Machine Learning and Quantum Computing

openlab Technical Workshop

Sofia Vallecorsa, Federico Carminati

CERN 24/01/2019

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?

*Biamonte et al. arxiv: 1611.09347

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

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 Boltzman 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

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]**

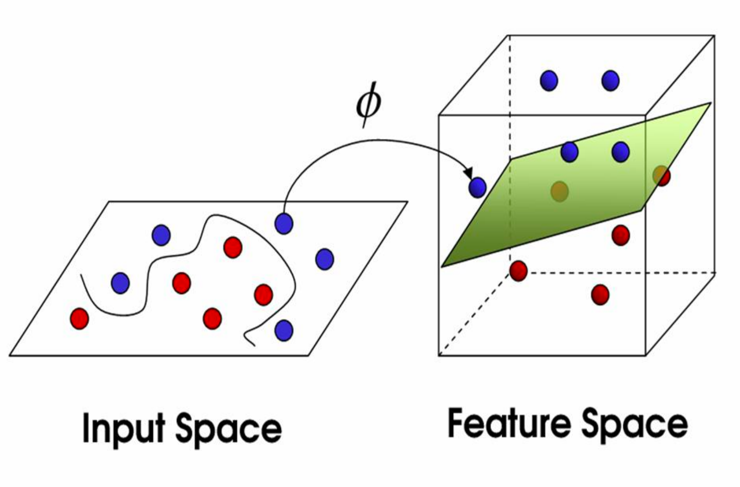
Quantum Support Vector Machines **[Rebentrost]**

Quantum Boltzmann Machines **[Amin]**

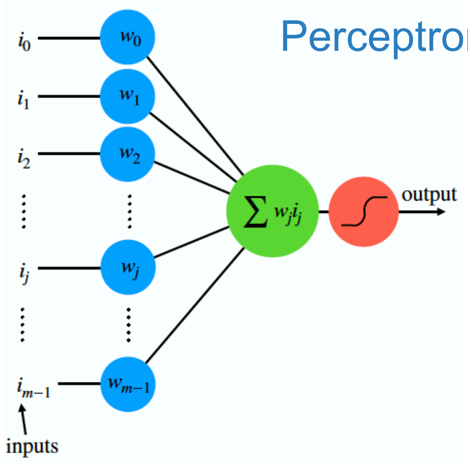
Quantum Generative Models **[Khoshaman]**

Quantum implementation of a single Perceptron **[Tacchino]**

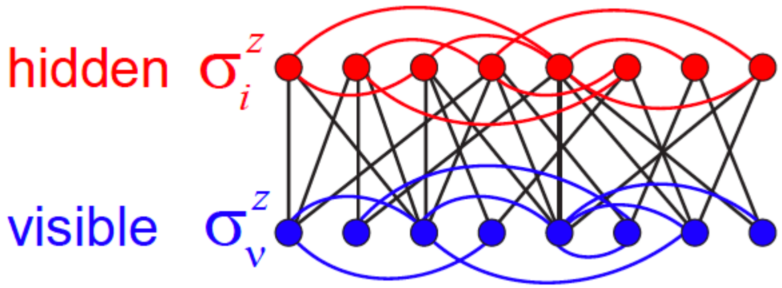
Principle of Support Vector Machines (SVM)



Perceptron



Boltzman Machine



[Amin]

Quantum Boltzmann Machines

- Classical Boltzmann Machine consists of visible and hidden binary units $\mathbf{z_a}$.
  - Trained by adjusting weights so that the thermal statistics of the units $\mathbf{P_z}$ 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

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

$$E_{\mathbf{z}} = -\sum_a b_a z_a - \sum_{a,b} w_{ab} z_a z_b.$$

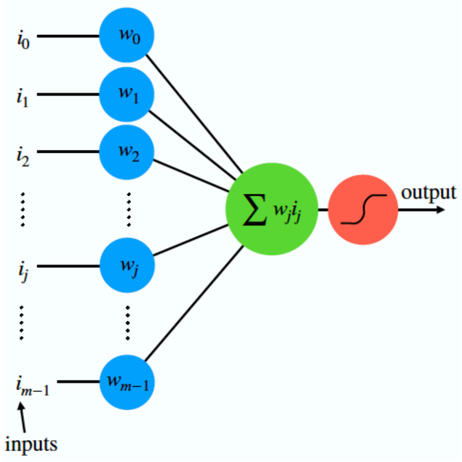
$$P_{\mathbf{v}} = Z^{-1} \sum_{\mathbf{h}} e^{-E_{\mathbf{z}}}, \qquad Z = \sum_{\mathbf{z}} e^{-E_{\mathbf{z}}},$$

$$\boxed{H = -\sum_a \Gamma_a \sigma_a^x - \sum_a b_a \sigma_a^z - \sum_{a,b} w_{ab} \sigma_a^z \sigma_b^z}$$

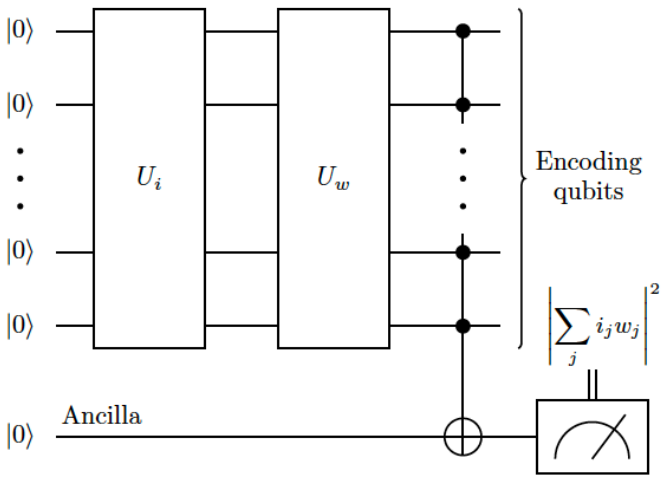
[Tacchino]

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



$$\vec{i} = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{m-1} \end{pmatrix} \qquad \vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix}$$



$$|\psi_i\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j |j\rangle$$

$$|\psi_w\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j |j\rangle$$

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

More in W. Guan's talk

QSVM is simulated on IBM Qiskit simulator

- different numbers of qubits and events

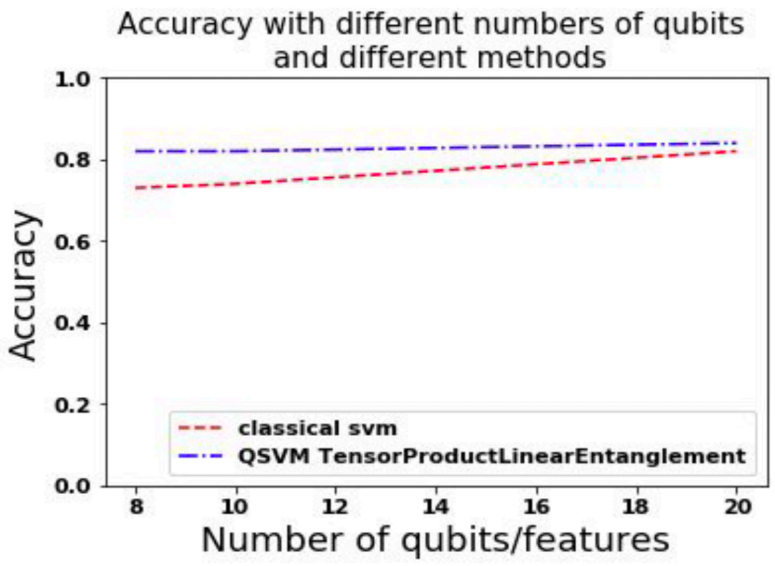
Entanglement is used to encode relationships between features

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



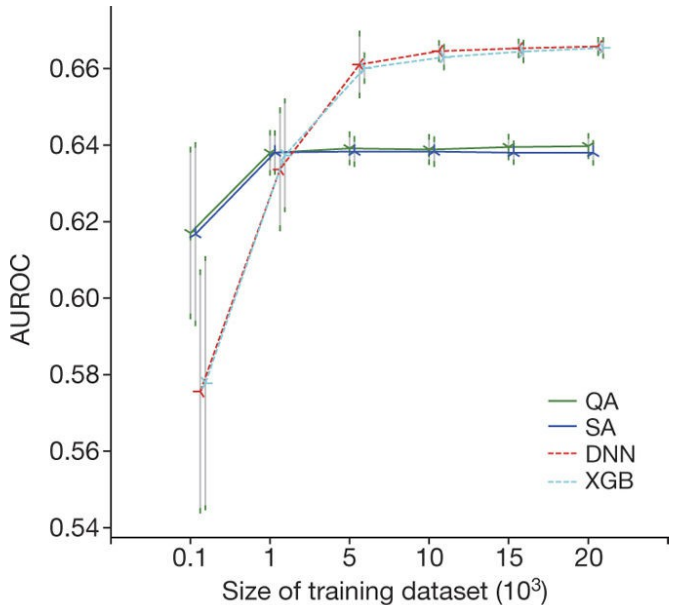
$$Accuracy = \frac{\text{Number of correct prediction}}{\text{Total number of predictions}}$$

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->γγ 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

D-Wave
The Quantum Computing Company™



nature
International journal of science

Letter | Published: 18 October 2017

Solving a Higgs optimization problem with quantum annealing for machine learning

Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar & Maria Spiropulu

Nature **550**, 375–379 (19 October 2017) | Download Citation

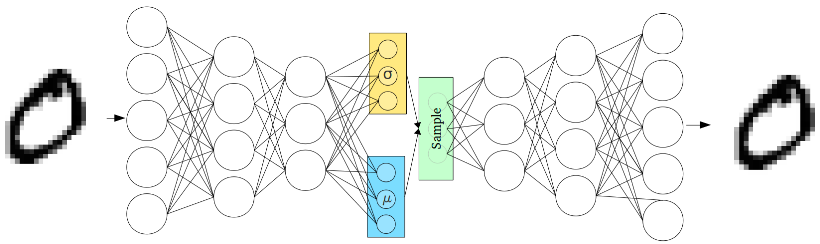
Quantum VAE

[Khoshaman]

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

[Lloyd] Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component

analysis. Nat. Phys. 10, 631{633 (2014). DOI 10.1038/nphys3029

[Zhan] Zhan, Justin Zhijun, Chang, LiWu, and Matwin, Stan. Pri-

vacy preserving k-nearest neighbor classification. IJ Network Security, 1(1):46–51, 2005

[Rebentrost] Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classication. Phys. Rev. Lett. 113 , 130503 (2014). DOI 10.1103/PhysRevLett.113.130503.

[Amin] Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. Quantum Boltzmann machine. arXiv:1601.02036 (2016).

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

[Giovannetti] Giovannetti, V., Lloyd, S. & Maccone, L. Quantum random access

memory. Phys. Rev. Lett. 100 , 160501 (2008).

[Tacchino] F. Tacchino, C. Macchiavello, D. Gerace and D. Bajoni, arxiv:1811.02266 (2018).