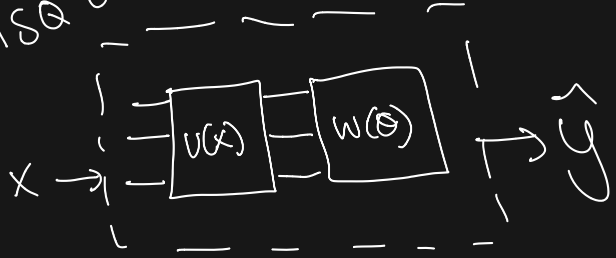
An introduction to quantum machine learning

Amira Abbas

IBM Quantum, University of KwaZulu-Natal

NISQ era



$\langle 0|M|0\rangle$

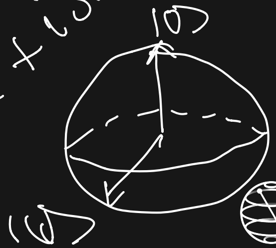
Barren plateaus!

θ

$\hat{H}$

$f(z)$

$\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}$



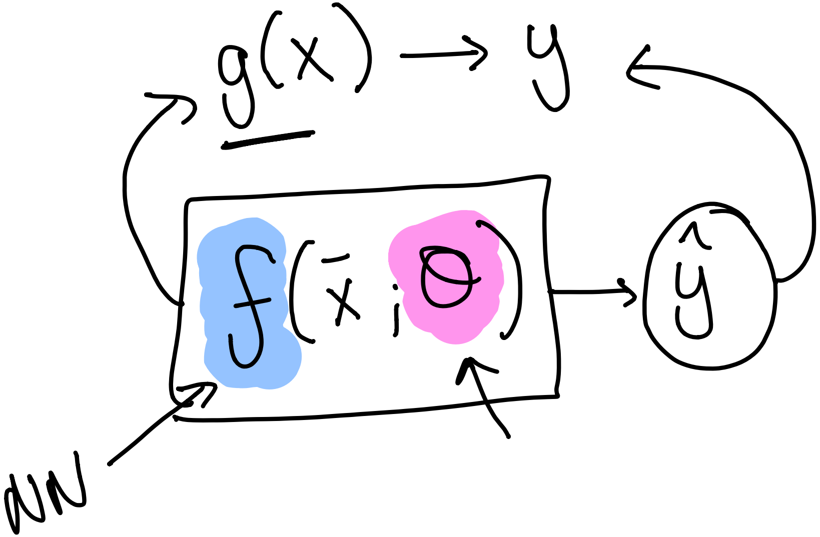
$|\psi\rangle \otimes |\psi\rangle$

QML vs ML

Qiskit

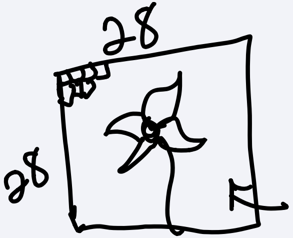
Classical machine learning

“Learning patterns from data in order to draw inferences”



g(x)
y
f(x̄; θ)
ŷ
NN

DATA



784 pixel $\in [-1, 1]$

dark light

$$\vec{x}_1 = \begin{array}{c} 1 \\ 2 \\ \vdots \\ \\ 784 \end{array} \begin{bmatrix} 0.5 \\ -0.2 \\ \vdots \\ \\ 0.2 \end{bmatrix}$$

features = dimension d

DATA

MODEL

$\vec{x}_i \longrightarrow \boxed{f(\vec{x}_i ; \theta)} \longrightarrow \hat{y} \in \begin{matrix} -1 & \text{DOG} \\ +1 & \text{CAT} \end{matrix}$

DATA

MODEL

COST

$\hat{y}$ compares y

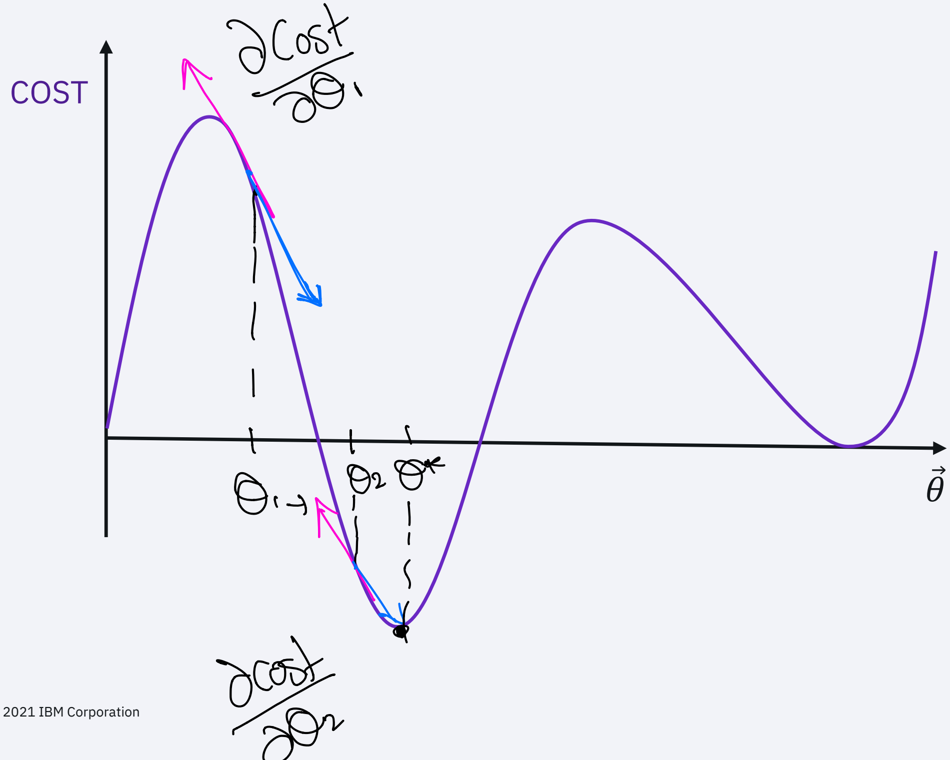
$$MSE := (\hat{y} - y)^2$$

$$(-1 - 1)^2 = 4$$

$$(1 - 1)^2 = 0$$

How do we optimize our model?

How do we optimize our model?



Gradient-descent

How do we optimize our model?

Gradient-descent

$$\theta^{i+1} = \theta^{i} - \eta \frac{\partial C}{\partial \theta^{i}}$$

Quantum machine learning

Hilbert space is a big place!

- Carlton Caves

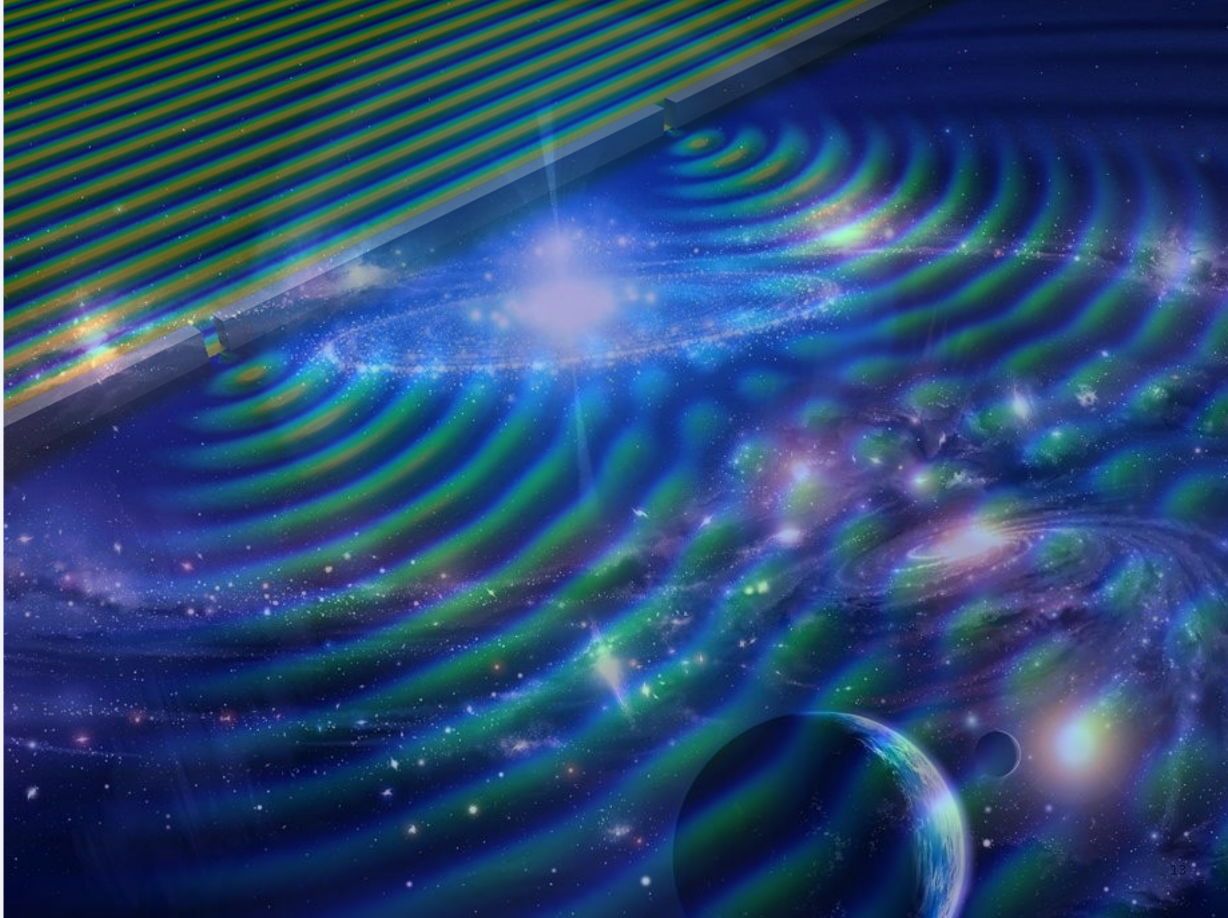
With just 275 qubits, we can represent more states than the number of atoms in the observable universe

$$2^{275}$$

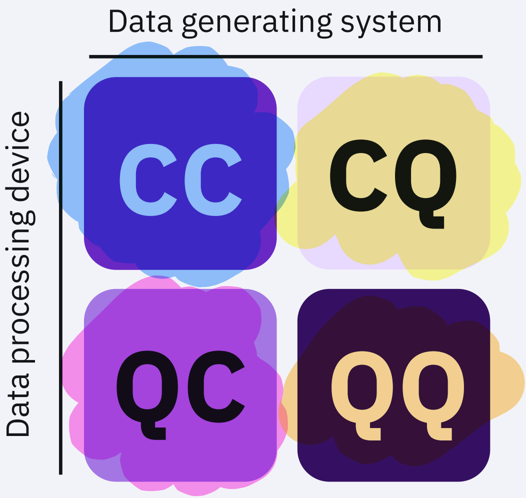
Hilbert space is a big place!

- *Carlton Caves*

Interference



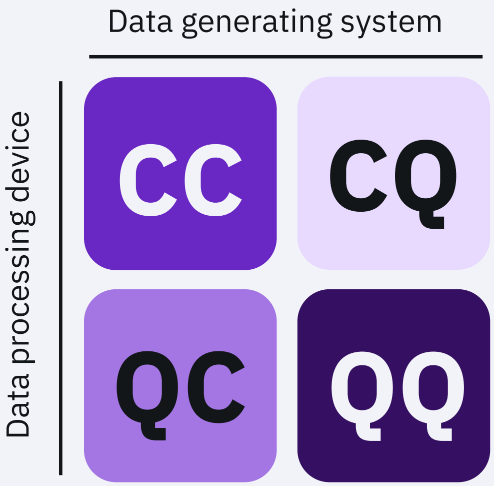
Quantum machine learning



C – *classical*, Q – *quantum*

Image credit: *Maria Schuld and Francesco Petruccione. Supervised learning with quantum computers. Vol. 17. Springer, 2018.*

Quantum machine learning



C – *classical*, Q – *quantum*

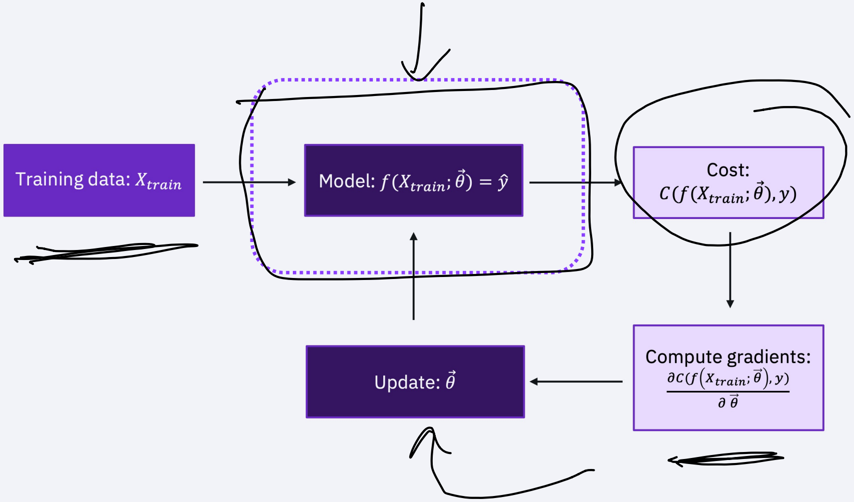


Image credit: *Maria Schuld and Francesco Petruccione. Supervised learning with quantum computers. Vol. 17. Springer, 2018.*

Near-term vs fault-tolerant

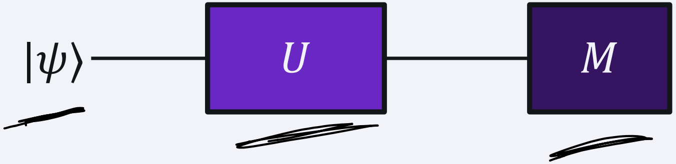
Noisy, error-prone, small devices

What can we do now?

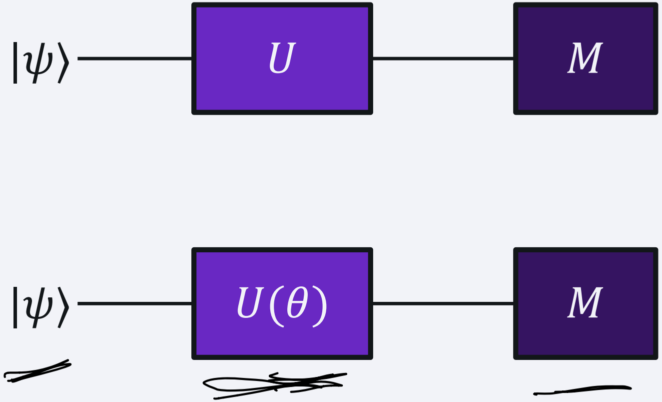
Variational models

Variational models

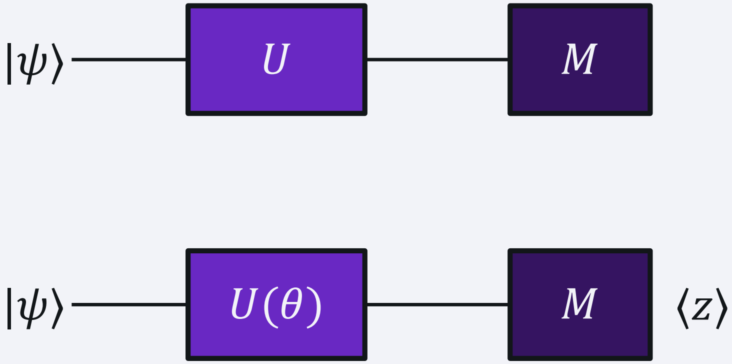
Variational models



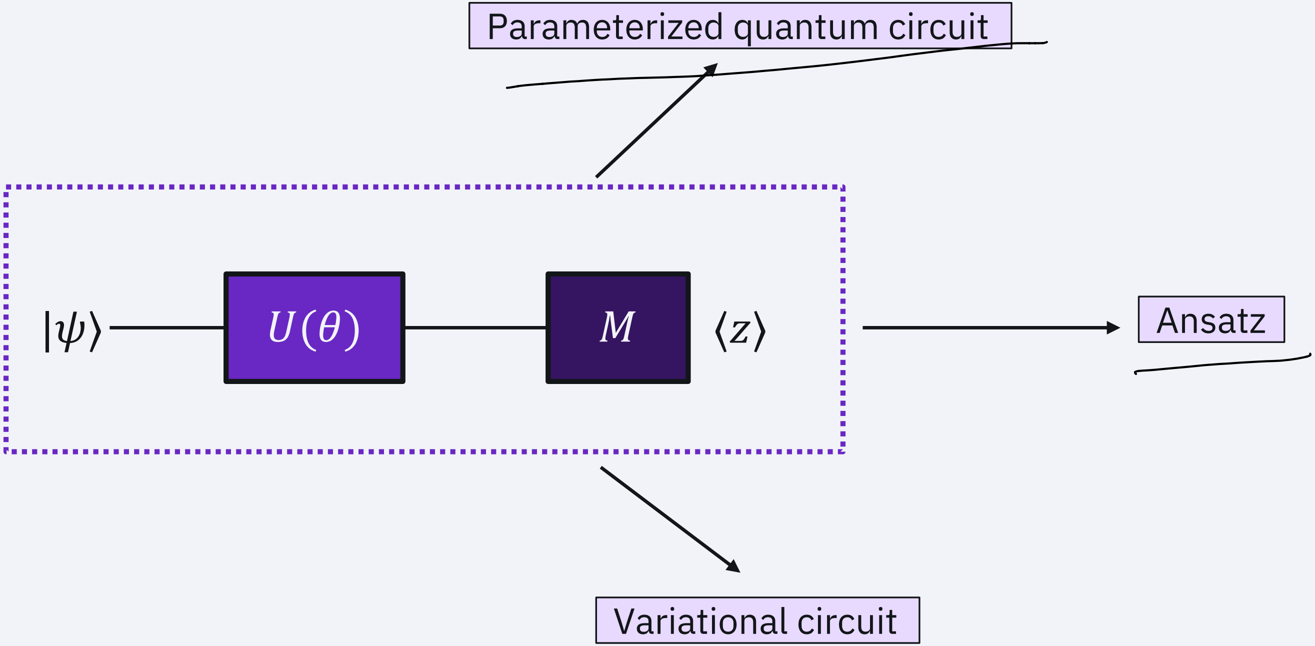
Variational models



Variational models



Variational models



A first attempt

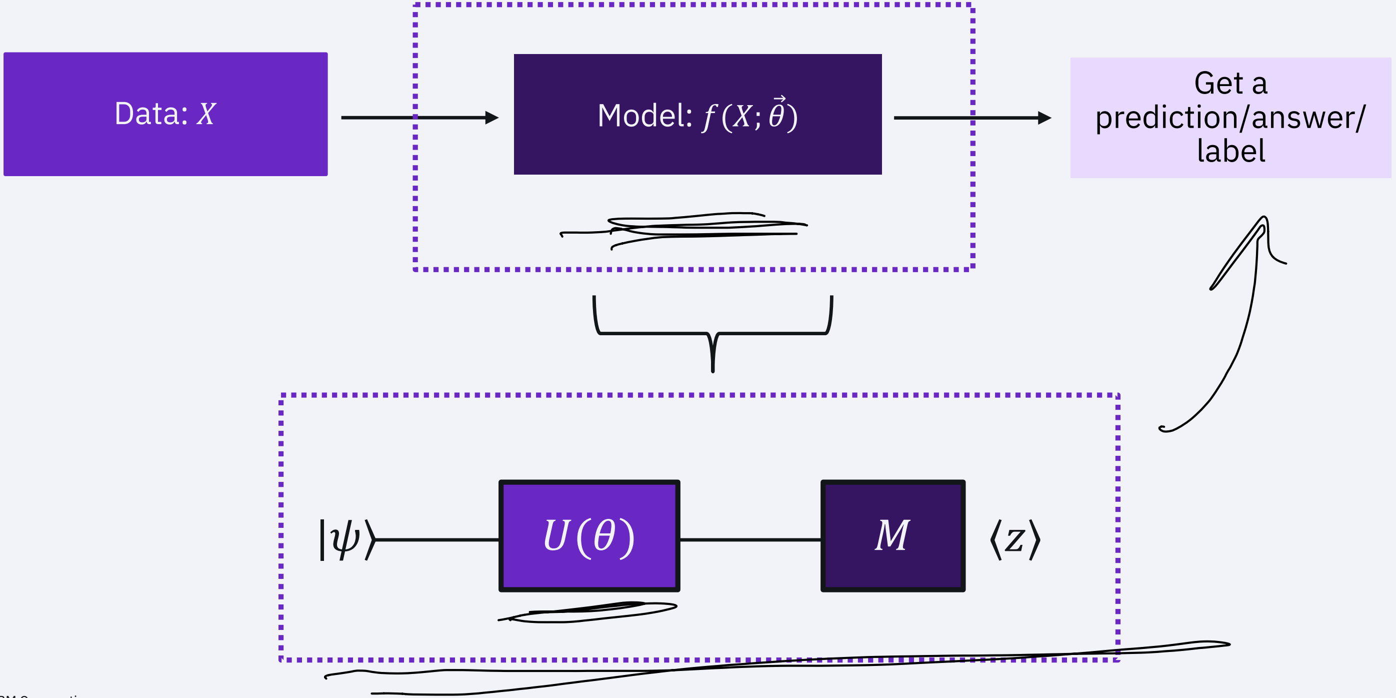
Variational circuit as a classifier

Variational circuit as a classifier

Qiskit

Data: X → Model: $f(X; \vec{\theta})$ → Get a prediction/answer/label

Variational circuit as a classifier



Variational circuit as a classifier

Task: Train a quantum circuit on labelled samples in order to predict labels for new data

Variational circuit as a classifier

Task: Train a quantum circuit on labelled samples in order to predict labels for new data

Step 1: Encode the classical data into a quantum state

Variational circuit as a classifier

Task: Train a quantum circuit on labelled samples in order to predict labels for new data

Step 1: Encode the classical data into a quantum state

Step 2: Apply a parameterized model

Variational circuit as a classifier

Task: Train a quantum circuit on labelled samples in order to predict labels for new data

Step 1: Encode the classical data into a quantum state

Step 2: Apply a parameterized model

Step 3: Measure the circuit to extract labels

Variational circuit as a classifier

Task: Train a quantum circuit on labelled samples in order to predict labels for new data

Step 1: Encode the classical data into a quantum state

Step 2: Apply a parameterized model

Step 3: Measure the circuit to extract labels

Step 4: Use optimization techniques (like gradient descent) to update model parameters

Data encoding

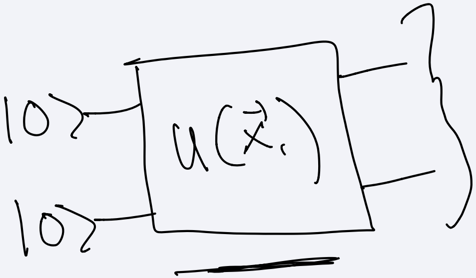
Basis encoding

$$\left.\begin{matrix}\vec{x}_1 \\ \vec{x}_2\end{matrix}\right\} \quad \begin{matrix}00 \\ 10\end{matrix} \longrightarrow \left.\begin{matrix}|00\rangle \\ |10\rangle\end{matrix}\right\} \quad \left.\begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array}\begin{bmatrix}\sqrt{\frac{1}{2}} \\ 0 \\ \sqrt{\frac{1}{2}} \\ 0\end{bmatrix}\right\}$$

Data encoding

Amplitude encoding

$$\vec{x}_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{bmatrix} 0.1 \\ \vdots \\ \vdots \\ 0.2 \end{bmatrix}$$

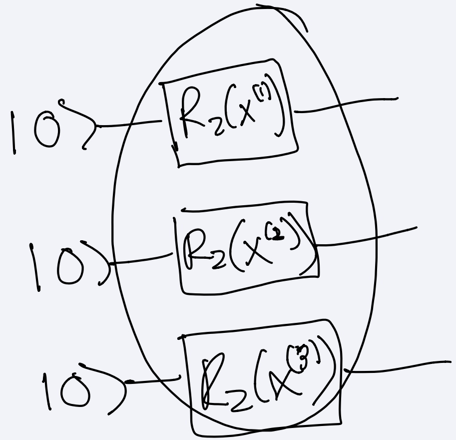


$$|\psi\rangle = \begin{bmatrix} 0.1 \\ \vdots \\ 0.2 \end{bmatrix} = \vec{x}_1$$

Data encoding

Angle encoding

$$\vec{x}_1 = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}$$



Data encoding

Qiskit

Angle encoding: Qiskit circuit library

qiskit.circuit.library.ZFeatureMap ¶

CLASS **ZFeatureMap**(*feature_dimension*, *reps=2*, *data_map_func=None*, *insert_barriers=False*) [SOURCE] ¶

The first order Pauli Z-evolution circuit.

On 3 qubits and with 2 repetitions the circuit is represented by:

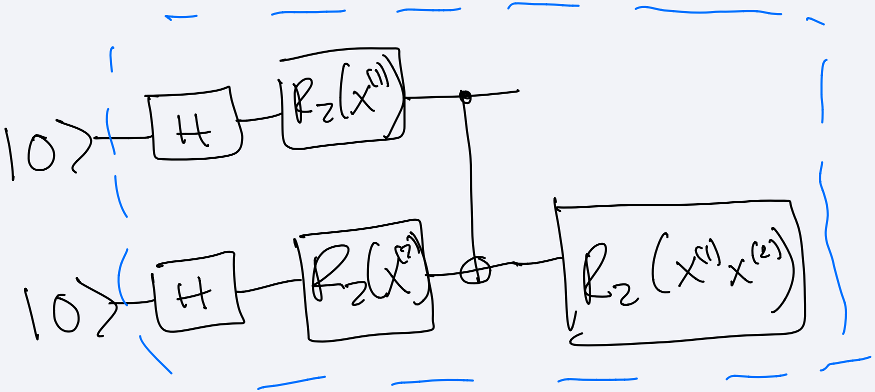
```
┌───┐┌──────────────┐┌───┐┌──────────────┐
┤ H ├┤ U1(2.0*x[0]) ├┤ H ├┤ U1(2.0*x[0]) ├
├───┤├──────────────┤├───┤├──────────────┤
┤ H ├┤ U1(2.0*x[1]) ├┤ H ├┤ U1(2.0*x[1]) ├
├───┤├──────────────┤├───┤├──────────────┤
┤ H ├┤ U1(2.0*x[2]) ├┤ H ├┤ U1(2.0*x[2]) ├
└───┘└──────────────┘└───┘└──────────────┘
```

This is a sub-class of `PauliFeatureMap` where the Pauli strings are fixed as *['Z']*. As a result the first order expansion will be a circuit without entangling gates.

Data encoding

Higher order encoding

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}$$



Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Data encoding

Qiskit

Higher order encoding

qiskit.circuit.library.ZZFeatureMap ¶

CLASS **ZZFeatureMap**(*feature_dimension*, *reps=2*, *entanglement='full'*, *data_map_func=None*, *insert_barriers=False*) [SOURCE] ¶

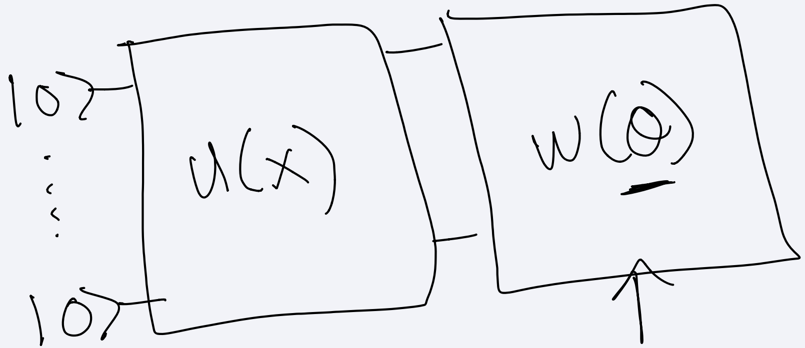
Second-order Pauli-Z evolution circuit.

For 3 qubits and 1 repetition and linear entanglement the circuit is represented by:

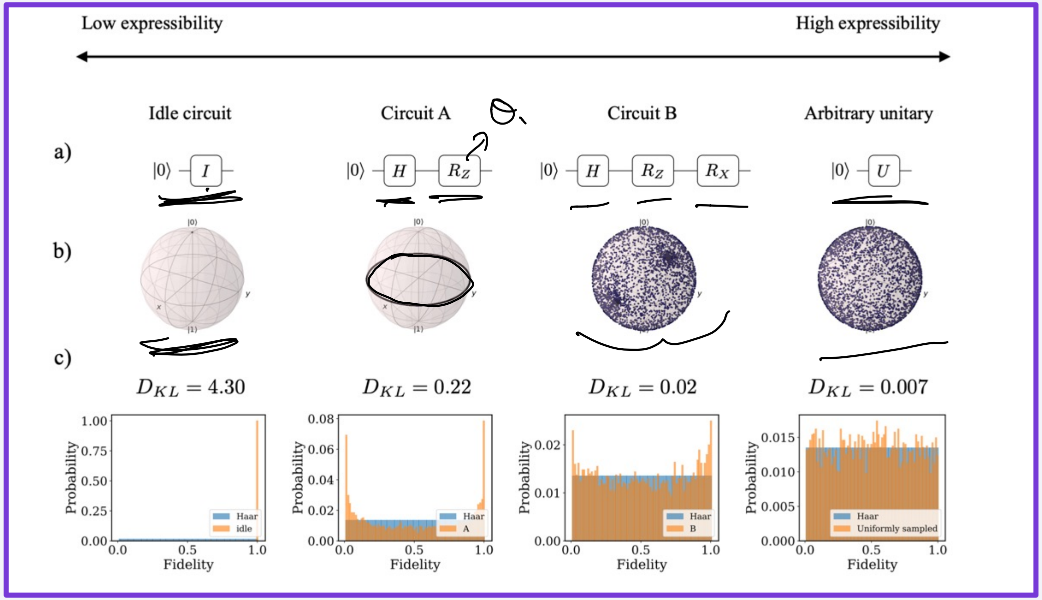
```
┌───┐┌─────────────────┐
┤ H ├┤ U1(2.0*φ(x[0])) ├──■────────────────────────────■────────────────────────────────────────
├───┤├─────────────────┤┌─┴─┐┌──────────────────────┐┌─┴─┐
┤ H ├┤ U1(2.0*φ(x[1])) ├┤ X ├┤ U1(2.0*φ(x[0],x[1])) ├┤ X ├──■────────────────────────────■──
├───┤├─────────────────┤└───┘└──────────────────────┘└───┘┌─┴─┐┌──────────────────────┐┌─┴─┐
┤ H ├┤ U1(2.0*φ(x[2])) ├──────────────────────────────────┤ X ├┤ U1(2.0*φ(x[1],x[2])) ├┤ X ├
└───┘└─────────────────┘                                  └───┘└──────────────────────┘└───┘
```

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Applying a variational model

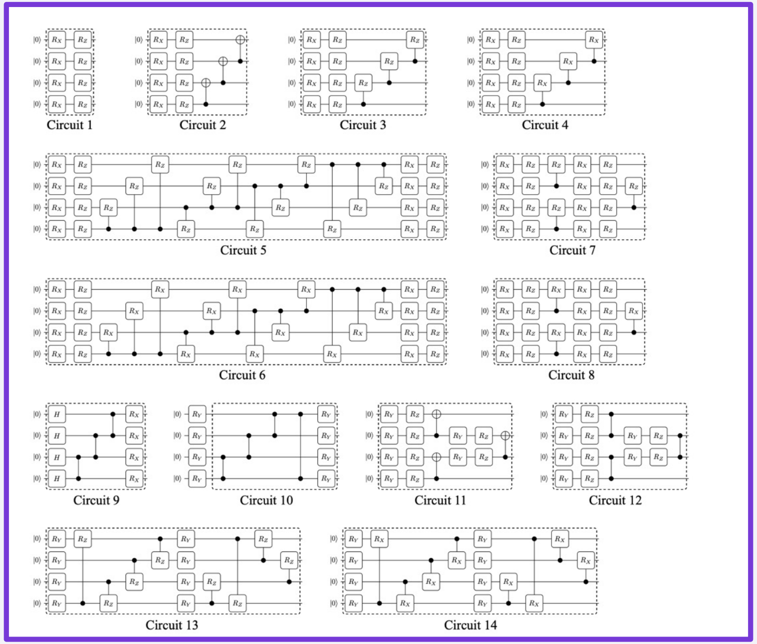


Applying a variational model



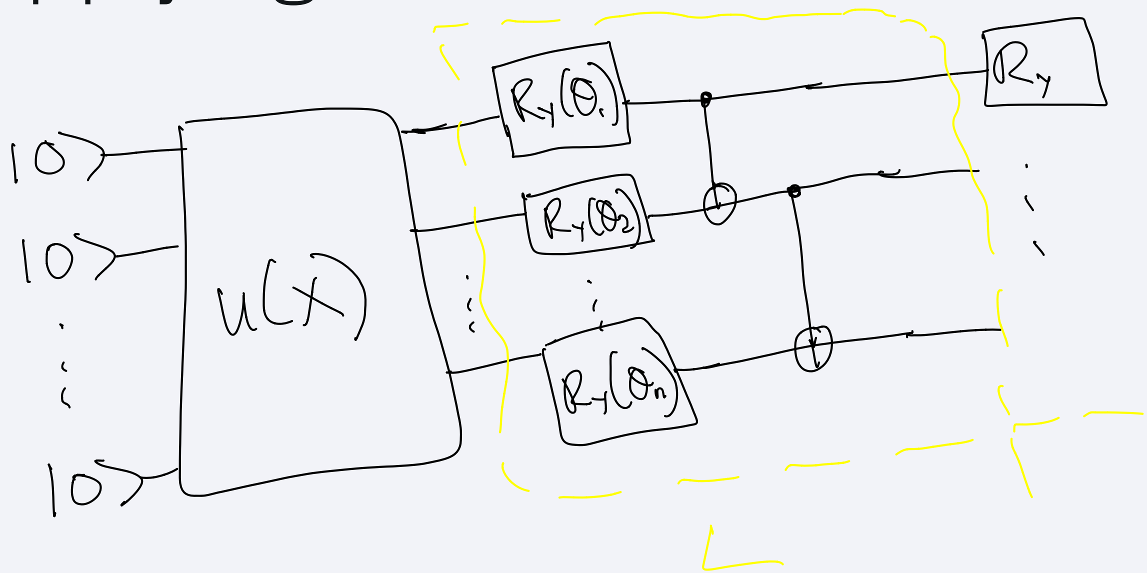
Sim et al. "Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms." Advanced Quantum Technologies 2.12 (2019): 1900070.

Applying a variational model



Sim et al. "Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms." Advanced Quantum Technologies 2.12 (2019): 1900070.

Applying a variational model



Applying a variational model

Qiskit

qiskit.circuit.library.RealAmplitudes ¶

CLASS **RealAmplitudes**(*num_qubits*=*None*, *entanglement*='*full*', *reps*=*3*, *skip_unentangled_qubits*=*False*, *skip_final_rotation_layer*=*False*, *parameter_prefix*='*θ*', *insert_barriers*=*False*, *initial_state*=*None*) [SOURCE] ¶

The real-amplitudes 2-local circuit.

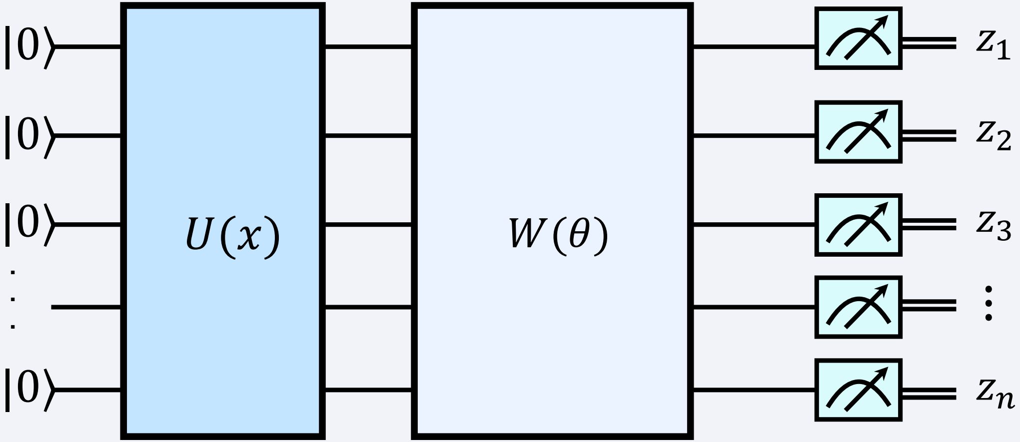
The `RealAmplitudes` circuit is a heuristic trial wave function used as Ansatz in chemistry applications or classification circuits in machine learning. The circuit consists of of alternating layers of Y rotations and CX entanglements. The entanglement pattern can be user-defined or selected from a predefined set. It is called `RealAmplitudes` since the prepared quantum states will only have real amplitudes, the complex part is always 0.

For example a `RealAmplitudes` circuit with 2 repetitions on 3 qubits with `'full'` entanglement is

```
┌──────────┐ ░                 ░ ┌──────────┐ ░                 ░ ┌──────────┐
┤ RY(θ[0]) ├─░───■────■────────░─┤ RY(θ[3]) ├─░───■────■────────░─┤ RY(θ[6]) ├
├──────────┤ ░ ┌─┴─┐  │        ░ ├──────────┤ ░ ┌─┴─┐  │        ░ ├──────────┤
┤ RY(θ[1]) ├─░─┤ X ├──┼────■───░─┤ RY(θ[4]) ├─░─┤ X ├──┼────■───░─┤ RY(θ[7]) ├
├──────────┤ ░ └───┘┌─┴─┐┌─┴─┐ ░ ├──────────┤ ░ └───┘┌─┴─┐┌─┴─┐ ░ ├──────────┤
┤ RY(θ[2]) ├─░──────┤ X ├┤ X ├─░─┤ RY(θ[5]) ├─░──────┤ X ├┤ X ├─░─┤ RY(θ[8]) ├
└──────────┘ ░      └───┘└───┘ ░ └──────────┘ ░      └───┘└───┘ ░ └──────────┘
```

Extracting a label

Extracting a label



$f(z_1 z_2 \ldots z_n) \in C = \{0,1\}$

Extracting a label

Parity post-processing

$f(z_1 z_2 \ldots z_n) \in C = \{0,1\}$

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Parity post-processing

$f(z_1 z_2 \ldots z_n) \in C = \{0,1\}$

In the case of 2 qubits, there are $2^2 = 4$ possible basis states

$00 \quad 01 \quad 10 \quad 11$

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Parity post-processing

$f(z_1 z_2 ... z_n) \in C = \{0,1\}$

In the case of 2 qubits, there are $2^2 = 4$ possible basis states

$00 \quad 01 \quad 10 \quad 11$

There will be an expectation value associated with each possible basis state, 1000 shots

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Parity post-processing

$f(z_1 z_2 \ldots z_n) \in C = \{0,1\}$

In the case of 2 qubits, there are $2^2 = 4$ possible basis states

$00 \quad 01 \quad 10 \quad 11$

There will be an expectation value associated with each possible basis state, 1000 shots

$00 \longrightarrow 100$

$01 \longrightarrow 10$

$10 \longrightarrow 400$

$11 \longrightarrow 490$

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Parity post-processing

$f(z_1 z_2 ... z_n) \in C = \{0,1\}$

In the case of 2 qubits, there are $2^2 = 4$ possible basis states

$00 \quad 01 \quad 10 \quad 11$

There will be an expectation value associated with each possible basis state, 1000 shots

$00 \longrightarrow 100 \longrightarrow 0.1$

$01 \longrightarrow 10 \longrightarrow 0.01$

$10 \longrightarrow 400 \longrightarrow 0.4$

$11 \longrightarrow 490 \longrightarrow 0.49$

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Parity post-processing

$f(z_1 z_2 ... z_n) \in C = \{0,1\}$

In the case of 2 qubits, there are $2^2 = 4$ possible basis states

$00 \quad 01 \quad 10 \quad 11$

There will be an expectation value associated with each possible basis state, 1000 shots

$00 \longrightarrow 100 \longrightarrow 0.1 \quad f(00) \longrightarrow$ even

$01 \longrightarrow 10 \longrightarrow 0.01 \quad f(01) \longrightarrow$ odd

$10 \longrightarrow 400 \longrightarrow 0.4 \quad f(10) \longrightarrow$ odd

$11 \longrightarrow 490 \longrightarrow 0.49 \quad f(11) \longrightarrow$ even

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Parity post-processing

$f(z_1 z_2 \ldots z_n) \in C = \{0,1\}$

In the case of 2 qubits, there are $2^2 = 4$ possible basis states

$00 \quad 01 \quad 10 \quad 11$

There will be an expectation value associated with each possible basis state, 1000 shots

$00 \longrightarrow 100 \longrightarrow 0.1 \quad f(00) \longrightarrow$ even

$01 \longrightarrow 10 \longrightarrow 0.01 \quad f(01) \longrightarrow$ odd

$10 \longrightarrow 400 \longrightarrow 0.4 \quad f(10) \longrightarrow$ odd

$11 \longrightarrow 490 \longrightarrow 0.49 \quad f(11) \longrightarrow$ even

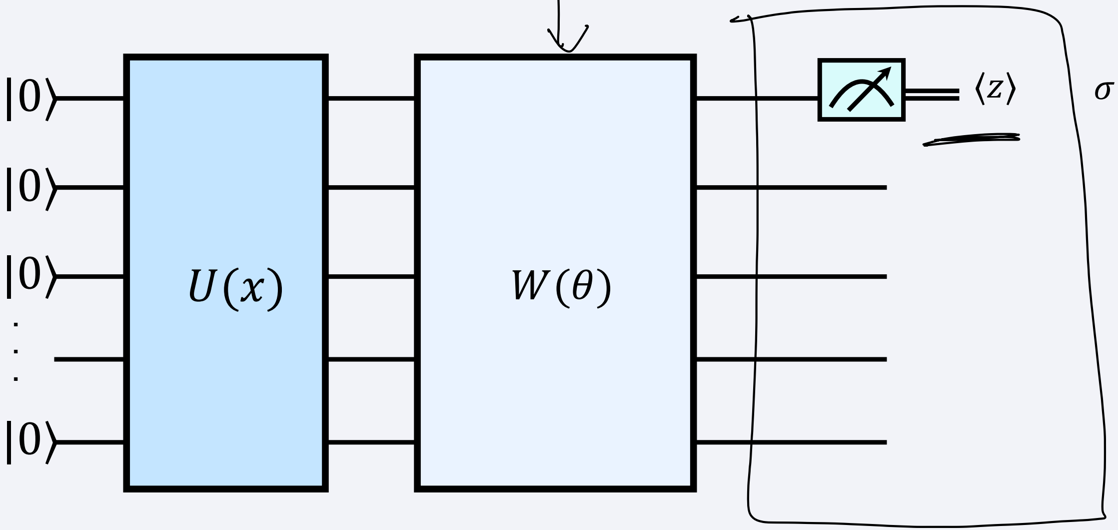
$P(\hat{y} = 0) = 0.59$

$P(\hat{y} = 1) = 0.41$

Source: Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

Extracting a label

Measuring the first qubit



$\sigma(\langle z\rangle) \in C = \{0,1\}$

$M \rightarrow$

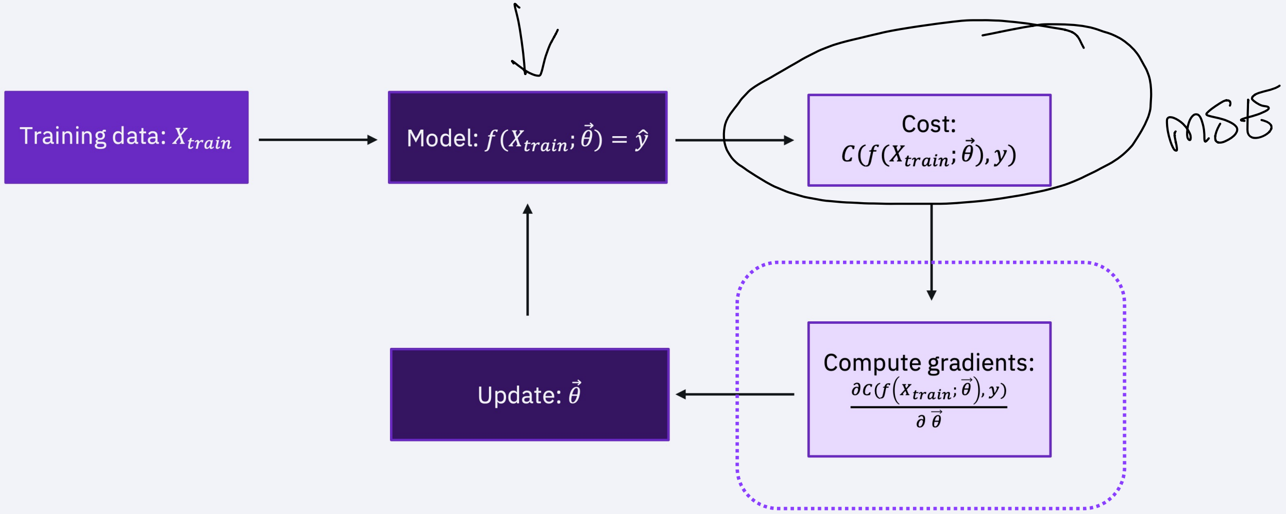
$M^c \rightarrow$ rotations + M

$\langle 0| U(x)^\dagger W(\theta) M \; W(\theta) U(x) |0\rangle$

Optimization

Optimization

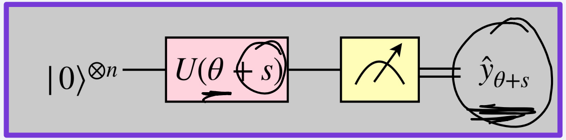
Qiskit



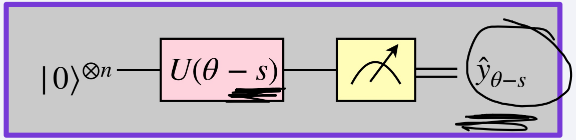
Optimization

Gradient descent?

Gradient =



−



Source: Schuld, Maria, et al. "Circuit-centric quantum classifiers." Physical Review A 101.3 (2020): 032308.

Optimization

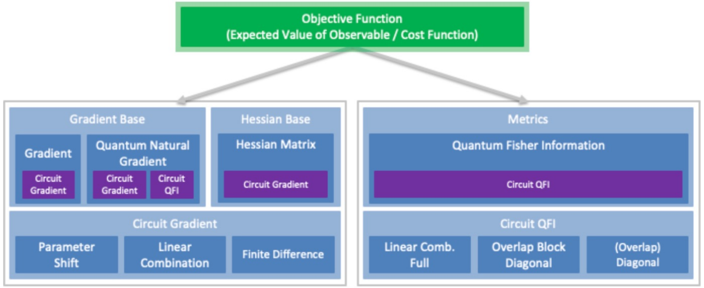
Gradient descent?

Qiskit Gradient Framework

The gradient framework enables the evaluation of quantum gradients as well as functions thereof. Besides standard first order gradients of expectation values of the form

$$\langle\psi(\theta)|\hat{O}(\omega)|\psi(\theta)\rangle$$

The gradient framework also supports the evaluation of second order gradients (Hessians), and the Quantum Fisher Information (QFI) of quantum states $|\psi(\theta)\rangle$.



Jupyter notebook

https://nithecs.ac.za/wp-content/uploads/2020/09/NITheP_mini_school_L3-variational_classifier-Amira-Abbas-Lecture-3-Jupyter-Notebook-22-September-2020.pdf

1 Variational quantum classifier with Qiskit

In this notebook, we build the variational quantum classifier using Qiskit

In [44]:
```
from qiskit.ml.datasets import *
from qiskit import QuantumCircuit
from qiskit.aqua.components.optimizers import COBYLA
from qiskit.circuit.library import ZZFeatureMap, RealAmplitudes
import numpy as np
import matplotlib.pyplot as plt
from qiskit.quantum_info import Statevector
%matplotlib inline
```

Is this advantageous?

This attempt does **not** provide any known advantage

They are linear classifiers in a ***feature space***

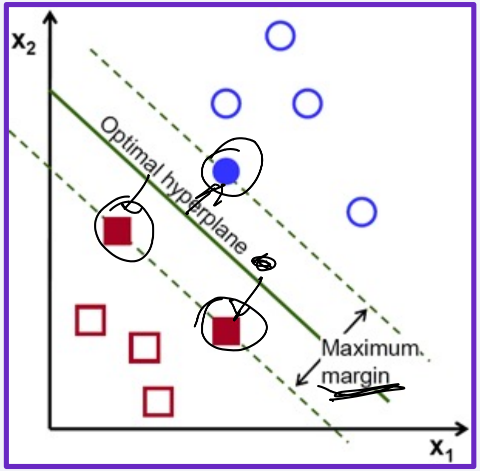
Support vector machines

Support vector machines

Classification problems

Support vector machines

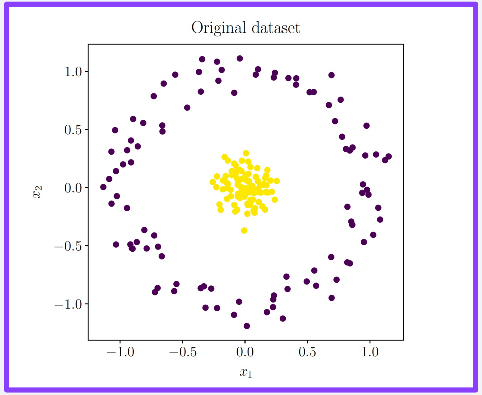
Classification problems



Support vector machines

Qiskit

Classification problems – data not linearly separable?

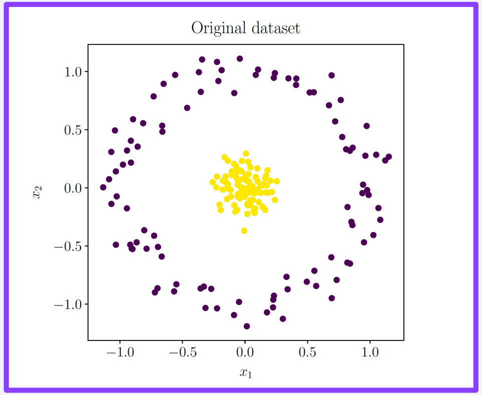


$\in \mathbb{R}^2$

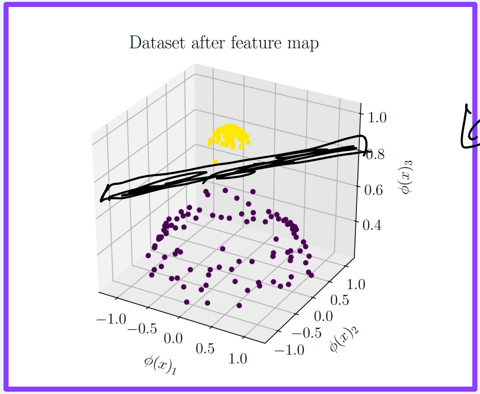
Support vector machines

Qiskit

Classification problems – data not linearly separable? Apply a **feature map**



$$\xrightarrow{\phi}$$

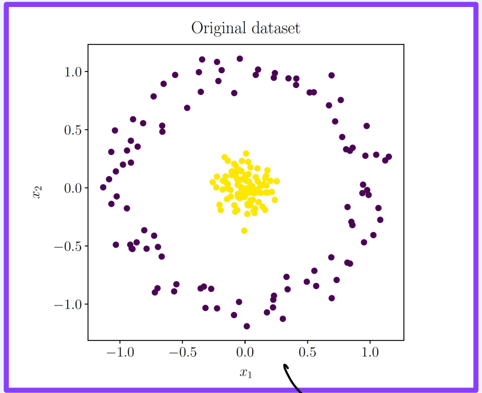


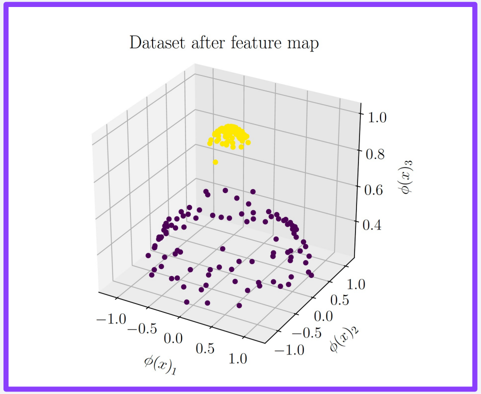
$\mathbb{R}^2$ $\qquad\qquad$ $\mathbb{R}^3$

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} \longrightarrow \phi(\vec{x}) = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(1)} x^{(2)} \end{bmatrix}$$

Support vector machines

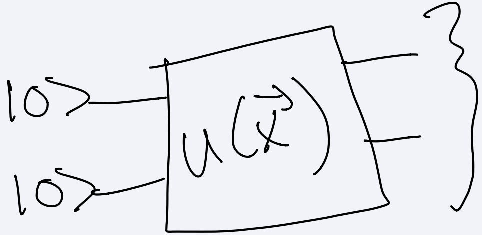
Classification problems – data not linearly separable? Apply a **feature map**





Data encoding

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}$$



$$|\psi\rangle = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}$$

Resources

- Schuld, M. (2021). Quantum machine learning models are kernel methods. *arXiv e-prints*, arXiv-2101.
- Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, *567*(7747), 209-212.
- Glick, J. R., Gujarati, T. P., Corcoles, A. D., Kim, Y., Kandala, A., Gambetta, J. M., & Temme, K. (2021). Covariant quantum kernels for data with group structure. *arXiv preprint arXiv:2105.03406*.
- Kübler, J. M., Buchholz, S., & Schölkopf, B. (2021). The Inductive Bias of Quantum Kernels. *arXiv preprint arXiv:2106.03747*.
- https://pennylane.ai/qml/demos/tutorial_kernel_based_training.html

Recap

Qiskit

Data encoding

Variational models

Optimization

Feature maps

Thank you!

Amira Abbas

@AmiraMorphism