TATA CONSULTANCY SERVICES
Experience certainty.

TATA

Quantum-Enhanced Optimization and Machine Learning

Few Points, Pointers and a Detailed Image Processing Use Case

M Girish Chandra and Sayantan Pramanik
TCS Research, India

ACM Winter School on Quantum Computing at IITM, 14th Jan 2022, Sponsored by TCS Research

Acknowledgements

Many Courses and Lectures influenced us in our journey of learning from last two years. Our deepest gratitude to the intellectuals involved. Too many to list, but cannot miss the Gem Course, “Quantum Machine Learning” by Dr. Peter Wittek, 2019.

Thanks to many interns, who worked with us on Quantum so far- We grew together!

Sincere thanks to Prof. Steven Garvin, Yale University and Prof. Román Orús, Donostia International Physics Center (DIPC), Spain, for generously sharing their slides- MGC

TCS RESEARCH & INNOVATION

Few Points on Quantum Computing

Quantum Computing

Quantum Computing: a completely new way to store and process information based on the principles of Quantum Theory.

Quantum theory (used synonymously with the term quantum mechanics) is “first and foremost a calculus for computing the probabilities of outcomes of measurements made on physical systems” called quantum systems[+].

A quantum system is *typically* a collection of microscopic physical objects for which classical Newtonian mechanics does not explain experimental observations. Examples are a hydrogen atom, light of very low intensity, and a small number of electrons in a magnetic field[+].

Information is physical:
Quantum information is stored in the *physical states* of a quantum system

- atoms, molecules, ions,
- photons,
- superconducting circuits (behave as macroscopic artificial atoms),
- mechanical oscillators, …

+ Maria Schuld and Francesco Petruccione, “Machine Learning with Quantum Computers”, Second Edition, Springer, 2021

Quantum Computing (QC): Information Processing is Physical

TCS RESEARCH & INNOVATION

Quantum Computing: Popular Quantum Computing “models” use quantum bits or **qubits.**

Four Different Models of Quantum Computing+

1. Circuit model of qubits and gates 2. Adiabatic Computing (Quantum Annealing, which can be understood as a heuristic to adiabatic quantum computing) 3. One-way or measurement-based quantum computing 4. Continuous-Variable (CV) quantum computing

-quantum annealing and one-way quantum computing still refer to qubits as the basic computational unit, CV systems encode information not in a discrete quantum system such as a qubit, but in a quantum state with a continuous basis. The Hilbert space of such a system is infinite-dimensional, which can potentially be leveraged to build machine learning models (Get initiated through the available Pennylane Demos/Tutorials on CV-QC)

Qu*bits*- A qubit is realised as a quantum mechanical two-level system and as such can be measured in two states, called the basis states. Traditionally, they are denoted by the Dirac vectors $|0\rangle$ and $|1\rangle$

A quantum computer can be understood as a physical implementation of *n* qubits (or other basic quantum systems) with precise control on the evolution of the state. A quantum algorithm is a *controlled manipulation* of the quantum system with a subsequent measurement to retrieve information from the system. In this sense, a quantum computer can be understood as a special kind of sampling device: We choose certain experimental configurations—such as the strength of a laser beam, or a magnetic field—and read out samples from a distribution defined by the quantum state and the *observable*+.

+ Maria Schuld and Francesco Petruccione, “Machine Learning with Quantum Computers”, Second Edition, Springer, 2021

Gate Model Quantum Computing

The time evolution of an isolated or closed quantum mechanical system is described by the Schrödinger equation, which involves a special observable known as the Hamiltonian of the system.

A time-independent Hamiltonian *H* generates a Unitary Operator $U(t) = e^{-i\frac{Ht}{\hbar}}$ $\quad |\psi(t)\rangle = U(t)|\psi(0)\rangle$

The time evolution of a quantum system in a mixed state described by density operator evolves according to von Neumann equation, and to evolve such a density operator we have to apply the unitary from the left, and its adjoint (complex conjugate) from the right.

In Gate Circuit Quantum Computing, we can work with finite-dimensional Hilbert Spaces, and hence Operators are (complex) matrices. Recollect Unitary refers to:

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

where $U^\dagger$ is the conjugate transpose of U.

In the gneral case, *U* is a bounded operator and *U* dagger is the adjoint and *I* the identity operators, respectively.

In reality, trying to write down a Hamiltonian that does a complex calculation for a large number of qubits would be a nightmare. To simplify the matter, we'll break up the time evolution of the qubit into bite-sized **gates** that we can reliably execute. Since, at the end of the day, these gates represent evolution of the qubit over small chunks of time, we write them down as unitary matrices. We'll leave the fine details of the Hamiltonian required to execute these gates to the experimentalists building our quantum computer: we'll assume for now that we can apply a gate described by the (unitary) matrix $\mathbf{U}$ to our qubit, which will evolve the state vector of the qubit from $|\psi\rangle \rightarrow \mathbf{U}|\psi\rangle$.

In quantum mechanics, the Hamiltonian of a system is an operator corresponding to the total **energy** of that system, including both kinetic energy and potential energy.

Gate Model Quantum Computing, Contd..

+Any quantum evolution can be approximated by a sequence of only a handful of elementary manipulations, called quantum gates,which only act on one or two qubits at a time. Based on this insight, quantum algorithms are widely formulated as quantum circuits of these elementary gates. A **universal** quantum computer consequently only has to know how to perform a small set of operations on qubits, just like classical computers are built on a limited number of logic gates. Runtime considerations usually count the number of quantum gates it takes to implement the entire quantum algorithm. Efficient quantum algorithms are based on evolutions whose decomposition into a circuit of gates grows at most polynomially with the input size of the problem.

Some Remarks:

All Unitary operations (of state manipulations) have an inverse- **reversible computing**.
Each such (Unitary) matrix is a possible quantum **gate** in a quantum **circuit**
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)

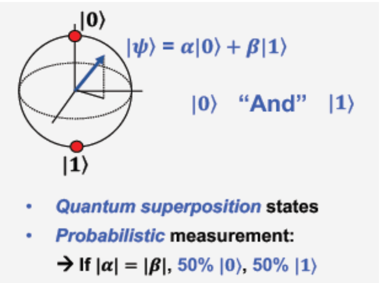
Quantum computation is completely reversible (unitary): if a quantum computer consumes any energy, the computation fails!
(Thermodynamics/information theory guarantees that the support structure, initialization, error correction and readout consumes energy)
Due to the nature of quantum physics, the destruction of information in a gate will cause heat to be evolved which can destroy the superposition of qubits.

+ Maria Schuld and Francesco Petruccione, "Machine Learning with Quantum Computers", Second Edition, Springer, 2021

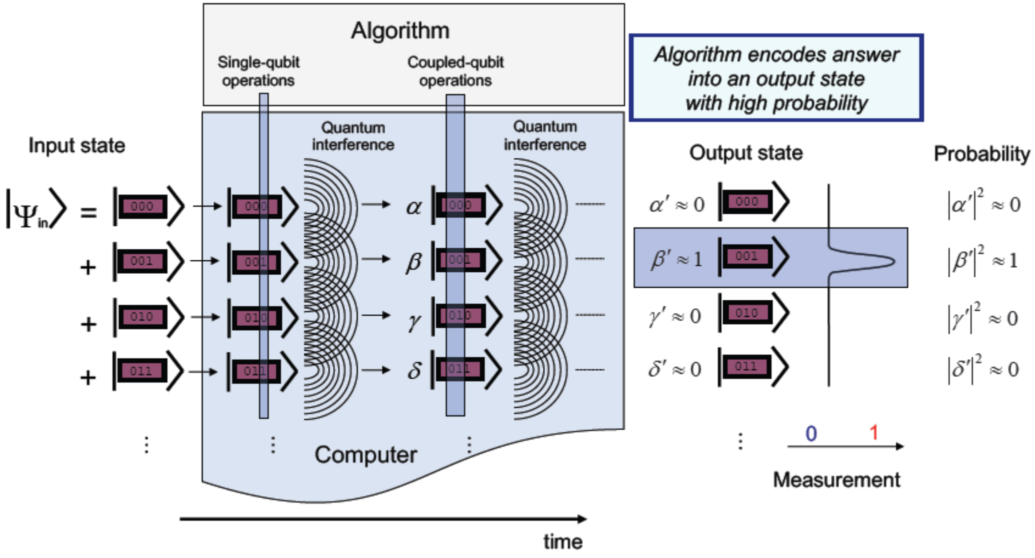
A Quick Feel for Quantum Algorithm (Universal - Gate Circuit Model of Computation)

Quick Recap

Quantum computers use **qubits**, which can be in **superposition** states and are **entangled** with each other.
Qubits- two distinguishable (coherent) states 0 and 1; also can have aspects of both 0 and 1 simultaneously -superposition; **uncertainty is a feature!**; can be anywhere on the Bloch sphere.



Quantum states- half angle bracket- ket notation; **braket** - Dirac notation
Measurement of the state of a qubit yields one classical bit of information: Measurement result is always 0 or 1.



Source: From the slides on "Introduction to Quantum Computing" by Prof. William D. Oliver, Dec 2019, Q2B

How do we reconcile discrete (binary) measurement results and continuous (analog) quantum states (encodings)? Randomness is the only solution!- From Prof. Steven Girvin's slides.

How a Quantum Algorithm Unfolds or Work on a Quantum Computer?

In many ways, it's just like a classical computer- in the sense that classical computers work on bits and gates- 1 bit gates and 2 bit gates. With a handful of those, NOT gate, AND gate, etc., we can perform any boolean logic or we can perform any classical algorithm we wish; and similarly, quantum computer has a handful of one qubit and two qubit gates, and with them we can carry out any quantum logic.

It starts with a massive superposition state, which carries aspects of all 2^n states; Put this into a computer and run an algorithm; the type of things typically done is to apply a single qubit gate operation; even one qubit operation affects all aspects of the state simultaneously (why?); that followed by quantum interference that change the weighting coefficients in front of these various aspects; next, there is other type of gates called 2-qubit gates, where depending on the state of qubit1, an operation is performed on qubit 2, that followed by again a quantum interference which modifies these coefficients further.

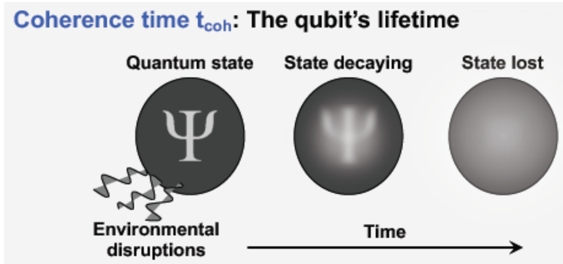
The goal of an algorithm designer is by the end of the algorithm, all of the weighting is in one of the aspects- in this example, 001 (and $\beta' = 1$). Why?

"When we make mesurements, we have an exponential slow down. We being classical beings, we measure things in a classical sense. We can get one of these component states at the output. But, we get them at a probability which goes as the square of the magnitude of these coefficients (**Born Rule**). So, with probability 1 we get the right answer. That's the goal of an algorithm designer"

Source: From the video of "Introduction to Quantum Computing" by Prof. William D. Oliver, Dec 2019, Q2B

Some Technological Problems in Quantum Computing and NISQ

Decoherence: Qubits are unstable
⇒ State of a qubit decays over time (often fast!)
⟶ Implementations of qubits even results in disturbances
⇒ Increasing number of qubits is difficult
Gate Fidelity: individual operations are (a bit) imprecise
⇒ Error of an algorithm increases with number of operations
⇒ Only algorithms with "a few" operations can be correctly performed
Readout Error: Measuring a qubit is imprecise
⇒ Results are distorted
Qubit Connectivity: Not all qubits have a physical connection
⇒ 2-qubit operations cannot be performed on arbitrary pairs of qubits
⟶ Reminder: 2-qubit operations are key for universal sets of operations
⇒ Additional SWAP operations needed
⇒ Number of operations to implement an algorithm increases



Gate time t_{gate}: Time required for a single gate operation

A Less Rigorous Figure of Merit:

# of gates per coherence time = t_{coh}/t_{gate}

Rigorous metric: gate & readout fidelity

Long coherence times are not sufficient, it's the number of gates before an error.

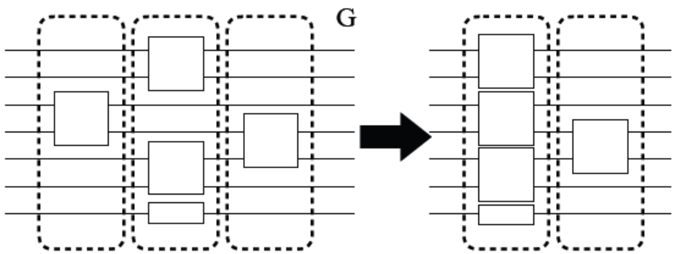
…attempt to do meaningful quantum computing in such a noisy situation!
NISQ (Noisy Intermediate-Scale Quantum) Technology uses these…
• 50 to100+ Qubits
• with a limited number of operations in algorithms
…to provide significant proof of quantum supremacy

Sources: 1. The slides on "Quantum Computing: A Brief Introduction " by Prof. Frank Leymann, SummerSoC 2019
2. Slides of "Introduction to Quantum Computing" by Prof. William D. Oliver, Dec 2019, Q2B

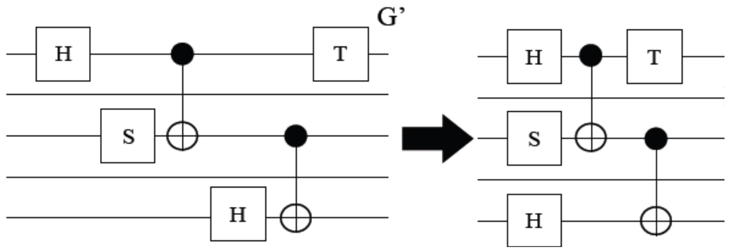
Algorithms on Noisy Devices and Depth of an Algorithm (Quantum Circuit)

TCS RESEARCH & INNOVATION

The *depth* (*D*) of a quantum circuit is the number of layers of 1- or 2-qubit gates that operate *in parallel on disjoint qubits*.
The breadth or *width* (*W*) of a quantum circuit is the number of manipulated qubits.

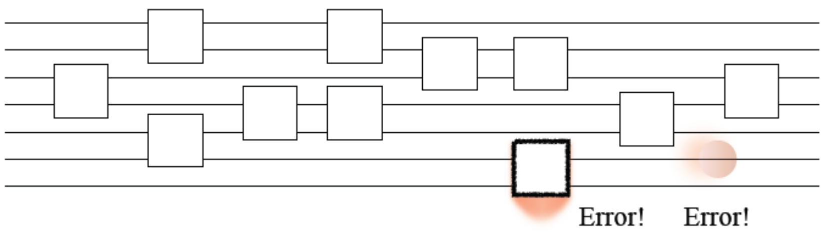


D = 2 and W = 7



D = 3 and W = 3

Noisy Algorithm:



Rough estimation of the "size" of a quantum algorithm that can be performed without errors:

$$W\,D << \frac{1}{\varepsilon} \qquad \varepsilon : \text{error rate}$$

Consequence:

Deep quantum algorithms ⇒ few qubits
⇒ efficient classical simulation possible

Shallow quantum algorithms ⇒ many qubits
⇒ potential for quantum advantage

Source: From the slides "Quantum Computing: A Brief Introduction " by Prof. Frank Leymann, SummerSoC 2019

Adiabatic Quantum Computing and Quantum Annealer

TCS RESEARCH & INNOVATION

Adiabatic quantum computing is in a sense the analog version of quantum computing in which the solution of a computational problem is encoded in the ground state (i.e., lowest energy state) of a Hamiltonian which defines the dynamics of a system of *n* qubits. Starting with a quantum system in the ground state of another Hamiltonian which is relatively simple to realise in a given experimental setup, and slowly adjusting the system so that it is governed by the desired Hamiltonian ensures that the system is afterwards found in the ground state[+].

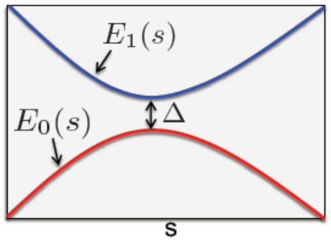
A Nice Summary of Quantum Annealer

"Non-ideal" implementation of adiabatic quantum computation

$$H(s(t)) = s(t)H_0 + (1 - s(t))H_P$$

$$s(0) = 1; s(T) = 0$$

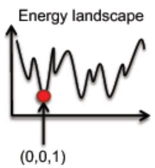
Usually: $s(t) = \left(1 - \frac{t}{T}\right) \quad T = O\left(\frac{1}{\Delta^2}\right)$



$$H_0 = \sum_i \sigma_i^x \qquad H_P = \sum_i h_i \sigma_i^z + \sum_{i,j} J_{ij} \sigma_i^z \sigma_j^z$$

(not quantum universal!)

How to move towards Universal Adiabatic Quantum Computation?



Run the process, sample the outcome; repeat and choose the best found solution (non-ideal conditions imply solution may not be optimal)

In **optimization problems**, it is routine to construct a cost or "energy function" whose global minimum value represents an optimal solution to the problem. In Quantum Mechanical Formulation we have Hamiltonian (Operator).
Notice the Problem or Cost Hamiltonian H_P and the initial "mixing" Hamiltonian H_0.
Also, the Hamiltonian here is an **Ising Model**- which means we can formulate it in terms of interacting "spins".
Then, we are seeking the "ground" state of the quantum system governed by that Hamiltonian. Ground state is also the preferred state of a physical quantum computing system.

Source: Slides of the talk, "Quantum Computing for Finance" by Román Orús, Sept. 2020, for Centre for Quantum Technologies, Singapore

+ Maria Schuld and Francesco Petruccione, "Machine Learning with Quantum Computers", Second Edition, Springer, 2021

Our Research, QUBO and Other Related Points

Our Research "New / Enhanced Quantum Algorithms for NISQ & Beyond+"
-New algorithms make the biggest computational leaps
-Competitive advantage come from the inhouse algorithms and IPs
Quantum-Enhanced Optimization, Machine Learning, Sampling/Stochastic Modeling
There is a research activity on Quantum Chemistry as well.
We are considering **both** Gate-Model Based Computing and D-Wave Annealer.
We are interested in *Quantum-Inspired Algorithms* also- new classical algorithms.
Now- From Initial Explorations to Use Cases Consolidation and Proof of Concepts (PoCs)
At TCS- we have highly diverse Business Units and Research Groups
Use Cases- Banking and Finace, Life Sciences, Materials Design, Travel and Hospitality related problems, Retail, Telecom, Manufacturing, Energy, ...
Already have strong presence in cutting-edge end-to-end "classical" solutions
Exclusive Data Sets are available

See also our QIQT, July 2021 video
https://www.youtube.com/watch?v=ABGHFaGDeuE

Did you carefully observe the previous slide? - we were considering binary optimization, where the values of (binary) variables were sought after. Further, the Ising Hamiltonian was having quadratic (product) terms.

Enter Quadratic Unconstrained Binary Optimization (QUBO)
QUBO model encompasses many **important** optimization problems; about 20+ are listed in:
G.Kochenberger and F. Glover (2006) "A Unified Framework for Modeling and Solving Combinatorial Optimization Problems: A Tutorial," In: Multiscale Optimization Methods and Applications, eds. W. Hager, S-J Huang, P. Pardalos, and O. Prokopyev, Springer, pp. 101-124. **Check it**

- They are hard problems (There is no (classical) polynomial-time approximation scheme (PTAS) which gets arbitrarily close to the solution (unless P = NP)
- **Good QUBO Solvers and Fast Architectures needed**

QUBO involves finding the binary vector ***x*** such that the function $\sum_{i<j} x_i Q_{ij} x_j + \sum_i Q_{ii} x_i$ is minimized; x_i and x_j are binary decision variables; maximization, if required can be handled trivially.

+Beyond- Future fault-tolerant quantum computing; FT QC uses error correction

Optimization, D-Wave Annealer

TCS RESEARCH & INNOVATION

Any general QUBO cost function $C(x)$ can be translated to Hamiltonian Operator such that

$$H_C|x\rangle = C(x)|x\rangle$$

The bit string *x* can be identfied with the computational basis state of our quantum system (e.g., D-Wave Annealer).

What do we gain?
With the “required” quantum Ising Hamiltonian acting on our system, we can seek its ground state through either Quantum Annealer (like, D-Wave Machine) or on a Gate-Model Quantum Computing through Quantum Approximate Optimization Algorithm (QAOA).
Because of this- we often see the remark that a Quantum Computers are *better matched* processors for solving QUBO problems.

Our QUBO problem of interest may lead to a cost Hamiltonian, for example (illustration only):

$$\begin{array}{c c} & \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix} \end{array}$$

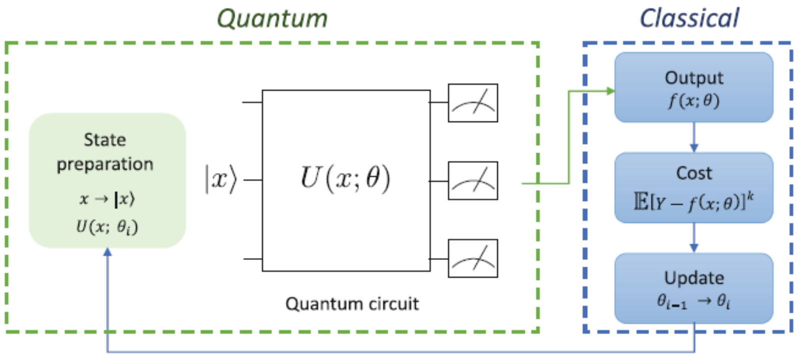
This two qubit system settles for the state $|10\rangle$, why?
Hint: eigen-based reasoning

D-Wave Annealers- special kind of Quantum Computers to solve many useful optimization (and sampling) problems. They utilize Superposition, Quantum Tunneling and Entanglement; https://www.dwavesys.com/learn/quantum-computing/

Mapping a given optimization problem to QUBO may require careful considerations.

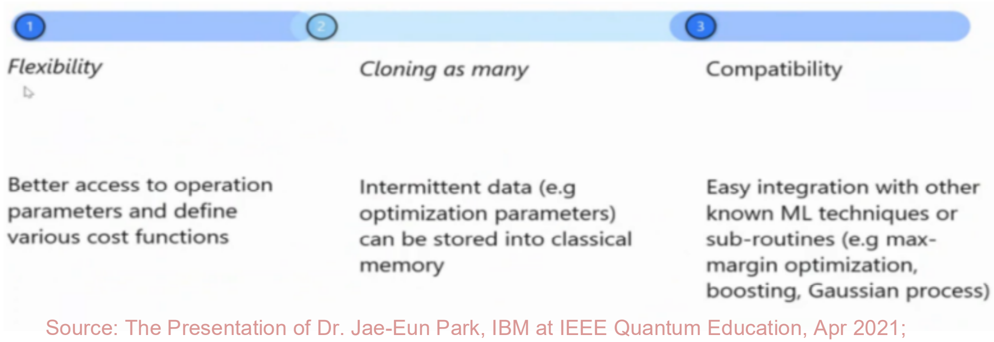
Quantum-Classical Hybrid will stay

In the NISQ era, the Hybrid Variational Approach is the norm

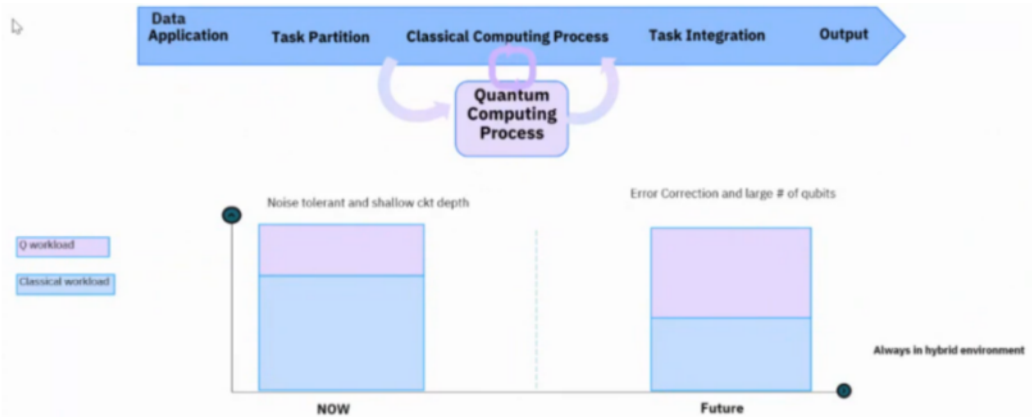


Source: Antonio Macaluso, et al, A Variational Algorithm for Quantum Neural Networks, ICCS 2020

Advantage:



Source: The Presentation of Dr. Jae-Eun Park, IBM at IEEE Quantum Education, Apr 2021;
ML- Machine Learning



Source: The Presentation of Dr. Jae-Eun Park, IBM at IEEE Quantum Education, Apr 2021

Note- Even though the figures are biased towards ML, the arguments are applicable for other problems.
Combinatorial Optimization Problems with complex constraints have to be split into QUBO+Classical Oprtimization
- D-Wave Hybrid

Notice the terms like QPU-CPU Hybrid and Quantum Accelerator in the literature/blogs
Advantage for Solution providers like TCS

QPU- Quantum Processing Unit

QAOA and VQE

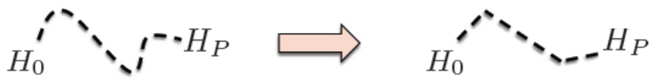
We cannot miss the two variational circuits- Variational Quantum Eigensolver and QAOA

Quant. Approx. Opt. Algorithm (QAOA)

E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

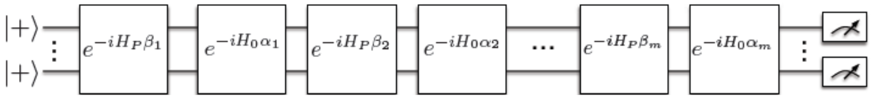
"QUA-WA"

"Break" adiabatic evolution into discrete optimized steps



$$U(t) \Rightarrow \prod_{i=1}^{m} \left(e^{-iH_0\alpha_i} e^{-iH_P\beta_i}\right)$$

Choose m, sample the energy form the outcome of the quantum circuit, use it to **optimize over alphas and betas**, and repeat until convergence (i.e., optimize the discrete path in Hamiltonian space)



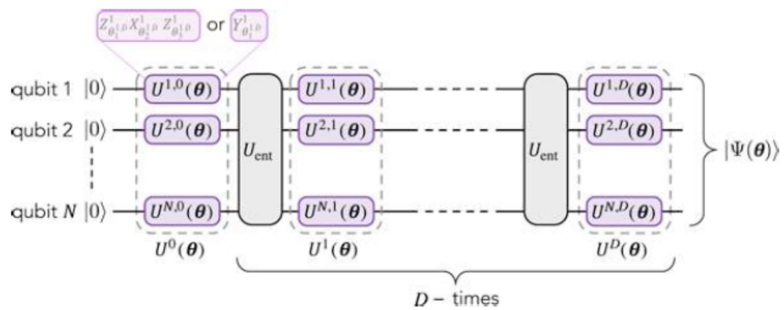
It has the correct structure of entanglement in the variational quantum circuit
The circuit can be difficult to implement (greater depth)

Source: Slides of the talk, "Quantum Computing for Finance" by Román Orús, Sept. 2020, for Centre for Quantum Technologies, Singapore

Variational Quantum Eigensolver (VQE)

A. Peruzo et at., Nat. Comm. 5 (2014).

Similar to QAOA, but this time we choose the circuit



We choose the circuit, therefore we can control very well its implementation
We choose the circuit, therefore the entanglement structureof the ansatz may not be the one of the problem

A variational principle is one that enables a problem to be solved using calculus of variations, which concerns finding such functions which optimize the values of quantities that depend upon those functions. The Variational Method is used to approximately calculate the energy levels of difficult quantum systems (ground and excited).

Quantum Machine Learning

Some of the Sources Used for the Content

For Classical Machine Learning:

1. Slides of Pradeep Raviumar and Peter Stone, The Machine Learning Summer School MLSS 2020, Max Planck Institute for Intelligent Systems, Tübingen, Germany, June-July 2020

2. Slides and Video of Michael Kagan, Introduction to Machine Learning, CERN OpenLab Summer School, July 2021

3. Slides and Video of Sofia Vallecorsa, Introduction to Deep Learning: Examples from High Energy Physics, CERN OpenLab Summer School, July 2021

For Quantum Machine Learning:

Many videos, slides, tutorials, blog articles, and the book by Maria Schuld and Francesco Petruccione (2nd Edition)

Few Points on Machine Learning (ML)

Giving computers the ability to learn without explicitly programming them (Arthur Samuel, 1959).

Machine Intelligence = Data/Distributions + Algorithms/Hardware + Models

Models learnt from data that characterize the patterns, regularities, and relationships amongst variables in the system.

Model
- Learning
- Inference: using model to infer properties of system(s)

Use data samples to construct model that minimizes cost on **unseen** data

Model Learning and Inference can be cast as an optimization problem

It's useful to note that “generalization” is the goal of ML; Optimization enters as a part

Over-parametrizing a deep model often improves test performance, contrary to bias-variance tradeoff prediction; “double-decent” risk curve. But we must control that: Gradients don’t vanish, Gradient amplitude is homogeneous across network and Gradients are under control when weights change

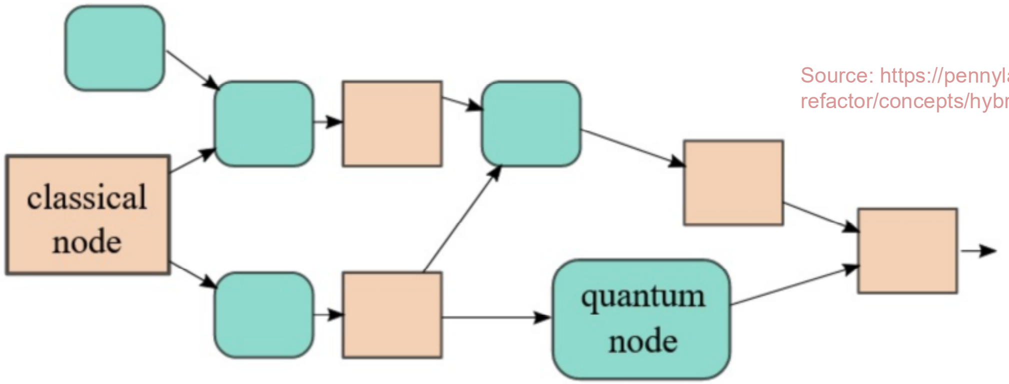
Exercise: Study the generalization aspects in the Unsupervised ML

Quantum Machine Learning

Quantum Machine Learning (QML) is at the crossroads of two of the most exciting current areas of research: Quantum Computing and classical Machine Learning. In QML, the proposition is to enhance the performance of machine learning tasks by using quantum computers, like, improving the accuracy, reducing the number of training samples, speeding up the process of training and inference, interpretability, trust, energy savings, among others.

In the present Noisy Intermediate Scale Quantum (NISQ) computing era, the hybrid quantum-classical processing has established itself an essential combination, and this “cooperation” will continue for a long time as mentioned earlier. In fact, Quantum and classical nodes can be combined into an arbitrary directed acyclic graph. You will see a hybrid quantum-classical Image Processing pipeline soon.

QML- A relatively a young field with extensive activity and “hope”.

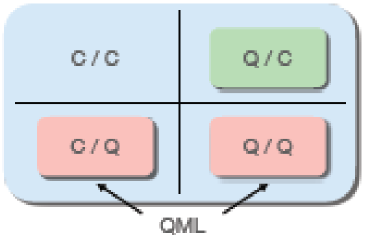


Source: https://pennylane.readthedocs.io/en/user-docs-refactor/concepts/hybrid_computation.html

Practical quantum advantage “Solve a problem that is useful either for academia or industry faster or *better* than any known classical algorithm on the best classical computer” (M. Troyer, Microsoft)

Quantum supremacy refers to quantum computers that “.. can do things that classical computers can’t, regardless of whether those tasks are useful” (John Preskill, Caltech)

QML - Few Remarks



First letter - Data (or Problem)
Second letter - Algorithm (Hardware)

Source: Patrick Huembeli, "Machine Learning for Quantum Physics and Quantum Physics for Machine Learning", PhD Thesis, ICFO, Feb. 2021

Presently QML is dominated by C/Q
In the NISQ and beyond Quantum Processing works together with the Classical Computing

Emergence of QML: When NISQ Computers bacame availbe, people were looking for applications which can work in the noisy scenarios, give access to multi-billion $ markets and attract young researchers - (Q) Machine Learning!

(From the slides of Maria Schuld, Quantum Machine Learning, SMBQ, Sept. 2020)

At present, the circuit model of qubits and gates is the more common formalism.
Of course, that doesn't mean that Annealers are not getting touched (there are different aspects of ML which can be outsourced to Quantum Computing).

- Example, using D-Wave Annealer to train Restricted Boltzmann Machines based on sampling (a decade back)

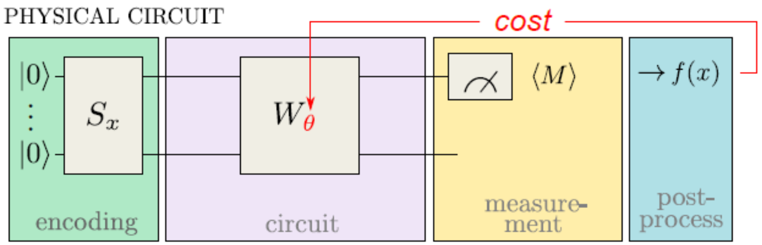
OPEN **QUBO formulations for training machine learning models** Nature Scientific Reports, May 2021

Prasanna Date[1], Davis Arthur[2] & Lauren Pusey-Nazzaro[3]

The exploroations are trying to answer ultimately - Hardware for Artificial Intelligence/ ML : CPU, GPU, QPU?

Quantum Circuit Models

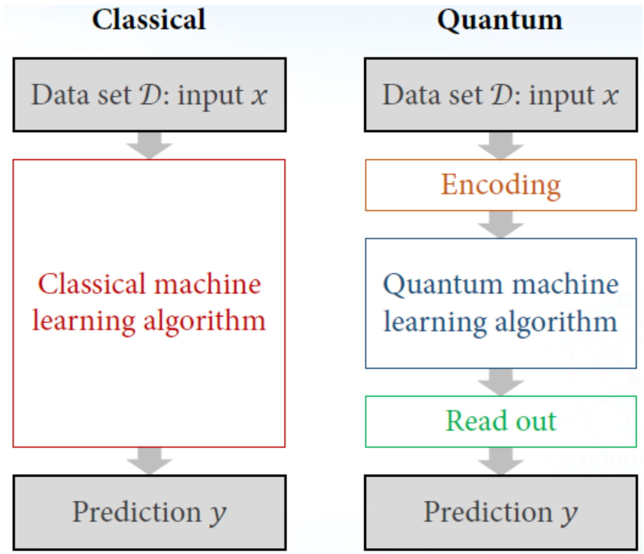
Similar to classical ML, we have to focus on three main ingredients: 1. Data 2. Model Family and 3. Loss (Cost) Function



Source: Slides of Maria Schuld, Quantum Machine Learning, SMBQ, Sept. 2020

Variational Circuit: Quantum-Classical Processing
Purple Circuit Portion: Ansatz
-Consists of layers of 1 and 2 qubit gates
- Different suggestions have emerged.
-Gradient computation of the circuit possible
- Analytical, Natural Gradients
- Parameter Shift Rule
Classical Optimization
Different “clever” loss or cost functions are necessary

Exercise- Study the loss function of Variational Least Squares Algorithm



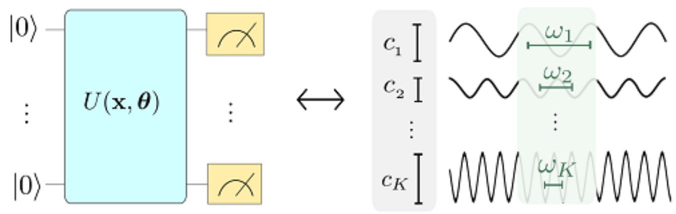
Source: Slides of Hao-Chung Cheng, A Glimpse of Quantum Machine Learning, Machine Learning Guest Lecture, National Taiwan University, May 2021

Encoding is absolutely crucial for QML
Can be learnt as well

Quantum Circuit Models: Additional Points

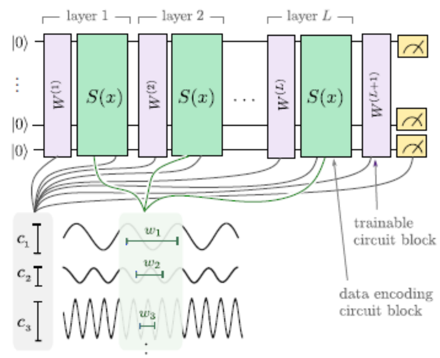
Apart from “build and run” approach (with some thoughts of course!), rigorous examination of fundamental issues like expressivity, approximation property, role of data, barren plateus, (re)formulation as kernel models, etc are getting aggressively worked out in the last couple of years.

Quantum Circuits provide Partial Fourier Series representation



$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 0|U(\boldsymbol{x},\boldsymbol{\theta})MU(\boldsymbol{x},\boldsymbol{\theta})|0\rangle = \sum_{\omega\in\Omega} c_\omega e^{i\omega x}$$

Available frequencies are determined by the encoding (the necessaity of repetition of the encoding is brought out).
Trainable unitaries and the observable fix the coefficients.
1-d data input example; higher-dimensional features simply generalize to multi-dimensional Fourier series.



Source: https://pennylane.ai/qml/demos/tutorial_expressivity_fourier_series.html

Many open problems- How to demonstrate quantum advantages (defining advantage itself- runtime speed up, sample complexity, representational power,..), to provide theoretical evidence, practical implementation versus asymptotic complexity, data input/output issues, benchmarking, etc
Very interesting research area. A lot more work is needed!
Even if QML does not help classical problems, it might help quantum problems -QQ (some initial thoughts with more hope are emerging).

Exercise- Study about representational power, expressivity; useful to read some discussions, e.g., https://stats.stackexchange.com/questions/469493/capacity-and-expressivity-of-a-neural-network

Some Useful Remarks when we look at Real-Life Use Cases

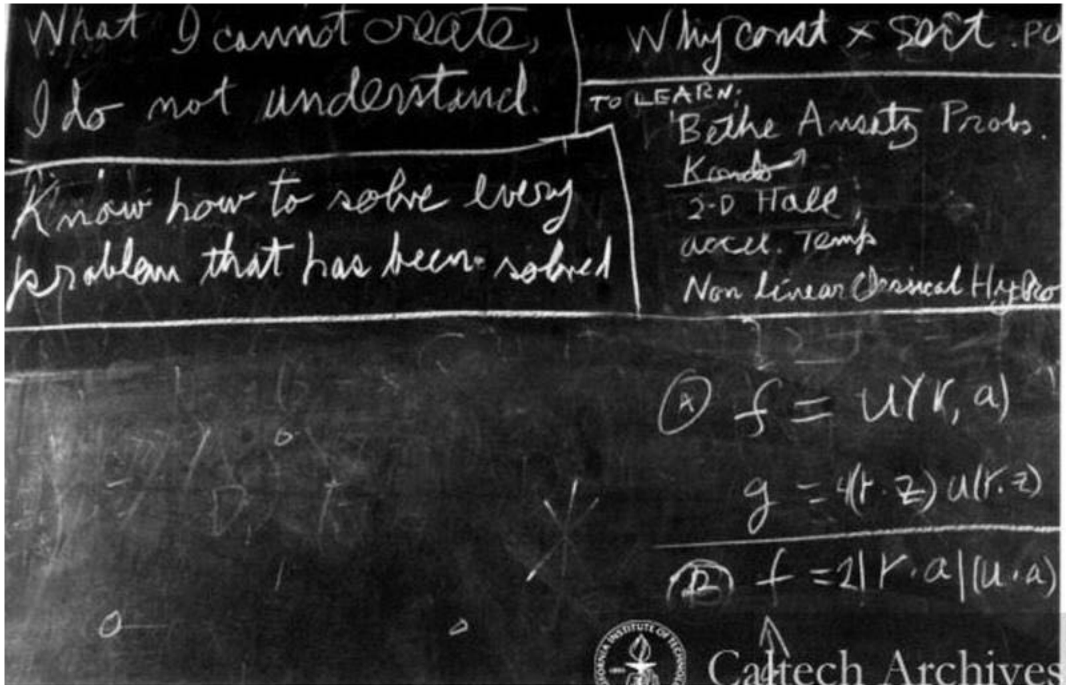
Efficiently loading classical data onto quantum computers and reading out classical outputs resulting from quantum computations is still the field of ongoing research. The majority of the quantum algorithms devised so far is based on the existence of a Quantum Random Access Memory (QRAM) for accessing the classical data. The realization of a QRAM has been theoretically proven, but concrete hardware implementations are still undergoing. Alternatively, classical data can be loaded into a quantum states via specialized circuits.

It is not always possible to apply a quantum linear-algebra algorithm out of the box to solve a specific use case; several conditions must be met and customizations are often necessary to address unique use-case-dependent requirements. **Furthermore, multiple classical and quantum algorithmic components are usually involved in the end-to-end solution of a real-life use case, with the potential for any such component to become the bottleneck and negate the overall quantum advantage. The task of computing the quantum speedup of the solution of a specific use case is, therefore, not always intuitive.**

Source: Marco Pistoia, et al, “Quantum Machine Learning for Finance”, https://arxiv.org/abs/2109.04298, Sept 2021

Time to Look into a Use Case

Over to Sayantan!



Thank You

Questions?

Emails: m.gchandra@tcs.com and sayantan.pramanik@tcs.com

Back Up Slides

Quantum Register; Quantum Parallelism

TCS RESEARCH & INNOVATION

Recollect, classical register is a series of *n* (classical) bits.
Quantum register is a series of *n* qubits;

Consequence of Superposition: A quantum register can be in a superposition state composed of 2^n "components" or aspects
$|00...00\rangle,\ |00...01\rangle,\ |00...10\rangle,...,|11...11\rangle$

2- Qubit Register; Description through tensor product

$$R = |x_1\rangle \otimes |x_0\rangle$$
$$|x_0\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle$$
$$|x_1\rangle = \beta_0|0\rangle + \beta_1|1\rangle$$

$$\begin{aligned} R &= |x_1\rangle \otimes |x_0\rangle \\ &= (\beta_0|0\rangle + \beta_1|1\rangle)\cdot(\gamma_0|0\rangle + \gamma_1|1\rangle) \\ &= \beta_0\gamma_0|0\rangle|0\rangle + \beta_0\gamma_1|0\rangle|1\rangle + \beta_1\gamma_0|1\rangle|0\rangle + \beta_1\gamma_1|1\rangle|1\rangle \end{aligned}$$

With $\alpha_{ij} = \beta_i\gamma_j$
$$\begin{aligned} R &= \alpha_{00}|0\rangle|0\rangle + \alpha_{01}|0\rangle|1\rangle + \alpha_{10}|1\rangle|0\rangle + \alpha_{11}|1\rangle|1\rangle \\ &= \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \end{aligned}$$

$|00...00\rangle,\ |00...01\rangle,\ |00...10\rangle,...,|11...11\rangle$ Each is a tensor product of *n* states

Classical register of *n* bits- 1 value at a time for compuation.
Quantum register of *n* qubits- 2^n values at a time for computation by a Quantum Computer --> massive (quantum) parallelism. Quantum computer manipulates 2^n values at the same time.
Even a small quantum computer of 53 qubits has a state space dimension so large ($d = 2^{53} = 9\text{x}10^{15}$) that its operation is very difficult to simulate on a conventional supercomputer (Peta- 10^{15})

A quantum register can hold an exponentially large superposition of all possible 2^n states that can be created in one time step.
Feature: Since this superposition includes all possibilities, the answer to our calculation is in here!

“Bug” in Quantum Parallelism; Quantum Interference

TCS RESEARCH & INNOVATION

Bug: Measurement yields an almost completely random bit string.
How can quantum algorithms produce useful results instead of random answer (noise)?

A quantum register can hold an exponentially large superposition of all possible 2^N states

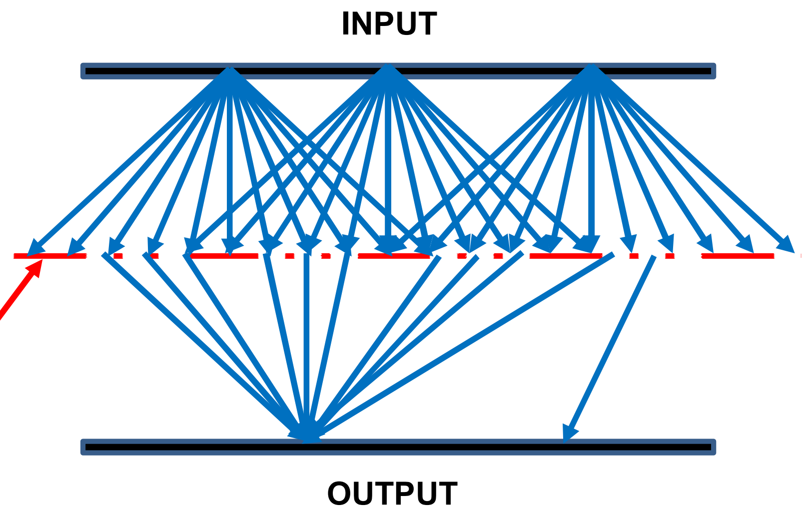
$$|000\rangle \Rightarrow |000\rangle + |001\rangle - |010\rangle - |011\rangle + |100\rangle + |101\rangle + |110\rangle - |111\rangle$$

Quantum computer program

Apply wave-like destructive interference to eliminate (many of) the ‘wrong’ answers.

$$\cancel{|000\rangle} + \cancel{|001\rangle} - |010\rangle - |011\rangle + \cancel{|100\rangle} + \cancel{|101\rangle} + |110\rangle - \cancel{|111\rangle}$$

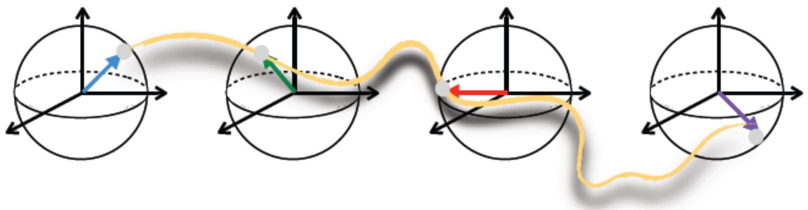
Measurement then yields the correct answer with high probability.



‘programmable diffraction grating’

Superposition, Entanglement and Interference

Quantum register is a series of *n* qubits; recollect, classical register is a series of *n* bits.
Entanglement: In a quantum register, qubits can influence each other non classically;
It can be seen as a global phenonmenon in the sense that manipulating a single qubit of a quantum register influences the state of the other qubits - Entanglement is unique to quantum computing;
Every computation *not* involving entangled qubits can be performed with the same efficiency using classical computations.



Source: From the slides on "Quantum Computing: A Brief Introduction " by Prof. Frank Leymann, SummerSoC 2019

Consequence of Superposition: A quantum register can be in a superposition state composed of 2^n "components" or aspects

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

Recollect- the tensor product and the fact that entangled states are non-separable

Also note- the "components" **interfere**; see the figure in the previous slide also; more in the next slide.
Classical register of *n* bits- 1 value at a time for compuation.
Quantum register of *n* qubits- 2^n values at a time for computation by a Quantum Computer --> massive (quantum) parallelism.
Even a small quantum computer of 53 qubits has a state space dimension so large ($d = 2^{53} = 9\text{x}10^{15}$) that its operation is very difficult to simulate on a conventional supercomputer.

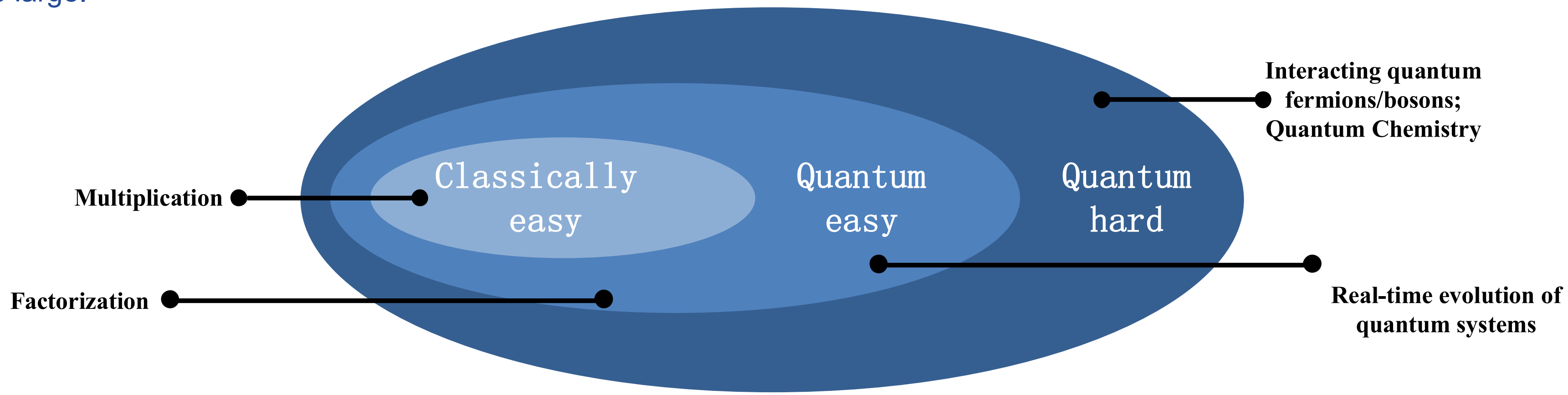
Every quantum algorithm showing "exponential speed up" compared to classical algorithms **must exploit entanglement**- Why?
Richard Jozsa and Noah Linden, "On the role of entanglement in quantum-computational speed-up", Proceedings of the Royal Society, Aug 2003

What Powers Do Quantum Machines Have?

Massive parallelism: perform certain computations that are impossible on **ANY** conventional computer. *Not because of clock speed but rather complexity of tasks scales differently with problem size N.*

$$\text{Time } T \; \alpha \; N^q \text{ vs } e^N$$

Note: the proportionality constant sometimes called prefactor or human time scale should also be considered carefully; they can be large.



Classification of problem complexity is an on-going research topic in quantum computer science;many a times focuses only on exponential gains and is better with worst-case analysis than average case or with heuristics (e.g. for optimization problems). Quantum CS heuristics will be important and must advance with the hardware.

Source: From the Slides of Prof. Steven Garvin -**The Second Quantum Revolution and the Race to Build 'Impossible' Computers**

Little More on Computational Complexity

TCS RESEARCH & INNOVATION

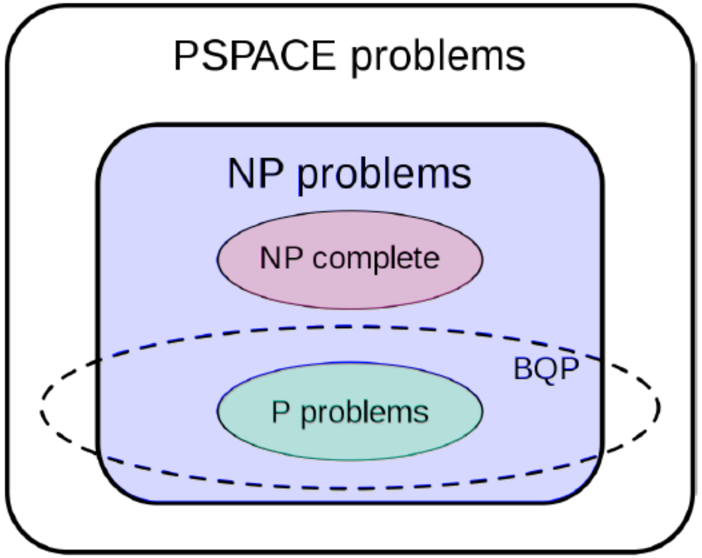


Image credits: wikipedia.org

In computational complexity theory, PSPACE is the set of all decision problems that can be solved by a Turing machine using a polynomial amount of space.

Bounded-error quantum polynomial time (BQP) is the class of decision problems solvable by a quantum computer in polynomial time, with an error probability of at most 1/3 for all instances.

A problem is *Bounded Error Quantum Polynomial time* (BQP) if it can be solved on a quantum computer with error probability $\leq \frac{1}{2} - \varepsilon$

BQP is for quantum computing what P is for classical computing!

- Let *A* be a BQP algorithm
- Let *A'* be the following algorithm:
- *A* is repeated *N* times
- The result with highest frequency will be output

The probability to output the correct result increases exponentially with the number *N* of repetitions

$$\boxed{P(\text{wrong majority}) \leq e^{-2N\varepsilon^2}}$$

(Chernoff Bound)

Also from the slides on "Quantum Computing: A Brief Introduction " by Prof. Frank Leymann, SummerSoC 2019

Success Amplification:

Let ω be the maximal probability to accept a wrong result.

After $N \geq \frac{1}{2\varepsilon^2} \ln \frac{1}{\omega}$ repetitions of a BQP algorithm the result is correct with probability $1 - \omega$

Example: $\varepsilon=1/4$, $\omega=1/1000 \Rightarrow N=56$

$\Rightarrow$ After 56 repetitions the result is correct with probability 99.99%

Noise in Quantum Computing

The huge information content of quantum superpositions comes with a price:

$$|000\rangle + |001\rangle - |010\rangle - |011\rangle + |100\rangle + |101\rangle + |110\rangle - |111\rangle$$

Great sensitivity to noise, perturbations and dissipation.

Quantum operators are typically implemented by rotation operators

E.g.: $H = i \cdot R_z(\pi) \cdot R_y\left(-\frac{\pi}{2}\right) \cdot R_z(0)$

Typically, these are rotations by non-rational angels

Such rotations can<u>not</u> be performed precisely

$\boxed{\Rightarrow \text{Quantum operators are typically } \textit{noisy} \text{ (i.e. erroneous)}}$

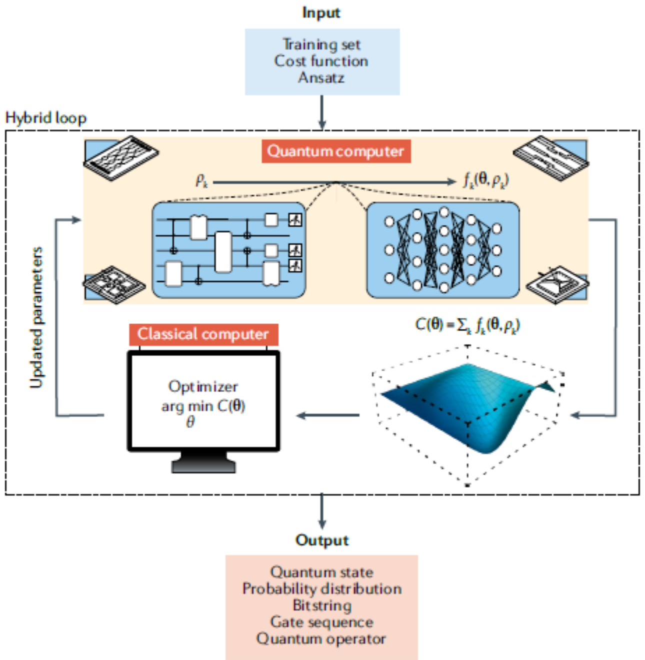
Qubits are typically interacting with their environment, i.e. they are unstable
⇒ Qubits "decay" over time (decoherence)
⇒ A quantum algorithm cannot per performed for an arbitrary long time, i.e. it cannot contain arbitrary many steps

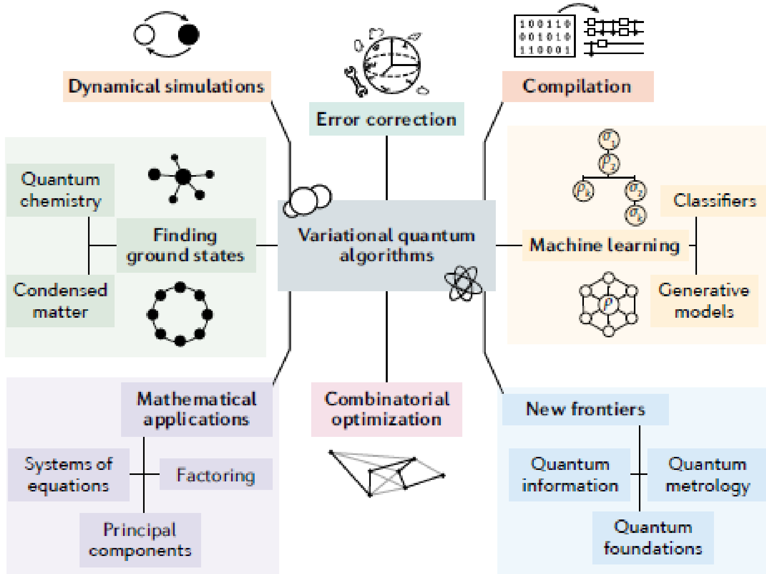
Even with today's most powerful classical computers, a quantum computer with 50 qubits is difficult to simulate
A 50+ Qubit quantum computer is already available to selected user groups ("intermediate-scale")
But, these qubits are noisy, i.e. their usability is limited (precision of operations, number of sequentially executed operations,…)
No practical use of quantum error correction yet
"Error Proneness": No more than about 1000 basic 2-qubit operations can be performed in sequence ⇒ Limitation of NISQ Technology

Source: from the slides on "Quantum Computing: A Brief Introduction " by Prof. Frank Leymann, SummerSoC 2019

TCS RESEARCH & INNOVATION

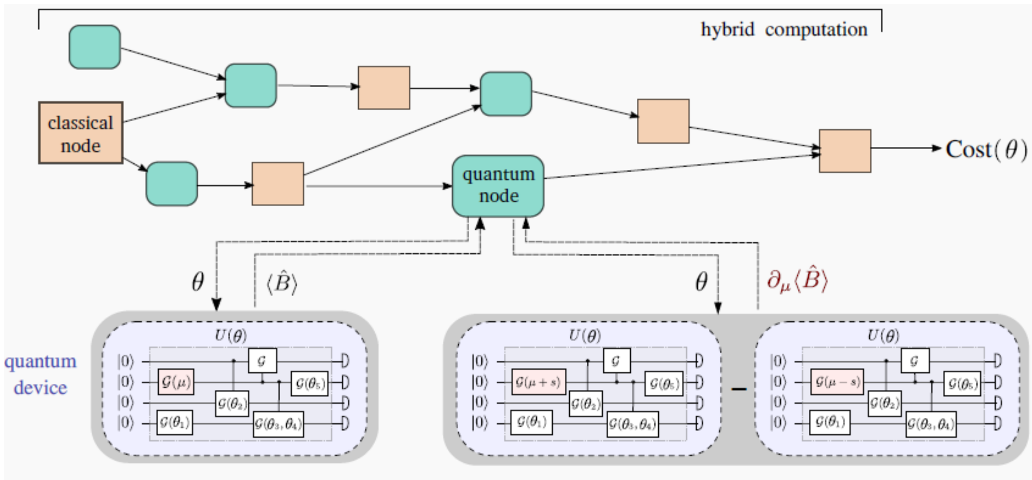
Variational Quantum Algorithms





Source:M. Cerezo, et al, Variational quantum algorithms, Nature Reviews, Sept 2021

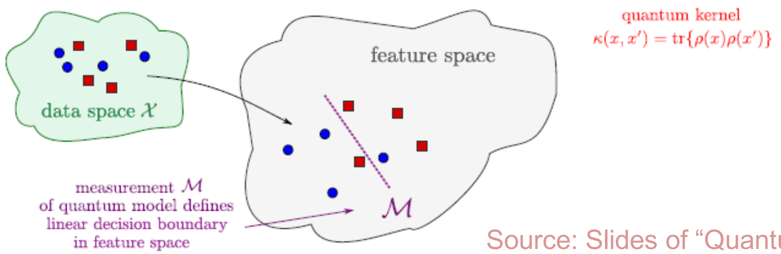
Gradients of Quantum Computations



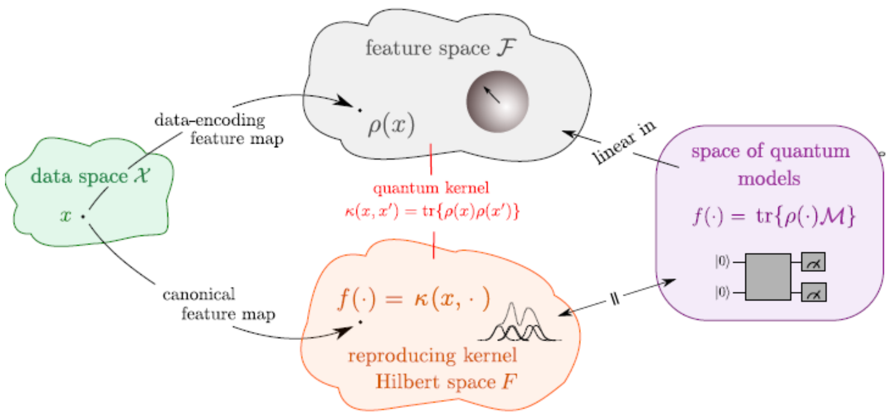
Source: Slides of Maria Schuld, Quantum Machine Learning, SMBQ, Sept. 2020

Quantum Circuit Models: Kernel Methods and Beyond

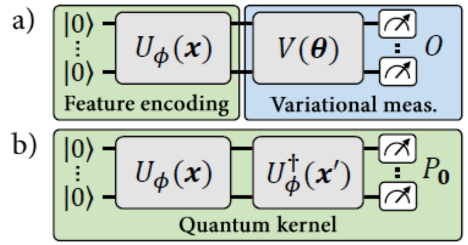
$$f(x) = \langle M_\theta \rangle_x = \mathrm{tr}\{\rho(x) M_\theta\}$$



Source: Slides of “Quantum Machine Learning”, SMBQ Sept. 2020, CERN Seminar Feb. 2021



... the power of parametrized quantum circuits can lie not only in the way they encode data in quantum states, but also in a restricted variational processing.



studied in this Letter. a) An explicit quantum classifier, where the label of a data point x is specified by the expectation value of a variational measurement on its associated quantum feature state $\rho(x)$. b) The quantum kernel associated to these quantum feature states. The expectation value of the projector $P_0 = |\mathbf{0}\rangle\langle\mathbf{0}|$ corresponds to the inner product between encoded data points $\rho(x)$ and $\rho(x')$. An implicit quantum classifier is defined by a linear combination of such inner products, for x an input point and x' training data points.

Source: Sofiene Jerbi, et al, “Quantum Machine Learning Beyond Kernel Methods”, https://arxiv.org/abs/2110.13162, Oct 2021

Support Vector Machines (SVMs) and Quantum SVM

c) Support Vector Machines: An *SVM* [39] consists of solving a convex quadratic optimization problem to find the hyperplane that results in the maximum margin between two classes of data. The dual problem

$$\max_{\alpha_i} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k K(x_i, x_k) \mid \alpha_i \geq 0 \quad (1)$$

is the one usually solved. $K : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$ is symmetric and positive semi-definite; a common example is the *Radial Basis Function*. K induces the, potentially *non-linear*, feature map $\phi(x) = K(\cdot, x) = K_x$. The Hilbert Space built from such maps is called the *Reproducing Kernel Hilbert Space* (RKHS) with reproducing kernel K. The optimal classifier in RKHS is one that is a linear combination of K_x's over a subset of the training data [34].

One of proposed quantum enhancements to the SVM is based on evidence that universal quantum computation, most likely, cannot be efficiently simulated on a classical computer [40]. Thus, one should be able to construct a quantum circuit for the map $\vec{x} \mapsto \mathcal{U}_{\Phi(\vec{x})} |0\rangle$; $\mathcal{U}_{\Phi(\vec{x})}$ is a unitary operation applied to the computational basis state consisting of all qubits in the $|0\rangle$ state, such that this operation is not classically feasible. This is called a *quantum feature map* and maps classical data $\vec{x}$ into a quantum Hilbert Space that is exponentially large in the dimension of $\vec{x}$. A potential quantum kernel is

$$K(x_i, x_j) = \| \langle 0 | \mathcal{U}^\dagger_{\Phi(\vec{x_i})} \mathcal{U}_{\Phi(\vec{x_j})} |0\rangle \|^2$$

which is symmetric and positive semi-definite. In this case the RKHS is spanned by the functionals, $K(\cdot, x)$, that are constructed from quantum circuits. The coefficients α_i of the decision function in RKHS can be computed by a convex optimizer running on a classical computer; the quantum computer is used to evaluate the kernel. This hybrid model is called QSVM [40]. There is potential quantum advantage in the expressability of the feature map, as long as the associated kernel is infeasible for a classical device to compute. This kernel can be computed on a quantum device utilizing either the *destructive SWAP* [41] or the *controlled-SWAP* [42] tests. The latter has better asymptotic complexity, but is not as feasible on NISQ devices. While the former's asymptotic scaling is prohibitive, it can efficiently be implemented on small quantum computers; this allows for experimentation in the near-term.

Source: Marco Pistoia, et al, "Quantum Machine Learning for Finance", https://arxiv.org/abs/2109.04298, Sept 2021

Variational Quantum Classifiers

d) Variational Quantum Classifiers: *Variational Quantum Classifiers* (VQCs) are hybrid quantum-classical ML architectures meant for classification tasks that utilize the quantum state space as a feature space to potentially obtain a quantum advantage. A VQC circuit mainly consists of a quantum embedding, a PQC for processing the quantum data, a measurement routine, and a classical optimization loop for updating the parameters of the PQC. First, classical input data $\vec{x}$ is mapped to a quantum state non-linearly using the feature-map circuit, $\mathcal{U}_{\Phi(\vec{x})}$, defined in Equation 2. Applying $\mathcal{U}_{\Phi(\vec{x})}$ to $|0\rangle^n$ results in the state $|\Phi(\vec{x})\rangle$.

Next, a PQC, $W(\vec{\theta})$, is constructed with parameters $\vec{\theta}$. An example of such a PQC is one made from compositions of single qubit rotations and entangling gates. PQC architectures have been discussed where descriptors, such as the entangling capability and expressibility, are used to characterize the performance of the PQCs [45].

In case of a binary-classification problem, a measurement routine is used to get a binary output. This is accomplished by measuring state $W(\vec{\theta})\mathcal{U}_{\Phi(\vec{x})}|0\rangle^n$ in the Pauli Z-basis and mapping the output bit-string to a function with binary outcome $f:\{0,1\}^n \rightarrow \{+1,-1\}$. The probability of obtaining an outcome, $y = \pm 1$, is

$$p_y(\vec{x}) = \sum_{i \in f^{-1}(y)} \|\langle i| W(\vec{\theta}) |\Phi(\vec{x})\rangle\|^2.$$

We repeat this step for R measurement shots, which gives an empirical distribution, $\hat{p}_y(\vec{x})$.

Then, a classical cost function is formulated to enable optimizing the parameters $(\vec{\theta}, b)$, where $b \in [-1, +1]$ is an added bias parameter. Once the classifier is trained on the training data set using a classical optimizer, the trained circuit can now be used to assign labels to unlabelled data. Several optimizers have been proposed and used, both gradient based, such as ADAM and SPSA [46, 47], and gradient-free ones, such as COBYLA [48].

VQCs have some limitations, and solving these drawbacks is an active area of research. Barren plateaus occur in optimization algorithms of quantum ML when the parameter search space turns flat once the optimizer is run [49, 50]. Architecture design problems, such as choosing the correct cost functions and initializing the parameters, is a very complex process that has not been completely understood yet [51]. Additionally, a given variational quantum circuit with fixed form may not be able to capture all of the necessary states in the Hilbert space in its parameterization, and as a result, work on adaptive variational quantum algorithms, such as the Evolutionary Variational Quantum Eigensolver (EVQE), may be applicable to VQC [52].

PQC- Parameterized Quantum Circuit

Source: Marco Pistoia, et al, “Quantum Machine Learning for Finance”, https://arxiv.org/abs/2109.04298, Sept 2021

Generative Modeling, Boltzmann Machines

A *Generative Model* learns a probability distribution over data [79]. In supervised learning, where the model is provided as a set of input/label pairs $\{(x_i, y_i)\}$, the model learns $P(X, Y)$, the joint probability distribution of inputs and labels [80]. In unsupervised learning, these models can be used to generate new data given only samples [81]. Since measuring a quantum state naturally results in a probability distribution over the outcomes, it makes sense to see if quantum computation can be utilized for generative modeling.

a) Boltzmann Machine: The *Boltzmann Machine* [82] is defined by a collection of *visible* (observed) and *hidden* (marginalized out) random variables, and an undirected graph of conditional dependencies among them. It originates from thermodynamics where the nodes represent a system of correlated classical spins, s_i, under an external magnetic field. The classical *Ising Hamiltonian*

$$\mathcal{H} = -\sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i$$

represents the energy of the system. Probabilistic inference is performed by sampling from the steady-state distribution—a Gibbs state—over the visible nodes. This is usually done utilizing Markov Chain Monte Carlo (MCMC) methods [82]. In most cases, the graph is restricted to being bipartite to make sampling feasible, resulting in the *Restricted Boltzmann Machine* (RBM) [83].

To formulate the *quantum* Boltzmann Machine, we quantize the Ising Hamiltonian by making the replacements $s_i \mapsto \sigma_i^z$, where σ_i^z is the Pauli Z spin operator for the i-th qubit. This results in a quantum Hamiltonian, and thus nodes are associated with qubits, and sampling is performed by projective measurements on the visible qubits.

One potential quantum method to sample from the visible nodes of the Gibbs state is to utilize *Quantum Annealing* (QA) [83–85]. For example, QA can be performed using the D-Wave devices [86].

Alternatively, we can prepare the quantum Gibbs state for this system by performing *Imaginary Time Evolution* (ITE) [87]. If the initial state is maximally mixed, performing ITE according to a quantum Hamiltonian will result in the associated Gibbs State. ITE can be performed variationaly, via McLachlan's principle, on a gate-based quantum computer [88]. Interestingly, the model introduced by Zoufal *et al.* [87] can be utilized to formulate a Boltzmann Machine without restricted connections that is tractable on a quantum device.

Source: Marco Pistoia, et al, "Quantum Machine Learning for Finance", https://arxiv.org/abs/2109.04298, Sept 2021

Some Points Worth Noting about QML

TCS RESEARCH & INNOVATION

Reasons for optimism

1. Powerful quantum tools for Linear Algebra
Machine Learning is mostly Linear Algebra – Matrix Multiplication, Linear Systems (training neural nets, linear regression, Support Vector Machines,…)
Quantum algorithms for Linear Algebra can offer great speedups in certain cases
2. Fast Distance Estimations
Simple quantum circuits for estimating distances between quantum states
3. Noise resilient algorithms
There is a lot of noise in ML data but the algorithms can deal with it
4. Improvements: Efficiency, Accuracy, Interpretability, Energy

Reasons for caution

1. Subtle quantum tools for Linear Algebra
One needs to be very careful about when quantum algorithms can offer speedups
2. Loading classical data as quantum states
Taking full advantage of quantum ML algorithms needs efficient quantum loaders
3. Getting classical information out of quantum algorithms
The quantum output encodes a classical solution that needs to be extracted
4. Benchmarking QML algorithms is difficult in the absence of hardware
Machine Learning must work in practice! How do we test?

Very interesting research area. A lot more work is needed!
Can we take advantage of the much bigger Hilbert space to learn better? (still not fully answered)

Seem to work OK for extremely small instances / simple data
Hope: they will continue to work well for bigger instances on larger hardware
Reality: Very difficult to benchmark

Quantum Data Loaders
• As powerful as Quantum Random Access Memory (QRAM) and much simpler to implement
• Optimal constructions on qubits and depth.
• Robust to noise

Source: Slides and Presentation of Iordanis Kerenidis, "Quantum Machine Learning: Algorithms and Practical Applications" QC Ware Q2B 2019

Quantum Linear Algebra
• Shallow Matrix Multiplications (NNs, classification, clustering)
• NISQer versions of Amplitude Estimation and Phase Estimation

Quantum Distance Estimators
• Shallow and noise robust circuits

Extracting classical information
• Optimal number of samples, shallow depth, no QRAM