Why Quantum Machine Learning?

ML & DL applications multiply across all fields of science, society and industry

HEP is quickly catching up

Large data sets, HPC and cloud resources and (new) dedicated hardware drive the increase in complexity and range of applications

Quantum approach to ML could solve more complicated problems… faster

ML based tool can recognize complicated (hidden) patterns in data

Quantum processors can produce statistical patterns that are computationally difficult to produce with classical approaches

→ Could quantum processors recognize more complicated patterns in data?
A quantum speed-up* for ML?

Defining what quantum speed-up means is a complicated task

I/O, data transfer and query complexity

Quantum states preparation, output retrieval and memory access

Computational

How many computing steps are needed to solve a problem

Need to compare to the “best available” classical algorithm

For ML/DL the “best” classical algorithm is often not known

*Biamonte et al. arxiv: 1611.09347
A quantum speed-up* for ML?

Quantum linear algebra is generally faster than classical counterpart

Quantum Basic Linear Algebra subroutines (qBLAS) exhibit exponential speedup

Fourier transforms, eigenvectors and eigenvalues calculation, matrix multiplication and inversion

Some standard ML techniques estimate the ground state of Hamiltonians

Quantum approach may have an advantage

Quantum Boltzmann Machines

ML algorithms have some tolerance to errors

Less affected by quantum instability of results

Specific quantum techniques can be exploited to bring further improvement

Amplitude amplification and quantum annealing

Advantage from special purpose processors, such as quantum annealers

*Biamonte et al. arxiv: 1611.09347
Quantum ML

... and ML for Quantum Computing

QML introduces quantum algorithms as part of a larger implementation
  Fully quantum or hybrid classical/quantum approaches
  Input data could be quantistic → ML for QC

How do we construct Quantum Neural Networks (QNN)?
  Direct association between neurons and qubits
  Encode information into amplitudes of a quantum state

How do we represent learning rules?
  Need association rule between NN activation patterns and pure quantum states

How do we address data loading?
  Quantum state preparation
  Direct access through qRAM?
  Possible to train on large datasets by only loading a small number of samples!
Some Examples

Quantum Nearest Neighbors Clustering [Zhan]
Quantum Principal Component Analysis [Lloyd]
Quantum Support Vector Machines [Rebentrost]
Quantum Boltzmann Machines [Amin]
Quantum Generative Models [Khoshaman]
Quantum implementation of a single Perceptron [Tacchino]
Quantum Boltzmann Machines

Classical Boltzmann Machine consists of visible and hidden binary units $z_a$.

Trained by adjusting weights so that the thermal statistics of the units $P_z$ reproduces the statistics of the data.

QBM replaces units with quantum spins and rewrite the Hamiltonian according to QFT formalism.

Classical Ising Hamiltonian is augmented with a transverse field.

Training process is inspired to Gradient descent approach but it is not trivial.

Trained QBM performed better on simple examples (~10 units) than classical counterpart.

$$z_a \in \{-1, +1\}$$

$$E_z = - \sum_a b_a z_a - \sum_{a,b} w_{ab} z_a z_b.$$
Quantum implementation of a binary perceptron

Represent $m$ classical inputs and weights with $N$ qubit: $m=2^N$

Quantum system is initialized in his idle state

Apply two unitary transformations as a series of gates:
- Prepare the quantum state
- Apply weights

Store output in ancilla qubit
- Apply activation function by measuring ancilla
  - An additional ancilla allows coherent propagation of output to second perceptron

$N=2$ perceptron tested on IBM Q-5
Some HEP related applications

Classification with Quantum Annealing on the D-Wave System (J-R. Vlimant)
https://indico.cern.ch/event/719844/contributions/3047935

Quantum Support Vector Machines (W. Guan) -- next talk --
https://indico.cern.ch/event/719844/contributions/3197680

Quantum Variational AutoEncoder (Vinci, D-Wave)
https://indico.cern.ch/event/719844/contributions/3101600

Applications in Astrophysics (ORNL, FNAL)
https://indico.cern.ch/event/719844/contributions/3105972

Machine Learning for Quantum Computing
Deep reinforcement learning approach for fast qubit control (A. Ustyuzhanin)
https://indico.cern.ch/event/719844/contributions/3167608
Quantum Support Vector Machine

Quantum SVM for $ttH (H \rightarrow \gamma\gamma)$ classification

QSVM is simulated on IBM Qiskit simulator
- different numbers of qubits and events
Entanglement is used to encode relationships between features
Apply PCA to input data features
- Reduced from 45 to 8, 10 or 20 (limited by number of qubits)
Running full training with quantum simulators requires large computing resources
- Memory increases with qubit, training events and complexity

More in W. Guan’s talk
Quantum Annealing for classification

First application of D-Wave quantum annealing to HEP

Transform binary classification into Ising Hamiltonian optimisation problem

Use boosting as learning rule and run adiabatic quantum optimisation

Build Weak/strong classifiers from <40 features for $H \rightarrow \gamma \gamma$ signal/background separation

Define objective as a Quantum Unconstrained Binary Optimisation

Comparison to different classical methods (decision trees and DNN) and exact solution obtained from simulation
Quantum VAE

Quantum Variational Auto-Encoder

Example of a hybrid model

- **classical AutoEncoder**: forward-propagation through DNN
- **quantum generative process**: sampling from quantum Boltzmann distributions

Demonstrate QVAE can be trained with quantum annealers on non-trivial datasets

- Can generate realistic samples!

Additional work to prove quantum advantage

trained with 128 qubit on D-Wave 2000Q
Summary

Diversified research in Quantum Computing and Machine Learning
  R&D on quantum version of building blocks and models for Machine Learning
    Fully quantum or hybrid quantum/classical
  A lot of work on ML for QC optimization and to quantum data

Many research areas
  Simulators require large computing resources to train
  Efficient encoding of NN models and learning strategies
  Training data input (especially for Deep Learning applications)
  Error mitigation

First applications are promising
  Tested on simulators but also on available hardware
  Highlight scope and limitation and can provide important feedback to hardware R&D

Quantum Annealing hardware can already solve realistic problems

We are interested in collaborating with research institutes and industry partners
Some References


[Khoshaman] Khoshaman, Vinci et al., QST, 4, 1
