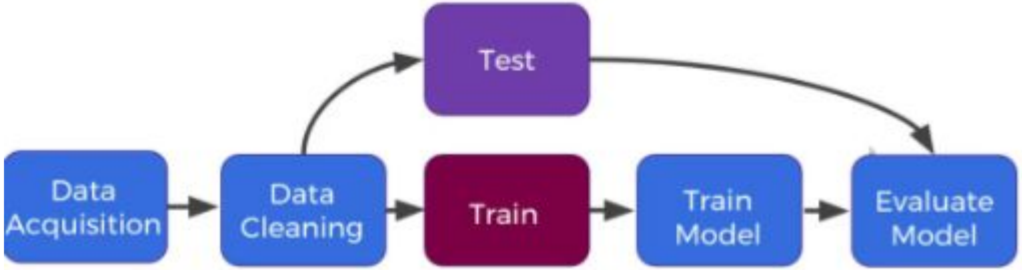
Quantum Machine Learning

Guo Zheng
Aug 15, 2018

Outline

1. Classical Machine Learning
    a. Overview
    b. Sample problems
    c. Limitations
2. Quantum Machine Learning
    a. Overview
    b. Basic ideas
    c. Deep quantum learning
    d. Sample algorithms
3. Summary
    a. Compare of CML & QML
    b. Major challenges
    c. Future potential

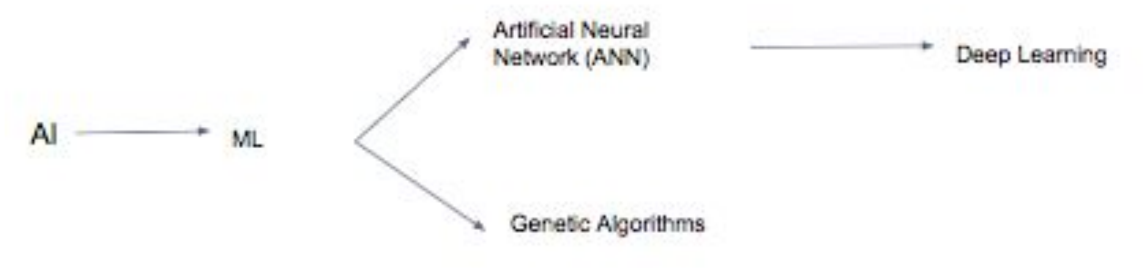
Classical machine learning



Using statistical techniques to progressively improve performance on a specific task.

CML structure

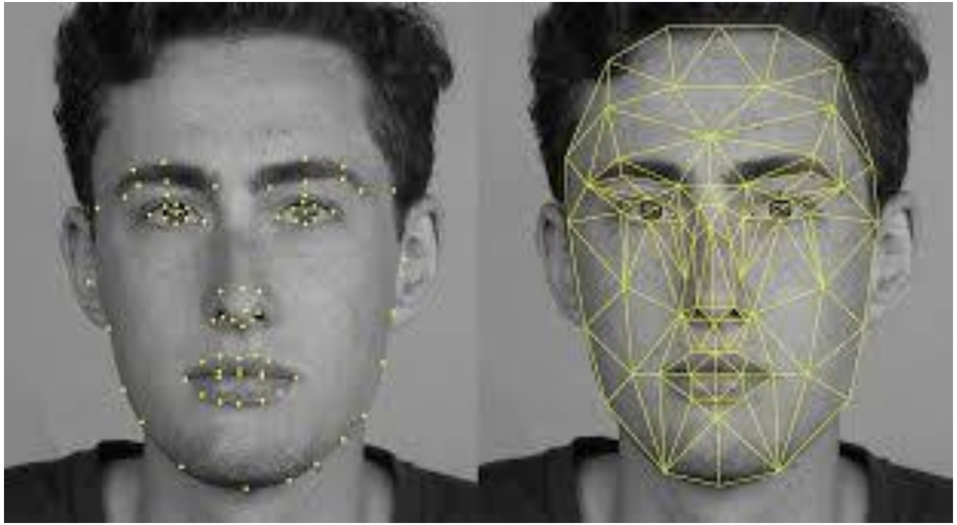
By basic idea:



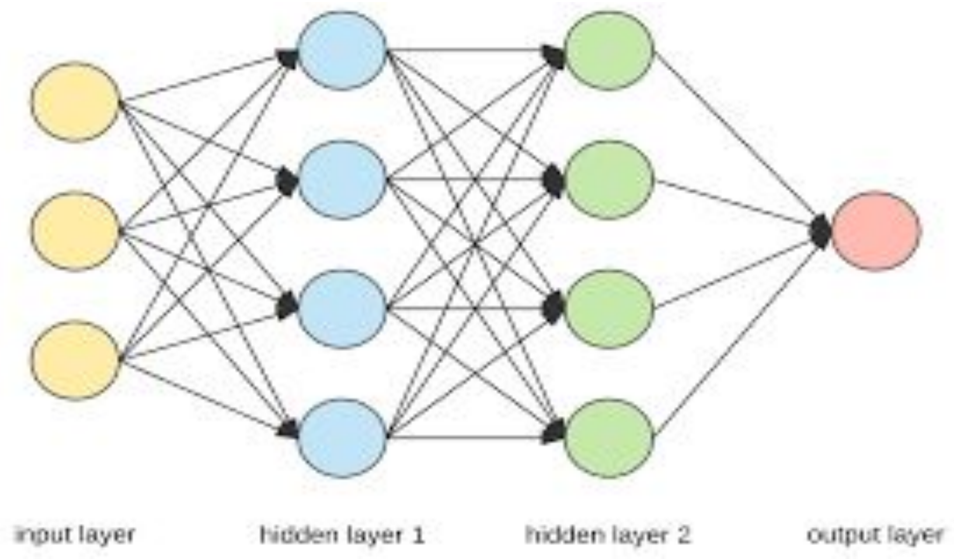
By method: ML

- Supervised learning, e.x. Classification problem
- Unsupervised learning, e.x. Grouping problem
- Reinforcement learning, e.x. Chess-playing problem

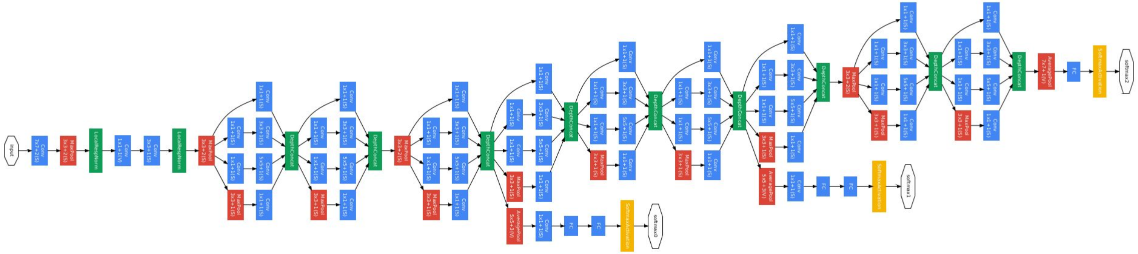
Sample problems - face recognition



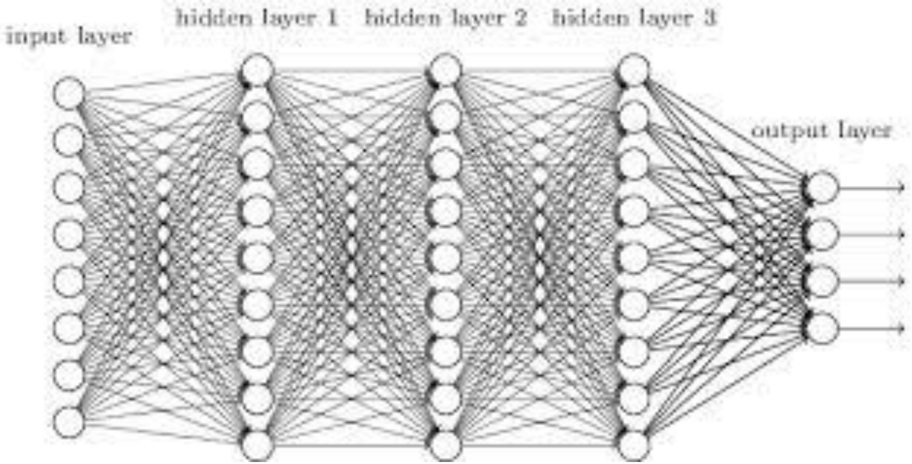
ANN & deep learning



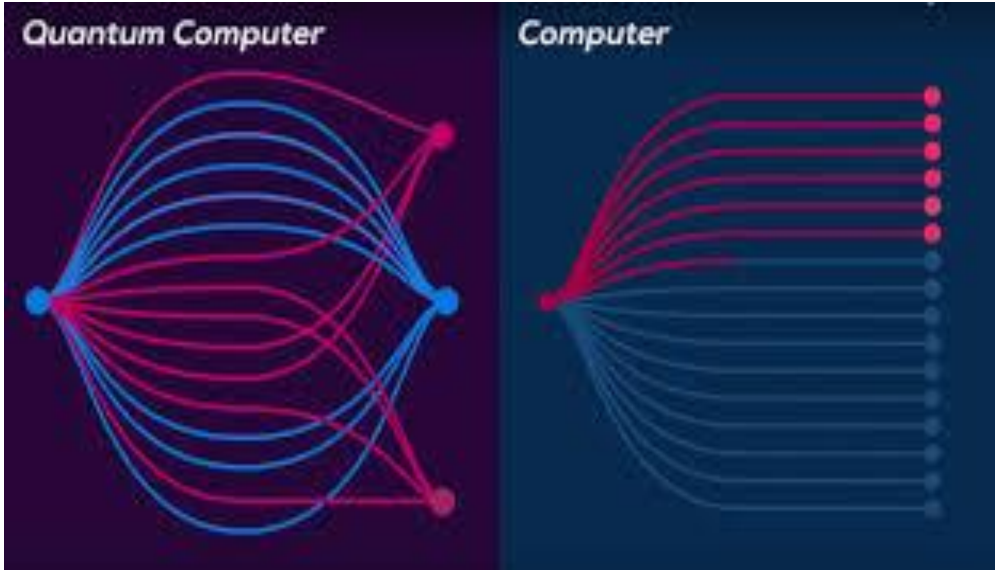
GoogleNet 2015, 22 layers



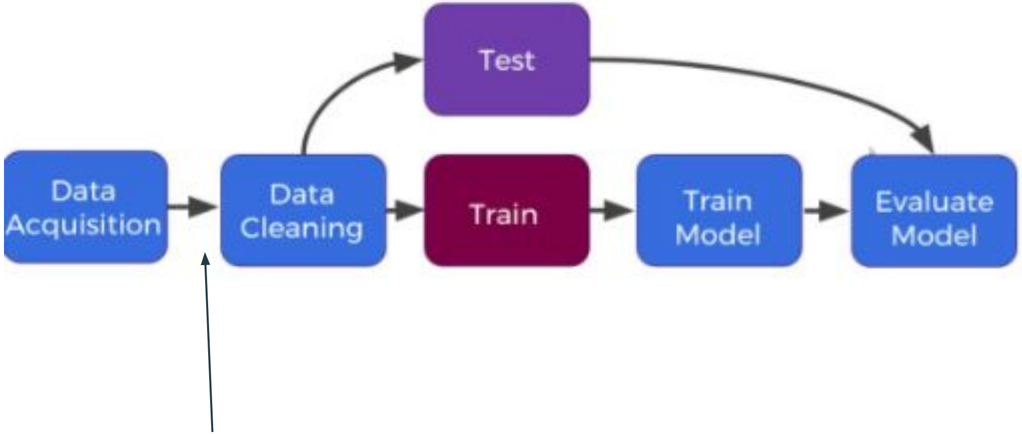
Conclusion: out of the control of classical computers (at least for now)



Quantum machine learning - basic ideas



Quantum Machine Learning



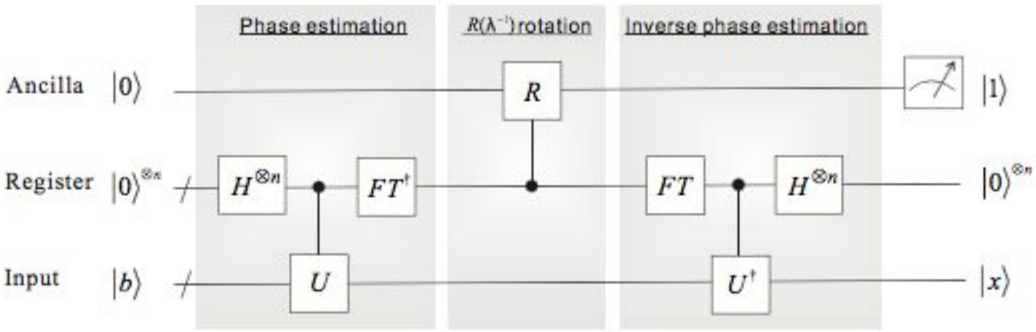
Convert data from their classical form
to quantum states

Theoretical basis

- Quantum RAM (qRAM) & Quantum Basic Linear Algebra Subroutines (qBLAS) (Exponential speedup)
    - Fourier transform
    - Finding eigenvectors and eigenvalues
    - Solving linear equations (HHL, 2009)
        - O(log(N))

$$A\vec{x} = \vec{b}$$

HHL algorithm



Computation:

- Get eigenvector and eigenvalue of A: Unitary Operator (Hamiltonian simulation) e^{iAt}
- Decompose |b> in the eigenvector basis: phase estimation $\vec{b} \longrightarrow \sum_{j=1}^{N} \beta_j |u_j\rangle$
- Inverse eigenvectors: non unitary operator, un-computation
- Compute A^-1

Limitations of HHL

Result: only in the form of $\vec{x}^{\dagger} M \vec{x}$

Requirements on A:

1. Condition number small
2. Sparse

Algorithms

1. Shor's algorithm (integer factorization)
    a. $O(e^{1.9 (\log N)^{1/3} (\log \log N)^{2/3}}) \longrightarrow O((\log N)^2 (\log \log N)(\log \log \log N))$
2. Grover's algorithm (search algorithm)
    a. O(N) to O(N^0.5)

QML - applicability

Designed to be used on quantum machines:

1. Large, general purpose quantum computer: Shor's algorithm, Grover's algorithm, etc

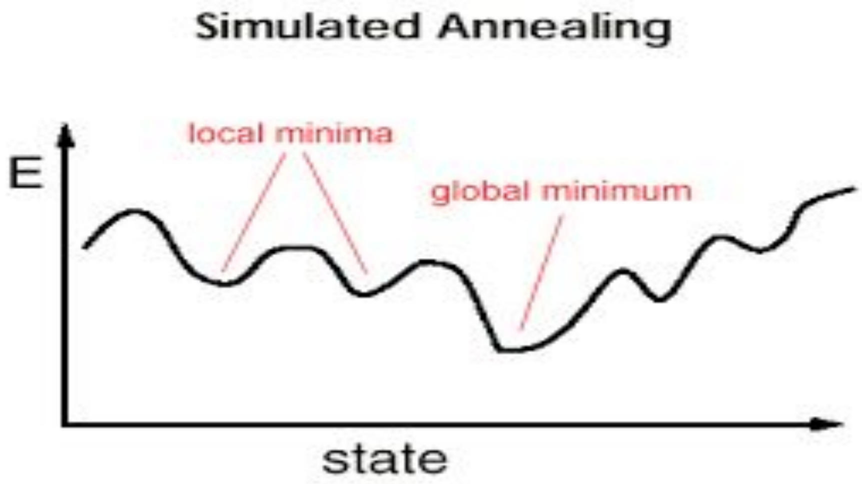
2. Special purpose information processor (quantum annealers, etc): deep quantum learning, HHL, etc

Change-making problem

Simulated annealing (SA)

https://en.wikipedia.org/wiki/Simulated_annealing

Probability of moving: P(e,e′,T)

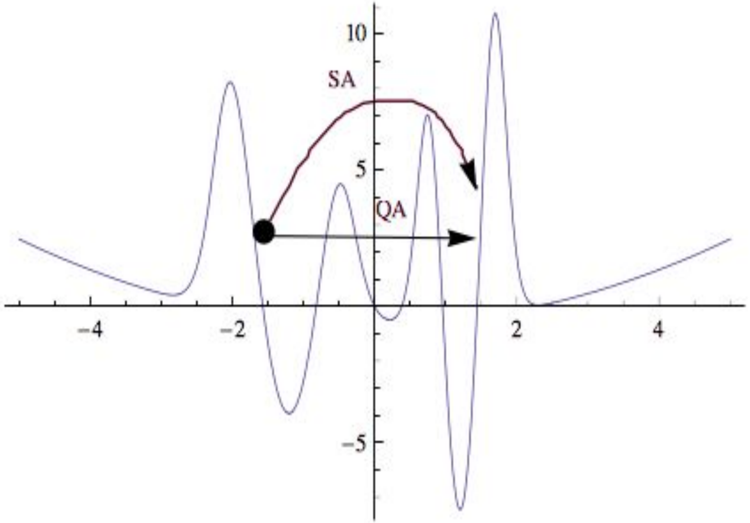


Quantum annealing (quantum stochastic optimization)

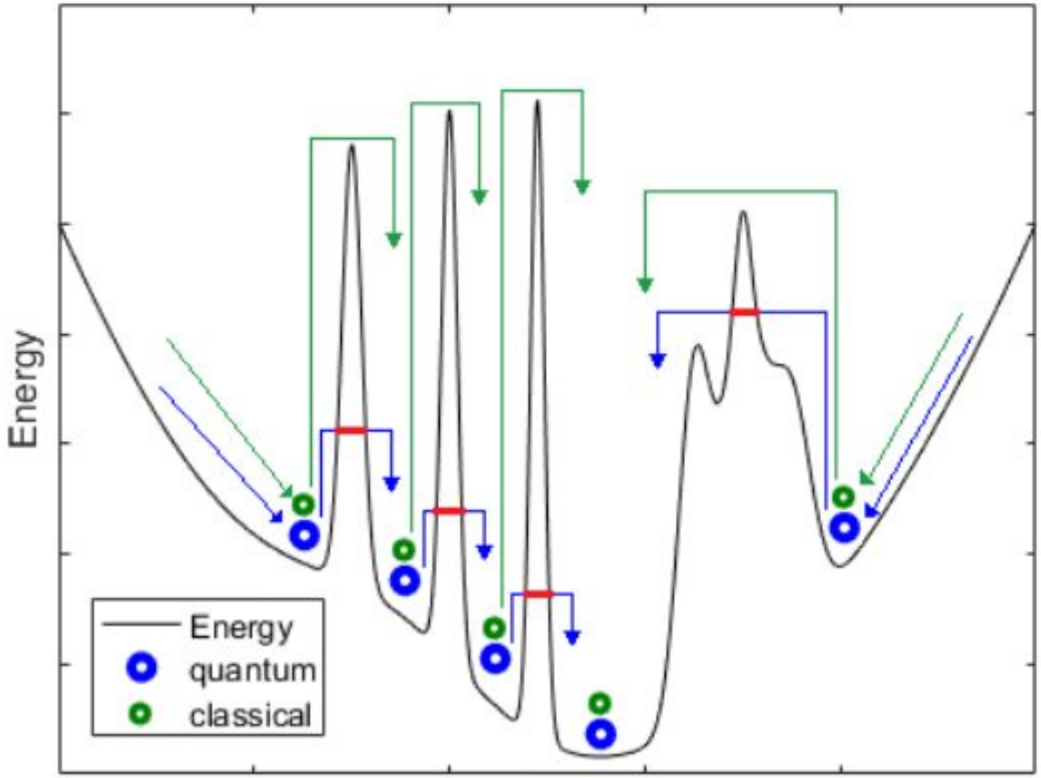
Evolves according to time-dependent schrodinger equation.

$$H_\nu = -\frac{\nu^2}{2}\frac{\partial^2}{\partial x^2} + V(x)$$

The time factor in the previous case is replaced by the potential energy term in the equation.



Pros



D-wave

Latest:

2017 D-wave 2000Q 2048 qubits

However, no particular quantum speedup in deep learning, compared to classical deep learning, has been found in their systems

Major challenges

1. Input
    a. Process of reading data still takes a certain amount of time, which might be a restriction on the complexity of the algorithm
    b. Construction of qRAM: principle demonstration of qRAM has been made, but constructing large arrays of quantum switches are still a large technical problem.
2. Output
    a. Expressing quantum solutions of the algorithms needs exponential number of classical bits relative to quantum bits. (Can be potentially overcome by reading the summary statistics of the solution states)
3. Costing
    a. Complexity-wise, QML is less than CML but it’s hard to determine the number of quantum gates needed for the a certain amount of data (hard to determine the crossover point).
4. Benchmarking
    a. Difficult to say QML is more efficient than all CML. Need to make more test of QML to determine it’s lower bounds etc compared to CML.

Future developments

- Quantum hardwares are needed to perform QML
  - Besides general purpose quantum computers, quantum annealers, quantum simulators, and other smaller quantum hardware should also be focused at.
- Can perform QML on quantum data first and use the results to design next generation processors as CML did

References

1. Jacob Biamonte. Quantum machine learning. https://arxiv.org/pdf/1611.09347.pdf
2. Krishnakemarsekar. Awesome quantum machine learning. https://github.com/krishnakumarsekar/awesome-quantum-machine-learning#quantumai
3. Cristian Romero Garcia. Quantum Machine Learning.
4. Carlo Ciliberto. Quantum machine learning: a classical perspective. http://rspa.royalsocietypublishing.org
5. Quantum Algorithms for linear systems of equations. Wikipedia. https://en.wikipedia.org/wiki/Quantum_algorithm_for_linear_systems_of_equations