QML (QUANTUM MACHINE LEARNING)

WHAT IS QUANTUM MACHINE LEARNING?

Quantum machine learning is a research area that explores the interplay of ideas from quantum computing and machine learning.

For example, we might want to find out whether quantum computers can speed up the time it takes to train or evaluate a machine learning model. On the other hand, we can leverage techniques from machine learning to help us uncover quantum error-correcting codes, estimate the properties of quantum systems, or develop new quantum algorithms.

QUANTUM COMPUTERS AS AI ACCELERATORS

The limits of what machines can learn have always been defined by the computer hardware we run our algorithms on for example, the success of modern-day deep learning with neural networks is enabled by parallel GPU clusters.
Quantum machine learning extends the pool of hardware for machine learning by an entirely new type of computing device the quantum computer. Information processing with quantum computers relies on substantially different laws of physics known as *quantum theory*.

MACHINE LEARNING ON NEAR-TERM QUANTUM DEVICES

Some research focuses on ideal, universal quantum computers ("fault-tolerant QPUs") which are still years away. But there is rapidly-growing interest in quantum machine learning on near-term quantum devices.

We can understand these devices as special-purpose hardware like Application-Specific Integrated Circuits (ASICs) and Field-Programmable Gate Arrays (FPGAs), which are more limited in their functionality.

USING QUANTUM COMPUTERS LIKE NEURAL NETWORKS

In the modern viewpoint, quantum computers can be used and trained like neural networks. We can systematically adapt the physical control parameters,

such as an electromagnetic field strength or a laser pulse frequency, to solve a problem.

For example, a trained circuit can be used to classify the content of images, by encoding the image into the physical state of the device and taking measurements.

THE BIGGER PICTURE: DIFFERENTIABLE PROGRAMMING

But the story is bigger than just using quantum computers to tackle machine learning problems. Quantum circuits are differentiable, and a quantum computer itself can compute the change in control parameters needed to become better at a given task.

Differentiable programming is the very basis of deep learning, implemented in software libraries such as TensorFlow and PyTorch. Differentiable programming is more than deep learning: it is a programming paradigm where the algorithms are not hand-coded, but learned.

Similarly, the idea of training quantum computers is larger than quantum machine learning. Trainable quantum circuits can be leveraged in other fields like quantum chemistry or quantum optimization. It can help in a variety of applications such as the design of quantum algorithms, the discovery of quantum error correction schemes, and the understanding of physical systems.

PENNYLANE FOR QUANTUM DIFFERENTIABLE PROGRAMMING

PennyLane is an open-source software framework built around the concept of quantum differentiable programming. It seamlessly integrates classical machine learning libraries with quantum simulators and hardware, giving users the power to train quantum circuits.

QML TYPES

- **DelegateModel - Encapsulates a model and delegate**
- **DelegateModelGroup - Encapsulates a filtered set of visual data items**
- **Instantiator - Dynamically creates objects**

- **ItemSelectionModel - Instantiates a QItemSelectionModel to be used in conjunction with a QAbstractItemModel and any view supporting it**
- **ListElement - Defines a data item in a ListModel**
- **ListModel - Defines a free-form list data source**
- **ObjectModel - Defines a set of items to be used as a model**
- **Package - Specifies a collection of named items**

EXPERIMENTAL QML TYPES

- **DelegateChoice - Encapsulates a delegate and when to use it**
- **DelegateChooser - Allows a view to use different delegates for different types of items in the model**
- **TableModel - Encapsulates a simple table model**
- **TableModelColumn - Represents a column in a model**