Quantum Computing: An Intro

Heman Gandhi

Rutgers – HackRU RnD

hemang@ndhi.ninja

November 29, 2018

Overview

1. Intro: what is this presentation
2. QC Applications and Questions
3. QC Basics
4. Messing with QC
5. An example: the Deutsch Oracle (time-permitting)

UwU, What This?

This is an intro to quantum computing. So what I'll be doing is going over brief, sometimes not very technical answers to the following (in chronological order):

- What should I have told my *insert family member here* about this last week?
- What are some questions people are asking about it?
- What is quantum computing?
- Where can I go to mess with quantum computing?

How to use this?

This is a subset:

- Fast factoring (Shor's Algorithm)
- Computational (micro)-biology
- Quantum Machine Learning
- Quantum Cryptography

How to Measure the Benefit?

There is a field called Quantum complexity theory. We know that quantum computers are at most exponentially faster from this.

We also get that we can solve circuit satisfiability in a square-root of the time (with an error bound).

Searches are also theorized to be in a square-root of the time. [Cleve]

Only recently can we verify whether a quantum computer even used quantum magic to compute. [Quanta, 2018]

Open Questions

This is a subset:

- How to implement...anything?
- What should programming languages look like?
- How to scale quantum computers to match the requirements?

What is Quantum Computing?

(See [WIRED, 2018] for more.)

To a Child:

Computers represent computers as coins with “heads” and “tails”. A quantum computer also lets you rotate the coin.

To a Teen:

Superposition is like trying to tell if a coin if heads or tails while it’s being tossed.
Entanglement is when two coins are forced to have the same state.

More rigorously...

This is why I used $\LaTeX$. Much of what follows about how Q-bits work is thanks to [Microsoft, 2018].

Definition

We write bits as vectors. So $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ is the 0 bit, written in "Dirac notation" as $|0\rangle$.

Any guesses about representing a 1-bit?

Classical Systems

Bit	**As a Vector**	**Dirac Notation**
1	$\begin{pmatrix}0\\1\end{pmatrix}$	$\|1\rangle$
0	$\begin{pmatrix}1\\0\end{pmatrix}$	$\|0\rangle$

Table: The Classical Bit States

Mnemonic: the Dirac notation gives you the index of the 1 in the vector.

Matrix Multiplication as Bit Operators

Bit operations can be thought of as certain matrices.

Bit Operation	**Matrix**
Identity	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
Not	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Set to 0	$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$
Set to 1	$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$

Table: The Classical Unary Bitwise Operators

What does invertibility mean here?

$\otimes$: Tensors

Definition (Tensor Product of vectors)

$$\begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ \vdots \\ y_m \end{pmatrix} \\ \vdots \\ x_m \begin{pmatrix} y_0 \\ \vdots \\ y_m \end{pmatrix} \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \\ 5 \\ 10 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = ???$$

Using Tensors

You can treat multiple bits as tensors of single bits:

- $|2\rangle = |1,0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
- $|4\rangle = |1,0,0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

Note that the mnemonic still works.

CNOT: A Building Block

Takes in 2 bits. If the first bit is 1, flip the second. Leave the first bit alone.

Here is the matrix:

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Example: $C|1,0\rangle = C|2\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |3\rangle = |1,1\rangle.$

This is an important building block.

Q-Bits

- The vectors we've been messing with are just special Q-bits. Any vectors $\begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{C}^2$ with $|a|^2 + |b|^2 = 1$ work.
- This is superposition. Each component is the square root of the probability of that component "collapsing" to a 1.
- You can see this as a unit circle for most of our purposes. The axis the bit is closer to is the bit it's more likely to collapse to.
- We can prove that for any vectors u, v, $|u \otimes v| = |u||v|$. This means that tensoring Q-Bits gives us valid Q-Bits.

Hadamard Gate

Takes a 0 or 1 and puts it in perfect superposition. This is a 45 degree rotation.

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix}$$

This is self-inverse, so you can go from prefect super-position to classical bits too.

Composition

So we have X as the bit flip and H as the Hadamard, giving us our operators. This results in the below (thanks to [Tatourian, 2018]):

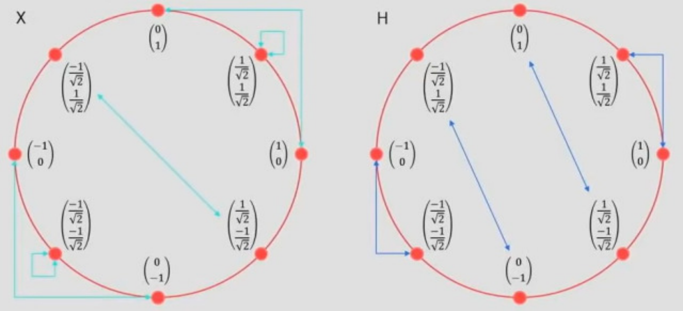
X
$\binom{0}{1}$
$\binom{\frac{-1}{\sqrt{2}}}{\frac{1}{\sqrt{2}}}$
$\binom{\frac{1}{\sqrt{2}}}{\frac{1}{\sqrt{2}}}$
$\binom{-1}{0}$
$\binom{1}{0}$
$\binom{\frac{-1}{\sqrt{2}}}{\frac{-1}{\sqrt{2}}}$
$\binom{\frac{1}{\sqrt{2}}}{\frac{-1}{\sqrt{2}}}$
$\binom{0}{-1}$
H
$\binom{0}{1}$
$\binom{\frac{-1}{\sqrt{2}}}{\frac{1}{\sqrt{2}}}$
$\binom{\frac{1}{\sqrt{2}}}{\frac{1}{\sqrt{2}}}$
$\binom{-1}{0}$
$\binom{1}{0}$
$\binom{\frac{-1}{\sqrt{2}}}{\frac{-1}{\sqrt{2}}}$
$\binom{\frac{1}{\sqrt{2}}}{\frac{-1}{\sqrt{2}}}$
$\binom{0}{-1}$

Figure: The Map for Moving Q-Bits around

IBM's API

Qiskit is bae.
You write QASM (Quantum-ASM) files and Qiskit runs them on a local simulator or online on a real quantum computer.
There is an online interface: and it's free for like 5 runs on a Quantum computer.

The Deutsch Oracle

Let $f\colon \{0,1\} \rightarrow \{0,1\}$ (so f is a bit operator). How do we know if it's constant? How can you do it on a standard computer?

The Deutsch Oracle: the Quantum Advantage

Let $f\colon \{0,1\} \to \{0,1\}$ (so f is a bit operator). How do we know if it's constant?
One query? He superpose.

The Deutsch Oracle: Reversibility

$f(x) = 0$ is not reversible.

Reversibility Hack

The hack: operate on two bits: $g(|0, x\rangle) = |f(x), x\rangle$ (Do not look at $|1, x\rangle$.)
The idea is that the 0 bit is the output wire and x the input.

This g is reversible. (Only proof I know: matrices.)
Asking about g is equivalent to asking about f, but now you can use quantum operators.

The Deutsch Oracle: What the Constant Functions Look Like

Constant 0

nothing on either wire.

Constant 1

X (the flip) on the output wire.

The Deutsch Oracle: What the Variable Functions Look Like

Identity

output $=$ CNOT(input, output)

Negation

output $=$ Not(CNOT(input, output))

The Deutsch Oracle: Solution

Solution

input = Hadamard(second bit of g(Hadamard(1), Hadamard(1))

input is 1 iff g constant.

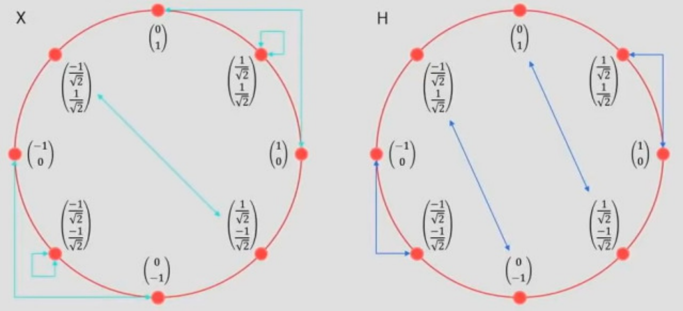
X
$\begin{pmatrix}0\\1\end{pmatrix}$
$\begin{pmatrix}\frac{-1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}-1\\0\end{pmatrix}$
$\begin{pmatrix}1\\0\end{pmatrix}$
$\begin{pmatrix}\frac{-1}{\sqrt{2}}\\\frac{-1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{-1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}0\\-1\end{pmatrix}$
H
$\begin{pmatrix}0\\1\end{pmatrix}$
$\begin{pmatrix}\frac{-1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}-1\\0\end{pmatrix}$
$\begin{pmatrix}1\\0\end{pmatrix}$
$\begin{pmatrix}\frac{-1}{\sqrt{2}}\\\frac{-1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{-1}{\sqrt{2}}\end{pmatrix}$
$\begin{pmatrix}0\\-1\end{pmatrix}$

Figure: The Map for Moving Q-Bits around

The Deutsch Oracle: How the CNOT Works

$$
\begin{aligned}
C\left(\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\end{pmatrix}\right) &= C\begin{pmatrix}1/2\\ -1/2\\ -1/2\\ 1/2\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\ -1\\ -1\\ 1\end{pmatrix}\\
&= \frac{1}{2}\begin{pmatrix}1\\ -1\\ 1\\ -1\end{pmatrix} = \left(\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\end{pmatrix}\right)
\end{aligned}
$$

The Deutsch Oracle: Why Care?

It turns out you can do this for functions with n inputs. Shor's algorithm uses this to factor.

References

WIRED (2018)
Quantum Computing Expert Explains Once Concept in 5 Levels of Difficulty
youtube.com `https://www.youtube.com/watch?v=OWJCfOvochA`

Microsoft Research (2018)
Quantum Computing for Computer Scientists
youtube.com `https://youtu.be/F_Riqjdh2oM`

Alan Tatourian (2018)
Quantum Computing for Computer Scientists
tatourian.blog `https://tatourian.blog/2018/09/01/quantum-computing-for-computer-scientists/`

Richard Cleve
An Introduction to Quantum Complexity Theory
University of Calgary
`https://cds.cern.ch/record/392006/files/9906111.pdf`

Erica Klarreich
Graduate Student Solves Quantum Verification Problem
Quanta Magazine `https://www.quantamagazine.org/graduate-student-solves-quantum-verification-problem-20181008/`

The End