Bayesian Deep Learning on a Quantum Computer

Peter Wittek

9 October 2018

We won an award!

October 4, 2018
Posted in: Quantum Computing

We Have Winners! ... Of The IBM Q Best Paper Award

To encourage more teachers and students to take advantage of the IBM Q Experience and the IBM Qiskit development platform, we announced in January a number of challenges and prizes to inspire people to take the quantum leap.

We're happy to announce the winners of the third IBM Q Award: the IBM Q Best Paper Award, which offered one first place prize of $1,500, one second place prize of $1,000, and a travel stipend of up to $1,500 for the authors of the best five papers to attend a quantum event held at an IBM Research lab.

The prizes were awarded for the highest-impact scientific papers by a master's student, PhD student or postdoctoral researcher using the IBM Q Experience and Qiskit as tools to achieve the presented results.

Congratulations to our first-place winner, Christophe Vuillot of QuTech and TU Delft; our second-place winner, Clement Javerzac-Galy, and his students, of EPFL; and our three runners-up, Maria Schuld of the University of KwaZulu-Natal, Shantanu Debnath of the University of Maryland, College Park; and Alejandro Pozas-Kerstjens of ICFO – The Institute of Photonic Sciences.

"My research showed an average improvement of the task of sampling from states that can be fault-tolerantly prepared in the [[4,2,2]] code, when using a fault-tolerant technique well suited to the layout of the IBM five-qubit chip, showing that fault-tolerant quantum computation is already within our reach," says Christophe.

Christophe Vuillot

Clement's paper started as a student project when his master's class on quantum information at EPFL started using the IBM Q Experience. "Very quickly we decided to shift to Qiskit to program some more complex experiments," he says. "We realized how powerful the full system is for learning and for training the next generation of quantum engineers."

Clement Javerzac-Galy

Clement Javerzac-Galy's students

"In my paper, I propose that quantum machine learning should be thought of from the direction of quantum circuits which lead to classifiers, instead of the other way around," says Maria. "We implemented a proof-of-principle experiment with the IBM Q Experience, and together with numerical simulations, showed that this classifier works surprisingly well in simple benchmarks, providing a minimal example of a quantum machine learning algorithm that can be implemented and understood by beginners to quantum computing."

Maria Schuld

Shantanu's experimental comparison of two quantum computing architectures supported the idea that quantum computer applications and hardware should be co-designed, and Alejandro developed a quantum algorithm for performing Bayesian learning on deep neural networks.

Shantanu Debnath

Our sincere thanks goes to all the participants. It was difficult to pick a winner, as all of the entries were of high quality.

Visit the Qiskit blog over the coming weeks to read blog posts by the winners!

IBM Research editorial staff

Share

Related articles

August 20, 2018
We Have Winners! ... of the IBM Qiskit Developer Challenge
To encourage more teachers and students to take advantage of the IBM Q Experience and the IBM Qiskit development platform,...

June 1, 2018
Teach Me QISKit: We have a winner!
We announced in January four challenges and prizes to encourage people to take the quantum leap by taking advantage of...

May 17, 2018
First IBM Q Hub in Asia to Spur Academic, Commercial Quantum Ecosystem
IBM built the first vacuum tube "supercomputer" in 1944, the Mark I. The first digital mainframe, the IBM 1401, came...

Archive

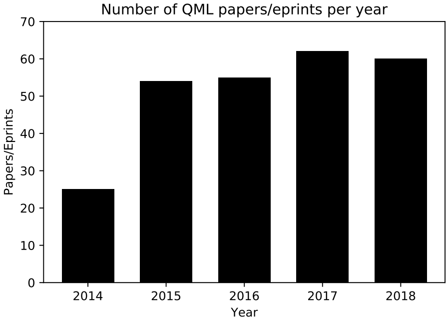
Select Month

What's wrong with quantum machine learning?

2013: First exponential speedup result in "machine learning" by using quantum resources.

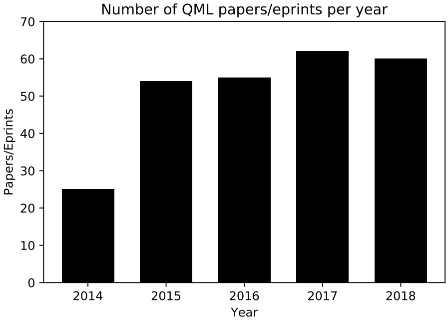
What’s wrong with quantum machine learning?

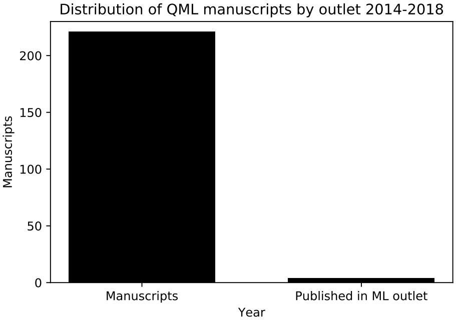
2013: First exponential speedup result in “machine learning” by using quantum resources.



What's wrong with quantum machine learning?

2013: First exponential speedup result in "machine learning" by using quantum resources.





Where it goes wrong: quantum BLAS

- Quantum states are vectors in a Hilbert space.
- Quantum computing transforms these vectors.
- Therefore you can do linear algebra fast on quantum computers.
- Terms and conditions apply.

Where it goes very wrong: quantum neural networks

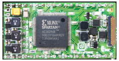
- Bad idea #1: Put all weights in superposition. Grover's search. Done.

Where it goes very wrong: quantum neural networks

- Bad idea #1: Put all weights in superposition. Grover's search. Done.
- Bad idea #2: Build some vaguely defined quantum optical system and interfere.
- It gets worse:
  - Quantum cognition and the quantum brain.
  - Add blockchain and black holes to it for additional impact.

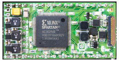
Why quantum-enhanced machine learning?

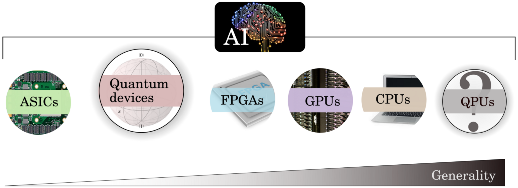
CPU → GPU → FPGA → ASIC



Why quantum-enhanced machine learning?

CPU → GPU → FPGA → ASIC



Supervised
Discriminative
Deterministic
Parallelizable

CLASSICAL

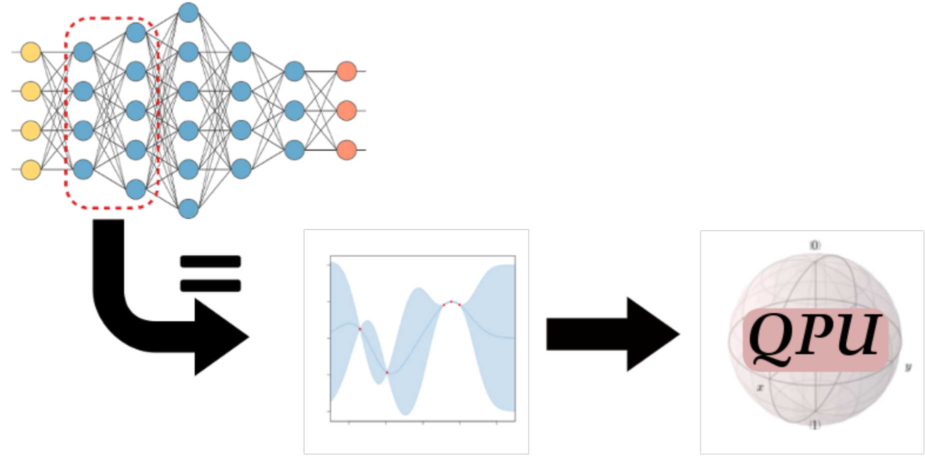
Unsupervised
Generative
Probabilistic
Difficult on GPU

QUANTUM

The premise: you need quantum data

- State preparation from classical data.
  - **Reverse is hard**: state tomography.
  - This is why exponential speedup claims have to be taken with a pinch of salt.
- QRAM.
- Quantum simulations.
- Internal workings of a quantum computer

What you need for Bayesian deep learning on a quantum computer



Quick refresher: Gaussian processes

- We are in the supervised regime: $\{(x_i, y_i)\}_{i=1}^{N}$.
- A Gaussian distribution is defined by its mean and variance. In other words, its first and second moments.
- Thing of the variance as a kernel: $f \sim \mathcal{N}(0, K)$, where $K_{ij} = k(x_i, x_j)$.
- Forward pass: calculating the posterior. Also Gaussian.

Quick refresher: Gaussian processes

- We are in the supervised regime: $\{(x_i, y_i)\}_{i=1}^{N}$.
- A Gaussian distribution is defined by its mean and variance. In other words, its first and second moments.
- Thing of the variance as a kernel: $f \sim \mathcal{N}(0, K)$, where $K_{ij} = k(x_i, x_j)$.
- Forward pass: calculating the posterior. Also Gaussian.
- $p(f(x)|\mathrm{data}) = \mathcal{N}(m, s^2)$
- Where:
  - $m = {k^*}^T(K + \sigma_n^2\mathbb{I})^{-1}y$, where $k_i^* = k(x^*, x_i)$;
  - $s^2 = k(x^*, x^*) - {k^*}^T(K + \sigma_n^2\mathbb{I})^{-1}k^*$.

Things Bayesian, things quantum

In Bayesian methods, we like ‘easy’ posteriors.
In quantum computing, we like linear operations.
Consequence: the kernel of the GP must be well behaved. ReLU works – it goes back to arccos calculations.

Three tricks

- Equivalence between (Bayesian) deep networks and Gaussian processes.
  - Recent result in machine learning.
- Modify the self-exponentiation routine of quantum PCA.
  - Allows clever element-wise matrix manipulation and calculating the outer product.
  - This is where the new science is in this paper.
- Quantum matrix inversion for Gaussian processes.
  - First author's earlier paper.

The main steps in quantum matrix inversion

We want to calculate $Ax = b$ for some b.
Instead, we will **sample** the solution x
We want to expand the b vector in the eigenbasis of A.

1. State preparation.
2. Quantum simulation
3. Quantum phase estimation to extract the eigenvalues.
4. Controlled rotation.
5. Uncomputation and post-selection.

A fantastic primer on HHL: arXiv:1802:08227

State preparation

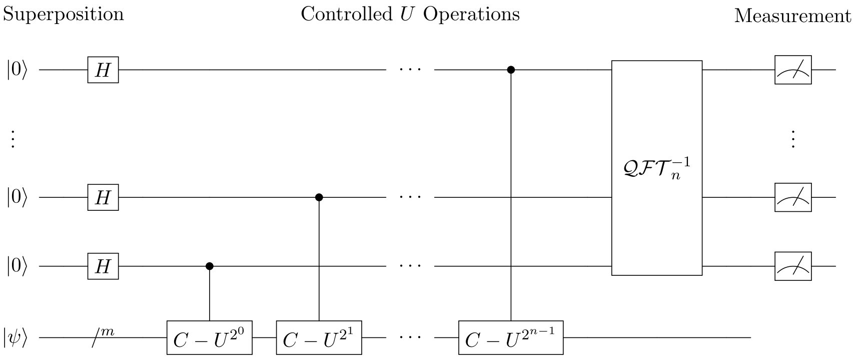
- Assume that A is Hermitian (i.e. $A = A^\dagger$).
- Assume that b is normalized and can be efficiently prepared. We write it as a quantum state in the bra-ket notation as $|b\rangle$.
- The thing about **kets**: yes, it is just a normalized vector in $\mathbb{C}^d$. But it is also a probability distribution.
- Big assumption: $|b\rangle$ can be efficiently prepared.
- An eigenvector of A is u_k. Taking it as a column vector, we write it in the bra-ket notation as $|u_k\rangle$.
- We want to expand the $|b\rangle$ vector in the eigenbasis of A.

Quantum simulation

- Next we prepare a carefully chosen ancilla state $|\psi_0\rangle = \sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin \frac{\pi(\tau+\frac{1}{2})}{T} |\tau\rangle$.
- Apply the conditional Hamiltonian evolution $U = \sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau| \otimes e^{\imath A \tau t_0 / T}$ on $|\psi_0\rangle \otimes |b\rangle$ for some time t_0.
- Why do we do this?
- How is classical information encoded?
- T: bit width for accuracy.
- Huge assumption: Hamiltonian to be simulated must be sparse.

Quantum phase estimation

Goal: given $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, where $|\psi\rangle$ is an eigenvector of U, write an approximation of the $\lambda = e^{2\pi i\theta}$ into an ancilla register.



This is where the scaling becomes bad: QFT needs $O(n^2)$ gates.
The state of the system after this decomposition is approximately: $\sum_{j=1}^{N} \beta_j |u_j\rangle|\lambda_j\rangle$.

Uncomputing the eigenvalue register and post-selection

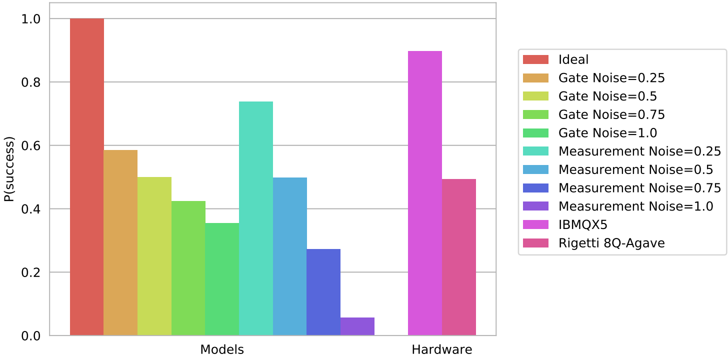
- A technicality: we have to uncompute the phase estimation. Why?
- Finally, we postselect on the last ancilla qubit:
  $\sum_{j=1}^{N} \beta_j |u_j\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_k^2}} |0\rangle + \frac{C}{\lambda_k} |1\rangle \right)$.

So why does it break again?

- Condition number: $\kappa^2 \log N$.
- Size of ancillas in error terms.
- Circuit depth.

How bad is it?

Quantum matrix inversion of a 2×2 matrix on real hardware in 2018.



Want to go deeper?

Download the slides: https://peterwittek.com/tdls_qml.pdf
Code repo: https://gitlab.com/apozas/bayesian-dl-quantum/
Upcoming events:

- Every other Thursday at 4pm, we have a quantum lecture in the Vector Institute. Next one: Oct 11.
- Half-day QC & QML workshop: Dec 14, Rotman.
- First MOOC on QML: February 2019.

A twelve-minute intro to QML:
https://medium.com/xanaduai/quantum-machine-learning-1-0-76a525c8cf69
LinkedIn Group: https://www.linkedin.com/groups/8592758/

Questions to consider

- Variational quantum algorithms for achieving the same end?
- Personal take: people avoid Bayesian methods because they require more thinking to get the posterior.
    - Quantum computing imposes additional constraints. Yet, can it be a killer app?
- Other kernels are hard: think reverse, starting from what the hardware can do.