Quantum Computing

How can it save the world?

Debajyoti Bera
Department of Computer Sc. & Engg.
Center for Quantum Technologies
IIIT-Delhi

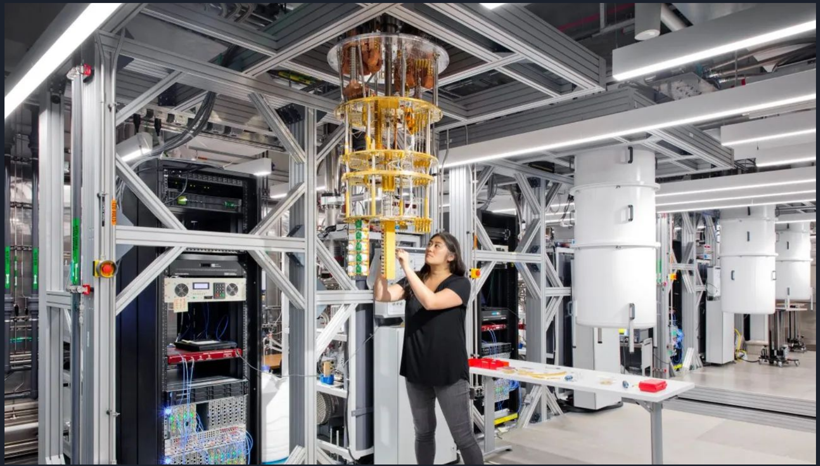
⟨C|Q|T⟩

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

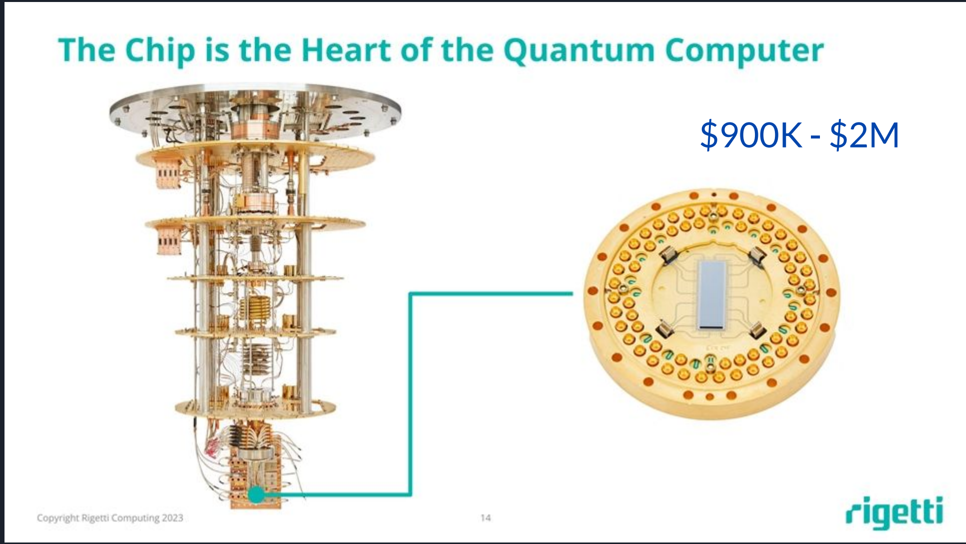
https://en.wikipedia.org/wiki/List_of_quantum_processors

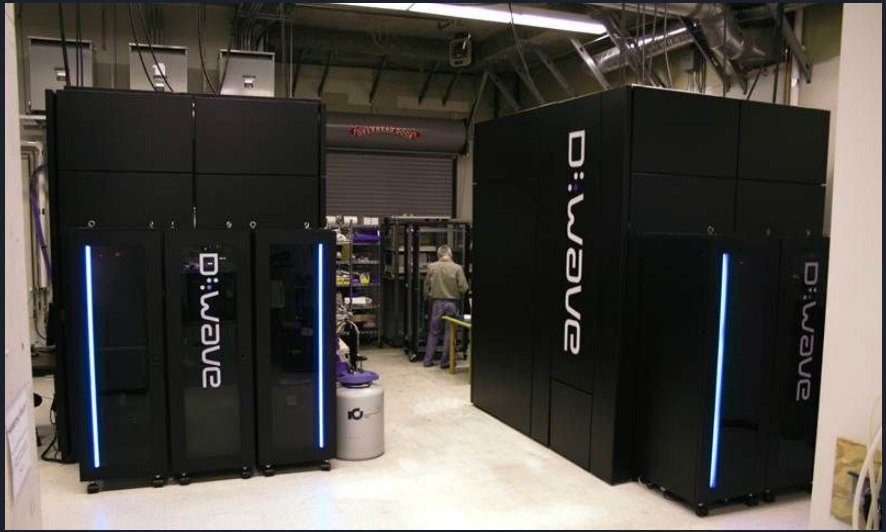
Modern QPU

Company	Technology of qubits	Architecture
IBM	Superconducting	Gate-based
Google	Superconducting	
Intel	Superconducting & semiconductor spin	
IonQ	Trapped ion	
Quantinuum	Trapped ion	
Rigetti	Superconducting transmon	
SpinQ	Nuclear magnetic resonance (NMR)	
DWAVE	Superconducting	Quantum annealing




$9K - $58K
My personal
stand-alone room
temperature 2-qubit
quantum computer


The Chip is the Heart of the Quantum Computer
$900K - $2M
Copyright Rigetti Computing 2023
14
rigetti


D:Wave
D:Wave
D:Wave

How to access QPUs ?

Amazon Braket Pricing

Hardware Provider	D- Wave	D- Wave	IonQ	Rigetti
QPU Family	2000Q	Advantage	IonQ device	Aspen-8
Per-task price	$0.30000	$0.30000	$0.30000	$0.30000
Per-shot Price	$0.00019	$0.00019	$0.01000	$0.00035

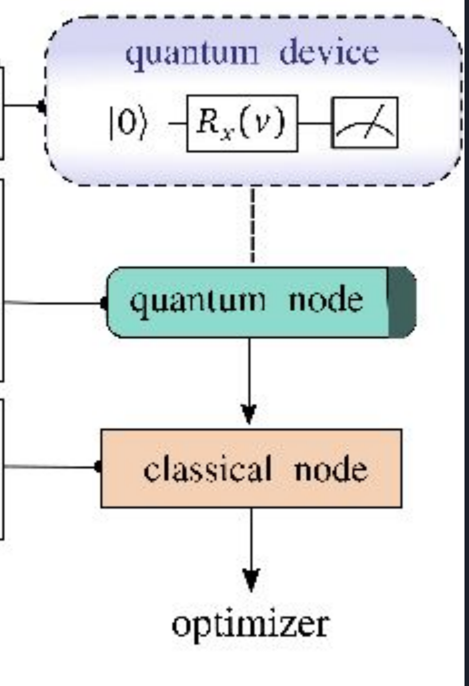
WORKFALL

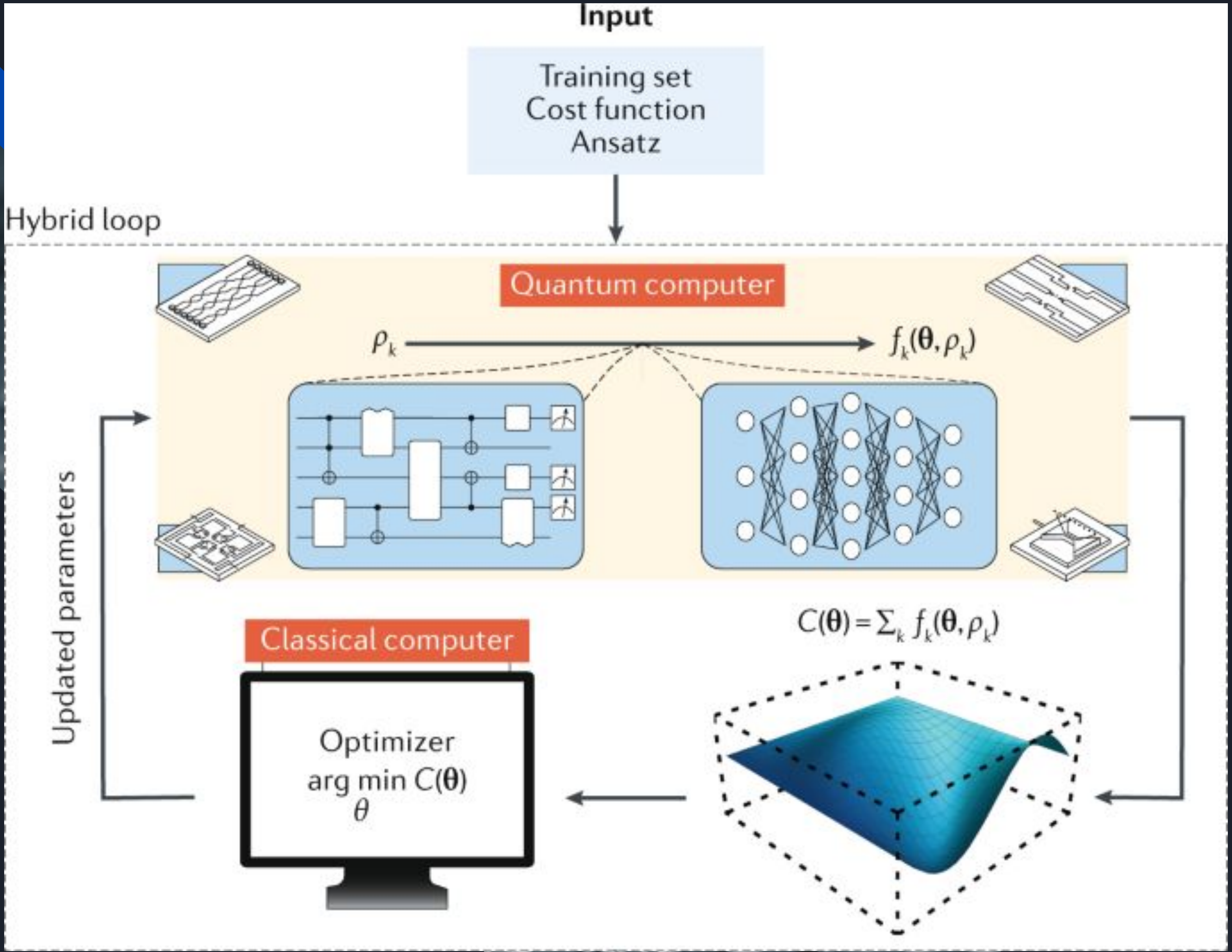
PENNYLANE

Pennylane supports IBM Quantum, Amazon Braket, IonQ, AQT, Rigetti, QuTech, Honeywell.

How to program QPUs ?

```python
import pennylane as qml

# Create backend to run the quantum node
dev = qml.device('default.qubit', wires=1)

# Define quantum node
@qml.qnode(dev)
def circuit(var):
    qml.RX(var, wires=0)
    return qml.expval.PauliZ(0)

# Define classical node/cost
def cost(var):
    return var[0] * circuit(var[1])

# Update initial variable
opt = qml.GradientDescentOptimizer()
var_new = opt.step(cost, var=[0.1, 0.2])
```



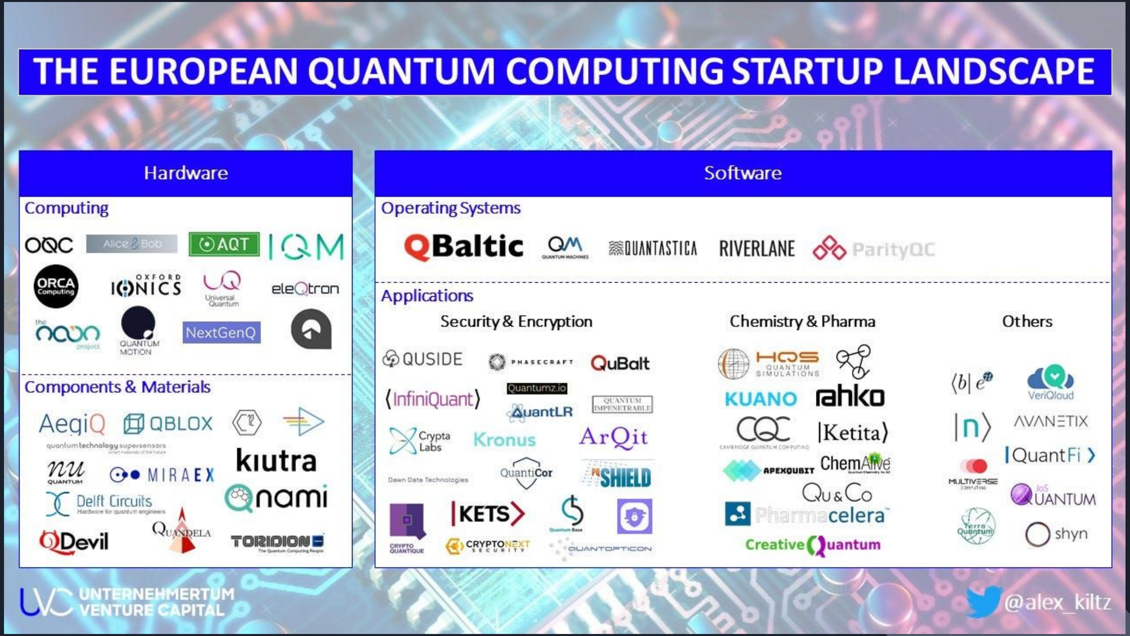


Variational Quantum Algorithms

Quantum development ecosystem

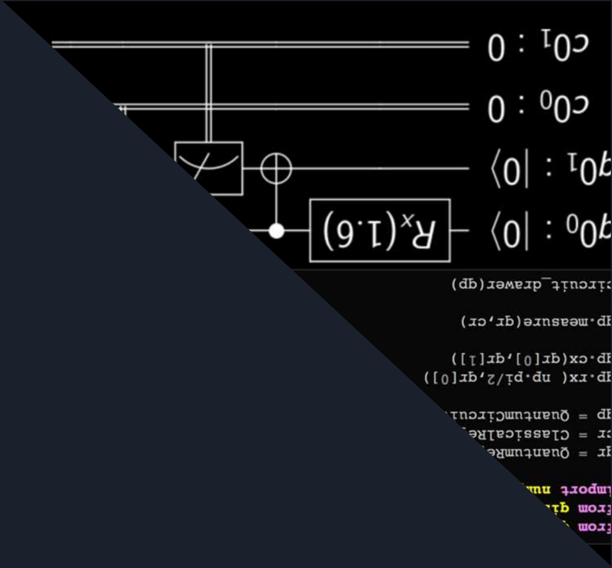
Many programming languages.
Lot of libraries.
Active community.
Several tutorial, blogs, open Github projects,

SDK
Qiskit by IBM
Ocean by D-Wave
Strawberry Fields & Pennylane
Forest by Rigetti
Circ by Google
QDK by Microsoft
CUDA-Quantum by NVIDIA



Quantum Algorithms for …

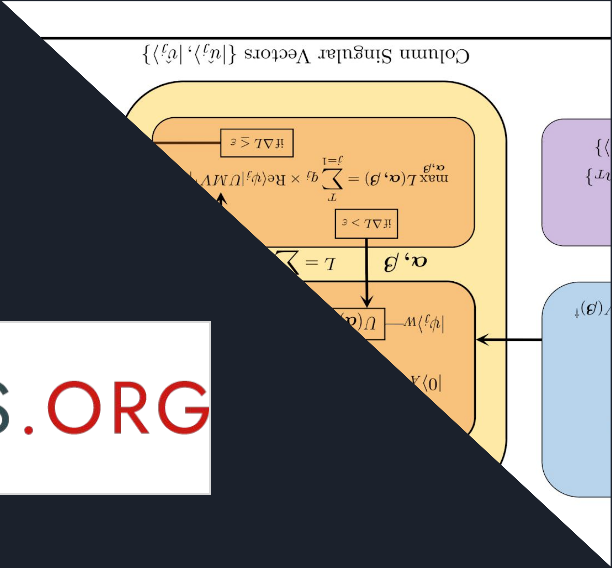
Traditional CS problems



Quantum Algorithm Zoo

Number theoretic algorithms
Linear algebraic algorithms
Algebraic algorithms
Sorting, searching, …
Graph algorithms
String algorithms
Group-theoretic algorithms
Linear programs
Semi-definite programs
…

https://quantumalgorithmzoo.org/



Quantum algorithms for …

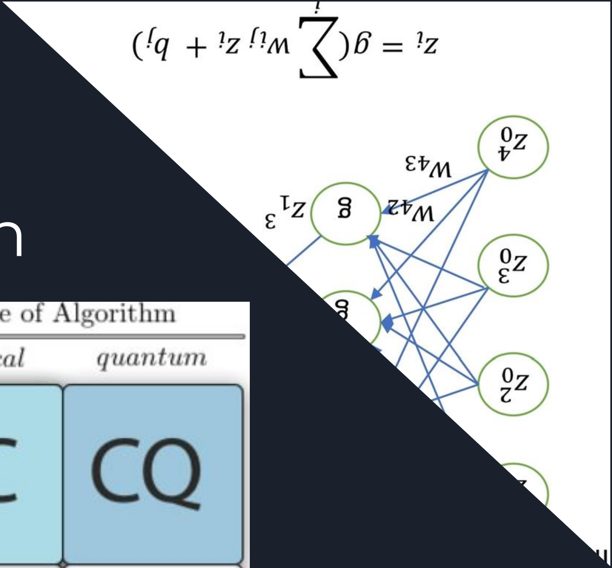
Data analysis

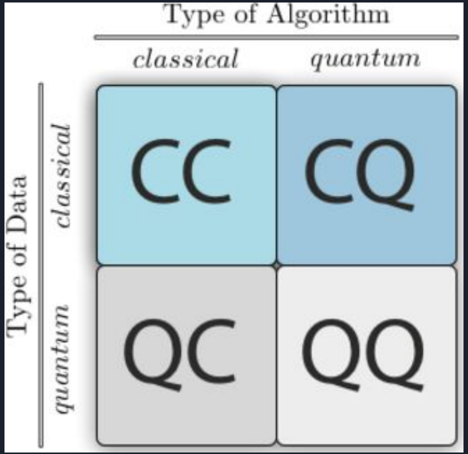
Singular-value estimation based algorithms
Monte-carlo techniques
PCA and other dimensionality reduction methods
Clustering algorithms like k-means, k-median, …
Matrix operations like inversion, solving linear system, …

Quantum algorithms for …

Machine learning & optimization

https://en.wikipedia.org/wiki/Quantum_machine_learning





Quantum-enhanced reinforcement learning
Quantum annealing
Training Boltzmann machines
Quantum convolutional neural networks (QCNN)
Dissipative QNN
Quantum generative adversarial networks (QGAN)
Hidden quantum markov models
Quantum graph neural networks
Quantum physics-inspired neural networks (QPINN)

Quantum Online Portfolio Optimization

Input: n, s, η, T, C, δ

1: Initialize $U_{\rho^{(0)}} = \mathbb{I}$, $\tilde{I} = 1$ and $q^{(1)} = (1, \cdots, 1)$, $U_{q^{(1)}} = \mathbb{I}$.

2: **for** $t = 1$ to T **do**

3: $q_{\max} \leftarrow$ Find the largest element of $q^{(t)}$ using $U_{q^{(t)}}$ and quantum maximum finding [56] with success probability $1 - \frac{\delta}{4T}$.

4: $\tilde{Z}^{(t)} \leftarrow$ Estimate the norm of $\frac{q^{(t)}}{q_{\max}}$ using $U_{q^{(t)}}$, $q_{\max}$, Lemma 2, and Lemma 3(i), with relative error $\epsilon_Z = \frac{\eta^2}{r_{\min}^2}$ and success probability $1 - \frac{\delta}{4T}$.

5: $U_{w^{(t)}} \leftarrow$ Prepare quantum circuit for approximating $\left|\tilde{w}^{(t)}\right\rangle$ of the quantum state $\left|w^{(t)}\right\rangle$, where $w^{(t)} = \frac{q^{(t)}}{\|q^{(t)}\|_1}$, using $U_{q^{(t)}}$, $q_{\max}$, $\tilde{Z}^{(t)}$, Lemma 2, Lemma 3(ii), with success probability $1 - \frac{\delta}{4T}$.

6: $\Gamma, W, V \leftarrow$ Determine using Lem. 5 applied to $\tilde{w}^{(t)}$ with probability $1 - \frac{\delta}{4T}$.

7: $i_1^{(t)}, \cdots, i_s^{(t)}$ Perform multi-sampling using Γ, W, V and Lemma 6 with probability $1 - \frac{\delta}{4T}$.

8: Invest the amount $1/s$ in each asset $i_1^{(t)}, \cdots, i_s^{(t)}$ at cost C each.

9: Wait until end of day.

10: Receive price relative oracle $U_{\rho^{(t)}}$.

11: $\rho_{j^{(t)}}^{(t)} \leftarrow$ Query $U_{\rho^{(t)}}$ with $\left|j^{(t)}\right\rangle|0\rangle$.

12: $\tilde{I}^{(t)} \leftarrow$ Estimate $\tilde{w}^{(t)} \cdot \rho^{(t)}$ using $U_{w^{(t)}}$, $U_{\rho^{(t)}}$, and Lemma 4, with relative err success probability $1 - \frac{\delta}{4T}$.

13: $U_{q^{t+1}} \leftarrow$ Prepare quantum circuit to compute $q^{(t+1)} = \exp\left(\eta \sum_{t'=1}^{t} \frac{\rho^{(t')}}{\tilde{I}^{(t')}}\right)$ usi Lemma 2.

14: **end for**

Output: $LS_{\text{samp}}^{Q} := \frac{1}{T} \sum_{t=1}^{T} \log\left(\frac{1}{s} \sum_{\ell=1}^{s} \rho_{i_\ell^{(t)}}^{(t)}\right)$.

Name	Alg.	Regret	Run time
Online	1	$\frac{1}{r_{\min}}\sqrt{\frac{\log n}{2T}}$	$O(Tn)$
Sampling-based Online	2	$\frac{2}{r_{\min}}\sqrt{\frac{\log n}{2T}}$	$O(T^2 n \log \frac{T}{\delta})$
Approximate Sampling-based Online	3	$\frac{8}{r_{\min}}\sqrt{\frac{\log n}{2T}}$	$O\left(Tn + \frac{T^2}{r_{\min}} \log \frac{T}{\delta}\right)$
Quantum Online	4	$\frac{12}{r_{\min}}\sqrt{\frac{\log n}{2T}}$	$\tilde{O}\left(T^3 \sqrt{\frac{n}{r_{\min}}} \log^{1.5}\left(\frac{1}{\delta}\right)\right)$

Why quantum ?

Fundamentally different point of view! Backed by quantum mechanics.

10 qubits represent a distribution over 1024 elements ≡ 1024-sized "stochastic" vector

+

Distribution allows "negative" probabilities ⇒ Cancellation of probabilistic scenarios

+

Certain impossibility results advocated by quantum mechanics

Classically impossible tasks are now possible.

Better speedup compared to classical techniques.

Better quality of optimization solutions.

Why not (yet) quantum ?

No Quantum RAM (yet)!

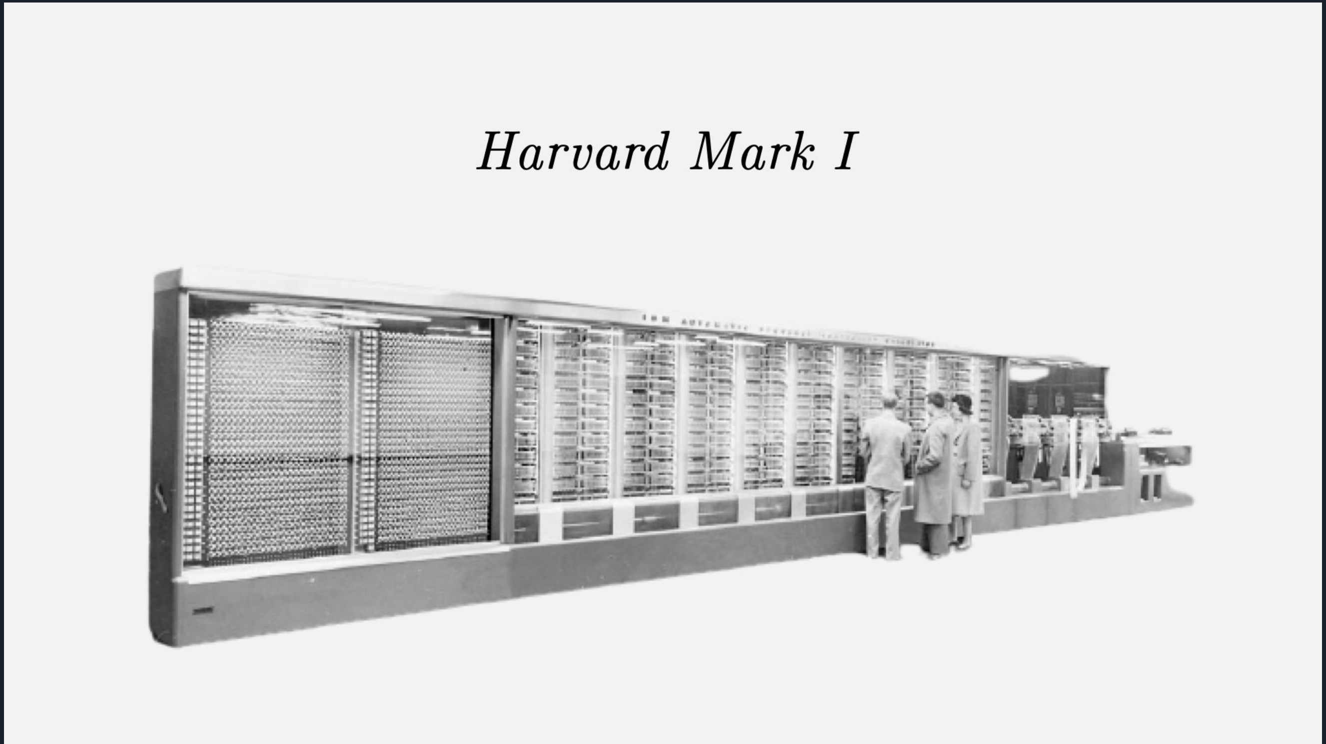
Lots of errors - T1, T2, Gate, Readout, …

Barren plateau observed in variational QA

No clear case of quantum supremacy!

Perceived learning curve.

First generation CPU



Specification

- + : 0.3 sec
- x : 6 secs
- ÷ : 15 secs
- sin, cos, etc. : 1 min +

- Operated on 72 73-bit registers

- 51 ‘ x 8 ‘ x 2 ‘
- 4.3 ton

Thank you!

https://braqiiit.github.io/