Quantum Machine Learning: Strategies based on Quantum Annealing and Gated Quantum Computing

Yaroslav Koshka

Electrical and Computer Eng., Mississippi State University

Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs "apples to bees"..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, $\otimes$
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms?
- Apples to oranges… vs "apples to bees"..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, ⊗
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

Can QC help my branch of ML? Now or how soon?

- HHL
- Linear Regression
- WBL
- Q clustering
- Q PCA
- Q SPV
- Q perceptron
- Q NN
- Q CNN
- Q DL

…

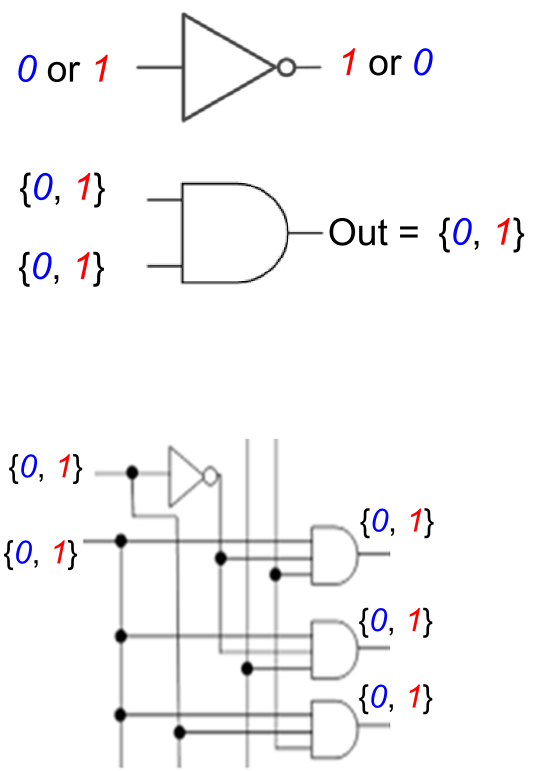
Difficulties… (N of qBits, de-coherence, noise…)

Outline

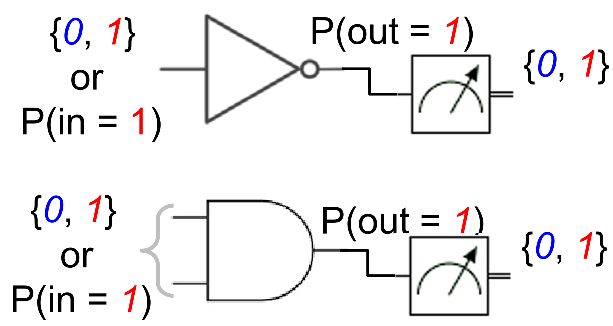
- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs "apples to bees"..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, ⊗
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

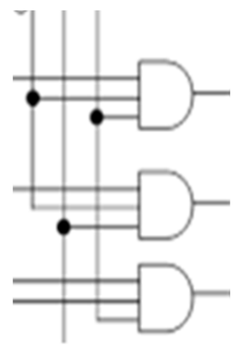
Classical, classical Probabilistic and Quantum computers

Classical gates



Probabilistic gages





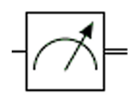
$\mathrm{P}(\mathrm{o}_1 = 1)$

$\mathrm{P}(\mathrm{o}_2 = 1)$

$\mathrm{P}(\mathrm{o}_3 = 1)$

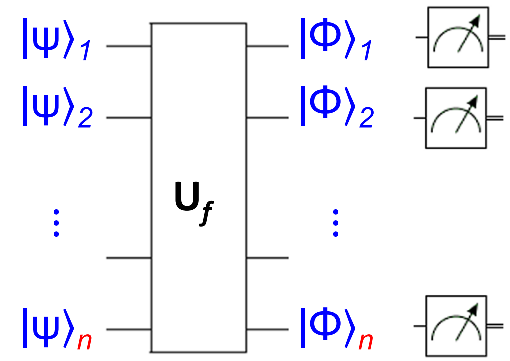
$\vdots$

$\mathrm{P}(\mathrm{o}_n = 1)$



$$out_i \equiv \begin{pmatrix} \mathrm{P}(\mathrm{o}_i = 1) \\ \mathrm{P}(\mathrm{o}_i = 0) \end{pmatrix}_i$$

Quantum gages



$|\ldots\rangle_i \neq \{0, 1\}$ $\quad |\ldots\rangle_i \neq \mathrm{P}(\text{bit} = 1)$

$$|\ldots\rangle_i = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}_i \neq \begin{pmatrix} \mathrm{P}(\text{bit}_i = 0) \\ \mathrm{P}(\text{bit}_i = 1) \end{pmatrix}_i$$

$$|\alpha_1|^2 = \mathrm{P}(\text{bit}_i = 0)$$

$$|\alpha_2|^2 = \mathrm{P}(\text{bit}_i = 1)$$

α is a complex number, called a probability amplitude.

bit $_i$ is not "1 or 0"

bit $_i$ is "simultaneously" 1 and 0

MISSISSIPPI STATE UNIVERSITY

Y. Koshka, 2021

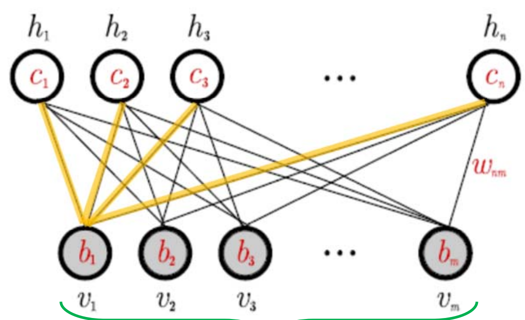
Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs “apples to bees”..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, ⊗
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)

- We do not usually focus on individual "bits".
- We care about a particular simultaneous state of many neurons.

Probabilistic ML



$$P\big(V_j = 1 \mid \mathbf{h}\big) = \sigma\left(\sum_{i=1}^{n} \omega_{ij} h_i + b_j\right) \qquad P(\mathbf{v} \mid \mathbf{h}) = \prod_{j=1}^{m} P\big(V_j = 1 \mid \mathbf{h}\big)$$

$$\underbrace{(\ 0000000011000011\ \dots\ \dots\ \dots}_{\text{Pixels of the image}}\ \underbrace{00010000}_{\text{Labels}}\ )_m$$

$$2^m \left\{ \begin{pmatrix} 00000000\dots\ 00000000 \\ 00000000\dots\ 00000001 \\ 00000000\dots\ 00000010 \\ \dots\dots\dots\dots\dots\dots \\ 00000000\dots\ 00000101 \\ \dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots \\ 11111111\dots\ 11111111 \end{pmatrix} \right. \equiv \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{2^m} \end{pmatrix} \leftarrow \begin{pmatrix} P(\mathbf{v}_1 \mid \mathbf{h}) \\ P(\mathbf{v}_2 \mid \mathbf{h}) \\ \vdots \\ P(\mathbf{v}_{2^m} \mid \mathbf{h}) \end{pmatrix}$$

Quantum ML

$$|\psi\rangle_m = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^m} \end{pmatrix} \neq \begin{pmatrix} P(\mathbf{v} = \mathbf{v}_1 \\ P(\mathbf{v} = \mathbf{v}_2 \\ \vdots \\ P(\mathbf{v} = \mathbf{v}_{2^m} \end{pmatrix}$$

$$|\alpha_1|^2 = \mathrm{P}(\mathbf{v} = \mathbf{v}_1)$$
$$|\alpha_2|^2 = \mathrm{P}(\mathbf{v} = \mathbf{v}_2)$$
$$\dots$$
$$|\alpha_{2^m}|^2 = \mathrm{P}(\mathbf{v} = \mathbf{v}_{2^m})$$

$$|\psi_{\text{out}}\rangle_m = \mathrm{U}\ |\psi_{\text{in}}\rangle_m$$

Our ML network handles a superposition of states

α is a complex number, called a probability amplitude.

MISSISSIPPI STATE UNIVERSITY

Y. Koshka, 2021

Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs “apples to bees”..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, ⊗
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

Fundamentals: dead-or-alive vs simultaneously dead-and-alive

$$|\psi\rangle_m = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^m} \end{pmatrix}$$

$$|\psi\rangle_1 = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} \langle 0|\psi\rangle \\ \langle 1|\psi\rangle \end{pmatrix}$$

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \text{“}0\text{”} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \text{“}1\text{”}$$

α_0 – the component of $|\psi\rangle$ along $|0\rangle$ direction $\qquad (1\ 0)\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \boxed{\langle 0|\psi\rangle}$

α_1 – the component of $|\psi\rangle$ along $|1\rangle$ direction $\qquad (0\ 1)\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \boxed{\langle 1|\psi\rangle}$

$$|\psi\rangle_2 = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \begin{pmatrix} \langle 00|\psi\rangle \\ \langle 01|\psi\rangle \\ \langle 10|\psi\rangle \\ \langle 11|\psi\rangle \end{pmatrix}$$

$$P(00) = |\alpha_{00}|^2 = \boxed{|\langle 00|\psi\rangle|^2}$$

$$P(01) = |\alpha_{01}|^2 = \boxed{|\langle 01|\psi\rangle|^2}$$

…

MISSISSIPPI STATE UNIVERSITY

Y. Koshka, 2021

Fundamentals: dead-or-alive vs simultaneously dead-and-alive

$|\psi\rangle_m = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^m} \end{pmatrix}$ $\quad |\psi\rangle_1 = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$ $\quad |\psi\rangle_2 = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \begin{pmatrix} \langle 00|\psi\rangle \\ \langle 01|\psi\rangle \\ \langle 10|\psi\rangle \\ \langle 11|\psi\rangle \end{pmatrix}$

$$|Cat\rangle_2 = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \begin{pmatrix} \langle 00|Cat\rangle \\ \langle 01|Cat\rangle \\ \langle 10|Cat\rangle \\ \langle 11|Cat\rangle \end{pmatrix} = \begin{pmatrix} \langle Alive, Black|Cat\rangle \\ \langle Alive, White|Cat\rangle \\ \langle Dead, Black|Cat\rangle \\ \langle Dead, White|Cat\rangle \end{pmatrix}$$

For example

$|Cat\rangle_1 = \begin{pmatrix} \langle Black|Cat\rangle \\ \langle White|Cat\rangle \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\quad P(Black|Cat) = |\langle \ldots | \ldots \rangle|^2 = \frac{1}{2}$ $\quad P(White|Cat) = \frac{1}{2}$

$|Black\rangle = \begin{pmatrix} \langle Alive|Black\rangle \\ \langle Dead|Black\rangle \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\quad |White\rangle = \begin{pmatrix} \langle Alive|White\rangle \\ \langle Dead|White\rangle \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Just an example

$P(Alive|Black) = |\langle \ldots | \ldots \rangle|^2 = \frac{1}{2}$ $\quad P(Dead|Black) = \frac{1}{2}$ $\quad P(Alive|White) = \frac{1}{2}$ $\quad P(Dead|White) = \frac{1}{2}$

Calculate: $P(Alive|Cat) = ?$

Using <u>classical</u> probabilities:

$$P(Alive|Cat) = P(Alive|Black) * P(Black|Cat) + P(Alive|White) * P(White|Cat) = \frac{1}{2} * \frac{1}{2} + \frac{1}{2} * \frac{1}{2} = \mathbf{\frac{1}{2}}$$

Using <u>probability amplitudes</u>:

$P(Alive|Cat) = |\langle Alive|Cat\rangle|^2 = \; P(\ldots) + P(\ldots)$ + Interference Term = **1**

MISSISSIPPI STATE UNIVERSITY

Y. Koshka, 2021

Fundamentals - Continue…

Kronecker product

$$|\,\psi\,\rangle_2 = \begin{pmatrix}\alpha_{00}\\ \alpha_{01}\\ \alpha_{10}\\ \alpha_{11}\end{pmatrix} = \begin{pmatrix}\langle 00|\psi\rangle\\ \boxed{\langle 01|}\psi\rangle\\ \langle 10|\psi\rangle\\ \langle 11|\psi\rangle\end{pmatrix}$$

$$\langle 01| \equiv \langle 0| \otimes \langle 1|$$

$$|01\rangle \equiv |0\rangle \otimes |1\rangle$$

$$|\psi_{qBit1}\rangle = \begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix} \qquad |\psi_{qBit2}\rangle = \begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix}_{qBit2} \qquad |\psi_{combined}\rangle = |\psi_{qBit1}\rangle \otimes |\psi_{qBit2}\rangle$$

If there is no entanglement between the 2 qBits.

Entanglement

$$\begin{pmatrix}\boxed{\alpha_{00}}\\ \boxed{\alpha_{01}}\\ \alpha_{10}\\ \alpha_{11}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\\0\\0\end{pmatrix}$$

qBit2 is *0*, *qBit1* can be *0* or *1*

$$\begin{pmatrix}\boxed{\alpha_{00}}\\ \alpha_{01}\\ \boxed{\alpha_{10}}\\ \alpha_{11}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$$

qBit1 is *0*, *qBit2* can be *0* or *1*

$$\begin{pmatrix}\boxed{\alpha_{00}}\\ \boxed{\alpha_{01}}\\ \boxed{\alpha_{10}}\\ \boxed{\alpha_{11}}\end{pmatrix} = \frac{1}{4}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Both *qBits* can be *0* or *1* independently

$$\begin{pmatrix}\boxed{\alpha_{00}}\\ \alpha_{01}\\ \alpha_{10}\\ \boxed{\alpha_{11}}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

When *qBit1* is *0*, *qBit2* must also be *0*.

When *qBit1* is *1*, *qBit2* must also be *1*.

$$\begin{pmatrix}\alpha_{00}\\ \boxed{\alpha_{01}}\\ \boxed{\alpha_{10}}\\ \alpha_{11}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}0\\1\\1\\0\end{pmatrix}$$

When *qBit1* is *1*, *qBit2* must be *0*.

When *qBit1* is *0*, *qBit2* must be *1*.

Fundamentals - Continue…

Density matrix

$$\rho = |\psi\rangle\langle\psi| = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{2^n} \end{pmatrix} (\alpha_1 \cdots \alpha_{2^n}) = \begin{pmatrix} 2^n \ x \ 2^n \end{pmatrix} = \begin{pmatrix} P(\mathbf{1}) & \cdots & \alpha_1\alpha_{2^n}^* \\ \vdots & \ddots & \vdots \\ \alpha_{2^n}\alpha_1^* & \cdots & P(\mathbf{2^n}) \end{pmatrix}$$

$$\alpha_i\alpha_i^* = |\alpha_i|^2 = P(\boldsymbol{\upsilon_i})$$

What if there is no quantum superposition state $|\psi\rangle$?

00000000… 00000000
00000000… 00000001
00000000… 00000010
……………………………
00000000… 00000101
……………………………
……………………………
11111111… 11111111

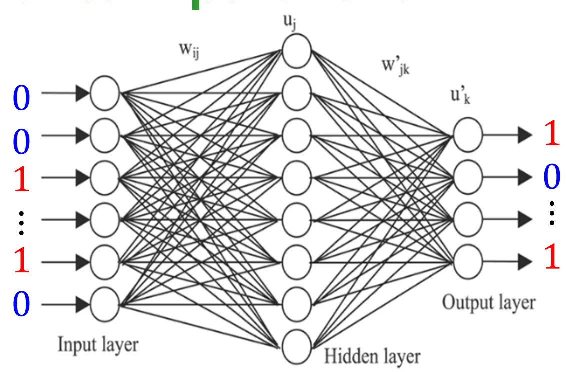
$$\equiv \begin{pmatrix} \boldsymbol{\upsilon}_1 \\ \boldsymbol{\upsilon}_2 \\ \vdots \\ \boldsymbol{\upsilon}_{2^n} \end{pmatrix} \leftarrow \begin{pmatrix} P(\boldsymbol{\upsilon_1}) \\ P(\boldsymbol{\upsilon_1}) \\ \vdots \\ P(\boldsymbol{\upsilon_{2^n}}) \end{pmatrix}$$

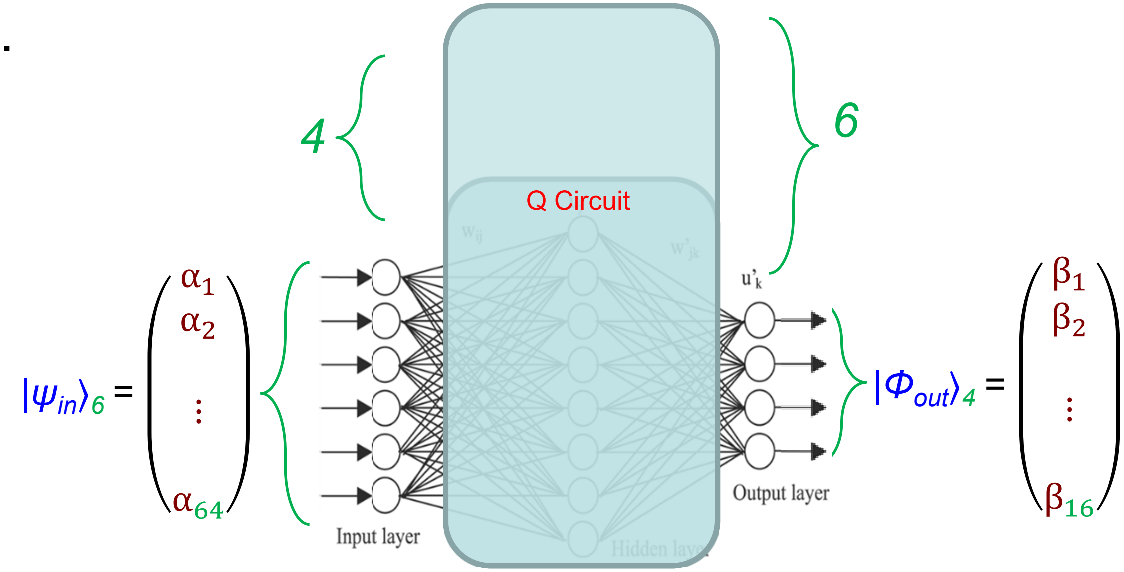
$$\rho = \begin{pmatrix} P(1) & 0 & 0 & 0 \\ 0 & P(2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & P(2^n) \end{pmatrix}$$

- For superposition states and for classical probabilistic models, the same operations on density matrix can be used to calculate probabilities, expectation values, etc.
- Also, density matrices make it possible to access sub-sets of qBits (partial trace, etc.)

Fundamentals - Continue…

Quantum parallelism





$$\begin{pmatrix} 000000 \\ 000001 \\ 000010 \\ \cdots \\ \cdots \\ 100000 \\ 100001 \\ \cdots \\ \cdots \\ 111111 \end{pmatrix} 2^6 = 64 \;\Rightarrow\; |\psi_{in}\rangle_6 = \begin{pmatrix} 0 \\ 0 \\ \alpha_{i1} \\ 0 \\ \vdots \\ \alpha_{i2} \\ \vdots \\ \alpha_{i_N} \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0000 \\ 0001 \\ \cdots \\ 1000 \\ 1001 \\ \cdots \\ 1111 \end{pmatrix} 2^4 = 16 \;\Rightarrow\; |\Phi_{out}\rangle_4 = \begin{pmatrix} 0 \\ \beta_{o1} \\ \beta_{o2} \\ \vdots \\ \beta_{o_N} \\ 0 \end{pmatrix}$$

Q operations must be reversible

$$\boxed{|\Phi_{out}\rangle_{6+4} = U_Q\,|\psi_{in}\rangle_{6+4}}$$

- *A superposition of 2^{6+4} strings of bits*
- *Each string has 6 input bits and 4 output bits*
- *Every non-zero probability string contains an input and the corresponding label*
- *So, we simultaneously calculated answers for all input patterns*

This is called ***Quantum Parallelism***

Fundamentals - Continue…

The catch of Quantum parallelism

After only one computation, up to 2^n evaluations of the function are encoded in the final state as possibilities for extracting those values.

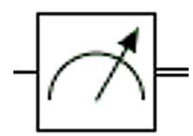
If we have 100 sensory neurons

=> $2^{100} \approx 10^{30}$ evaluations of the function represented by the network.

The catch:

$$|\Phi_{out}\rangle_{n+m} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_{2^{n+m}} \end{pmatrix} \quad <=> \quad \begin{pmatrix} 000000\ 0000 \\ 000000\ 0001 \\ \boxed{\dots\dots\dots\dots} \\ \dots\dots\dots\dots \\ \underbrace{111111}_{n}\ \underbrace{1111}_{m} \end{pmatrix}_{2^{n+m}}$$

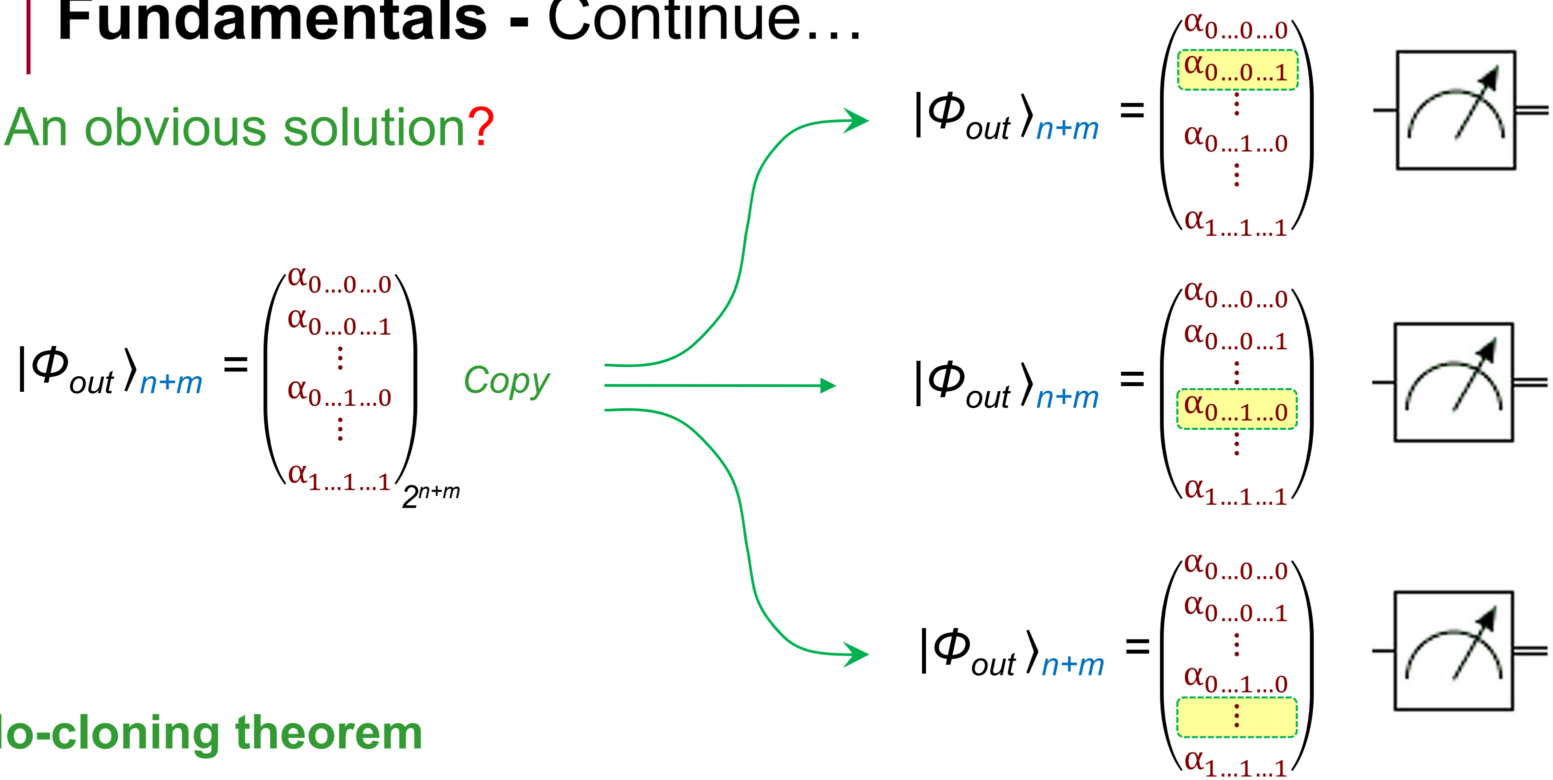
2^n evaluations
in parallel



After the measurement, the state of the input-output registers reduces to a string of bits for just one input pattern, and the bits of the corresponding label.
We are no longer able to learn anything about the labels for any other input pattern.

Fundamentals - Continue…

An obvious solution?



No-cloning theorem

It is impossible to create an independent and identical copy of an arbitrary unknown quantum state.

(BTW, it is extremely valuable for Q communications/cryptography)

Despite this, potential **benefits** of **Q Parallelism** are enormous, e.g.:

- Find relations between different results (e.g., a period of a function)
- Reliable sampling from a probability distribution
- Maybe, can use the entire superposition of all inputs-w-labels for training

Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs "apples to bees"..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - $\langle$ Bra|Ket $\rangle$, $\otimes$
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm => linear regression, PCA, SPV…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

Huge promise – linear algebra.

- The most mature directions in Q ML:
  - Linear Algebra
  - Optimization
  - Sampling

- QLSA (Q linear system algorithm)
  - Also called HHL [Harrow, Hassidim, and Lloyd, 2009]
  - Solves in Logarithmic time => a “mini-revolution” in QML.
  - It was extended or used as a subroutine in other Q ML algorithms.

HHL was extended or used as a subroutine in other Q ML algorithms

- WBL - Quantum Algorithm for Data Fitting: N. Wiebe, D. Braun, and S. Lloyd (2012)
- Linear Regression: Schuld, M., Sinayskiy, I. and Petruccione, F. (2016), “Prediction by Linear Regression on a Quantum Computer”
- Q PCA: Lloyd, S., Mohseni, M. & Rebentrost, P. (2014), “Quantum principal component analysis.”
- Q SPV: P. Rebentrost, M. Mohseni, S. Lloyd (2014), “Quantum support vector machine for big data classification. ”

Q linear system algorithm (HHL algorithm)

Given:

A – an N x N matrix

$\vec{b}$ – a vector

Find:

$\vec{x}$ – a vector, satisfying: $A\vec{x} = \vec{b}$

HHL algorithm:

- Represent $\vec{b}$ as a Q state: $|b\rangle = \sum_{i=1}^{N} b_i |i\rangle.$ $|b\rangle = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$
- Apply e^{iAt} to $|b\rangle$ for various values of the time t.
- Phase estimation technique,
  decompose $|b\rangle$ into eigenbasis of A,
  find corresponding eigenvalues λ_j:
  $$|b\rangle = \sum_{j=1}^{N} \beta_j |u_j\rangle.$$
  $$\sum_{j=1}^{N} \beta_j |u_j\rangle |\lambda_j\rangle$$
- After a few additional steps, left with a state proportional to:
  $$\sum_{j=1}^{N} \beta_j \lambda_j^{-1} |u_j\rangle = A^{-1}|b\rangle = |x\rangle.$$

MISSISSIPPI STATE UNIVERSITY

Caveats of HHL

(1) The input state preparation into $|b\rangle$ - steals from the exponential speed up of the HHL (common problem for many Q algorithm).

(2) The are restrictions on A in $A\vec{x} = \vec{b}$

(sparsity, time needed for A inversion).

(3) The solution is not $\vec{x}$, but a Q superposition state: $|x\rangle = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$

- To get individual values x_i, need many repetitions of the algorithm, the number of repetitions proportional to N.

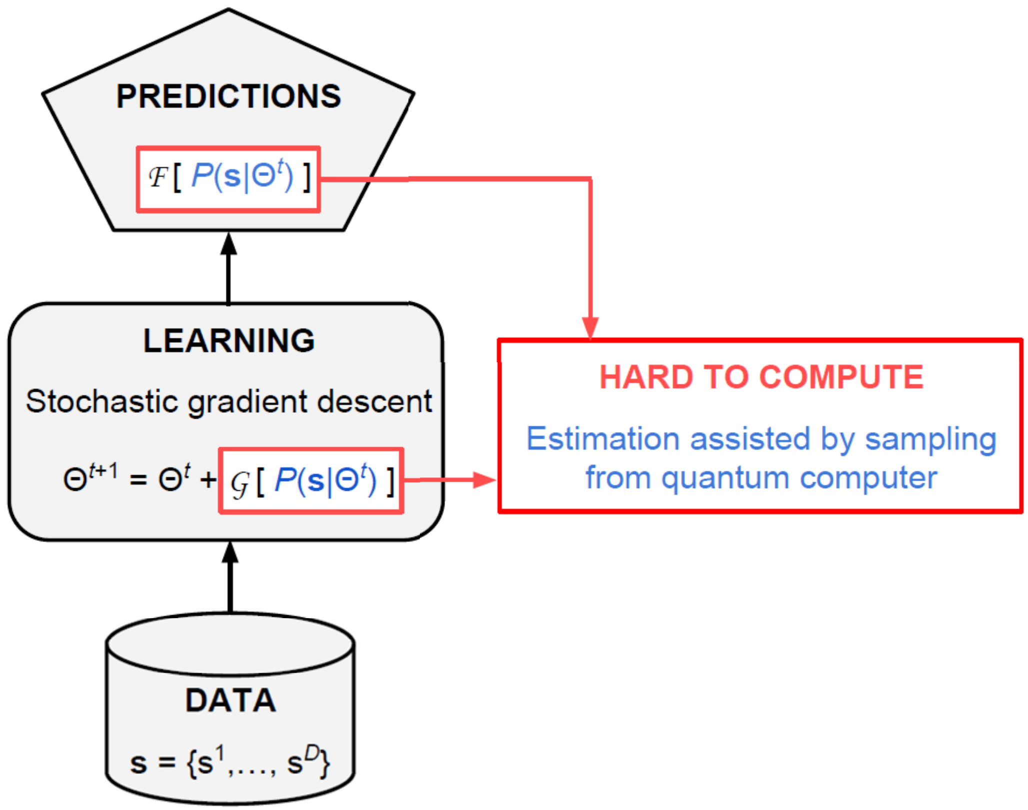
- Realistically, $|x\rangle$ easily reveals only the biggest entries of $\vec{x}$, or, the expectation value of some operator/matrix $\langle k|M|k\rangle$.

NISQ

- 2011 – first commercial Quantum Annealer by D-Wave.
- 2016 – first cloud-based (gated) quantum computer by IBM.
- NISQ (Noisy Intermediate-Scale Quantum) computers:
  - Small number of qBits (50-100 qBits today on Gated QC; a few thousands on QAs).
  - De-coherence, noise/errors – limited circuit depth.
  - Limited connectivity between qBits.
- Focus on the areas where Classical ML struggles (generative models, sampling, etc).
- Hybrid Classical-Quantum ML algorithms – particularly promising for NISQ.

General scheme for hybrid quantum-classical algorithms

[A. Perdomo-Ortiz *et al* (2018) "Opportunities and challenges for quantum-assisted machine learning in near-term quantum computer," *Quantum Sci. Technol.]*



- Besides AQA, sampling can be done using QAOA on a Gated QC.

Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs "apples to bees"..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, ⊗
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QA for NISQ
  - Optimization
  - Sampling
- Gated QC => Variational ML, Q NNs

A million-dollar problem

Find the global energy minimum of the objective function:

$$E(s) = -\sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i \qquad s_i \in \{-1,1\}$$

A broad range of hallmark optimization problems can be mapped onto quadratic unconstrained binary optimization problems.

E.g.

- Satisfiability problems (*k*-SAT)
- The traveling salesman problem
- Knapsack problem
- Graph coloring
- *MRFs*

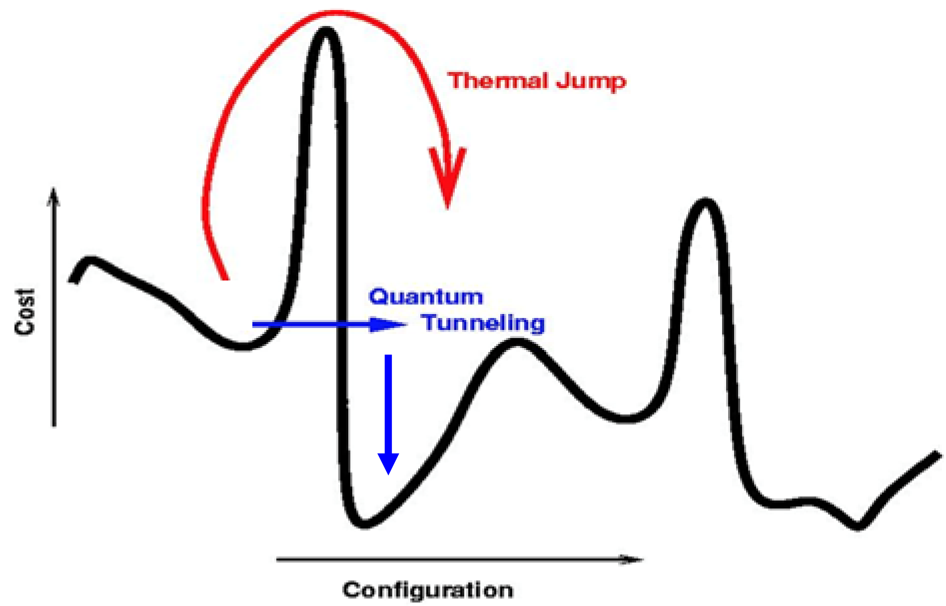
This problem is suited for NISQ:

- *A difficult part (find global min) can be done on AQC*
- *Parameter (J_{ij}, h_i) adjustment/learning – on Classical Computer.*

Simulated Annealing vs Quantum Annealing

Simulated (thermal) annealing (SA)

- Escape local minima via thermal fluctuations.
- "Jump over" energy barriers.



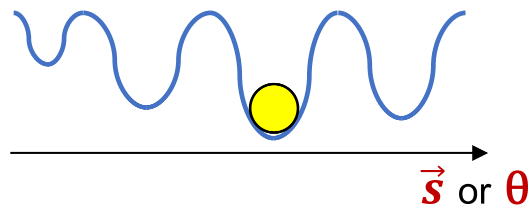
Quantum annealing (QS)

- Escape local minima via quantum fluctuations.
- Tunnel through energy barriers.

Y. Koshka, 2021

Two applications of Adiabatic Quantum Annealing in ML

Parameter optimization

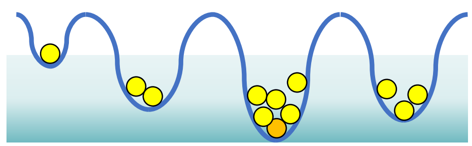


Our problem: $\theta_{opt} = arg\ min_{\theta} f(\theta)$

$f(\theta) \rightarrow E(s)$ of Adiabatic Quantum Annealer

$$D\text{-}Wave: \quad E(\mathbf{s}) = -\sum_{i=1}^{N-1}\sum_{j=1+1}^{N} J_{ij} s_i s_j - \sum_{j=1}^{N} h_j s_j$$

Sampling

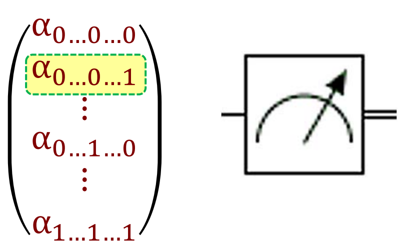


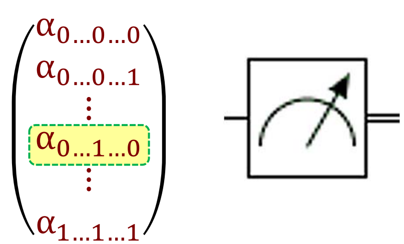
e.g., Restricted Boltzmann Machines (RBMs)

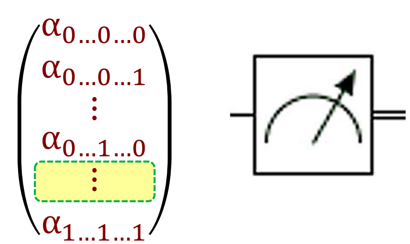
$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})/T}$$

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} \omega_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

$$E(s) = -\sum_{i=1}^{N-1}\sum_{j=1+1}^{N} J_{ij} s_i s_j - \sum_{j=1}^{N} h_j s_j$$





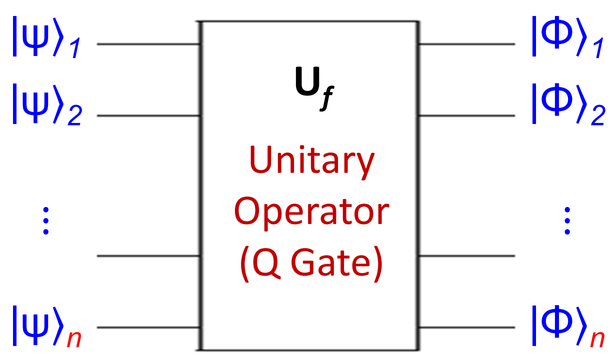


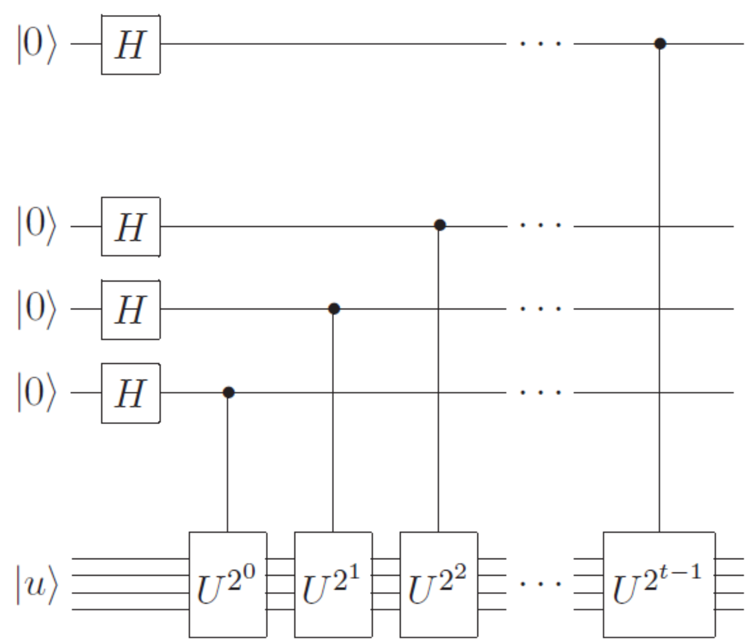
Outline

- Why Quantum Computing (QC)
  - Can QC help my branch of ML? Now or how soon?
  - What do I need to know to understand published Q algorithms
- Apples to oranges… vs "apples to bees"..
  - Classical, classical Probabilistic and Quantum computers
  - Probabilistic ML and Q ML (dead-or-alive vs simultaneously dead-and-alive)
- Fundamentals
  - ⟨ Bra|Ket ⟩, ⊗
  - Why does do want to use density matrix formalism in Q ML
  - Popular misconceptions about Q parallelism
  - No-cloning
- Huge promise – linear algebra.
  - HHL algorithm, linear regression, PCA…
  - Difficulties
- NISQ (Noisy intermediate-scale quantum)
- Adiabatic QC for NISQ
  - Optimization
  - Sampling
- **Gated QC => Variational ML, Q NNs**

Gate model vs Adiabatic QA

- ❑ While an Adiabatic Q Annealer has become the first commercial QC…
- ❑ … the Gate model is behind all the famous Q algorithms.
- ❑ AQA and Gated are equivalent (with a significant conversion overhead).
- ❑ Gated QC - the promise for the universal/general-purpose QC.



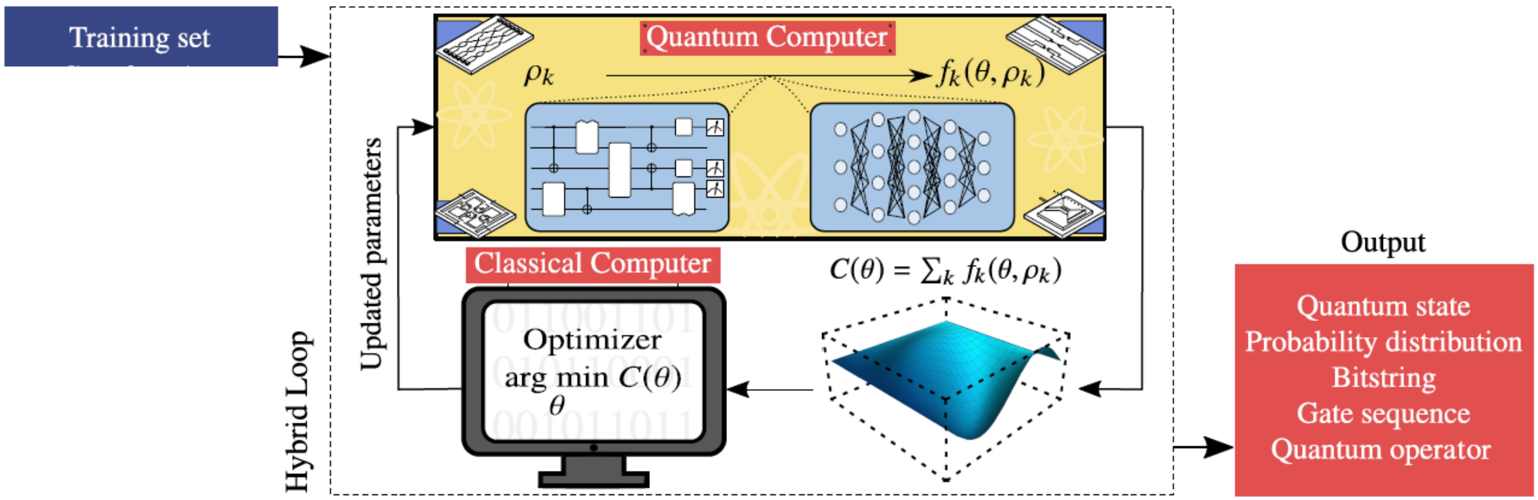


Why Hybrid Classical-Quantum algorithms?

- Difficult to expect killer apps (based on Q linear algebra) on 100-1000 qBit devices.
- Q algorithms offering Polynomial-Exponential speed-up may require high circuit depth…
- … but the noise limits the circuit depth.
- Instead, focus on the areas where Classical ML struggles (generative models, sampling, etc).
- Hybrid Classical-Quantum ML algorithms – particularly promising for NISQ.
- Variational Q algorithms (VQAs) – a classical optimizer to train parametrized Q circuit.

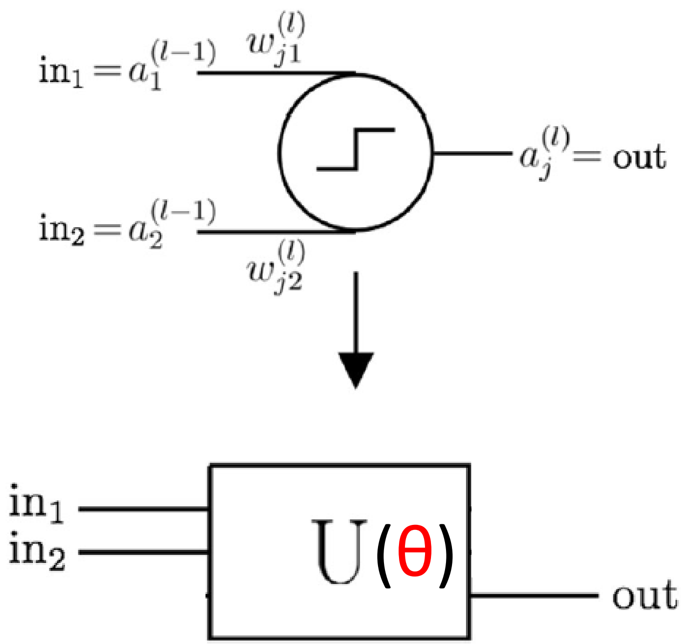
Schematic diagram of a Variational Quantum Algorithm (VQA).

[M. Cerezo *et al* (2020), "Variational Quantum Algorithms," *arXiv:2012.09265]*



Quantum NNs

[Kwok Ho Wan *et al* (2017), "Quantum generalization of feedforward neural networks," *npj Quantum Information]*



$$\delta w_{jk}^{(l)} = -\eta \frac{\partial C}{\partial w_{jk}^{(l)}},$$

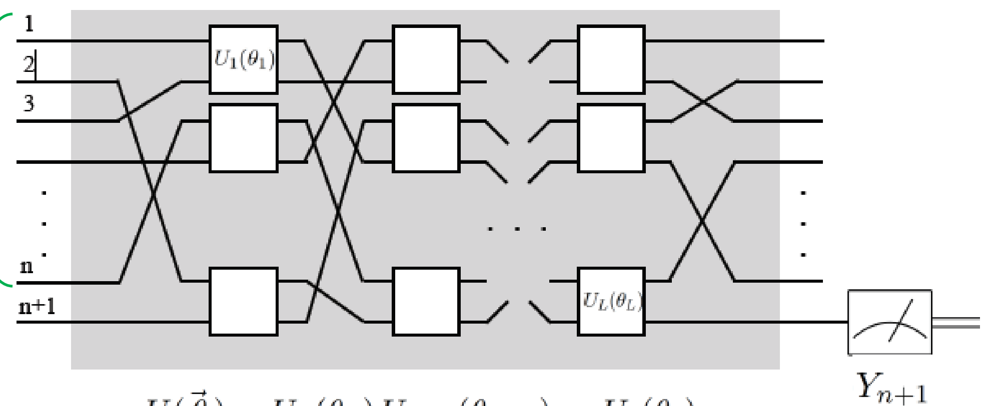
- Classical NNs: non-linearities are critical.
- Q Unitaries: linear operations.
- Q NN: non-linearities come from Q measurements.

- Demonstrated a way of building Q autoencoder (but no-cloning may be an issue).

Quantum NNs

[E. Farhi *et al* (2018), Classification with Quantum Neural Network on Near Term Processors”.

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix}_n \Rightarrow \begin{matrix} |0\rangle \\ |0\rangle \\ |1\rangle \\ \vdots \\ |1\rangle \\ |0\rangle_n \end{matrix} = |\psi\rangle$$



$$U(\vec{\theta}) = U_L(\theta_L)U_{L-1}(\theta_{L-1})\ldots U_1(\theta_1)$$

- A possibility to have $|\psi\rangle$ = a superposition of all training patterns for training.

[M. Schuld *et al* (2018), “Circuit-centric quantum classifiers]

Can be used for classifying also real-valued data-vectors $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2^N} \end{pmatrix}$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix}_{n=2^M} \Rightarrow |\psi\rangle_M = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \Rightarrow$$

Use only M = log n qBits

qBits

bits

- Amplitude encoding rather than encoding into qBits.
- Used angles of a Q state as learnable parameters (instead of parametrization with Pauli matrices).
- Emphasized importance of the circuit preparing strongly entangled quantum states.

MISSISSIPPI STATE UNIVERSITY

Y. Koshka, 2021

Conclusion

- There are realistic opportunities to benefit from the near-term QC hardware
- While the near-term benefits of QC in general remain uncertain, Q ML research is likely to bring some killer apps.
- It is important to understand caveats associated with the most promising algorithms
- The potential benefits are huge and exciting, but usually not what science journalist promise the public.