

# **Will Google & NASA Joining Hands Make Quantum Computing A Household Word**

**22/11/2018**

[RAM SAGAR](#)

I have a master's degree in Robotics and I write about machine learning advancements. email:ram.sagar@analyticsindiamag.com

## Intro

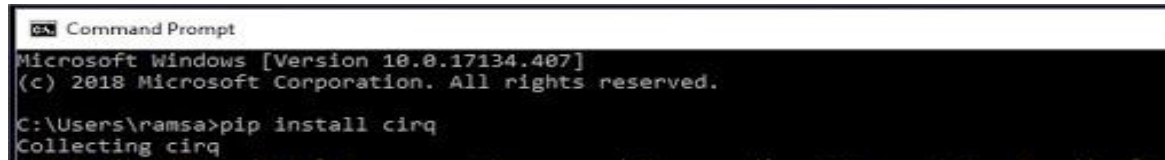
- With quantum computers, the next wave of transition might not take that long. The next 5 decades of [advancements](#) might reduce present-day [supercomputer](#) to the level of an abacus in comparison with super quantum computers.
- Google and NASA have joined hands to commercialize quantum computing back in 2013. And, they have made extraordinary progress with their smart team at [Quantum AI](#). They tried to address the simplest of quantum problems with [NISQ](#) computers and as a result, Cirq has been designed. Licensed under Apache2, this open source framework can be modified by the users for any commercial purpose.

## Evolution Of Cirq

- A quantum computer requires a quantum transistor unlike CMOS in a classical computer. We are no longer dealing with low and high voltage thresholds (0 or 1); electrons behave like waves in quantum states and the smallest measurable unit of this information is a qubit or quantum bit. A qubit can be in two states at any given point (coherent superposition). And, for controlling or measuring these qubits, we need metals, such as Niobium which assist superconductivity. The circuitry of a typical quantum processor is sophisticated; not to mention the cooling systems which can go as [low](#) as 15 milliKelvin (180 times colder than the interstellar space).
- 
- With such high standards involved in a hardware installation, the need for having a virtual environment to simulate quantum effects only makes sense.
- 
- Cirq package allows users to develop processor specific quantum algorithms. These algorithms can further be used to simulate the quantum environment like specifying the gating behavior and testing the constraints. These results can then be used to integrate with quantum hardware via a cloud.

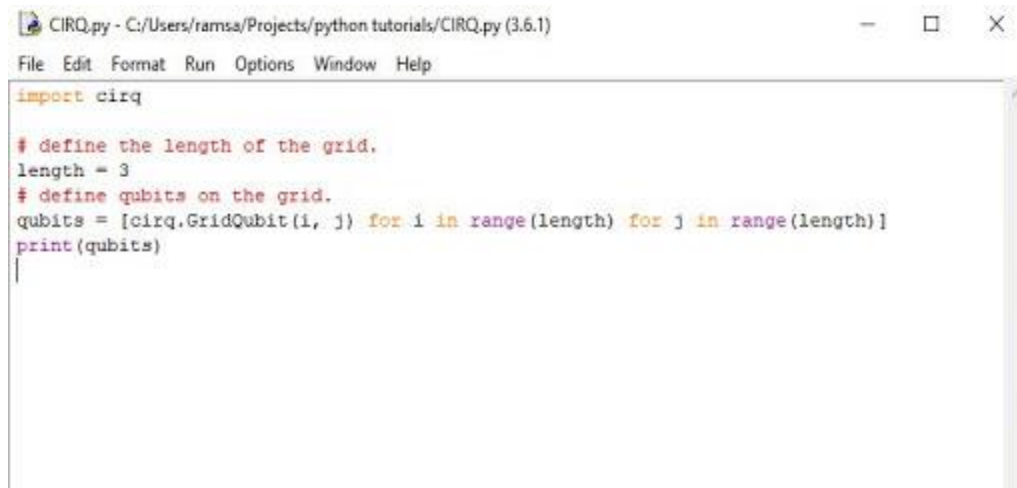
# Installing Cirq Package For Python

- This will install Cirq on to Windows and now we shall see how we can use Cirq's built-in modules to create qubits



```
Command Prompt
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ramsa>pip install cirq
Collecting cirq
```



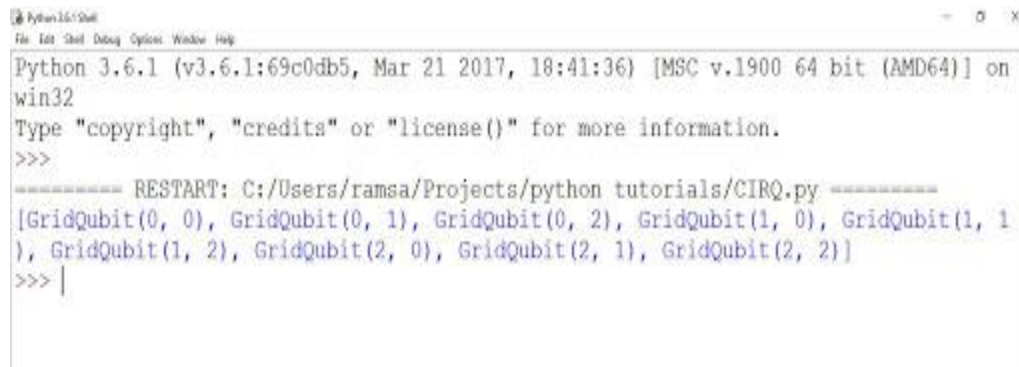
```
CIRQ.py - C:/Users/ramsa/Projects/python tutorials/CIRQ.py (3.6.1)
File Edit Format Run Options Window Help

import cirq

# define the length of the grid.
length = 3

# define qubits on the grid.
qubits = [cirq.GridQubit(i, j) for i in range(length) for j in range(length)]
print(qubits)
```

- The above code will list the qubits as follows:



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help

Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)] on
win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:/Users/ramsa/Projects/python tutorials/CIRQ.py -----
[GridQubit(0, 0), GridQubit(0, 1), GridQubit(0, 2), GridQubit(1, 0), GridQubit(1, 1
), GridQubit(1, 2), GridQubit(2, 0), GridQubit(2, 1), GridQubit(2, 2)]
>>> |
```

## construct a quantum circuit

- We can now use this to construct a quantum circuit as follows:

```
circuit = cirq.Circuit()
circuit.append(cirq.H.on(q) for q in qubits if (q.row + q.col) % 2 == 0)
circuit.append(cirq.X(q) for q in qubits if (q.row + q.col) % 2 == 1)
print(circuit)
# prints
# (0, 0): —H—
#
# (0, 1): —X—
#
# (0, 2): —H—
#
# (1, 0): —X—
#
# (1, 1): —H—
#
# (1, 2): —X—
#
# (2, 0): —H—
#
# (2, 1): —X—
#
# (2, 2): —H—
```

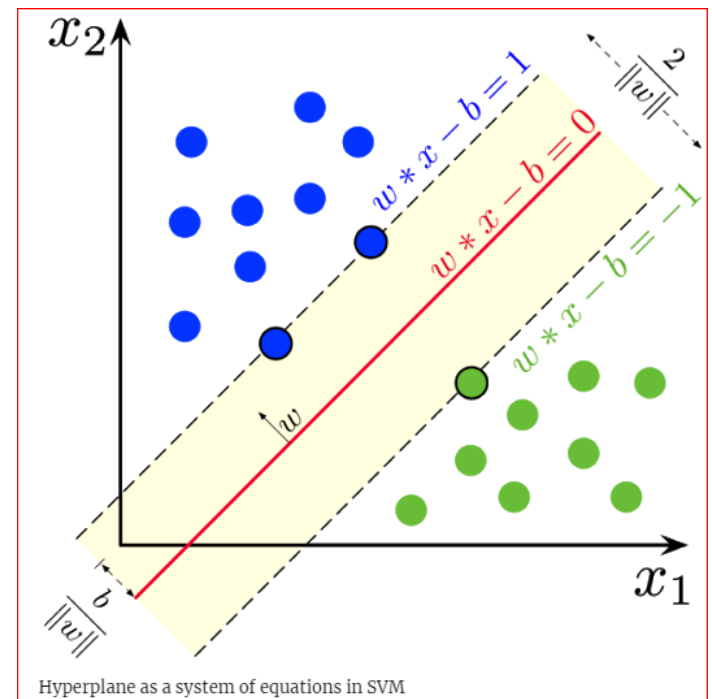
- Similarly, we can also create gates and test constraints with few lines of python code. The hassles surrounding building hardware quantum experiments are exceedingly high and running simulations [cirq.google.XmonSimulator()] in virtual environments like illustrated above will make it easy for the researchers to test multiple combinations and integrate later with the hardware.

# Probing Infinities With Quantum ML

- Amplitude encoding is one of the fundamental concepts in quantum theory where amplitudes of a quantum state can be defined in terms of a probability density [function](#). This function reveals how a quantum state vector relates to the observations made.
- 
- For every qubit, say  $x$ , there are  $2^x$  states. Now, imagine the number of states a 100 qubit processor can handle! Two qubits can store four states, three qubits can store eight states. So, 50 qubits in quantum computers can be approximated to 1 quadrillion bits in classical computers.
- This alone is the single most important feature and reason behind the arduous efforts to improve quantum computing. The exponentially increasing complex amplitudes can have as much information encoded. Now, this endless chain of information can be thought of as a large matrix representing equations. ML models run on a system of equations like these.
- 
- For example, a hyperplane in SVM can be written as,

$$wx - b = 0$$

Hyperplane as a system of equations in SVM



# Probing Infinities With Quantum ML

- So with increasing feature set, the complexity of the equation increases or let's say the size of a matrix increases and complexity is where quantum machine learning comes into [picture](#).
- By initiating a quantum state where amplitudes, that give away the probability density of the quantum state vectors can be related to the feature vectors in our dataset used for training the model. So anyone who has had an idea on how to run a standard regression model would comprehend how complex and time consuming it gets with increasing number of rows/columns (think millions).
- Building on Cirq, the researchers have also released application specific library [OpenFermion](#). This library contains necessary modules to tackle the problems related to quantum chemistry; the study which will allow the researchers to gain insights about how cathodes in a battery degrade or the electrical properties of superconductors.
- 
- Companies like D-Wave and Google have been trying to develop a sophisticated hybrid system which incorporates quantum effects into the classical ML problems since the turn of this decade and they will continue to do so. With the current fabrication techniques, D-Wave has managed to develop 2000 qubits processors. And, we have just read about the capabilities of a 100 qubits processor. So, it does make sense when [futurists](#) say that classical computers will be made to look like an abacus in front of quantum computers.

END