Quantum Machine Learning (for Physics)

Mohammad Hassanshahi
University College London

30 Dec 2023

IUT physics seminar

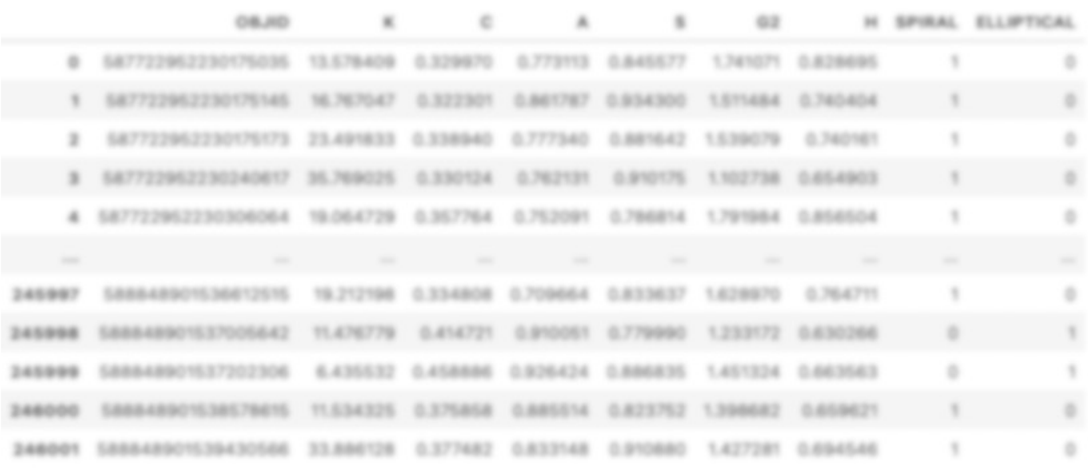
Outline

- Machine Learning (ML) + examples
- Quantum Machine Learning + examples
- Challenges of Quantum Machine Learning

Machine Learning

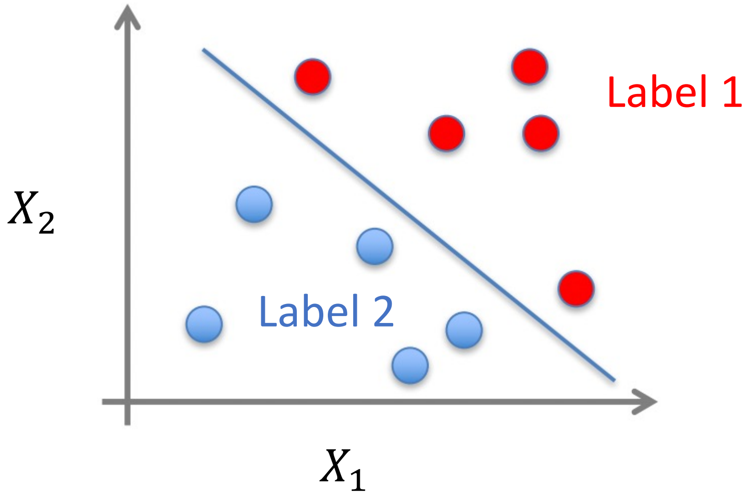
What does ML do?

- Task: Learn from given data to predict new data
  - Learn = find a model



- Model found by optimizing a *loss function*

- Examples: classification and regression



Example: Support Vector Machine

- Task:
  - classifying data with a hyperplane.
  - penalty for misclassified data.

- Optimization:
  - Primal problem:

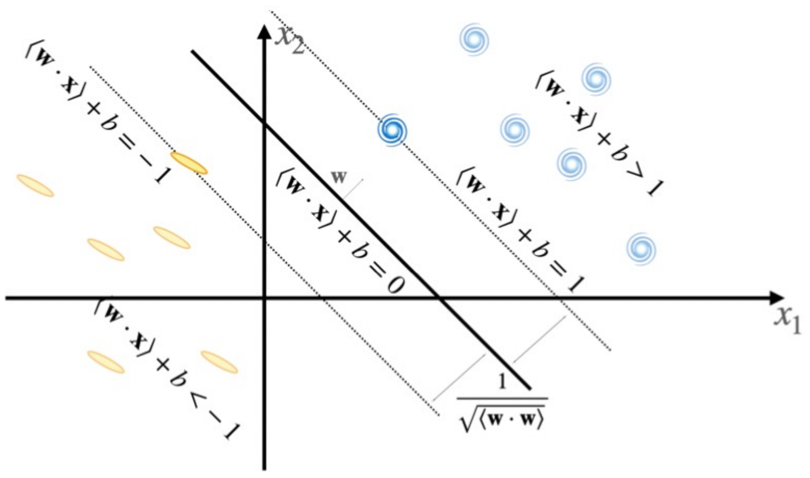
$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda\|\mathbf{w}\|^2$$
$$\text{subject to } y_i\left(\mathbf{w}^\mathsf{T}\mathbf{x}_i - b\right) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, \text{ for all } i.$$

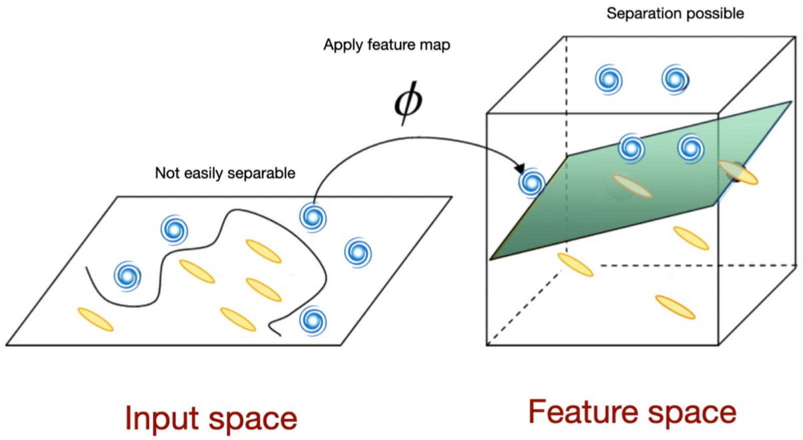
  - Dual problem:

$$\text{maximize } f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (\mathbf{x}_i^\mathsf{T}\mathbf{x}_j) y_j c_j,$$
$$\text{subject to } \sum_{i=1}^{n} c_i y_i = 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i.$$

- If data not linearly separable: kernel trick

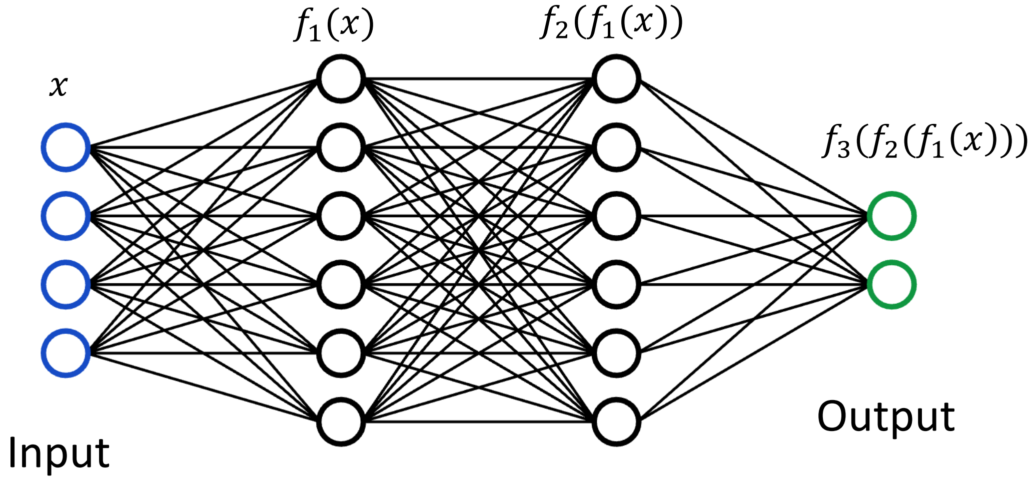
$$\text{maximize } f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (\varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_j)) y_j c_j$$
$$= \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i k(\mathbf{x}_i, \mathbf{x}_j) y_j c_j$$





Example: Neural Networks (NN)

- In NN, output is a function of input.
- Weights to be optimised optimised
- Min of loss is found with gradient descent method
- Good scaling of derivative calculation (with backpropagation method)



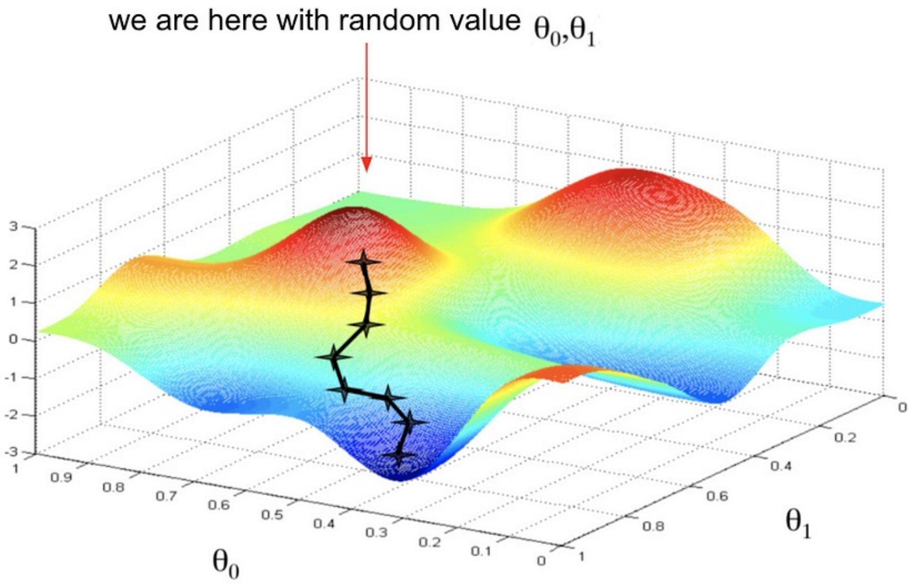
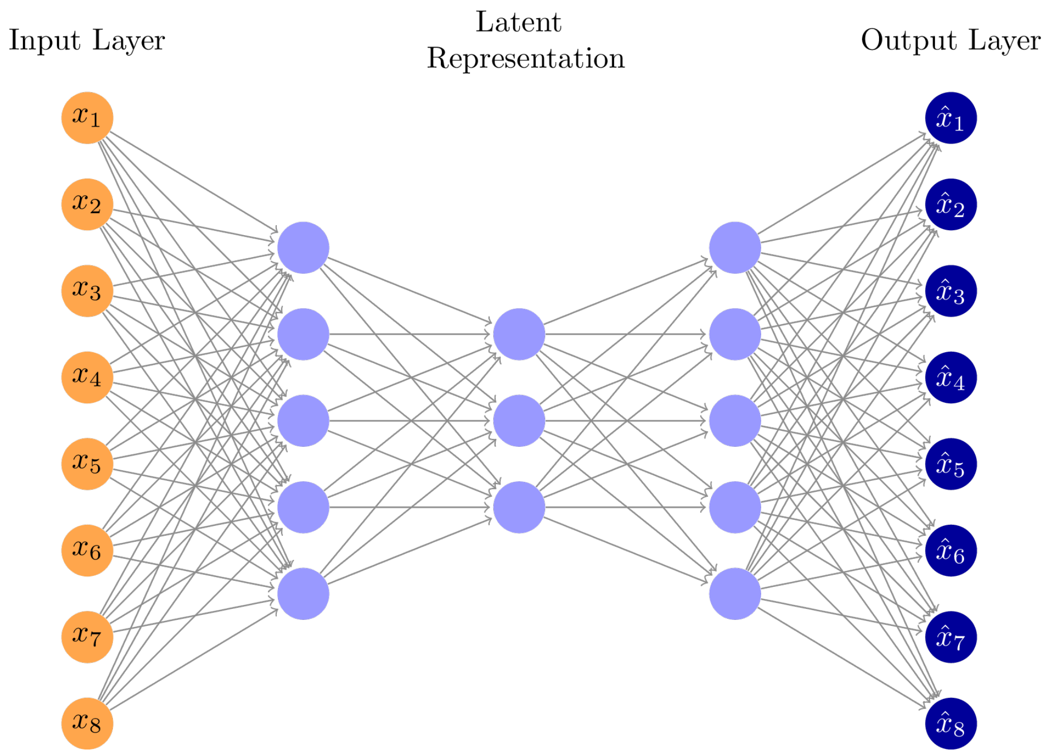
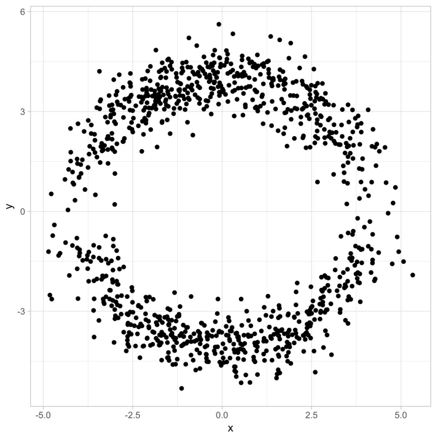


Photo: https://www.kdnuggets.com/2020/05/5-concepts-gradient-descent-cost-function.html

Examples: Autoencoders

- Neural network with a bottle-neck in the middle
- Require output to be as similar as possible to input
- Used for dimensionality reduction
- Also used for anomaly detection
  - e.g. in HEP: model-independent searches



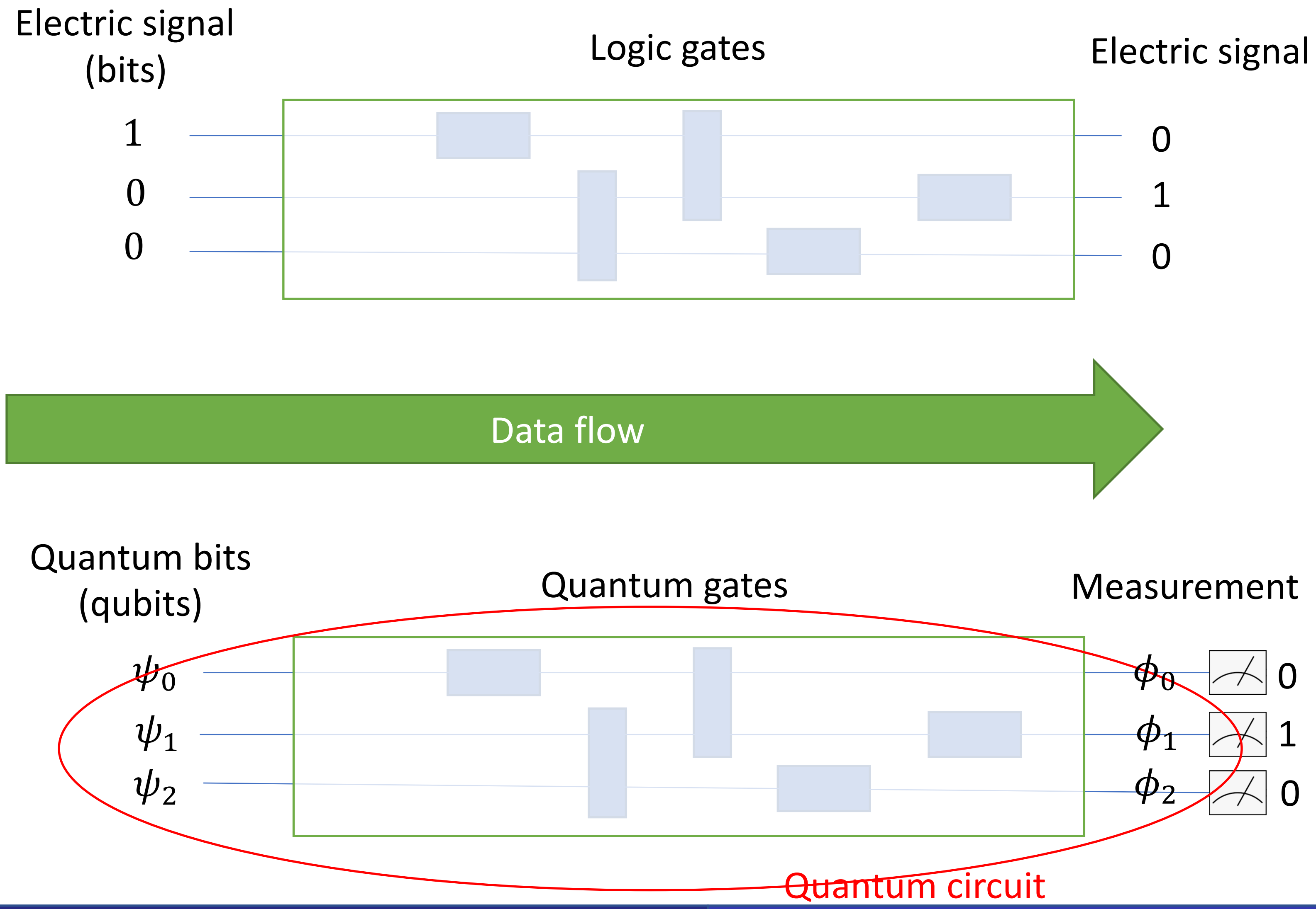


Quantum Machine Learning

Quantum vs classical

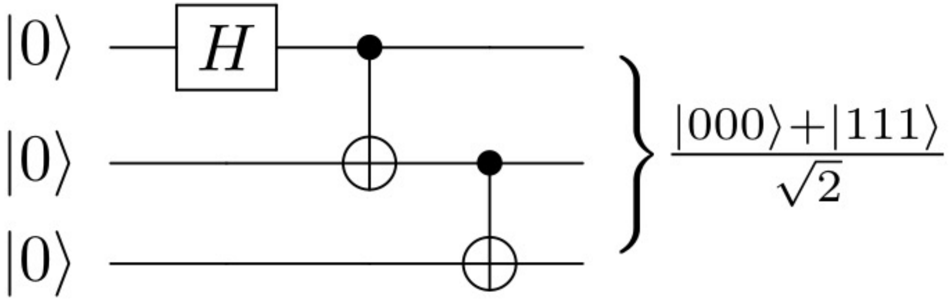
Classical (=conventional) computers

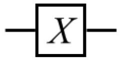
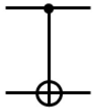
Quantum computers



Quantum circuit example

➢ Making Greenberger–Horne–Zeilinger (GHZ) state



Gate	Notation	Matrix
NOT (Pauli-X)		$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Pauli-Z		$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
Hadamard		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
CNOT (Controlled NOT)		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$

Quantum properties

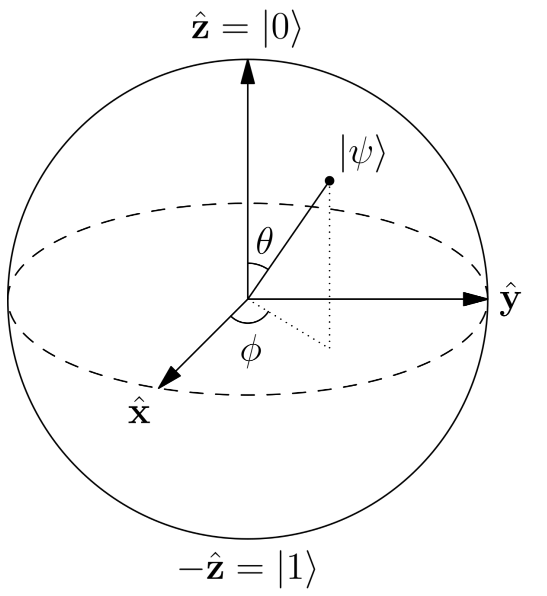
Two important quantum properties

Superposition

Entanglement

A quantum bit (qubit)

- Quantum bit:
  - Instead of 0 or 1, $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
  - Equal to the points on a sphere



- Quantum gates: changing α_i
  - E.g. $|\psi\rangle = \sqrt{\frac{1}{7}}|0\rangle + \sqrt{\frac{6}{7}}|1\rangle$ ----------applying gate------> $|\psi\rangle = \sqrt{\frac{3}{5}}|0\rangle + \sqrt{\frac{2}{5}}|1\rangle$
  - As if rotating the vector in the sphere

- Measurement:
  - We need 0 or 1 to work with
  - All/Some of qubits are measured at the end

More qubits

- Two qubits:
  - Notation: e.g. $|01\rangle$ means the first qubit is 0 and second qubit is 1.
  - $|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle \rightarrow$ 4 basis states

- Three qubits: $|\psi\rangle = \alpha_0|000\rangle + \alpha_1|001\rangle + \cdots + \alpha_8|111\rangle \rightarrow$ 8 basis states

- N qubits: $|\psi\rangle = \alpha_0\big|0 \ldots 0\rangle + \cdots + \alpha_{2^N}\big|1 \ldots 1\rangle \rightarrow 2^N$ basis states

- Quantum gates: changing α_i

- Measurement

Quantum "advantage"

- Applying gates on a $|\psi\rangle = \alpha_0 \big|0 \ldots 0\rangle + \cdots + \alpha_{2^N} \big|1 \ldots 1\rangle$:
  - Classical computers: **Simulating** quantum circuits → **keeping track of 2^N coefficients** → Could get **intractable**.
  - Quantum computers: Performed **naturally**.
  - Quantum advantage!

- Is $\alpha_0 \big|0 \ldots 0\rangle + \cdots + \alpha_{2^N} \big|1 \ldots 1\rangle$ **always** interactable in classical (conventional) computers?
  - Answer: No!
  - When is it intractable?

- Product vs entangled state

  e.g. of product: $|\psi\rangle = ½\,(|00\rangle + |10\rangle + |01\rangle + |11\rangle) = \frac{1}{2}\,(|0\rangle + |1\rangle)\ (|0\rangle + |1\rangle)$

  e.g. of entangled: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

- Entanglement is (almost) always essential for potential quantum advantage

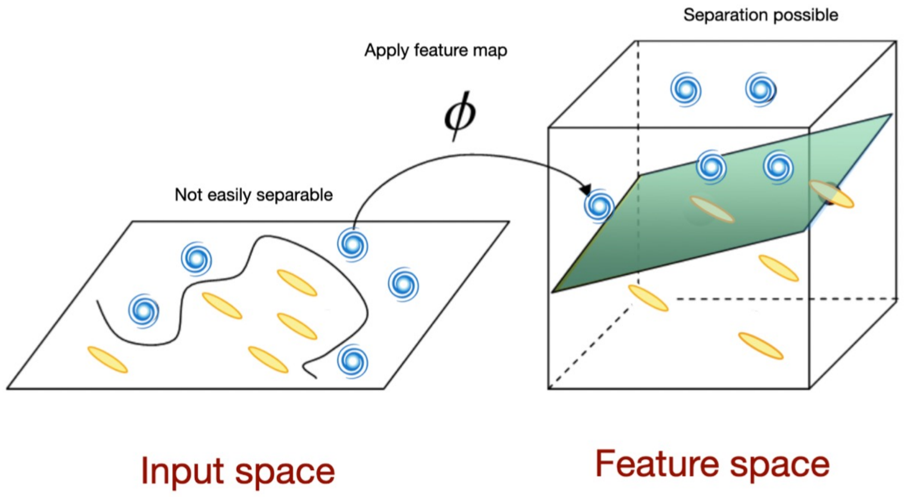
Example: Quantum SVM

- Compute kernels in quantum computers

- Features are mapped into the Hilbert space of qubits
  - Feature vector $\boldsymbol{x}$ is mapped to $\boldsymbol{x} \rightarrow \rho(\boldsymbol{x}) = |\psi(\boldsymbol{x})\rangle\langle\psi(\boldsymbol{x})|$, where $|\psi(\boldsymbol{x})\rangle = \mathcal{U}(\boldsymbol{x})|0\rangle^n$.
  - Kernel is defined as $\mathrm{k}(\boldsymbol{z}, \boldsymbol{x}) \coloneqq \mathrm{tr}\big(\rho(\boldsymbol{z})\rho(\boldsymbol{x})\big) = |\langle\psi(\boldsymbol{z})|\psi(\boldsymbol{x})\rangle|^2 = \left|\langle 0|\mathcal{U}^\dagger(\boldsymbol{z})\mathcal{U}(\boldsymbol{x})|0\rangle\right|^2$.

- There are infinite possible feature maps. Interesting ones:
  - are hard to simulate classically
  - are computable in *near-term* devices
  - provide good classification performance

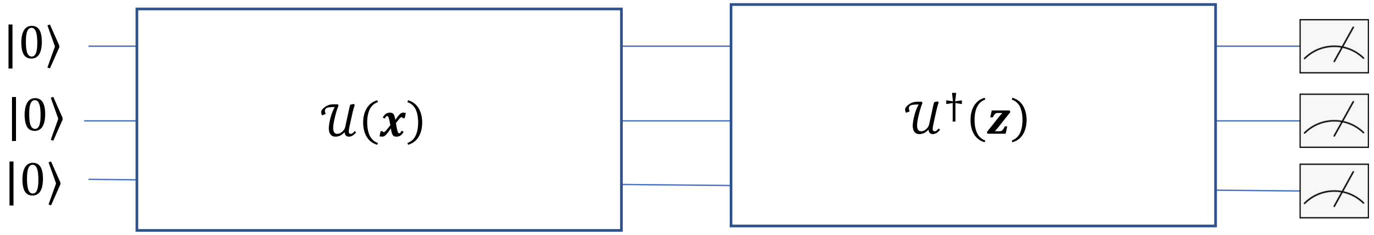
- Example of U → rotating in Bloch sphere



Example: Quantum SVM

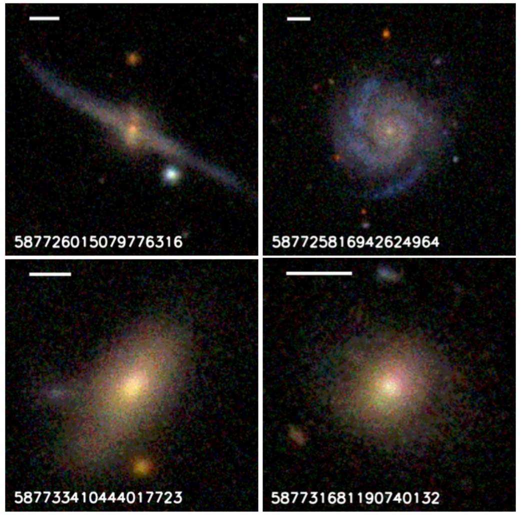
- The kernel is then **estimated** by measuring $\mathcal{U}^\dagger(\boldsymbol{z})\mathcal{U}(\boldsymbol{x})|0\rangle^n$ for O(1000) times.

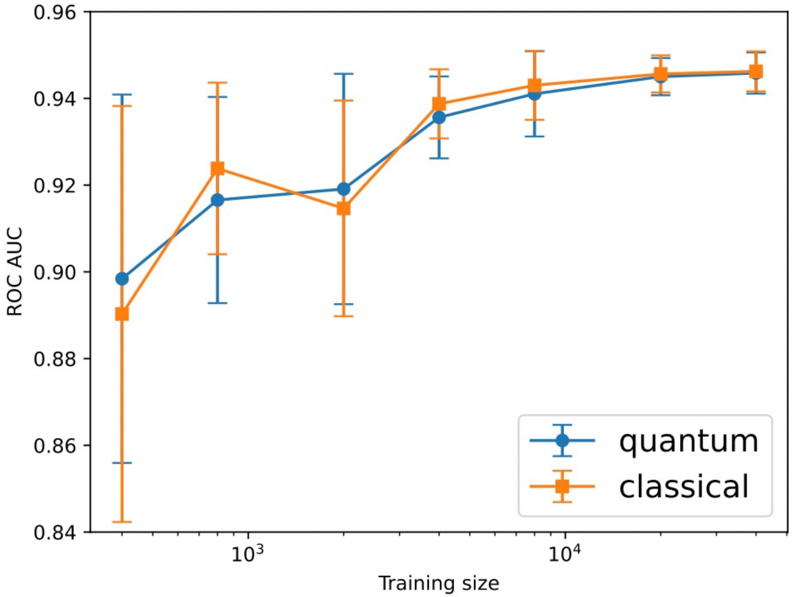
- Kernel $= \dfrac{\#\,00 \ldots 00}{all\ meas.}$



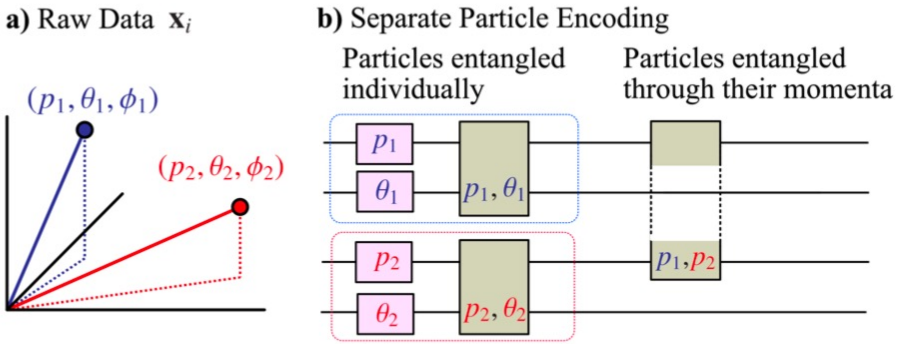
Example:
Galaxy classification

(Also tested a small sample on a real device)





Example: B meson decay

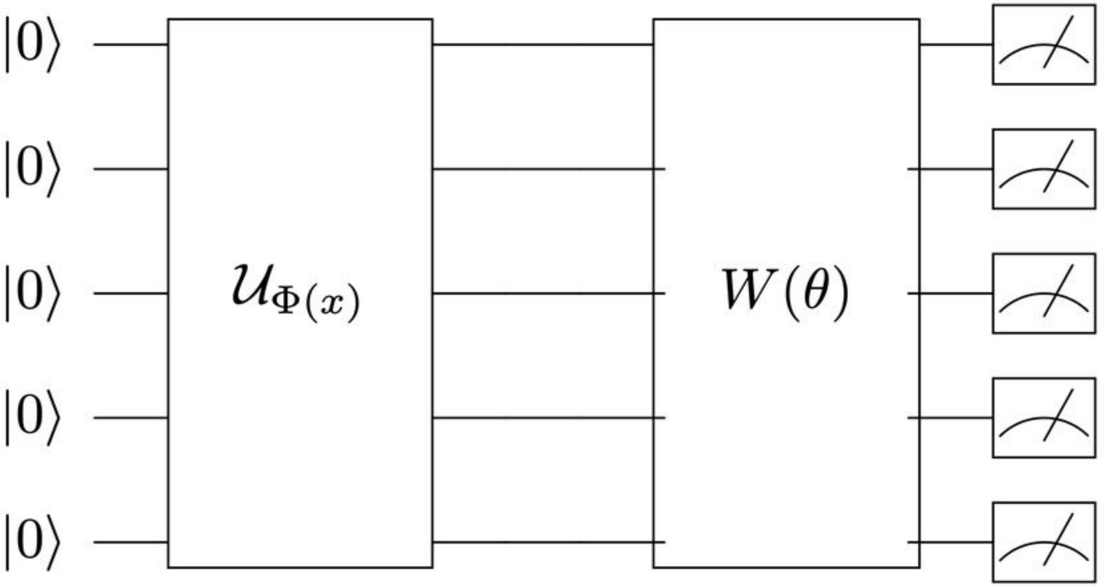


Encoding Circuit	Accuracy	AUC
Combinatorial Encoding	0.762	0.822
Separate Particle Encoding	0.776	0.835
Bloch Sphere Encoding	0.764	0.836
Separate Particle with Bloch	0.771	0.848
Classical RBF Kernel SVM	0.728	0.793
XGBoost	0.590	0.621

Quantum Neural Network

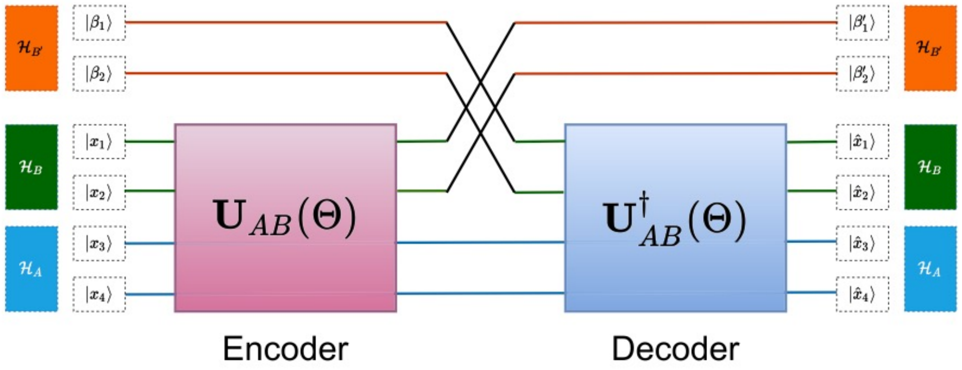
- Some similarities with (conventional) neural network

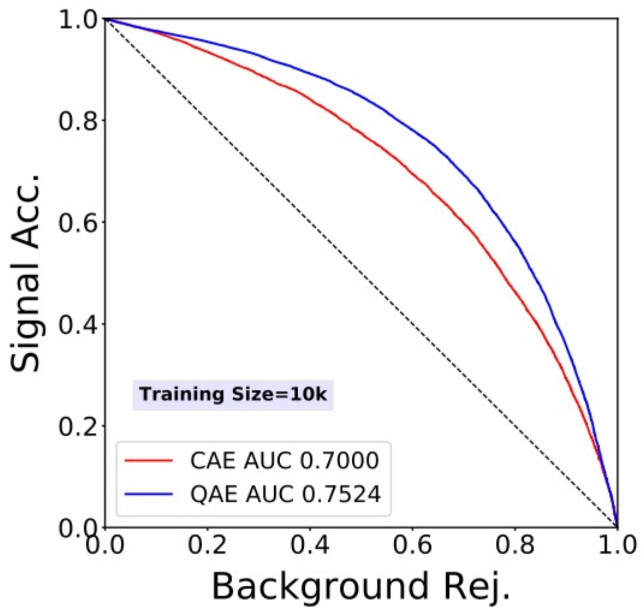
- Parameter(s) θ updated until optimum found



Quantum Autoencoder

- Similar to classical autoencoder



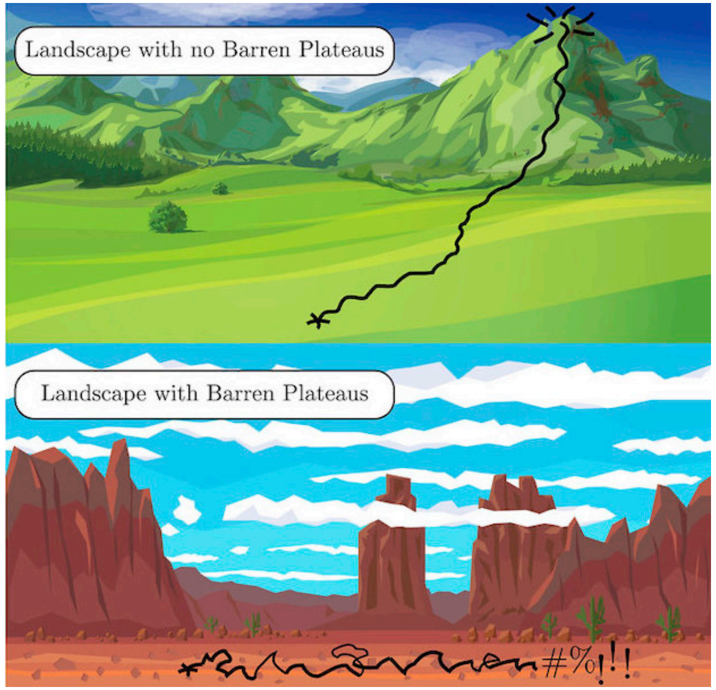


Example: Bkg: Z ->nu nu
Sig: H-> two dark matter

Challenges

Challenges

- Noise
- Barren plateau
- Backpropagation scaling
- Exponential concentration



Summary

- Quantum computing algorithms can be used for machine learning tasks

- Data are mapped to Hilbert space and the optimized model is found by minimizing a loss function

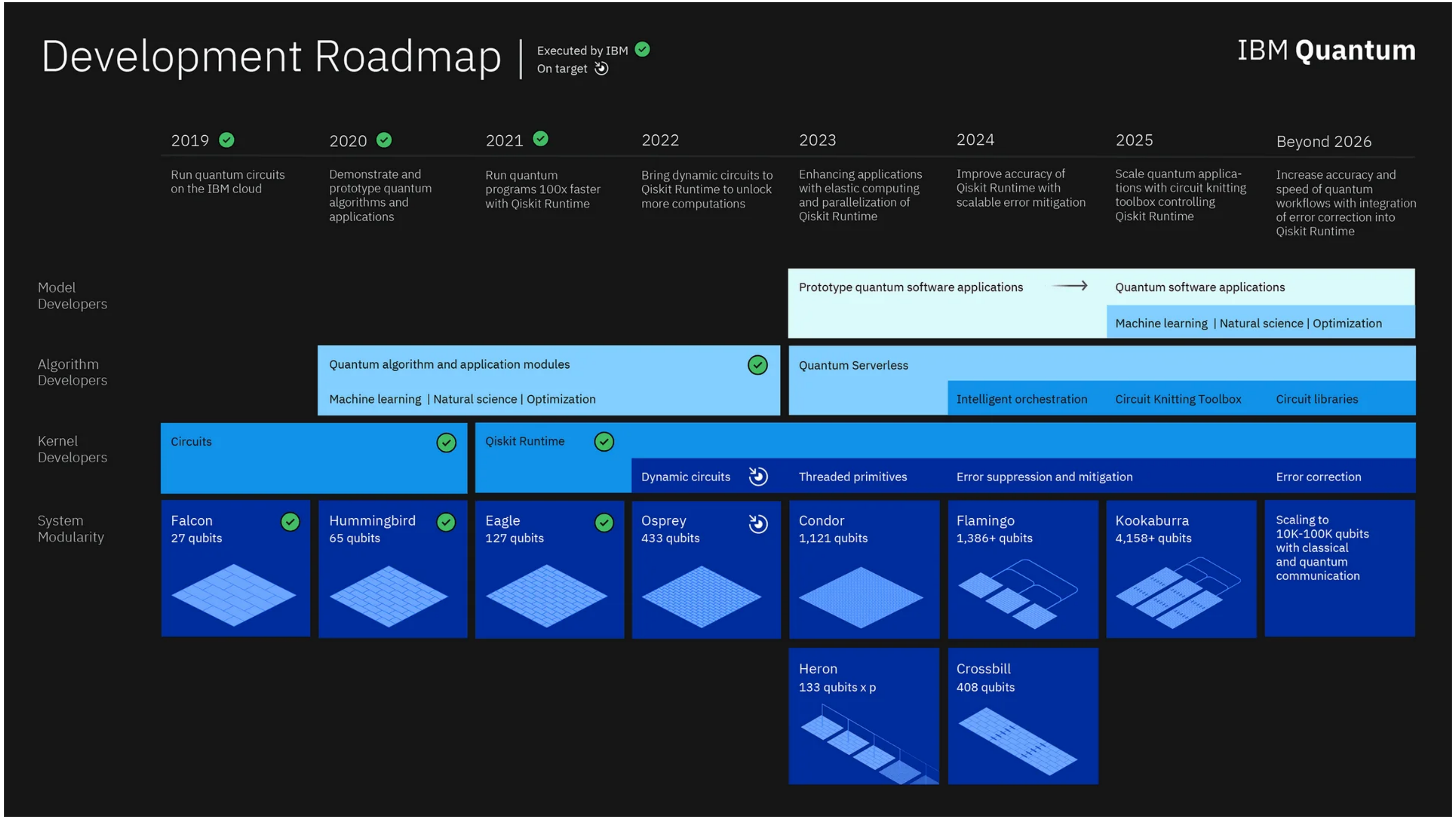
- Such tasks could in principle be intractable in classical computers

- However, there are challenges ahead!

Thank you!

Backup

IBM Plan



Development Roadmap | Executed by IBM On target
IBM Quantum
2019
Run quantum circuits on the IBM cloud
2020
Demonstrate and prototype quantum algorithms and applications
2021
Run quantum programs 100x faster with Qiskit Runtime
2022
Bring dynamic circuits to Qiskit Runtime to unlock more computations
2023
Enhancing applications with elastic computing and parallelization of Qiskit Runtime
2024
Improve accuracy of Qiskit Runtime with scalable error mitigation
2025
Scale quantum applications with circuit knitting toolbox controlling Qiskit Runtime
Beyond 2026
Increase accuracy and speed of quantum workflows with integration of error correction into Qiskit Runtime
Model Developers
Prototype quantum software applications
Quantum software applications
Machine learning | Natural science | Optimization
Algorithm Developers
Quantum algorithm and application modules
Machine learning | Natural science | Optimization
Quantum Serverless
Intelligent orchestration
Circuit Knitting Toolbox
Circuit libraries
Kernel Developers
Circuits
Qiskit Runtime
Dynamic circuits
Threaded primitives
Error suppression and mitigation
Error correction
System Modularity
Falcon 27 qubits
Hummingbird 65 qubits
Eagle 127 qubits
Osprey 433 qubits
Condor 1,121 qubits
Flamingo 1,386+ qubits
Kookaburra 4,158+ qubits
Scaling to 10K-100K qubits with classical and quantum communication
Heron 133 qubits x p
Crossbill 408 qubits

Superconducting qubit