Efficient Quantum Machine Learning w/ Hybrid Computational Resources

Ara G.

agnostiq.ai

AgnostiqHQ

@agnosticHQ

Ara Ghukasyan, PhD
Quantum Algorithms Researcher.

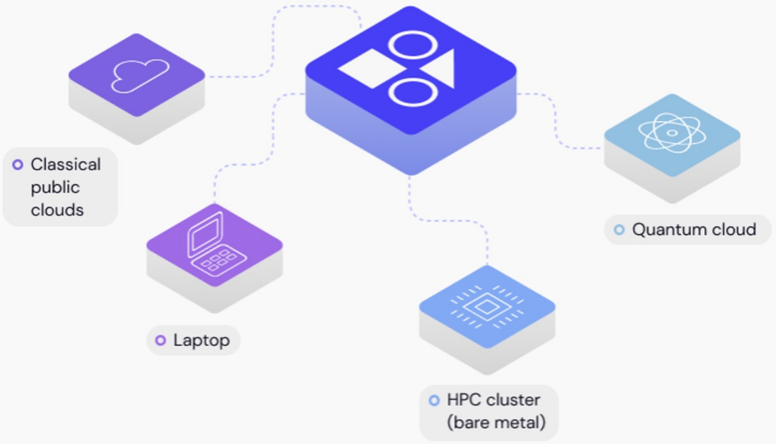
BSc, McMaster, Math & Physics

PhD, McMaster, Engineering Physics

ara@agnostiq.ai

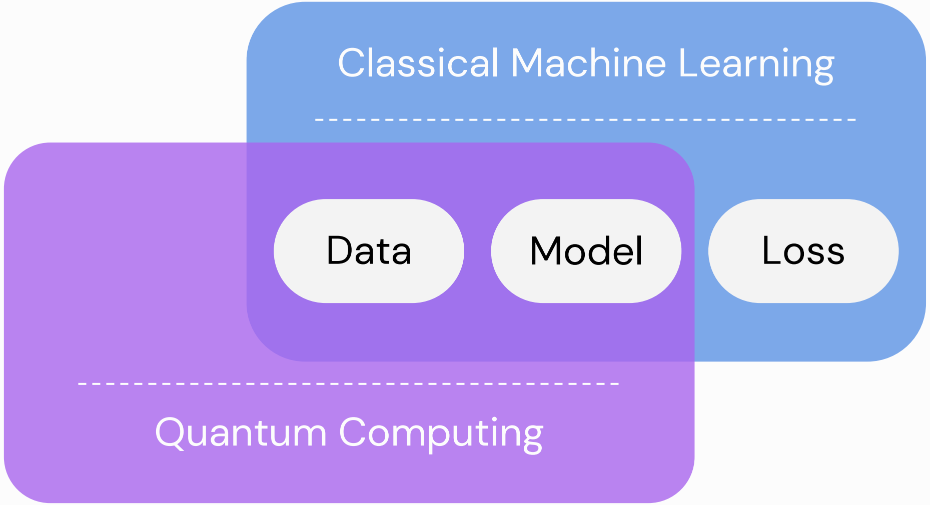
Outline.

- ❑ What is *Quantum Machine Learning*?
- ❑ A kernel algorithm for QML
- ❑ QML in practice



- ❑ **Tutorial:** Running QML workflows with *Covalent*

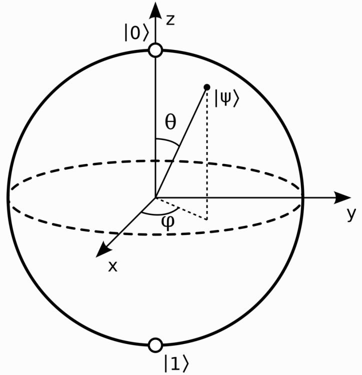
Quantum Machine Learning



- Quantum information as input data
- Quantum computations as ML models

Quantum Computing.

- Processing information using the laws of quantum mechanics.
- Processing information using a quantum system.



3-qubit register
(quantum system)

$$\underbrace{|\phi\rangle|\chi\rangle|\psi\rangle}_{} = \underbrace{\alpha_0|000\rangle + \alpha_1|001\rangle + \cdots + \alpha_6|110\rangle + \alpha_7|111\rangle}_{} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{pmatrix} \in \mathbb{C}^8$$

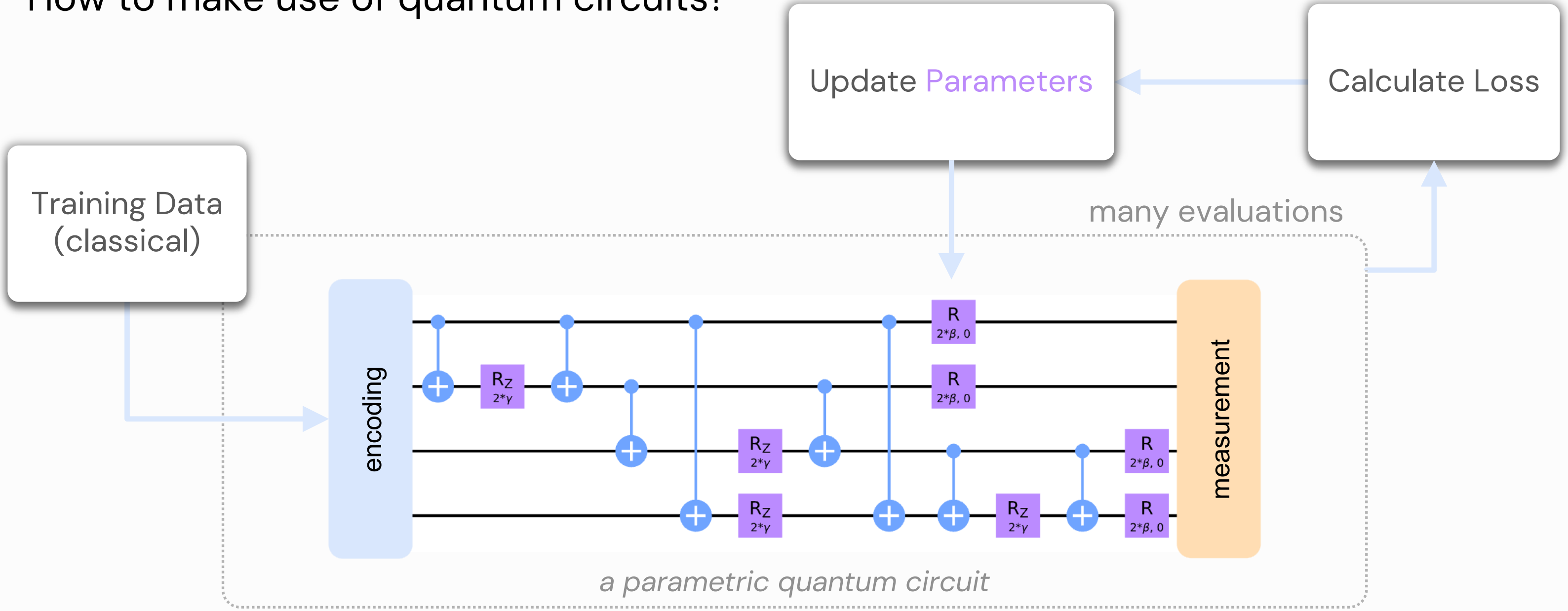
state vector
(superposition over $2^3 = 8$ basis states)

elements of quantum computation

- **operations** (*unitary matrices*) manipulate amplitudes, α_i
- **measurements** return outcomes with probabilities related to $|\alpha_i|^2$
- **algorithms** obtain "desirable" amplitudes, reveal solutions

Quantum Circuits for ML.

- How to make use of quantum circuits?



Support-Vector Machines

SVM Overview.

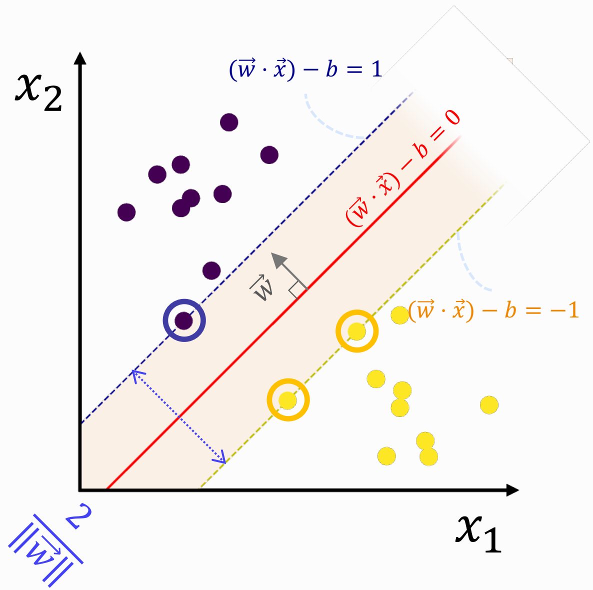
- The naïve SVM is a **linear classifier.**

Objective

find $\vec{w}$ and b, such that

$$\vec{w} \cdot \vec{x} - b = 0$$

determines the line with **largest** possible margin.



SVM Overview.

Feature Map

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{pmatrix}$$

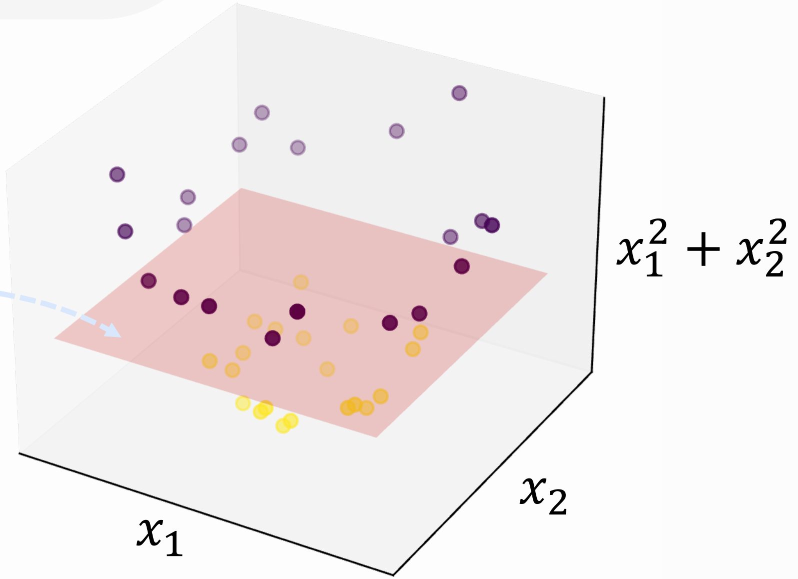
- The non-linear SVM.

Objective

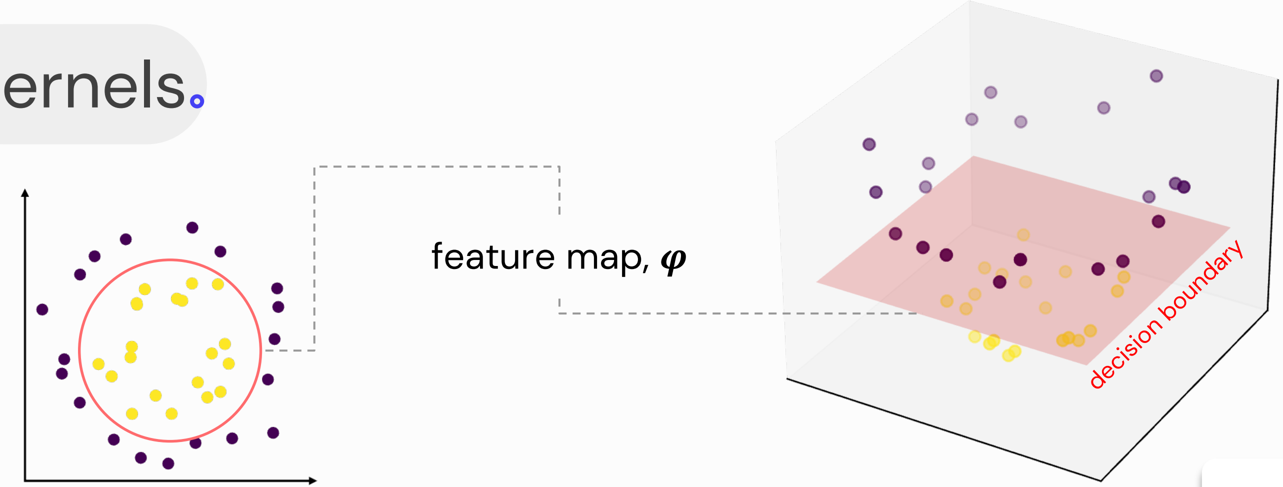
find $\vec{w}$ and b, such that

$$\vec{w} \cdot \vec{x} - b = 0$$

determines the *hyperplane* with **largest** possible margin.



Kernels.



- SVM only cares about inner products on the inputs.
- We can transform the inputs as we like.*

$$\vec{x} \cdot \vec{x}' = (x_1, x_2) \cdot (x_1', x_2')$$

$$\varphi(\vec{x}) \cdot \varphi(\vec{x}') = (x_1, x_2, x_1^2 + x_2^2) \cdot \left(x_1', x_2', {x_1'}^2 + {x_2'}^2\right)$$

feature map

$$\kappa(\vec{x}, \vec{x}') = \varphi(\vec{x}) \cdot \varphi(\vec{x}')$$

kernel function

Quantum Kernels.

- How to evaluate a kernel function on a quantum circuit?

parametric operations

$$|\phi(\vec{x})\rangle = \hat{S}_{\vec{x},\vec{\theta}}\,|0000\rangle$$

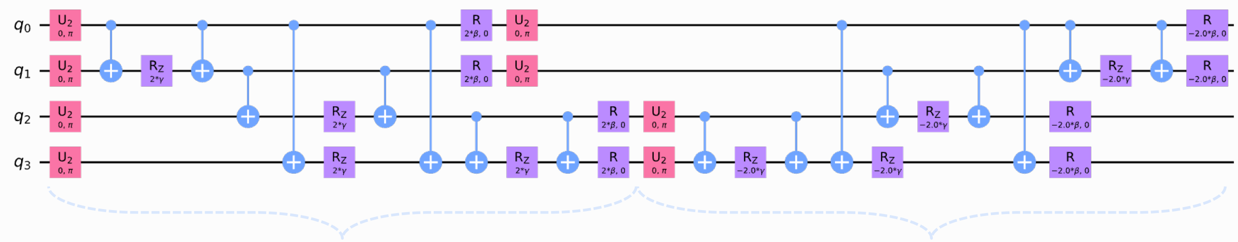
encoding operations

$$\kappa(\vec{x},\vec{x}') = |\langle\phi(\vec{x}')|\phi(\vec{x})\rangle|^2$$

quantum kernel

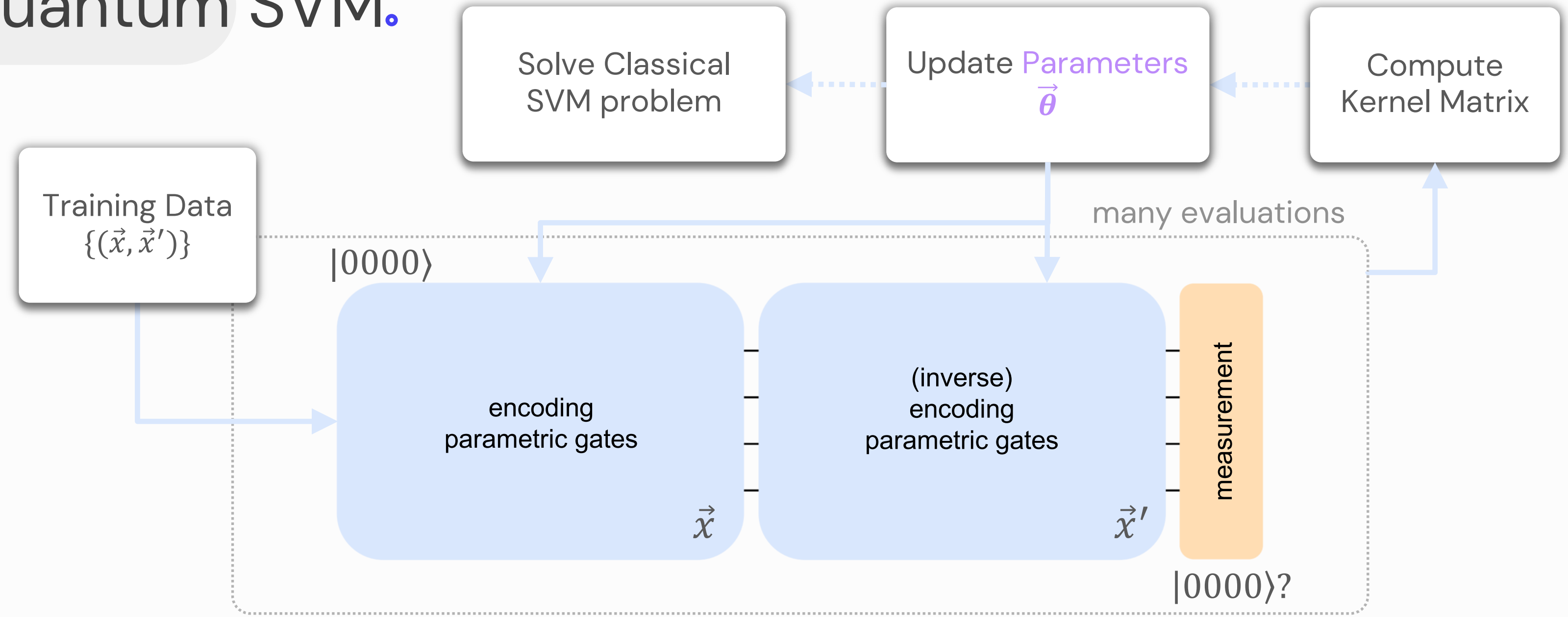
feature map

$$\varphi(\vec{x}) = |\phi(\vec{x})\rangle\langle\phi(\vec{x})|$$



$\hat{S}(\vec{x},\vec{\theta})$ $\hat{S}^\dagger(\vec{x}',\vec{\theta})$

Quantum SVM.



$$\left(\hat{S}_{\vec{x}',\vec{\theta}}\right)^{\dagger} \hat{S}_{\vec{x},\vec{\theta}} \, |0000\rangle = \begin{cases} |0000\rangle \ \text{ if } \ \vec{x}' = \vec{x} & \rightarrow \ \kappa(\vec{x}, \vec{x}') = 1 \\ |\Psi\rangle & \rightarrow \ 0 \leq \kappa(\vec{x}, \vec{x}') < 1 \end{cases}$$

QML in Practice

Implementing a QML algorithm.

quantum programming frameworks, simulators

PENNYLANE

HPC and quantum cloud

aws

classical algorithms

NumPy

scikit learn

Covalent.

`experiment.py / experiment.ipynb`

Tutorial

https://github.com/AgnostiqHQ/tutorials_covalent_qsite_2022

AgnostiqHQ

Resources.

Quantum Computing (IBM)

qiskit.org/textbook/preface.html

Quantum Machine Learning (Xanadu)

pennylane.ai/qml/

Covalent (Agnostiq)

covalent.xyz

covalent.readthedocs.io