Quantum Machine Learning

Iordanis Kerenidis

Director, QC Ware France

Research Director, CNRS, U Paris

iordanis.kerenidis@qcware.com

Why care now about Quantum Computing?

The quantum revolution 2.0

The quantum revolution 2.0

- Quantum is NOT a faster processor

The quantum revolution 2.0

- Quantum is NOT a faster processor

- Quantum is a fundamentally DIFFERENT way of performing computation that can be MUCH FASTER for CERTAIN tasks

The quantum revolution 2.0

- Quantum is NOT a faster processor
- Quantum is a fundamentally DIFFERENT way of performing computation that can be MUCH FASTER for CERTAIN tasks
- Quantum computing will NOT replace classical computing. It will open the way to new applications in areas like Climate, Energy, Materials, Automotive, Finance, Telecommunications, Pharmaceuticals, Healthcare, ...

The quantum revolution 2.0

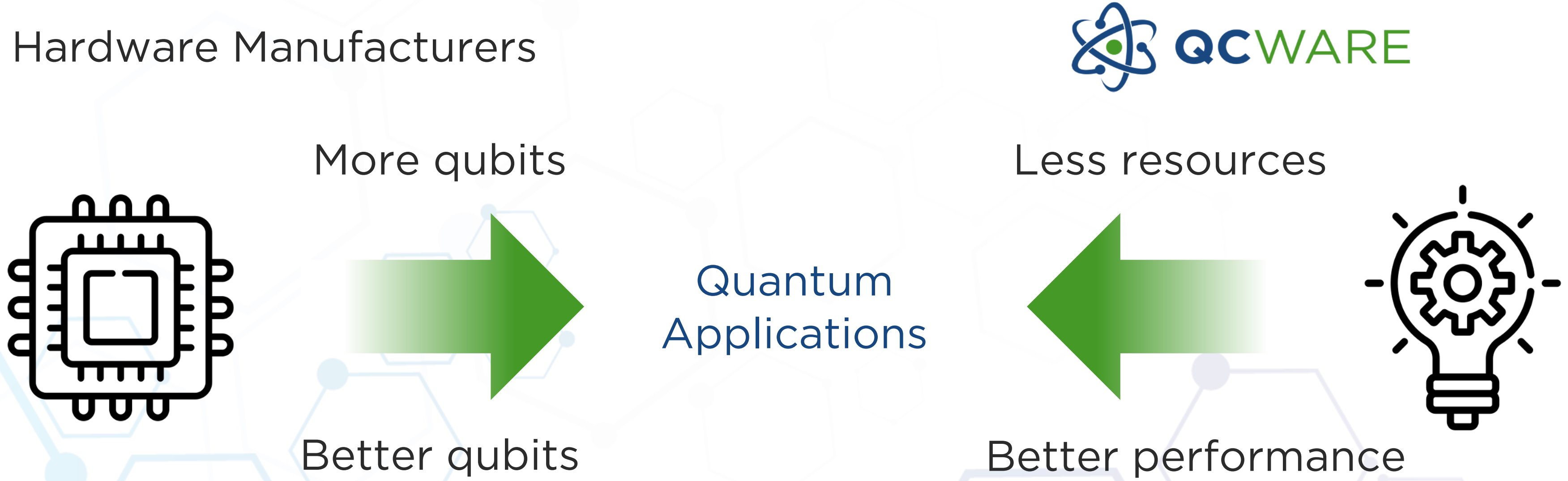
- Quantum is NOT a faster processor

- Quantum is a fundamentally DIFFERENT way of performing computation that can be MUCH FASTER for CERTAIN tasks

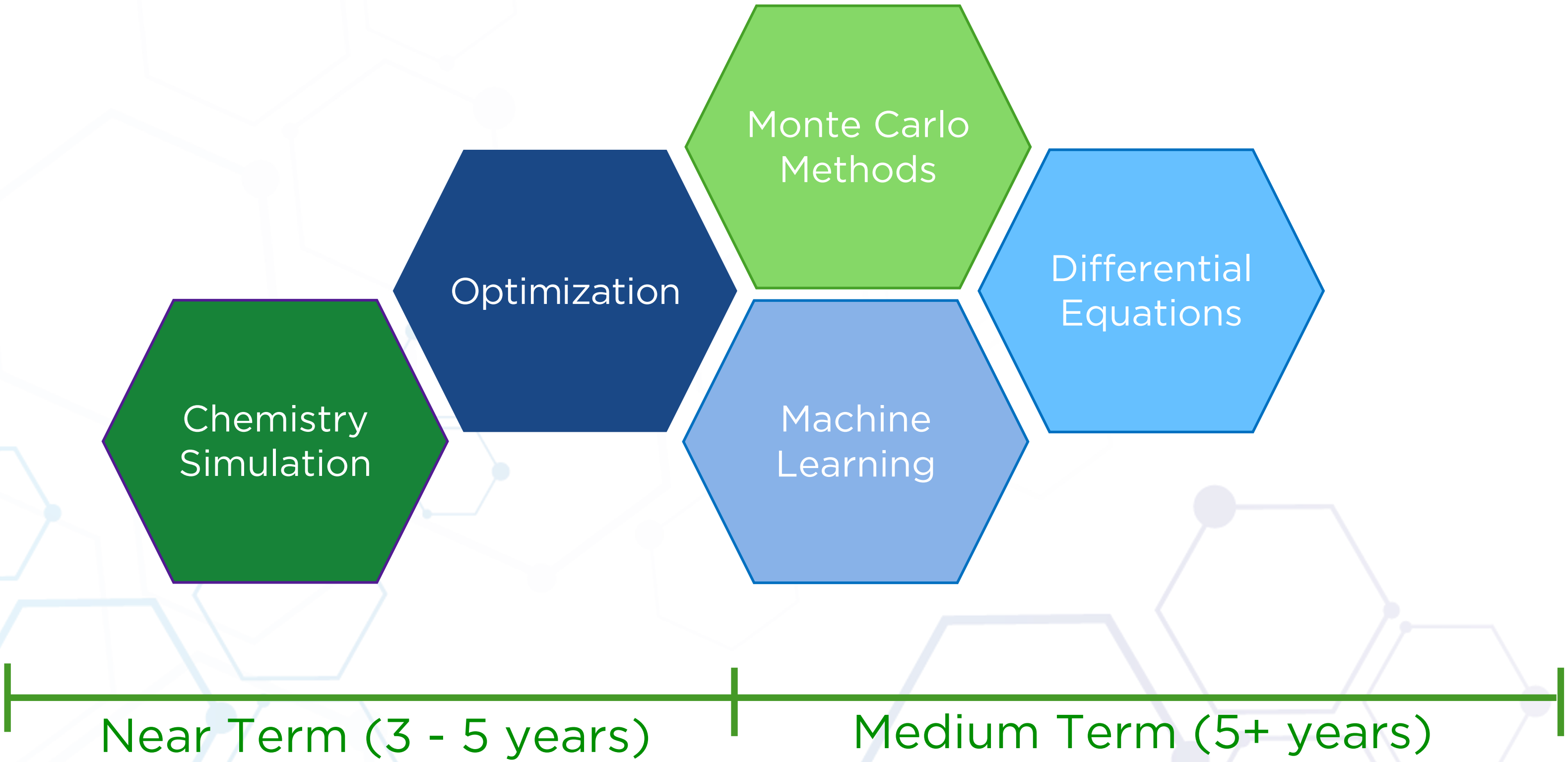
- Quantum computing will NOT replace classical computing. It will open the way to new applications in areas like Climate, Energy, Materials, Automotive, Finance, Telecommunications, Pharmaceuticals, Healthcare, …

- We need to rethink and invent new algorithmic solutions

Getting to quantum applications from the software side



We focus on hard computing problems



Quantum Machine Learning

Quantum Machine Learning

Machine Learning

- Large-scale quantum computers offer big advantages

Quantum Machine Learning

Machine Learning

- Large-scale quantum computers offer big advantages
  - We have developed many of these algorithms — Classifiers, Recommenders, q-means, …

    Quantum Algorithms for Deep Convolutional Neural Networks – ICLR 2020
    https://arxiv.org/abs/1911.01117
    Quantum Expectation-Maximization for Gaussian Mixture Models – ICML 2020
    https://arxiv.org/abs/1908.06657
    q-means: A quantum algorithm for unsupervised machine learning – NeurIPS 2019
    https://arxiv.org/abs/1812.03584
    Quantum algorithms for Second-Order Cone Programming and Support Vector Machines, Quantum 2021
    https://arxiv.org/abs/1908.06720
    Quantum Algorithms for feedforward neural networks – ACM ToQC 2020
    https://arxiv.org/abs/1812.03089
    Quantum classification of the MNIST dataset via Slow Feature Analysis – PRA 2020
    https://arxiv.org/abs/1808.09266
    Quantum gradient descent for linear systems and least squares – PRA 2020
    https://arxiv.org/abs/1704.04992
    Quantum recommendation systems (2017) – Innovations on TCS 2017
    https://arxiv.org/abs/1603.08675

Quantum Machine Learning

Machine Learning

- Large-scale quantum computers offer big advantages
  - We have developed many of these algorithms — Classifiers, Recommenders, q-means, ...
- We HAVE concrete avenues for bringing QML closer to reality
  - Data Loaders
  - Distance Estimators for Similarity Learning
  - Linear algebra for ML

Quantum Machine Learning

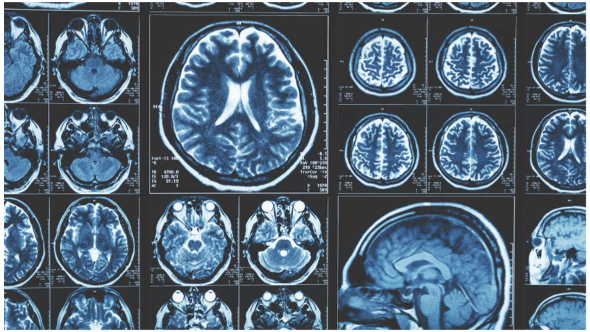
Machine Learning

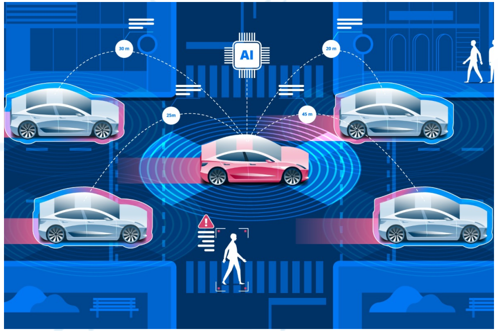
- Large-scale quantum computers offer big advantages
  - We have developed many of these algorithms — Classifiers, Recommenders, q-means, …
- We HAVE concrete avenues for bringing QML closer to reality
  - Data Loaders
  - Distance Estimators for Similarity Learning
  - Linear algebra for ML
- QML may offer: Efficiency, Accuracy, Interpretability, Trust, Energy savings

A first example

Classification

Data



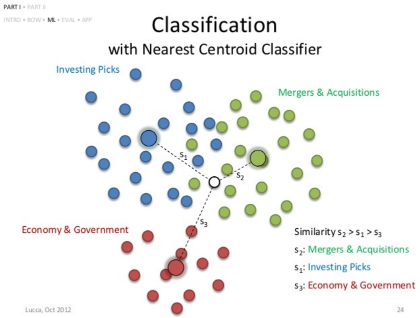


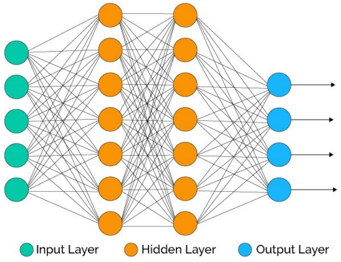
Classification

Data

→

Quantum algorithms



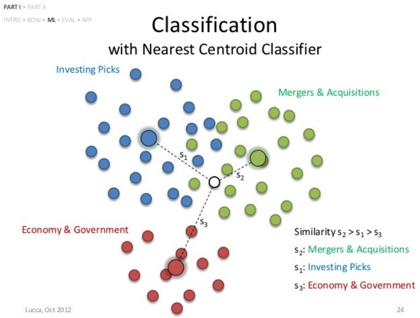


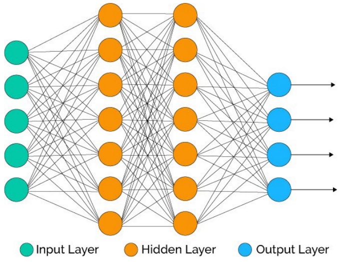
Classification

Data

→

Quantum algorithms





↓

Quantum software

```
# let's create some synthetic data
X, y = generate_data_clusters()

# let's run the quantum classifier
qlabels = fit_and_predict(X,y=y,model='QNearestCentroid')

#import NearestCentroid from scikit-learn for benchmarking
clabels = sklearn.neighbors.NearestCentroid().fit(X,y).predict(X)

print('Quantum labels\n',qlabels)
print('Classical labels\n',clabels)

# let's plot the data (only for dimension=2)
plot(X,qlabels,'QNearestCentroid')
plot(X,clabels,'KNearestCentroid')
```

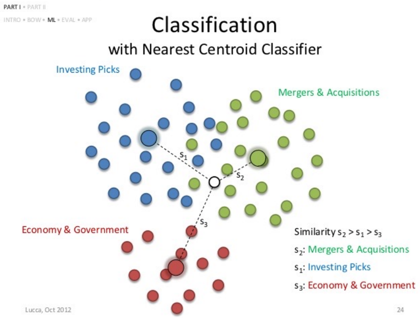
```
Quantum labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
Classical labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
```

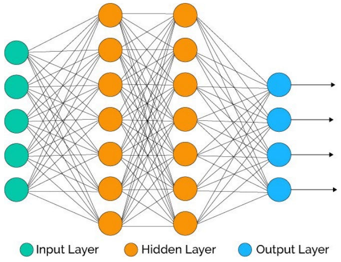
Classification

Data

→

Quantum algorithms





↓

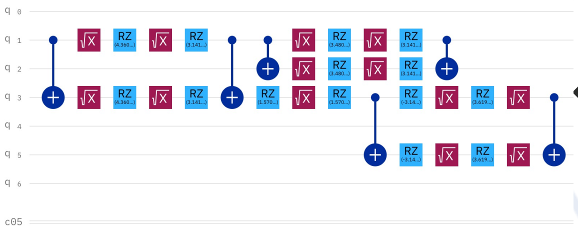
Quantum software

```
# let's create some synthetic data
X, y = generate_data_clusters()

# let's run the quantum classifier
qlabels = fit_and_predict(X,y=y,model='QNearestCentroid')

#import NearestCentroid from scikit-learn for benchmarking
clabels = sklearn.neighbors.NearestCentroid().fit(X,y).predict(X)

print('Quantum labels\n',qlabels)
print('Classical labels\n',clabels)

# let's plot the data (only for dimension=2)
plot(X,qlabels,'QNearestCentroid')
plot(X,clabels,'KNearestCentroid')
```

```
Quantum labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
Classical labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
```

←

Quantum circuits

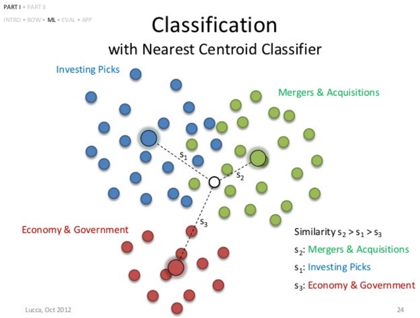


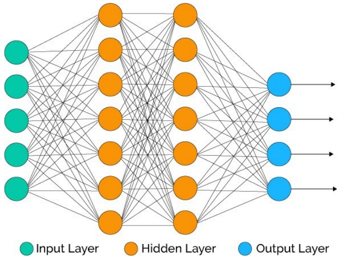
←

Classification

Data

Quantum algorithms



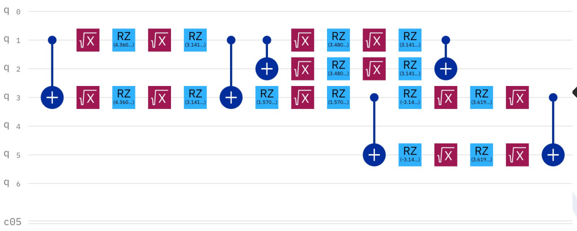


Quantum software

```
# let's create some synthetic data
X, y = generate_data_clusters()

# let's run the quantum classifier
qlabels = fit_and_predict(X,y=y,model='QNearestCentroid')

#import NearestCentroid from scikit-learn for benchmarking
clabels = sklearn.neighbors.NearestCentroid().fit(X,y).predict(X)

print('Quantum labels\n',qlabels)
print('Classical labels\n',clabels)

# let's plot the data (only for dimension=2)
plot(X,qlabels,'QNearestCentroid')
plot(X,clabels,'KNearestCentroid')
```

```
Quantum labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
Classical labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
```

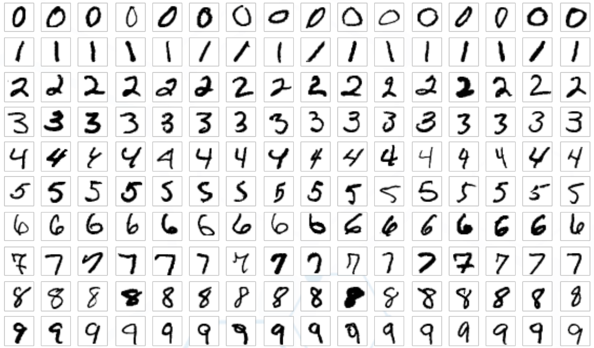
Quantum circuits

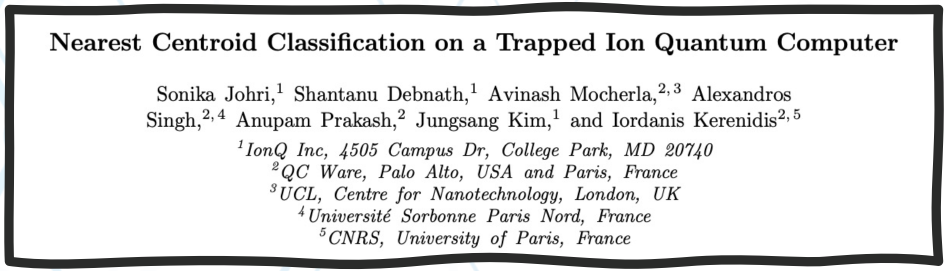


Quantum computer

Classification

Data

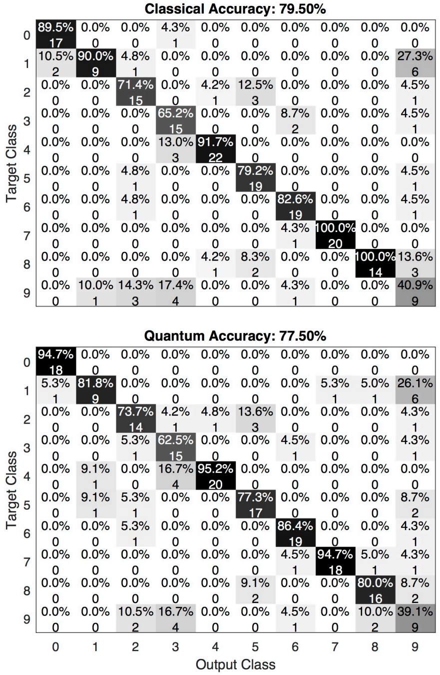



Nearest Centroid Classification on a Trapped Ion Quantum Computer

Sonika Johri,[1] Shantanu Debnath,[1] Avinash Mocherla,[2,3] Alexandros Singh,[2,4] Anupam Prakash,[2] Jungsang Kim,[1] and Iordanis Kerenidis[2,5]

[1]*IonQ Inc, 4505 Campus Dr, College Park, MD 20740*
[2]*QC Ware, Palo Alto, USA and Paris, France*
[3]*UCL, Centre for Nanotechnology, London, UK*
[4]*Université Sorbonne Paris Nord, France*
[5]*CNRS, University of Paris, France*

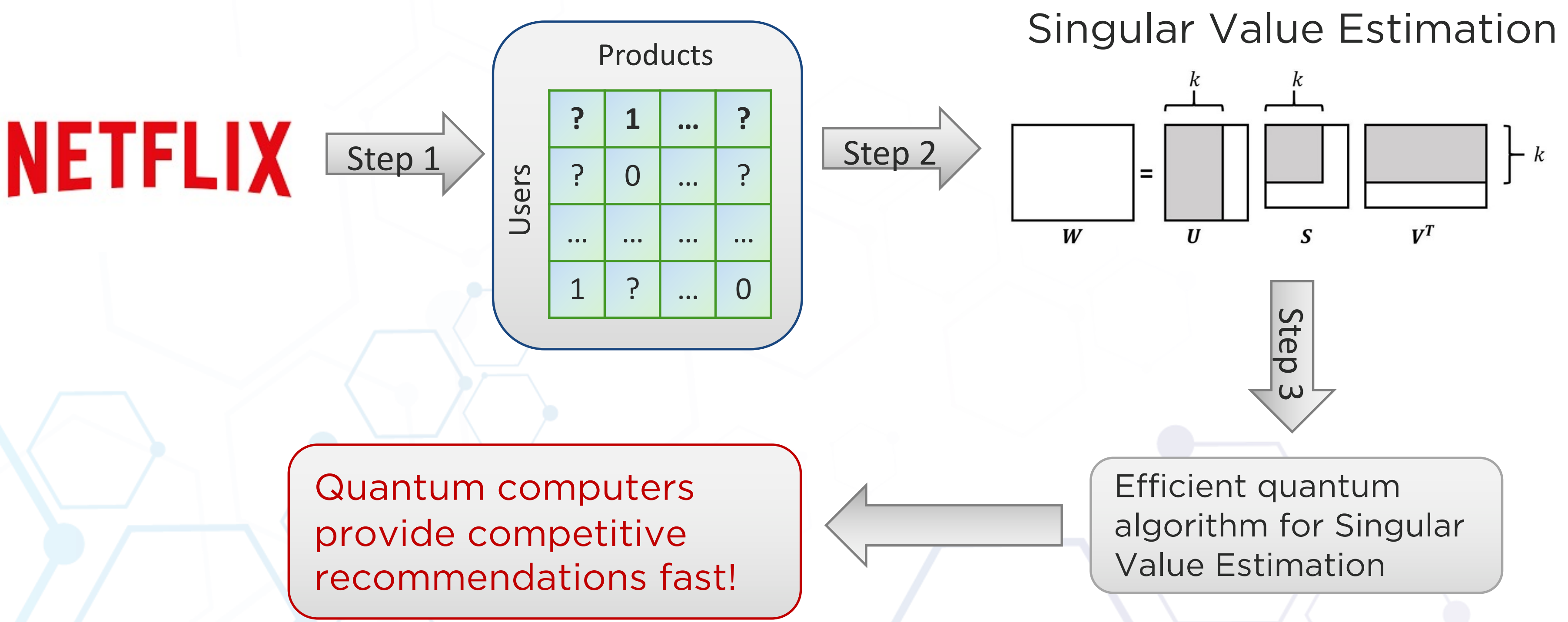
Results



An 11-qubit QC can recognise 8 out of 10 handwritten digits

Many more applications

Recommendation Systems [Kerenidis, Prakash, ITCS 17]



Clustering

q-means++ [Kerenidis,Landman,Luongo, Prakash NeurIPS 2019]

Input: N points in d-dimensions (quantum access)
Output: K clusters/centroids

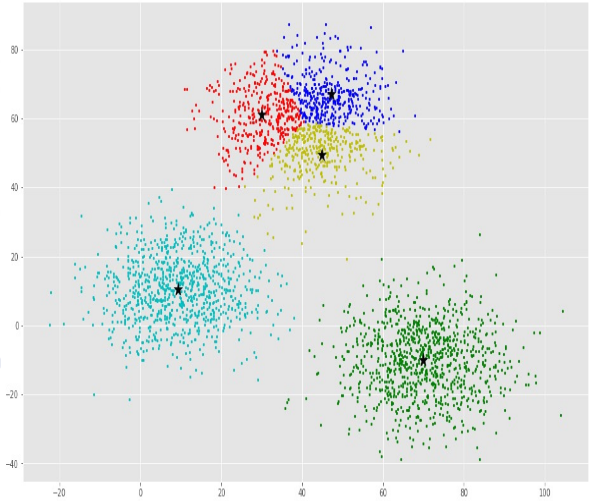
1. Start with some initial centroids (e.g. ++-method)

Repeat until convergence

2. For all points in superposition

   estimate distances to centroids quantumly
   and assign to nearest centroid

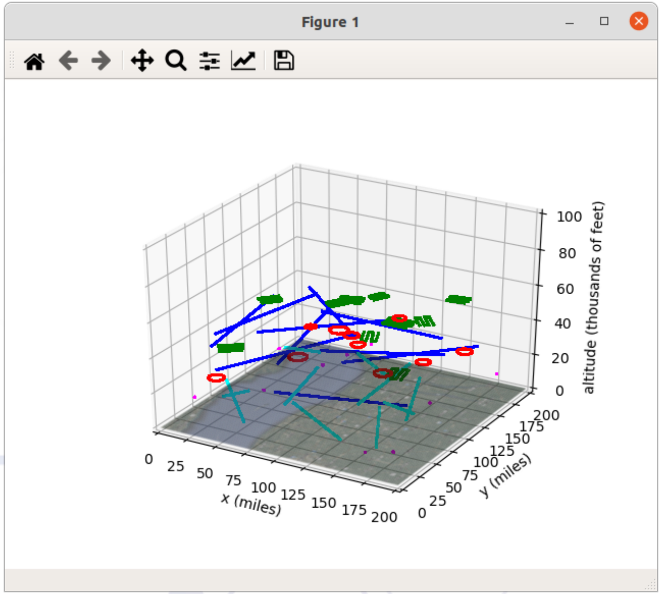
3. Update the centroids
   i. Quantum linear algebra to find new centroid
   ii. Tomography to recover classical description



Clustering: QC Ware-AFRL collaboration

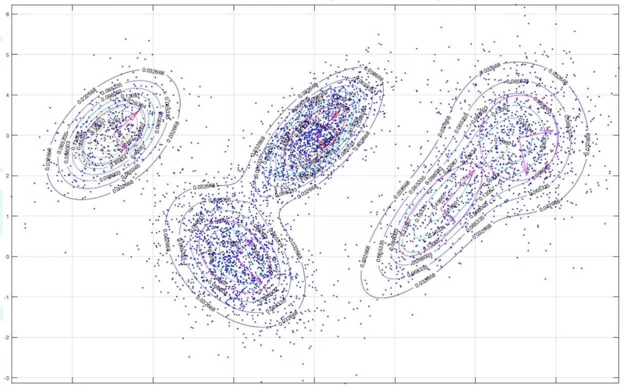
q-means++ [NeurIPS2019]

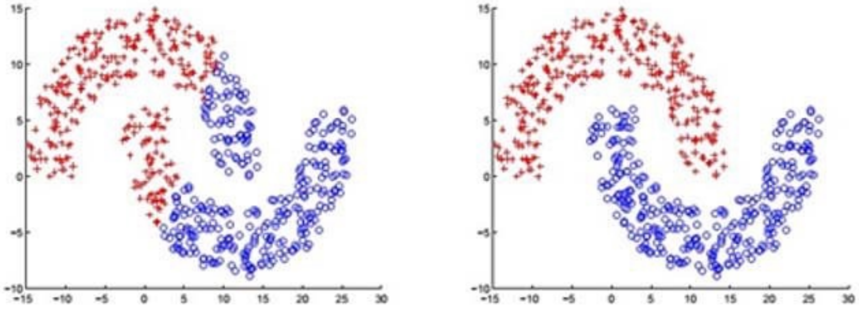
- NISQ implementation
- Simulations achieve comparable performance
- Faster quantum running times
- Preparing for hardware demonstration



Unsupervised learning

- Feature selection
  - Methods related to determinantal sampling/volume sampling
- Expectation Maximization for Gaussian Mixture Models [K,Luongo,Prakash ICML 2020]
- Spectral Clustering [K, Landman PRA 2021]





(a) K-means

(b) Spectral Clustering

Reinforcement Learning

Quantum Policy Iteration [Cherrat,Kerenidis,Prakash 2021]

Input: states S, actions A, transitions P, Rewards R
Output: policy π

1. Start with some initial π_0

Repeat until convergence

2. solve $(I-\gamma P^{\pi})Q=R$ quantum linear systems
3. update π from Q by measurements

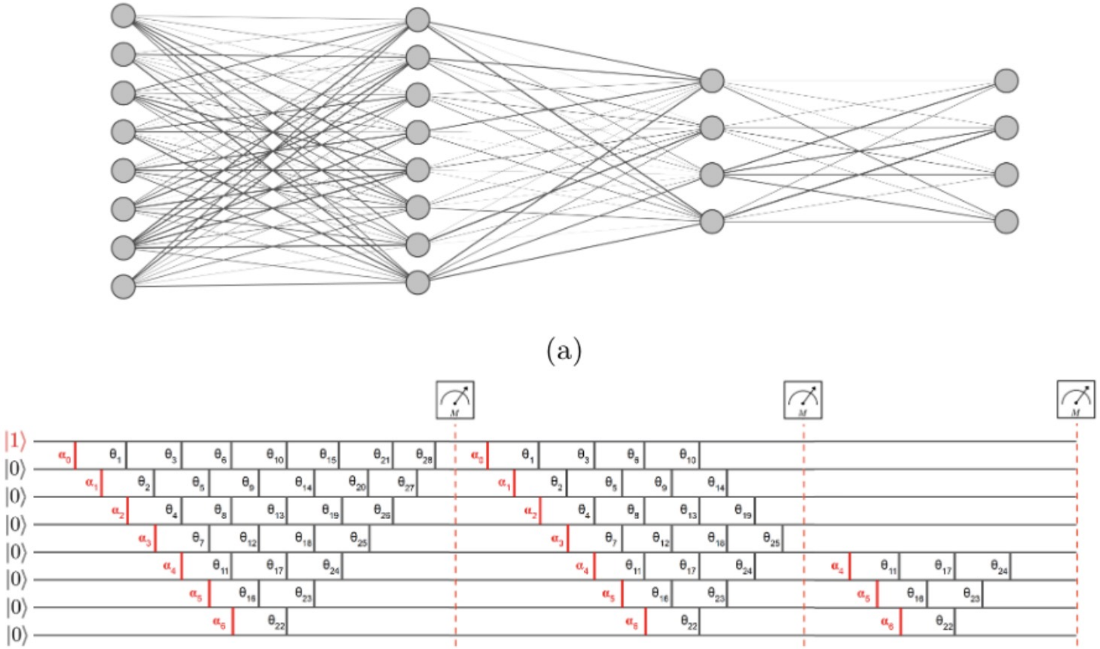
Remarks

"No input" / Well-conditioned / ℓ^{∞} guarantees

Quantum Neural Networks

Quantum Orthogonal NNs

- New classical training in $O(n^2)$
- NISQ implementations
- provable efficiency
- A new optimization landscape

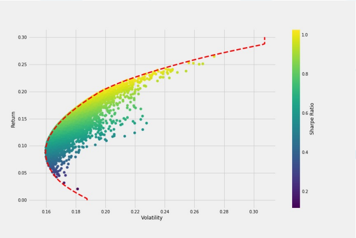


(a)

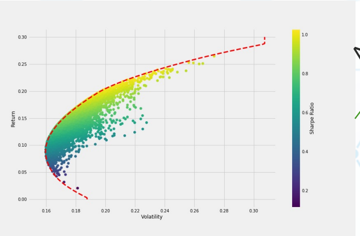
(b)

Quantum Optimization

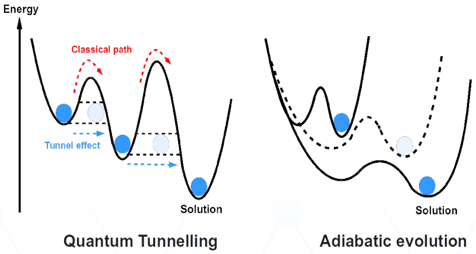
Optimization

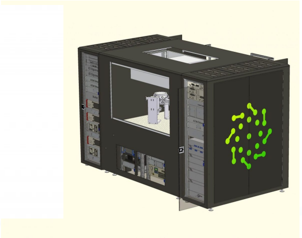


Optimization

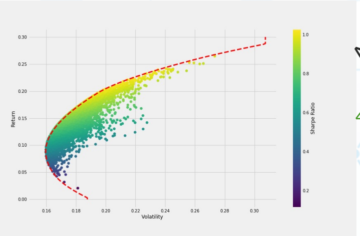


Analog Combinatorial opt



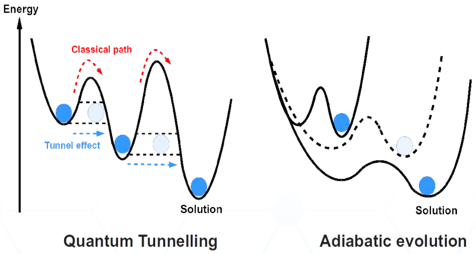


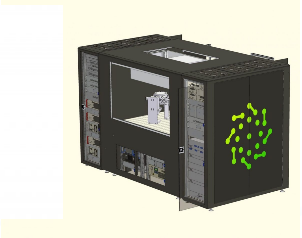
Optimization

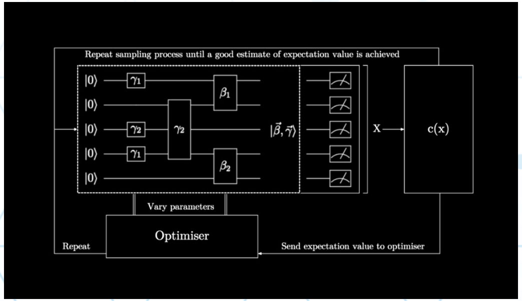


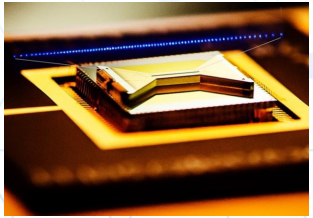
Analog Combinatorial opt

Discrete Combinatorial opt

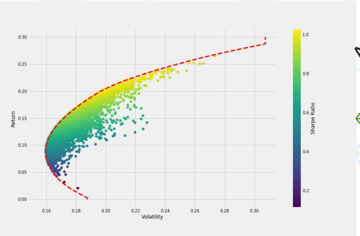




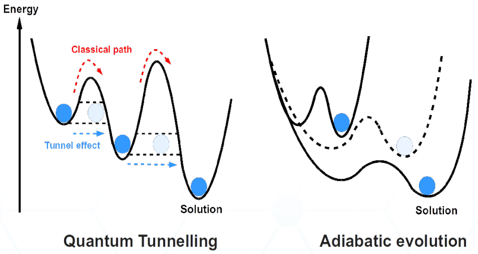


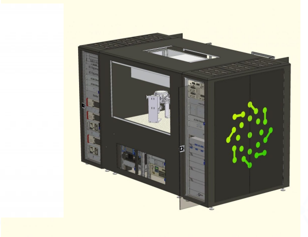


Optimization

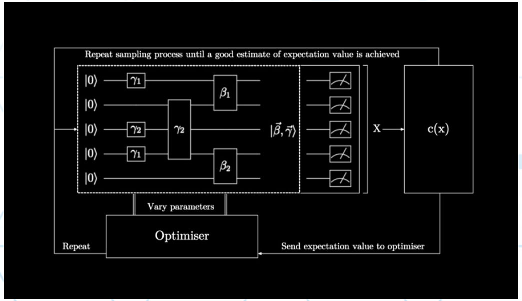


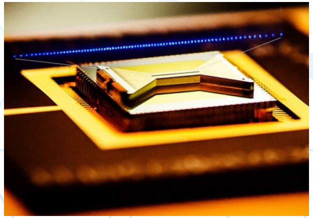
Analog Combinatorial opt



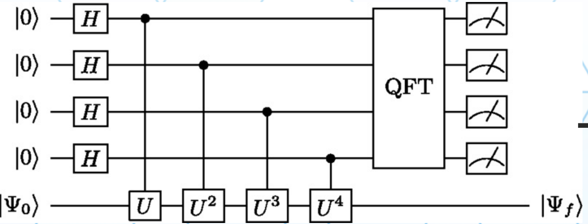


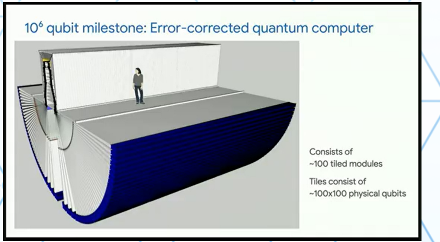
Discrete Combinatorial opt





Convex opt

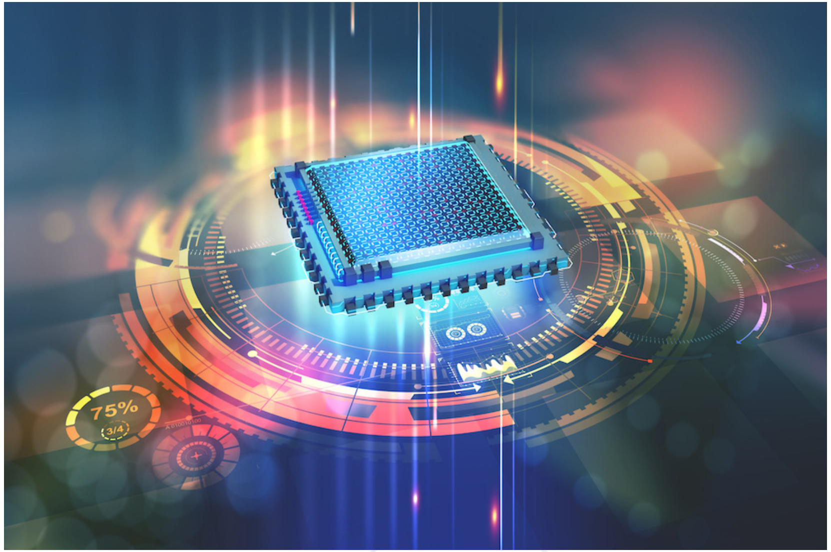




[Kerenidis,Prakash 2018]
[Kerenidis,Prakash,Szilágyi 2020]

Conclusions

- ✓ Quantum technologies will very soon impact most industry sectors
- ✓ Competitive advantage comes from algorithms and models
- ✓ The time to engage with quantum technologies is now




Thank you!
Iordanis Kerenidis
jkeren@irif.fr, iordanis.kerenidis@qcware.com
cnrs
dépasser les frontières
QCWARE