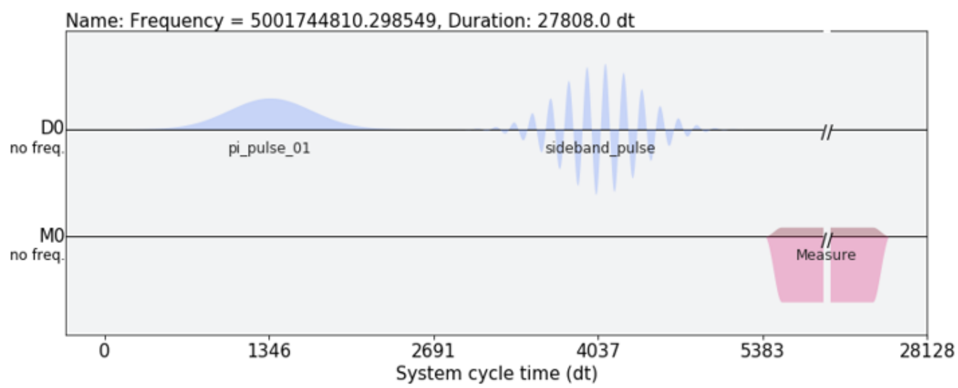

Figure 6: 012 Sweep. We first send in a 01 pi-pulse and then send in a sideband to access the $|2\rangle$ state in our qubit.

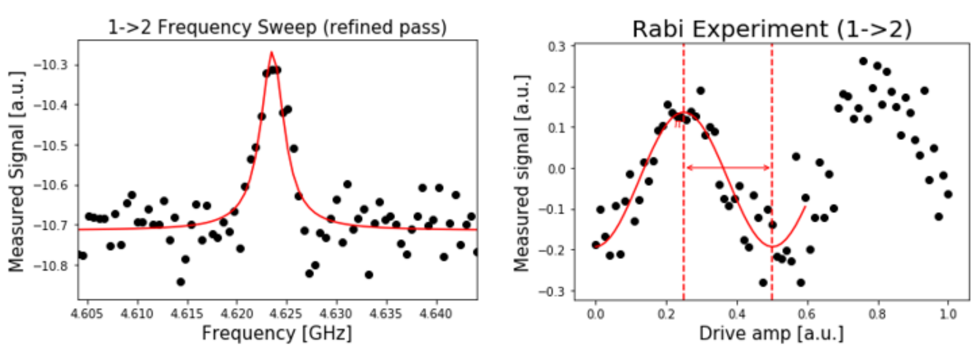

Figure 7: 012 Sweep Result. The frequency sweep result and Rabi experiment result.

3 Classification of Higher Energy States

So far, we have done the qubit calibration, and find out the pi-amplitude for both $|0\rangle \rightarrow |1\rangle$ and $|1\rangle \rightarrow |2\rangle$ transition. Now it is finally time to do the classification task for higher energy states in our qubit. In this

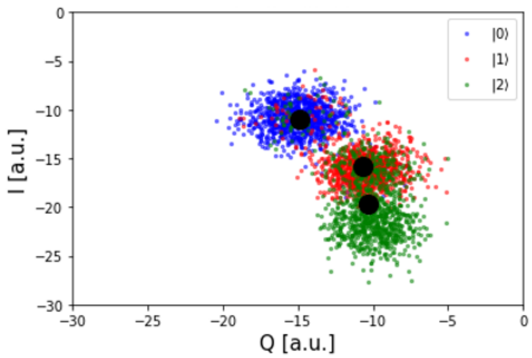

Figure 8: 012 State Result.

section we will explore several algorithms including simple linear discriminant anasysis and fully-connected neural networks to classify our higher energy state. And we will compare the result of each method to find out the one with highest score.

3.1 Linear Discriminant Analysis(LDA)

To start, we first test LDA method to classify the higher energy state. The LDA function deployed here is the built-in function in sklearn. The result is shown in Fig.9(a). The black line is two linear function combined to show the result. We can see that $|1\rangle$ has been zoned out but the borders between $|0\rangle$,$|1\rangle$ and $|1\rangle$,$|2\rangle$ are not clearly defined. The score for LDA analysis is 0.808.

3.2 Simple Multi-layer Perceptron (MLP)

Now we switch to a buld-in MLP function in Sklearn with 1 hidden layer of 100 neurons. We use lbfgs as solver with 1000 iterations. The result is shown in Fig.9(b), we can see that the boundaries of each state is slightly more accurate and it gives us a slightly higher score of 0.839. The sklearn built-in function did not allowed too much flexibility in terms of tuning hyperparameters of the MLP so next we try to build our own fully-connected layers for the task.

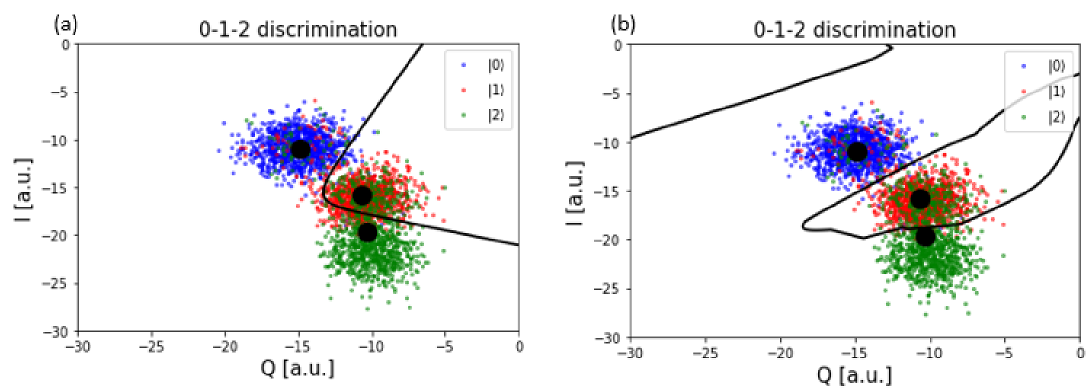

Figure 9: Higher Energy State classification. (a) using LDA function for classification (b) using MLP function for classification

3.3 Custom-Built Fully-Connected Layers (FC)

The following we provide the structure of our custom-built FC layers in pytorch. The network consist of 2 hidden layers with 10 and 50 neurons. Use use ReLu() and Sigmoid() as activation functions. Loss function used here is CrossEntropyLoss(), learning rate is 0.05, and the optimizer is Adam. The loss and accuracy of the training process is shown in Fig.10. The training accuracy is maximized at 0.84. We then look at the comparison between the predicted states and the real state side-by-side, as shown in Fig.10. In the predicted plot we see that the classifier can not distinguish the overlapped $|1\rangle$,$|2\rangle$ state, so all the prediction made in the same region are the same. The final testing accuracy is also around 0.84.

```
#Set-up a pytorch non-linear discriminte
import torch
import torch.nn as nn

X_train = IQ_012_train
Y_train = state_012_train
X_test = IQ_012_test
Y_test = state_012_test

# Define network dimensions
n_input_dim = X_train.shape[1]
# Layer size
n_hidden =10 # Number of hidden nodes
n_output = 3 # Number of output nodes

# Build your network
net = nn.Sequential(
    nn.Linear(n_input_dim, n_hidden),
    nn.ReLU(),
    nn.Linear(n_hidden,50),
    nn.ReLU(),
    nn.Linear(50,50),
    nn.ReLU(),
    nn.Linear(50, n_output),
    nn.Sigmoid(),
        )
```

3.4 Comparing Different Methods

Here we summarize all the different method used in this report and show the test score of each methods. As shown in Table.1, the score maximized at around 84%. This is probably it is difficult to classify the point overlapping in the same area. Given the data sampled from the quantum hardware are just tuple of complex numbers, the classification accuracy is limited by the data sources.

Table 1: Score Comparison of Different Classification Method

Methods		
Name	Description	Score (%)
LDA	2 LDA functions	~80.3
QDA	1 QDA functions	~82.4
MLP	1x100 1 Hidden layer	~83
FC	1x10, 1x50 2 Hidden layers	~84

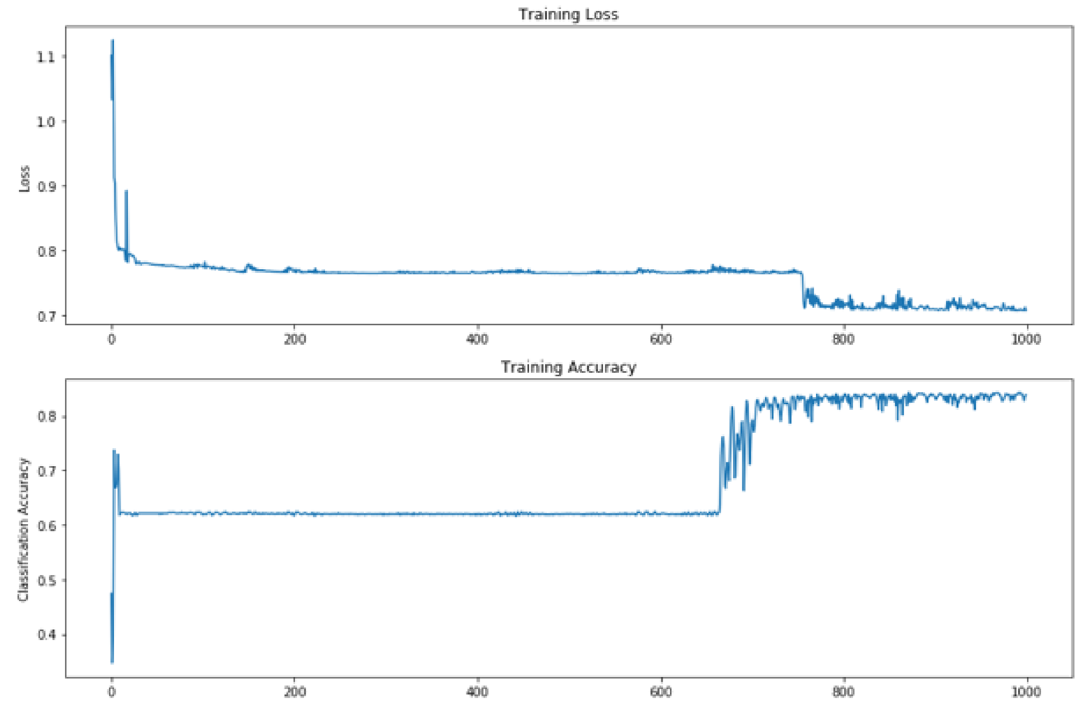

Figure 10: Training loss and the accuracy of the costum-built FC layer for classification

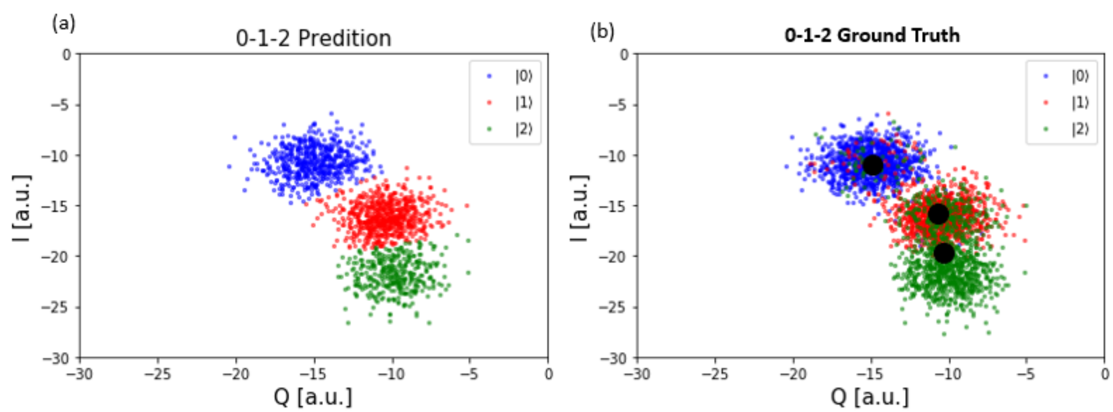

Figure 11: Higher Energy State classification Using Custom-built FC. (a)the predicted state of the classifier (b) groud truth of the states

4 Summary and future directions

In this report we access the higher energy state via a real quantum hardware from IBM. Calibrations are performed and higher energy states are identified. Then we continued with the classification tasks. Various methods were employed and test scores were recorded, the custom-built FC layer has the highest classification accuracy with 84%. Future directions of this project can be done in expanding the dimension of the data sets, currently the limitation of the accuracy is from the 2D nature of the plot, if in a higher dimension, the higher states may be easier to distinguish comparing to the current setting. Different networks might be employed too since in this report only the simplest form the NN are used, but it is demonstrated that machine learning can help with classifying higher energy states in a real quantum computer. So far we only worked on one qubit, if using a multiple qubit devices, complex NN can be used to improve the fidelity of the qubits.

Acknowledgments

Thanks to Dr. Wei for giving precious advises on performing qubit calibration and the inspiration for this report, and also big thanks to the wonderful Qiskit community for helping resolving my question when running the program.

References

[1] Qiskit textbook, IBM. https://qiskit.org/textbook/preface.html.

[2] QCHack 2021, Mingweii, https://github.com/mingweii/QCHack2021

[3] The source code used for this project can be found: https://github.com/ytchen2010/Accessing-Higher-Energy-State-with-ML-Qiskit