



Overview of Quantum Machine Learning

Samyak Surti

Spring 2022

Quantum Computing at Berkeley

But first some history!

- Machine Learning → Recognizing and learning patterns in data + producing data with desired patterns

Where It All Began...Astronomy?

- 16th-Century - Kepler analyzed Copernicus's and Brahe's astronomical data → revealed true orbital nature of planets
- Analyzing astronomical data gave us many important mathematical techniques
 1. Linear system of equations (Newton-Gauss)
 2. Gradient Descent-Based Learning (Newton)
 3. Polynomial interpolation (Lagrange)
 4. Least-squares fitting (Lagrange)



Brief Synopsis of Classical Machine Learning

- Divided into several categories:
 1. 'Classic' data analysis
 2. Machine learning protocols
 - Supervised
 - Unsupervised
 3. Machine learning tasks
 - Mixture of Supervised and Unsupervised

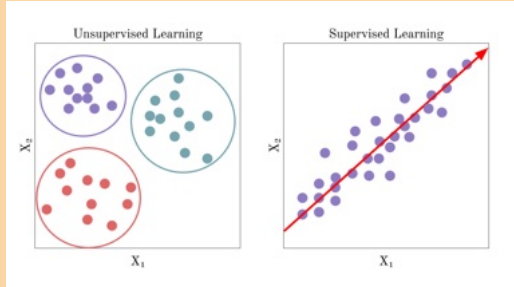


Figure 1: Supervised vs. Unsupervised Classical ML

Quantum Machine Learning: The Buzz-Word Collision

- Already well-acquainted with seemingly ubiquitous presence of Machine Learning in every field imaginable.

Quantum Machine Learning: The Buzz-Word Collision

- Already well-acquainted with seemingly ubiquitous presence of Machine Learning in every field imaginable.
 - Medical Diagnoses
 - Image Processing
 - Speech Recognition
 - Audio/Video Generation

Quantum Machine Learning: The Buzz-Word Collision

- Already well-acquainted with seemingly ubiquitous presence of Machine Learning in every field imaginable.
 - Medical Diagnoses
 - Image Processing
 - Speech Recognition
 - Audio/Video Generation
 - Etc...

Quantum Machine Learning: The Buzz-Word Collision

- Already well-acquainted with seemingly ubiquitous presence of Machine Learning in every field imaginable.
 - Medical Diagnoses
 - Image Processing
 - Speech Recognition
 - Audio/Video Generation
 - Etc...
- The **collision**
 - **Quantum Computing** → Theory promises sizeable computational advantage
 - **Machine Learning** → Computationally demanding

Quantum Machine Learning: The Buzz-Word Collision

- Already well-acquainted with seemingly ubiquitous presence of Machine Learning in every field imaginable.
 - Medical Diagnoses
 - Image Processing
 - Speech Recognition
 - Audio/Video Generation
 - Etc...
- The **collision**
 - **Quantum Computing** → Theory promises sizeable computational advantage
 - **Machine Learning** → Computationally demanding
 - **Quantum Machine Learning** is born!

Things are not as sweet as they seem... For now...

- Hardware Limitations
 - Quantum Decoherence
 - Gate Errors

Things are not as sweet as they seem... For now...

- Hardware Limitations
 - Quantum Decoherence
 - Gate Errors
 - Hope is that this can be addressed soon

Things are not as sweet as they seem... For now...

- **Hardware Limitations**
 - Quantum Decoherence
 - Gate Errors
 - Hope is that this can be addressed soon
- **Software Limitations**
 - Model of Computation
 - Scalability

Things are not as sweet as they seem... For now...

- **Hardware Limitations**
 - Quantum Decoherence
 - Gate Errors
 - Hope is that this can be addressed soon
- **Software Limitations**
 - Model of Computation
 - Scalability
 - Again, hopefully this is just a matter of time

Things are not as sweet as they seem... For now...

- **Hardware Limitations**
 - Quantum Decoherence
 - Gate Errors
 - Hope is that this can be addressed soon
- **Software Limitations**
 - Model of Computation
 - Scalability
 - Again, hopefully this is just a matter of time
- I will continue to reference some of these issues.

So what does Quantum Machine Learning look like right now?

- **Linear Algebra** integral part of Quantum Mechanics.
 - Quantum state $\rightarrow |\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$
 - Quantum Logic operations $\rightarrow \hat{A} \in \mathbb{C}^{2^n \times 2^n}$

So what does Quantum Machine Learning look like right now?

- **Linear Algebra** integral part of Quantum Mechanics.
 - Quantum state $\rightarrow |\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$
 - Quantum Logic operations $\rightarrow \hat{A} \in \mathbb{C}^{2^n \times 2^n}$
- **Basic Linear Algebra Subroutines (BLAS)** exponentially faster
 - Fourier transform
 - Computing eigenvectors and eigenvalues
 - Solving Linear Systems of Equations (HHL Algorithm)

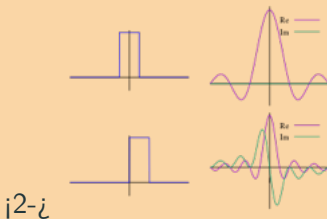


Figure 2: Fourier Transform

How much faster are we talking?

Method	Speedup	Amplitude amplification	HHL	Adiabatic	qRAM
Bayesian inference ^{106,107}	$O(\sqrt{N})$	Yes	Yes	No	No
Online perceptron ¹⁰⁸	$O(\sqrt{N})$	Yes	No	No	Optional
Least-squares fitting ⁹	$O(\log N)^*$	Yes	Yes	No	Yes
Classical Boltzmann machine ²⁰	$O(\sqrt{N})$	Yes/No	Optional/No	No/Yes	Optional
Quantum Boltzmann machine ^{22,61}	$O(\log N)^*$	Optional/No	No	No/Yes	No
Quantum PCA ¹¹	$O(\log N)^*$	No	Yes	No	Optional
Quantum support vector machine ¹³	$O(\log N)^*$	No	Yes	No	Yes
Quantum reinforcement learning ³⁰	$O(\sqrt{N})$	Yes	No	No	No

*There exist important caveats that can limit the applicability of the method⁵¹.

Figure 3: Speedups for Various QML Subroutines

Quantum Principal Component Analysis

- **Principal Component Analysis** used to compress our data's representation
- Simplest form \rightarrow diagonalizing the covariance matrix:

$$C = \sum_k e_k c_k c_k^\dagger$$

- Performing qPCA on classical data:
 - Use **qRAM (quantum Random Access Memory)** - classical data vector gets mapped to quantum state ($v_j \mapsto |v_j\rangle$)

Before we continue... What is qRAM?

- We know **Random-Access Memory (RAM)** uses " n bits to randomly address $N = 2^n$ distinct memory cells" [\[GLM08\]](#)
- **quantum Random-Access Memory (qRAM)** theoretically uses " n qubits to address any quantum superposition of N memory cells" [\[GLM08\]](#)
 - Large qubit-overhead \implies Not feasible in near-term
 - Costly memory call

Quantum Principal Component Analysis (Cont.)

- Suppose vectors live in d -dimensional space so that $d = 2^n = N$
- Principal components:

$$v = \sum_k v_k c_k$$

- Classical time complexity $\rightarrow \mathcal{O}(d^2)$
- Quantum time complexity $\rightarrow \mathcal{O}[(\log N)^2]$
 - Quantum state has $\log d$ qubits

Quantum Support Vector Machine (SVM)

- Simplest supervised machine learning algorithms:
 - Linear SVMs
 - Perceptrons
- Quantum SVM → canonical example for QML techniques
 1. Data input (qRAM or other subroutine)
 2. Process data with Quantum Phase Estimation and Matrix Inversion
- Operations to construct hyperplane take $\log N$

qBLAS-based optimization

- Optimization is integral to data analysis and ML techniques
- Methods:
 1. Quantum annealing to solve combinatorial optimization problems
 2. Single-shot solution to linear system → Use HHL Algorithm
 3. Modified quantum PCA implements iterative gradient descent and Newton's method → polynomial optimization
 4. Quantum Approximate Optimization Algorithm (QAOA)



Figure 4: D-Wave's Quantum Annealer

- Quantum Annealers and Photonics Circuits suitable for deep quantum learning networks
- Simplest deep neural network → Boltzmann Machine
 - Trained by adjusting interactions between bits described by a Boltzmann-Gibbs distribution
- Quantum Boltzmann Machine
 - Take neural network and express as interacting quantum spins (tunable Ising model)

Deep Quantum Learning (Cont.)

- Deep quantum learning networks do not require large, fault-tolerant, error-corrected Quantum Computers.
 - Quantum annealers well-suited
- Advantages Quantum Computers can provide:
 1. Quadratically-faster thermalization of system
 2. Can accelerate Boltzmann training
 3. Quantum access to data \implies quadratically less accesses

Where does it make most sense to apply QML... Quantum Data!

- This is the most immediate application
- QML on classical data → have to map classical data to quantum states
- Instead... QML on quantum data!
 - Allows for direct application of algorithms
- Example 1: Quantum PCA on quantum data
 - Quantum Time complexity - $\mathcal{O}[(\log N)^2]$
 - Classical Time complexity - $\mathcal{O}(N^2)$
- Example 2: Quantum simulators to probe quantum dynamics
 - Dynamics can be learned using approximate Bayesian inference
- These methods do not require qRAM and can realize exponential speed-ups!

Designing Suitable Quantum Systems

- **Major Challenge:** Tuning quantum gates to meet threshold requirements prescribed by quantum error correction
 - Genetic algorithms have been employed to reduce digital and experimental gate errors
 - Stochastic gradient-descent and two-body interactions
 - Using dynamical decoupling sequences, which can be designed using recurrent neural networks, to protect quantum states from decohering.

Controlling Quantum Systems

- Having control over your quantum systems is equally important
- Approaches:
 - Genetic Algorithms
 - Reinforcement Learning

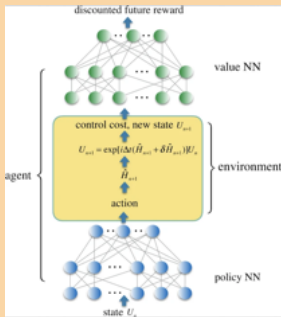


Figure 5: Using reinforcement learning to control quantum systems 17/19

Future Prospects and Areas of Research

- Small NISQ-era quantum computers along with special purpose quantum devices (quantum annealers, quantum simulators, integrated photonics chips, etc...) have potential use in machine learning applications
- Required area of focus to realize QML algorithms → Hardware
- Quantum Devices and Hardware Advances:
 1. **Quantum Annealers** - Improving connectivity and more tunable couplings
 2. **Quantum Optic Arrays** - Loss of quantum effects upon scaling system size
 3. **qRAM** - Discussed prior

4 Main Problems We Need to Address for QML

1. The Input Problem

- Pertaining to method of reading data

4 Main Problems We Need to Address for QML

1. The Input Problem

- Pertaining to method of reading data

2. The Output Problem

- Quantum state \rightarrow classical bit string

4 Main Problems We Need to Address for QML

1. The Input Problem

- Pertaining to method of reading data

2. The Output Problem

- Quantum state \rightarrow classical bit string

3. The Costing Problem

- Determining true number of quantum gates for QML algorithms

4 Main Problems We Need to Address for QML

1. The Input Problem
 - Pertaining to method of reading data
2. The Output Problem
 - Quantum state \rightarrow classical bit string
3. The Costing Problem
 - Determining true number of quantum gates for QML algorithms
4. The Benchmarking Problem
 - Are Quantum Algorithms even better?