A Practical Introduction to Quantum Computing: From Qubits to Quantum Machine Learning and Beyond

Elías F. Combarro
combarro@gmail.com

CERN openlab (Geneva, Switzerland) - University of Oviedo (Oviedo, Spain)

CERN - November/December 2020

Part I

Introduction: quantum computing... the end of the world as we know it?

I, for one, welcome our new quantum overlords

NEWS QUANTUM PHYSICS

Google officially lays claim to quantum supremacy

A quantum computer reportedly beat the most powerful supercomputers at one type of calculation

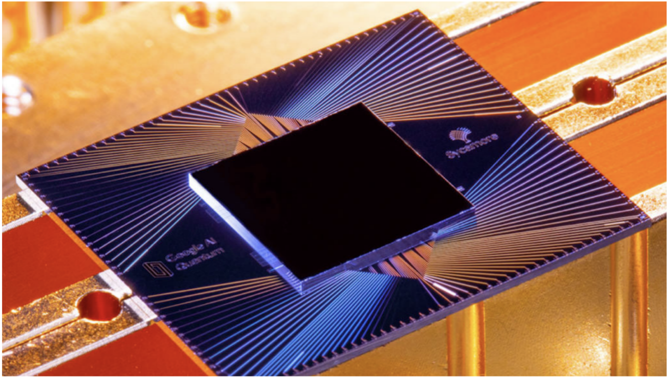


Image credits: sciencenews.org

Philosophy of the course

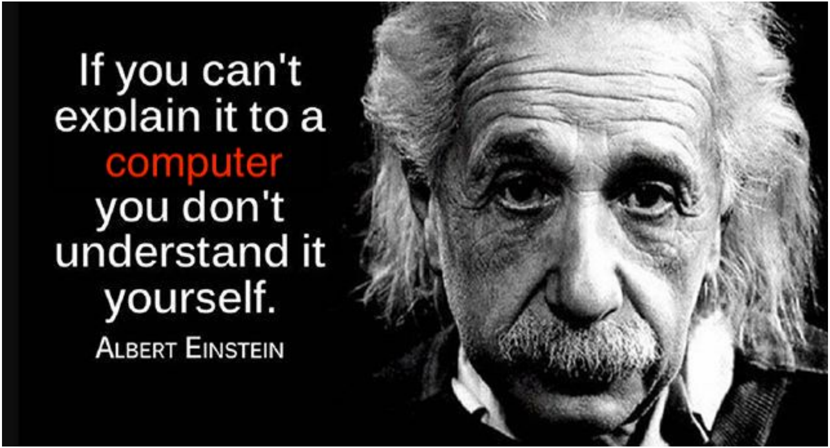


Image credits: Modified from an Instagram image by Bob MacGuffie

Tools and resources

- Jupyter Notebooks
  - Web application to create and execute notebooks that include code, images, text and formulas
  - They can be used locally (Anaconda) or in the cloud (mybinder.org, Google Colab...)
- IBM Quantum Experience
  - Free online access to quantum simulators (up to 32 qubits) and **actual quantum computers** (1, 5 and 15 qubits) with different topologies
  - Programmable with a visual interface and via different languages (python, qasm, Jupyter Notebooks)
  - Launched in May 2016
  - https://quantum-computing.ibm.com/

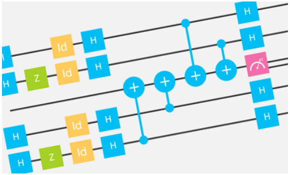


Image credits: IBM

Tools and resources (2)

- Quirk
  - Online simulator (up to 16 qubits)
  - Lots of different gates and visualization options
  - http://algassert.com/quirk
- D-Wave Leap
  - Access to D-Wave quantum computers
  - Ocean: python library for quantum annealing
  - Problem specific (QUBO, Ising model...)
  - https://www.dwavesys.com/take-leap

The shape of things to come

Image credits: Created with wordclouds.com

What is quantum computing?

Quantum computing

Quantum computing is a computing paradigm that exploits quantum mechanical properties (superposition, entanglement, interference...) of matter in order to do calculations

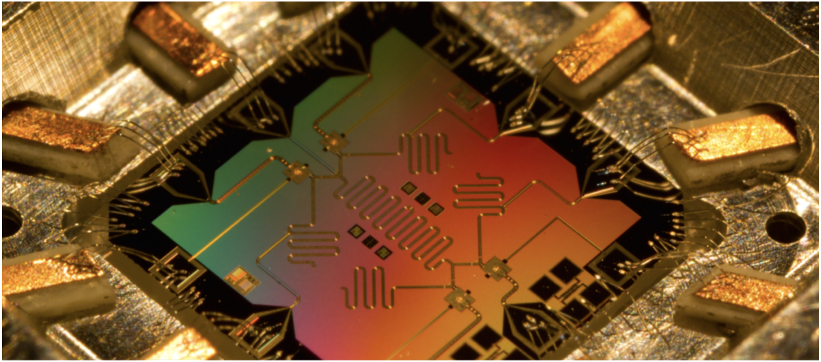


Image credits: Erik Lucero

Models of quantum computing

- There are several models of quantum computing (they're all equivalent)
  - Quantum Turing machines
  - **Quantum circuits**
  - Measurement based quantum computing (MBQC)
  - Adiabatic quantum computing
  - Topological quantum computing
- Regarding their **computational capabilities**, they are equivalent to classical models (Turing machines)

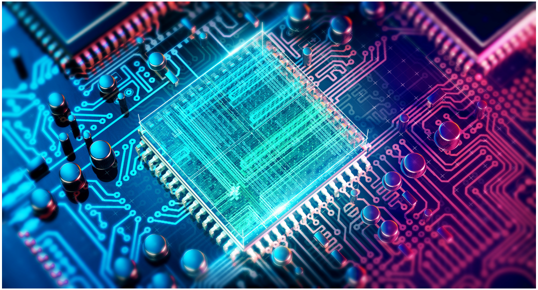


Image credits: Getty Images

Quantum and classical computational complexity

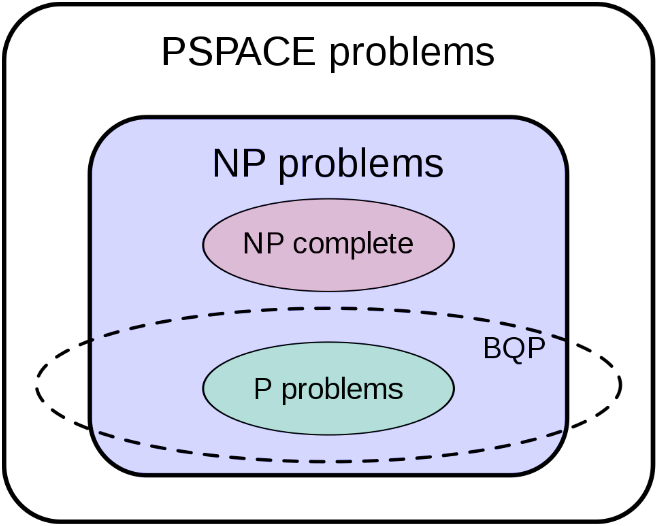


Image credits: wikipedia.org

What technologies are used to build quantum computers?

	Superconducting loops	**Trapped ions**	**Silicon quantum dots**	**Topological qubits**	**Diamond vacancies**
Company support	Google, IBM, Quantum Circuits	ionQ	Intel	Microsoft, Bell Labs	Quantum Diamond Technologies
Pros	Fast working. Build on existing semiconductor industry.	Very stable. Highest achieved gate fidelities.	Stable. Build on existing semiconductor industry.	Greatly reduce errors.	Can operate at room temperature.
Cons	Collapse easily and must be kept cold.	Slow operation. Many lasers are needed.	Only a few entangled. Must be kept cold.	Existence not yet confirmed.	Difficult to entangle.

Image credits: Graphic by C. Bickle/Science data by Gabriel Popkin

What is a quantum computer like?

Image credits: IBM

The Sounds of IBM: IBM Q

Programming a quantum computer

- Different frameworks and programming languages:
  - qasm
  - Qiskit (IBM)
  - Cirq (Google)
  - Forest/pyqil (Rigetti)
  - Q# (Microsoft)
  - Ocean (D-Wave)
  - ...
- Most of them for quantum circuit specification

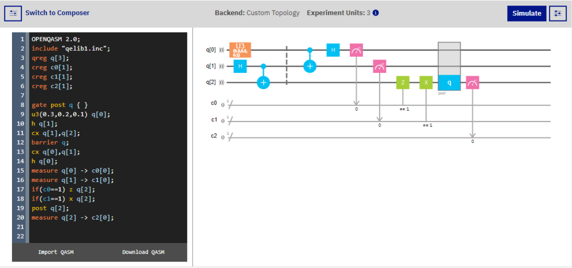


Image credits: IBM

What are the elements of a quantum circuit?

- Every computation has three elements: data, operations and results
- In quantum circuits:
  - Data = **qubits**
  - Operations = **quantum gates** (unitary transformations)
  - Results = **measurements**

Image credits: Adobe Stock

Part II

One-qubit systems: one qubit to rule them all

What is a qubit?

- A classical bit can take two different values (0 or 1). It is discrete.
- A qubit can "take" **infinitely** many different values. It is continuous.
- Qubits live in a **Hilbert vector space** with a basis of two elements that we denote $|0\rangle$ y $|1\rangle$.
- A generic qubit is in a **superposition**

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

where α and β are **complex numbers** such that

$$|\alpha|^2 + |\beta|^2 = 1$$

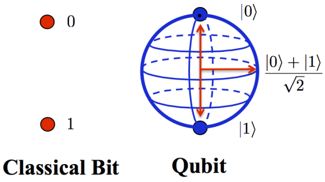


Image credits: https://prateekvjoshi.com/

Measuring a qubit

- The way to know the value of a qubit is to perform a measurement. However
  - The result of the measurement is random
  - When we measure, we only obtain one (classical) bit of information
- If we measure the state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.
- Moreover, the new state after the measurement will be $|0\rangle$ or $|1\rangle$ depending of the result we have obtained (wavefunction colapse)
- We cannot perform several independent measurements of $|\psi\rangle$ because we cannot copy the state (**no-cloning theorem**)

What are quantum gates?

- Quantum mechanics tells us that the evolution of an isolated state is given by the Schrödinger equation

$$H(t)|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle$$

- In the case of quantum circuits, this implies that the operations that can be carried out are given by unitary matrices. That is, matrices U of complex numbers verifying

$$UU^\dagger = U^\dagger U = I$$

  where $U^\dagger$ is the conjugate transpose of U.
- Each such matrix is a possible quantum gate in a quantum circuit

Reversible computation

- As a consequence, all the operations have an inverse: **reversible computing**
- Every gate has the same number of inputs and outputs
- We cannot directly implement some classical gates such as *or*, *and*, *nand*, *xor*...
- But we can simulate any classical computation with small overhead
- Theoretically, we could compute without wasting energy (Landauer's principle, 1961)

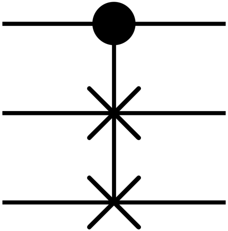


Image credits: wikipedia.org

One-qubit gates

- When we have just one qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, we usually represent it as a column vector $\begin{pmatrix}\alpha \\ \beta\end{pmatrix}$
- Then, a one-qubit gate can be identified with a matrix $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ that satisfies

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \overline{a} & \overline{c} \\ \overline{b} & \overline{d} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

  where $\overline{a}, \overline{b}, \overline{c}, \overline{d}$ are the conjugates of complex numbers a, b, c, d.

Action of a one-qubit gate

- A state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is transformed into

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix}$$

  that is, into the state $|\psi\rangle = (a\alpha + b\beta)|0\rangle + (c\alpha + d\beta)|1\rangle$
- Since U is unitary, it holds that

$$|(a\alpha + b\beta)|^2 + |(c\alpha + d\beta)|^2 = 1$$

The X or NOT gate

- The X gate is defined by the (unitary) matrix

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

- Its action (in quantum circuit notation) is

$$|0\rangle \;-\!\boxed{X}\!-\; |1\rangle$$

$$|1\rangle \;-\!\boxed{X}\!-\; |0\rangle$$

  that is, it acts like the classical *NOT* gate

- On a general qubit its action is

$$\alpha\,|0\rangle + \beta\,|1\rangle \;-\!\boxed{X}\!-\; \beta\,|0\rangle + \alpha\,|1\rangle$$

The Z gate

- The Z gate is defined by the (unitary) matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Its action is

$$|0\rangle \quad -\!\boxed{Z}\!- \quad |0\rangle$$

$$|1\rangle \quad -\!\boxed{Z}\!- \quad -|1\rangle$$

The H or Hadamard gate

- The H or Hadamard gate is defined by the (unitary) matrix

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

- Its action is

$$|0\rangle \; -\boxed{H}- \; \tfrac{|0\rangle+|1\rangle}{\sqrt{2}}$$

$$|1\rangle \; -\boxed{H}- \; \tfrac{|0\rangle-|1\rangle}{\sqrt{2}}$$

- We usually denote

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

and

$$|-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Other important gates

- Y gate

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

- S gate

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$$

- T gate

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

- The gates X, Y and Z are also called, together with the identity, the Pauli gates. An alternative notation is σ_X, σ_Y, σ_Z.

The Bloch sphere

- A common way of representing the state of a qubit is by means of a point in the surface of the Bloch sphere
- If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$ we can find angles γ, δ, θ such that
$$\alpha = e^{i\gamma}\cos\frac{\theta}{2}$$
$$\beta = e^{i\delta}\sin\frac{\theta}{2}$$
- Since an overall phase is physically irrelevant, we can rewrite
$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$
with $0 \leq \theta \leq \pi$ and $0 \leq \varphi < 2\pi$.

The Bloch sphere (2)

- From $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$ we can obtain spherical coordinates for a point in $\mathbb{R}^3$

$$(\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$$

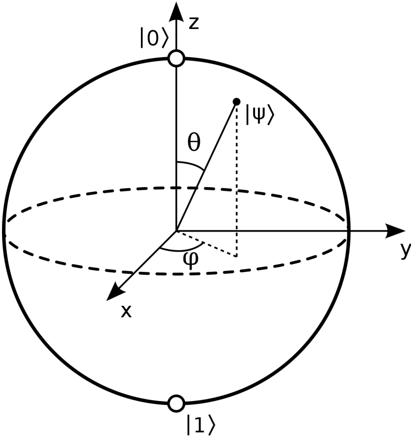


Image credits: wikipedia.org

Rotation gates

- We can define the following rotation gates

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

- Notice that $R_X(\pi) \equiv X$, $R_Y(\pi) \equiv Y$, $R_Z(\pi) \equiv Z$, $R_Z(\frac{\pi}{2}) \equiv S$, $R_Z(\frac{\pi}{4}) \equiv T$

Using rotation gates to generate one-qubit gates

- For any one-qubit gate U there exist a unit vector $r = (r_x, r_y, r_z)$ and an angle θ such that

$$U \equiv e^{-i\frac{\theta}{2} r\cdot\sigma} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2}(r_x X + r_y Y + r_z Z)$$

- For instance, choosing $\theta = \pi$ and $r = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$ we can see that

$$H \equiv e^{-i\frac{\theta}{2} r\cdot\sigma} = -i\frac{1}{\sqrt{2}}(X + Z)$$

- Additionally, it can also be proved that there exist angles α, β and γ such that

$$U \equiv R_Z(\alpha) R_Y(\beta) R_Z(\gamma)$$

Inner product, Dirac's notation and Bloch sphere

- The inner product of two states $|\psi_1\rangle = \alpha_1 |0\rangle + \beta_1 |1\rangle$ and $|\psi_2\rangle = \alpha_2 |0\rangle + \beta_2 |1\rangle$ is given by

$$\langle\psi_1|\psi_2\rangle = \begin{pmatrix}\overline{\alpha_1} & \overline{\beta_1}\end{pmatrix} \begin{pmatrix}\alpha_2 \\ \beta_2\end{pmatrix} = \overline{\alpha_1}\alpha_2 + \overline{\beta_1}\beta_2$$

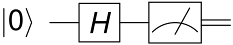
- Notice that $\langle 0|0\rangle = \langle 1|1\rangle = 1$ and $\langle 0|1\rangle = \langle 1|0\rangle = 0$
- This allows us to compute

$$\begin{aligned}\langle\psi_1|\psi_2\rangle &= \left(\overline{\alpha_1}\langle 0| + \overline{\beta_1}\langle 1|\right)\left(\alpha_2 |0\rangle + \beta_2 |1\rangle\right) \\ &= \overline{\alpha_1}\alpha_2 \langle 0|0\rangle + \overline{\alpha_1}\beta_2 \langle 0|1\rangle + \overline{\beta_1}\alpha_2 \langle 1|0\rangle + \overline{\beta_1}\beta_2 \langle 1|1\rangle \\ &= \overline{\alpha_1}\alpha_2 + \overline{\beta_1}\beta_2\end{aligned}$$

- Orthogonal states are antipodal on the Bloch sphere

Hello, quantum world!

- Our very first quantum circuit!



- After applying the H gate the qubit state is

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

- When we measure, we obtain 0 or 1, each with 50% probability: we have a circuit that generates perfectly uniform random bits!

Part III

The BB84 protocol: Alice and Bob's hotline

One-time pad: a Catch-22 situation

- Alice wants to send Bob a message m without Eve being able to learn anything about its content
- This can be achieved if Alice and Bob share in advance a string k of random bits:
  - Alice computes $x = m \oplus k$ and sends x to Bob
  - Eve cannot learn anything from x ($Pr(M = m|X = x) = Pr(M = m)$)
  - But Bob can recover m by computing $x \oplus k$
- The main problem is that k has to be as long as m and cannot be reused so... how to agree on k?

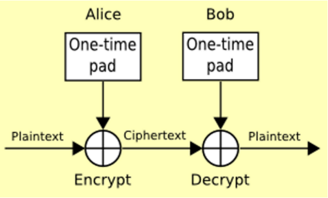
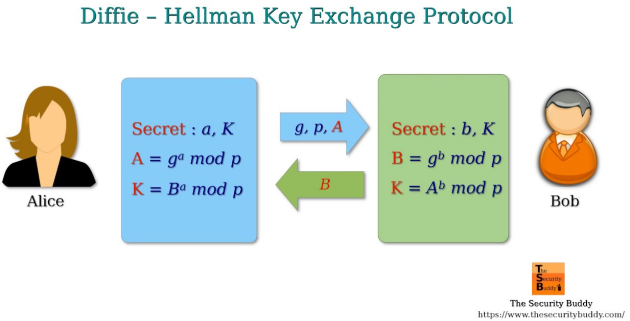


Image credits: nullprogram.com

The problem of key distribution

- Alice and Bob may share several keys for later use when they are together
- But... what if they cannot meet each other?
- There exist key distribution methods like the Diffie-Hellman protocol but...
    - They are not unconditionally secure (they usually rely on hardness assumptions)
    - In fact, DH can be broken with quantum computers!



BB84: Alice's part

- In 1984, Charles Bennett and Gilles Brassard proposed the first protocol for quantum key distribution (QKD)
- Alice generates a (private) string of random bits
- She could even do this with a quantum computer (H gate + measure)
- Then, for each bit she randomly chooses if she encodes it in the $\{|0\rangle, |1\rangle\}$ basis or in the $\{|+\rangle, |-\rangle\}$ basis (remember that $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$)
- She can easily do this by using H and X gates (recall that $H|0\rangle = |+\rangle, H|1\rangle = |-\rangle, X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$)
- Alice sends the resulting qubits to Bob (through a quantum but not necessarily secure channel)

BB84: Bob's part

- Each time Bob receives a qubit, he randomly decides whether he will measure it in the $\{|0\rangle, |1\rangle\}$ basis or in the $\{|+\rangle, |-\rangle\}$ basis
- He does this by applying (or not) the *H* gate before measuring
- He writes down the results and the basis he used:
  - If he used $\{|0\rangle, |1\rangle\}$ he writes down 0 if he gets $|0\rangle$ and 1 if he gets $|1\rangle$
  - If he used $\{|+\rangle, |-\rangle\}$ he writes down 0 if he gets $|+\rangle$ and 1 if he gets $|-\rangle$

BB84: Alice and Bob on the phone

- After this process, Alice and Bob talk on a classical channel (authenticated but not necessarily secure)
- Bob announces the bases he has used for the measurements and Alice announces the bases she used to code the bits
- Bob does NOT announce the results of his measurements
- For those bits in which Bob measured with the same basis that Alice used for coding, he has got the bit that Alice intended to send
- The rest are discarded (they will keep about half of the bits)

BB84: The protocol in an image

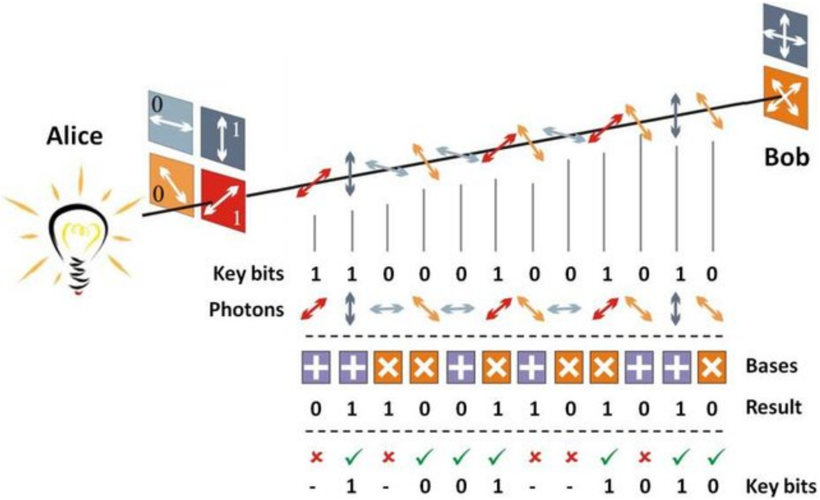


Image credits: A. Carrasco-Casado, V. Fernández, N. Denisenko

Eve tries to intercept and resend...

- Imagine Eve has access to the qubits that Alice sends to Bob
- Eve could try to measure and resend the qubit to Bob
- It is imposible for Eve to distinguish the four possibilities $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ because she does not know the basis that Alice has chosen
- If Eve chooses a basis at random, she will make an error half of the time and Alice and Bob may detect it (by sharing some of the bits of the key to check that they are equal)
- Eve cannot copy the qubits and wait to check the basis that Alice and Bob have used (no cloning theorem)
- Other more complex attacks are possible, but can be shown to fail

Information reconciliation and privacy amplification

- Because of imperfections in the channel and devices or because of eavesdropping, some of the bits that Alice and Bob have may be different
- They can conduct a process of information reconciliation (for instance, with the cascade protocol)
- After this phase (or even before), some information may have leaked to Eve
- Alice and Bob can perform privacy amplification (for instance, with randomness extractors)

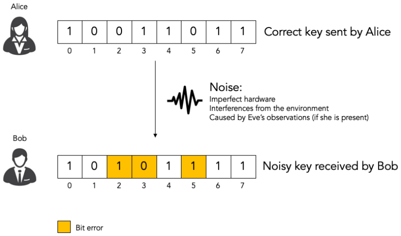


Image credits: hikingandcoding.wordpress.com

QKD at CERN

Image credits: https://arxiv.org/pdf/1203.4940.pdf

Kak's three-stage protocol

- Proposed by Kak in 2006
- It needs an authenticated quantum channel
- Suppose Alice wants to send $|x\rangle \in \{|0\rangle, |1\rangle\}$ to Bob:
  - Alice chooses θ_A at random and sends $R_Y(\theta_A)|x\rangle$ to Bob
  - Bob choose θ_B at random and sends $R_Y(\theta_B)R_Y(\theta_A)|x\rangle$ back to Alice
  - Alice applies $R_Y(-\theta_A)$ and sends

    $$R_Y(-\theta_A)R_Y(\theta_B)R_Y(\theta_A)|x\rangle = R_Y(\theta_B)|x\rangle$$

    to Bob
- Bob can now recover $|x\rangle$ by applying $R_Y(-\theta_B)$

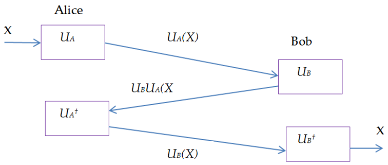


Image credits: wikipedia.org

The quantum one-time pad

- The analagous of the one-time pad with quantum operations would be to choose $a \in \{0, 1\}$ at random and encode $|x\rangle \in \{|0\rangle, |1\rangle\}$ as

$$X^a |x\rangle = |x \oplus a\rangle$$

- This cannot be extended to general qubits $|\psi\rangle$ because $X|+\rangle = |+\rangle$ and $X|-\rangle \equiv |-\rangle$
- We need to choose two bits a and b at random and encode $|\psi\rangle$ as

$$Z^b X^a |\psi\rangle$$

- Bob can now recover $|\psi\rangle$ by applying $X^a Z^b$
- It can be proved that this is unconditionally secure
- The QOTP is the basis of some blind quantum computing protocols

Other protocols that use independent qubits

- The use of independent qubits does not fully exploit the possibilities of quantum information, but there are some additional interesting applications
- For instance:
  - Other QKD protocols: B92, SARG04, Six-state protocol...
  - The concept of quantum money (Wiesner)
  - The Elitzur-Vaidman bomb tester
  - Quantum position verification
  - One-qubit classifier

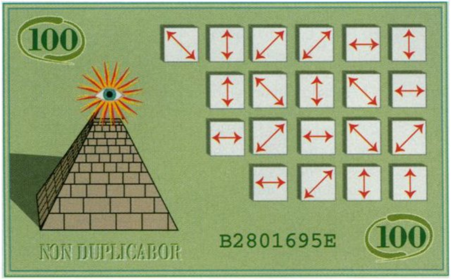


Image credits: The American Association for the Advancement of Science

Part IV

Two-qubit systems: more than the sum of their parts

Working with two qubits

- Each of the qubits can be in state $|0\rangle$ or in state $|1\rangle$
- So for two qubits we have four possibilities:

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

  that we also denote

$$|0\rangle |0\rangle, |0\rangle |1\rangle, |1\rangle |0\rangle, |1\rangle |1\rangle$$

  or

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

- Of course, we can have superpositions so a generic state is

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

  where α_{xy} are complex numbers such that

$$\sum_{x,y=0}^{1} |\alpha_{xy}|^2 = 1$$

Measuring a two-qubit system

- Suppose we have a state

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

- If we measure both qubits, we will obtain:
  - 00 with probability $|\alpha_{00}|^2$ and the new state will be $|00\rangle$
  - 01 with probability $|\alpha_{01}|^2$ and the new state will be $|01\rangle$
  - 10 with probability $|\alpha_{10}|^2$ and the new state will be $|10\rangle$
  - 11 with probability $|\alpha_{11}|^2$ and the new state will be $|11\rangle$
- It is an analogous situation to what we had with one qubit, but now with four possibilities

Measuring just one qubit in a two-qubit system

- If we have a state

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

  we can also measure just one qubit
- If we measure the first qubit (for the second one is analogous):
  - We will get 0 with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$
  - In that case, the new state of $|\psi\rangle$ will be

$$\frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

  - We will get 1 with probability $|\alpha_{10}|^2 + |\alpha_{11}|^2$
  - In that case, the new state of $|\psi\rangle$ will be

$$\frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$$

Two-qubit states and vector representation

- A general two-qubit quantum state is

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle$$

- We can represent with the column vector

$$\begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

- We can compute inner products by noticing that

$$\langle 00|00\rangle = \langle 01|01\rangle = \langle 10|10\rangle = \langle 11|11\rangle = 1$$

$$\langle 00|01\rangle = \langle 00|10\rangle = \langle 00|11\rangle = \cdots = \langle 11|00\rangle = 0$$

- A two-qubit quantum gate is a unitary matrix U of size 4×4

Tensor product of one-qubit gates

- The simplest way of obtaining a two-qubit gate is by having a pair of one-qubit gates A and B acting on each of the qubits
- In this case, the matrix for the two-qubit gate is the tensor product $A \otimes B$
- It holds that

$$(A \otimes B)(|\psi_1\rangle \otimes |\psi_2\rangle) = (A |\psi_1\rangle) \otimes (B |\psi_2\rangle)$$

- Of course, either A or B may be the identity
- This does NOT exhaust all posible two-qubit gates

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{bmatrix}$$

Image credits: wikipedia.org

The *CNOT* gate

- The *CNOT* (or controlled-*NOT* or *cX*) gate is given by the (unitary) matrix

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

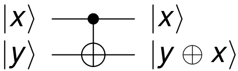
- If the first qubit is $|0\rangle$, nothing changes. If it is $|1\rangle$, we flip the second bit (and the first stays the same)
- That is:

$$|00\rangle \to |00\rangle \qquad |01\rangle \to |01\rangle$$

$$|10\rangle \to |11\rangle \qquad |11\rangle \to |10\rangle$$

Action of the *CNOT* gate

- Its action on $x, y \in \{0, 1\}$ is, then:



- This is an extremely important gate for it allows to:
  - Create entanglement (more on this soon)
  - Copy *classical* information, because:

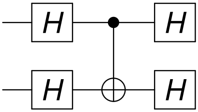
$$|00\rangle \rightarrow |00\rangle$$

$$|10\rangle \rightarrow |11\rangle$$

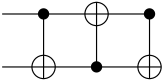
  - Construct other controlled gates

Equivalences with CNOT gates

- Sometimes, CNOT gates are not implemented between all pairs of qubits in a quantum computer
- We can use H gates to change the control and target of a CNOT gate



- We can swap states using three CNOT gates

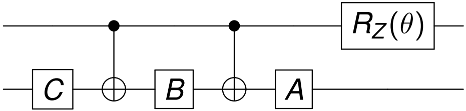


Constructing controlled gates by using the *CNOT* gate

- Any one-qubit gate U can be decomposed in the form

$$e^{i\theta}AXBXC$$

  with $ABC = I$
- Then, the circuit



  implements a U gate on the lower qubit controlled by the upper qubit

The no-cloning theorem

- There is **no** quantum gate that makes copies of an arbitrary (unknown) qubit
- The proof is easy: suppose we have a gate U such that $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$
- Then $U|00\rangle = |00\rangle$ and $U|10\rangle = |11\rangle$ and by linearity

$$U(\frac{1}{\sqrt{2}}(|00\rangle+|10\rangle)) = \frac{1}{\sqrt{2}}(U|00\rangle+U|10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

- But

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}} = (\frac{|0\rangle + |1\rangle}{\sqrt{2}})|0\rangle$$

  so we should have

$$U(\frac{|00\rangle + |10\rangle}{\sqrt{2}}) = \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \neq \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Quantum entanglement: the spooky action at a distance

- We say that a state $|\psi\rangle$ is a product state if it can be written in the form

$$|\psi\rangle = |\psi_1\rangle\,|\psi_2\rangle$$

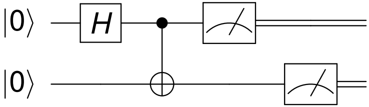
where $|\psi_1\rangle$ and $|\psi_2\rangle$ are two states (of at least one qubit)
- An **entangled** state is a state that is not a product state
- Example of entangled states (Bell states):

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$\frac{|01\rangle + |10\rangle}{\sqrt{2}} \qquad \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Hello, entangled world!

- We can construct (and measure) Bell states with simple circuits



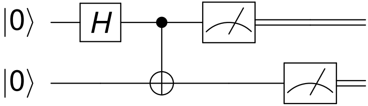
- Initially, the state of the system is $|00\rangle$
- After we apply the H gate, the state is

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

- When we apply the *CNOT* gate, the state changes to

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Hello, entangled world!



- Before we measure the first qubit, we have the state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$
- We will get 0 or 1, each with probability $\frac{1}{2}$
- Suppose we obtain 0. Then, the new state will be $|00\rangle$
- Then, when we measure the second qubit we will obtain 0 with probability 1!
- Also, if we obtain 1 in the first qubit, in the second we will also obtain 1!

Part V

The CHSH game: Nature isn't classical, dammit

The CHSH game

- Based in an inequality proposed in 1969 by Clauser, Horne, Shimony and Holt based on previous work by John Bell
- Alice and Bob receive bits x and y from a referee
- They have to respond with bits a and b
- They win if

$$a \oplus b = x \cdot y$$

- They can decide on a joint strategy beforehand, but they cannot communicate during the game

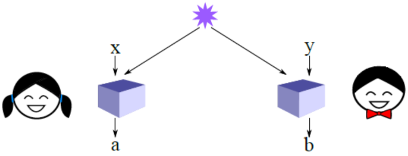


Image credits: quantumcomputing.stackexchange.com

Classical strategies for the CHSH game

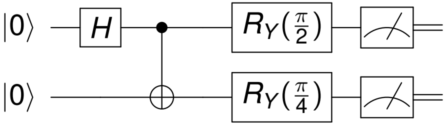
- Alice and Bob can win 75% of the time if they always answer '0'
- No other deterministic strategy can do better
- And probabilistic strategies are convex combinations of classical strategies so they cannot improve the 75% success rate

	$a = 0$	$a = 1$	$a = x$	$a = \neg x$
$b = 0$	$3/4$	$1/4$	$3/4$	$1/4$
$b = 1$	$1/4$	$3/4$	$1/4$	$3/4$
$b = y$	$3/4$	$1/4$	$1/4$	$3/4$
$b = \neg y$	$1/4$	$3/4$	$3/4$	$1/4$

Image credits: Ryan O'Donnell

Quantum strategy for the CHSH game

- Alice and Bob share a Bell pair $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ before the start of the game
- If Alice receives 0, she measures her qubit and ouputs the result
- If she receives 1, she applies $R_Y(\frac{\pi}{2})$ to her qubit and then she measures it
- If Bob receives 0, he applies $R_Y(\frac{\pi}{4})$. Else, he applies $R_Y(-\frac{\pi}{4})$.
- Then, he measures his qubit
- The probability of winning is now $\cos^2(\frac{\pi}{8}) \approx 0.85 > 0.75$



Some comments on the CHSH game

- It can be proved that $\cos^2(\frac{\pi}{8})$ is the highest possible success rate for a quantum strategy (Tsirelson's bound)
- The CHSH game can be used to rule out local realism
- Several experiments have been conducted, including:
  - Aspect et al. (1981-82)
  - Hensen et al. (2005) - Eliminate the locality and detection loopholes
- All of them agree with the predictions of quantum theory

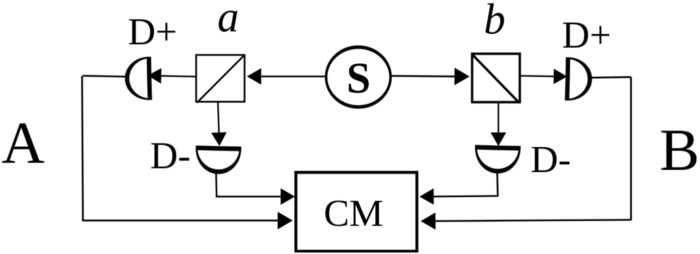


Image credits: George Stamatiou based on png file of C.Thompson

The GHZ game

- Introduced by Greenberger, Horne and Zeilinger
- A referee selects *rst* from $\{000, 011, 101, 110\}$ and sends *r* to Alice, *s* to Bob and *t* to *Charlie*
- They produce *a*, *b* and *c* and win if

$$a \oplus b \oplus c = r \vee s \vee t$$

- Classically, they can only win with 75% probability
- Quantumly, they can win every single time
  - They share the state

$$\frac{1}{2}(|000\rangle - |011\rangle - |101\rangle - |110\rangle)$$

  - They apply *H* to their qubit if the receive 1
  - They measure and return the answer
- This is sometimes called “quantum pseudo-telepathy” (Brassard, Cleve, Tapp)
- Both the CHSH and the GHZ game can be used for randomness certification (and expansion)

Part VI

Quantum teleportation and superdense coding: entangled up in blue

Quantum teleportation: Quantum me up, Scotty!

- Can Alice sent a qubit $|\psi\rangle$ to Bob it there is no quantum channel available?
- We are interested in the most general case, even if Alice does not know which state she has
- The problem can be solved if Alice and Bob share an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

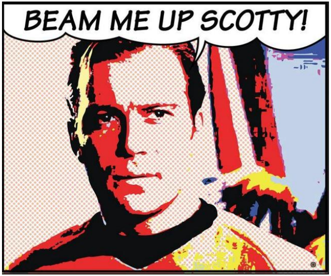


Image credits: www.geeksaresexy.net

Quantum teleportation: Alice's part

- Alice and Bob share an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
  - This can be done in advance
  - Or they can rely on a source that distributes entangled pairs
- Alice applies a CNOT gate to the qubit she wants to teleport $|\psi\rangle = a|0\rangle + b|1\rangle$ and to her part of the Bell pair. We will have

$$\frac{1}{\sqrt{2}}(a(|000\rangle + |011\rangle) + b(|110\rangle + |101\rangle))$$

- Alice further applies the H gate to the qubit she wants teleported. Then, we have

$$\frac{1}{2}\big(|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(b|0\rangle + a|1\rangle) \\ + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(-b|0\rangle + a|1\rangle)\big)$$

- Alice measures her two qubits and sends the result (two classical bits) to Bob (through a classical channel)

Quantum teleportation: Bob's part

- Bob uses the second bit received from Alice to decide if he applies X to his qubit
- And he uses the first bit to decide if he applies Z

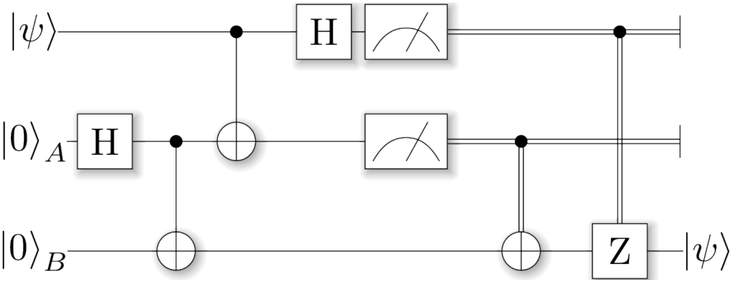


Image credits: ProjectQ

Quantum teleportation: some comments

- It is not matter that is teleported but information
- When Alice measure her qubit, she looses it (if not, we would be contradicting the no-cloning theorem)
- To teleport a qubit, we need two classical bits and one entangled pair:

$$2 bits + 1 ebit \geq 1 qubit$$

- Teleportation is not instantaneous, we need classical communication (no-communication theorem)
- Quantum teleportation has been shown experimentally (current record is 1,400 km)
- Demonstration of quantum teleportation in Quirk

Entanglement swapping

- Quantum teleportation can also be used with entangled qubits
- Alice shares a Bell pair with Bob and another one with Charlie
- In the figure, the top and bottom qubits belong to Alice. The second from the top belongs to Bob and the other to Charlie
- Alice teleports her top qubit to Charlie
- Now Bob's and Charlie's qubits are entangled (although maybe they were never in direct contact)

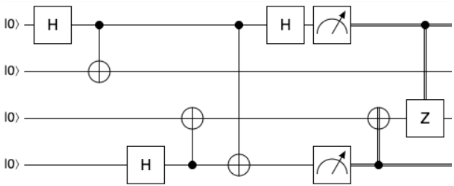
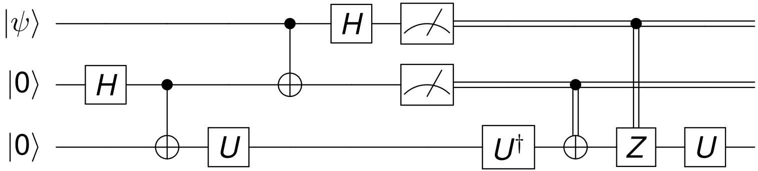


Image credits: Created with Quirk. Click here to access the circuit

Gate teleportation

- We can generalize the idea of quantum teleportation to teleport the action of gates
- With the circuit of the figure, we can apply gate U to an arbitrary state $|\psi\rangle$
- This is useful if preparing $\frac{1}{\sqrt{2}}(|0\rangle\, U\,|0\rangle + |1\rangle\, U\,|1\rangle)$ and applying $UXU^\dagger$, $UZU^\dagger$, $UZXU^\dagger$ are easy compared to applying U to a general qubit
- Such a situation can happen when $U = T$ in the context of fault-tolerant quantum computing



Superdense coding: two for the price of one (more or less)

- As we have seen, in the presence of a Bell pair, we can send a qubit with just two classical bits
- But... how many classical bits can we communicate with one qubit?
- Holevo's bound: the accesible information of one qubit is just one bit
- However, if Alice and Bob share in advance a Bell pair... we can send two bits of information with just one qubit!

$$1 qubit + 1 ebit \geq 2 bits$$

- This protocol is, in some sense, the inverse of quantum teleportation

Superdense coding: Alice's part

- Alice and Bob share a Bell pair in advance $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Alice wants to send to Bob two classical bits b_1 and b_2
- If $b_2 = 1$, she applies X to her qubit
- If $b_1 = 1$, she applies Z to her qubit
- Then, she sends her qubit to Bob

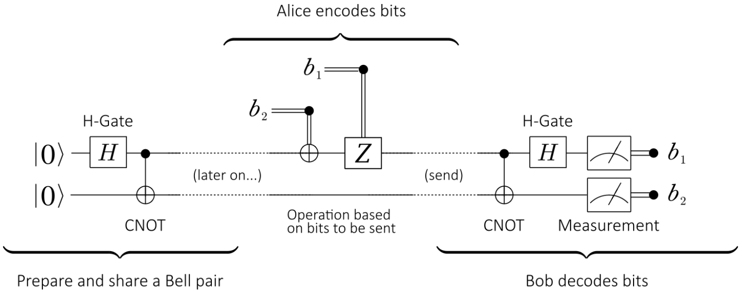


Image credits: www.quantum-bits.org

Superdense coding: Bob's part

- Bob receives Alice's qubit
- He applies a *CNOT* gate controlled by Alice's qubit
- He applies H to Alice's qubit
- He measures and recovers b_1 and b_2

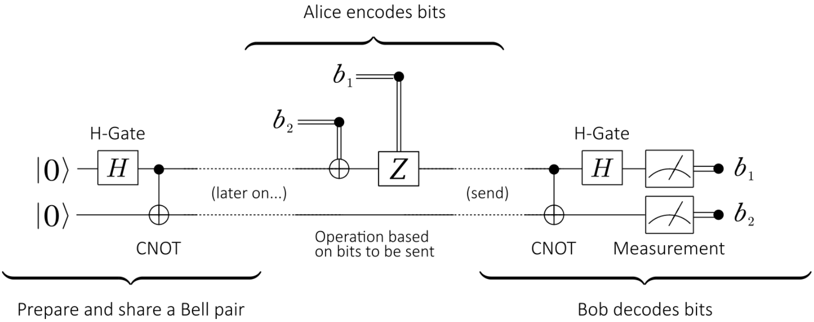


Image credits: www.quantum-bits.org

Superdense coding: an example

- Suppose Alice wants to send 11
- We start with $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- After Alice's operations, we will have $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$
- When Bob applies *CNOT* he obtains

$$\frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,|1\rangle$$

- And with the *H* gate he gets $|11\rangle$ that now he can measure

Part VII

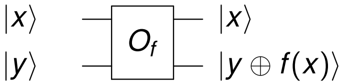
Deutsch's algorithm: the grandfather of all quantum algorithms

Deutsch's algorithm: statement of the problem

- In 1985, David Deutsch proposed a very simple algorithm that, nevertheless, hints at the capabilities of quantum computing
- The problem it solves is only of theoretical relevance and was later generalized in a joint work with Jozsa
- We are given a circuit (an **oracle**) that implements a one-bit boolean function and we are asked to determine whether the function is constant (returns the same value for all inputs) or balanced (returns 1 on one input and 0 on the other)
- Alternatively, we can think of the oracle as indexing a bit string of length two and we are asked to compute the XOR of the bits of the string
- In the classical case, we would need to consult the oracle twice, to compute both values of the function
- In the quantum case, we can make just one oracle call... but in superposition

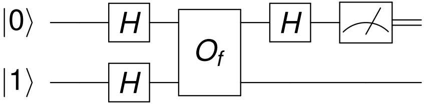
Deutsch's algorithm: the oracle

- An oracle is treated as a black box, a circuit whose interior we cannot know
- This circuit computes, in a reversible way, a certain function f (in our case, of just one input)
- For the computation to be reversible, it uses as many inputs as outputs and "writes the result" with an XOR



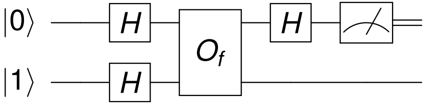
Deutsch's algorithm: the circuit

- The quantum circuit that we need to use to solve the problem is very simple



- If the function is constant, we will measure 0
- If the function is balanced, we will measure 1

Deutsch's algorithm: the magic



- The initial state is $|0\rangle\,|1\rangle$
- After the H the gates we have

$$\frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2}$$

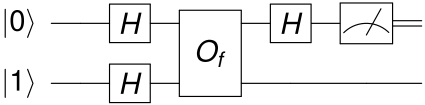
which is the same as

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{|1\rangle\,(|0\rangle - |1\rangle)}{2}$$

- When we apply the oracle, by linearity we obtain

$$\frac{|0\rangle\,(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)}{2} + \frac{|1\rangle\,(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)}{2}$$

Deutsch's algorithm: the magic (2)



- If $f(0) = 0$, we have

$$|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle = |0\rangle - |1\rangle$$

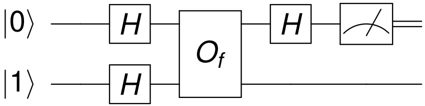
- However, if $f(0) = 1$ we get

$$|0 + f(0)\rangle - |1 \oplus f(0)\rangle = |0 \oplus 1\rangle - |1 \oplus 1\rangle = |1\rangle - |0\rangle = -(|0\rangle - |1\rangle)$$

- For $f(1)$ the situation is the same so the global state is

$$\frac{(-1)^{f(0)}\,|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{(-1)^{f(1)}\,|1\rangle\,(|0\rangle - |1\rangle)}{2}$$

Deutsch's algorithm: the magic (3)



- We can also write that state as

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{(-1)^{f(0)+f(1)}\,|1\rangle\,(|0\rangle - |1\rangle)}{2}$$

- So if $f(0) = f(1)$, we will have

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{|1\rangle\,(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2}$$

  and when we apply the last H and measure we obtain 0.
- But if $f(0) \neq f(1)$, the state is

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} - \frac{|1\rangle\,(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2}$$

  and, then, we obtain 1.

Deutsch's algorithm: some comments

- When we apply the oracle we have a phase kickback: we only act on one qubit, but it affects the whole state
- Deutch's algorithm exploits an interference phenomenon similar to that found in some physical experiments (double-slit experiment, Mach-Zehnder interferometer)

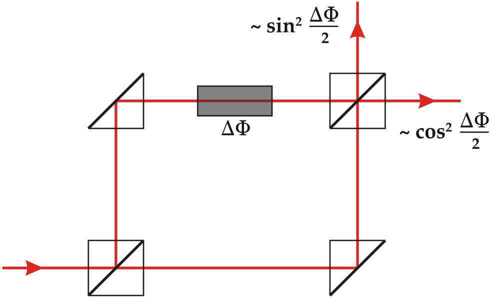


Image credits: Wikipedia

Part VIII

Multiqubit systems: growing up!

n-qubit systems

- When he have n qubits, each of them can be in state $|0\rangle$ and $|1\rangle$
- Thus, for the n-qubit state we have 2^n possibilities:

$$|00\ldots0\rangle\,,|00\ldots1\rangle\,,\ldots,|11\ldots1\rangle$$

or simply

$$|0\rangle\,,|1\rangle\,,\ldots,\left|2^n-1\right\rangle$$

- A generic state of the system will be

$$|\psi\rangle=\alpha_0\,|0\rangle+\alpha_1\,|1\rangle+\ldots+\alpha_{2^n-1}\left|2^n-1\right\rangle$$

where α_i are complex numbers such that

$$\sum_{i=0}^{2^n-1}|\alpha_i|^2=1$$

Measuring a n-qubit state

- Suppose we have the n-qubit state

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

- If we measure all its qubits, we obtain:
  - 0 with probability $|\alpha_0|^2$ and the new state will be $|0\ldots00\rangle$
  - 1 with probability $|\alpha_1|^2$ and the new state will be $|0\ldots01\rangle$
  - ...
  - $2^n - 1$ with probability $|\alpha_{2^n-1}|^2$ and the new state will be $|1\ldots11\rangle$
- It is analogous to what we had with one and two qubits, but now with 2^n possibilities

Measuring one qubit in a n-qubit state

- We have

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

- If we measure the j-th qubit
  - We will get 0 with probability

$$\sum_{i \in I_0} |\alpha_i|^2$$

    where I_0 is the set of numbers whose j-th bit is 0
  - In that case, the new state $|\psi\rangle$ will be

$$\frac{\sum_{i \in I_0} \alpha_i |i\rangle}{\sqrt{\sum_{i \in I_0} |\alpha_i|^2}}$$

  - The case in which we obtain 1 is analogous

n-qubit quantum gates

- A n-qubit state is

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

- We can represent it by the column vector

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix}$$

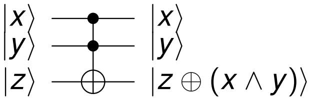
- To compute inner products with Dirac notation we only need to note that

$$\langle i|j\rangle = \delta_{ij}$$

- Thus, a n-qubit quantum gate is a unitary matrix U of size $2^n \times 2^n$

The Toffoli gate

- The Toffoli gate (or *CCNOT*) is a 3-qubit gate. Thus, it can be represented as a 8×8 matrix
- Its action on elements $x, y, z \in \{0, 1\}$ is:



- The Toffoli gate is **universal for classical logic**, and thus **any classical circuit can be simulated with a quantum circuit**
- However, the Toffoli gate, on its own, **is not universal for quantum computing** (and it is not even necessary, because it can be simulated with one and two-qubit gates)

Universal gates in quantum computing

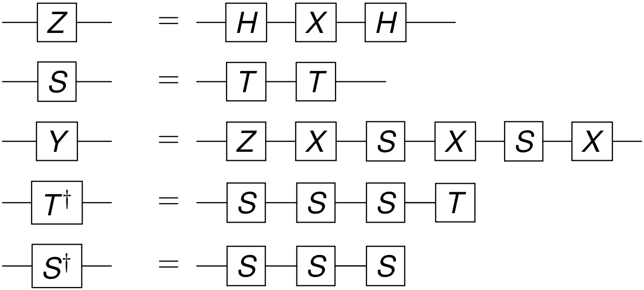
- The number of quantum gates (even for a single qubit) is uncountably infinite. Thus, no finite set of gates is universal in the classical sense
- However, we can obtain finite sets of gates that allow us to **approximate** any other gate as much as we want

Theorem

The one-qubit gates together with the CNOT gate are universal for quantum computing

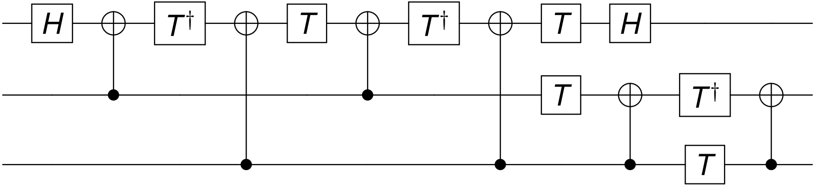
Theorem

The gates X, H, T and CNOT are universal for quantum computing



However, Z, S, Y, $S^\dagger$ and $T^\dagger$ are usually included among the available gates in most quantum computers (such as the ones in the IBM Q Experience).

Equivalence of the Toffoli gate



Part IX

Everything you always wanted to know about quantum parallelism but were afraid to ask

Urban legends about quantum parallelism

- But... don't quantum computers try all 2^n possibilities in parallel?
- The answer is... yes *and* no (this is *quantum* computing after all!)

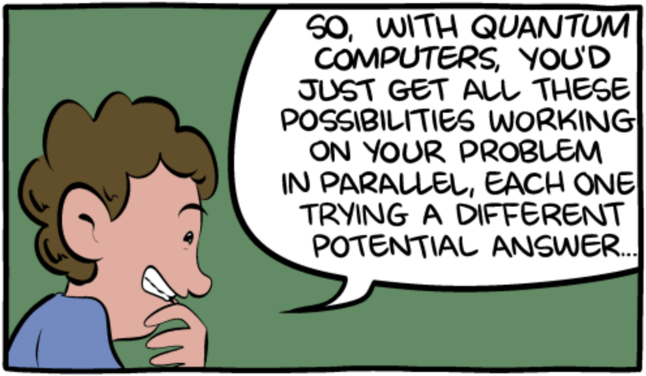
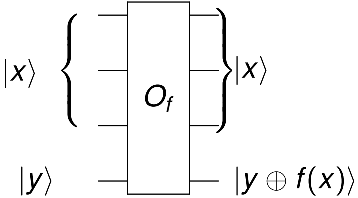


Image credits: The Talk, by Scott Aaronson and Zach Weinersmith

Evaluating a function: querying the oracle

- As we know, in quantum computing every gate is reversible
- To compute a function f we keep the inputs unchanged and *xor* the result to the output qubits
- This type of circuit is called and oracle for f (we have already used an oracle for a one-bit function in Deutsch's algorithm)



Evaluating a function in parallel: the superposition hocus-pocus

- Suppose that we have an oracle O_f for a function $f(x)$ with a one-bit input
- We know that, using the H gate, we can put a qubit in superposition
- If we start with the state $|0\rangle |0\rangle$ and we apply H on the first qubit, we will have

$$\frac{1}{\sqrt{2}} |0\rangle |0\rangle + \frac{1}{\sqrt{2}} |1\rangle |0\rangle$$

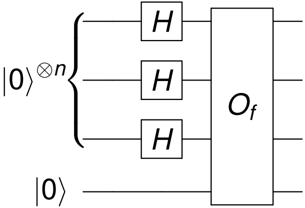
- If we now apply O_f, by linearity we have

$$\frac{1}{\sqrt{2}} |0\rangle |f(0)\rangle + \frac{1}{\sqrt{2}} |1\rangle |f(1)\rangle$$

- We have evaluated the function on two different inputs with just one call!

Evaluating a function in parallel: the tensor-product abracadabra

- We can do something similar with a function $f(x_1, x_2, \dots, x_n)$ on n-variables by using the following circuit



- When we apply the H gates we obtain

$$\frac{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) \cdots (|0\rangle + |1\rangle) |0\rangle}{\sqrt{2^n}}$$

Evaluating a function in parallel: the tensor-product abracadabra (2)

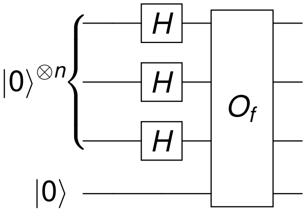
- If we expand the product we get

$$\frac{(|0\dots 0\rangle + |0\dots 1\rangle + \dots + |1\dots 1\rangle)\,|0\rangle}{\sqrt{2^n}} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|0\rangle$$

- And, when we apply the oracle, we will get the state

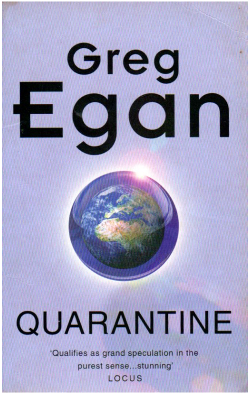
$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|f(x)\rangle$$

- An exponential number of function evaluations with just one call!



Quantum parallelism vs. non-deterministic machines

- With a non-deterministic machine, we could choose at will some value f
- This would allow us to solve *NP*-complete problems
- A similar idea is used in the plot of *Quarantine*, a science-fiction novel by Greg Egan



All that glitters ain't gold

- And now... how do we retrieve the values $f(x)$?
- To obtain a result, we need to perform a measurement
- But then we will get a state of the form

$$|c\rangle\,|f(c)\rangle$$

- That is, we only obtain the result of the function for a randomly chosen input (this may be even worse than classically evaluating the function)

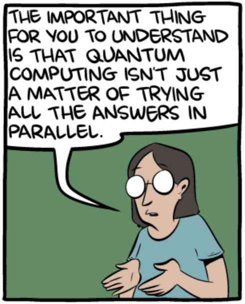


Image credits: The Talk, by Scott Aaronson and Zach Weinersmith

Interferences come to the rescue

- How can we use the 2^n evaluations to extract useful information?
- One possibility is... to produce interferences!
- The amplitudes of some states can be negative
- If we manage to “annihilitate” the amplitudes of states we are not interested in, the probability of obtaining the answer that we need will grow
- This is, in general, no easy task, but we know how to achieve it in some interesting cases

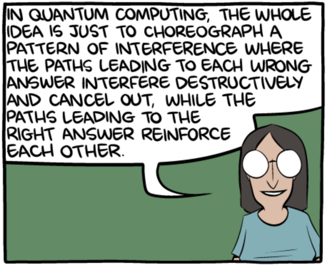


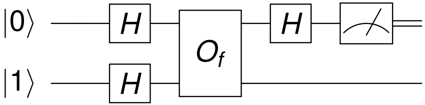
Image credits: The Talk, by Scott Aaronson and Zach Weinersmith

Part X

The Deutsch-Jozsa algorithm: a very fast way of solving a problem that nobody asked to solve

Reminder: Deutsch's algorithm

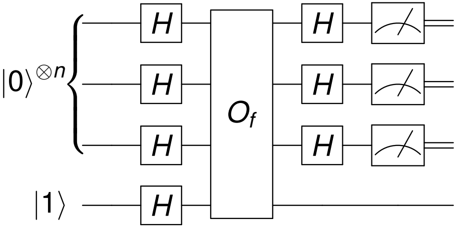
- We have an oracle O_f for a boolean function $f(x)$
- f can be constant (returns the same value for all inputs) or balanced (returns 1 on one input and 0 on the other)
- Distinguishing one situation from the other requires, in the classical case, evaluating the function on the two possible inputs
- With a quantum computer, we can solve the problem with just one call to O_f
- The key is to use quantum parallelism together with interference



Upping the ante: the Deutsch-Jozsa algorithm

- The Deutsch-Jozsa algorithm solves a type of problem called **promise problem**
  - We are given a boolean function $f(x_1, \ldots, x_n)$
  - We are **promised** that f is either constant (always 0 or 1) or balanced (0 for half of the inputs and 1 for the rest)
  - We have to decide which of the two cases we are in by calling the function as few times as possible
- With a classical deterministic algorithm we need (in the worst case) $2^{n-1} + 1$ calls to f
- With the Deutsch-Jozsa quantum algorithm it is enough to evaluate f **just once**

Circuit for the Deutsch-Jozsa algorithm



Steps in the Deutsch-Jozsa algorithm

1. We create the state $|0\dots0\rangle\,|1\rangle$
2. We use Hadamard gates to create the superposition

$$\sum_{x\in\{0,1\}^n}\frac{1}{\sqrt{2^{n+1}}}\,|x\rangle\,(|0\rangle-|1\rangle)$$

3. We apply the oracle, getting

$$\sum_{x\in\{0,1\}^n}\frac{1}{\sqrt{2^{n+1}}}\,|x\rangle\,(|0\oplus f(x)\rangle-|1\oplus f(x)\rangle)=$$

$$\sum_{x\in\{0,1\}^n}\frac{(-1)^{f(x)}}{\sqrt{2^{n+1}}}\,|x\rangle\,(|0\rangle-|1\rangle)$$

Steps in the Deutsch-Jozsa algorithm (2)

4. We apply again Hadamard gates to the n first qubits and we obtain

$$\sum_{y\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)+x\cdot y}}{2^n\sqrt{2}} |y\rangle (|0\rangle - |1\rangle)$$

5. Finally, we measure the n first qubits.
6. If the function is constant, we will obtain $|0\rangle$. Otherwise (if the function is balanced), we will get a string different from $|0\rangle$.

Correctness of the algorithm

- The probability of measuring $|0\rangle$ is exactly

$$\left(\sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)+x\cdot 0}}{2^n}\right)^2 = \left(\sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}}{2^n}\right)^2$$

- If f is constant, the sum is 1
- If f is balanced, the sum is 0

Some comments on the Deutsch-Jozsa algorithm

- The problem we have solved is academical, with no practical interest
- But... it shows how quantum computing can obtain global information about a function with just one evaluation
- The key is to use:
    - Quantum parallelism (because of superposition)
    - Interference (constructive and destructive)
- Similar ideas are used in other algorithms, like the Bernstein-Vazirani and Simon methods

Part XI

Grover's algorithm: finding the needle in the haystack

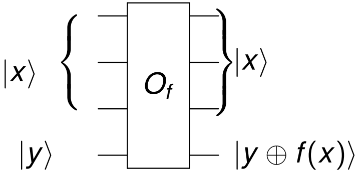
Statement of the problem

- Grover's algorithm is used to solve search problems
- Imagine we have an unsorted list of N elements
- One of them verifies a certain condition and we want to find it
- Any classical algorithm requires $O(N)$ queries to the list in the worst case
- Grover's algorithm can find the element with $O(\sqrt{N})$ queries

Image credits: Downloaded from www.usnewsglobaleducation.com

The oracle

- As in Deutsch-Jozsa's algorithm, we will use an oracle
- This oracle computes the function $f : \{0,1\}^n \Rightarrow \{0,1\}$ (with $N = 2^n$)
- The element we want to find is the one that verifies $f(x) = 1$



The idea behind the algorithm

- Grover's algorithm is based on the idea of **inversion about the mean**

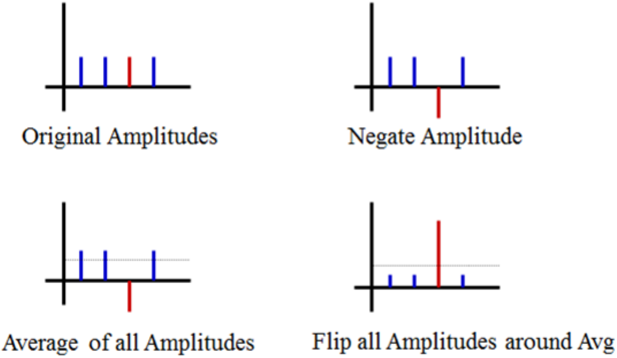
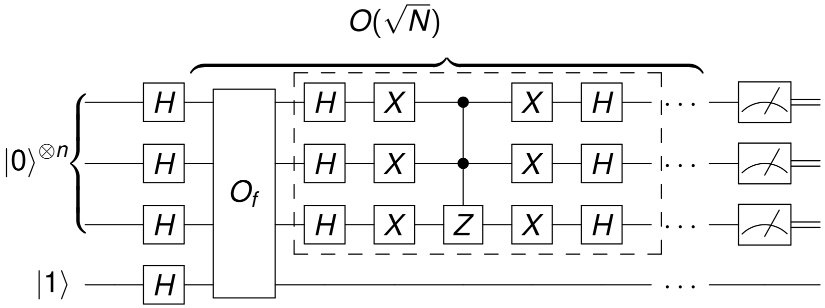


Image credits: quantumcomputing.stackexchange.com

Grover's algorithm

- Grover's algorithm performs $O(\sqrt{N})$ iterations, each one consisting in an oracle query and a call to Grover's diffusion operator
- The oracle "marks" those states that verify the condition
- The diffusion operator "amplifies" the amplitudes of the marked states



Grover's algorithm as a rotation

- Let us denote by $|x_1\rangle$ the marked element
- Then, the initial state of the upper n qubits is

$$\sqrt{\frac{N-1}{N}}\,|x_0\rangle + \sqrt{\frac{1}{N}}\,|x_1\rangle$$

where

$$|x_0\rangle = \sum_{x\in\{0,1\}^n, x\neq x_1} \sqrt{\frac{1}{N-1}}\,|x\rangle$$

- We can choose $\theta \in (0, \frac{\pi}{2})$ such that

$$\cos\theta = \sqrt{\frac{N-1}{N}} \qquad \sin\theta = \sqrt{\frac{1}{N}}$$

Grover's algorithm as a rotation (2)

- Define D to be Grover's diffusion operator and $G = DO_f$
- It can be shown that G acts on the 2-dimensional space spawned by $|x_0\rangle$ and $|x_1\rangle$ as a rotation of angle 2θ
- That is

$$G\,|x_0\rangle = \cos 2\theta\,|x_0\rangle + \sin 2\theta\,|x_1\rangle$$

$$G\,|x_1\rangle = -\sin 2\theta\,|x_0\rangle + \cos 2\theta\,|x_1\rangle$$

$$|x_0\rangle = \sum_{x\in\{0,1\}^n, x\neq x_1} \sqrt{\frac{1}{N-1}}\,|x\rangle$$

- Since the initial state is $\cos\theta\,|x_0\rangle + \sin\theta\,|x_1\rangle$, after m iterations we will have

$$\cos\,(2m+1)\theta\,|x_0\rangle + \sin\,(2m+1)\theta\,|x_1\rangle$$

Grover's algorithm as a rotation (3)

- In order to obtain $|x_1\rangle$ with high probability when we measure we need

$$(2m+1)\theta \approx \frac{\pi}{2}$$

and this gives

$$m \approx \frac{\pi}{4\theta} - \frac{1}{2}$$

- Since

$$\sin\theta = \sqrt{\frac{1}{N}}$$

we will have

$$\theta \approx \sqrt{\frac{1}{N}}$$

and then we can choose

$$m = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor$$

The case with multiple marked elements

- If the number of marked elements is $k > 1$, a similar argument can be made by defining

$$|x_0\rangle = \sum_{f(x)=0} \sqrt{\frac{1}{N-k}}\,|x\rangle$$

$$|x_1\rangle = \sum_{f(x)=1} \sqrt{\frac{1}{k}}\,|x\rangle$$

- In this case

$$\sin\theta = \sqrt{\frac{k}{N}}$$

and if $k \ll N$ we can choose

$$m = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{k}} \right\rfloor$$

The case with unknown number of marked elements

- If we do not know how many elements are marked, we can still user Grover's algorithm
- We can use Grover's circuit combined with the Quantum Fourier Transform to estimate k
- Or we can choose m at random. For instance:
  - Uniformly from the set $\{0, \ldots, \lceil \sqrt{N} + 1 \rceil\}$
  - With an incremental scheme, starting with an upper bound for m of $b = 1$ and increasing it exponentially up to $\sqrt{N}$
- In all the cases, it can be shown that a marked element will be found with high probability with $O(\sqrt{N})$ queries to the oracle

Some comments on Grover's algorithm

- When we measure, we will obtain x such that $f(x) = 1$ with probability depending on:
  - The number m of iterations
  - The fraction of values x that satisfy the condition
- If we perform too many iterations, we can overshoot and not find a marked element
- On the other hand, if $k = \frac{N}{4}$ then one iteration will find a marked element with certainty
- Grover's algorithm can be used to find minima of functions (Dürr-Hoyer's algorithm)
- It can be shown that no other quantum algorithm can obtain more than a quadratic speed-up over over classical algorithms in the same setting
- A generalization of Grover's algorithm called Amplitude Amplification can be used with states prepared by an arbitrary unitary A

Part XII

Shor's algorithm: breaking the Internet

Shor's algorithm and factoring

- Shor's algorithm is, probably, the most famous quantum algorithm
- It finds a factor of a n-bit integer in time $O(n^2(\log n)(\log\log n))$
- The best classical algorithm that we know of for the same task needs time $O(e^{cn^{\frac{1}{3}}(\log n)^{\frac{2}{3}}})$
- Dramatic consequences for current cryptography (RSA)

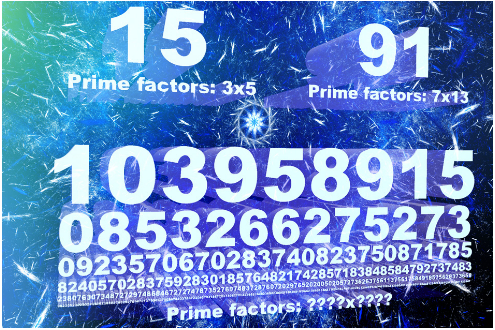


Image credits: Jose-Luis Olivares/MIT

Steps of Shor's algorithm

1. Given N, check that N is not a prime or power of a prime. If it is, stop.
2. Choose $1 < a < N$ at random
3. If $b = gcd(a, N) > 1$, output b and stop
4. Find the order of a mod N, that is, $r > 0$ such that $a^r \equiv 1 \mod N$
5. If r is odd, go to 2
6. Compute
$$x = a^{\frac{r}{2}} + 1 \mod N$$
$$y = a^{\frac{r}{2}} - 1 \mod N$$
7. If $x = 0$, go to 2. If $y = 0$, take $r = \frac{r}{2}$ and go to 5.
8. Compute $p = gcd(x, N)$ and $q = gcd(y, N)$. At least one of them will be a non-trivial factor of N

Correctness of Shor's algorithm

- We know that

$$a^r \equiv 1 \mod N$$

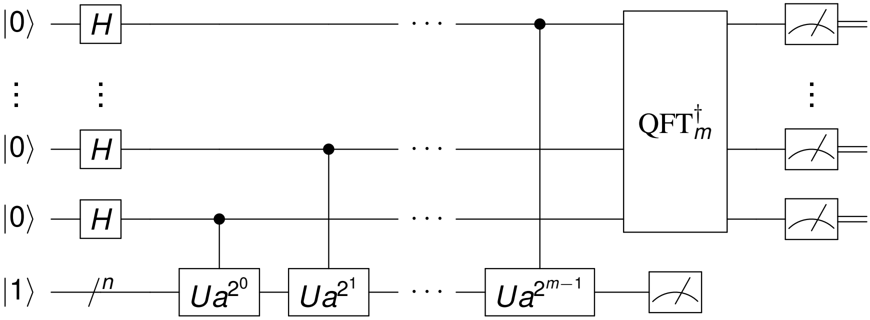
- Thus

$$x \cdot y \equiv (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) \equiv (a^r - 1) \equiv 0 \mod N$$

- This means that $x \cdot y$ is a multiple of N
- Since neither x nor y are multiples of N, either p or q divides N
- It can be proved that step 8 will be reached with high probability

Implementation of Shor's algorithm

- Every step but number 4 are carried out on a classical computer (efficient algorithms exist)
- For step 4, there exists a quantum circuit with a number of gates that is polynomial on n (the number of bits of N)



Preparing a periodic sequence

- The first part of the circuit computes

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle \, |a^x \mod N\rangle$$

- When we measure the bottom qubits, we obtain

$$\frac{1}{\sqrt{|C|}} \sum_{x \in C} |x\rangle \, |c\rangle$$

  where c is some value in $\{0, \ldots, N-1\}$ and $C = \{x : a^x \mod N = c\}$.

Preparing a periodic sequence (2)

- For example, if $a = 2$, $N = 5$, $m = 4$, we would have

  $$\frac{1}{4}\left(|0\rangle\,|1\rangle + |1\rangle\,|2\rangle + |2\rangle\,|4\rangle + |3\rangle\,|3\rangle + |4\rangle\,|1\rangle + \ldots + |15\rangle\,|3\rangle\right)$$

  and when we measure we could obtain, for instance

  $$\frac{1}{2}\left(|1\rangle\,|2\rangle + |5\rangle\,|2\rangle + |9\rangle\,|2\rangle + |13\rangle\,|2\rangle\right)$$

- Notice that the values of the first register are exactly 4 units apart and that $2^4 = 1 \mod 5$.
- In general, we will obtain values that are r units apart, where $a^r = 1 \mod N$.

Measuring the period

- To retrieve the period r we use the (inverse) of the Quantum Fourier Transform (QFT)
- Two properties of the QFT are central here:
  - Shift-invariance (up to an unobservable phase)
  - QFT transforms sequences with period r into sequences with period $\frac{M}{r}$ (where $M = 2^m$)
- After the use of the inverse QFT, we can measure a value of the form $\frac{Mc}{r}$ with high probability and, from it, obtain r

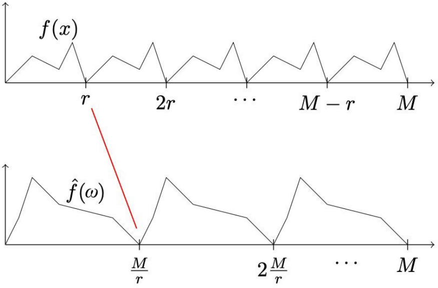


Image credits: Umesh Vazirani

Quantum Fourier Transform: definition and circuit

- The QFT of order m is the unitary transformation defined by

$$QFT\,|j\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{\frac{2\pi ijk}{2^m}} |k\rangle$$

- The circuit in the figure implements the QFT
- The R_k gates in the circuit are what we call $R_Z(\frac{2\pi}{2^k})$
- The number of gates is quadratic in m, an exponential speed-up over the classical case (FFT)
- For Shor, m can be chosen to be about $2n$

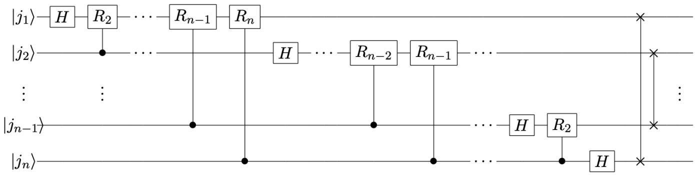


Image credits: Jurgen Van Gael

Using the QFT for phase estimation

- Suppose we are given a unitary operation U and one of its eigenvectors $|\psi\rangle$
- We know that there exists $\theta \in [0, 1)$ such that $U|\psi\rangle = e^{2\pi i\theta}$
- We can estimate θ with the circuit shown below
- With the first part, we will obtain $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i\theta k} |k\rangle$
- By using the inverse QFT we can measure $j \approx 2^n\theta$

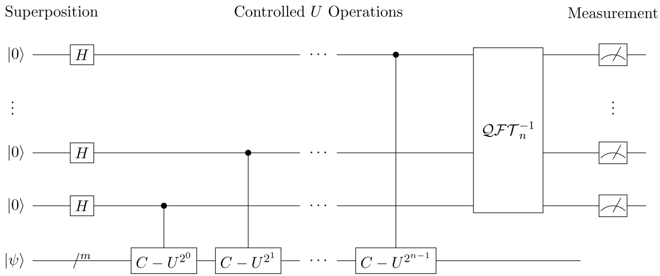


Image credits: Wikipedia

Shor's algorithm as a particular case of quantum phase estimation

- Clearly, the circuit used in Shor's algorithm is a case of quantum phase estimation
- It can be shown that the (unitary) operation of modular mutiplication by a has eigenvalues

$$e^{2\pi i \frac{k}{r}} \qquad k = 0, \ldots, r-1$$

  where r is the period of a
- It is not easy to prepare one of the eigenvectors $|\psi_k\rangle$ of the unitary operation
- But we use the fact that

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$$

- We will then measure a value close to $\frac{2^m k}{r}$ for some k

Using quantum phase estimation to count the number of marked elements

- We can use Grover's algorithm together with the QFT to count the number of elements marked by a boolean function
- The eigenvalues of Grover's operator are $e^{\pm 2i\theta}$ where $\sin\theta = \sqrt{\frac{k}{N}}$
- Then, with quantum phase estimation we can recover k, the number of marked elements

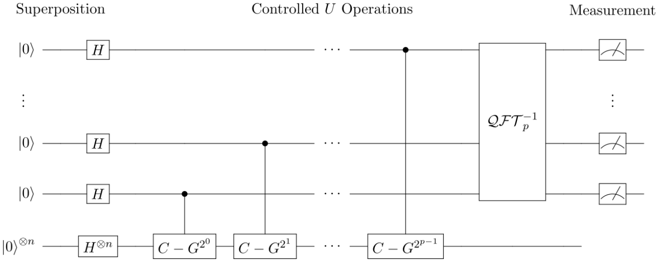


Image credits: Wikipedia

HHL: Applying quantum phase estimations to solve linear systems of equations

- A quantum algorithm proposed in 2009 by Harrow, Hassidim and Lloyd can be used to solve linear systems of equations
- The main steps of the algorithm are
  - Computation of the eigenvalues (quantum phase estimation)
  - Inversion of the eigenvalues
  - Uncomputation of the eigenvalues (inverse of quantum phase estimation)

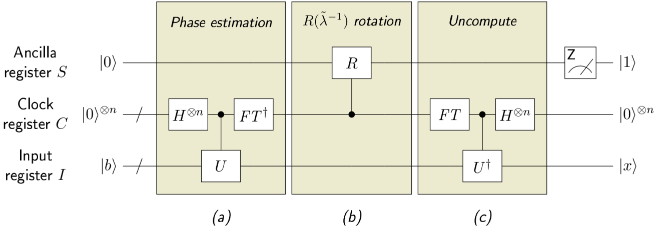


Image credits: Niel de Beaudrap

Visualizing Shor's algorithm with Qirk

- Case $a = 2$ and $N = 15$
- Case $a = 4$ and $N = 15$
- Case $a = 14$ and $N = 15$
- Case $a = 26$ and $N = 55$

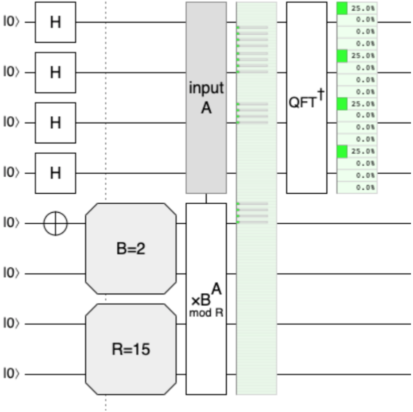


Image credits: Created with Quirk

Part XIII

Quantum annealing: when time is gold

The maximum cut or Max-Cut problem

- Consider the problem of dividing the vertices of a graph into two sets such that the number of edges with extremes in both sets is the maximum possible
- It can be proved that this problem, called “maximum cut” or “Max-Cut”, is NP-hard
- It is also APX-Hard and thus there is no (classical) polynomial-time approximation scheme (PTAS) which gets arbitrarily close to the solution (unless $P = NP$)

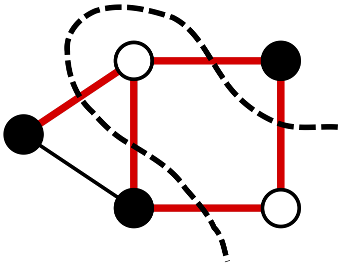


Image credits: Wikipedia.org

Stating Max-Cut with *spins*

- We can identify each vertex i of the graph with a variable Z_i
- We assign value 1 to the vertices of one group and -1 to the others
- Then, if E is the set of edges, the problem can be stated as

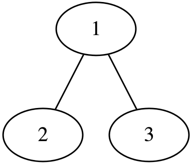
$$\text{Minimize} \sum_{(i,j)\in E} Z_i Z_j$$

since vertices in different groups contribute -1 and vertices of the same group contribute 1

Example of Max-Cut problem

- For the graph of the figure we need to minimize

$$H = Z_1 Z_2 + Z_1 Z_3$$



- By inspection (or enumerating the eight possibilities) it is easy to see that the solutions are 011 and 100

Enter quantum computing

- Remember that the matrix of gate Z is

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and that the vector $|0\rangle$ has coordinates

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

- Then

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

- Using Dirac notation, we can denote this by

$$\langle 0| Z |0\rangle = 1$$

Enters quantum computing (2)

- Analogously

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- And thus

$$\langle 1| Z |1\rangle = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1$$

- If we have more qubits, we evaluate each product independently and multiply the results. For instance:

$$\langle 01| Z_1 Z_2 |01\rangle = (\langle 0| Z_1 |0\rangle) \cdot (\langle 1| Z_2 |1\rangle) = 1 \cdot (-1) = -1$$

and

$$\langle 101| Z_1 Z_3 |101\rangle = (\langle 1| Z_1 |1\rangle) \cdot (\langle 1| Z_3 |1\rangle) = (-1) \cdot (-1) = 1$$

Back to the Max-Cut example

- We had the Max-Cut problem given by

$$H = Z_1 Z_2 + Z_1 Z_3$$

- We can identify a possible cut with $|011\rangle$ (vertices 2 and 3 in one set and 1 in the other) and evaluate its cost by

$$\begin{aligned} \langle 011| H |011\rangle &= \langle 011| (Z_1 Z_2 + Z_1 Z_3) |011\rangle \\ &= \langle 011| Z_1 Z_2 |011\rangle + \langle 011| Z_1 Z_3 |011\rangle = -1 + (-1) = -2 \end{aligned}$$

- Analogously

$$\begin{aligned} \langle 010| H |010\rangle &= \langle 010| (Z_1 Z_2 + Z_1 Z_3) |010\rangle \\ &= \langle 010| Z_1 Z_2 |010\rangle + \langle 010| Z_1 Z_3 |010\rangle = -1 + 1 = 0 \end{aligned}$$

Hamiltonians, Hamiltonians everywhere

- Then, we are interested in finding a (basis) quantum state $|x\rangle$ such that

$$\langle x| H |x\rangle$$

  is minimum, with $H = \sum_{(i,j)\in E} Z_i Z_j$ the cost function of the Max-Cut problem
- This is a particular case of a very important problem: finding the **ground state** or minimum energy state of a Hamiltonian
- A Hamiltonian is a Hermitian matrix H (i.e. it verifies $H = H^\dagger$)
- The (expected) energy of a state $|\psi\rangle$ is

$$\langle\psi|H|\psi\rangle$$

Example: the Ising model

- We have n spins that interact with their neighbours
- The Hamiltonian of the system is

$$H = \sum_{1 \leq i < j \leq n} J_{ij} Z_i Z_j + \sum_{i=1}^{n} h_i Z_i$$

  with J_{ij} and h_i real coefficients
- We want to find a value assignment (1 or -1) that minimizes the sum
- The problem is NP-hard (it includes the Max-Cut problem)

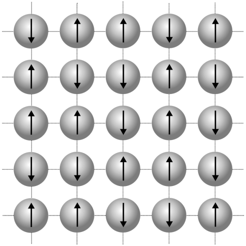


Image credits: Peter Eastman

QUBO: Quadratic Unconstrained Binary Optimization

- A closely related family of problems is that of Quadratic Unconstrained Binary Optimization (QUBO)
- These problems are stated as

$$\text{Minimize} \sum_{1 \leq i \leq j \leq n}^{n} w_{ij} x_i x_j$$

  where each x_i is a binary variable and w_{ij} are real coefficients
- We can transform the problem into an Ising model via

$$x_i = \frac{1 - z_i}{2}$$

  and get back to QUBO with

$$z_i = 1 - 2x_i$$

Adiabatic quantum computing

- How to obtain the *ground state* of H?
- A natural approach is to apply H itself to reach the solution
- The **adiabatic theorem** (roughly) says that if we start in the ground state of a Hamiltonian and we change this Hamiltonian slowly, we will stay in a ground state
- The idea behind adiabatic quantum computing is
  - Start with the ground state of a simple Hamiltonian H_i
  - Evolve the the system to the ground state of the problem Hamiltonian H_f
  - To achieve that, we apply a time-dependent Hamiltonian

$$H(t) = (1 - \frac{t}{T})H_i + \frac{t}{T}H_f$$

  for time T

Adiabatic quantum computing (2)

- To guarantee adiabaticity, T must grow as the inverse of the square of the *spectral gap* of $H(t)$ (difference between the first and the second energy levels)
- The spectral gap is **hard** to compute
- In practice, *quantum annealing* is used:
  - We take $H_i = -\sum_{j=1}^{n} X_j$ (with ground state $\sum_{x=0}^{2^n-1} |x\rangle$)
  - H_f is an Ising Hamiltonian that encodes our problem
  - We let the system evolve for time T (no necessarily adiabatic)
  - We measure to obtain a candidate solution
  - We repeat the process a number of times and keep the best solution
- This is the basis of D-Wave's quantum computers

D-Wave's quantum computers

- These are special-purpose computers: they find approximate solutions of the Ising model
- Free access (1 minute/month) at `https://www.dwavesys.com/take-leap`
- We will test them with this example

Published: 19 October 2017

Solving a Higgs optimization problem with quantum annealing for machine learning

Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar & Maria Spiropulu

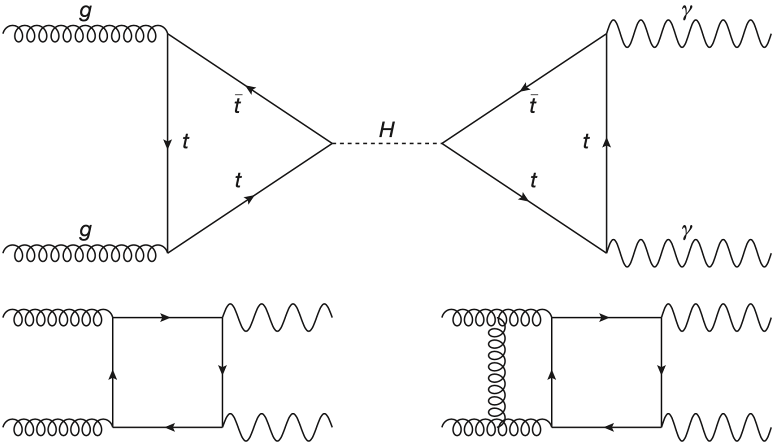
Nature **550**, 375–379(2017) | Cite this article

Abstract

The discovery of Higgs-boson decays in a background of standard-model processes was assisted by machine learning methods[1,2]. The classifiers used to separate signals such as these from background are trained using highly unerring but not completely perfect simulations of the physical processes involved, often resulting in incorrect labelling of background processes or signals (label noise) and systematic errors. Here we use quantum[3,4,5,6] and classical[7,8] annealing (probabilistic techniques for approximating the global maximum or minimum of a given function) to solve a Higgs-signal-versus-background machine learning optimization problem, mapped to a problem of finding the ground state of a corresponding Ising spin model. We build a set of weak classifiers based on the kinematic observables of the Higgs decay photons, which we then use to construct a

An application in High Energy Physics (2)

- Signal: production of a Higgs boson through the fusion of two gluons which then decay into two photons
- Background: standard-model two-photon production processes



An application in High Energy Physics (3)

- The authors consider 36 weak classifiers $c_i(x)$ and combine them to form a strong classifier

$$O(x) = \sum_i w_i c_i(x)$$

  with $w_i \in \{0, 1\}$
- They minimize

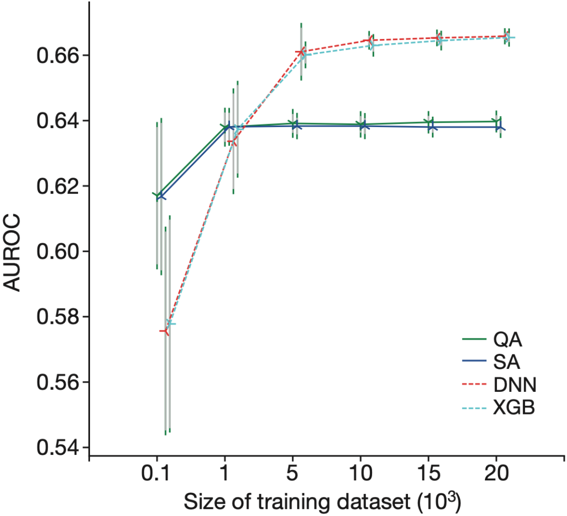
$$\sum_x (y(x) - \sum_i w_i c_i(x))^2$$

  which, when an additional regularization parameter λ is added, is equivalent to minimizing

$$\sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_i) w_i$$

  where $C_{ij} = \sum_x c_i(x) c_j(x)$ and $C_i = \sum_x c_i(x) y(x)$.

An application in High Energy Physics (4)



Part XIV

Quantum Approximate Optimization Algorithm: going digital

The Quantum Approximate Optimization Algorithm

- The Quantum Approximate Optimization Algorithm (QAOA) was proposed by Farhi, Goldstone and Gutmann (2014) to obtain approximate solutions of the problem of minimizing

$$C(x) = \sum_{a} w_a C_a(x)$$

  where x is n-bit string, w_a are real weights and each C_a is a boolean function
- Max-Cut is one such problem, with every w_a equal to 1 and each C_a of the form

$$x_i \oplus x_j$$

- The maximum satisfiability (MAX-SAT) and weighted maximum satisfiability (weighted MAX-SAT) are other examples of that kind of problems

Boolean functions and Hamiltonians

- For each boolean C_a we can find a Hamiltonian H_a of the form

$$a_0 I + \sum_i a_i Z_i + \sum_{i<j} a_{ij} Z_i Z_j + \sum_{i<j<k} a_{ijk} Z_i Z_j Z_k + \cdots$$

such that for every string x it holds that $C_a(x) = \langle x| H_a |x\rangle$

- Then, minimizing $C(x)$ is equivalent to finding the ground state of

$$H_f = \sum_a w_a H_a$$

since H_f is diagonal and $\langle x| H_f |x\rangle = C(x)$.

x	$\frac{1}{2}I - \frac{1}{2}Z$	$\overline{x}$	$\frac{1}{2}I + \frac{1}{2}Z$
$x_1 \oplus x_2$	$\frac{1}{2}I - \frac{1}{2}Z_1 Z_2$	$\bigoplus_{j=1}^k x_j$	$\frac{1}{2}I - \frac{1}{2}Z_1 Z_2 \ldots Z_k$
$x_1 \wedge x_2$	$\frac{1}{4}I - \frac{1}{4}(Z_1 + Z_2 - Z_1 Z_2)$	$\bigwedge_{j=1}^k x_j$	$\frac{1}{2^k}\prod_j (I - Z_j)$
$x_1 \vee x_2$	$\frac{3}{4}I - \frac{1}{4}(Z_1 + Z_2 + Z_1 Z_2)$	$\bigvee_{j=1}^k x_j$	$I - \frac{1}{2^k}\prod_j (I + Z_j)$
$\overline{x_1 x_2}$	$\frac{3}{4}I + \frac{1}{4}(Z_1 + Z_2 - Z_1 Z_2)$	$x_1 \Rightarrow x_2$	$\frac{3}{4}I + \frac{1}{4}(Z_1 - Z_2 + Z_1 Z_2)$

Image credits: Stuart Hadfield https://arxiv.org/pdf/1804.09130.pdf

The parametrized states of QAOA

- QAOA is an adaptation of the adiabatic model to gate-based quantum computers
- Remember that the adiabatic Hamiltonian was

$$H(t) = (1 - \frac{t}{T})H_i + \frac{t}{T}H_f$$

with $H_i = -\sum_{j=1}^{n} X_j$

- As an approximation of the evolution of the system, we consider parametrized states of the form

$$|\beta, \gamma\rangle = e^{-i\beta_p H_i} e^{-i\gamma_p H_f} \dots e^{-i\beta_2 H_i} e^{-i\gamma_2 H_f} e^{-i\beta_1 H_i} e^{-i\gamma_1 H_f} |s\rangle$$

where $p \geq 1$ and

$$|s\rangle = \sum_{i=0}^{2^n-1} |x\rangle$$

Optimization with QAOA

- QAOA is a hybrid method in which both a quantum and classical computer are used
- The steps are:
  1. Choose a value for p and some initial angles β, γ
  2. Prepare the state $|\beta, \gamma\rangle$
  3. Estimate the energy $E(\beta, \gamma) = \langle\beta, \gamma| H_f |\beta, \gamma\rangle$ of $|\beta, \gamma\rangle$
  4. Vary β and γ in order to minimize $E(\beta, \gamma)$
  5. If the stopping criterium is met, stop. Else, go to 2
- Step 2 is carried out on the quantum computer and steps 1, 3 and 4, on a classical one

How to prepare $|\beta, \gamma\rangle$

- We already know that $|s\rangle = \sum_{i=0}^{2^n-1} |x\rangle$ can be prepared with Hadamard gates
- Each $e^{-i\beta_k X_j}$ is a rotation $R_X(2\beta_k)$ or equivalently

$$|x_j\rangle \; - \boxed{H} - \boxed{R_Z(2\beta_k)} - \boxed{H}$$

- To implement $e^{-i\gamma_k H_f}$ we only need to consider cases of the form

$$e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$$

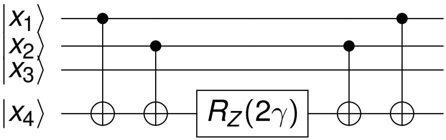
  because
  - All terms of the form $Z_{i_1} \cdots Z_{i_j}$ commute
  - The weights in $H_f = \sum_a w_a H_a$ are “absorbed” by the angles γ

Implementing $e^{-i\gamma_k Z_{i_1}\cdots Z_{i_j}}$

- Notice that $e^{-i\gamma_k Z_{i_1}\cdots Z_{i_j}}$ is diagonal in the computational basis
- In fact, for a binary string $x = x_1 \dots x_n$ it acts on $|x\rangle$ as
  - $|x\rangle \to e^{-i\gamma_k} |x\rangle$ if $x_{i_1} \oplus \cdots \oplus x_{i_j} = 0 \mod 2$
  - $|x\rangle \to e^{i\gamma_k} |x\rangle$ if $x_{i_1} \oplus \cdots \oplus x_{i_j} = 1 \mod 2$
- This is very similar to the action of a R_Z rotation
- Then, we can:
  - Compute the parity $x_{i_1} \oplus \cdots \oplus x_{i_j}$ with CNOT gates
  - Apply $R_Z(2\gamma_k)$ on the qubit where we have computed the parity
  - Uncompute the parity

An example

- Imagine that we are working with 4 qubits and we want to implement $e^{-i\gamma Z_1 Z_2 Z_4}$
- We can use the following circuit:



Estimating the energy

- Estimating the energy is very easy in the case of QAOA
- We repeat the following process a fixed number of times:
  1. Prepare the state $|\beta, \gamma\rangle$
  2. Measure it to obtain a string x
  3. Compute $C(x)$

  and then we average the results
- This works because if

$$|\beta, \gamma\rangle = \sum_{x \in \{0,1\}^n} a_x |x\rangle$$

  then

$$\langle \beta, \gamma | H_f | \beta, \gamma \rangle = \sum_{x \in \{0,1\}^n} |a_x|^2 C(x)$$

- It is also interesting to keep the string x with minimum value $C(x)$ over all we obtain when we measure

Some comments on QAOA

- For the procedure to be efficient, H_f must have a number of terms $e^{-i\gamma_k Z_{i_1}\cdots Z_{i_j}}$ that is polynomial in the number n of qubits and the number m of clauses $C_a(x)$ of $C(x)$
- If a clause $C_a(x)$ only involves k bits, then its translation H_a will involve terms with at most k Pauli matrices Z_i
- Thus, if p is a constant independent of n and m and all clauses involve at most k bits (also independent of n and m) then the number of gates will be polynomial in n and m
- This is the case, for example, of problems such as MaxCut or Max 3-SAT
- When $p \to \infty$, the ground state of $|\beta, \gamma\rangle$ tends to the ground state of H_f
- Interesting results can be obtained in some cases even for small p
- The choice of classical optimizer is important

Applying QAOA for particle track reconstruction

Computing and Software for Big Science (2020) 4:1
https://doi.org/10.1007/s41781-019-0032-5

ORIGINAL ARTICLE

A Pattern Recognition Algorithm for Quantum Annealers

Frédéric Bapst[1] · Wahid Bhimji[2] · Paolo Calafiura[2] · Heather Gray[2,3] · Wim Lavrijsen[2] · Lucy Linder[1] · Alex Smith[3]

Received: 1 March 2019 / Accepted: 29 November 2019

Abstract
The reconstruction of charged particles will be a key computing challenge for the high-luminosity Large Hadron Collider (HL-LHC) where increased data rates lead to a large increase in running time for current pattern recognition algorithms. An alternative approach explored here expresses pattern recognition as a quadratic unconstrained binary optimization (QUBO), which allows algorithms to be run on classical and quantum annealers. While the overall timing of the proposed approach and its scaling has still to be measured and studied, we demonstrate that, in terms of efficiency and purity, the same physics performance of the LHC tracking algorithms can be achieved. More research will be needed to achieve comparable performance in HL-LHC conditions, as increasing track density decreases the purity of the QUBO track segment classifier.

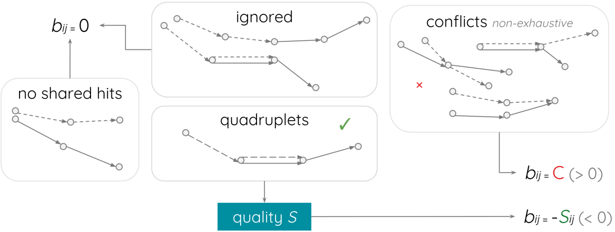
Keywords Quantum annealing · Pattern recognition · HEP particle tracking

Applying QAOA for particle track reconstruction (2)

- QUBO formulation to select the best pairs of triplets by minimizing

$$O(a, b, T) = \sum a_i T_i + \sum b_{ij} T_i T_j$$

  where the a_i are bias weights expressing quality of the triplets (all equal) and the b_{ij} are coupling strengths between triplets
- From this formulation, QAOA is planned to be applied on Rigetti computers (work by Eric Rohm at Lawrence Berkeley National Laboratory)



Part XV

Variational Quantum Eigensolver: endless forms most beautiful

VQE: Variational Quantum Eigensolver

- QAOA can be seen as a particular case of a more general algorithm: the Variational Quantum Eigensolver (VQE)
- Now, we will have a general Hamiltonian H_f (with a polynomial number of terms) and we want to approximate its ground state
- Instead of the parametrized state $|\beta, \gamma\rangle$ of QAOA we will use
  - An initial state $|\psi\rangle$ that is easy to prepare (it could be just $|0\rangle$)
  - A parametrized unitary $U(\theta)$ that is called a **variational form**
- We can create an ansatz

  $$|\psi(\theta)\rangle = U(\theta) |\psi\rangle$$

  and try to minimize its energy with respect to H_f by varying the parameters

The variational principle

- Since H_f is a Hermitian matrix, it has real eigenvalues λ_i and an associated orthonormal basis of eigenvectors $|\phi_i\rangle$
- Then, we can write $|\psi(\theta)\rangle$ as a linear combination

$$|\psi(\theta)\rangle = \sum_i \alpha_i |\phi_i\rangle$$

- The energy of $|\psi(\theta)\rangle$ is

$$\langle\psi(\theta)| H_f |\psi(\theta)\rangle = \sum_i |\alpha_i|^2 \lambda_i$$

- If λ_{min} is the minimum of the eigenvalues then

$$\min_\theta \langle\psi(\theta)| H_f |\psi(\theta)\rangle \geq \lambda_{min}$$

Approximating the ground state with VQE

- VQE is also a hybrid method in which both a quantum and classical computer are used
- The steps are:
  1. Choose an initial state $|\psi\rangle$, a variational form $U(\theta)$ and some initial vector θ
  2. Prepare the state $|\psi(\theta)\rangle = U(\theta)\,|\psi\rangle$
  3. Estimate the energy $E(\theta) = \langle\psi(\theta)|\,H_f\,|\psi(\theta)\rangle$ of $|\psi(\theta)\rangle$
  4. Vary θ in order to minimize $E(\theta)$
  5. If the stopping criterium is met, stop. Else, go to 2
- Step 2 is carried out on the quantum computer and steps 1, 3 and 4, on a classical one

Estimating the energy of a state

- The Hamiltonian can be always expressed as a linear combination of tensor product of Paulis
- For instance
$$H_f = \frac{1}{4} Z_1 Z_3 - 3 X_1 Y_3 Z_4$$
- Given $|\psi\rangle$, we can use linearity and evaluate
$$\langle\psi| H_f |\psi\rangle = \frac{1}{4} \langle\psi| Z_1 Z_3 |\psi\rangle - 3 \langle\psi| X_1 Y_3 Z_4 |\psi\rangle$$
- To estimate $\langle\psi| Z_1 Z_3 |\psi\rangle$ we can just measure $|\psi\rangle$ in the computational basis and average the energies of the results (which will be 1 or -1 for each individual measurement result).

Estimating the energy of a state (2)

- To estimate $\langle\psi| X_1 Y_3 Z_4 |\psi\rangle$ we can notice that

$$X = HZH$$

and

$$Y = SHZHS^\dagger$$

- Then $\langle\psi| X_1 Y_3 Z_4 |\psi\rangle$ is equal to

$$\langle\psi| (H \otimes I \otimes SH \otimes I) Z_1 Z_3 Z_4 (H \otimes I \otimes HS^\dagger \otimes I) |\psi\rangle$$

- Thus, we can just measure the energy on $Z_1 Z_3 Z_4$ of $(H \otimes I \otimes HS^\dagger \otimes I) |\psi\rangle$ because

$$\Big((H \otimes I \otimes HS^\dagger \otimes I) |\psi\rangle\Big)^\dagger = \langle\psi| (H \otimes I \otimes SH \otimes I)$$

- Notice that this is equivalent to measuring in a different basis

Simulating molecules with VQE

- VQE has been used to estimate ground states of several molecules
- The fermionic Hamiltonian has to be translated into a qubit Hamiltonian (Jordan-Wigner, Bravyi-Kitaev...)
- Information of the problem is used for:
  - The initial state (vacuum state $|0\rangle$, Hartree-Fock...)
  - The variational form (Unitary Coupled-Cluster Single and Double excitations...)

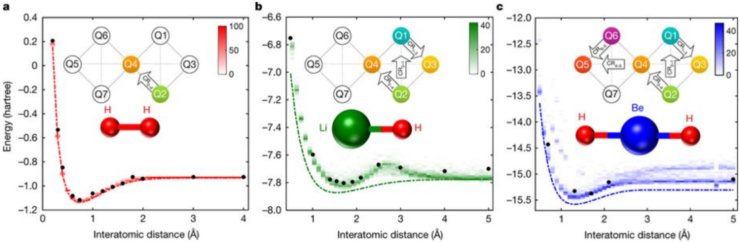


Image credits: Kandala, Mezzacapo, Temme, Takita, Brink, Chow, Gambetta. Nature 549, 242–246 (2017)

Finding excited states

- We can also use VQE to find excited states (eigenstates that are not the ground state)
- Once we have the ground state $|\psi_0\rangle = U(\theta_0)|\psi\rangle$, we consider the Hamiltonian

$$H'_f = H_f + C|\psi_0\rangle\langle\psi_0|$$

- Then, we have that $\langle\varphi| H'_f |\varphi\rangle$ is

$$\langle\varphi| H_f |\varphi\rangle + C\langle\varphi|\psi_0\rangle\langle\psi_0|\varphi\rangle = \langle\varphi| H_f |\varphi\rangle + C|\langle\psi_0|\varphi\rangle|^2$$

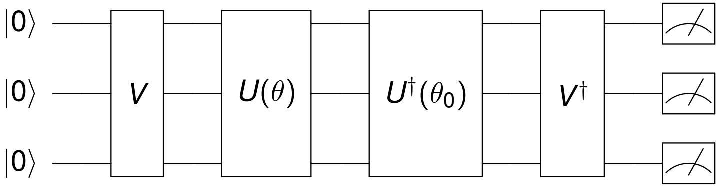
- if C is bigger than the difference between the ground energy and the next energy level of H_f, then $|\psi_0\rangle$ is not the ground state of H'_f

Computing inner products of parametrized states

- To compute the inner product in the expression of the energy we can notice that $|\psi_0\rangle = U(\theta_0)|\psi\rangle$ and that the new states that we try will be of the form $|\varphi\rangle = U(\theta)|\psi\rangle$ for some θ
- Then, it is easy to estimate $|\langle\psi_0|\varphi\rangle|^2$ by running the circuit of the figure and computing the relative frequency of $|0\rangle$ because

  $$|\langle\psi_0|\varphi\rangle|^2 = |\langle 0| V^\dagger U(\theta_0)^\dagger U(\theta) V |0\rangle|^2$$

  where V is a unitary such that $V|0\rangle = |\psi\rangle$



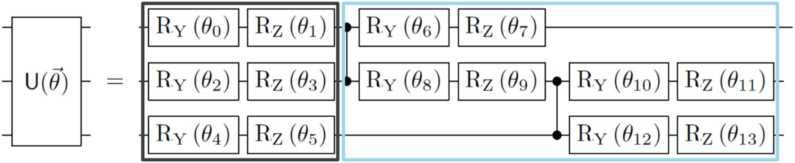
An application of VQE in High Energy Physics

- Work by Li, Macridin, Spentzouris - Fermilab (2019)
- Rabi Hamiltonian: two-level system (TLS) coupled to a photon mode

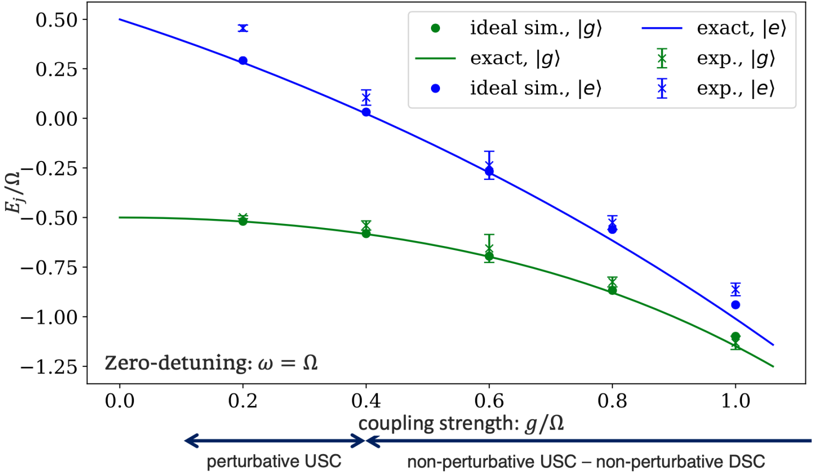
$$H = \omega a a^\dagger + \frac{\Omega}{2} Z + g(a^\dagger + a)X$$

- Number-basis binary encoding: photon mode truncated to up to 3 photons

$$H = \omega Z_0 + \frac{\omega}{2} Z_1 + \frac{\Omega}{2} Z_2 + g\sqrt{Z_0 + 2} X_1 X_2$$
$$+ \frac{g}{\sqrt{2}} X_0 X_1 X_2 + Y_0 Y_1 X_2 + \frac{3\omega}{2}$$



Results on simulator and Rigetti's quantum computers



Part XVI

Quantum Machine Learning: a marriage made in heaven

What I talk about when I talk about Quantum Machine Learning

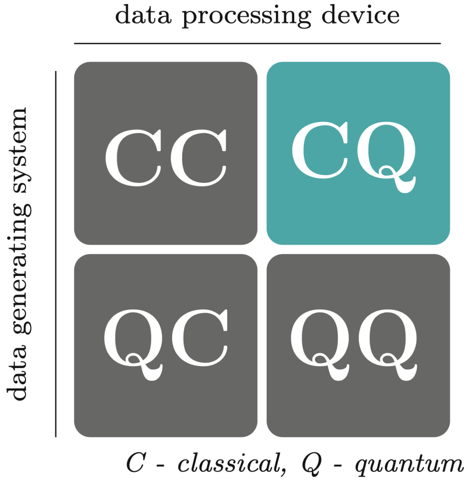


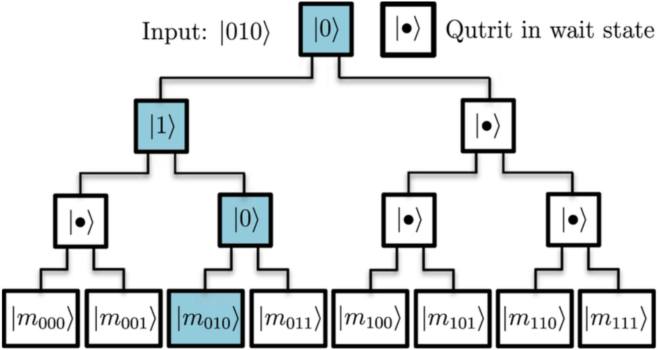
Image credits: Figure taken from *Supervised Learning with Quantum Computers*. Schuld, Petruccione (2018)

QBLAS: The Quantum Basic Linear Algebra Subroutines

- A number of algorithms in Quantum Machine Learning (QML) rely on the exponential speedup of methods such as
  - Quantum Fourier Transform
  - Quantum Phase Estimation
  - HHL
- We refer to these methods as Quantum Basic Linear Algebra Subroutines (QBLAS)
- Other quantum subroutines used in QML include amplitude amplification and quantum annealing
- Some common problems are how to load the **input**, how to read the **output** and the **size** of the circuits

QRAM: The elephant in the room

- A Quantum Random Access Memory should allow queries in superposition
- Several architectures have been proposed (for instance, the “bucket brigade”) but further investigation is needed
- Loading data can become a bottleneck for many QML algorithms



Translational QML and speedups

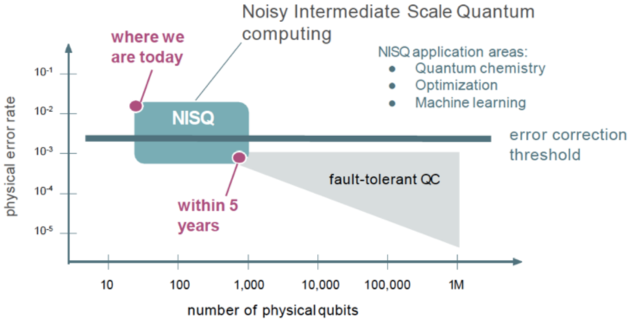
Method	Speedup	Amplitude amplification	HHL	Adiabatic	qRAM
Bayesian inference[106,107]	$O(\sqrt{N})$	Yes	Yes	No	No
Online perceptron[108]	$O(\sqrt{N})$	Yes	No	No	Optional
Least-squares fitting[9]	$O(\log N)$*	Yes	Yes	No	Yes
Classical Boltzmann machine[20]	$O(\sqrt{N})$	Yes/No	Optional/No	No/Yes	Optional
Quantum Boltzmann machine[22,61]	$O(\log N)$*	Optional/No	No	No/Yes	No
Quantum PCA[11]	$O(\log N)$*	No	Yes	No	Optional
Quantum support vector machine[13]	$O(\log N)$*	No	Yes	No	Yes
Quantum reinforcement learning[30]	$O(\sqrt{N})$	Yes	No	No	No

*There exist important caveats that can limit the applicability of the method[51].

Image credits: Table taken from Biamonte, Wittek, Pancotti, Rebentrost, Wiebe, Lloyd. Nature 549, 195–202(2017)

QML in the times on NISQ

- Noisy Intermediate-Scale Quantum computers are
  - Subjet to noise (not fault-tolerant)
  - Limited in the number of qubits (50-100)
  - Not fully-connected
- Despite these drawbacks, they may be useful for QML



"Quantum computing in the NISQ era and beyond" Preskill, 2018 https://arxiv.org/abs/1801.00862

Part XVII

Quantum Support Vector Machines: exploiting the kernel trick

Support Vector Machines

- Support Vector Machines (SVM) are a very popular machine learning algorithm used for data classification
- The main idea is to find a hyperplane that separates data from two different classes with the maximum possible margin

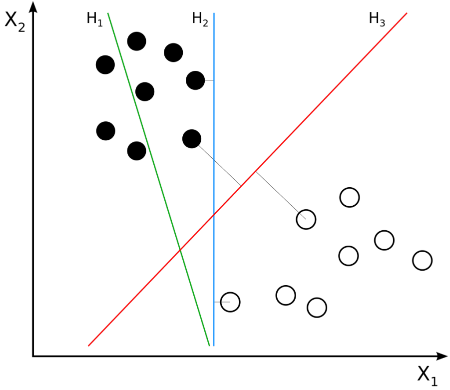


Image credits: wikipedia.org

Finding the hyperplane

- We are given training data points (x_i, y_i) where the x_i are vectors of real numbers and $y_i \in \{1, -1\}$
- The problem of finding the separating hyperplane with the biggest margin can be formulated as

$$\text{Minimize } \frac{1}{2}||w||^2 \text{ subject to } y_i(w \cdot x_i + b) \geq 1$$

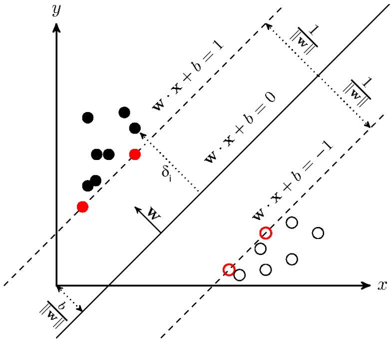


Image credits: John Daniel, towardsdatascience.com

The soft-margin case

- In the “soft-margin” case we introduce a hyperparameter $C \geq 0$ and reformulate the problem as

$$\text{Minimize } \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

subject to

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0$$

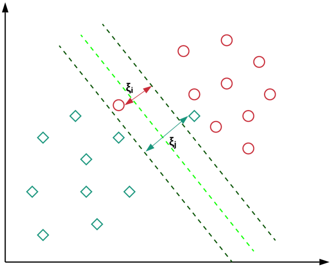


Image credits: Rishabh Misra, towardsdatascience.com

Dual formulation of SVM

- An equivalent formulation of the SVM optimization problem is this dual formulation

$$\text{Maximize } \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \left(x_i \cdot x_j\right)$$

subject to

$$0 \leq \alpha_i \leq C \quad \sum_i \alpha_i y_i = 0$$

- From the values α_i we can recover b and w. In fact

$$w = \sum_i \alpha_i y_i x_i$$

and to classify a point x we compute

$$w \cdot x + b = \sum_i \alpha_i y_i \left(x_i \cdot x\right) + b$$

Non-linear separation

- A common technique to improve classification with Support Vector Machines is to embed the data points x_i into a higher-dimensional space using a feature map $\phi(x_i)$

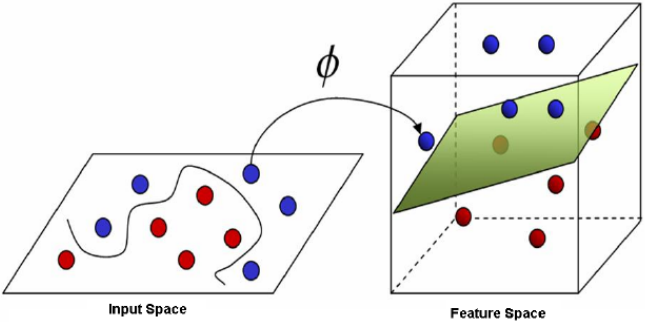


Image credits: C. Moreira, “Learning To Rank Academic Experts’, Master Thesis, Technical University of Lisbon, 2011

The Kernel Trick

- We can easily incorporate the feature map in our formulation of the dual problem for the SVM

$$\text{Maximize } \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \left(\phi(x_i) \cdot \phi(x_j)\right)$$

  subject to

$$0 \leq \alpha_i \leq C \quad \sum_i \alpha_i y_i = 0$$

- Again, we can obtain w as

$$w = \sum_i \alpha_i y_i \phi(x_i)$$

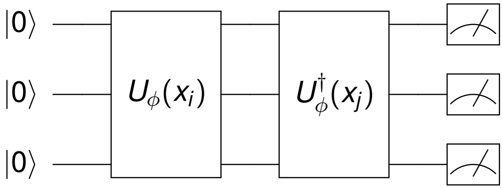
  and to classify a point x we only need to compute

$$w \cdot x + b = \sum_i \alpha_i y_i \left(\phi(x_i) \cdot \phi(x)\right) + b$$

- The function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is called “kernel”

Computing kernel functions with quantum computers

- In 2019, Havlíček, Córcoles, Temme et al. proposed using quantum computers as kernel estimators
- Each data point x_i is embedded in a Hilbert space by means of a variational circuit $U_\phi(x_i)$ such that $U_\phi(x_i)\left|0\right\rangle = \left|\phi(x_i)\right\rangle$
- We know we can $|\left\langle \phi(x_j) \middle| \phi(x_i) \right\rangle|^2$ by running the circuit of the figure and computing the relative frequency of $\left|0\right\rangle$



Application of Quantum Machine Learning to HEP Analysis at LHC using Quantum Computer Simulators and Quantum Computer Hardware

Sau Lan Wu

I am new in this field, since two years.
I have assembled an international and interdisciplinary team of High Energy Physicists and Quantum Computing Scientists:

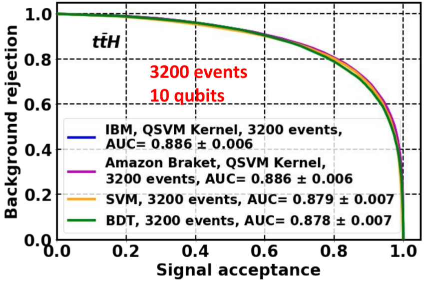
Jay Chan, Alkaid Cheng, Wen Guan, Shaojun Sun, Alex Wang, Sau Lan Wu, Rui Zhang, Chen Zhou
Physics Department, University of Wisconsin-Madison
Miron Livny
Computer Sciences Department, University of Wisconsin-Madison
Federico Carminati, Alberto Di Meglio
CERN Quantum Technology Initiative, IT Department, CERN
Panagiotis Barkoutsos, Ivano Tavernelli, Stefan Woerner, Jennifer Glick
IBM Research Zurich and IBM T.J. Watson Research Center
Andy Li, Joseph Lykken, Panagiotis Spentzouris
Quantum Institute, Fermilab
Samuel Yen-Chi Chen, Shinjae Yoo
Computational Science Initiative, BNL
Tzu-Chieh Wei
C.N. Yang Institute for Theoretical Physics, State University of New York at Stony Brook
Pavel Lougovski, Sanjay Padhi, Simone Severini, Dewayne Walker
Quantum Computing and AI Research, Amazon Web Services

4 November, 2020
QuantHEP Seminar

Using QSVM in High Energy Physics (2)

- Classification of Higgs events ($H \to \gamma\gamma$ and $H \to \mu\mu$)

Method 2: Employing Quantum SVM Kernel method with Amazon simulator for ttH (H → $\gamma\gamma$) analysis



	AUC (ttH, 3200 events)
QSVM Kernel IBM simulator	0.886
QSVM Kernel Amazon simulator	0.886
BDT	0.879
SVM	0.878

Image credits: Sau Lan Wu et al.

Using QSVM in High Energy Physics (2)

Method 2: Employing Quantum SVM Kernel method with Amazon simulator for ttH (H $\rightarrow \gamma\gamma$) analysis

AUC vs number of events

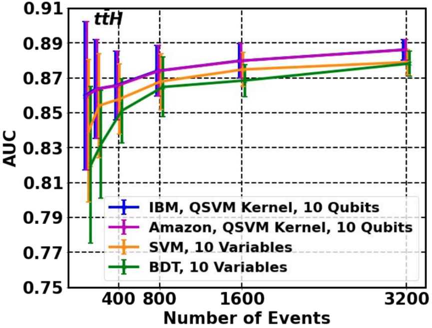


Image credits: Sau Lan Wu et al.

Part XVIII

Quantum Neural Networks: Deep Learning meets Quantum Computing

What is a Quantum Neural Network

- Quantum Neural Networks or Variational Quantum Classifiers are parametrized quantum circuits that can be "trained" on data and used for classification tasks
- The most common architecture is shown in the figure below: a feature map that embeds the data point into the Hilbert space and a variational form that performs the classification

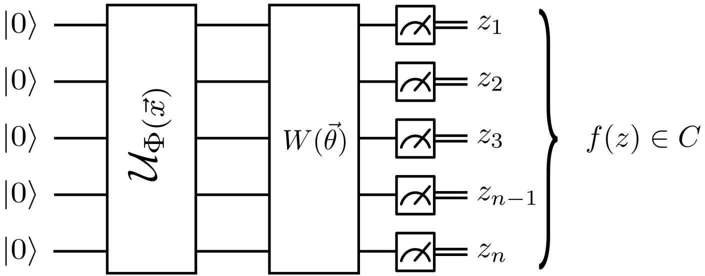


Image credits: Vojtech Havlíček et al. https://arxiv.org/pdf/1804.11326.pdf

Training and classifying with a Quantum Neural Network

- A QNN prepares a state $|\psi(x,\theta)\rangle$ that depends on the input data x and the parameters θ
- We measure the state and compute an average value, for instance

$$f(x,\theta) = \langle\psi(x,\theta)|\, Z_1 \cdots Z_n \,|\psi(x,\theta)\rangle$$

- For each training example x_i we have a class y_i
- We choose a loss function L and we want to find θ minimizing

$$\sum_i L(y_i, f(x,\theta))$$

- Once we obtain the optimal value θ_{min} we can predict a class for x using $f(x,\theta_{min})$

Gradients and the parameter shift rule

- To obtain θ_{min}, we can use a classical minimizer
- If we need to compute gradients of f the parameter-shift rule is useful
- Suppose
  $$U(\theta) = e^{-i\theta H}$$
  with H a Hermitian matrix with eigenvalues $\pm r$ (r real)
- This is the case, for instance, if U is a one-qubit rotation
- Then, we have
  $$\frac{\partial f(x,\theta)}{\partial \theta} = r \cdot [f(x,\theta+s) - f(x,\theta-s)]$$
  where $s = \frac{\pi}{4r}$
- This requires just two extra evaluations of the same circuit with shifted parameters

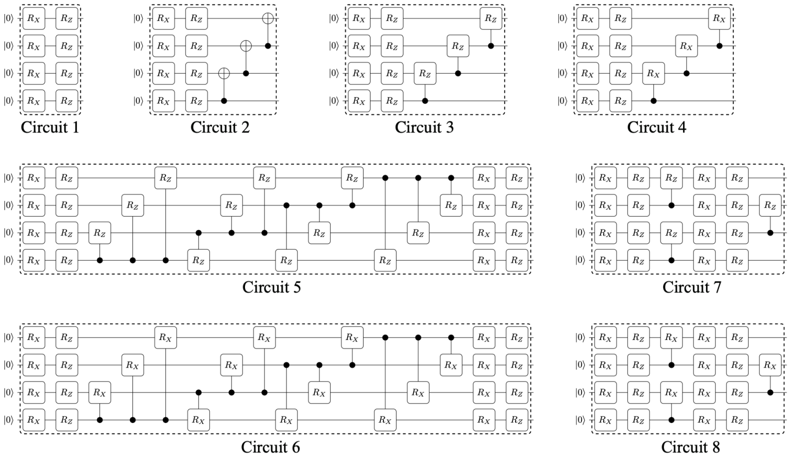


Image credits: Sim, Johnson, Aspuru-Guzik. Adv. Quantum Tech. 2(12) (2019)

The power of quantum neural networks

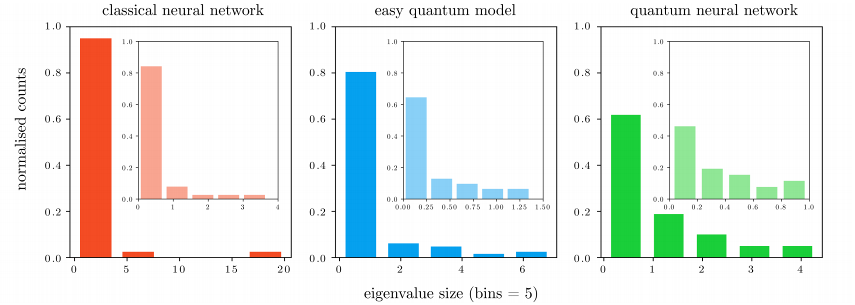


Image credits: Amira Abbas et al. https://arxiv.org/pdf/2011.00027.pdf

Method 1: Employing VQC (Variational Quantum Classifier) with IBM Q simulator for ttH ($H \to \gamma\gamma$) analysis and $H \to \mu\mu$ analysis

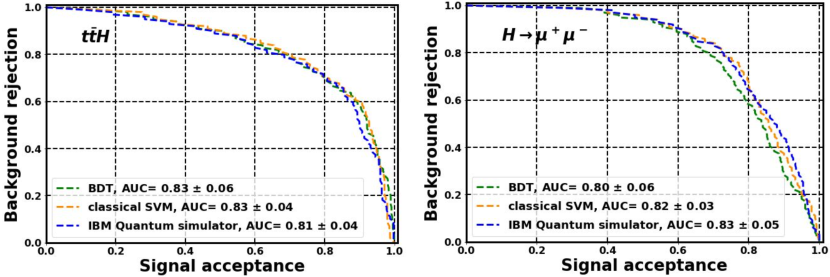
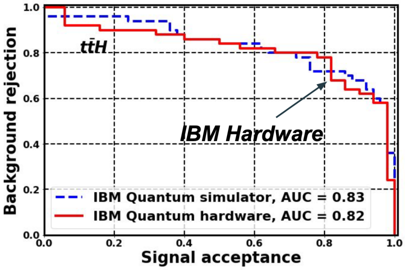


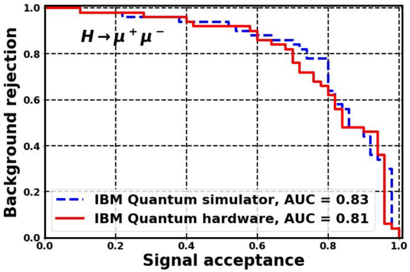
Image credits: Sau Lan Wu et al.

Quantum Neural Networks in HEP (2)

Method 1: Employing VQC (Variational Quantum Classifier) with IBM hardware for ttH (H → $\gamma\gamma$) analysis and H → $\mu\mu$ analysis



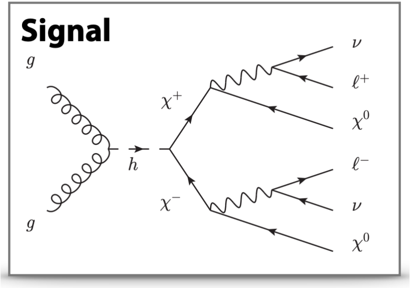
hardware AUC = 0.82, simulator AUC = 0.83



hardware AUC = 0.81, simulator AUC = 0.83

Image credits: Sau Lan Wu et al.

Quantum Neural Networks in HEP (3)



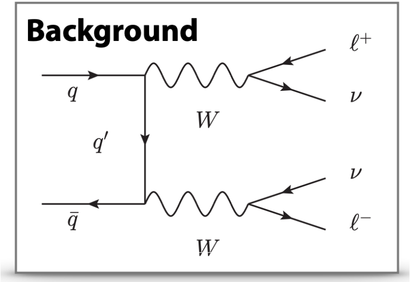


Image credits: Koji Terashi

Quantum Neural Networks in HEP (4)

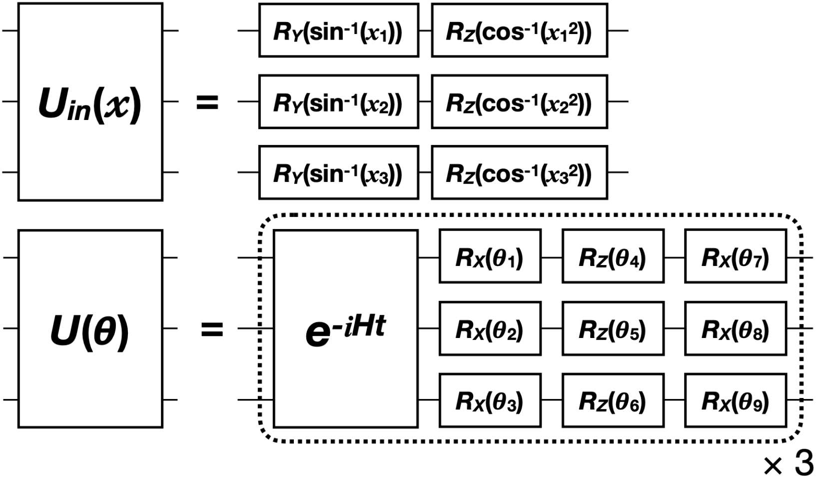


Image credits: Koji Terashi et al. https://arxiv.org/pdf/2002.09935.pdf

Quantum Neural Networks in HEP (5)

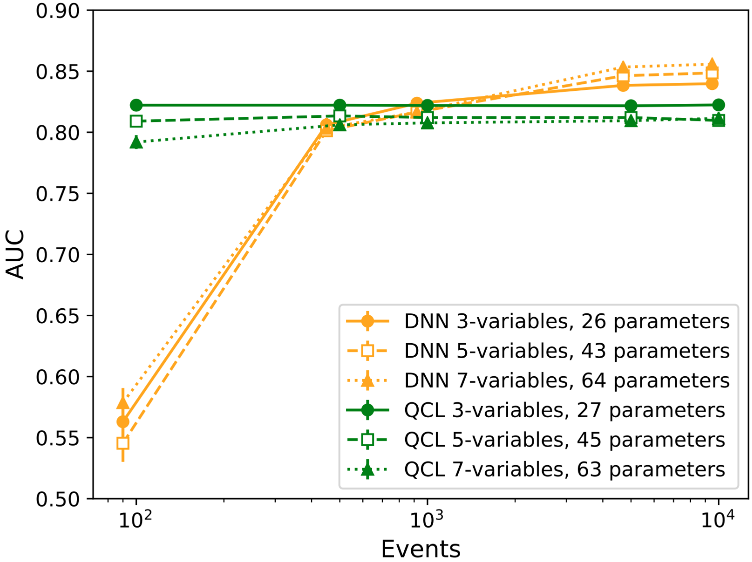


Image credits: Koji Terashi et al. https://arxiv.org/pdf/2002.09935.pdf

Part XIX

Quantum Generative Adversarial Networks: this quantum image does not exist

GANs: Generative Adversarial Networks

- Generative Adversarial Networks (GANs) were introduced by Ian Goodfellow and his collaborators in 2014
- The objective, is, given a training dataset, learning to generate new, unseen data with the same distribution
- Impressive results have been achieved in several different applications

Image credits: Nvidia/StyleGAN `https://arxiv.org/abs/1812.04948`

Architecture of a GAN

- Two neural networks: generator and discriminator
- The generator tries to “fool” the generator
- The discriminator tries to distinguish between real and fake images

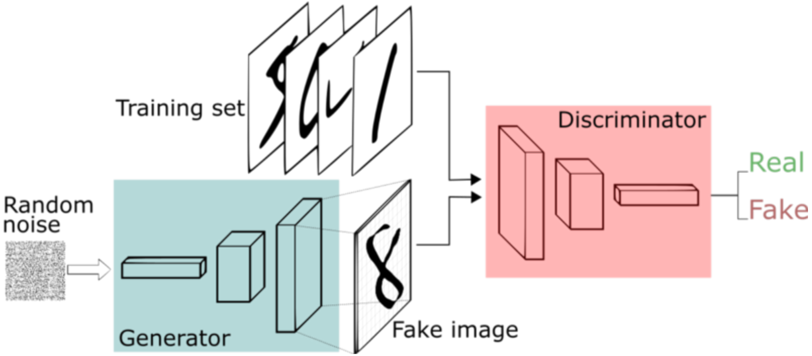


Image credits: Thalles Silva - www.freecodecamp.org

GAN training

- The generator and discriminator are trained in alternating phases
- The discriminator tries to maximize

$$E_x[\log D(x)] + E_z[\log(1 - D(G(z)))]$$

- The generator can try to minimize

$$E_z[\log(1 - D(G(z)))]$$

or (in practice) to maximize

$$E_z[\log D(G(z))]$$

Quantum GANs

- A Quantum GAN replaces the generator or the discriminator (or both) with a quantum circuit

PHYSICAL REVIEW LETTERS **121,** 040502 (2018)

Editors' Suggestion   Featured in Physics

Quantum Generative Adversarial Learning

Seth Lloyd[1] and Christian Weedbrook[2]
[1]*Massachusetts Institute of Technology, Department of Mechanical Engineering, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*
[2]*Xanadu, 372 Richmond Street W, Toronto, Ontario M5V 1X6, Canada*

(Received 30 April 2018; published 26 July 2018)

PHYSICAL REVIEW A **98**, 012324 (2018)

Quantum generative adversarial networks

Pierre-Luc Dallaire-Demers* and Nathan Killoran
Xanadu, 372 Richmond Street W, Toronto, Ontario M5V 1X6, Canada

(Received 7 May 2018; published 23 July 2018)

Using a QGAN to load a probability distribution

npj Quantum Information

www.nature.com/npjqi

ARTICLE OPEN

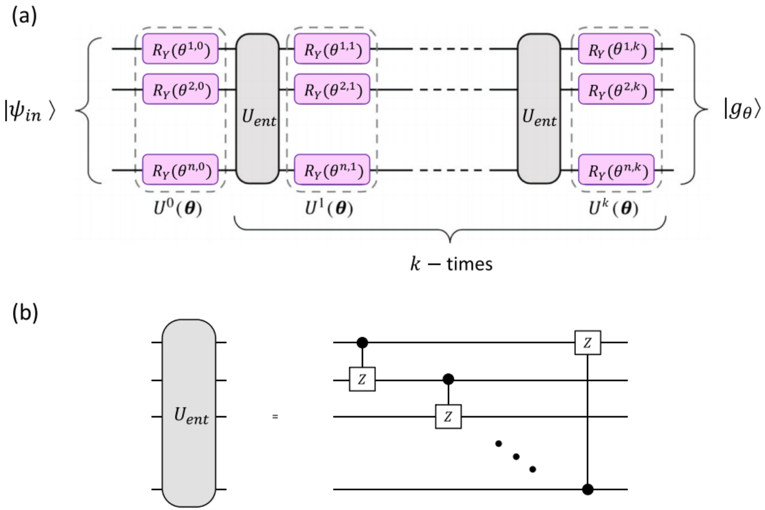
Quantum Generative Adversarial Networks for learning and loading random distributions

Christa Zoufal[1,2*], Aurélien Lucchi[2] and Stefan Woerner[1]

Quantum algorithms have the potential to outperform their classical counterparts in a variety of tasks. The realization of the advantage often requires the ability to load classical data efficiently into quantum states. However, the best known methods require $\mathcal{O}(2^n)$ gates to load an exact representation of a generic data structure into an n-qubit state. This scaling can easily predominate the complexity of a quantum algorithm and, thereby, impair potential quantum advantage. Our work presents a hybrid quantum-classical algorithm for efficient, approximate quantum state loading. More precisely, we use quantum Generative Adversarial Networks (qGANs) to facilitate efficient learning and loading of generic probability distributions - implicitly given by data samples - into quantum states. Through the interplay of a quantum channel, such as a variational quantum circuit, and a classical neural network, the qGAN can learn a representation of the probability distribution underlying the data samples and load it into a quantum state. The loading requires $\mathcal{O}(poly(n))$ gates and can thus enable the use of potentially advantageous quantum algorithms, such as Quantum Amplitude Estimation. We implement the qGAN distribution learning and loading method with Qiskit and test it using a quantum simulation as well as actual quantum processors provided by the IBM Q Experience. Furthermore, we employ quantum simulation to demonstrate the use of the trained quantum channel in a quantum finance application.

npj Quantum Information (2019)5:103 ; https://doi.org/10.1038/s41534-019-0223-2

Quantum generator in IBM's QGAN



- Two-dimensional projection of 3D energy shower

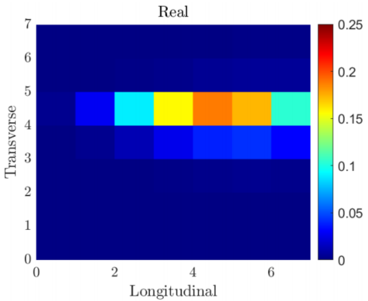


Image credits: Su Yeon Chang, Sofia Vallecorsa (CERN openlab)

To learn more...

Quantum Machine Learning in High Energy Physics

Wen Guan[1], Gabriel Perdue[2], Arthur Pesah[3], Maria Schuld[4], Koji Terashi[5], Sofia Vallecorsa[6], Jean-Roch Vlimant[7]

[1] University of Wisconsin-Madison, Madison, WI, USA 53706
[2] Fermi National Accelerator Laboratory, Fermilab Quantum Institute, PO Box 500, Batavia, IL, USA 60510-0500
[3] Technical University of Denmark, DTU Compute, Lyngby, DK
[4] University of KwaZulu-Natal School of Chemistry and Physics, Durban, ZA 4000
[5] ICEPP, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, JP 300-1153
[6] CERN, IT, 1, Esplanade des Particules, Geneva, CH 1211
[7] California Institute of Technology, PMA, Pasadena, CA, USA 91125-0002

Image credits: `https://arxiv.org/pdf/2005.08582.pdf`

Quantum machine learning and its supremacy in high energy physics

Kapil K. Sharma

DY Patil International University,
Sect-29, Nigdi Pradhikaran, Akurdi, Pune, Maharashtra 411044, India

Image credits: Modern Physics Letters A

Part XX

Errare quantum est: quantum error correction

Quantum computers and errors: a problem without solution?

- A series of problems seem to prevent the possibility of fault-tolerant quantum computing:
  - The no-cloning theorem
  - The collapse of the state after measurement
  - Unitary operations are continuous (not discrete)
- Despite these problems, quantum error correction is possible

A classical error-correcting code

- For quantum error correction, we can try use ideas from classical error correction
- The simplest approach is to use redundancy to code the information

$$0 \to 000$$

$$1 \to 111$$

- We use majority voting to "correct" errors

$$000, 001, 010, 100 \to 000$$

$$111, 110, 101, 011 \to 111$$

- In this way, we can correct errors that affect only a single bit

A quantum code that corrects flip errors in a single qubit

- We can extend the previous idea to the quantum domain
- We use three qubits to code one

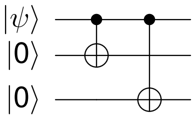
$$|0\rangle \to |000\rangle$$

$$|1\rangle \to |111\rangle$$

- By linearity

$$\alpha |0\rangle + \beta |1\rangle \to \alpha |000\rangle + \beta |111\rangle$$

- It does NOT violate the no-cloning theorem
- The circuit for encoding is simple

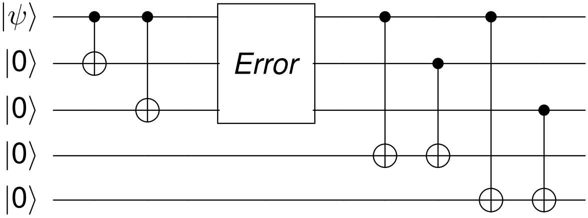


Detecting the flip of a single qubit

- How can we detect if a qubit has flipped without measuring it?
- We use ancillary qubits to detect the error *syndrome*

$$|000\rangle, |111\rangle \to |00\rangle \qquad |001\rangle, |110\rangle \to |01\rangle$$

$$|010\rangle, |101\rangle \to |10\rangle \qquad |100\rangle, |011\rangle \to |11\rangle$$



Correcting the flip of a single qubit

- Now, we can measure the syndrome qubits and apply the appropriate error correction operation
  - If we obtain 00, we do nothing
  - If we obtain 01, we invert the third qubit
  - If we obtain 10, we invert the second qubit
  - If we obtain 11, we invert the first qubit
- In this way, we are also "discretizing" the errors. If the error acts as

  $$|000\rangle \to \sqrt{1-\epsilon^2}\,|000\rangle + \epsilon\,|001\rangle$$

  with the syndrome register we would have

  $$\sqrt{1-\epsilon^2}\,|000\rangle\,|00\rangle + \epsilon\,|001\rangle\,|01\rangle$$

  and, when we measure, it will collapse to either $|000\rangle\,|00\rangle$ or $|001\rangle\,|01\rangle$ and then we can correct

A quantum code to correct phase inversion errors

- Another type of error that a qubit can suffer is a phase inversion (as if an unwanted Z gate was applied to it)

$$\alpha\,|0\rangle + \beta\,|1\rangle \to \alpha\,|0\rangle - \beta\,|1\rangle$$

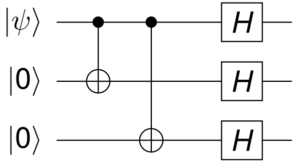
- We can use the following code

$$|0\rangle \to |+++\rangle$$

$$|1\rangle \to |---\rangle$$

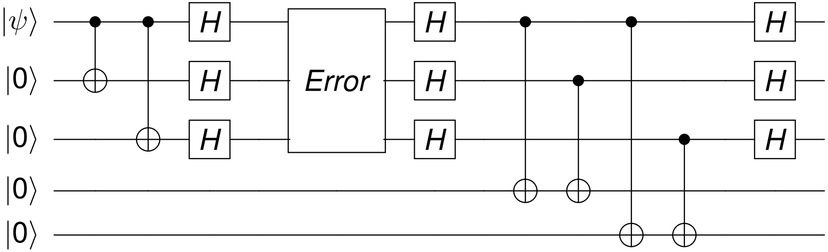
- By linearity

$$\alpha\,|0\rangle + \beta\,|1\rangle \to \alpha\,|+++\rangle + \beta\,|---\rangle$$



Detecting phase inversion errors

- It is almost equal to the case of qubit flip
- We only need to take into account that $HZH = X$
- But X acts as a qubit flip
- Thus, we can use the following circuit



Correcting phase inversion errors

- Again, it is enough to measure the syndrome and measure accordingly
  - If we obtain 00, we do nothing
  - If we obtain 01, we apply Z to the third qubit
  - If we obtain 10, we apply Z to the second qubit
  - If we obtain 11, we apply Z to the third qubit
- And we obtain “discretization” of the errors for free. For instance

$$|- - -\rangle \to \sqrt{1-\epsilon^2}\,|- - -\rangle + \epsilon\,|- + -\rangle$$

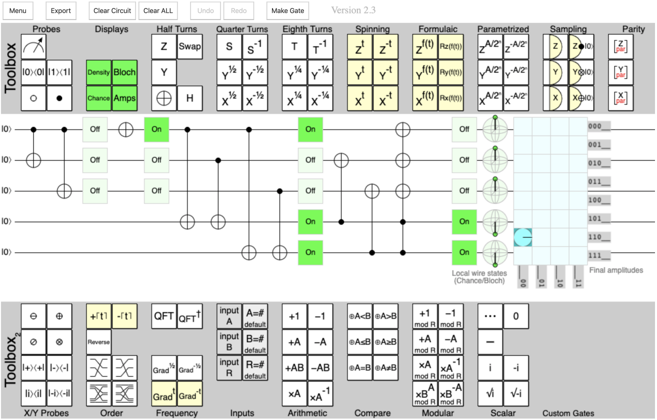
with the syndrome register would be

$$\sqrt{1-\epsilon^2}\,|- - -\rangle\,|00\rangle + \epsilon\,|- + -\rangle\,|10\rangle$$

and, when we measure, it would collapse to either $|- - -\rangle\,|00\rangle$ or $|- + -\rangle\,|10\rangle$ and we know how to correct.

The codes in action

- Seeing the codes in action can be illuminating
- We will use Quirk
- Qubit flip error-correcting code
- Phase inversion error-correcting code

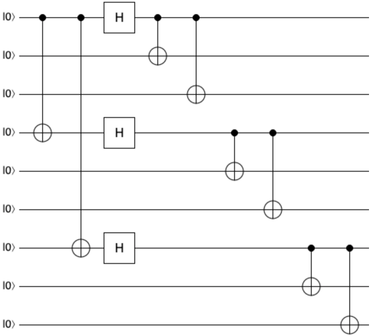


Combining both codes: Shor's code

- We can combine both codes using 9 qubits (Shor's code)
- The code is

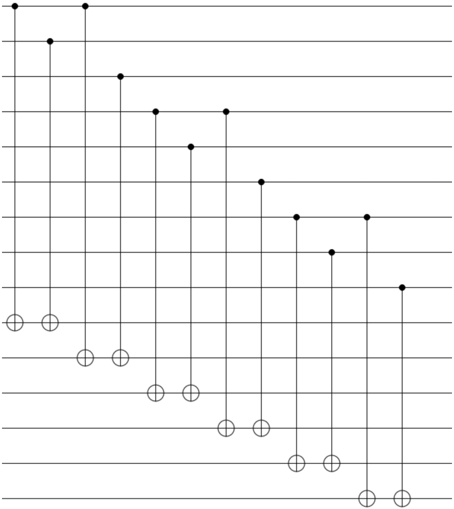
$$|0\rangle \to \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$

$$|1\rangle \to \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$



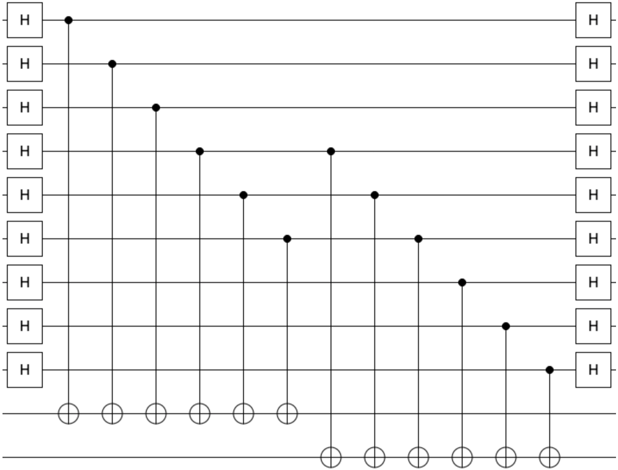
Detecting qubit flips with Shor's code

- We can use the following syndromes to detect qubit flips



Detecting phase inversion errors with Shor's code

- We can use the following syndromes to detect phase inversions



Correcting any one-qubit error with Shor's code

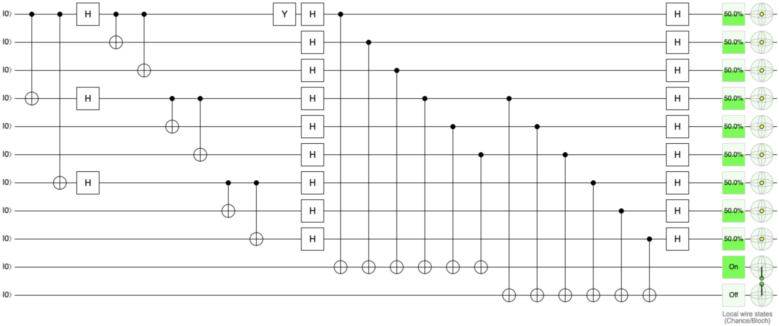
- Discretization, again, allows us to use Shor's code to correct errors in a single qubit
- The key is to note that $Y = iXZ$ and that any one-qubit gate G can be written in the form

$$G = a_I I + a_x X + a_Y Y + a_Z Z$$

- Each error will have different syndromes for qubit flip (X) and for qubit-inversion (Z)
- When we measure, we discretize and obtain a concrete type of error that we can correct with the syndrome information

Shor's code in action

- Qubit flip syndrome
- Phase inversion syndrome



Logical operations with Shor's code

- We can perform quantum transformations on the logical qubits of Shor's code
- For instance, if we use $Z_1 Z_2 Z_3 Z_4 Z_5 Z_6 Z_7 Z_8 Z_9$ its effect is

$$|0\rangle_L \to |1\rangle_L \qquad |1\rangle_L \to |0\rangle_L$$

  so it acts like X
- Analogously, $X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9$ acts like a Z gate

$$|0\rangle_L \to |0\rangle_L \qquad |1\rangle_L \to -|1\rangle_L$$

- Other gates can be implemented in a similar way or with other techniques (gate teleportation)

Stabilizer codes

- Many quantum error-correcting codes are examples of stabilizer codes
- In stabilizer codes, all the states are fixed points of certain tensor products of Pauli matrices
- For instance, for the code spanned by $\{|000\rangle, |111\rangle\}$ the stabilizers are

$$I_1 \otimes I_2 \otimes I_3 \quad Z_1 \otimes Z_2 \otimes I_3 \quad Z_1 \otimes I_2 \otimes Z_3 \quad I_1 \otimes Z_2 \otimes Z_3$$

  which, under multiplication, form a (commutative) group generated by

$$Z_1 \otimes Z_2 \otimes I_3 \quad Z_1 \otimes I_2 \otimes Z_3$$

- It is important to notice that the eigenvalues of tensor products of Paulis are 1 or -1

Shor's code as a stabilizer code

- Both

$$\frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$

and

$$\frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$

are stabilized by

$$X_1X_2X_3X_4X_5X_6 \qquad X_4X_5X_6X_7X_8X_9$$

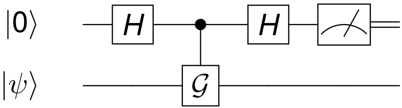
$$Z_1Z_2 \qquad Z_2Z_3 \qquad Z_4Z_5 \qquad Z_5Z_6 \qquad Z_7Z_8 \qquad Z_8Z_9$$

Measuring syndromes with stabilizer codes

- For a state to be in the code, it needs to be stabilized by each generator of the stabilizer group
- We can measure the syndrome associated to one of the generators $\mathcal{G}$ by using the circuit of the figure
- The state just before measuring is

$$\frac{1}{2}|0\rangle\left(|\psi\rangle+\mathcal{G}|\psi\rangle\right)+\frac{1}{2}|1\rangle\left(|\psi\rangle-\mathcal{G}|\psi\rangle\right)$$

so if $\mathcal{G}|\psi\rangle = |\psi\rangle$ we will measure $|0\rangle$ and if $\mathcal{G}|\psi\rangle = -|\psi\rangle$ we will measure $|1\rangle$



Surface codes

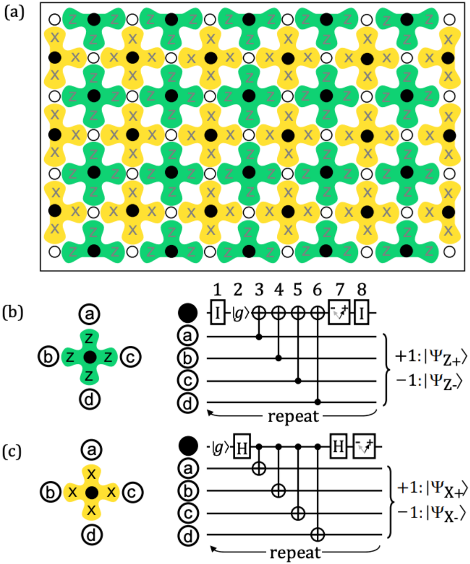


Image credits: Fowler, Mariantoni, Martinis, Cleland https://arxiv.org/abs/1208.0928

Fault-tolerant quantum computing

- In fault-tolerant quantum computing, every operation (state preparation, gate application, error correction and measurements) is performed with a probability $O(p^2)$ of two errors occurring in a given block (p being the probability of an individual error)
- This and code concatenation, allows us to prove a very important result: the threshold theorem

Theorem (Threshold theorem - informal version)

If the error probability of each physical operation is below a threshold p_{th}, it is possible to reduce arbitrarily the error probability of any quantum computation without increasing too much the size of the circuit, under reasonable assumptions on the error model.

Part XXI

Ad astra: quantum supremacy and the future of quantum computing

What is quantum supremacy?

Quantum supremacy

Quantum supremacy is a term coined by John Preskill that refers to the moment in which a quantum computer performs a task in much less time than it would take on a classical computer

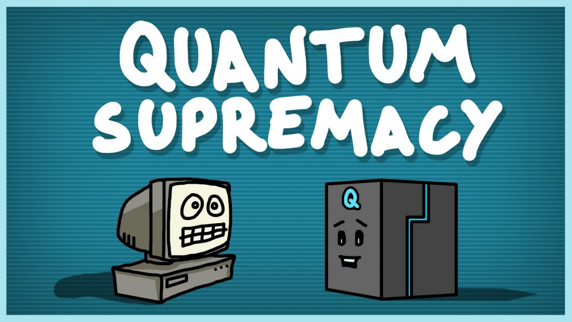


Image credits: Domain of Science https://www.youtube.com/watch?v=90U_SmKyfGI

Google's quantum supremacy

Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5

Received: 22 July 2019

Accepted: 20 September 2019

Published online: 23 October 2019

Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G. S. L. Brandao[1,4], David A. Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[1,5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Sergey Knysh[1], Alexander Korotkov[1,8], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[9], Salvatore Mandrà[3,10], Jarrod R. McClean[1], Matthew McEwen[5], Anthony Megrant[1], Xiao Mi[1], Kristel Michielsen[11,12], Masoud Mohseni[1], Josh Mutus[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Murphy Yuezhen Niu[1], Eric Ostby[1], Andre Petukhov[1], John C. Platt[1], Chris Quintana[1], Eleanor G. Rieffel[3], Pedram Roushan[1], Nicholas C. Rubin[1], Daniel Sank[1], Kevin J. Satzinger[1], Vadim Smelyanskiy[1], Kevin J. Sung[1,13], Matthew D. Trevithick[1], Amit Vainsencher[1], Benjamin Villalonga[1,14], Theodore White[1], Z. Jamie Yao[1], Ping Yeh[1], Adam Zalcman[1], Hartmut Neven[1] & John M. Martinis[1,5*]

Image credits: Arute et al. https://www.nature.com/articles/s41586-019-1666-5

How did Google achieve quantum supremacy?

- Google used a 53 qubit chip (Sycamore) to run (pseudo)-random circuits with one and two qubit gates
- This task is not especially useful in practice (might have an application in certified random bit generation)
- We believe this task to be impossible to do efficiently on a classical computer (it would cause a collapse of the polynomial hierarchy)

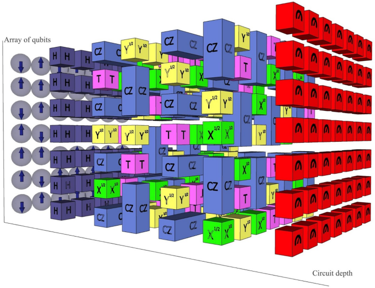


Image credits: https://ai.googleblog.com/

When can we simulate quantum circuits efficiently?

- The complexity of simulating quantum circuits on a classical computer does not depend only on the number of qubits and gates
- If parts of the circuit are not entangled, we can simulate them independently
- If the gates used in the circuit come from restricted sets, we may be able to simulate them efficiently

Theorem (Gottesman - Knill)

Any circuit that only uses gates from the set $\{H, X, CNOT, S\}$ *plus preparation of the state* $|0\rangle$ *and measurements in the computational basis can be simulated efficiently with a classical algorithm*

Sampling strings from random circuits

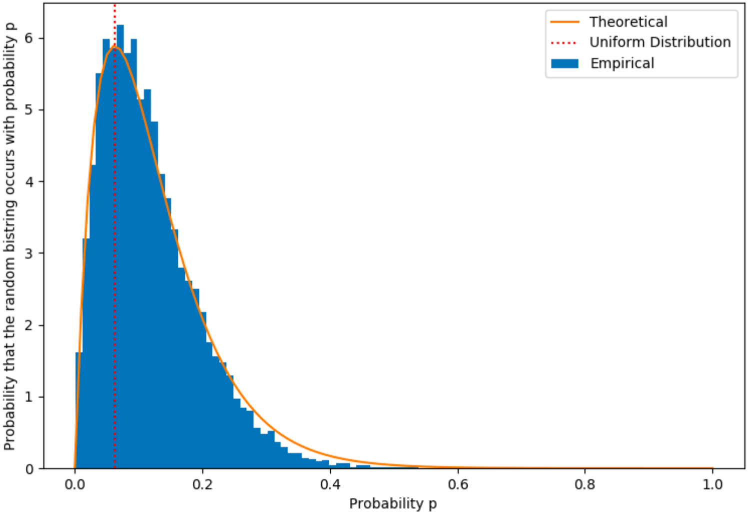


Image credits: M. Sohaib Alam and Will Zeng Medium post

Quantum supremacy experiment results

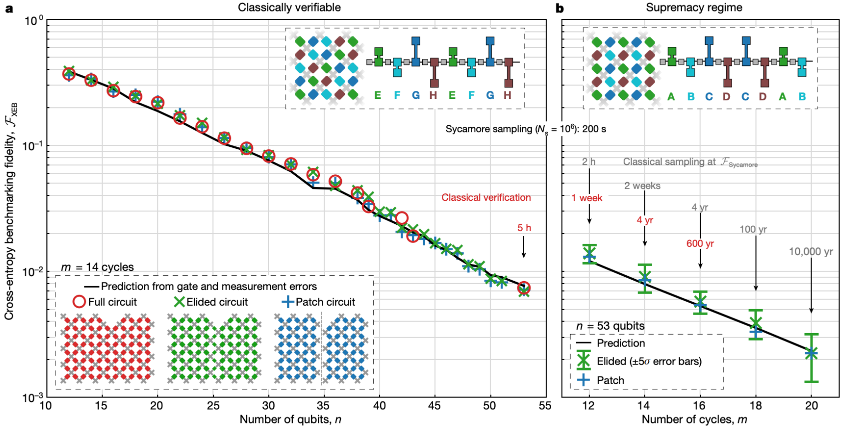


Image credits: Arute et al. https://www.nature.com/articles/s41586-019-1666-5

IBM's challenge to Google's quantum supremacy

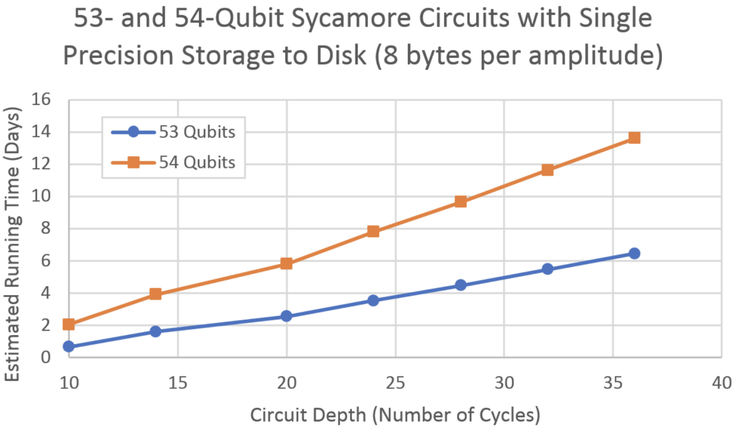


Image credits: https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/

Quantum circuit simulation in polynomial space

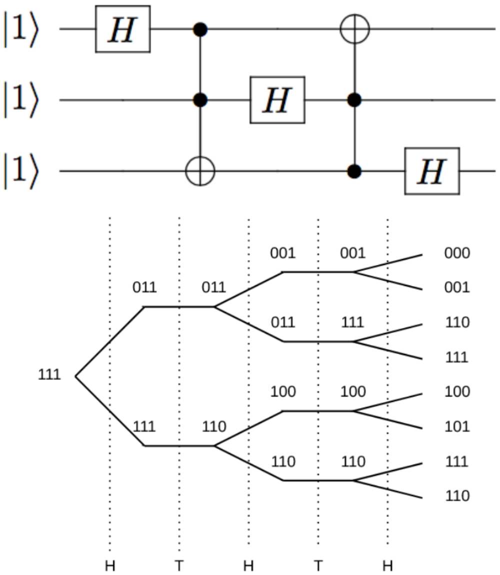


Image credits: Andrew Shi https://arxiv.org/abs/1710.09364

Science

REPORTS

Cite as: H.-S. Zhong *et al.*, *Science* 10.1126/science.abe8770 (2020).

Quantum computational advantage using photons

Han-Sen Zhong[1,2]*, Hui Wang[1,2]*, Yu-Hao Deng[1,2]*, Ming-Cheng Chen[1,2]*, Li-Chao Peng[1,2], Yi-Han Luo[1,2], Jian Qin[1,2], Dian Wu[1,2], Xing Ding[1,2], Yi Hu[1,2], Peng Hu[3], Xiao-Yan Yang[3], Wei-Jun Zhang[3], Hao Li[3], Yuxuan Li[4], Xiao Jiang[1,2], Lin Gan[4], Guangwen Yang[4], Lixing You[3], Zhen Wang[3], Li Li[1,2], Nai-Le Liu[1,2], Chao-Yang Lu[1,2], Jian-Wei Pan[1,2]†

[1]Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China. [2]CAS Centre for Excellence and Synergetic Innovation Centre in Quantum Information and Quantum Physics, University of Science and Technology of China, Shanghai 201315, China. [3]State Key Laboratory of Functional Materials for Informatics, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China. [4]Department of Computer Science and Technology and Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China.

Boson sampling

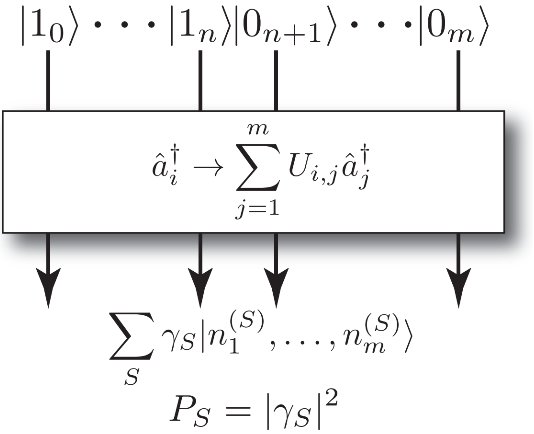


Image credits: Gard et al. https://arxiv.org/abs/1406.6767

Complexity of Boson sampling

$$
\begin{aligned}
\gamma_{\{1,2,3\}} &= U_{1,1}U_{2,2}U_{3,3} + U_{1,1}U_{3,2}U_{2,3} \\
&+ U_{2,1}U_{1,2}U_{3,3} + U_{2,1}U_{3,2}U_{1,3} \\
&+ U_{3,1}U_{1,2}U_{2,3} + U_{3,1}U_{2,2}U_{1,3} \\
&= \mathrm{Per}\begin{bmatrix} U_{1,1} & U_{2,1} & U_{3,1} \\ U_{1,2} & U_{2,2} & U_{3,2} \\ U_{1,3} & U_{2,3} & U_{3,3} \end{bmatrix}
\end{aligned}
$$

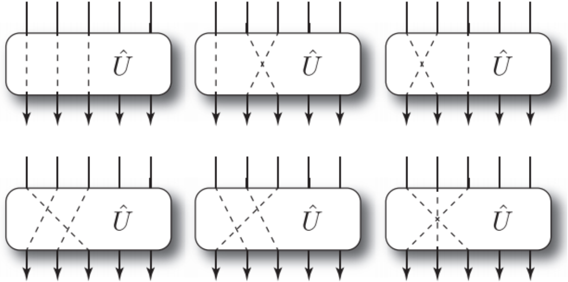


Image credits: Gard et al. https://arxiv.org/abs/1406.6767

Quantum computational advantage using photons: results

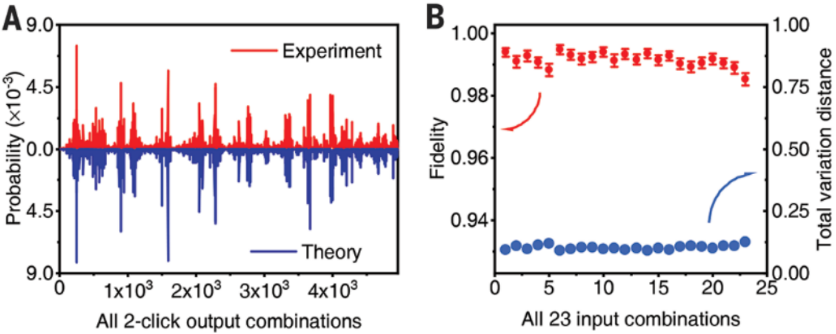


Image credits: Han-Sen Zhong et al. Science, 3 Dec 2020

Google Quantum Roadmap

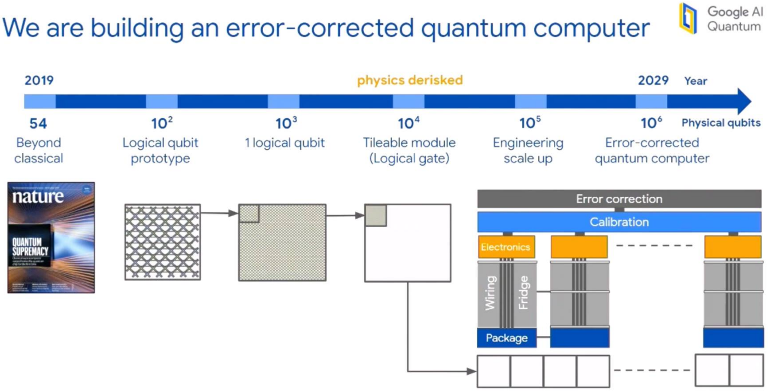


Image credits: H. Neven Google Quantum Summer Symposium 2020

Honeywell Quantum Roadmap

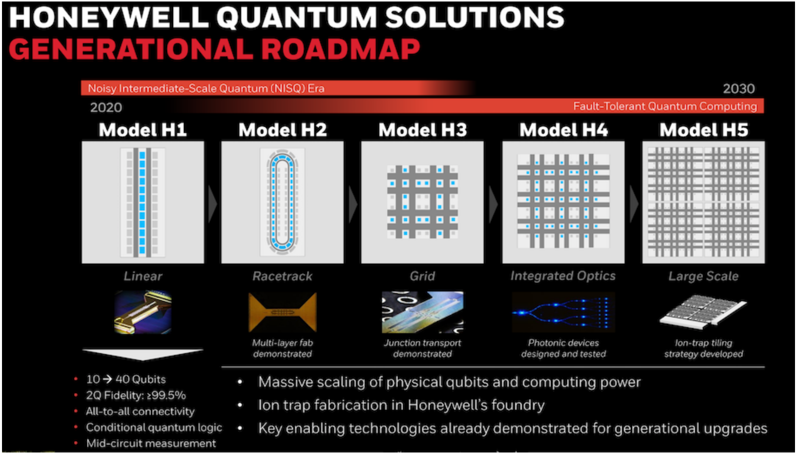


Image credits: Honeywell news post

IBM Quantum Roadmap

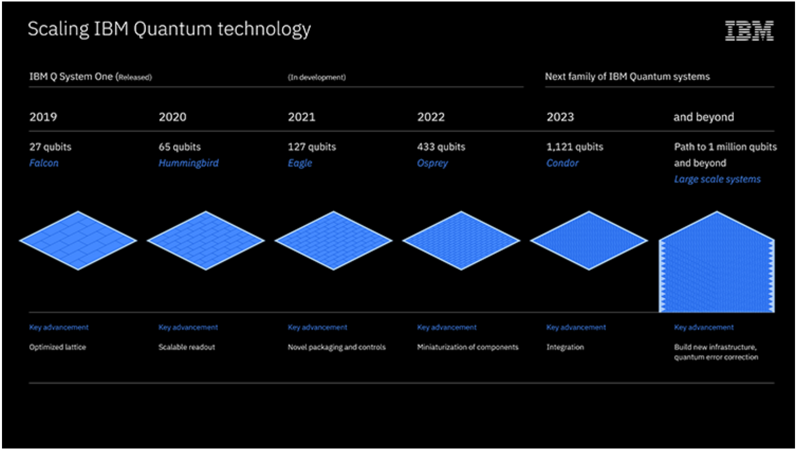


Image credits: `https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/`

IonQ Quantum Roadmap

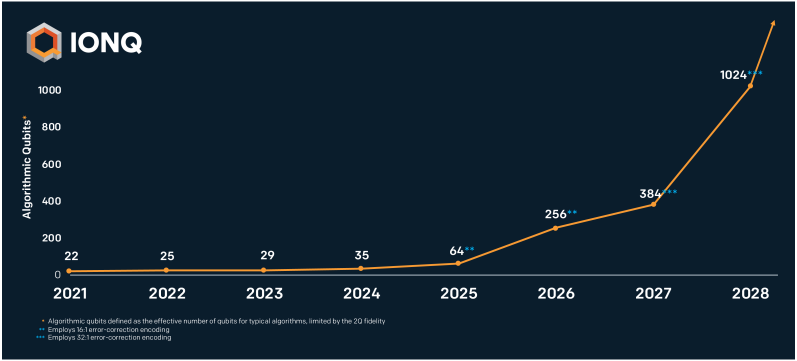


Image credits: https://ionq.com/posts/december-09-2020-scaling-quantum-computer-roadmap

My wishlist for the quantum computing future

- (More) confirmation of quantum supremacy
- Practical applications on NISQ computers
- Advances in qubit technologies
- Development of new quantum algorithms with (exponential) speed-ups
- Fault-tolerant quantum computing

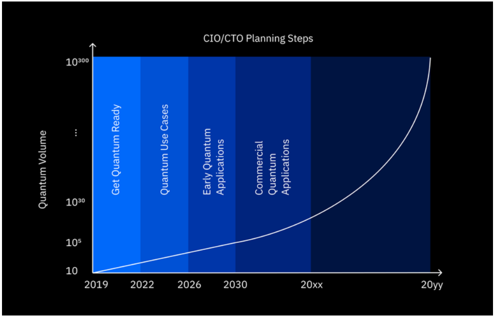


Image credits: Gartner/IBM

Thank you for your attention!

CERN openlab

Universidad de Oviedo

CERN