QML-Mod5-QML Algorithm Complexity

Riccardo Marega

March 2025

Indice

1 QML Algorithm Complexity -13/06/2025 **2**

2 QML: practical aspects -14/06/2025 **3**

3 Noise characterization and mitigation -28/06/2025 **50**

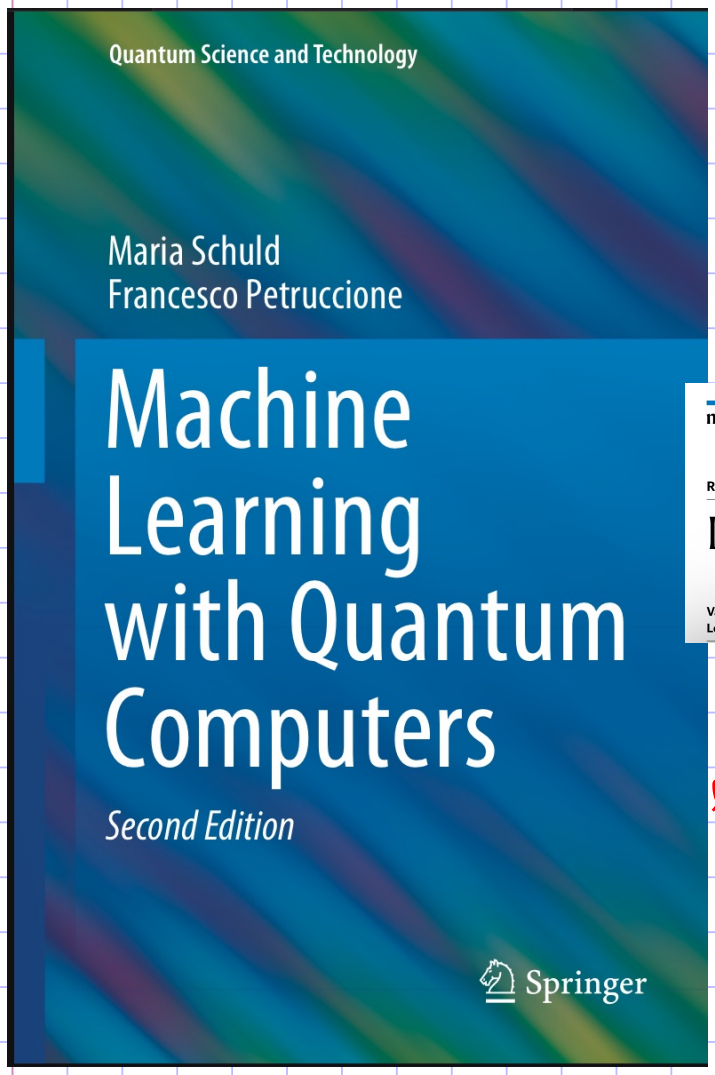
1 QML Algorithm Complexity -13/06/2025

All notes can be found on the ipynb.

2 QML: practical aspects -14/06/2025

QUANTUM MACHINE LEARNING
PRACTICAL ASPECTS

Classical C.S.



PROCEEDINGS A
rspa.royalsocietypublishing.org

Review

Cite this article: Ciliberto C, Herbster M, Ialongo AD, Pontil M, Rocchetto A, Severini S, Wossnig L. 2018 Quantum machine learning: a classical perspective. *Proc. R. Soc. A* **474**: 20170551.
http://dx.doi.org/10.1098/rspa.2017.0551

Quantum machine learning: a classical perspective

Carlo Ciliberto[1], Mark Herbster[1], Alessandro Davide Ialongo[2,3], Massimiliano Pontil[1,4], Andrea Rocchetto[1,5], Simone Severini[1,6] and Leonard Wossnig[1,5,7]

[1]Department of Computer Science, University College London, London, UK
[2]Department of Engineering, University of Cambridge, Cambridge, UK

nature reviews physics
https://doi.org/10.1038/s42254-022-00552-1

QC!

Review article

Check for updates

Learning quantum systems

Valentin Gebhart[1,2], Raffaele Santagati[3], Antonio Andrea Gentile[4], Erik M. Gauger[5], David Craig[6], Natalia Ares[7], Leonardo Banchi[8,9], Florian Marquardt[10,11], Luca Pezzè[1,2] & Cristian Bonato[5]

Quantum Experiments

LEARN

View

Quantum computing and artificial intelligence: status and perspectives

Giovanni Acampora,[1] Andris Ambainis,[2] Natalia Ares,[3] Leonardo Banchi,[4,5] Pallavi Bhardwaj,[6] Daniele Binosi,[7,8] G. Andrew D. Briggs,[9] Tommaso Calarco,[7,10] Vedran Dunjko,[11] Jens Eisert,[12] Olivier Ezratty,[13,14] Paul Erker,[15,16] Federico Fedele,[3] Elies Gil-Fuster,[12,17] Martin Gärttner,[18] Mats Granath,[19] Markus Heyl,[20,21] Iordanis Kerenidis,[22] Matthias Klusch,[23] Anton Frisk Kockum,[24] Richard Kueng,[25] Mario Krenn,[26] Jörg Lässig,[27,28] Antonio Macaluso,[23] Sabrina Maniscalco,[29] Florian Marquardt,[30] Kristel Michielsen,[10] Gorka Muñoz-Gil,[31] Daniel Müssig,[28] Hendrik Poulsen Nautrup,[31] Evert van Nieuwenburg,[11,32] Roman Orus,[33] Jörg Schmiedmayer,[15] Markus Schmitt,[10,34] Philipp Slusallek,[35,23] Filippo Vicentini,[36,37,38] Christof Weitenberg,[39] and Frank K. Wilhelm[35,10]

[1]University of Naples Federico II, I-80126 Naples, Italy

DATA PROCESSING

DATA		
	CC	CQ
	QC	QQ

C : Classical
Q : Quantum

CC : Classical Data
Classical Computer
(Classical (Quantum models)

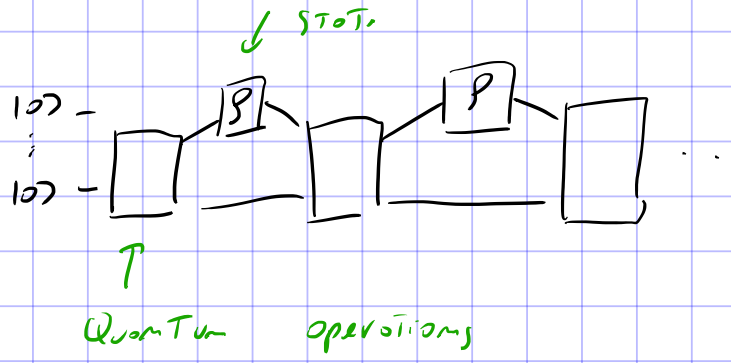
↳ Tensor Network Methods
Dequantized QML

QC : "Quantum Data"

↳ Wave Function collapse

No Cloning Theorem

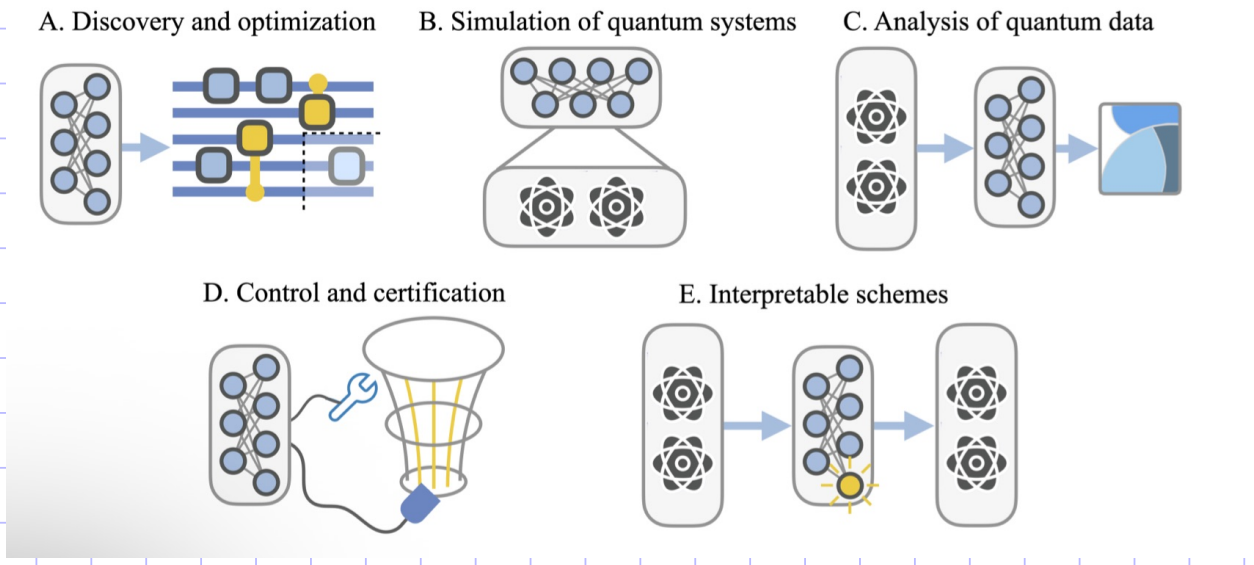
QQ : Quantum Parallelism



Exponential Advantage

CQ - Classical Data (Images)

- Quantum Computer



CQ

Classical

Quantum

Classical

CLASSICAL DATA

QUANTUM ENCODING

QUANTUM ALGORITHM

MEASUREMENTS

OUTPUT

N data

Load Data in Quantum Register

Unitary Gates (CNOT...)

Wave Function Collapse

Difficult

Quantum Parallelism

$O(N)$ $O(\log N)$

QRAM $O(\log N)$

Quantum Encoding / Data Loading

Classical data $\{x_i\}_{i=1 \dots N}$ $\quad x_i \in \mathbb{R}^d$

$x \Rightarrow |\psi(x)\rangle$

Examples

Amplitude Encoding

$$|\psi(x_i)\rangle = \sum_j \underbrace{(x_i)_j}_{\uparrow} |j\rangle$$

$$|j\rangle = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \leftarrow j^{th}$$

$$\prod_{j=1}^{d} e^{i \hat{G}(x_i)_j} |0\rangle$$

$\uparrow$ Hermitian operator / Pauli matrices

Parallel Quantum Strategies

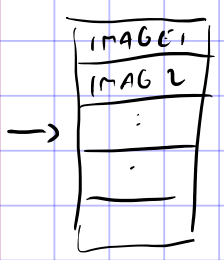
Quantum superposition

$$\{x_i\}_{i=1 \dots N} \rightarrow \sum_{i=1}^{N} |\psi(x_i)\rangle$$

Quantum RAM

$$\{x_i\}_{i=1 \dots N} \rightarrow \sum_{i=1}^{N} |i\rangle |\psi(x_i)\rangle$$

Address — memory content



Ensemble Strategies

$$\{x_i\} \Rightarrow \frac{1}{N} \sum_{i=1}^{\tilde{N}} |\psi(x_i)\rangle\langle\psi(x_i)|$$

$\{x_i\}$ → No Advantage

$|\psi(x_i)\rangle\langle\psi(x_i)|$ → Density Matrix

→ Test Quantum ideas

Quantum Computing Paradigms

Gate Based (Qubits)
- → Fault Tolerant Quantum Computer
- → NISQ
  - → Noisy Scale Intermediate Quantum Computers

Quantum Annealers

Continuous Variable / Photonics

→ Not Considered Here

FAULT TOLERANT

- Perfect Gates
- Many Qubits
- Algorithms : Everything in Quantum language

Quantum Circuits /
Shor (discrete log)
Grover (database search)

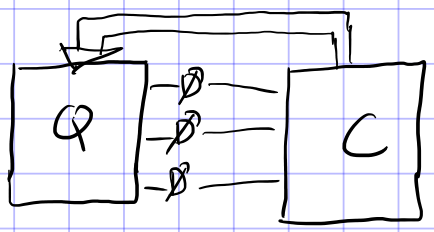
NISQ

Noisy out comes

10 - 100 Qubits (Free)

Quantum Resources are limited

Only use The Quantum Machine for Hard Tasks



Hybrid Quantum / Classical Computation

Theory : Asymptotics

$O(\log N)$

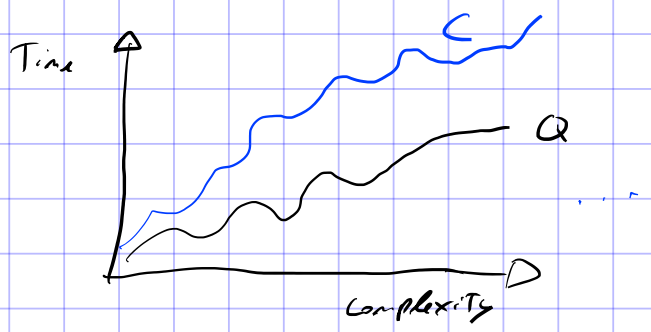
$N = 2^m$ $\quad O(\log N)$

Quantum $\quad$ Poly(m)

vs

Classical $\quad O(2^m) = O(N)$

"Hands on"



QML : Grover

↓

poly advantage

$O(N) \to O(\sqrt{N})$

HHL (linear system Eq.)

QSVD → Exp. adv.

QML
- Quantum Neural Networks
- Quantum Kernel Methods

"PRACTICAL" QML

IBM | Quantum | Technology | Qiskit | Research | Pricing | Blog | Community | Resources | Sign in to Platform

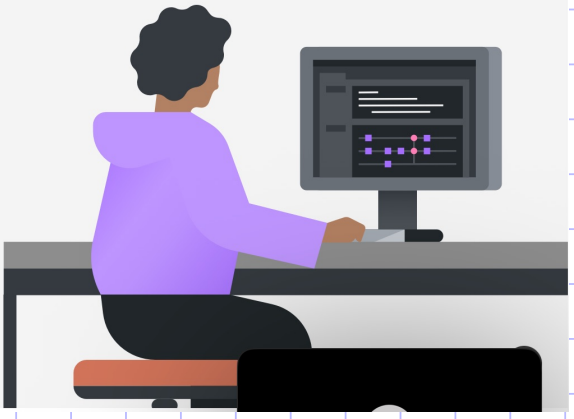
The new IBM Quantum Platform is here! The current experience will sunset on July 1. Prepare for the transition →

Qiskit SDK v2.0 is here
See what's new →

Qiskit

Qiskit is the world's most popular software stack for quantum computing. Build circuits, leverage Qiskit functions, transpile with AI tools, and execute workloads in an optimized runtime environment.

What is Qiskit? | Get started ↗



Learn quantum program | Codebook — PennyLane | Qiskit | IBM Quantum Co

pennylane.ai

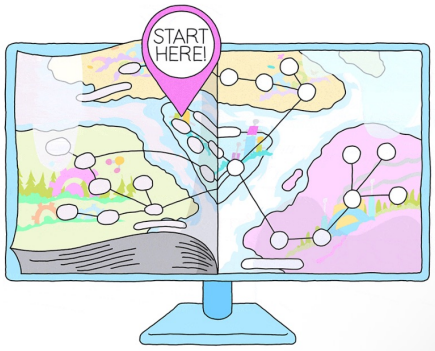
PENNYLANE | Sign in

Codebook

Learn quantum computing with PennyLane — the leading tool for programming quantum computers. Explore a specific module or follow a guided path to build your skills step-by-step.

Search

Open Codebook Map | Browse Modules

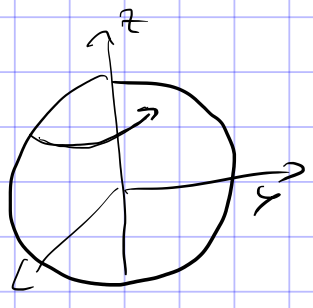


Quantum Gates

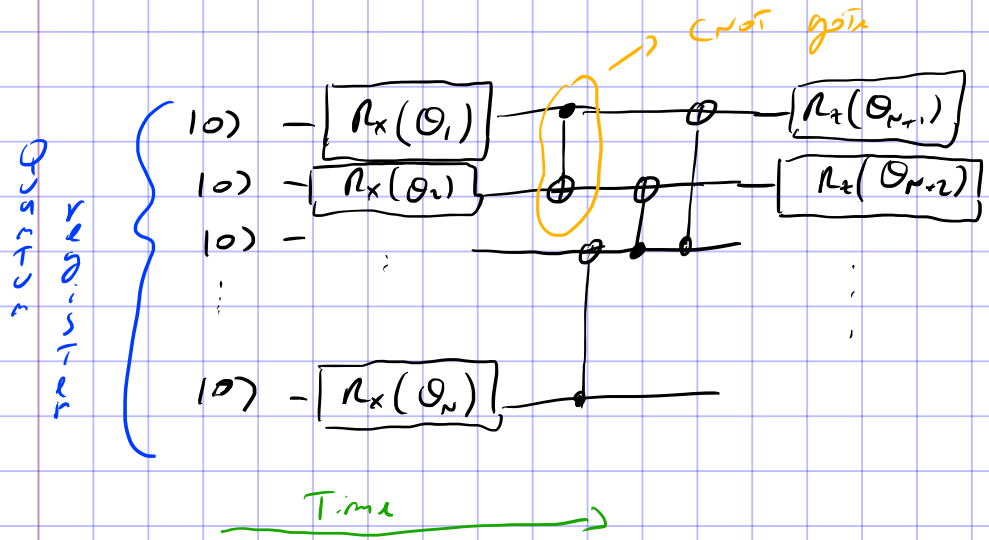
$e^{i\theta\hat{\sigma}_z} = R_z(\theta)$

Universality

- Single Qubit rotations R_x, R_y, R_z
- Entangling gate (CNOT)



Quantum Circuit

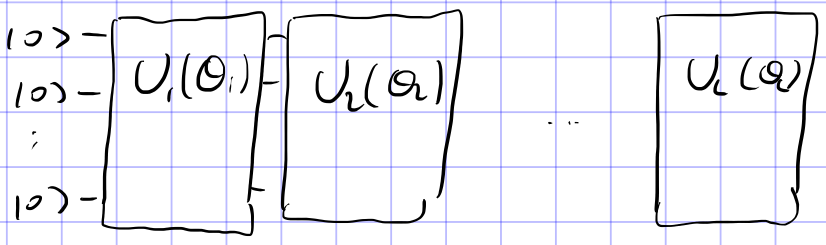


Parametric Quantum Circuit

$$|\psi(\underline{\theta})\rangle \qquad \underline{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \vdots \end{pmatrix}$$

Parameters

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



L = # Layers

Problem dependent "COST FUNCTION"

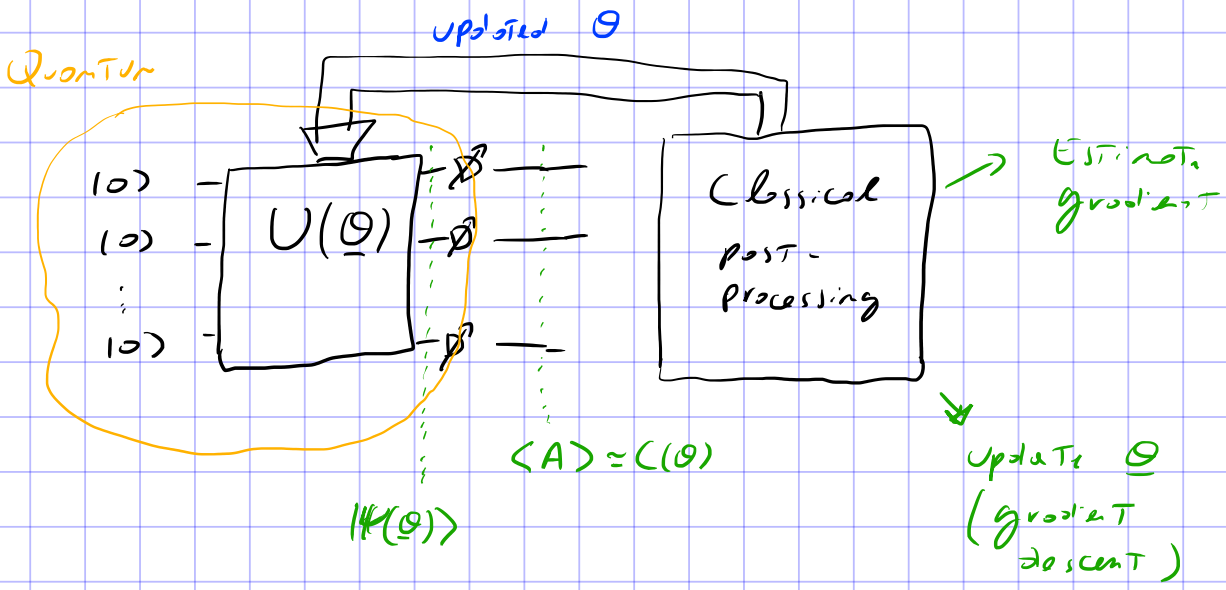
$$C(\theta) = \langle \psi(\theta) | \hat{A} | \psi(\theta) \rangle$$

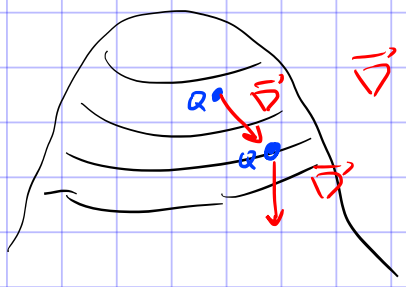
$\hat{A}$: Observable

$\langle \psi(\theta) | \hat{A} | \psi(\theta) \rangle$: Expectation value

"Training" = $\min_{\theta} C(\theta)$

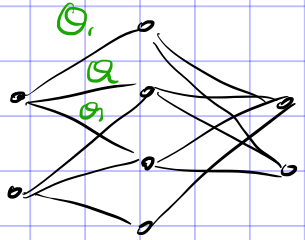
Hybrid Algorithm





θ

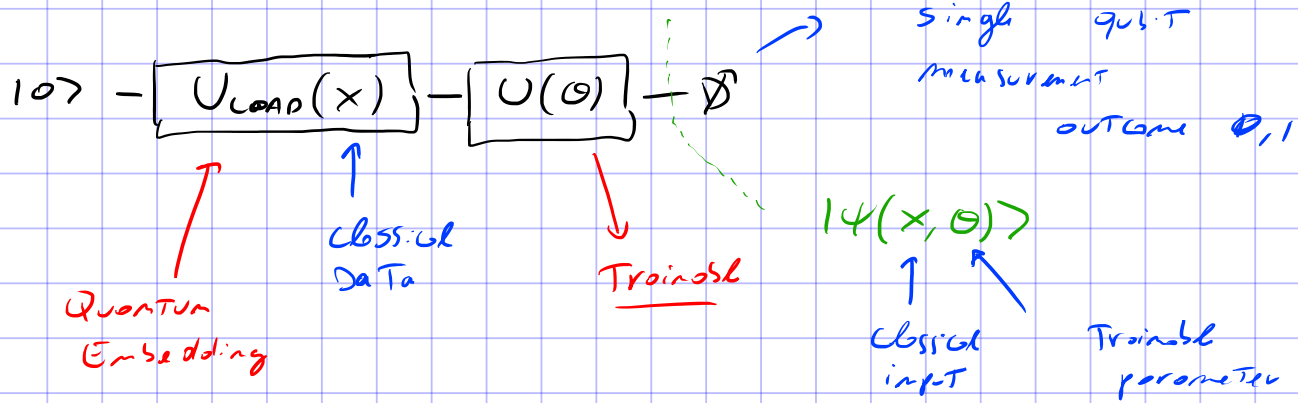
Quantum Neural Networks ≡ Parametric Quantum Circuit



Classical Feed forward Network

Universal Approximation Theorem

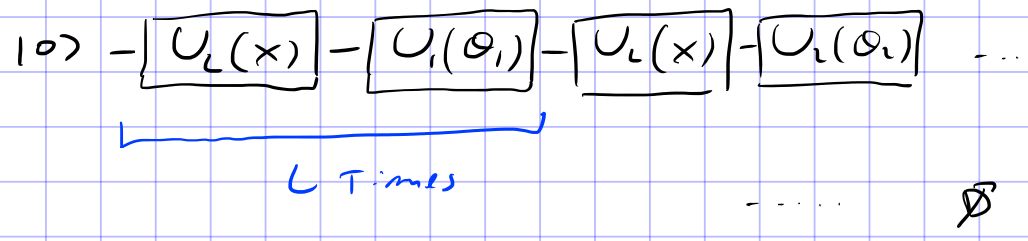
Example: Single-qubit variational quantum classifier



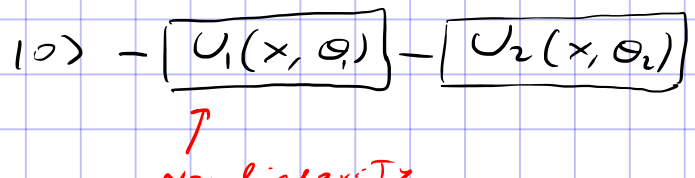
Outcomes

0	probability	$\lvert\langle 0 \lvert \psi(x,\theta)\rangle\rvert^2$
1	probability	$\lvert\langle 1 \lvert \psi(x,\theta)\rangle\rvert^2$

Many layers



Alternative



$$U_u(x,\theta) = U_u\left(\underline{\theta}_u + \underline{\underline{w}}^{(u)} \cdot \underline{x}\right)$$

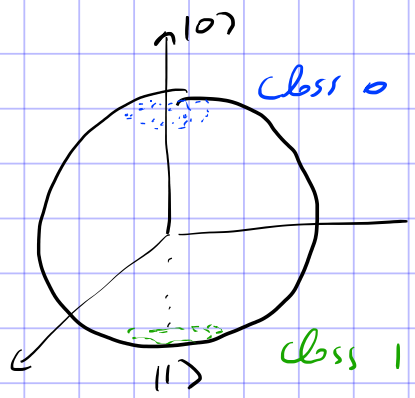
3-vector

$\underline{\theta}_u$: 3-vector; $\underline{\underline{w}}^{(u)}$: 3×d matrix; $\underline{x}$: d-vector

3 parameters

Trainable

Ideally a Trained model



binary classification problems

To Train a model we use Fidelity

$$F(x, y) = |\langle y | \psi(x, \theta) \rangle|^2 \quad \in [0, 1]$$

y: 0, 1 — $\psi(x, \theta)$: generic

$$loss(x, y) = 1 - F(x, y) \quad \in [0, 1]$$

(INFIDELITY)

Empirical Risk

$$R = \frac{1}{T} \sum_{i=1}^{T} loss(x_i, y_i)$$

T = # Training Data

Training data $\{(x_i, y_i)\}_{i=1 \dots T}$

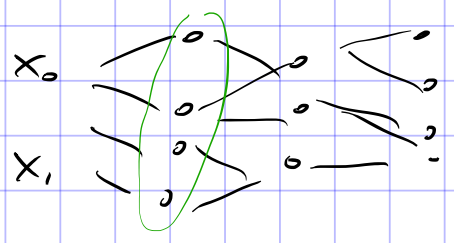
x_i: input — y_i: class

Training ≡ Empirical Risk Minimization

$$\underset{\theta}{\operatorname{argmin}} (R)$$

↳ Finding the θ that minimize R

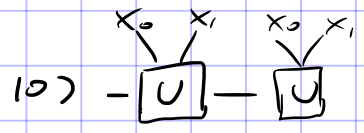
"Data Reuploading" idea



Non linear activation function

$$|0\rangle - |U_L(x)| - |U(\underline{\theta})| - \vec{\emptyset}$$

↑ Non linear



Data re-uploading for a universal quantum classifier

Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, José I. Latorre

A single qubit provides sufficient computational capabilities to construct a universal quantum classifier when assisted with a classical subroutine. This fact may be surprising since a single qubit only offers a simple superposition of two states and single-qubit gates only make a rotation in the Bloch sphere. The key ingredient to circumvent these limitations is to allow for multiple data re-uploading. A quantum circuit can then be organized as a series of data re-uploading and single-qubit processing units. Furthermore, both data re-uploading and measurements can accommodate multiple dimensions in the input and several categories in the output, to conform to a universal quantum classifier. The extension of this idea to several qubits enhances the efficiency of the strategy as entanglement expands the superpositions carried along with the classification. Extensive benchmarking on different examples of the single- and multi-qubit quantum classifier validates its ability to describe and classify complex data.

$$U(\underline{\Phi}) = e^{i\underline{\Phi}\cdot\hat{\underline{\sigma}}}$$

$$U(\underline{\phi}_1)\,U(\underline{\phi}_2) = e^{i\underline{\phi}_1\cdot\hat{\sigma}}\, e^{i\underline{\phi}_2\cdot\hat{\sigma}} = e^{i\vartheta(\underline{\phi}_1,\underline{\phi}_2)\cdot\hat{\sigma}}$$

↑ Non linear

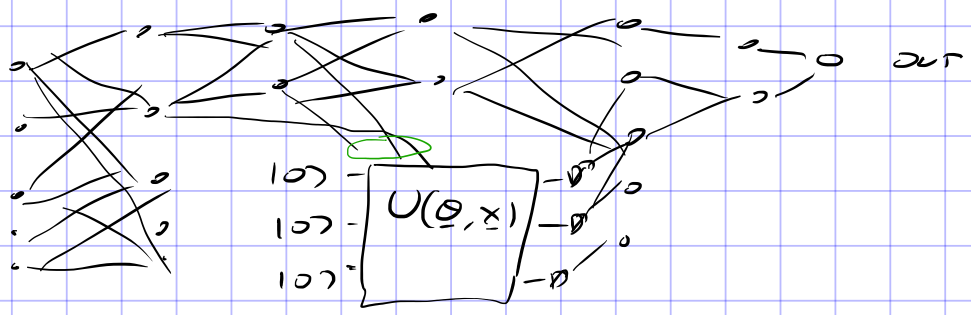
$$e^{ix\cdot\hat{\sigma}}\, e^{i\underline{\theta}_1\cdot\hat{\sigma}}\, e^{ix\cdot\hat{\sigma}}\, e^{i\theta_2\cdot\hat{\sigma}} = e^{i\vartheta(x,\underline{\theta}_1,\theta_2)\cdot\hat{\sigma}}$$

TRAINING QUANTUM NEURAL NETWORKS VIA GRADIENT DESCENT

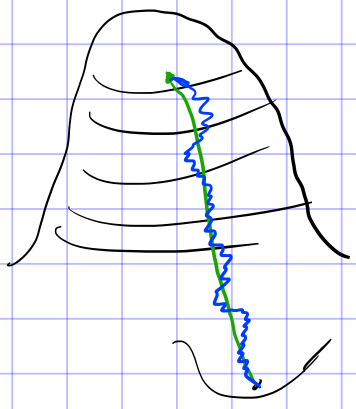
Why gradient descent?

1) Deep Learning → gradient descent

2) possible to integrate QNN with NN



3) Compatible with stochastic measurement outcomes
→ unbiased estimator of gradient



gradient descent

stochastic gradient descent

4) works very well in practice
→ good generalization

5) scaling $O(N)$ N: # Training data

Automatic Differentiation

F.F.N.

$$x^{(\ell+1)} = f_\ell\left(w_\ell \cdot x^{(\ell)} + b^{(\ell)}\right)$$

ℓ: layer

f_ℓ: activation function

w_ℓ, $b^{(\ell)}$: trainable weight

$x^{(0)} \equiv x$ input

$x^{(L+1)} = y$ output

$$C(\theta) = \mathbb{E}_{x,y \sim \text{data}}\ \text{loss}(x, y, \theta)$$

$$\frac{\partial C}{\partial w_{ij}^{\ell}} = \frac{\partial C}{\partial y}\frac{\partial y}{\partial w_{ij}^{\ell}} = \frac{\partial C}{\partial y}\frac{\partial x^{(L+1)}}{\partial x^{(\ell)}} \cdots \frac{\partial x^{(\ell+1)}}{\partial w_{ij}^{\ell}}$$

$$\frac{\partial x^{(u+1)}}{\partial x^{(u)}} = f_u' w^u$$

Integrate QNN. provided we can compute gradient

Suppose

$$f_\theta(x) = \langle \psi(x,\theta) | \hat{A} | \psi(x,\theta) \rangle$$

x: input

θ: Trainable parameters

$\psi(x,\theta)$: Trainable Embedding

$\hat{A}$: observable / measurement

Question

$$\frac{\partial f}{\partial \theta_i} \ ?$$

E.g. Finite differences ?

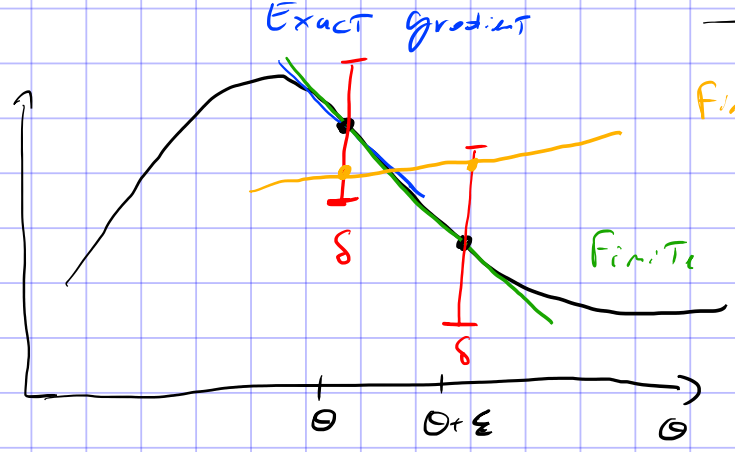
$$\frac{\partial f_\theta(x)}{\partial \theta_i} = \frac{f_{\underline{\theta} + \varepsilon \underline{e}_i}(x) - f_{\underline{\theta}}(x)}{\varepsilon} + O(\varepsilon^2)$$

$$\underline{e}_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad i^{th} \text{ element}$$

NOT GOOD

f_θ is estimated from measurements

→ Error bars → δ



if $\delta \ll \varepsilon$

Then Finite difference is ok

$$\hat{A} = \sum_e a_e |a_e\rangle\langle a_e|$$

a_e ↓ Eigenvalues — $|a_e\rangle\langle a_e|$ ↓ Eigenvectors

outcome a_e with probability $|\langle a_e|\psi(\theta,x)\rangle|^2$

$$\langle A \rangle = \langle \psi(\underline{\theta}, x) | \hat{A} | \psi(\underline{\theta}, x) \rangle \qquad \text{(block / exact point)}$$

$$= \frac{1}{M} \sum_{m=1}^{M} a_{\theta_m} \qquad \text{yellow part}$$

M # measurements

a_{θ_m} outcome of the m-th measurement

Variance $$\frac{\langle A^2 \rangle - \langle A \rangle^2}{M} = \delta^2$$

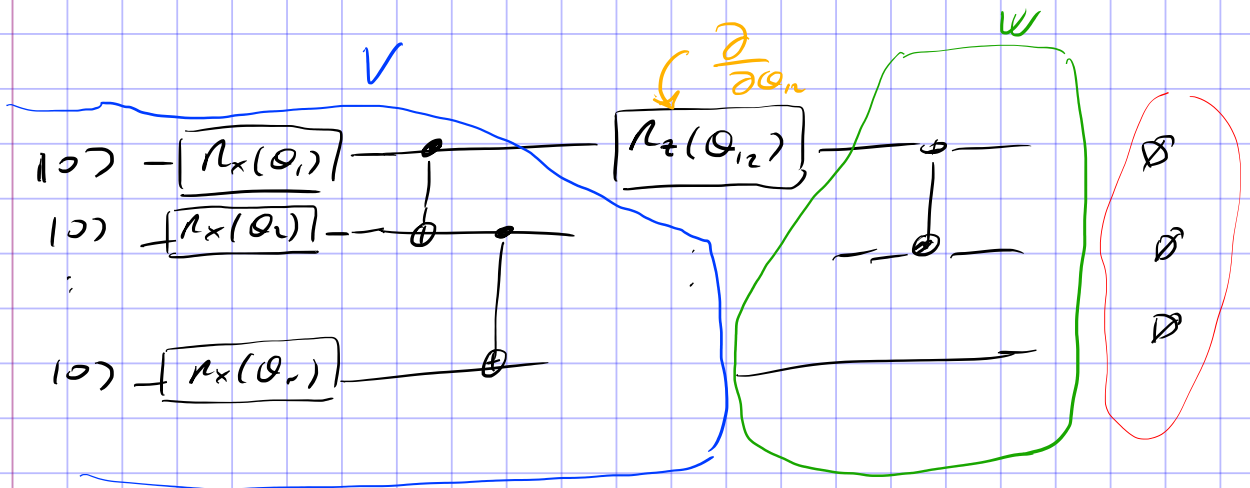
$$\delta \sim \frac{1}{\sqrt{M}} \ll \varepsilon$$

$$\Rightarrow \quad M \gg \frac{1}{\varepsilon^2}$$

$\varepsilon \sim 10^{-5}$

$M \sim 10^{10}$

EXACT CALCULATION OF GRADIENT



$$|\psi(\underline{\theta})\rangle = W e^{i\theta_m \hat{z}} V |0 \dots 0\rangle$$

$$g_\theta = \langle 0 \dots 0 | V^+ e^{-i\theta_m \hat{z}} W^+ A W e^{i\theta_m \hat{z}} V | 0 \dots 0 \rangle$$

$$\hat{z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \hat{z}^2 = \mathbb{1}$$

$\hat{\sigma}$ so that $\hat{\sigma}^2 = 1$

$$e^{i\theta\hat{\sigma}} = \sum_n \frac{(i\theta\hat{\sigma})^n}{n!} = \sum_{n\ \text{even}} \frac{(i\theta)^n}{n!}\mathbb{1} + \sum_{n\ \text{odd}} \frac{(i\theta)^n}{n!}\hat{\sigma}$$

$$= \cos\theta\,\mathbb{1} + i\sin\theta\,\hat{\sigma}$$

in general

$$f_\theta(x) = a(x)\cos^2\theta_{12} + b(x)\sin^2\theta_{12} + c(x)\cos(\theta)\sin(\theta)$$

$$\cos^2\theta_{12} = \frac{1+\cos(2\theta_{12})}{2} \qquad \sin^2\theta_{12} = \frac{1-\cos(2\theta_{12})}{2} \qquad \sin\theta_{12}\cos\theta_{12} = \frac{\sin(2\theta_{12})}{2}$$

$$= \frac{(a+b)}{2} + \frac{(a-b)}{2}\cos(2\theta_{12}) + \frac{c}{2}\sin(2\theta_{12})$$

$$\frac{\partial f}{\partial \theta_{12}} = -(a-b)\sin(2\theta_{12}) + c\cos(2\theta_{12})$$

$$\sin(a) = -\frac{\cos(a+u) - \cos(a-u)}{2\sin u} \qquad \forall u$$

$$\cos(a) = \ldots$$

$$\frac{\partial f}{\partial \theta_{12}} = \frac{f_{\theta+u} - f_{\theta-u}}{\sin(u)}$$

Exact

u not small

In practice we measure $\hat{A}$ with different circuits $\theta \to \theta \pm u$

$a_{\theta_n}^{(\pm)}$

$$\nabla f = \mathbb{E}\left(\frac{a_{0m}^{+} - a_{0m}^{-}}{\sin 4} \right)$$

Empirical Average over measurement shots

Unbiased estimator of the gradient

Variance $\sim (\sin 4)^{-2}$ small

$\Rightarrow$ $\sin 4$ large

$4 = \frac{\pi}{2}$

SUMMARY (PARAMETER SHIFT RULE)

$$\frac{\partial f_{\underline{\theta}}(\underline{x})}{\partial \theta_j} = f_{\underline{\theta} + \frac{\pi}{2}\underline{e}_j} - f_{\underline{\theta} - \frac{\pi}{2}\underline{e}_j}$$

Example Single qubit binary classifier

$$C(\theta) = \frac{1}{T} \sum_{x,y \in \text{Training}} \text{loss}(x, y, \theta)$$

$$= \frac{1}{T} \sum_{x,y} \left(1 - |\langle y | \psi(x, \theta) \rangle|^2 \right)$$

$$\langle \psi(x,\theta) | (|y\rangle\langle y|) | \psi(x,\theta) \rangle$$

$$y = 0 \quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$y = 1 \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\hat{z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\hat{A} = \frac{\mathbb{1} + \hat{z}(-1)^y}{2} \quad \begin{matrix} y=0 \to \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ y=1 \to \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

$$\partial_\theta(x) = \langle \psi(x,\theta) | \hat{z} | \psi(x,\theta) \rangle$$

$$C(\theta) = \frac{1}{T} \sum_{x,y} \left(1 - \frac{1 + (-1)^y \partial_\theta(x)}{2} \right)$$

$$\frac{\partial C}{\partial \theta_j} = \frac{1}{T} \sum_{x,y} -(-1)^y \left(\partial_{\theta + \frac{\pi}{4} e_j}(x) - \partial_{\theta - \frac{\pi}{4} e_j}(x) \right)$$

$$= C\left(\underline{\theta} + \tfrac{\pi}{4}\underline{e_j}\right) - C\left(\underline{\theta} - \tfrac{\pi}{4}\underline{e_j}\right) \quad \text{Exact}$$

Training via gradient descent

m'th iteration

$$\underline{\theta}_{m+1} = \underline{\theta}_m - \gamma \vec{\nabla} C$$

gradient

γ: Learning rate

BARREN PLATEAU

$$|\psi(\underline{x}, \underline{\theta})\rangle = \hat{W}(x)\, e^{i\theta_j \hat{H}_j}\, \hat{V}(x)\, |0 \cdots 0\rangle$$

$\hat{H}_j$ independent of θ_j

$$\hat{H}_j^2 = \mathbb{1}$$

$$f_{\underline{\theta}}(\underline{x}) = \langle \psi(\underline{x}, \underline{\theta}) | \hat{A} | \psi(\underline{x}, \underline{\theta}) \rangle$$

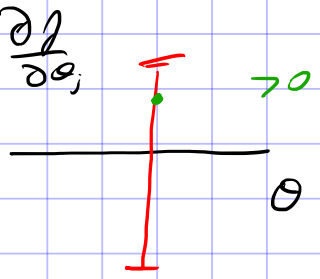
$$\frac{\partial f_{\underline{\theta}}}{\partial \theta_j} = -(a_j - b_j)\sin(2\theta_j) + c_j \cos(2\theta_j)$$

Suppose θ_j initialized randomly

uniform $[-\pi, \pi]$

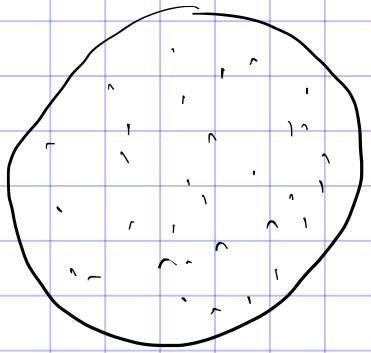
$$\mathbb{E}_{\theta_j}\left(\frac{\partial f}{\partial \theta_j}\right) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left(-(a_j - b_j)\sin(2\theta_j) + c_j \cos(2\theta_j)\right) d\theta_j$$

$$= 0$$



Variance with θ_j $\sim (a-s)^2 + c^2$

IF The circuit is Highly expressive



our circuit can explore a large part of space of unitaries

$$\mathbb{E}\left(\frac{\partial \mathcal{L}_\theta}{\partial \theta_j}\right) = 0$$

$$\mathrm{Var}\left(\frac{\partial \mathcal{L}_\theta}{\partial \theta_j}\right) = \mathcal{O}\left(\mathrm{poly}\left(\frac{1}{2^n}\right)\right)$$

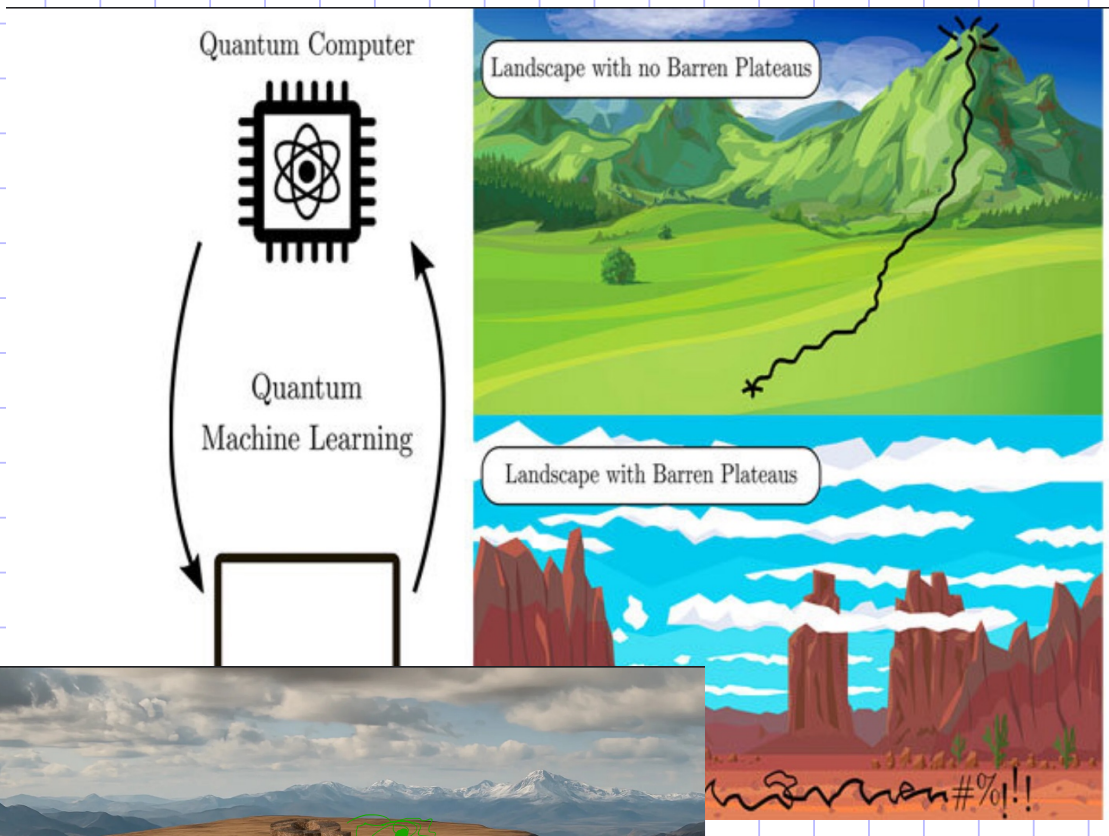
n # qubits

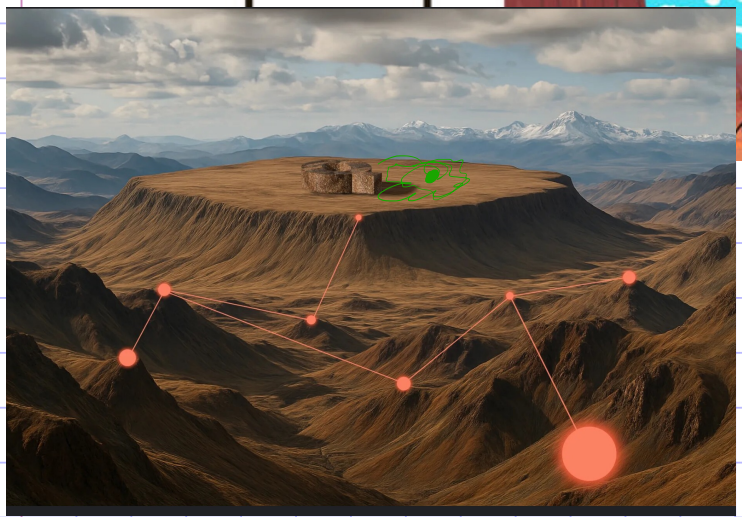
$$\mathbb{E}_{\substack{\text{all} \\ \text{other} \\ \text{parameters}}}\left((a-s)^2 + c^2\right)$$

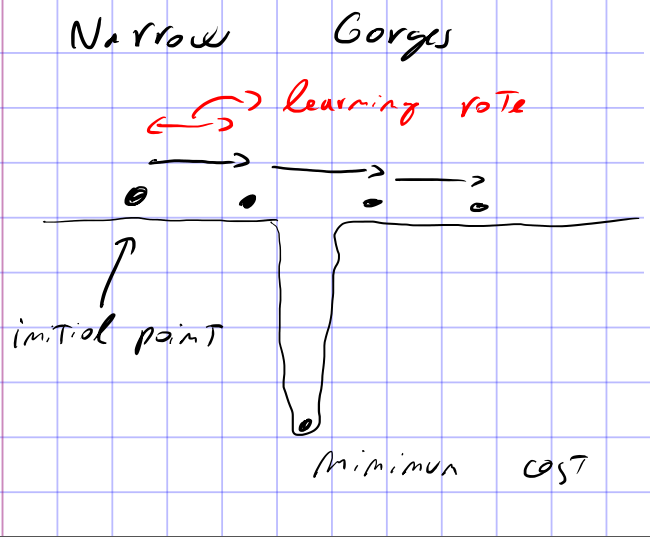
"Concentration of Measure"

Each sample with high prob. is close to the mean, but the mean is zero

The initial gradient can be very small





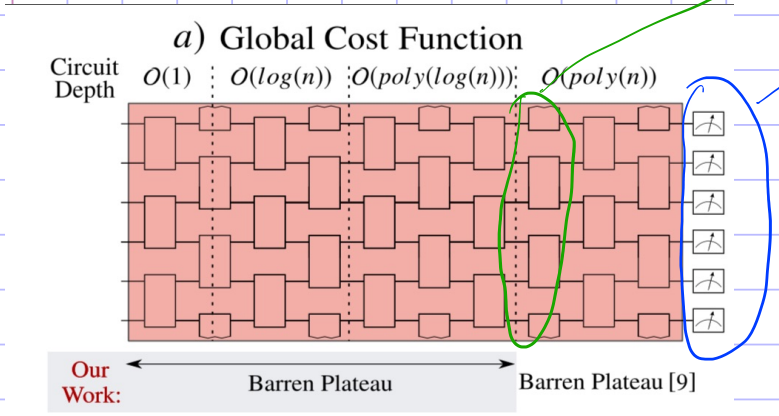


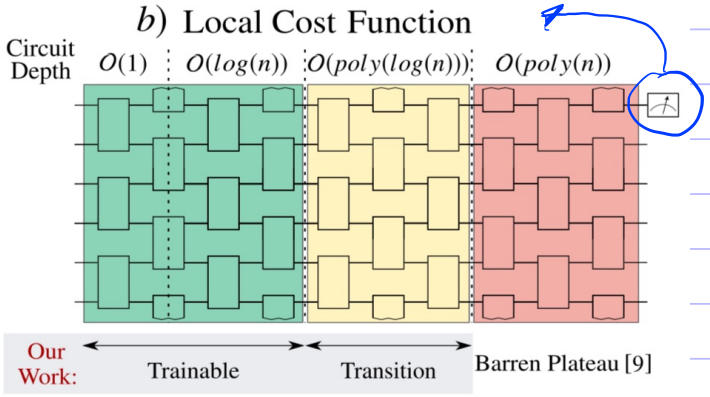
Article | Open access | Published: 19 March 2021

Cost function dependent barren plateaus in shallow parametrized quantum circuits

M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio & Patrick J. Coles

Nature Communications **12**, Article number: 1791 (2021) | Cite this article





No B.P.
No gorge

Code PlTeau.ipynb

Global cost function

$\hat{H} = -|00\cdots0\rangle\langle 00\cdots 0|$

zero Energy for all states except $|0\cdots 0\rangle$

Energy -1 for state $|0\cdots 0\rangle$

Local cost function with the same optimum of global cost function

$H = -\frac{1}{N}\sum_j |0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}$

identity on everything but j^{th} qubit

Quantum
the open journal for quantum science

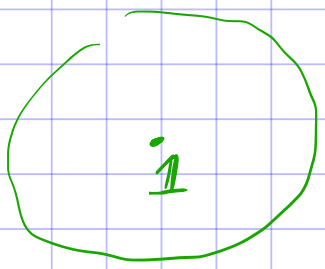
PAPERS PERSPECTIVES INST

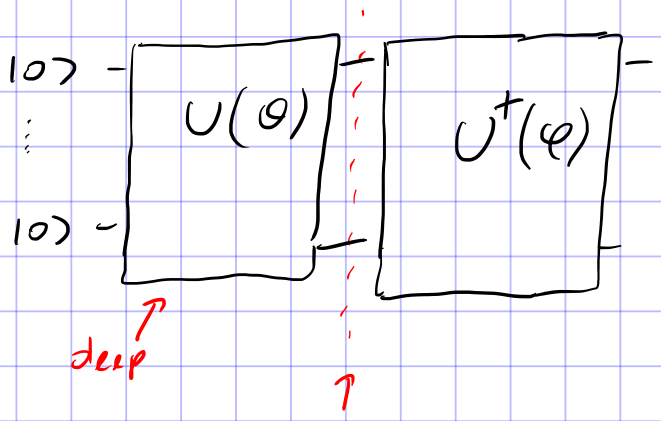
An initialization strategy for addressing barren plateaus in parametrized quantum circuits

Edward Grant[1], Leonard Wossnig[1], Mateusz Ostaszewski[2], and Marcello Benedetti[3]

[1]Rahko Limited & Department of Computer Science, University College London
[2]Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
[3]Cambridge Quantum Computing Limited & Department of Computer Science, University College London

Published: 2019-12-09, volume 3, page 214
Eprint: arXiv:1903.05076v3
Doi: https://doi.org/10.22331/q-2019-12-09-214
Citation: Quantum 3, 214 (2019).





initialize
$\theta = \varphi$

$U(\theta)U^\dagger(\varphi=\theta) = \mathbb{1}$

Cancel out
→ No B. P.

measurements → barren plateau

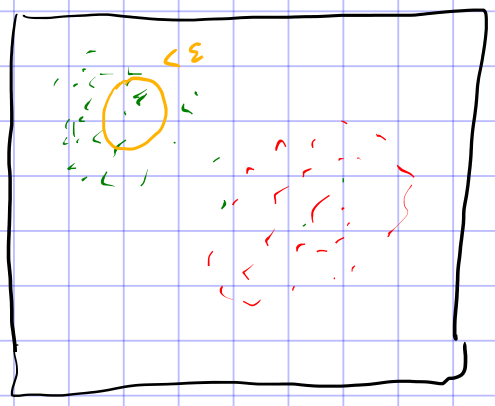
After initialization $\varphi = \theta$ Train independently
$\varphi \neq \theta$ during Training

UNSUPERVISED LEARNING

- Clustering

x_i y ← discrete

y ← cluster UNKnown



introduce a distance between points

$d(x_1, x_2)$

- Generative Models (Example Celebrity DataSet)

data $x \sim P(x)$ ← sampled from ← UNKnown

Task: find a parametric approximation of $P(x)$

$P_\theta(x)$ ← data

↑ Parameters

Goal: to ask to generate new samples

$x \sim P_\theta(x)$

Four samples should "look real"

Example "Celebrity DataSet"

Training minimize θ distance $\left(P_\theta(x), P(x) \right)$

Quantum Physics

[Submitted on 11 Apr 2018]

Differentiable Learning of Quantum Circuit Born Machine

Jin-Guo Liu, Lei Wang

Quantum circuit Born machines are generative models which represent the probability distribution of classical dataset as quantum pure states. Computational complexity considerations of the quantum sampling problem suggest that the quantum circuits exhibit stronger expressibility compared to classical neural networks. One can efficiently draw samples from the quantum circuits via projective measurements on qubits. However, similar to the leading implicit generative models in deep learning, such as the generative adversarial networks, the quantum circuits cannot provide the likelihood of the generated samples, which poses a challenge to the training. We devise an efficient gradient-based learning algorithm for the quantum circuit Born machine by minimizing the kerneled maximum mean discrepancy loss. We simulated generative modeling of the Bars-and-Stripes dataset and Gaussian mixture distributions using deep quantum circuits. Our experiments show the importance of circuit depth and gradient-based optimization algorithm. The proposed learning algorithm is runnable on near-term quantum device and can exhibit quantum advantages for generative modeling.

Comments: 9 pages, 7 figures, Github page for code this https URL
Subjects: **Quantum Physics (quant-ph)**; Machine Learning (cs.LG); Machine Learning (stat.ML)
Cite as: arXiv:1804.04168 **[quant-ph]**
(or arXiv:1804.04168v1 **[quant-ph]** for this version)
https://doi.org/10.48550/arXiv.1804.04168
Journal reference: Phys. Rev. A 98, 062324 (2018)
Related DOI: https://doi.org/10.1103/PhysRevA.98.062324

Quantum Physics

[Submitted on 23 Apr 2018 (v1), last revised 30 Apr 2018 (this version, v2)]

Quantum generative adversarial networks

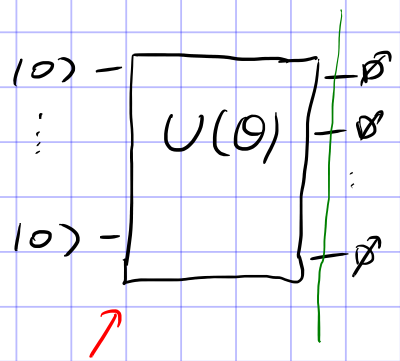
Pierre-Luc Dallaire-Demers, Nathan Killoran

Quantum machine learning is expected to be one of the first potential general-purpose applications of near-term quantum devices. A major recent breakthrough in classical machine learning is the notion of generative adversarial training, where the gradients of a discriminator model are used to train a separate generative model. In this work and a companion paper, we extend adversarial training to the quantum domain and show how to construct generative adversarial networks using quantum circuits. Furthermore, we also show how to compute gradients -- a key element in generative adversarial network training -- using another quantum circuit. We give an example of a simple practical circuit ansatz to parametrize quantum machine learning models and perform a simple numerical experiment to demonstrate that quantum generative adversarial networks can be trained successfully.

Comments: 10 pages, 8 figures
Subjects: **Quantum Physics (quant-ph)**; Machine Learning (cs.LG)
Cite as: arXiv:1804.08641 **[quant-ph]**
(or arXiv:1804.08641v2 **[quant-ph]** for this version)
https://doi.org/10.48550/arXiv.1804.08641
Journal reference: Phys. Rev. A 98, 012324 (2018)
Related DOI: https://doi.org/10.1103/PhysRevA.98.012324

Main idea

sample data from measurement of a Quantum State



parametric Quantum Circuit

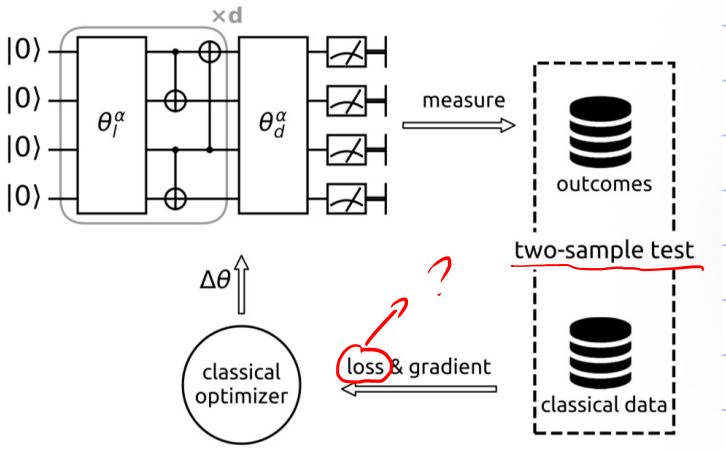
$|\psi(\theta)\rangle$

→ measured outcomes → x

Probability

$$P_\theta(x) = |\langle x|\psi(\theta)\rangle|^2$$

Quantum circuit Born Machine



Maximum Mean Discrepancy → MMD

Embedding Function

$$\text{Loss}(\theta) = C(\theta) = \left\| \mathop{\mathbb{E}}_{x\sim p_\theta}[\phi(x)] - \mathop{\mathbb{E}}_{x\sim p}[\phi(x)] \right\|_2^2$$

p_θ: parametric distribution; p: true distribution

$$= \mathop{\mathbb{E}}_{x,y\sim p_\theta}[\phi(x)\cdot\phi(y)] + \mathop{\mathbb{E}}_{x,y\sim p}[\phi(x)\cdot\phi(y)]$$

$\phi(x)\cdot\phi(y)$: inner product

$$- 2 \mathop{\mathbb{E}}_{\substack{x\sim p_\theta \\ y\sim p}}[\phi(x)\cdot\phi(y)]$$

Kernel Function

$$K(x,y) = \phi(x)\cdot\phi(y)$$

Choosing ϕ $\Longleftrightarrow$ Choosing K

good choice: Gaussian Kernel $K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

Thm MMD + Gaussian Kernel

$C = 0$ iff $P(x) = P_\theta(x)$

Training

$$P_\theta(x) = |\langle x|\psi(\theta)\rangle|^2$$

$$= \langle\psi(\theta)|\underbrace{|x\rangle\langle x|}_{\hat{A}}|\psi(\theta)\rangle$$

parameter shift rule

$$\frac{\partial P_\theta(x)}{\partial \theta_j} = P_{\theta + \frac{\pi}{s}e_j}(x) - P_{\theta - \frac{\pi}{s}e_j}(x)$$

arbitrary function

$$\frac{\partial}{\partial \theta_j} \mathop{\mathbb{E}}_{x\sim P_\theta}[f(x)] = \frac{\partial}{\partial \theta_j} \sum_x P_\theta(x) f(x)$$

$$= \sum_x f(x)\left(P_{\theta + \frac{\pi}{s}e_j}(x) - P_{\theta - \frac{\pi}{s}e_j}(x)\right)$$

$$= \mathop{\mathbb{E}}_{x\sim P_{\theta+\frac{\pi}{s}e_j}}[f(x)] - \mathop{\mathbb{E}}_{x\sim P_{\theta-\frac{\pi}{s}e_j}}[f(x)]$$

$$\theta_j^{\pm} = \theta \pm \frac{\pi}{s}e_j$$

$$\frac{\partial C}{\partial \theta_j} = 2\Big[\mathop{\mathbb{E}}_{x\sim P_{\theta_j^+}}[k(x,y)] - \mathop{\mathbb{E}}_{x\sim P_{\theta_j^-}}(k(x,y)) \quad (y\sim P_\theta)$$

Four

$$+ \mathop{\mathbb{E}}_{x\sim P_{\theta_j^+}}(k(x,z)) - \mathop{\mathbb{E}}_{x\sim P_{\theta_j^-}}(k(x,y))\Big] \quad (y\sim P)$$

True distribution

→ Unknown → in practice we use Training data

Qiskit $e^{i\frac{\theta}{2}\hat{\sigma}}$

My $e^{i\theta\hat{\sigma}}$

Algorithm

- Sample S elements "x" from P_θ, P_{θ^*} using S "shots"

Call outcomes x_s^0 $x_s^\pm$

$s = 1 \dots S$

- Take a batch of Training Data

x_s^b $s = 1 \dots S$

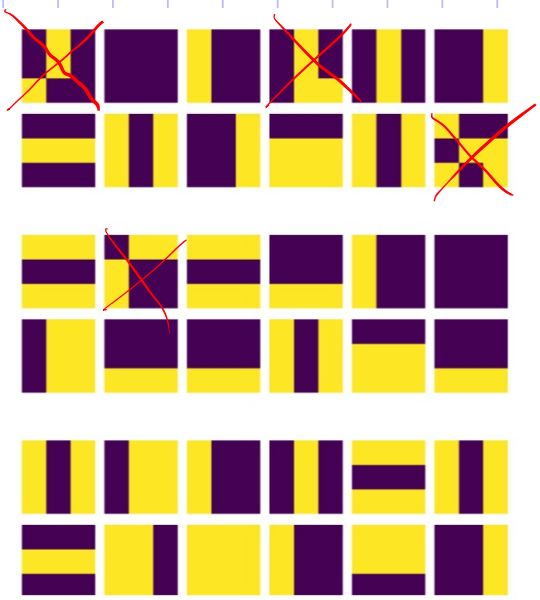
Training Data has to be bigger than S

Empirical average

$$\frac{\partial C}{\partial \theta_j} \simeq \frac{2}{S^2} \sum_{t,s=1}^{S} \Big[K(x_s^+, x_t^0) - K(x_s^-, x_t) + K(x_s^+, x_t^b) - K(x_s^-, x_t^s) \Big]$$

x_t^b: Training data; x_t^s: data generated by the Quantum Computer

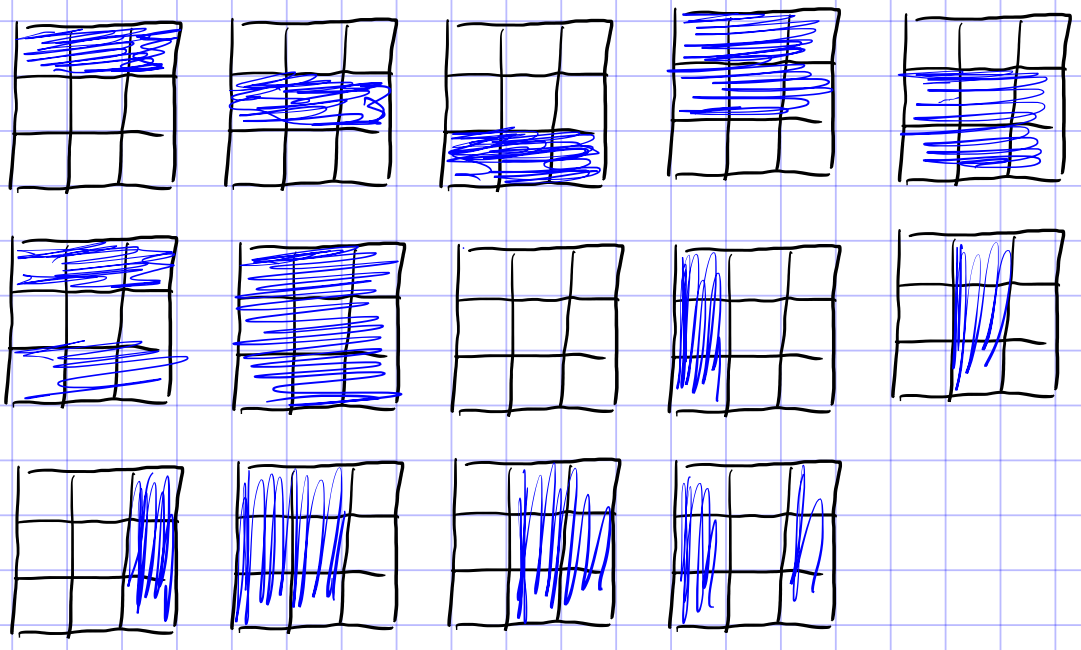
everything else is from data generated by the Quantum Computer



binary 3 × 3 images

$$\begin{array}{|c|c|c|}\hline x_{11} & x_{12} & x_{13} \\ \hline x_{21} & x_{22} & x_{23} \\ \hline x_{31} & x_{32} & x_{33} \\ \hline \end{array} \qquad x_{ij} \in \{0,1\}$$

Rule: True images are either BANDS or STRIPES



16 "True images" 2^9 possible configurations ~ 500

QUANTUM GENERATIVE ADVERSARIAL NETWORK

Classical GANs

2 players
- Generator G
- Discriminator D

G wants to generate fake data to fool D

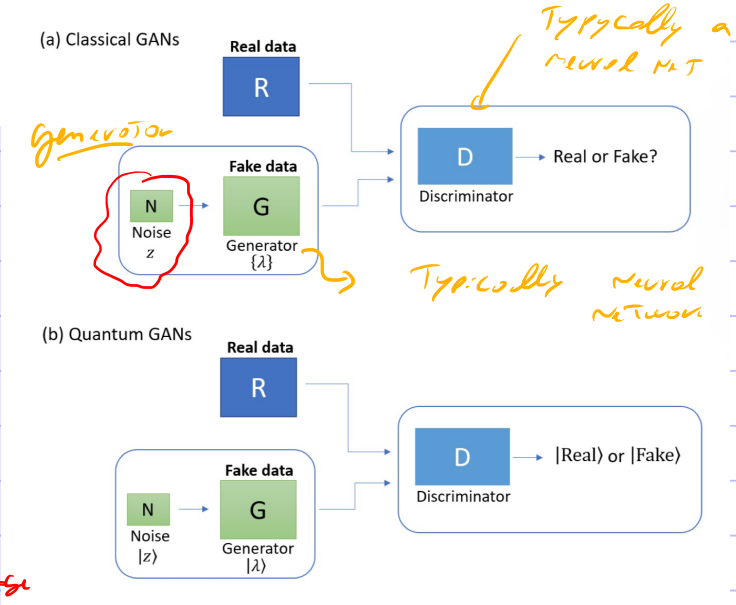
D judges if data is fake or real

Turn based game: G, D iteratively update their strategy to win

Theorem (Nash Equilibrium)
- G produces good data
- D cannot tell if data is real or fake

Quantum generative adversarial networks with Cirq + TensorFlow

Nathan Killoran



Classical GANs

$z \sim N(0, 1)$

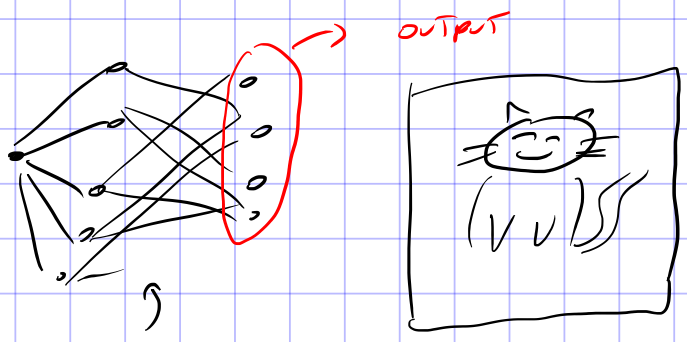
Generator

$x = g_\theta(z)$

↳ generated fake image

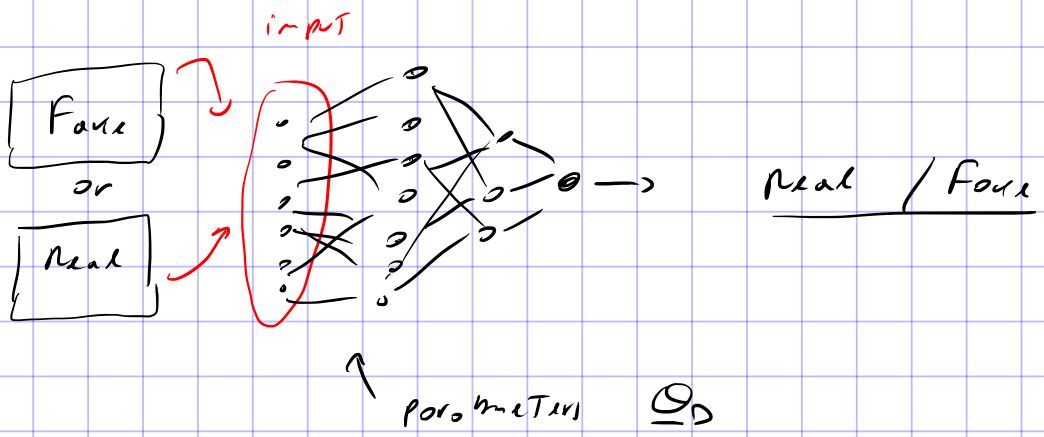
Classical Generator

$z \sim N(0,1)$



parameters θ_G

Classical Discriminator



parameters θ_D

$V(\theta_G, \theta_D)$ score function as misclassification probability

goal of G: minimize $V(\theta_G, \theta_D)$ over θ_G

goal of D: maximize $V(\theta_G, \theta_D)$ over θ_D

$\min_{\theta_D} \max_{\theta_G} V(\theta_G, \theta_D)$ GANs

Quantum GANs

ρ_R real Quantum states

ρ_F Fake Quantum states

if data is classical

$$\{x_i\}_{i=1\dots N} \rightarrow |\psi\rangle$$

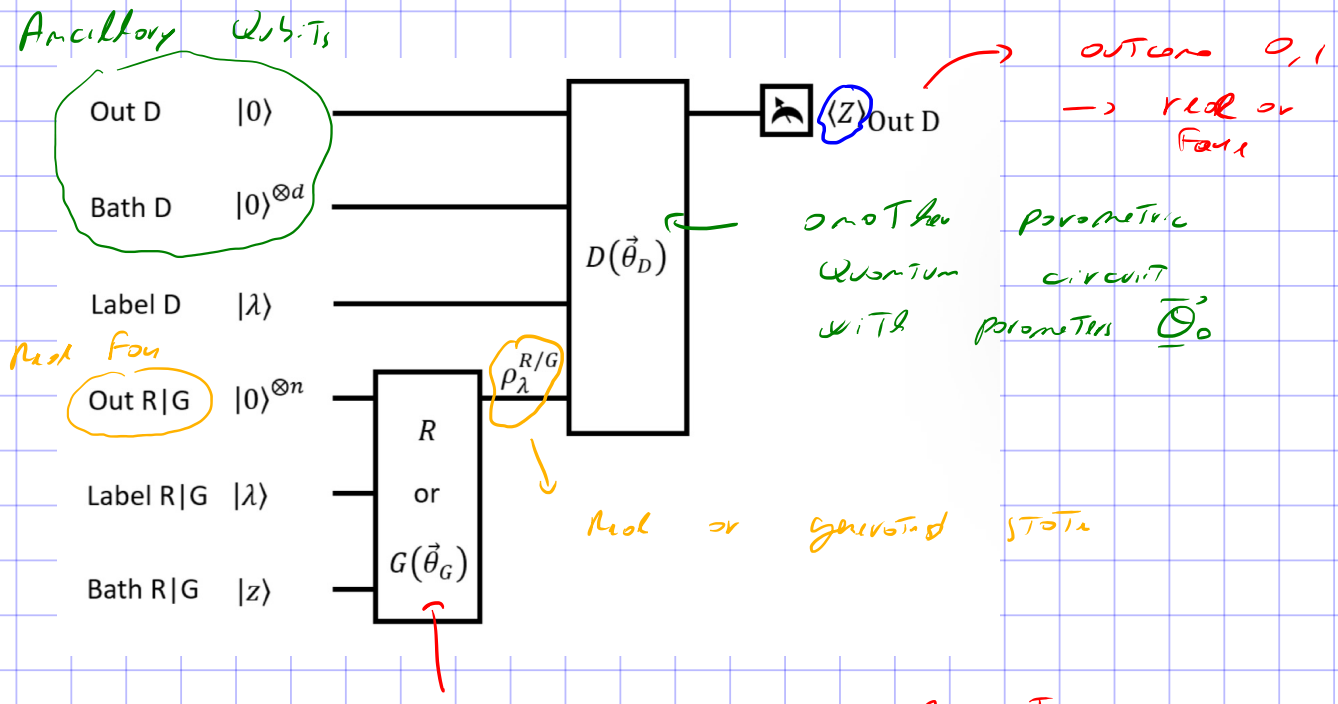
$$\rho = \frac{1}{N}\sum_{i=1}^{N} |\psi(x_i)\rangle\langle\psi(x_i)|$$

G outputs some quantum state

$$G(|\lambda\rangle, z) = |\psi_\lambda(z)\rangle\langle\psi_\lambda(z)|$$

D gets either real or fake state

R real data $R(|\lambda\rangle) = |\psi(x_\lambda)\rangle$



$\hat{z} = |\text{out}| - |1 - \text{out}|$ → 0 → real, 1 → fake

Why ancillary qubits ?

State discrimination ρ_0, ρ_1 Two possible states

Most general measurement strategy

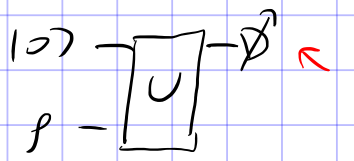
POVM (Positive Operator valued Measurement)

Π_0, Π_1 Hermitian

$\Pi_0, \Pi_1 \geq 0$

$\hat{\Pi}_0 + \hat{\Pi}_1 = \mathbb{1}$

Theorem (Naimark) $Tr(\Pi_i \rho)$ → probability of getting outcome i



↖ projective measurement with outcome 0, 1

↑ Circuit (Reversible)

$V(\underline{\theta}_G, \underline{\theta}_D) = \mathbb{E}_{data}$ P(real is real and fake is fake)

↑ discriminator prediction

↑ Measurement of the a Quantum circuit

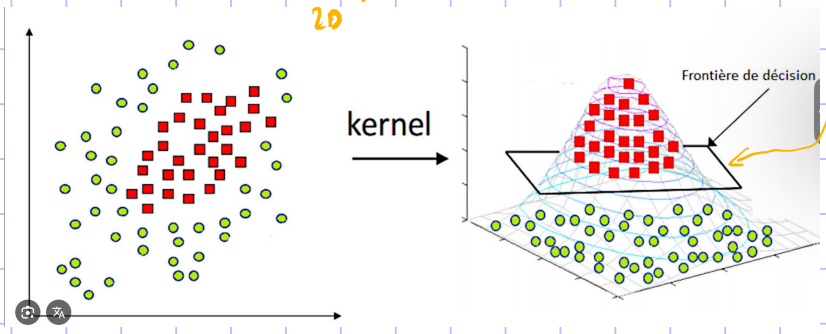
→ gradients Parameter shift rule

QUANTUM KERNEL METHODS

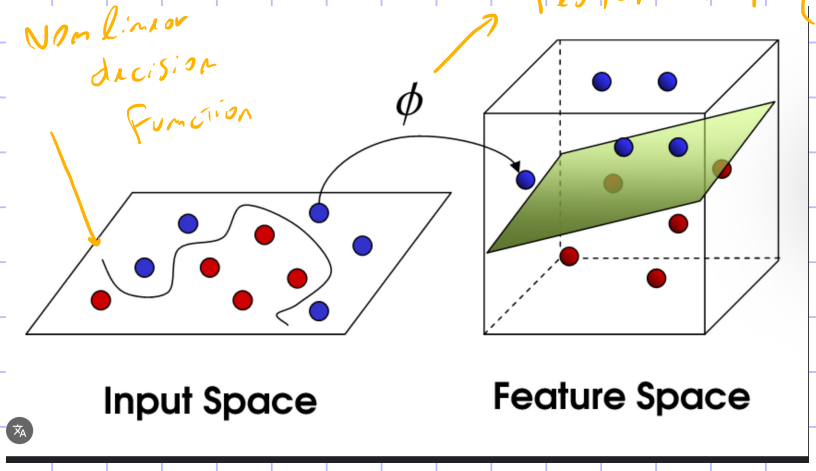
linear classifiers on a bigger space

→ intuition → data are easier to classify in a larger space

No linear classifier in 2D → line



linear classifier exists in 3D → plane



linear classifier → Hyperplane

blue points above

red points below

Non linear decision problems as linear decision problems on "feature space"

Feature map between input To "feature space"

$$\phi : x \rightarrow \phi(x)$$

x: input

$\phi(x)$: feature

Feature space → Hilbert space
(space of quantum states)

⇒ Quantum feature map

$\phi : x \to |\psi(x)\rangle$

Quantum Physics

{Submitted on 26 Jan 2021 (v1), last revised 17 Apr 2021 (this version, v2)}

Supervised quantum machine learning models are kernel methods

Maria Schuld

With near-term quantum devices available and the race for fault-tolerant quantum computers in full swing, researchers became interested in the question of what happens if we replace a supervised machine learning model with a quantum circuit. While such "quantum models" are sometimes called "quantum neural networks", it has been repeatedly noted that their mathematical structure is actually much more closely related to kernel methods: they analyse data in high-dimensional Hilbert spaces to which we only have access through inner products revealed by measurements. This technical manuscript summarises and extends the idea of systematically rephrasing supervised quantum models as a kernel method. With this, a lot of near-term and fault-tolerant quantum models can be replaced by a general support vector machine whose kernel computes distances between data-encoding quantum states. Kernel-based training is then guaranteed to find better or equally good quantum models than variational circuit training. Overall, the kernel perspective of quantum machine learning tells us that the way that data is encoded into quantum states is the main ingredient that can potentially set quantum models apart from classical machine learning models.

Comments: 26 pages, 9 figures - Version 2 emphasises focus on supervised learning, adds more references to existing literature, deletes section on state discriminator due to a technical error, and updates the comparison between kernel-based and variational training
Subjects: **Quantum Physics (quant-ph)**; Machine Learning (stat.ML)
Cite as: arXiv:2101.11020 **[quant-ph]**
(or arXiv:2101.11020v2 **[quant-ph]** for this version)
https://doi.org/10.48550/arXiv.2101.11020

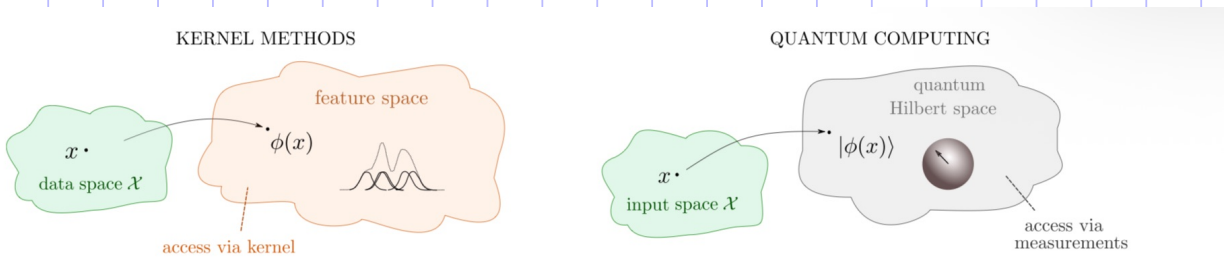


FIG. 1. **Quantum computing and kernel methods are based on a similar principle.** Both have mathematical frameworks in which information is mapped into and then processed in high-dimensional spaces to which we have only limited access. In kernel methods, the access to the feature space is facilitated through *kernels* or inner products of feature vectors. In quantum computing, access to the Hilbert space of quantum states is given by measurements, which can also be expressed by inner products of quantum states.

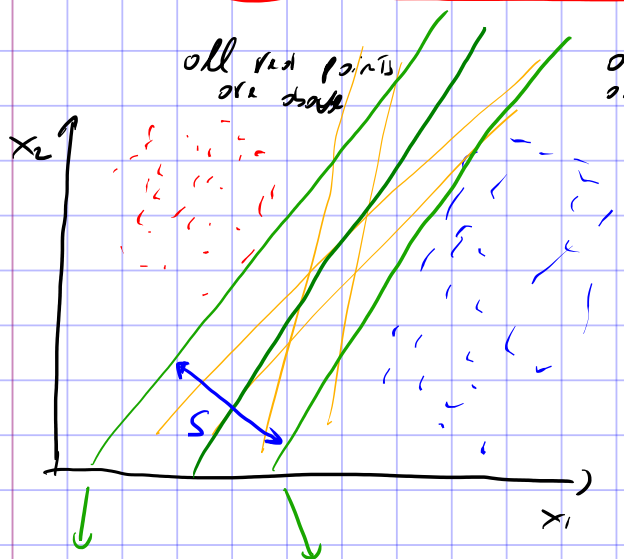
Classically → Kernel Trick

$k(x,y) = \phi(x) \cdot \phi(y)$

inner product between two feature maps

Choosing Feature Map ⟷ Choosing Kernel

SUPPORT VECTOR MACHINES



(linear SVM)

several possible data separation Hyperplanes

optimal Hyperplane

learnable coefficients

Data

$w^T \cdot x - b = 0$

w, x: d-dimensional vector; b: $\mathbb{R}$

parallel Hyperplanes

separation $S = \frac{2}{\|w\|}$

2D $w_1 x_1 + w_2 x_2 - b = 0$ → line

Select The Hyperplanes with maximum "margin" $\frac{2}{\|w\|}$

We need To solve

$w^T x_i - b \geq 1$ red points

$w^T x_i - b \leq -1$ blue points

$y_i = 1$ red points

$y_i = -1$ blue points

(binary classification)

$y_i (w^T x_i - b) \geq 1$ Compact form

$1 - y_i (w^T x_i - b) \leq 0$ (Hard Constraint)

Soft Constraint — Loss whenever the inequality is violated

Hinge loss $\ell(x_i, y_i) = \max\{0, 1 - y_i(w^T x_i - b)\}$

$\ell(x_i, y_i) = 0$ if on the right side

otherwise $\ell(x_i, y_i) \geq 0$

Cost function

$$C(w, b) = \lambda \|w\| + \sum_{i=1}^{n} \ell(x_i, y_i)$$

λ: Hyper parameter — n: Training pairs

Training / Learning $\min_{w,b} C(w, b)$

$\downarrow$

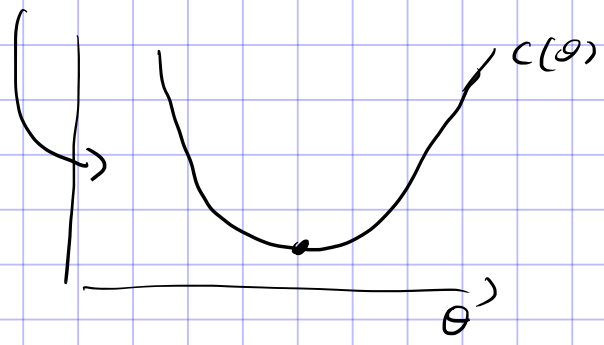
- max margin $\frac{2}{\|w\|}$

↳ min $\|w\|$

- minimize the loss

λ Hyperparameter To model the Trade off

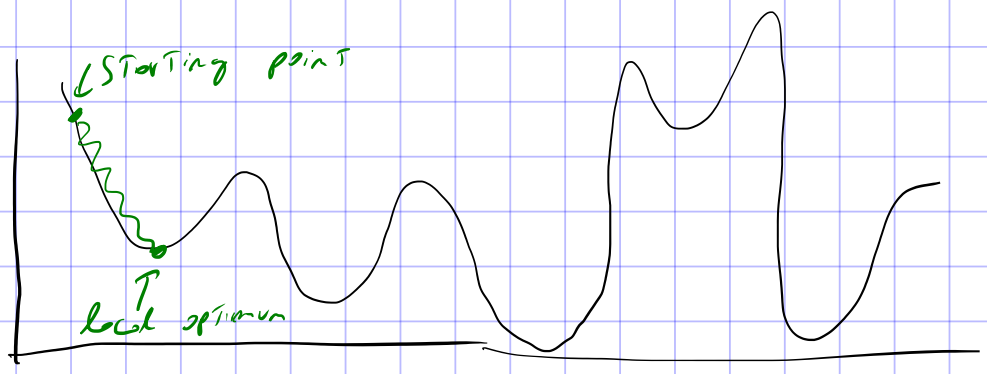
Convex optimization problem



$\theta = (w, b)$

Algorithm converges To global optimum

in Neural Networks



NON LINEAR SUM ≡ LINEAR SUM in FEATURE SPACE

$x \rightarrow \phi(x)$ Feature map

$y_i(w^T x_i - b) \geq 1 \longrightarrow y_i(w^T \phi(x_i) - b) \geq 1$

$loss(x_i, y_i) = \max\{0, 1 - y_i(w^T \phi(x_i) - b)\}$

Problem $\phi(x)$ can live in a very large space (infinite dimensional)

$dimension(w) = dimension(\phi(x))$

Dual Formulation $w \rightarrow \alpha$

$$\max_{\alpha_i \geq 0} \sum_{i=1}^{\tilde{}} \alpha_i - \sum_{i,j=1}^{\tilde{}} \alpha_i \alpha_j K(x_i, x_j) - \frac{1}{2\lambda} \sum_i \alpha_i^2$$

↑ much simpler

Primal Formulation

$$\min_{w,b} \left[\lambda \|w\| + \sum_{i=1}^{\tilde{}} l(x_i, y_i) \right]$$

α_i $\quad i = 1 \ldots N$ $\quad$ # Training Data

w_i $\quad i = 1 \ldots d_f$ $\quad d_f$ dimension of feature space (can be Huge)

Dual Formulation is independent on the dimension of feature space

Training

	primal	Dual
Time	$O(N)$	$O(N^2)$
Tuneable	$\phi(x)$	$K(x_i, x_j)$

$$K(x, x') = \phi(x) \cdot \phi(x')$$

Predictions

primal : Check the side of the plane

decision Hyperplane $\quad w^T \phi(x) - b = 0$

$$y_{predicted} = \text{sign}\left(w^T \phi(x) - b \right) = \pm 1$$

solutions to primal problem

Dual formulation $\quad w_{optimal} = \sum_i \alpha_i y_i \phi(x_i)$

Pairs in The Training set

$$y_{predicted} = sign\left(\sum_{i=1}^{\tilde{}} \alpha_i y_i \, K(x_i, x) \right)$$

learnt during Training

New data point

"Kernel Expansion of The Data"

Both For Training / Testing

No calculation in Feature space aside from Computing The Kernel

General Result : Representer Theorem

$$f_{opt}(x) = \underset{f}{\operatorname{argmin}} \left[\sum_i loss(x_i, y_i) + \text{regularization} \right]$$

$\lambda \|f\|^2$

$$= \sum_{i=1}^{\tilde{}} y_i \alpha_i \, K(x_i, x)$$

in classical sum, No calculations in Feature space

→ Choose Kernel

Example $K(x, x') = \exp\left(- \frac{\|x - x'\|^2}{2\sigma^2} \right)$

QUANTUM SVM

Quantum Computers CAN compute efficient kernel evaluations in large Hilbert spaces

Directly work with Feature map

Choice of Feature map

1) $x \longrightarrow |\psi(x)\rangle$

2) $x \longrightarrow |\psi(x)\rangle\langle\psi(x)| = \phi(x)$

↑ More natural

Use (2)

$$k(x, x') = \phi(x) \cdot \phi(x')$$

inner product between operators

Hilbert-Schmidt product

$$A \cdot B = \mathrm{Tr}(A^\dagger B)$$

$$k(x, x') = \mathrm{Tr}\left(|\psi(x)\rangle\langle\psi(x)| \; |\psi(x')\rangle\langle\psi(x')| \right)$$

$$= |\langle\psi(x)|\psi(x')\rangle|^2 \in \mathbb{R}$$

Representer Theorem

$$f_{opt}(x) = \sum_i \alpha_i y_i K(x_i, x)$$

$$= \sum_i \alpha_i y_i \, \mathrm{Tr}\left[|\psi(x_i)\rangle\langle\psi(x_i)| \, |\psi(x)\rangle\langle\psi(x)| \right]$$

$$= \langle \psi(x) | M | \psi(x) \rangle$$

$$M = \sum_i \alpha_i y_i \, |\psi(x_i)\rangle\langle\psi(x_i)|$$

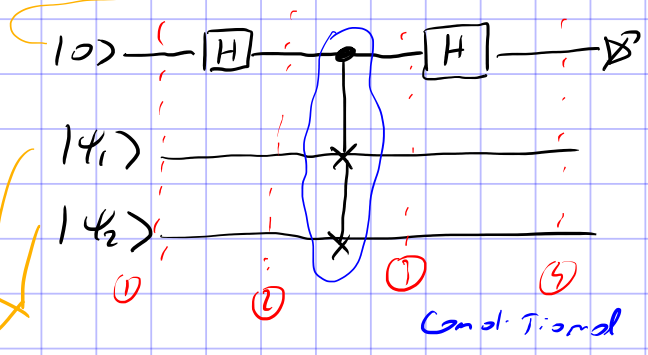
(M: measurable observable)

α_i: optimized during Training

QUANTUM CIRCUITS FOR COMPUTING THE KERNEL

1) SWAP TEST

2 registers + 2 ancilla



Two registers (possibly with many qubits)

① $|0\rangle|\psi_1\rangle|\psi_2\rangle$

② $\left(|0\rangle|\psi_1\rangle|\psi_2\rangle + |1\rangle|\psi_1\rangle|\psi_2\rangle \right)/\sqrt{2}$

③ $\left(|0\rangle|\psi_1\rangle|\psi_2\rangle + |1\rangle|\psi_2\rangle|\psi_1\rangle \right)/\sqrt{2}$ (SWAPPED)

⑤ $\left((|0\rangle+|1\rangle)|\psi_1\rangle|\psi_2\rangle + (|0\rangle-|1\rangle)|\psi_2\rangle|\psi_1\rangle \right)/2$

$$= |0\rangle \left(\frac{|\psi_1\psi_2\rangle + |\psi_2\psi_1\rangle}{2} \right) + |1\rangle \left(\frac{|\psi_1\psi_2\rangle - |\psi_2\psi_1\rangle}{2} \right)$$

$$P_0 = \left\| \frac{|\psi_1\psi_2\rangle + |\psi_2\psi_1\rangle}{2} \right\|^2 = \frac{1}{4}\left(1 + 1 + \langle\psi_1\psi_2|\psi_2\psi_1\rangle + \langle\psi_2\psi_1|\psi_1\psi_2\rangle \right)$$

$$= \frac{1}{2}\left(1 + |\langle\psi_1|\psi_2\rangle|^2 \right)$$

$$P_1 = \left\| \frac{|\psi_1\psi_2\rangle - |\psi_2\psi_1\rangle}{2} \right\|^2 = \frac{1}{2}\left(1 - |\langle\psi_1|\psi_2\rangle|^2 \right)$$

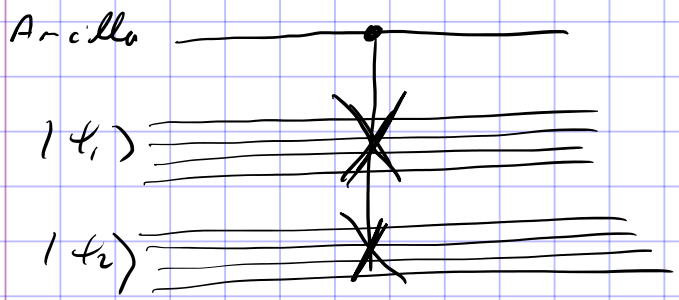
$$P_0 + P_1 = 1$$

$$|\langle\psi_1|\psi_2\rangle| = P_0 - P_1 = 2P_0 - 1$$

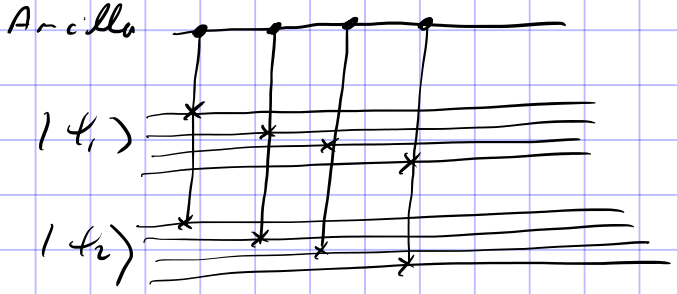
For many qubits

$$C\text{-SWAP} = \prod_{i=1}^{n} C\text{-SWAP}_{i,i+n}$$

3 qubit Fredkin?



=



C SWAP$_{i,i+n}$ 5 CNOTs + single qubit

2) SWAP MEASUREMENT

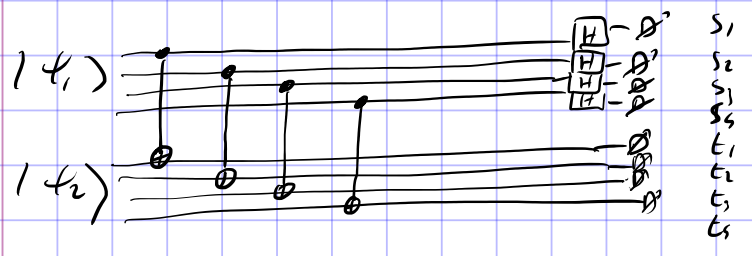
$$|\langle \psi_1 | \psi_2 \rangle|^2 = \langle \psi_1 \psi_2 | \psi_2 \psi_1 \rangle = \langle \psi_1 \psi_2 | SWAP | \psi_1 \psi_2 \rangle$$

$$SWAP = \prod_{i,i+n} SWAP_{i,i+n}$$

$$SWAP_{i,i+n} = U \lambda U^\dagger$$

$$\lambda = (-1)^{s_i t_i} |s_i t_i\rangle\langle s_i t_i|$$

$$U = CNOT \cdot H_i$$



$$\prod_{i=1} (-1)^{s_i t_i} = (-1)^{\sum s_i t_i}$$

$$|\langle \psi_1 | \psi_2 \rangle|^2 = \mathbb{E}\left((-1)^{\sum s_i t_i} \right)$$

average over measurement shots

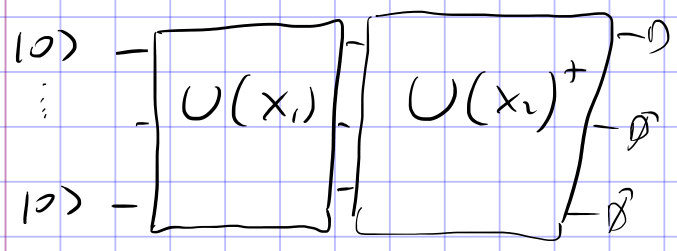
3) DIRECT FIDELITY ESTIMATION

$$|\psi_1\rangle = U(x_1) |0 \cdots 0\rangle$$

$$|\psi_2\rangle = U(x_2) |0 \cdots 0\rangle$$

$$K(x_1, x_2) = |\langle \psi_1 | \psi_2 \rangle|^2 = |\langle 0 \cdots 0 | U(x_1)^\dagger U(x_2) | 0 \cdots 0 \rangle|^2$$

$$= \langle 0 \cdots 0 | U(x_1)^\dagger U(x_2) | 0 \cdots 0 \rangle \langle 0 \cdots 0 | U(x_2)^\dagger U(x') | 0 \cdots 0 \rangle$$



Extract probability $P_{0\ldots0}$ → probability to get all outcomes zero

Advantage: single register

Disadvantage: - Higher depth
- $P_{0\ldots0}$ might be small

SUMMARY: Quantum Hybrid Kernel Methods Algorithms

Classical Kernel Training

Kernels estimate from Quantum Computer

Article | Published: 12 July 2021

A rigorous and robust quantum speed-up in supervised machine learning

Yunchao Liu, Srinivasan Arunachalam & Kristan Temme

Nature Physics **17**, 1013–1017 (2021) | Cite this article

There Exist a Dataset where Quantum Kernel Methods can offer Exponential Advantage

Quantum Physics

[Submitted on 5 Oct 2020 (v1), last revised 30 Nov 2020 (this version, v2)]

A rigorous and robust quantum speed-up in supervised machine learning

Yunchao Liu, Srinivasan Arunachalam, Kristan Temme

Over the past few years several quantum machine learning algorithms were proposed that promise quantum speed-ups over their classical counterparts. Most of these learning algorithms either assume quantum access to data -- making it unclear if quantum speed-ups still exist without making these strong assumptions, or are heuristic in nature with no provable advantage over classical algorithms. In this paper, we establish a rigorous quantum speed-up for supervised classification using a general-purpose quantum learning algorithm that only requires classical access to data. Our quantum classifier is a conventional support vector machine that uses a fault-tolerant quantum computer to estimate a kernel function. Data samples are mapped to a quantum feature space and the kernel entries can be estimated as the transition amplitude of a quantum circuit. We construct a family of datasets and show that no classical learner can classify the data inverse-polynomially better than random guessing, assuming the widely-believed hardness of the discrete logarithm problem. Meanwhile, the quantum classifier achieves high accuracy and is robust against additive errors in the kernel entries that arise from finite sampling statistics.

Comments: 27 pages, 2 figures
Subjects: **Quantum Physics (quant-ph)**; Machine Learning (cs.LG)
Cite as: arXiv:2010.02174 **[quant-ph]**
(or arXiv:2010.02174v2 **[quant-ph]** for this version)
https://doi.org/10.48550/arXiv.2010.02174
Journal reference: Nature Physics, 2021
Related DOI: https://doi.org/10.1038/s41567-021-01287-z

Advantages : String Theory, global optimum

Disadvantages

Compute Kernel

$O(N^2)$

↓

Neural Networks

$O(N)$

→ Quantum String No back propagation.

QNN	$O(NP)$	P # parameters
QUM	$O(N^2)$	N # Training Data

3 Noise characterization and mitigation -28/06/2025

NOISE CERTIFICATION & CHARACTERIZATION

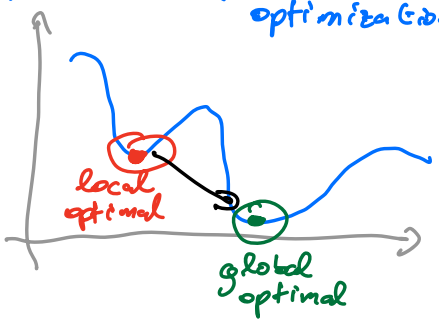
noise mitigation

master executive quantum machine learning 2025

Motivation

Quantum computing promises breakthrough in a variety of domains = drastic computational advantage

- material simulation (Deutsch)
- pharma discovery
- optimization problems (finance: portfolio optimization)



Shor's algorithm (1994)
(integer factorization)

$N = N_1 \cdot N_2$ (N_1, N_2: prime number)

- N: public key
- N_1, N_2: private keys

→ asymmetric cryptography

Problem: these algorithm are in principle meant to be carried on ideal, noiseless quantum computers ✗ reality

"noise" is present in every quantum computation

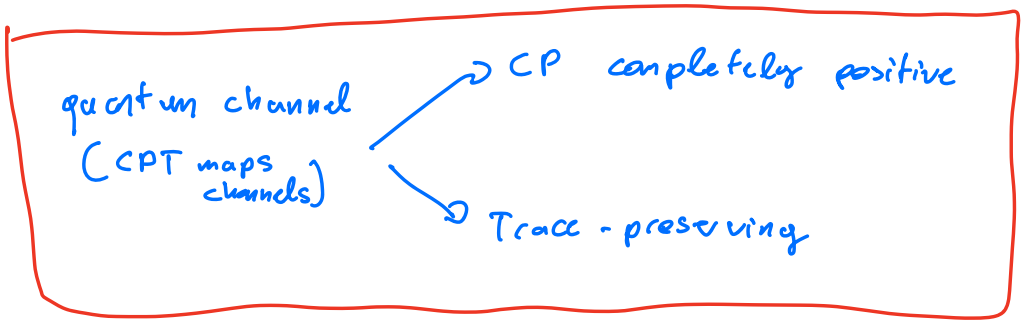
sources: "(thermal)" environment + noise from device itself

Def (noise) (i) unwanted effects changing the quantum state (where computation/information is encoded) during the process

(ii) difference between ideal vs experimentally implemented quantum gate/operation

(*) Typical noise examples

requisity: noise is modeled as a proper physical quantum channel



$\mathcal{E}$ positive

$\rho \geq 0 \qquad \mathcal{E}(\rho) \geq 0$

⚠ there can be positive channels

$$C \otimes \tilde{I}\left(\underset{\forall}{\tilde{\rho}}\right) < 0$$

$\forall \rho \geq 0$

④ Completely positive channels $\mathcal{E}$ $(\rho \geq 0)$

$\mathcal{E} \otimes \mathcal{I}_m(\rho) \geq 0$ for every m ≤ 1

② Trace preservation of $\mathcal{E}$ $\mathrm{Tr}[\mathcal{E}(\rho)] = \mathrm{Tr}[\rho] = 1$

$$\rho = \begin{pmatrix} p_1 & & & \\ & p_2 & & \\ & & \ddots & \\ & & & p_n \end{pmatrix} \qquad p_j \geq 0 \qquad |\psi_j\rangle$$

$$\mathcal{E} \otimes \mathbb{I}(\rho) = \sigma = \begin{pmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_m \end{pmatrix} \quad q_1 q_2 \cdots q_m \geq 0 \quad m \geq n \qquad |\phi_j\rangle$$

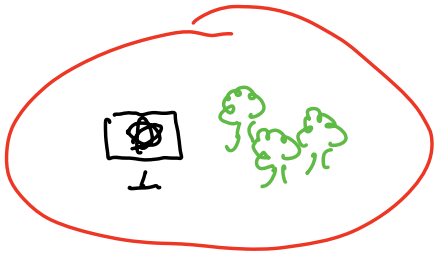
$$\mathrm{Tr}[\mathcal{E}(\rho)] = \sum_{j=1}^{m} q_j = \sum_{j=1}^{n} p_j$$

Define quantum channels (Kraus operators $\{E_j\}_j$)

$$\mathcal{E}(\rho) = \sum_{j=1}^{R} E_j \rho E_j^\dagger \qquad \text{such that} \quad \sum_j E_j^\dagger E_j = \mathbb{1}$$

R = "Kraus Rank"

$U \rho U^\dagger \qquad E_1 = U \qquad U^\dagger U = U U^\dagger = \mathbb{1}$



Examples

(1) Depolarizing noise

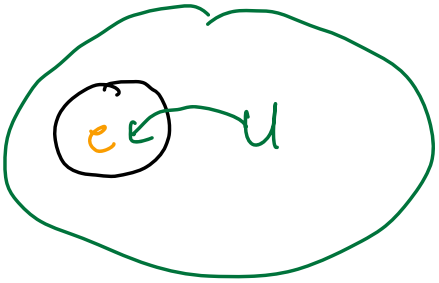
$$\mathcal{E}_{dep}(\rho) = (1-p)\rho + p\frac{\mathbb{1}}{d} \qquad \frac{\mathbb{1}}{d} = \begin{pmatrix} 1/d & & & \\ & 1/d & & \\ & & \ddots & \\ & & & 1/d \end{pmatrix}$$

depolarization strength $0 \leq p \leq 1$

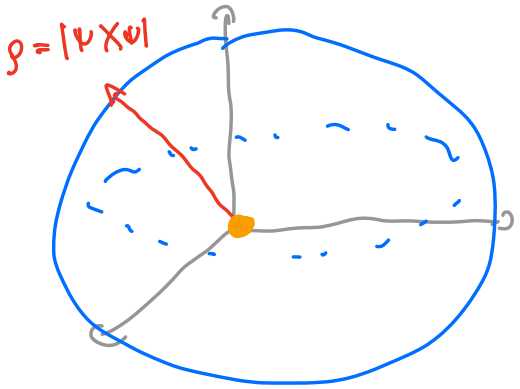
Alternatively to Kraus representation,

Stirling dilation Thm

$$\mathcal{E} = \mathrm{Tr}_R\left[U\,\rho\otimes\sigma\,U^\dagger\right]$$



2 axiom of QM: every evolution in a closed system is unitary



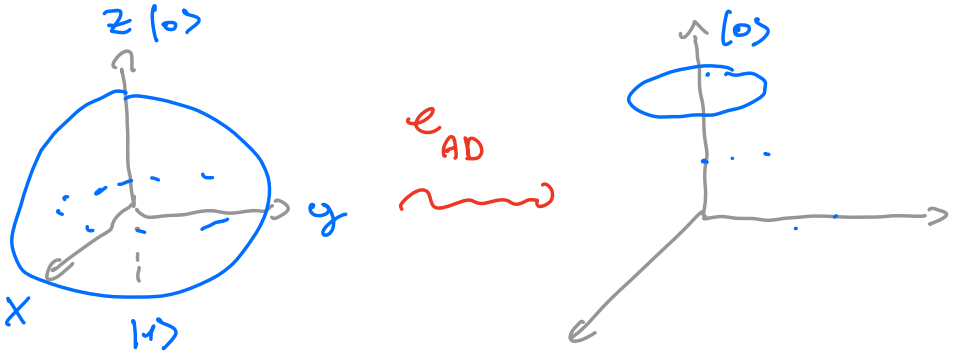
(2) Amplitude damping

$$\mathcal{E}_{AD} = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger \quad \text{with} \quad E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}$$

$$E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}$$

effect: leave $|0\rangle$ unchanged, but reduces the amplitude of $|1\rangle$ to a factor of γ

$\rightsquigarrow$ the quantum system is skewed towards $|0\rangle$



(3) bit flip / phase flip: X gate

$$\mathcal{E}_{Bf}(\rho) = (1-p)\rho + p X\rho X \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\mathcal{E}_{pf}(\rho) = (1-p)\rho + p Z\rho Z \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$X \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} X = \begin{pmatrix} \beta & 0 \\ 0 & \alpha \end{pmatrix}$$

$$X|\psi\rangle = X \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \alpha_2 \\ \alpha_1 \end{pmatrix}$$

(-) How to access / certify / understand noise

↝ most direct method "quantum process tomography"

Notation: when referring to noise channel, we use Λ (unknown)

reconstructing the noise channel Λ entirely

Step 1: choose a set of pure quantum states

$$|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_{d^2}\rangle$$

d = dimension of our Hilbert space

such that this is a basis of the set of matrices

$d = 2^n$

"tomographically complete set of states"

$n =$ ## qubits

Step 2: apply Λ on each density operator $|\psi_j\rangle\langle\psi_j|$ separately

$$\Lambda(|\psi_j\rangle\langle\psi_j|) \quad \text{for all } j = 1, \ldots, d^2$$

Step 3: perform quantum state tomography (// measuring in complete basis) on each resulting output

→ reconstruct a matrix

$$\Lambda_{jn} = \mathrm{Tr}\left[|\tilde{\psi}_j\rangle\langle\tilde{\psi}_j| \Lambda \left(|\tilde{\psi}_n\rangle\langle\tilde{\psi}_n|\right)\right] =$$

$$\langle\tilde{\psi}_j| \Lambda\left(|\tilde{\psi}_n\rangle\langle\tilde{\psi}_n|\right) |\tilde{\psi}_j\rangle \qquad j = 1, \ldots, d^2 \quad n = 1, \ldots, d^2$$

Λ_{jn} → $d^2 \times d^2$ matrix

$$\Lambda(\rho) = \chi \rho \chi$$

$d^2 \times d^2$

$\rho = d$

→ $\chi = d \times d$

$j, n = 1, \ldots, d^2$

→ quantum process tomography delivers a **complete** information about the noise channel Λ

② SPAM errors

(state preparation and measurement errors)

④ Scalability $d^2 \times d^2 = d^4$ operations

$d = 2^n$ $\quad n =$ ## qubits

→ $d^4 = 2^{4n} = 16^n$

3 qubits $16^3 = 4'096$

Addressing scalability : Randomized Benchmarking

References:

(1) Emerson, Alicki, Życzkowski
scalable noise estimation with random unitary gates
(2005)

(2) Magesan, Gambetta, Emerson
"Characterizing quantum gates via randomized benchmarking"
(2012)

(3) Helsen, Roth, Onorati, Werner, Eisert
"A general framework for randized benchmarking"
(2022)

Setting of RB with Clifford gates

Def (Pauli group)

$P_1 = \langle \{-1, 1, i, -i\} \times \{\mathbb{1}, X, Y, Z\} \rangle$ 1-qubit Pauli group

$P_n = \{P_1\}^{\otimes n}$

$\Uparrow$

$X \otimes Y \otimes Y \otimes \cdots \otimes Z$

n tensor product

Def (Clifford group)

$C_n := \{U \text{ unitary} : U \Theta U^\dagger \subset P_n \text{ for all } \Theta \subset P_n\}$

"the Clifford group is the normalizer of the n-qubit Pauli group"

$U_1, U_2 \in C_n \Rightarrow U_1 U_2 \in C_n$

$U_1 U_2 \Theta U_2^\dagger U_1 = U_1 \Theta' U_1 \subset P_1$

$\Theta' \in P_n$

Dimension of the Pauli group

$n =$	1	2	3	4
	24	11'520	$\sim$ 92 millions	$> 12 \cdot 10^{12}$

Why do we consider the Clifford group?

because of the twirling property:

$$\frac{1}{|\mathcal{C}|}\sum_{g\in\mathcal{C}} g\circ\Lambda\circ g^\dagger(\rho) = (1-p)\rho + p\frac{\mathbb{1}}{d}$$

$$= \Lambda^{dep}(\rho)$$

p is the "average gate fidelity of Λ"

$$F(\Lambda) = \int_{\text{pure states } \psi} \mathrm{Tr}\left[|\psi\rangle\langle\psi|\,\Lambda(|\psi\rangle\langle\psi|)\right] d\mu(\psi)$$

Ex - cursus

universal quantum computation = reproduce every possible unitary gate/evolution

Why the Clifford group is important?

Clifford group + one single non-Clifford gate (whatever) = universal set

Assumption: we implement ideal Clifford gates $g\in\mathcal{C}$

$g^2 = \mathbb{1}\circ g$ for all g

"gate independent noise assumption

$\tilde{g} = \Lambda \circ g$

Random procedure for RB

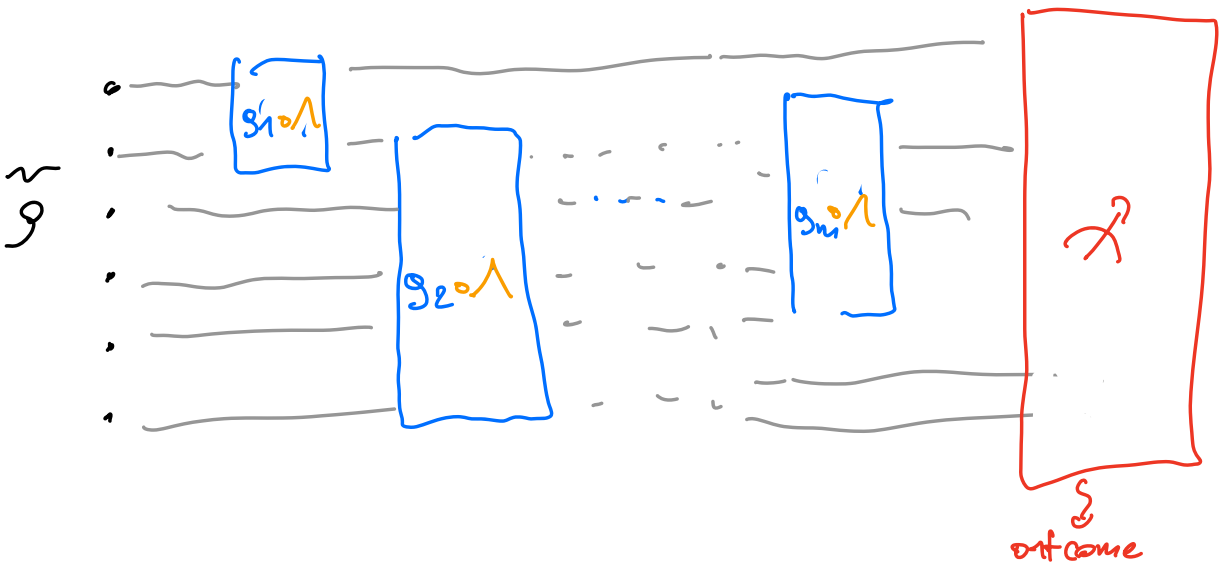
"build random quantum circuits, measure the outcome to obtain the average gate fidelity of the Clifford group"

(1) set a length for the circuit, m

(2) Generate a random sequence of Clifford gates

$g_1, g_2, \ldots, g_m$

(3) Implement (experimentally) the random sequence as quantum circuit and measure the final state



(4) repeat steps 2-3 for multiple random sequences for fixed m

$\rightsquigarrow$ compute the "measurement average"

$$Q_{\substack{avg \\ samples}}(m) = \frac{1}{\#samples} \sum_{\substack{j \\ \text{measurements outcomes} \\ \text{of random circuit } j}}$$

(5) increase the sequence length m and repeat the above

$\rightsquigarrow$ $Q_{avg}(m)$, $Q_{avg}(m+1)$, ..., $Q_{avg}(m+M)$

(6) fit the data $\{Q_{avg}(m)\}_m$ into a fitting model

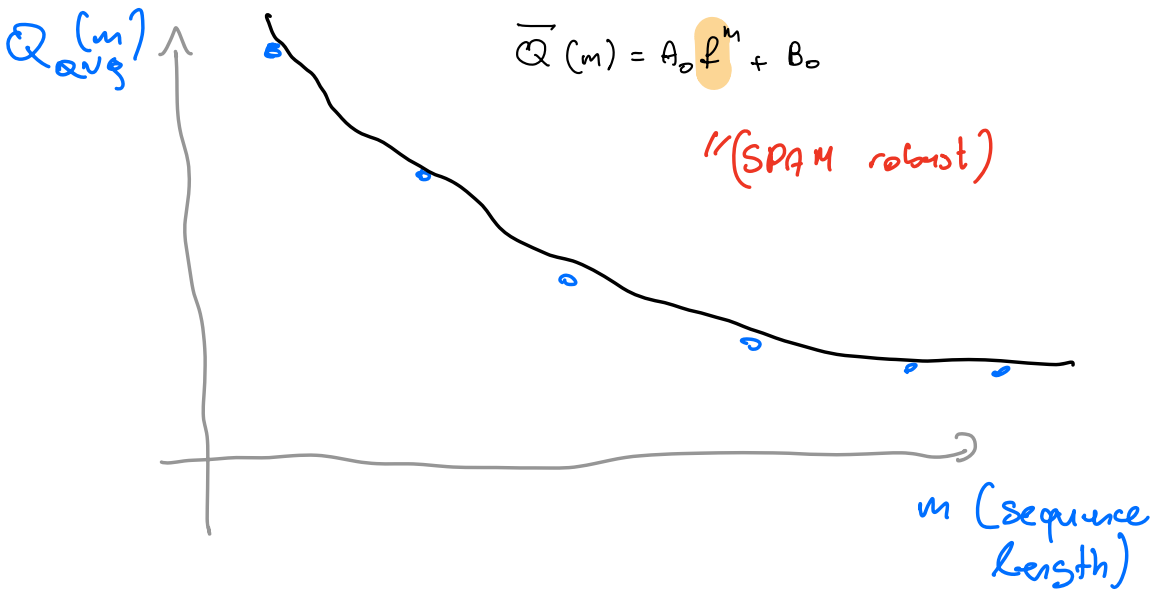
$$\overline{Q}(m) = A_0 f^m + B_0$$

to extract the parameter f which is connected to the average gate fidelity $F(\Lambda)$ of Λ

$$F(\Lambda) = f + \frac{(1-f)}{d}$$

↑

this is our final target quantity telling us how close is Λ to identity (or $g \neq \tilde{g}$)



Theoretical guarantee of RB

assume we take N samples with

$N = poly(n)$

$Q_{average}(m,n) \simeq \overline{Q}_{avg}(m)$

↑ theoretical mean over all possible realizations

How can we use information we retrieved about noise to improve / correct quantum computation?

→ improve hardware

error-mitigation

(classically)

+

some more quantum experiments

quantum error correction

(happening / affecting the quantum process itself)

simple example: bit flip error channel

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

trick redundancy

encode $|\psi\rangle$ into $|\phi\rangle = \alpha|000\rangle + \beta|111\rangle$

and assume $\mathcal{E}_{bf} = p\, X|\phi\rangle\langle\phi|X + (1-p)\rho$

assume $p \ll 1$, so that only one qubit at most is flipped per operation

strategy: use Projections or leverage "majority vote"

$$P_0 = |000\rangle\langle 000| + |111\rangle\langle 111| \qquad P^2 = P$$

$$P_1 = |100\rangle\langle 100| + |011\rangle\langle 011|$$

$$P_2 = |010\rangle\langle 010| + |101\rangle\langle 101|$$

$$P_3 = |001\rangle\langle 001| + |110\rangle\langle 110|$$

pauli X on qubit $j=1,2,3$

$$\mathcal{E}_{corr} = P_0 \rho P_0 + \sum_{j=1}^{3} X_j P_j \rho P_j X_j$$

$\psi = |0\rangle$

$\rightarrow |\phi\rangle = |000\rangle$

bit-flip error channel

$|\psi\rangle = |010\rangle$

$$\mathcal{E}_{corr}(|\psi\rangle\langle\psi|)$$

$$= \mathcal{E}_{corr}(|010\rangle\langle 010|) =$$

$$= \cancel{P_0 |010\rangle\langle 010| P_0} + \quad (=0)$$

$$\cancel{X_1 P_1 |010\rangle\langle 010| P_1 X_1} + \quad (=0)$$

$$+ X_2 P_2 |010\rangle\langle 010| P_2 X_2$$

$$+ \cancel{X_3 P_3 (|010\rangle\langle 010|) P_3 X_3} \quad (=0)$$

$$= 11 \otimes X \otimes 11 \left(|010\rangle\langle 010| + |101\rangle\langle 101|\right) |010\rangle\langle 010| \times$$

$$\left(|010\rangle\langle 010| + |101\rangle\langle 101|\right) 11 \otimes X_2 \otimes 11$$

$$= 11 \otimes X \otimes 11 \left(|010\rangle\langle 010|\right) 11 \otimes X \otimes 11$$

$=1$

$\neq 0$

$X|0\rangle = |1\rangle \quad X|1\rangle = |0\rangle$

$\neq |000X000|$

Exercise

$\mathcal{E}_{corr}(|100X100|) = |000X000|$

$\mathcal{E}(|001X001|) = |000X000|$

$\mathcal{E}(|000X000|) = |000X000|$

$|110\rangle \rightarrow |000\rangle$

Observation: overhead

physical vs logical qubits (a single logical error-corrected qubit)

3

Morning recap

1. Short overview about quantum platforms
2. noise concept + examples (depolarising, AD, BF)
   quantum channels "CPT maps"
3. Quantum process tomography
   - full information
   - SPAM-sensitive
   - non-scalable

Def

(4) "RB" // random circuits for Clifford gate

+ one single non Clifford = universality

we learn the "average gate fidelity" of the gate set

sample average (poly(n)) vs theoretical average (unfeasible)

(5) noise mitigation vs error correction

this afternoon

correcting single bit-flip error thanks to redundancy

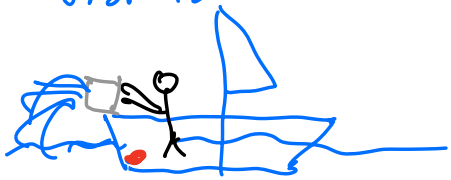
logical error-corrected qubits vs physical qubits

End goal of QC : Fault-tolerant quantum computer

"holy grail of QC"

= "QC that can carry out arbitrarily long and precise quantum computation"

"error is corrected faster than it is generated"



Aharonov, Ben-or

"Fault-tolerant quantum computation with constant error rate" '99

Threshold theorem for fault-tolerant QC

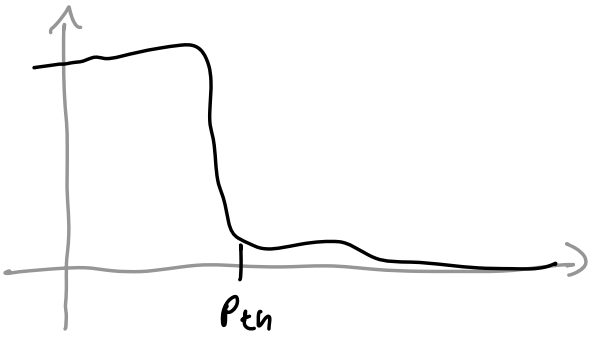
A quantum circuit containing $p(n)$ gates may be simulated with probability error at most ε using

$$\mathcal{O}\left(\text{poly}\left(\log \frac{p(n)}{\varepsilon}\right) p(n)\right)$$

gates on hardware whose components fails with probability at most $p_{th} < 10^{-5} \sim 10^{-6}$

10^{-3} 2-qubit gates

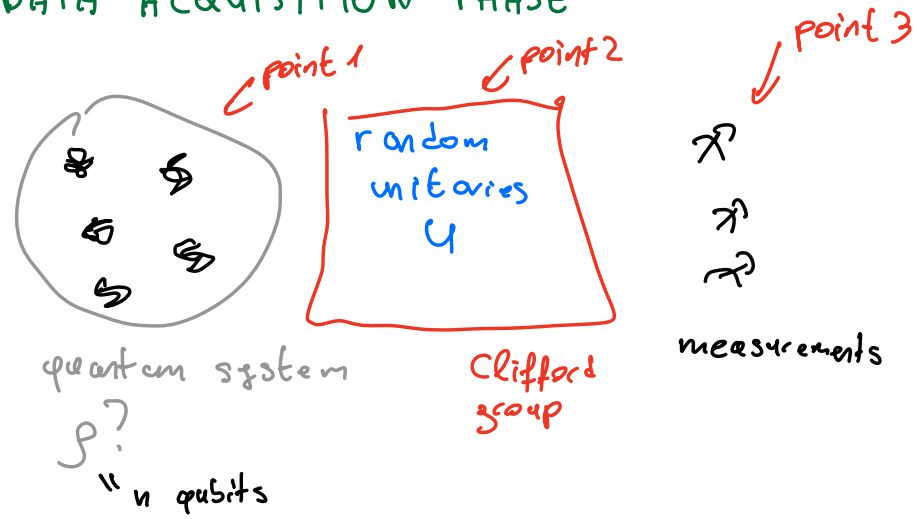
10^{-4} - 10^{-5} 1-qubits



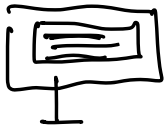
SHADOW ESTIMATION

Reference: "Predicting many properties of a quantum system from very few measurements"
Huang, Richard Küng, John Preskill
Nature 2020

DATA ACQUISITION PHASE



CLASSICAL POST-PROCESSING PHASE



$\langle\psi|O|\psi\rangle$ we are able to prepare it many times

$\langle\cdot\rangle_\sigma$

Goal: efficiently estimate $\mathrm{Tr}[O_i \rho]$ for unknown ρ
for many $\{O_i\}_i$ at once

$\mathrm{Tr}[\sigma_x |\psi\rangle\langle\psi|] = \langle\psi|\sigma_x|\psi\rangle$

Protocol : (1) prepare ρ (n-qubit states)

(2) randomly select a Clifford gate U and rotate ρ

$U \rho U^\dagger$

(3) Measure in computational basis $\{b\}$ $|0\ldots0\rangle$ $|0101\ldots\rangle$ $|0\ldots1\rangle$

$\rightsquigarrow$ obtain an outcome b' $\quad \Pr[b=b'] = \langle b'|U\rho U^\dagger|b'\rangle$

Quantum

(4) Classically reconstruct $U^\dagger|b'\rangle\langle b'|U$

(5) store efficiently the classical shadow

$$\hat{\rho} = \mathcal{M}^{-1}\left(U^\dagger|b'\rangle\langle b'|U\right)$$

frame operator (important!)

consisting in the weighted average of the process

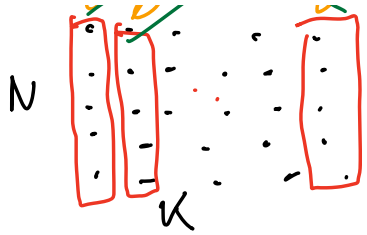
$$\mathcal{M}(\rho) = \mathbb{E}_{U\sim\text{Clifford}} \sum_{b\in\{0,1\}^{xn}} \langle b|U\rho U^\dagger|b\rangle\, U^\dagger|b'\rangle\langle b'|U$$

* later : we will see how to reconstruct the inverse easily

PREDICTION PHASE

(A) repeat the data acquisition phase $R = N \cdot K$ times

$\rightsquigarrow \{\hat{\rho}_{j,k}\}_{\substack{j=1,\dots,N \\ k=1,\dots,K}}$

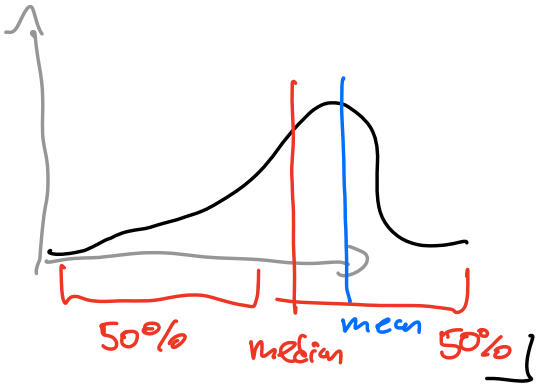


(B) Prepare the mean for K different sets of N data points each

$$\hat{\sigma}_i(k) = \frac{1}{N} \sum_{j=1}^{N} \mathrm{Tr}\left[\mathcal{O}_i \hat{\rho}_{j,k}\right] \qquad k = 1, \dots, K$$

(C) Estimate the observable $\mathcal{O}_i$ by a "median of means"

$$\hat{\sigma}_i(N,K) = \text{median}\left\{\sigma_i(1), \sigma_i(2), \dots, \sigma_i(K)\right\}$$



(D) repeat it for all wished $\{\mathcal{O}_i\}_{i=1}^{M}$ observables

Theorem (Assessment prediction guarantee for shadow estimation)

Set accuracy parameters ϵ and $\delta \in [0,1]$

Set $K = 2\log\left(\frac{2M}{\delta}\right)$ and $N = \frac{34}{\epsilon^2} \max_i \left\| O_i - \frac{\mathrm{Tr}[O_i]}{2^n} \right\|_{\text{shadow}}$

shadow norm

Then a collection of $N \cdot K$ independent classical shadows accurately estimates

$$\left| \hat{o}_i(N,K) - \mathrm{Tr}[O_i \rho] \right| \leq \epsilon \quad \text{for all } i = 1, \ldots, M$$

target

with probability at least $1-\delta$.

$$\|A\|_{\text{shadow}} = \max_{\sigma} \left(\mathop{\mathbb{E}}_{U} \sum_{b} \langle b|UOU^\dagger|b\rangle \langle b|UM^{-1}(\sigma)U^\dagger|b\rangle^2 \right)^{1/2}$$

U: Clifford group

$$\left\| \sigma - \frac{\mathrm{Tr}[\sigma]}{2^n} \mathbb{1} \right\|_{\text{shadow}} \leq 3\,\mathrm{Tr}[\sigma^2]$$

for the local 1-qubit Clifford group

$$\left\| O - \frac{\mathrm{Tr}[O]}{2^n} \right\|_{\text{shadow}} \leq 4^{\text{locality}(O)} \|O\|_\infty^2$$

Interpretation: + learn exponentially many observables M of a n-qubit quantum states with $R = O(1)\log(M)\frac{1}{\epsilon^2}$ (independent of system size)

– we can learn only local observables

most scalable procedure in the system size

Theoretical guarantee comes from

Weingarten calculus
(unitary 3-designs)
Clifford group! ⇝ scaling in n disappear

unitary design
"a distribution over the unitary group which approximates the uniform (Haar) measure"

Collins, Matsumoto, Novak
"The Weingarten calculus"
2022

median-of-means
⇝ guarantees on the variance

Berrum, Valiron, Vasirani
"Random generation of combinatorial structures from a uniform distribution"
1986

Nemirovsky, Yudin
"Problem complexity and method efficiency in optimization" 1983

Computing the inverse of frame operator M
and address noise

Frame operator

$$M(\rho) = \mathbb{E}_{U \sim \text{Clifford}} \sum_{b \in \{0,1\}^{xn}} \langle b | U \rho U^\dagger | b \rangle U^\dagger |b\rangle\langle b| U$$

taking an average $\mathbb{E}$ over a group (Clifford group)

By group theory, M is symmetric w.r.t action of the group itself

$\Rightarrow$ representation theory / Schur's Lemma

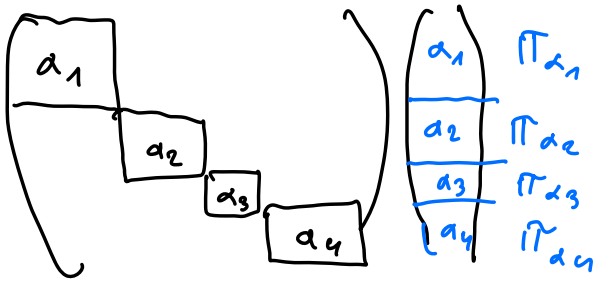
$$M = \sum_{\alpha} \frac{\mathrm{Tr}[B \Pi_\alpha]}{\mathrm{Tr}[\Pi_\alpha]} \Pi_\alpha =: \sum_\alpha f_\alpha \Pi_\alpha$$

α: irreps in the "gate representation"

irreducible representation = building blocks

representations = matrices

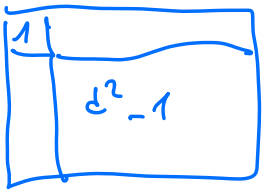
projection onto the invariant subspace of the irrep α



$$B = \sum_b |b,b\rangle\langle b,b|$$

Fact: the defining representation of a 3-design (like the Clifford group) is divided in 2 irreps only

- trivial irrep
- adjoin irrep



$|\omega\rangle\langle\omega|$ $\quad \omega = \sum_j |j,j\rangle$

$$\mathcal{M} = \Pi_{\text{trivial}} + \frac{\Pi_{\text{adj}}}{d+1}$$

$$\Pi_{\text{adj}} = \mathbb{1}_{d^2} - \Pi_{\text{trivial}}$$

$$\mathcal{M}^{-1} = \Pi_{\text{trivial}} + (d+1)\,\Pi_{\text{adj}}$$

$$\Pi^2 = \Pi$$

$$\Pi_j \Pi_n = 0$$

$$\mathcal{M}\mathcal{M}^{-1} = \left(\Pi_{\text{trivial}} + \frac{1}{d+1}\Pi_{\text{adj}}\right)\left(\Pi_{\text{trivial}} + (d+1)\,\Pi_{\text{adj}}\right)$$

$$= \Pi_{\text{trivial}}^2 + (d+1)\,\Pi_{\text{trivial}}\Pi_{\text{adj}} + \frac{1}{d+1}\Pi_{\text{adj}}\Pi_{\text{trivial}} +$$

$$+ \frac{d+1}{d+1}\Pi_{\text{adj}}^2 =$$

$$= \Pi_{\text{trivial}} + \Pi_{\text{adj}} = \Pi_{\text{trivial}} + \mathbb{1} - \Pi_{\text{trivial}} = \mathbb{1}$$

Frame operator

$$\mathcal{M}(\rho) = \mathbb{E}_{U\sim\text{Clifford}} \sum_{b\in\{0,1\}^{2n}} \langle b|\tilde{U}\rho\tilde{U}^\dagger|b\rangle\, U|b'\rangle\langle b'|U^\dagger$$

Can we calibrate the frame operator to cancel the effect of noise, like if the procedure was carried out with no noise at all?

There are different methods:

(1) "Robust shadow estimation" Chen et. al (2021)

(2) "Noise-mitigated randomized measurements and self-calibrating shadow estimation" Chirati et al (2024)

assume the noise to be gate independent

$$\tilde{g} = \Lambda \circ g \quad \text{for all } g \in \mathcal{C} \qquad \lambda(g) = \lambda$$

$$\hat{\mu} = \frac{\mathrm{Tr}[\Pi_{triv} B \Lambda]}{\mathrm{Tr}[\Pi_{triv}]} \Pi_{triv} + \frac{\mathrm{Tr}[\Pi_{adj} B \Lambda]}{\mathrm{Tr}[\Pi_{adj}]} \Pi_{adj}$$

$$= \Pi_{trivial} + \frac{\lambda_z(\Lambda)}{d+1} \Pi_{adj}$$

where we defined the $\lambda_z(\Lambda) := \frac{\mathrm{Tr}[\Pi_z \Lambda]}{d-1}$ (*)

and Π_z is the projection onto Pauli words made of identities and Pauli-Z operators only

$$\Pi_z = B \Pi_{adj} = \sum_{P \in \{1, Z\}^{\otimes n} / \mathbb{1}_n} \frac{|P\rangle\!\rangle\langle\!\langle P|}{2^n}$$

Lemma (error bias)

$$\mathcal{M}_{\text{ideal}}^{-1}\,\mathcal{M} - \mathbb{1} = \left[\Pi_{\text{trivial}} + (d+1)\,\Pi_{\text{adj}}\right]\left[\Pi_{\text{trivial}} + \frac{\lambda_z(1)}{d+1}\,\Pi_{\text{adj}}\right]$$

$$= \Pi_{\text{trivial}} + \lambda_z(1)\,\Pi_{\text{adj}} - \mathbb{1} = \left(\lambda_z(1) - 1\right)\Pi_{\text{adj}}$$

Idea to noise-mitigate shadow estimation is to substitute

$\mathcal{M}_{\text{ideal}}$ with $\hat{\mathcal{M}} = \underbrace{\hat{\lambda}_z(1)}_{\text{estimation of } \lambda_z(1)}\,\Pi_{\text{adj}} + \Pi_{\text{trivial}}$

$$\hat{\mathcal{M}}^{-1}\,\mathcal{M} - \mathbb{1} = \left(\frac{d+1}{\hat{\lambda}_z(1)}\,\Pi_{\text{adj}} + \Pi_{\text{trivial}}\right)\left(\Pi_{\text{trivial}} + \frac{\lambda_z(1)}{d+1}\,\Pi_{\text{adj}}\right)$$

$$= \Pi_{\text{trivial}} + \frac{\lambda_z(1)}{\hat{\lambda}_z(1)}\,\Pi_{\text{adj}} - \mathbb{1}$$

assume perfect estimation $\hat{\lambda}(1) = \lambda(1)$

$$= \Pi_{\text{trivial}} + \Pi_{\text{adj}} - \mathbb{1} =$$

$$= \Pi_{\text{trivial}} + \mathbb{1} - \Pi_{\text{trivial}} - \mathbb{1}$$

$$= \mathbb{1} - \mathbb{1} = 0$$

Recap: ① we assumed we were able to access the parameter $\lambda_z(1)$ by an estimate $\hat{\lambda}_z(1)$

② classically apply $\hat{\mathcal{M}}^{-1}$ instead of

the noise-unaware $\mathcal{U}_{ideal}^{-1}$

(3) Assuming our estimation is very close to the real $\lambda_z(1)$, then

$\hat{\mathcal{U}}^{-1} \mathcal{U}^{\epsilon} - \mathbb{1} \approx 0$

Goal: how to actually retrieve $\lambda_z(1)$?

(4) Ad-hoc calibration

(i) prepare $|\vec{0}\rangle = |0\rangle^{\otimes n}$, sample at random a Clifford gate

(ii) measure in computational basis $\rightarrow$ get b'

(iii) Compute classically $\hat{\lambda}_z^{(r)} = \dfrac{d\,|\langle b'|\mathcal{U}|0\rangle|^2}{d-1}$

(iv) repeat steps 1-3 for $R = N \cdot K$ rounds

(v) compute $\hat{\lambda}_z(1)$ as a median-of-means (R, N, K) (like shadow estimation)

(2) Use RB $\rightarrow$ scalable fashion
$\rightarrow$ SPAM robust

Recall: that the parameters of RB

are $\mu_\alpha = \frac{Tr[\Pi_\alpha \Lambda]}{Tr[\Pi_\alpha]}$ compared to the parameter of shadow estimation $f_\alpha = \frac{Tr[\Pi_\alpha B \Lambda]}{Tr[\Pi_\alpha]}$

(o) for depolarizing noise:

$$\mu_\alpha = f_\alpha \Rightarrow \text{perfect inversion}$$

(o) for the bit-flip noise

$$\mu_{adj} = \frac{d^2(1-p)^n - 1}{d^2 - 1} \quad \text{vs} \quad f_\alpha = \frac{d(1-p)^n - 1}{d-1}$$

mirror bias

$$\hat{\mu}^{-1}_{RB}\, \mu^\vee - \mathbb{1} = \left(\frac{d^2-1}{d^2(1-p)^n - 1} \frac{d(1-p)^n}{d-1} - 1 \right) \Pi_{adj}$$

$$= \left\{ (d+1) \frac{d(1-p)^n - 1}{d^2(1-p)^n - 1} - 1 \right\} \Pi_{adj}$$

$$\xrightarrow{d \gg 1} \left\{ \frac{d+1}{d+1} - 1 \right\} \Pi_{adj} = 0$$

(c) RB with CNOT-dihedral (substituting Clifford group)

WRAP-UP

Discussed shadow estimation

highly scalable method to learn many properties of unknown quantum states

randomization + statistical median-of-means techniques

(•) It is affected by noise, introducing an error bias

Separate RB to correct the error bias
Calibration In classical post-processing

(•) We discussed how to combine shadow estimation as classical post-processing of RB to learn more granular information about a noise channel Λ.

(•) Broader take-home message

interplay between quantum experiments and classical post-processing to push forward quantum learning / noise characterization