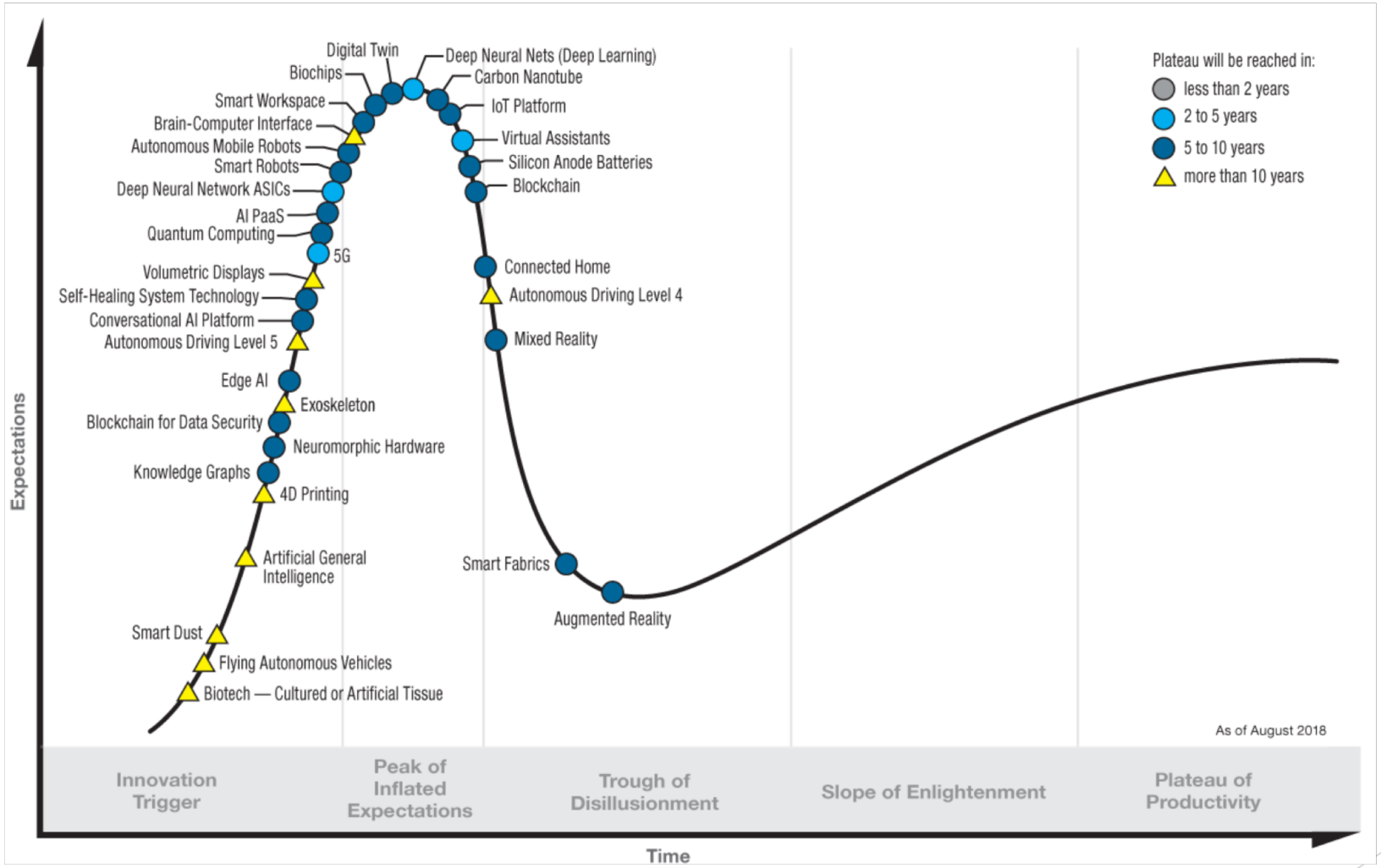
Quantum Machine Learning

Laurent Boué

Overall survey of the field

Live demo: Nearest neighbor classifier on a quantum computer

Expectation management: Gartner hype cycle for emerging technologies



Outline

From bits to **qubits**

Models of quantum computation:

- Domain-specific **quantum annealing**
- Universal quantum **logic gates**

Traditional quantum algorithms: a reality check

Quantum chips as **AI accelerators**

Live demo: nearest neighbor classifier on a quantum computer

From bits to qubits

$$0 \longrightarrow |0\rangle$$
$$1 \longrightarrow |1\rangle$$

General state of qubit: **superposition**
(linear combination basis vectors)

$$|\Psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$\alpha_0, \alpha_1 \in \mathbb{C}$$

Measurement of a quantum state:
"**wavefunction collapse**"

$$\text{Prob}(|1\rangle) = |\alpha_1|^2$$

$$\text{Prob}(|0\rangle) = |\alpha_0|^2$$

From bits to qubits

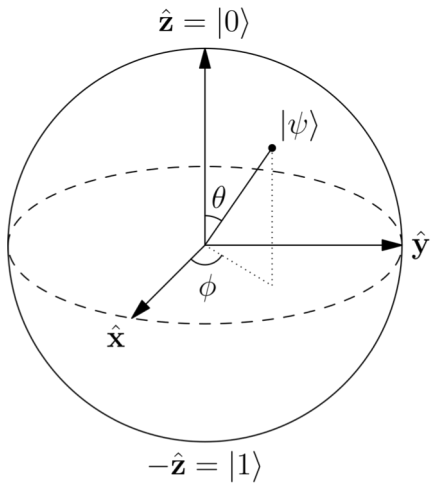
$$0 \longrightarrow |0\rangle$$
$$1 \longrightarrow |1\rangle$$

General state of qubit: **superposition**
(linear combination basis vectors)

$$|\Psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

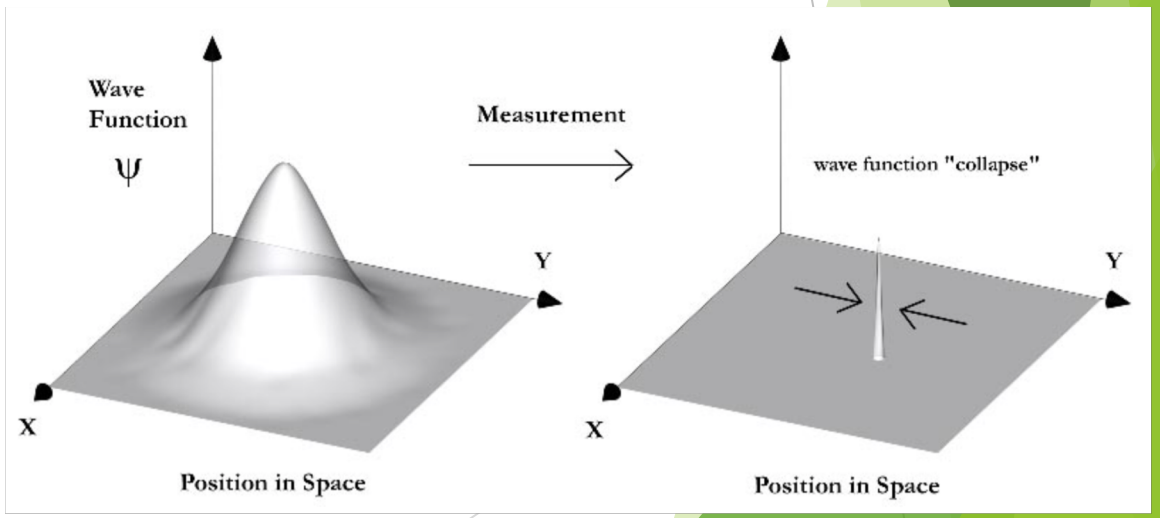
$$\alpha_0, \alpha_1 \in \mathbb{C}$$

Bloch sphere



Measurement of a quantum state:
"**wavefunction collapse**"

$$\text{Prob}(|1\rangle) = |\alpha_1|^2$$
$$\text{Prob}(|0\rangle) = |\alpha_0|^2$$



Quantum state defines a **probability density function**

Measurements sample that distribution

Multiple qubits

Tensor product between qubits

$$|\psi\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \qquad |\psi'\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$|\psi\rangle \otimes |\psi'\rangle = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

Not all vectors in tensor product space can be factorized as products!

"Bell state": $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ Cannot be expressed as $|\psi\rangle \otimes |\psi'\rangle$

Quantum entanglement Strong correlations between random variables that have **no classical analog**

Multiple qubits

Tensor product between qubits

$$|\psi\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \qquad |\psi'\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$|\psi\rangle \otimes |\psi'\rangle = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

Not all vectors in tensor product space can be factorized as products!

"Bell state": $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ Cannot be expressed as $|\psi\rangle \otimes |\psi'\rangle$

Quantum entanglement Strong correlations between random variables that have **no classical analog**

Simulating a quantum system on a classical computer:

Quantum computer with n qubits can be in any superposition of 2^n states simultaneously

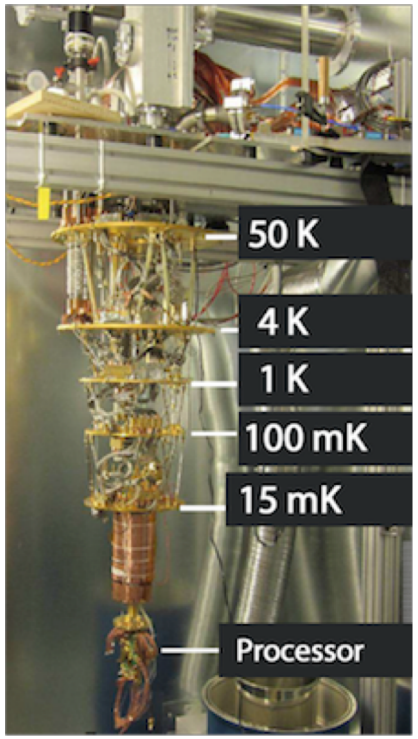
n = 50 qubits → 2^{50} complex numbers ~ 18 PB of memory. (PB = 1 million GB)

Quantum annealing (1/3)

D-Wave systems

First company to have commercial offering
Price tag ~ $15m

Shielded enclosure interfaced with IO subsystem



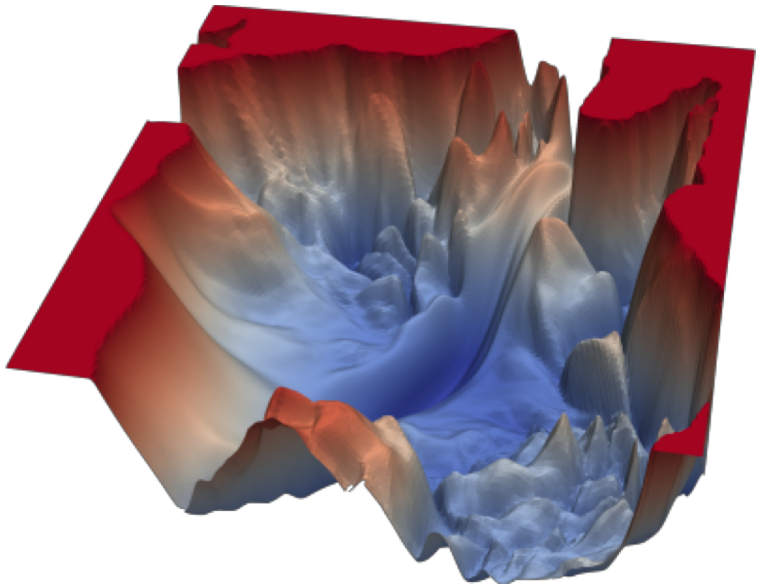
Liquid helium refrigeration

(for comparison interstellar space ~ 2.7K)

Customers include: Lockheed Martin, NASA, Google, Volkswagen, Temporal Defense Systems…

Quantum annealing (2/3)

Solving **optimization problems** on **rugged energy landscapes**

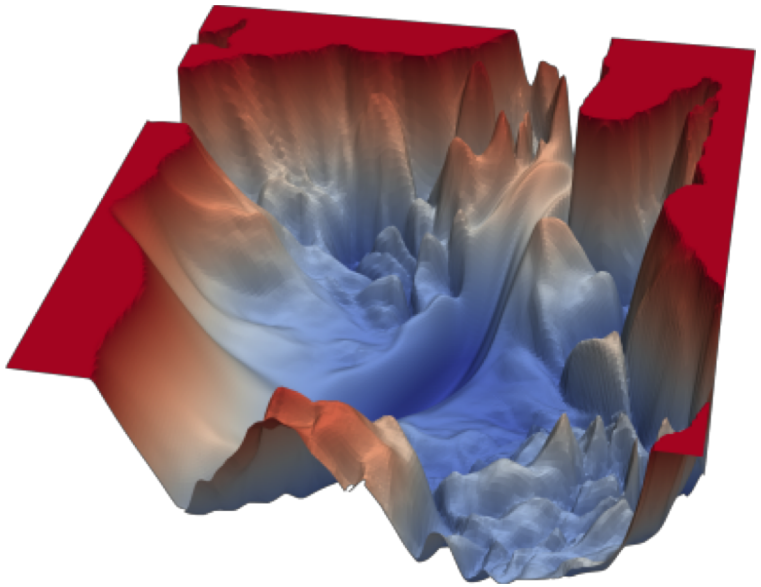


Protein folding
Modelling chemical reactions
Material discovery
Pharmaceutical drug design
Satisfiability problems, traveling salesman…
Portfolio management

Loss functions in machine learning

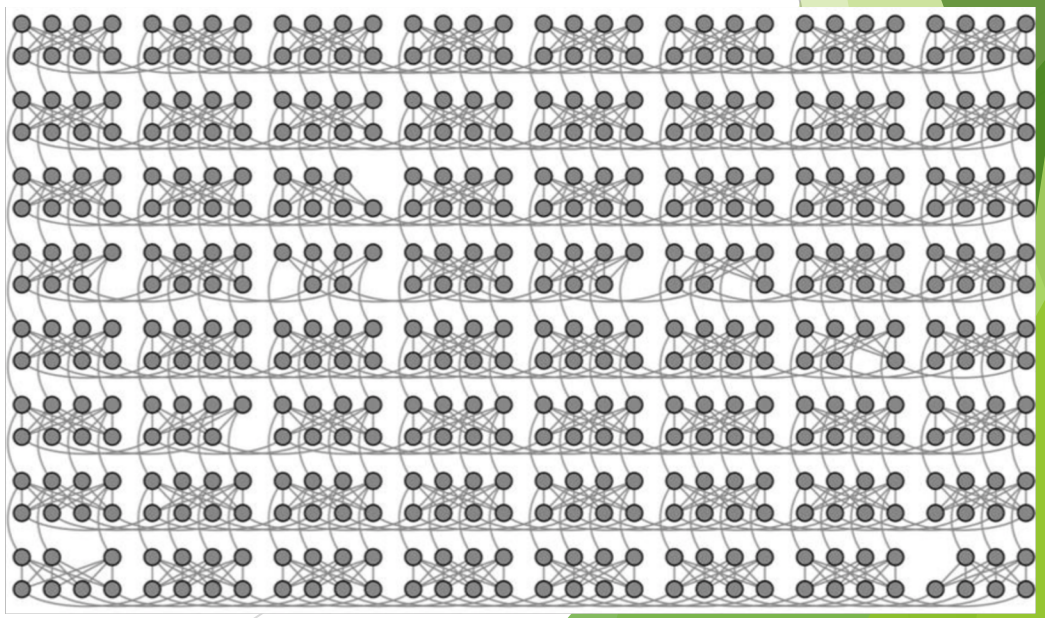
Quantum annealing (2/3)

Solving **optimization problems** on **rugged energy landscapes**



Protein folding
Modelling chemical reactions
Material discovery
Pharmaceutical drug design
Satisfiability problems, traveling salesman…
Portfolio management

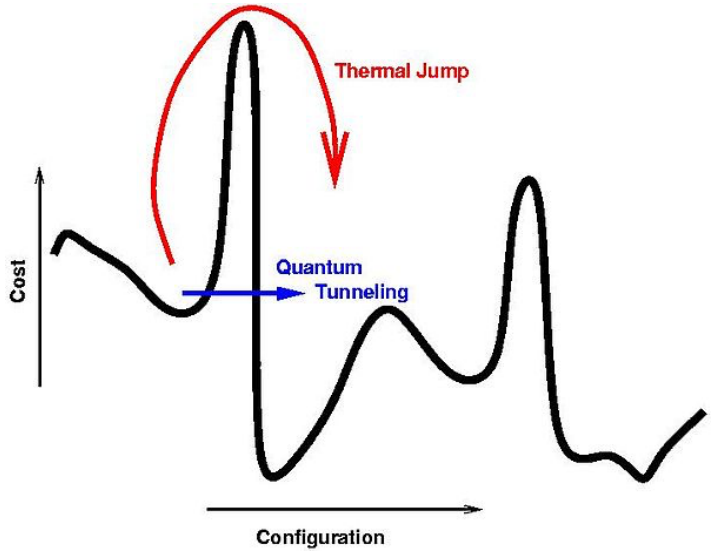
Loss functions in machine learning

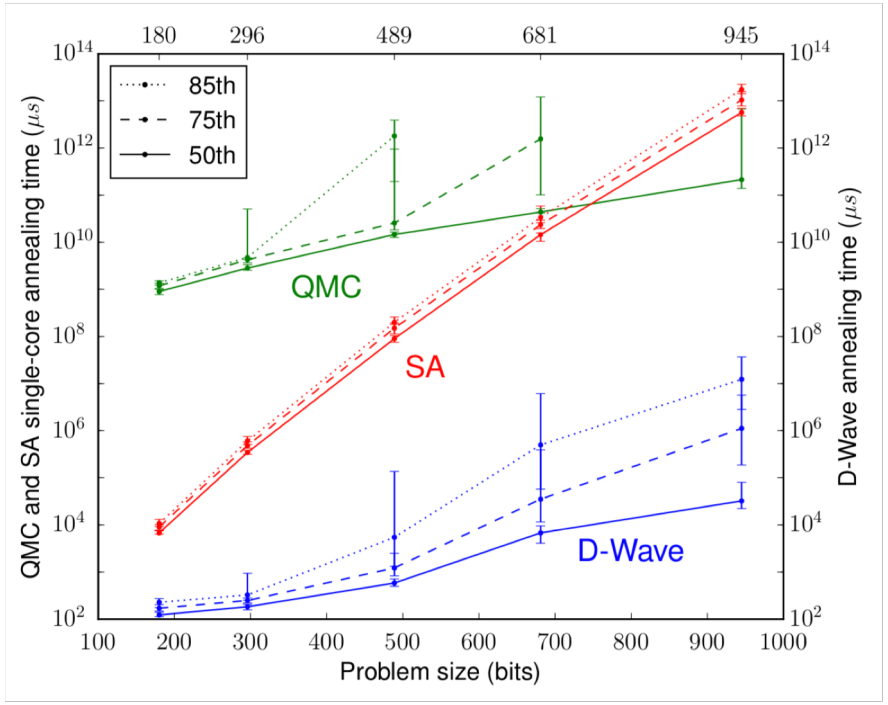


Constrained to **transverse field Ising model**
(~ quadratic unconstrained binary optimization **QUBO**)

$$H(\sigma) = -\sum_{\langle i\ j\rangle} J_{ij}\sigma_i\sigma_j - \mu\sum_j h_j\sigma_j$$

Quantum annealing (3/3)



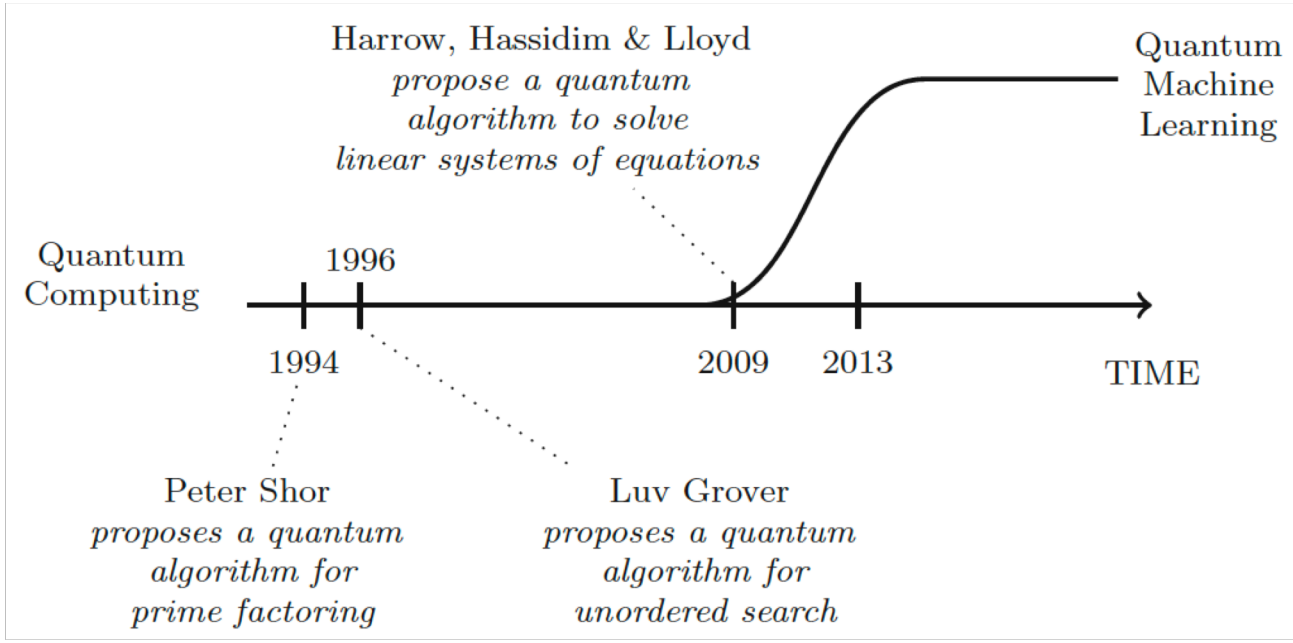


Google AI Quantum

Quantum annealing overcomes barriers **exponentially faster** than simulated annealing

~ 2000 qubits but **very specific architecture**

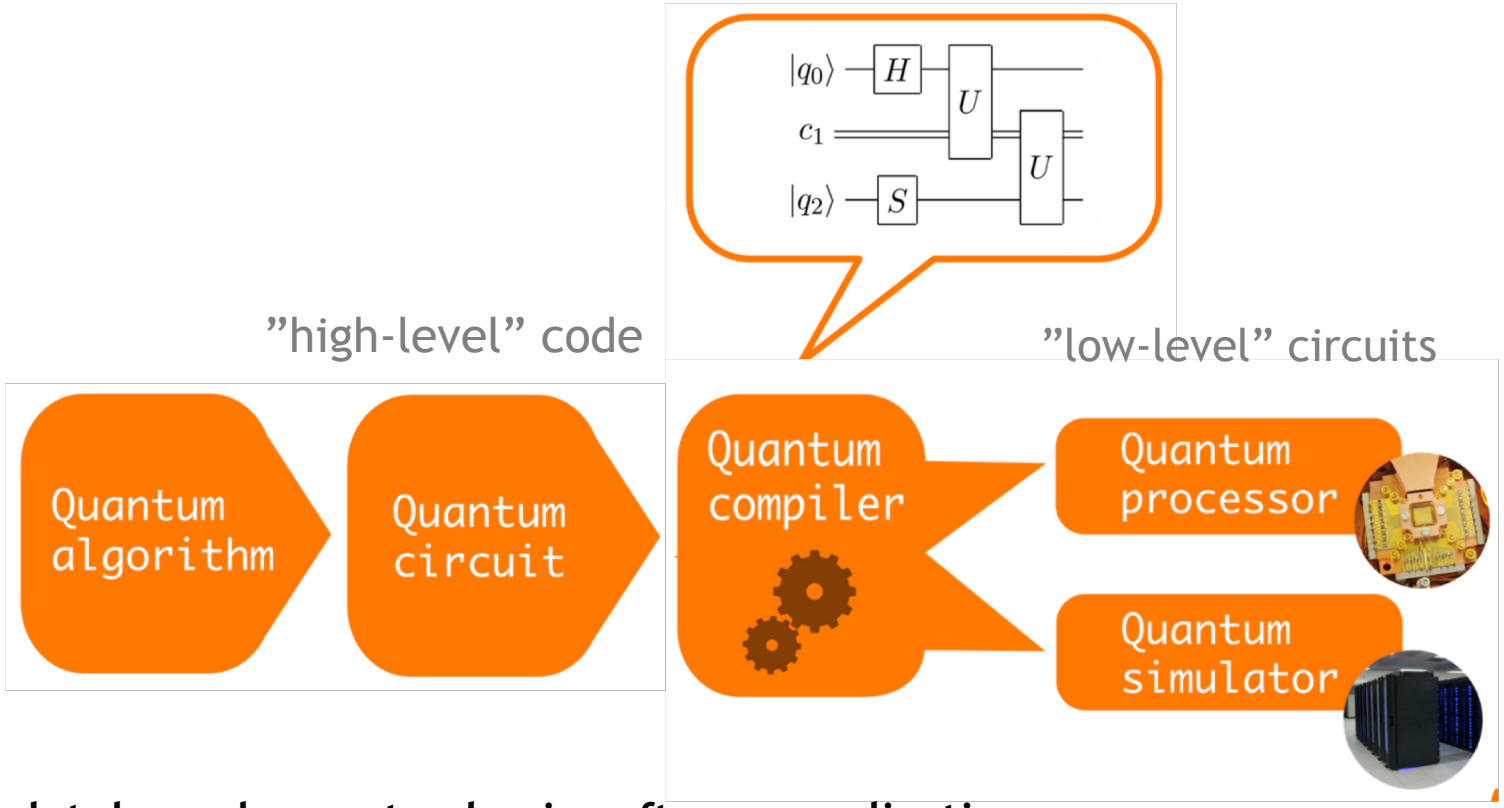
Traditional quantum algorithms



None of these algorithms can be formulated as quantum annealing problems…

Logic quantum gates: designing general (universal) circuits

Logic quantum gates (1/4)



Completely analogous to classic software application

Universal model: any algorithms can be expressed using this quantum gates

In practice, algorithms are usually written at gate-level
(compilers are still very new…)

Logic quantum gates (2/4)

Common gates include

Gate	Name	Matrix
X	Pauli-X or NOT gate	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Z	Pauli-Z gate	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
H	Hadamard gate	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Rx(θ)	Rotation around X	$\begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$
Ry(θ)	Rotation around Y	$\begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$
CNOT, CX	Controlled-NOT	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Unitary matrices:

2x2 for single qubit
4x4 for 2-qubit gate…

NOT gate

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

CNOT gate

$$|00\rangle \mapsto |00\rangle \qquad |10\rangle \mapsto |11\rangle$$
$$|01\rangle \mapsto |01\rangle \qquad |11\rangle \mapsto |10\rangle$$

Gates can be parametrized by a continuous value:

Playground for linear algebra!

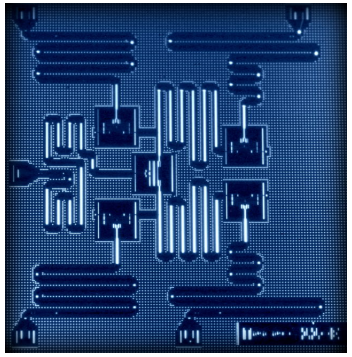
Logic quantum gates (3/4)

Manufacturer	Name/Codename/Designation	Architecture	Layout	Socket	Fidelity	Qubits	Release date
Google	Bristlecone	Superconducting	6×12 lattice	N/A	99% (readout) 99.9% (1 qubit) 99.4% (2 qubits)	72 qb [3][4]	5 March 2018
IBM	IBM Q 50 prototype	Superconducting	N/A	N/A	N/A	50 qb[6]	
IBM	IBM Q Experience 5	Superconducting	N/A	N/A	N/A	5 qb	2016 [1]
Google	N/A	Superconducting	7×7 lattice	N/A	99.7% [1]	49 qb [2]	Q4 2017 (planned)
Intel	Tangle Lake	Superconducting	N/A	108-pin cross gap	N/A	49 qb [9]	9 January 2018
Google	N/A	Superconducting	N/A	N/A	99.5% [1]	20 qb	2017
IBM	IBM Q 20	Superconducting	N/A	N/A	N/A	20 qb [6]	10 November 2017
Rigetti	19Q	Superconducting	N/A	N/A	N/A	19 qb [10]	December 2017
IBM	IBM Q 17	Superconducting	N/A	N/A	N/A	17 qb [5]	17 May 2017
Intel	17-Qubit Superconducting Test Chip	Superconducting	N/A	40-pin cross gap	N/A	17 qb [7][8]	10 October 2017
IBM	IBM Q Experience 16	Superconducting	2×8 lattice	N/A	N/A	16 qb [5]	17 May 2017

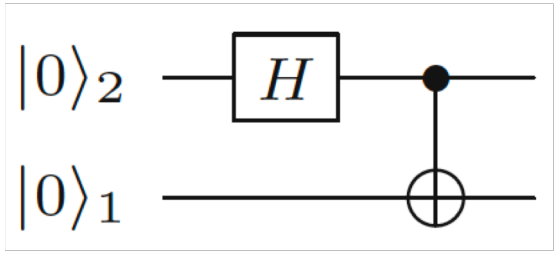
Existing systems: **Decoherence time ~ maximum 100 gates**

Error correction: **no-cloning theorem**…

Logic quantum gates (4/4)



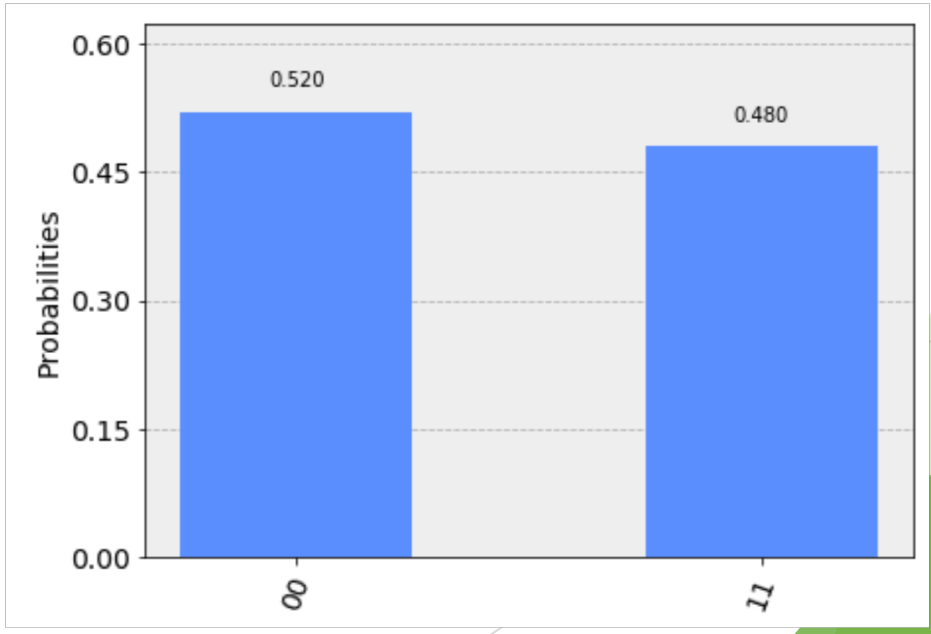
Example circuit: Create entangled state on IBM Q5 (bow-tie architecture)



$$= \frac{1}{\sqrt{2}}(|0\rangle_2 \otimes |0\rangle_1 + |1\rangle_2 \otimes |1\rangle_1).$$

```
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, execute, BasicAer
from qiskit.tools.visualization import plot_histogram, plot_bloch_multivector
```

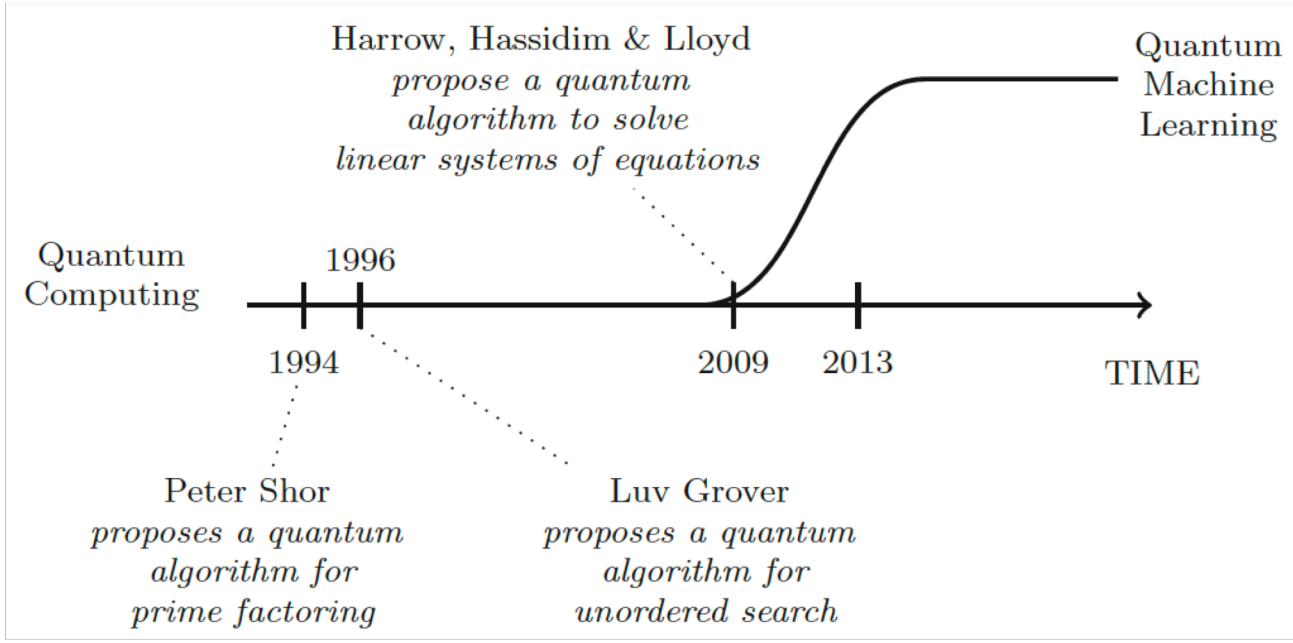
```
q = QuantumRegister(2)
c = ClassicalRegister(2)
circuit = QuantumCircuit(q, c)
circuit.h(q[0])
circuit.cx(q[0], q[1])
circuit.measure(q, c)
job = execute(circuit, backend, shots=100)
plot_histogram(job.result().get_counts(circuit))
```



Sampling superposition state 100 times

Longer circuit would suffer from noisy results…

Traditional quantum algorithms: back to the “usual suspects”



Traditional quantum algorithms: back to the “usual suspects” (1/3)

Shor’s algorithm: Given an integer N, find its prime factors

Break public-key cryptography schemes (RSA encryption...)

$$566{,}557 \times 896{,}479 = 507{,}906{,}452{,}803$$

prime numbers — 39-bit integer

Multiplication is easy to verify but factorization is much harder

Best classical result ~ exponential in n

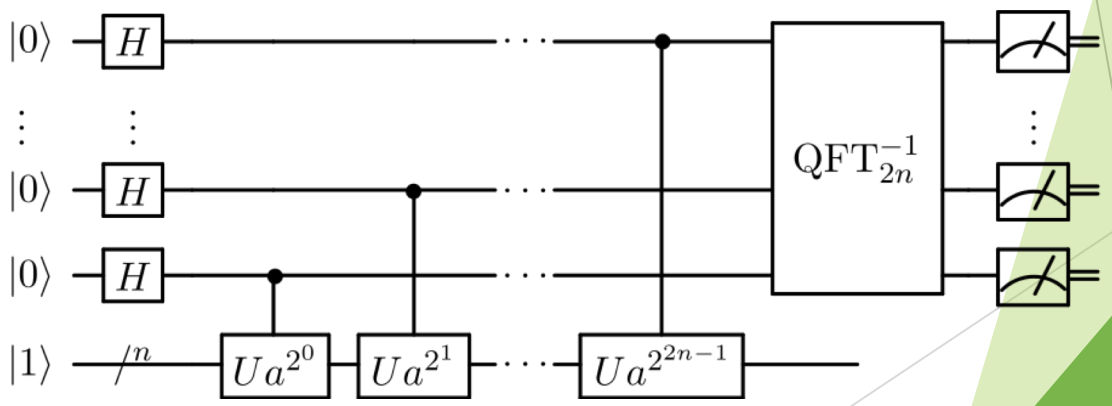
$O(n^3 \log n)$

Quantum: polynomial in n

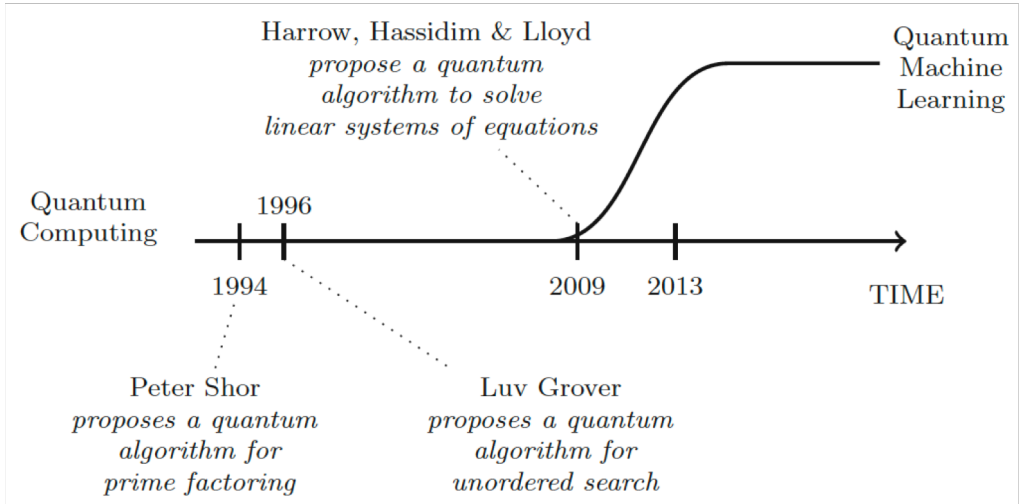
exponential speedup

$O(n^2 \log n \log\log n)$

Number of gates ~ 10^8 for n = 2048…



Traditional quantum algorithms: back to the "usual suspects" (2/3)



Grover's search algorithm

Search for subset of items that satisfy a criteria in an unstructured set of N items

Quadratic speedup in the expected number of necessary queries

$O(N)$ v.s $O(\sqrt{N})$

classical computing — quantum computing

(basic idea: evaluate search predicate on use superposition of all items in the database)

Traditional quantum algorithms: back to the “usual suspects” (3/3)

Solving linear systems: HHL algorithm

Given A and b, find x such that: $A\vec{x} = \vec{b}$

Traditional quantum algorithms: back to the “usual suspects” (3/3)

Solving linear systems: HHL algorithm

Given A and b, find x such that: $A\vec{x} = \vec{b}$

Exponential speedup in size of matrix NxN

Traditional quantum algorithms: back to the "usual suspects" (3/3)

Solving linear systems: HHL algorithm

Given A and b, find x such that: $A\vec{x} = \vec{b}$

Exponential speedup in size of matrix NxN

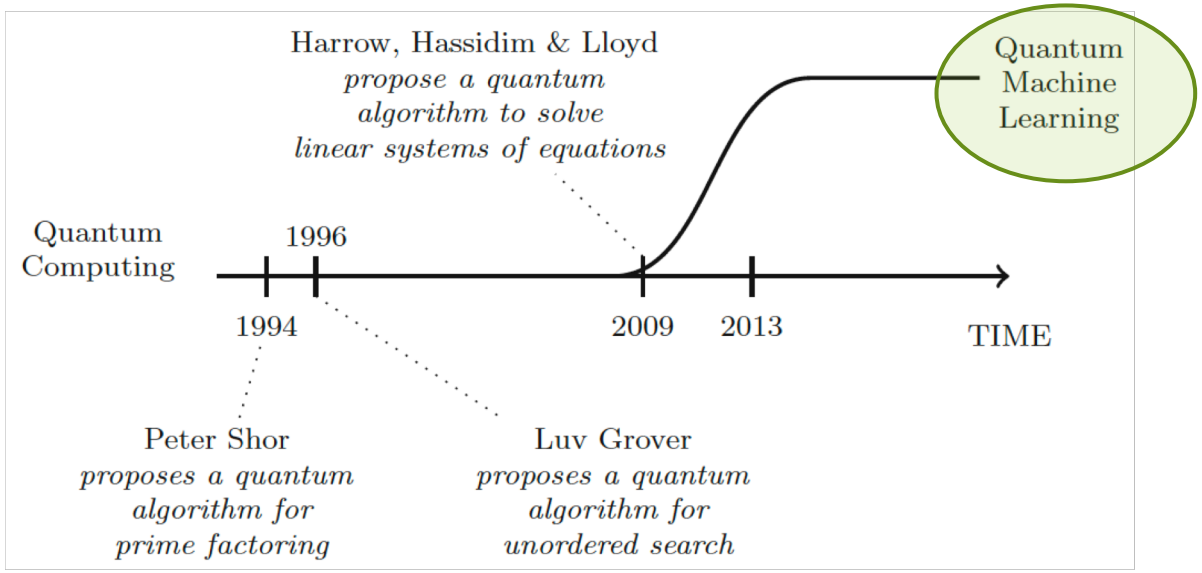
Unfortunately many constraints: sparse matrices with low condition number, expectation values of projections on x (no access to solution vector itself)…

Experimental (very complicated) quantum circuit implementing HHL: 2x2 matrix inversion…

"Experimental Quantum Computing to Solve Systems of Linear Equation", PRL

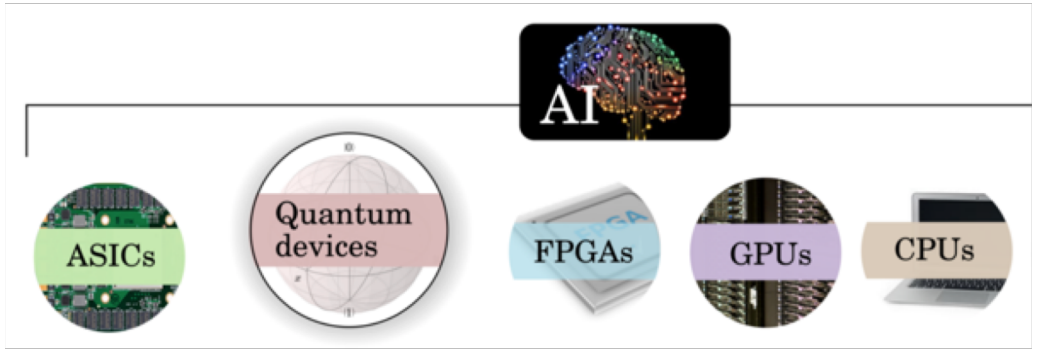
State preparation (loading data into quantum computer) and gathering **read-out statistics** (more or less) destroy quantum advantage…

What are quantum computers good for?

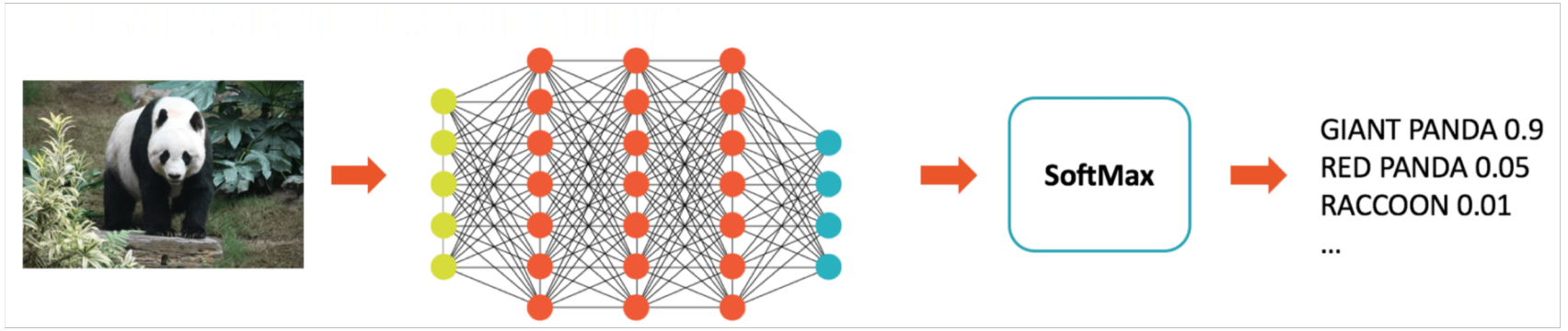


an emerging interdisciplinary research area

"Trendy" idea: use early generation **quantum devices as hardware accelerators** for **machine learning** tasks

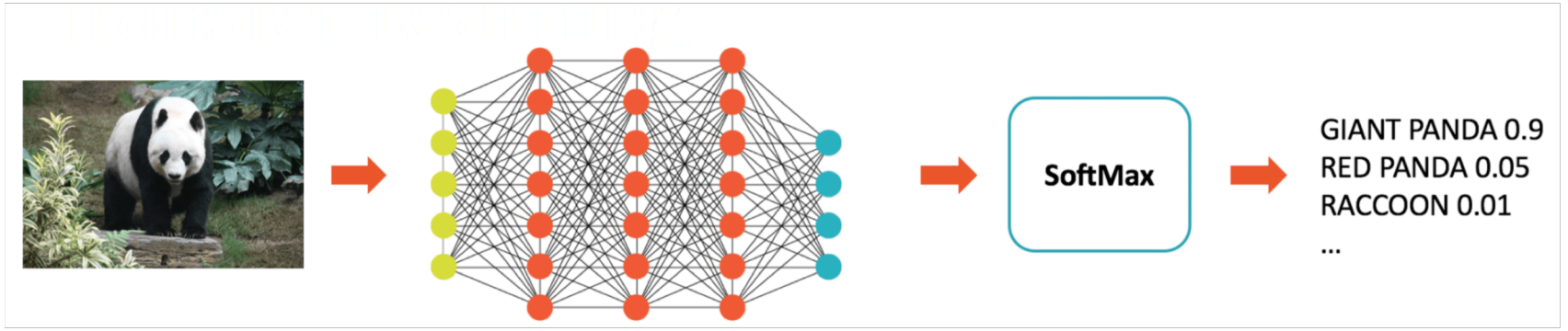


Deep learning in a nutshell: feedforward (memory-less) architectures



Linear mappings from layer to layer: fused matrix multiplications and accumulate (MACs)

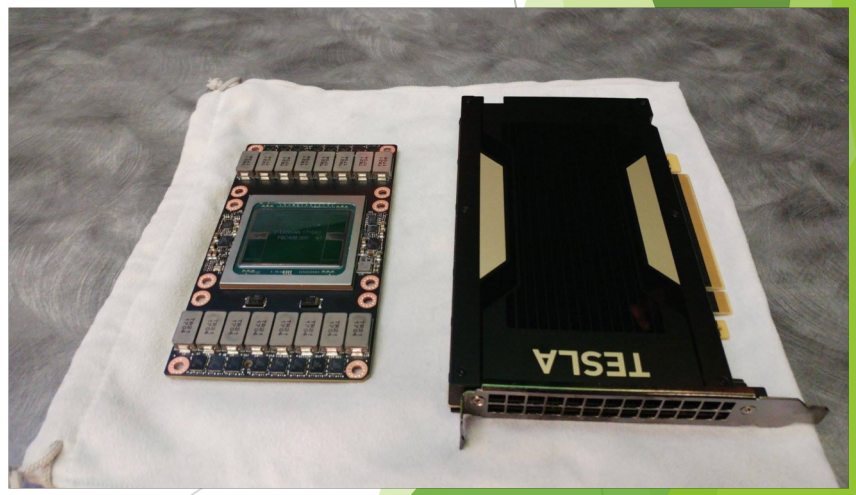
Deep learning in a nutshell: feedforward (memory-less) architectures



Linear mappings from layer to layer: fused matrix multiplications and accumulate (MACs)

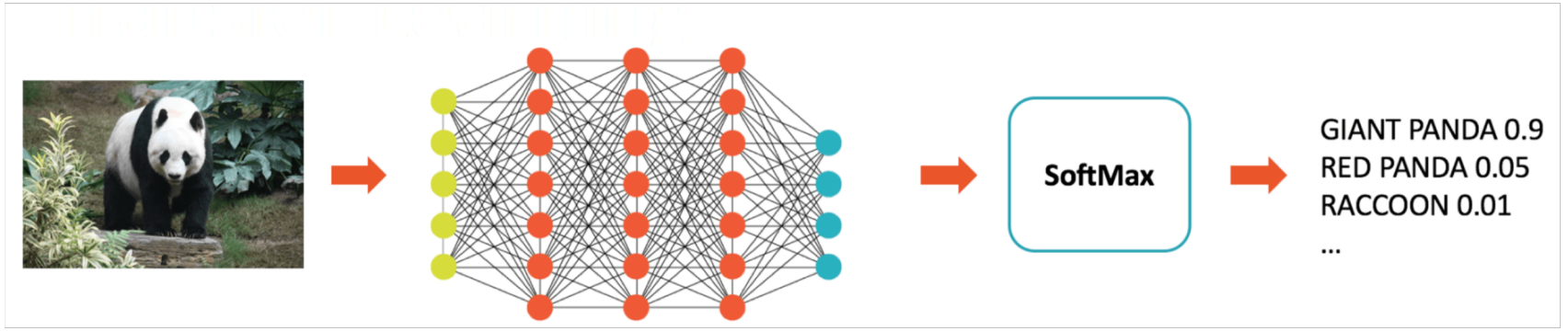
Current generation of specialized AI hardware accelerators:

$$D = \underbrace{\begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix}}_{\text{FP16 or FP32}} \underbrace{\begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix}}_{\text{FP16}} + \underbrace{\begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}}_{\text{FP16 or FP32}}$$



Tensor Core: 4x4 matrix multiply and accumulate / clock cycle
Tesla V100 GPU contains 640 Tensor Cores

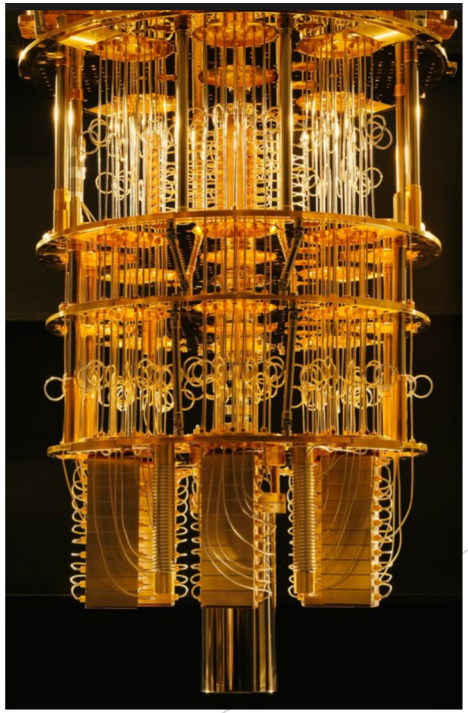
Deep learning in a nutshell: feedforward (memory-less) architectures



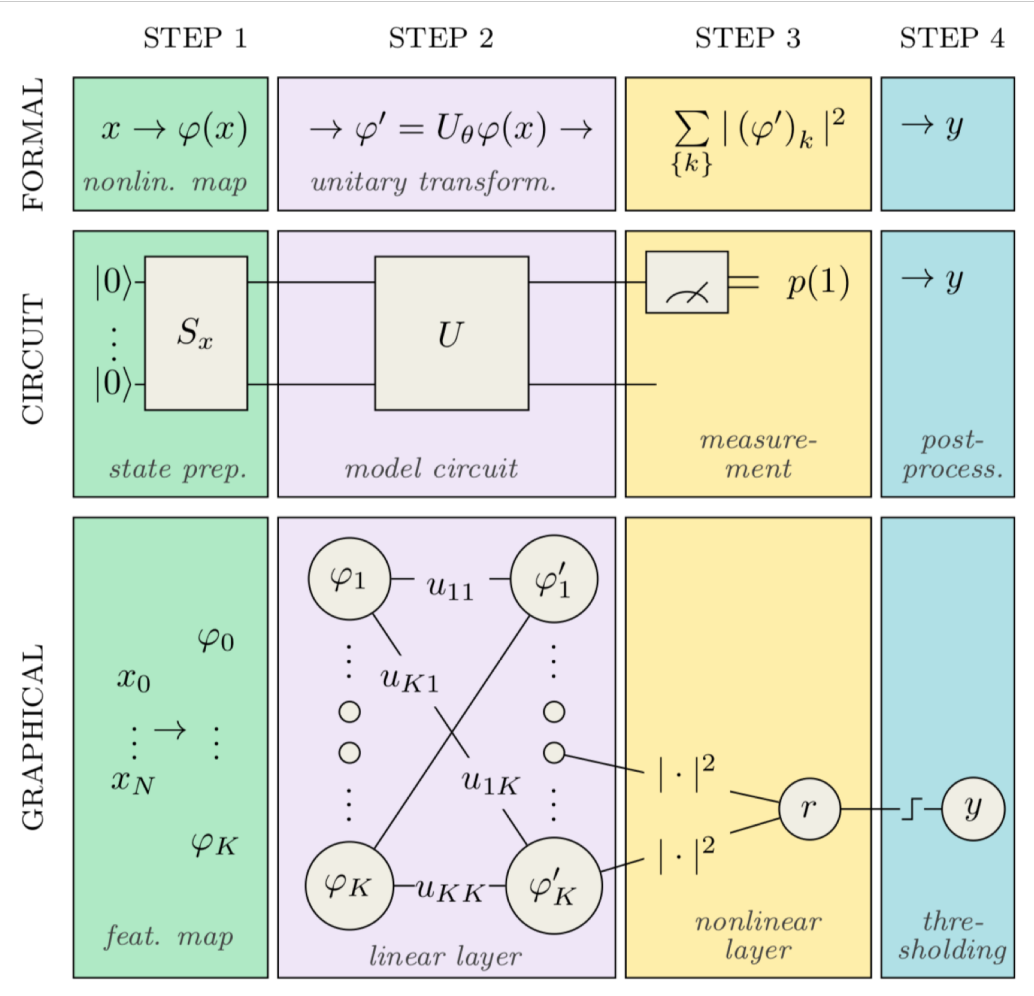
Linear mappings from layer to layer: fused matrix multiplications and accumulate (MACs)

Quantum gate = multiplication of large matrix (potentially infinite) with similarly large vector

Single operation on quantum computer!



Quantum chips as AI accelerators



"Circuit-centric quantum classifiers", arXiv:1804.00633

State preparation with tensor products generates **nonlinearities**

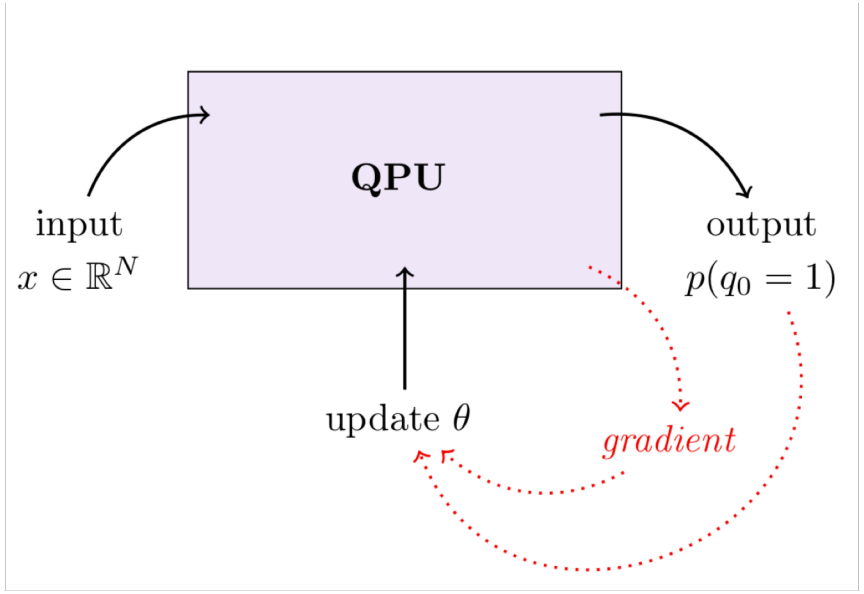
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \otimes \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_2 x_1 \\ x_2^2 \end{pmatrix}$$

Bottlenecks:

- Data encoding / Read out statistics
- Limited circuit depth
  (~ flexibility of ML model)

Variational circuits: hybrid quantum-classical technique for training!

Use QPU for "**hard to compute**" cost function



Gradients provided by rotations on input data

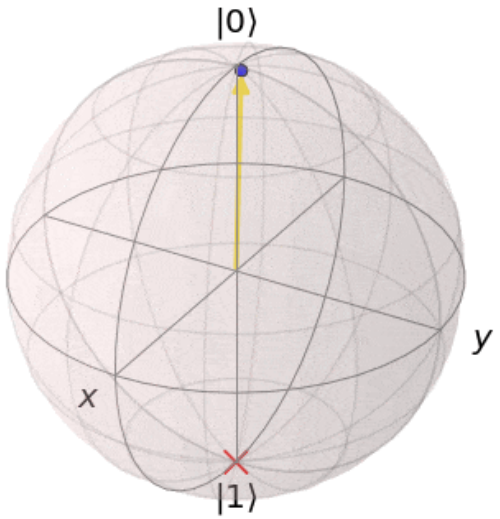
$$\partial_\alpha G = G(\alpha + \frac{\pi}{2}, \beta, \gamma)$$

"Classical" updates to the quantum system

```python
import pennylane as qml
import torch
from torch.autograd import Variable

qpu = qml.device('forest.qpu', device='Aspen-1-2Q-B')

@qml.qnode(dev, interface='torch')
def circuit(phi, theta):
    # Quantum node running on the Rigetti QPU
    qml.RX(theta, wires=0)
    qml.RZ(phi, wires=0)
    return qml.expval.PauliZ(0)

def cost(phi, theta, step):
    # Classical node
    target = -(-1)**(step // 100)
    return torch.abs(circuit(phi, theta) - target)**2

phi = Variable(torch.tensor(1.), requires_grad=True)
theta = Variable(torch.tensor(0.05), requires_grad=True)
opt = torch.optim.Adam([phi, theta], lr = 0.1)

for i in range(400):
    opt.zero_grad()
    loss = cost(phi, theta, i)
    loss.backward()
    opt.step()
```

Variational circuits: hybrid quantum-classical technique for training!

Use QPU for "**hard to compute**" cost function



Gradients provided by rotations on input data

$$\partial_\alpha G = G(\alpha + \frac{\pi}{2}, \beta, \gamma)$$

"Classical" updates to the quantum system

```python
import pennylane as qml
import torch
from torch.autograd import Variable

qpu = qml.device('forest.qpu', device='Aspen-1-2Q-B')

@qml.qnode(dev, interface='torch')
def circuit(phi, theta):
    # Quantum node running on the Rigetti QPU
    qml.RX(theta, wires=0)
    qml.RZ(phi, wires=0)
    return qml.expval.PauliZ(0)

def cost(phi, theta, step):
    # Classical node
    target = -(-1)**(step // 100)
    return torch.abs(circuit(phi, theta) - target)**2

phi = Variable(torch.tensor(1.), requires_grad=True)
theta = Variable(torch.tensor(0.05), requires_grad=True)
opt = torch.optim.Adam([phi, theta], lr = 0.1)

for i in range(400):
    opt.zero_grad()
    loss = cost(phi, theta, i)
    loss.backward()
    opt.step()
```

A Letters Journal Exploring the Frontiers of Physics

September 2017

EPL, **119** (2017) 60002
doi: 10.1209/0295-5075/119/60002

www.epljournal.org

Implementing a distance-based classifier with a quantum interference circuit

M. Schuld[1], M. Fingerhuth[1,2] and F. Petruccione[1,3]

[1] *Quantum Research Group, School of Chemistry and Physics, University of KwaZulu-Natal Durban 4000, South Africa*
[2] *Maastricht Science Programme, University of Maastricht - 6200 MD Maastricht, The Netherlands*
[3] *National Institute for Theoretical Physics - KwaZulu-Natal, Durban 4000, South Africa*

received 28 August 2017; accepted in final form 3 November 2017
published online 1 December 2017

PACS 03.67.Ac – Quantum algorithms, protocols, and simulations
PACS 03.67.Lx – Quantum computation architectures and implementations

Abstract – Lately, much attention has been given to quantum algorithms that solve pattern recognition tasks in machine learning. Many of these quantum machine learning algorithms try to implement classical models on large-scale universal quantum computers that have access to non-trivial subroutines such as Hamiltonian simulation, amplitude amplification and phase estimation. We approach the problem from the opposite direction and analyse a distance-based classifier that is realised by a simple quantum interference circuit. After state preparation, the circuit only consists of a Hadamard gate as well as two single-qubit measurements, and computes the distance between data points in quantum parallel. We demonstrate the proof of principle using the IBM Quantum Experience and analyse the performance of the classifier with numerical simulations.