Full-Stack Quantum Machine Learning

Matteo Robbiati

28 September 2023

Introduction to concepts

Machine Learning

Machine Learning helps in solving statistical problems, such as data generation, classification, regression, forecasting, etc.

Machine Learning

Machine Learning helps in solving statistical problems, such as data generation, classification, regression, forecasting, etc.

- we aim to know some hidden law between two variables: $\boldsymbol{y} = f(\boldsymbol{x})$;

Machine Learning

Machine Learning helps in solving statistical problems, such as data generation, classification, regression, forecasting, etc.

- we aim to know some hidden law between two variables: $\boldsymbol{y} = f(\boldsymbol{x})$;
- we define a parameteric model with returns $\boldsymbol{y}_{\text{est}} = f_{\text{est}}(\boldsymbol{x}; \boldsymbol{\theta})$;

Machine Learning

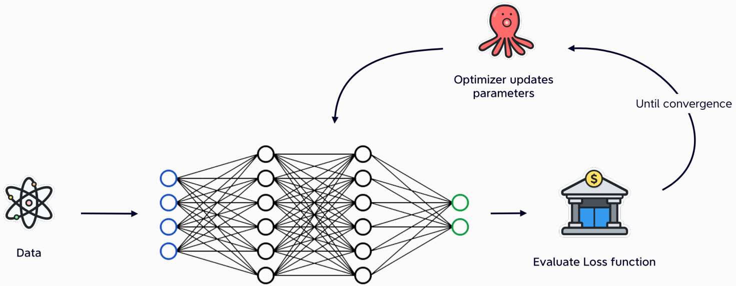
Machine Learning helps in solving statistical problems, such as data generation, classification, regression, forecasting, etc.

- we aim to know some hidden law between two variables: $\boldsymbol{y} = f(\boldsymbol{x})$;
- we define a parameteric model with returns $\boldsymbol{y}_{\mathrm{est}} = f_{\mathrm{est}}(\boldsymbol{x}; \boldsymbol{\theta})$;
- we define an optimizer, which task is to compute $\mathrm{argmin}_{\boldsymbol{\theta}} \left[J(\boldsymbol{y}_{\mathrm{meas}}, \boldsymbol{y}_{\mathrm{est}}) \right]$.

Machine Learning

Machine Learning helps in solving statistical problems, such as data generation, classification, regression, forecasting, etc.

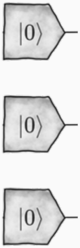
- we aim to know some hidden law between two variables: $\boldsymbol{y} = f(\boldsymbol{x})$;
- we define a parameteric model with returns $\boldsymbol{y}_{\mathrm{est}} = f_{\mathrm{est}}(\boldsymbol{x}; \boldsymbol{\theta})$;
- we define an optimizer, which task is to compute $\mathrm{argmin}_{\boldsymbol{\theta}}\left[J(\boldsymbol{y}_{\mathrm{meas}}, \boldsymbol{y}_{\mathrm{est}})\right]$.



Parametric Quantum Circuits

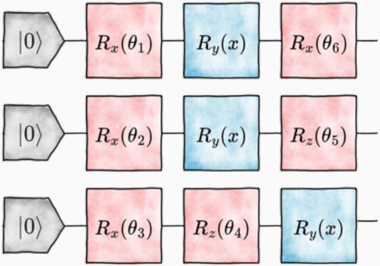
Parametric Quantum Circuits

- Classical bits are replaced by **qubits**: $|q\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$;



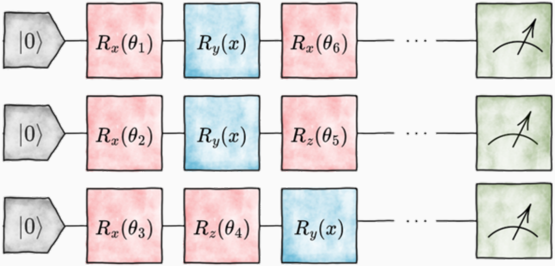
Parametric Quantum Circuits

- Classical bits are replaced by **qubits**: $|q\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$;
- we modify the qubits state by applying unitaries, which we call **gates**. Rotational gates $R_j(\theta) = e^{-i\theta\hat{\sigma}_j}$ are used to build parametric circuits $\mathcal{C}(\boldsymbol{\theta})$;



Parametric Quantum Circuits

- Classical bits are replaced by **qubits**: $|q\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$;
- we modify the qubits state by applying unitaries, which we call **gates**. Rotational gates $R_j(\theta) = e^{-i\theta\hat{\sigma}_j}$ are used to build parametric circuits $\mathcal{C}(\boldsymbol{\theta})$;
- information is accessed calculating expected values $\mathrm{E}[\hat{O}]$ of target observables $\hat{O}$ on the state obtained executing $\mathcal{C}$.



Quantum Machine Learning

Quantum Machine Learning - doing ML using QC

Machine Learning
$\mathcal{M}$: model;
$\mathcal{O}$: optimizer;
$\mathcal{J}$: loss function.
(x, y): data

Quantum Computation
$\mathcal{Q}$: qubits;
$\mathcal{S}$: superposition;
$\mathcal{E}$: entanglement.

Machine Learning

$\mathcal{M}$: model;

$\mathcal{O}$: optimizer;

$\mathcal{J}$: loss function.

(x, y): data

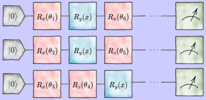
Quantum Computation

$\mathcal{Q}$: qubits;

$\mathcal{S}$: superposition;

$\mathcal{E}$: entanglement.

Circuit execution



Quantum Machine Learning - natural randomness

Machine Learning

$\mathcal{M}$: model;

$\mathcal{O}$: optimizer;

$\mathcal{J}$: loss function.

(x, y): data

Expected values

$y_{est} \equiv \langle q_f | \hat{O} | q_f \rangle$

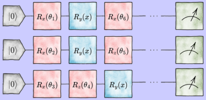
Quantum Computation

$\mathcal{Q}$: qubits;

$\mathcal{S}$: superposition;

$\mathcal{E}$: entanglement.

Circuit execution



Quantum Machine Learning - encoding the problem

Machine Learning

$\mathcal{M}$: model;

$\mathcal{O}$: optimizer;

$\mathcal{J}$: loss function.

(x, y): data

Expected values

$y_{est} \equiv \boxed{\langle q_f | \hat{O} | q_f \rangle}$

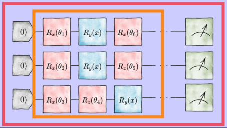
Quantum Computation

$\mathcal{Q}$: qubits;

$\mathcal{S}$: superposition;

$\mathcal{E}$: entanglement.

Circuit execution



Quantum Machine Learning!

Optimizer $\mathcal{O}$
Hybrid-strategy

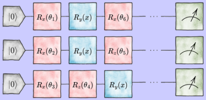
Machine Learning
$\mathcal{M}$: model;
$\mathcal{O}$: optimizer;
$\mathcal{J}$: loss function.
(x, y): data

loss function $\mathcal{J}$
$\mathcal{J}(y_{meas}, y_{est})$

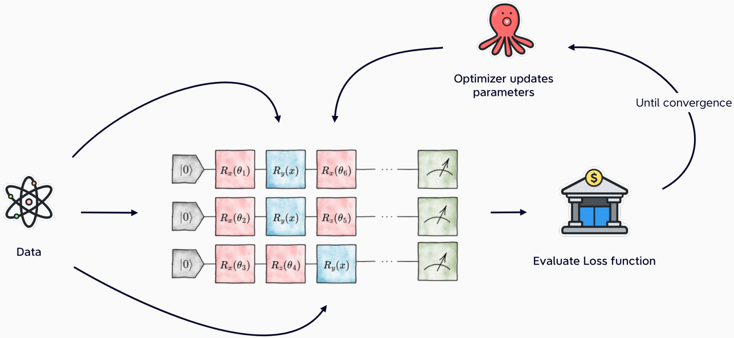
Expected values
$y_{est} \equiv \langle q_f | \hat{O} | q_f \rangle$

Quantum Computation
$\mathcal{Q}$: qubits;
$\mathcal{S}$: superposition;
$\mathcal{E}$: entanglement.

Circuit execution



From ML to QML



Why QML?

Why QML?

- we collect **high-dimensional** data, which challenge classical models;

Why QML?

- we collect **high-dimensional** data, which challenge classical models;
- an N-long input variable can be stored in a $\log N$ qubits system;

Why QML?

- we collect **high-dimensional** data, which challenge classical models;
- an N-long input variable can be stored in a $\log N$ qubits system;
- low power consuption[1];

[1] *Are quantum computers really energy efficient?*, Sophia Chen, Jun 2023

Why QML?

- we collect **high-dimensional** data, which challenge classical models;
- an N-long input variable can be stored in a $\log N$ qubits system;
- low power consuption[1];
- can **superposition** and **entanglement** be exploited to use less **parameters**?

[1] *Are quantum computers really energy efficient?*, Sophia Chen, Jun 2023

Why QML?

- we collect **high-dimensional** data, which challenge classical models;
- an N-long input variable can be stored in a $\log N$ qubits system;
- low power consuption[1];
- can **superposition** and **entanglement** be exploited to use less **parameters**?
- can superposition and entanglement better deal with quantum data?[2][3]

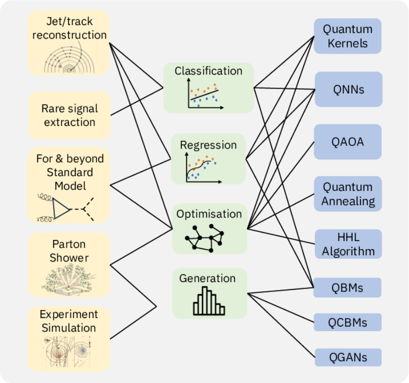
[1] *Are quantum computers really energy efficient?*, Sophia Chen, Jun 2023

[2] *Quantum Computing for High-Energy Physics: State of the Art and Challenges*, A. Di Meglio et al., Jul 2023

[3] *Quantum anomaly detection in the latent space of proton collision events at the LHC*, K. A. Woźniak et al., Jan 2023

Why QML?

- we collect **high-dimensional** data, which challenge classical models;
- an N-long input variable can be stored in a $\log N$ qubits system;
- low power consuption[1];
- can **superposition** and **entanglement** be exploited to use less **parameters**?
- can superposition and entanglement better deal with quantum data?[2][3]

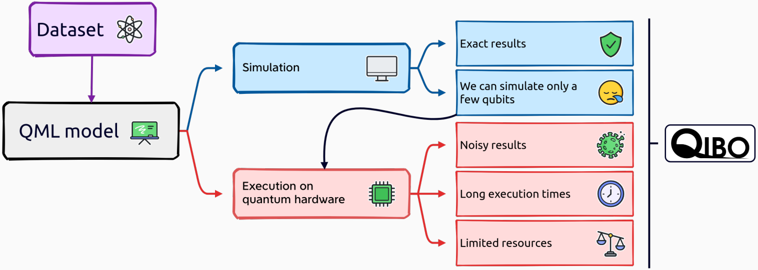


[1] *Are quantum computers really energy efficient?*, Sophia Chen, Jun 2023

[2] *Quantum Computing for High-Energy Physics: State of the Art and Challenges*, A. Di Meglio et al., Jul 2023

[3] *Quantum anomaly detection in the latent space of proton collision events at the LHC*, K. A. Woźniak et al., Jan 2023

Full-stack QML



Some results

High level API: Qibo

- </> define **prototypes** and models;
- </> **simulate** training and noise.

A full-stack PDF fit

arXiv:2308.06313

High level API: Qibo

- </> define **prototypes** and models;
- </> **simulate** training and noise.

Calibration: Qibocal

- **calibrate** qubits;
- generate **platform configuration**;

High level API: Qibo

- </> define **prototypes** and models;
- </> **simulate** training and noise.

Calibration: Qibocal

- **calibrate** qubits;
- generate **platform configuration**;

Execution: Qibolab

- allocate **calibrated** platform;
- **compile** and **transpile** circuits;
- execute and return **results**.

High level API: Qibo

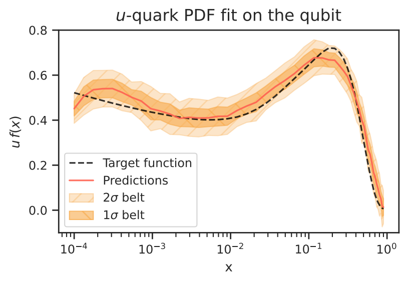
- </> define **prototypes** and models;
- </> **simulate** training and noise.

Calibration: Qibocal

- **calibrate** qubits;
- generate **platform configuration**;

Execution: Qibolab

- allocate **calibrated** platform;
- **compile** and **transpile** circuits;
- execute and return **results**.

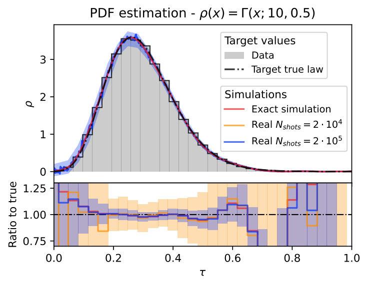


Parameter	Value
N_{data}	50
N_{shots}	500
MSE	$\sim 10^{-3}$
Electronics	Xilinx ZCU216
Training time	$\sim$ 2h

Some applications

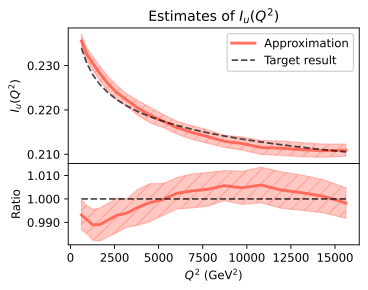
Some applications

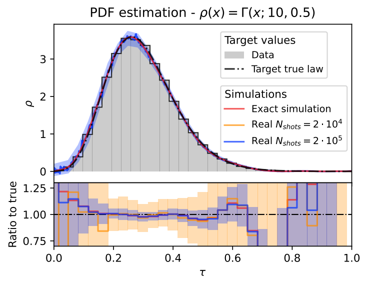
- Density estimation via Adiabatic Quantum Machine Learning, arXiv:2303.11346;



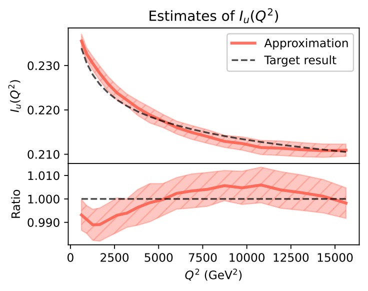
Some applications

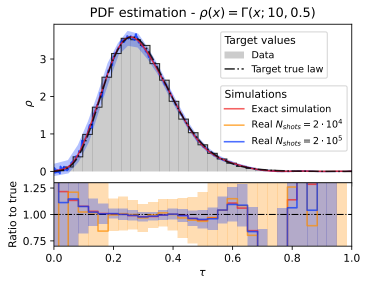
- Density estimation via Adiabatic Quantum Machine Learning, arXiv:2303.11346;
- Multi-dimensional integration using Variational Quantum Circuits, arXiv:2303.11346;

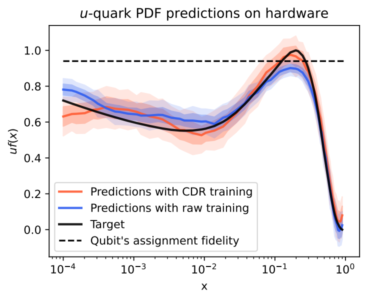




- Density estimation via Adiabatic Quantum Machine Learning, arXiv:2303.11346;
- Multi-dimensional integration using Variational Quantum Circuits, arXiv:2303.11346;
- Real-Time Quantum Error Mitigation to improve QML trainings, Coming soon!







Perspective

"One big challenge in the next three to five years is to figure out how to use the quantum hardware to improve a task in practice".

Iordanis Kerenidis.

Questions?

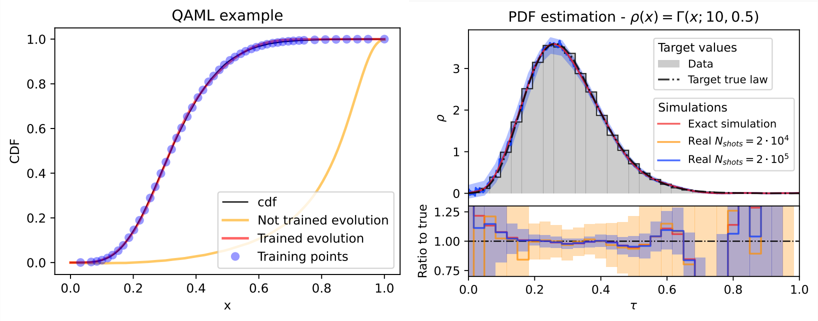
Determining PDFs with adiabatic quantum computing

- Determining Probability Density Functions (PDF).
- Algorithm's summary:

1. we optimize the parameters $\boldsymbol{\theta}$ of the following adiabatic evolution:

$$H_{\rm ad}(\tau;\boldsymbol{\theta}) = [1 - s(\tau;\boldsymbol{\theta})]\hat{\sigma}_x + s(\tau;\boldsymbol{\theta})\hat{\sigma}_z. \tag{1}$$

   we use the GS of the evolved $H_{\rm ad}$ to approximate the Cumulative Density Function (CDF);
2. we derivate from $H_{\rm ad}$ a circuit $\mathcal{C}(\tau;\boldsymbol{\theta})$ whose action on the GS of $\hat{\sigma}_x$ returns $|\psi(\tau)\rangle$;
3. we compute the PDF by derivating $\mathcal{C}$ w.r.t. τ using the Parameter Shift Rule (PSR).



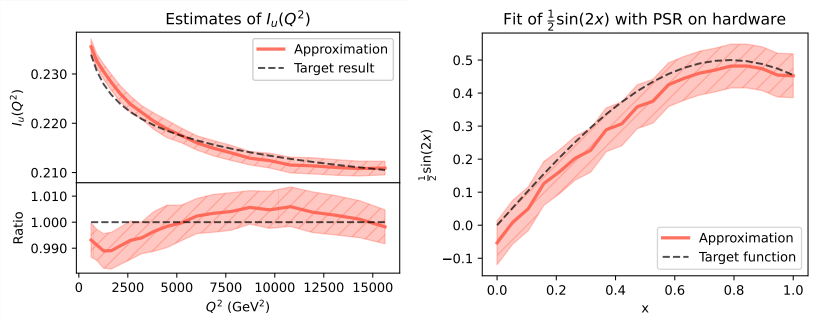
Multi-dimensional integration on hardware

- Use Variational Quantum Circuits to calculate multi-dimensional integrals of the form

$$I(\alpha) = \int_{\boldsymbol{x}_a}^{\boldsymbol{x}_b} g(\alpha; \boldsymbol{x}) \mathrm{d}^n \boldsymbol{x}. \tag{2}$$

- Algorithm's summary:

1. we train the derivative of a VQC w.r.t. the integral variables $\boldsymbol{x}$ to approximate $g(\boldsymbol{x})$;
2. we compute the derivatives using the PSR, which allows the same circuit $\mathcal{C}$ to be used for approximating any integrand marginalisation and the primitive! when varying α.



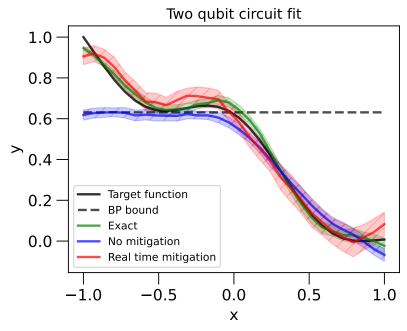
Real time error mitigation in QML trainings

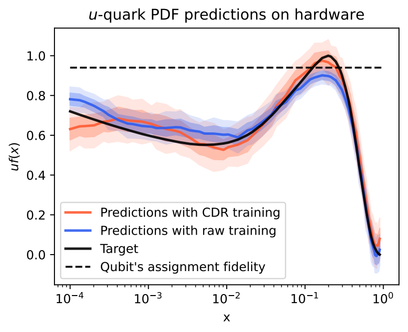
- Use Error Mitigation techniques to clean up the parameters space during the QML training.
- Algorithm's summary:

1. we mitigate all the expected values E through Clifford Data Regression (CDR):

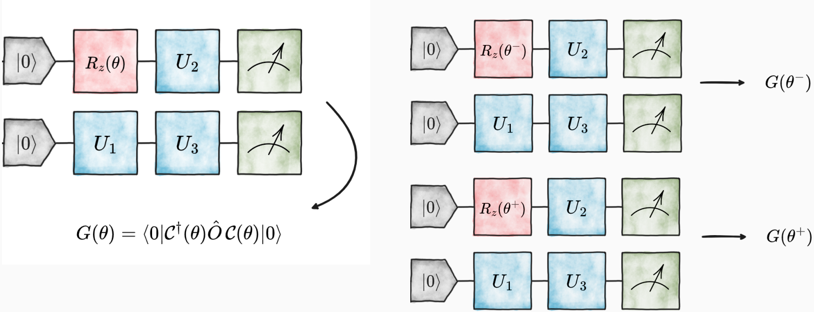
$$E_{\mathrm{mit}} = \alpha_{\mathrm{cdr}} E_{\mathrm{noisy}} + \beta_{\mathrm{cdr}}; \tag{3}$$

2. we update $(\alpha, \beta)_{\mathrm{cdr}}$ periodically during the training in order to track the noise;
3. the mitigation removes the bounds and accelerates the training process.





Parameter Shift Rule



The same circuit is executed twice with a shifted target parameter θ, then an exact evaluation of the derivative is given by:

$$\partial_\theta G = r\big[G(\theta^+) - G(\theta^-)\big].$$

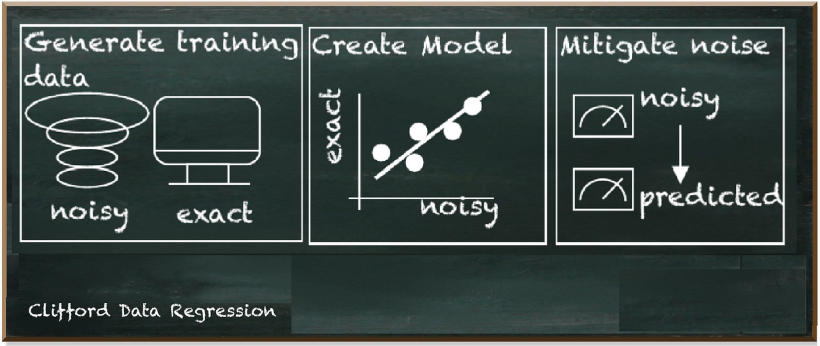


Figure 1: Credits: Frank Zickert.