Quantum Machine Learning

Adrián Pérez Salinas

[1]Lorentz Institute - Leiden University
[2]$\langle aQa \rangle^L$: Applied Quantum Algorithms

Spring school in near-term quantum computing
— Benasque 2024 —

Contents

Section 1

Introduction

Concepts in machine learning

Machine learning is...

- A set of techniques to teach computers to solve tasks, without explicitly programming them
- Capable of solving a variety of problems
  - Supervised learning (classification / regression)
  - Unsupervised learning (classification)
  - Reinforcement learning (interaction with environment)
  - Generative modeling (creating samples)

In this talk we focus primarily in supervised learning

Data plays a fundamental role, but also the way to interpret it

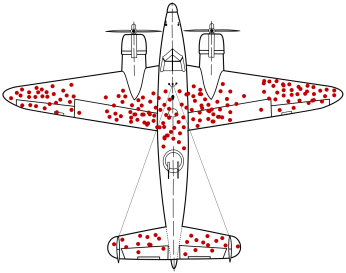


Figure: A real-world problem: in WWII the planes received these bullet shots. Where should we reinforce the planes?

Training We show the data to the model, and make the model minimize some loss function

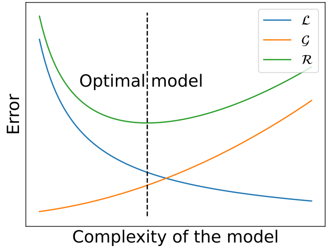
$$\text{Data} = (\mathcal{X}, \mathcal{Y}) = \{(x, y(x))\}$$

ML model provides $f_\theta(x)$
The (training) solution to the ML problem is to optimize the empiric risk

$$\mathcal{L}(\theta) = \sum_{x \in \mathcal{X}} D(f_\theta(x), y(x))$$

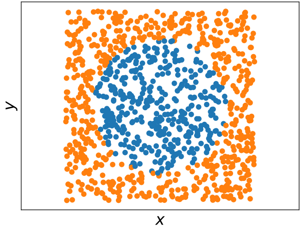
Generalization
We want to learn about a space of data, but have only limited access $\mathcal{X} \sim \mathcal{D}$
We want to minimize the true risk

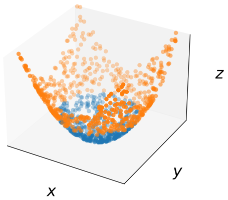
$$\mathcal{R}(\theta) = \mathbb{E}_{x \sim \mathcal{D}} D(f_\theta(x), y(x)) = \mathcal{G}(\theta) + \mathcal{L}(\theta)$$



Kernel trick

- Machine learning models are good at linearly separating data
- The kernel trick allows to linearly separate data that is non-separable otherwise
- Also used as a trick to compute distances
- This trick has been used to find quantum advantages in machine learning (more on that later)





Quantum machine learning

Quantum machines for learning classical data

- Data is produced in classical ways
- We use a quantum machine to learn it
- Examples
  - Variational algorithms
  - Discrete Logarithm Problem classification[a]
  - Linear-algebra-based machine learning

[a] Liu, Arunachalam, and Temme, "A Rigorous and Robust Quantum Speed-up in Supervised Machine Learning".

Quantum machines for learning quantum data

- Data is in quantum form (thus no loading overhead)
- Data is classical but extracted from a quantum source (e.g. properties of quantum materials)

Not covered in this lecture

Classical machines to learn about quantum data (but very interesting topic)

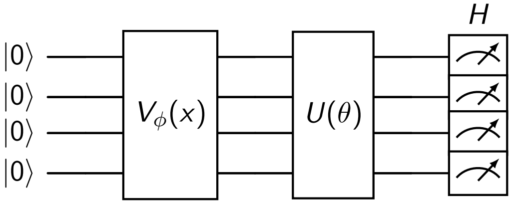
Section 2

Variational QML

Variational quantum machine learning

Machine learning where quantum computers follow variational models

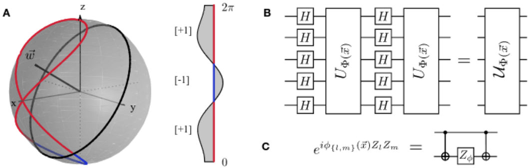
- Natural extension of VQAs to data
- Data can be understood as input state
  - If the data is classical, a feature map is required
- Optimization is done as in the case of variational algorithms

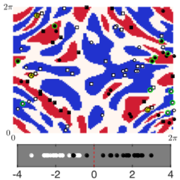


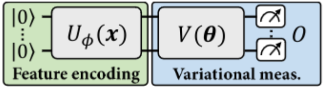
- We optimize θ
- Optionally, we can optimize the feature map

Linear models[1,2]

- Linear models are those where the feature maps appear at the beginning
- The choice of the correct feature map is crucial for the performance of the algorithm
- If $V_\phi(x)$ is fixed, performance in θ is bounded
- $U(\theta)$ can only find the optimal projection







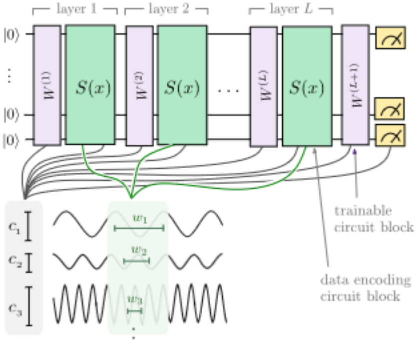
Finding feature maps...

becomes a task of utmost importance

[1] Havlíček et al., "Supervised Learning with Quantum-Enhanced Feature Spaces".
[2] Lloyd, Schuld, et al., "Quantum Embeddings for Machine Learning".

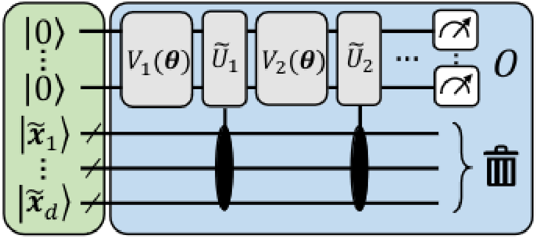
Data re-uploading[3,4,5]

- Data re-uploading introduces data several time and intersperses it with parameterized gates
- It can optimize the feature map on the fly
- Re-uploading models are universal in the output function (connection to Fourier analysis)



[3] Pérez-Salinas, Cervera-Lierta, et al., "Data Re-Uploading for a Universal Quantum Classifier".
[4] Schuld, Sweke, and Meyer, "The Effect of Data Encoding on the Expressive Power of Variational Quantum Machine Learning Models".
[5] Pérez-Salinas, López-Núñez, et al., "One Qubit as a Universal Approximant".

Data re-uploading vs. linear models



Model	Resources	
	Qubits	Data points
Re-uploading	1	$\mathcal{O}(\log d)$
Explicit	$\Omega(d)$	$\mathcal{O}(\log d)$
Implicit	$\Omega(d)$	$\Omega(2^d)$

It has ben proven that both methods are (asymptotically) equivalent

... at the expense of overhead in qubits and connectivity

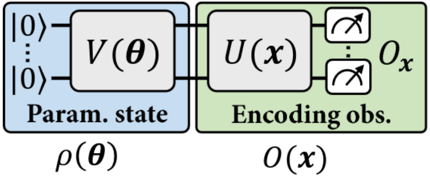
Jerbi, Fiderer, et al., "Quantum Machine Learning beyond Kernel Methods"

- Data is encoded in binary form
- These 0/1 act as control for trainable gates, which behaves as data-dependent gates
- Measurement in the work register are needed, data register is discarded

Flipped models

We have been encoding data before processing, why not turning it around?[a]

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$$



These models are more powerful than any classical model but less powerful than completely quantum models

[a] Jerbi, Gyurik, et al., *Shadows of Quantum Machine Learning*.

Properties

- The data is now in the measurement
- I can train the model with access to a quantum computer, and then measure classically
- Using classical shadows[a] we can predict any observables classically, provided a pool of information [b]

[a] H.-Y. Huang, Kueng, and Preskill, "Predicting Many Properties of a Quantum System from Very Few Measurements".

[b] Classical shadows are an amazing topic, but it will not be covered right now

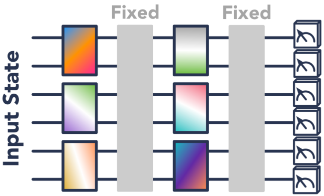
Generalization bounds

- Gen. bounds are tools to deal with generalization errors
- Gen. bounds *measure* the size of the function space
- How?
    - Create sets of functions that are *equally* equispaced
    - Measure how far these points are from every other function achievable by the model
    - Through mathematical foundations one can bound generalization errors[a]

In quantum machine learning

Even though the size of available space increases exponentially, the generalization error grows as

$$\text{gen} \in \mathcal{O}\left(\sqrt{\frac{T \log T}{N}}\right)$$



[a] Wolf, *Mathematical Foundations of Supervised Learning*.

Variational quantum machine learning

- Do not forget we are using variational methods, thus all problems from VQAs are inherited
- VQAs are a subset of variational QML, with just one data point
  (no overfitting is possible in this case)
- Generalization and optimization must be tackled independently
- Feature maps become a crucial aspect of variational QML, performance critically depends on them
- There exist methods to extend claims from variational methods to QML[6]

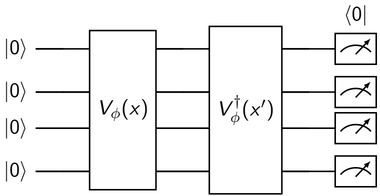
[6] Barthe and Pérez-Salinas, *Gradients and Frequency Profiles of Quantum Re-Uploading Models*.

Section 3

Kernel-based QML

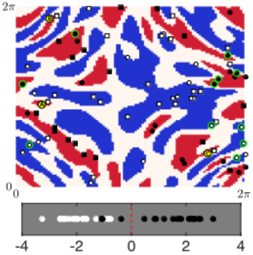
In kernel-based QML we utilize a quantum computer as a kernel

$$K(x, x') = \mathrm{Tr}\Big(V_\phi^\dagger(x') V_\phi(x) \rho_0\Big)$$



to combine with easy classical methods, e. g. support vector machine.

- Once the kernel is computed, we can perform classification by optimizing classical parameters
- If the quantum kernel is such that it performs an efficient non-classical linear separation, then we aim for quantum advantage



Discrete Logarithm Problem (DLP)-kernel

Discrete logarithm problem (DLP) Given (a, b), find k (if any) such that

$$\log_b a = k \to b^k = a$$

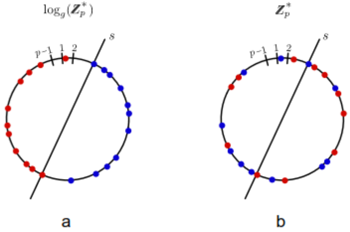
Shor's algorithm

The DLP is tightly connected to Shor's algorithm, and it is the main instance of problem that is quantumly tractable and believed to be classically impossible

ML problem[a]

$$f_s(x) = \begin{cases} 1 & \text{if} \quad \log_g x \in \left[s, s + \frac{p-3}{2}\right] \\ -1 & \text{else} \end{cases}$$

with $g \in \{1, 2, \ldots, p-1\}$ for a prime p.



[a] Liu, Arunachalam, and Temme, "A Rigorous and Robust Quantum Speed-up in Supervised Machine

DLP-kernel

Why this is interesting

- Show rigurously a case where QML surpasses classical
- The proof relies on Shor's algorithm, which is believed to be a problem in the class BQP
- Strong assumptions in complexity theory [a]

[a] I am trying to stay away from complexity theory, but in case of doubt, please do not hesitate to ask now or later!

Should I care about this?

- How relevant is this problem?
- How artificial is this problem?
- Complexity theory arguments are extremely strong, but they are usually constructed in very particular ways
- This result cannot be extended to other problems unsolvable for classical computers

Expressivity of quantum kernels

Quantum kernels for quantumly tractable problems

- Quantum kernels: kernels that are not efficient in classical computers
- For all problems that are tractable with quantum computers, there exists at least one kernel that allows for tackling a related classification problem[a]

[a] Jäger and Krems, "Universal Expressiveness of Variational Quantum Classifiers and Quantum Kernels for Support Vector Machines".

Quantum kernels for kernel functions

- For a kernel function $k(x, x')$, there always exist an embedding quantum kernel such that $k(x, x') = \langle 0| U^\dagger(x') U(x) |0\rangle$
- Under some conditions of the kernel function, the embedding quantum kernel is efficient[a].

[a] Gil-Fuster, Eisert, and Dunjko, *On the Expressivity of Embedding Quantum Kernels*.

Section 4

Linear-algebra-based QML

A linear algebra approach

> If we can invert matrices exponentially faster, then any ML task based on it has advantage

- There exist many ML algorithms consisting in inverting matrices
- There exist techniques to invert matrices with quantum computers more efficiently than with classical computers[a][b]

[a] Harrow, Hassidim, and Lloyd, "Quantum Algorithm for Solving Linear Systems of Equations".
[b] Childs, Kothari, and Somma, "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision".

HHL algorithms in a nutshell Problem to solve: $A|x\rangle = |b\rangle$,
How to:

1. Load matrix A
2. Load vector $|b\rangle$ (QRAM needed!)
3. Apply hamiltonian time evolution e^{-iAt}
4. Use Fourier Transform to extract eigevalues
5. Invert eigenvalues

Runtime: $\mathcal{O}(\log N s^2 \kappa^2/\epsilon)$
N: size, s: sparsity, κ: condition number of A, ϵ: error

Principal Component Analysis[a]
With input state $\rho = \sum \lambda_i |\psi_i\rangle \langle\psi_i|$, then

$$\text{qPCA}(\rho) = \sum_i \lambda_i |\lambda_i\rangle \langle\lambda_i| \otimes |\psi_i\rangle \langle\psi_i|$$

Quantum Support Vector Machine[b]
The SVM is given as a matrix F to invert
Using the HHL algorithm, we invert F

[a] Lloyd, Mohseni, and Rebentrost, "Quantum Principal Component Analysis".
[b] Rebentrost, Mohseni, and Lloyd, "Quantum Support Vector Machine for Big Data Classification".

Quantum recommendation systems[a]

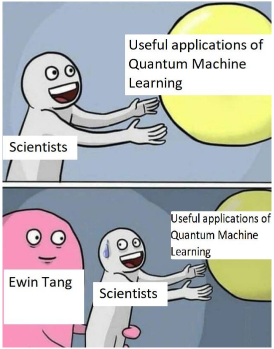
- Recommendation systems give advice to users for future purchases (e. g. Netflix, Amazon...)
- They function assuming low-rank in the recommendation matrix
- Most people belong to a pre-define types

We can invert recommendation systems leveraging the low-rank assumption

[a] Kerenidis and Prakash, *Quantum Recommendation Systems*.

Dequantization

The history of Tang



Dequantization

- It is the process of returning quantum algorithms to classical machines
- Many applications of HHL are succesfully dequantized
- After Tang, only a few applications still resist

Working principle

- Quantum algorithms are probabilistic, and thus admit some errors
- But most classical algorithms are deterministic
- Tang relaxes this condition for classical algorithms, yielding a classical version of the quantum algorithms

Quantum algorithms are still better, but only slightly[7]

[7] Tang, "A Quantum-Inspired Classical Algorithm for Recommendation Systems"; Tang, "Quantum Principal Component Analysis Only Achieves an Exponential Speedup Because of Its State Preparation Assumptions".

Section 5

Generative models

Generative models

- A generative model transforms a random number z into a sample x
- The samples x follow a probability distribution $x \sim p(x)$
- The goal is to mimic some target probability distribution $q(x)$

If $G_\theta : z \to x \sim p_\theta(x)$, then
$\theta^* = \text{argmin}_\theta \sum_x D(p_\theta(x), q(x))$

Examples:

- ChatGPT
- https://thispersondoesnotexist.com/[a]

[a] There is a Twitter account with the worst generated faces: @wedontexisthere

In quantum computing

- Exponential support
- Most experiments on quantum supremacy rely on sampling from probability distributions[a]
- Sampling is harder than simulating expectation values[b]
- Learning probability distribution is hard if they are only slightly quantum[c]

[a] Boixo et al., "Characterizing Quantum Supremacy in Near-Term Devices".

[b] Lund, Bremner, and Ralph, "Quantum Sampling Problems, BosonSampling and Quantum Supremacy".

[c] Hinsche et al., "A Single T-Gate Makes Distribution Learning Hard".

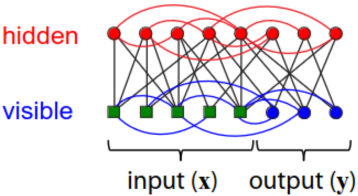
Quantum Generative models

Quantum Boltzmann machines

In this model, we aim to sample states according to the Boltzmann distribution of a Hamiltonian

$$H = \sum_i b_i Z_i + \sum_{i,j} w_{i,j} Z_i Z_j.$$

The weights b, w are tunable, to match whatever probability distribution.



Quantum Circuit Born Machines

Assume a circuit $U(\theta)$, then the QCBM is given by sampling from its output state

$$x \sim p_\theta(x) = |\langle x| U(\theta) |0\rangle|^2$$

The goal is to make $p_\theta(x)$ match some data-defined distribution[a].

If $U(\theta)$ is chosen at random, then $p_\theta(x)$ follows a Porter-Thomas distribution

$$\text{Prob}(p_\theta(x) = p) = N e^{-Np},$$

where N is the size of the support.

[a] Benedetti et al., "A Generative Modeling Approach for Benchmarking and Training Shallow Quantum Circuits".

Section 6

Learning quantum vs. classical data

Learning quantum vs. classical data

Folklore of quantum computing says:

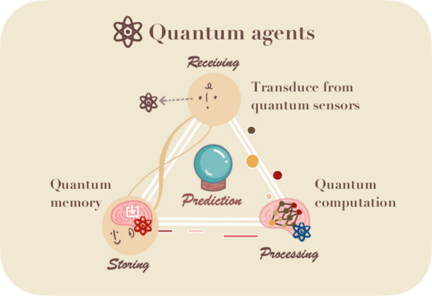
> *Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy.*

Feynman, "Simulating Physics with Computers"

For quantum machine learning

you better try to learn data coming from quantum processes to make QML advantageous

Can we obtain quantum advantage if we machine-learn physical processes?



R. Huang, *Tutorial in QTML 23'*

The power of data

Power of data

Consider $f(x) = \langle x| U^\dagger O U |x\rangle$. If $f(x)$ can be computed by a classical algorithm *without* data, then quantum computers are not more powerful than digital computers[a]

[a] BPP = BQP

With generalization error

$$\mathbb{E}_{x\in\mathcal{D}} |h(x) - f(x)| \leq c\sqrt{\frac{p^2}{N}}$$

where $c > 0$ is constant, N is the number of data points, and p is given by

$x_i = \sum_{k=1}^{p} x_i^k |k\rangle$ (i stands for data instance)

H.-Y. Huang, Broughton, et al., "Power of Data in Quantum Machine Learning"

- Quantum data is hard to generate
- But it does not mean it is hard to learn
- Therefore, classical ML models can take data coming from quantum sources, and learn it even more efficiently than quantum learners.

But this leaves an open question:

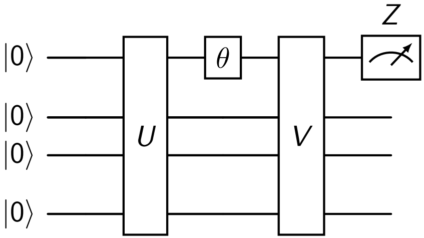
How hard is it to get data from a quantum source?

The power of data

A small flavour

- Data changes the game in quantum computing H.-Y. Huang, Broughton, et al., “Power of Data in Quantum Machine Learning”
- Quantum data *cannot* be created efficiently

An example



$\langle Z \rangle = A \sin(a\theta + b)$

Even though computing $\langle Z \rangle$ is difficult, having access to 3 sample points is enough to characterize it!

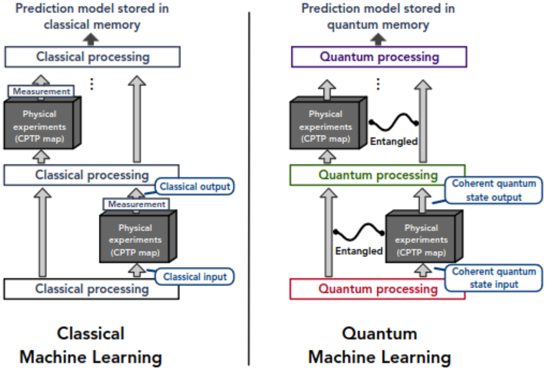
Learning quantum data

$\mathcal{E}(\cdot)$ is an unknown quantum process

$f(x) = \mathrm{Tr}(O\mathcal{E}(|x\rangle\langle x|))$

We allow for two different settings

1. Classical: sequential measurements
2. Quantum: coherent queries



Quantum advantage in machine learning

Average performance

Theorem

Assume a quantum ML model trained with N_Q samples such that

$$\sum_x \mathcal{D}(x)|h_Q(x) - \mathrm{Tr}(O\mathcal{E}(|x\rangle\langle x|))|^2 \leq \epsilon.$$

Then, there exists a classical ML model such that

$$\sum_x \mathcal{D}(x)|h_C(x) - \mathrm{Tr}(O\mathcal{E}(|x\rangle\langle x|))|^2 \leq \mathcal{O}(\epsilon),$$

trained with $N_c \in \mathcal{O}(mN_Q/\epsilon)$ samples

Worst-case performance

Consider a metric $\max_x |h_Q(x) - \mathrm{Tr}(O\mathcal{E}(|x\rangle\langle x|))|^2 \leq \epsilon$.

Quantum model

$N_Q = \mathcal{O}(\log(M)/\epsilon^4)$ copies of ρ to predict M Pauli observables with accuracy ϵ

Classical theorem

Any classical ML model must use at least $N_C \in \Omega(2^n)$ copies of ρ to predict Pauli expectation values

H.-Y. Huang, Kueng, and Preskill, *Information-Theoretic Bounds on Quantum Advantage in Machine Learning*

Quantum advantage is lost in measurement

Classical shadows

This technique allows to obtain a lot of information from a quantum state

- Select observables with bodyness up to k
- Measure the quantum state through random measurements
- Reconstruct expectation values of the observables
- The error of the approximation scales as $\log M \|\hat{O}\|_\infty^k$

Learning

- Consider a quantum state ρ
- Obtain the representation of this state in classical shadows
- There exist classical ML algorithms to predict ground states and phase transitions of Hamiltonians from data taken through classical shadows
- Any advantage in ML is lost in the measurement!

H.-Y. Huang, Kueng, Torlai, et al., "Provably Efficient Machine Learning for Quantum Many-Body Problems"

Section 7

Conclusions

Conclusions

- Quantum Machine Learning exists in many forms
- This lecture was an overview for different techniques, but it is not exhaustive
- There exist a few cases where quantum advantage has been proven
  - From kernels, without data
  - From data coming from quantum sources
- Quantum machine learning is in general universal, and theoretically as powerful as classical machine learning

Apart from the seen exceptions...

Is quantum machine learning useful for general purpose algorithms?

Quantum Machine Learning

Adrián Pérez Salinas

[1]Lorentz Institute - Leiden University
[2]$\langle aQa \rangle^L$: Applied Quantum Algorithms

Spring school in near-term quantum computing
— Benasque 2024 —