Tutorial on QuantumFlow: A Co-Design Framework of Neural Network and Quantum Circuit towards Quantum Advantage

Session 5: Roadmap of Quantum Machine Learning

Weiwen Jiang, Ph.D.

Assistant Professor

Electrical and Computer Engineering

George Mason University

wjiang8@gmu.edu

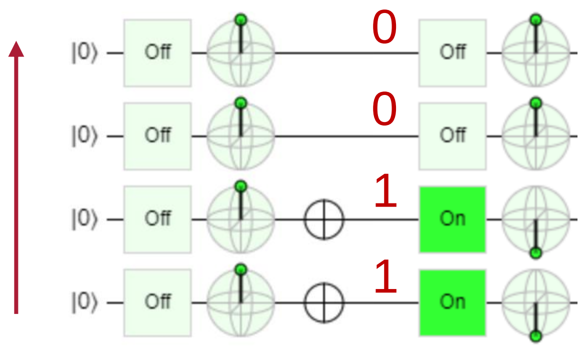
https://jqub.ece.gmu.edu

Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
- **Call for paper at “Electronics”**
- **Conclusion**
- **Q&A**

What Data Can Be Encoded to Quantum Computers, and how?

- **Can we encode an arbitrary number into quantum computer? Is it efficient?**
  - **Yes / No**



Yes, we can!

Number: 12

Number: 1.5

Number: 0.75

No, because it uses too many qubits!

This encoding is similar to classical bits, where each qubit is regarded as a binary number!

1-to-N mapping! (Boolean Function)

What Data Can Be Encoded to Quantum Computers, and how?

- Can we encode an arbitrary number into quantum computer? Is it efficient?
  - Yes / No

Yes, we can!

Number: 12

Number: 1.5

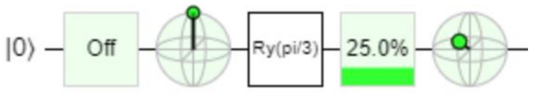
Number: 0.75

No, because it uses too many qubits!

This encoding is similar to classical bits, where each qubit is regarded as a binary number!

1-to-N mapping! (Boolean Function)

- **Can we take use of superposition of qubits to encode data? Is this solution perfect?**
  - **Yes / No**



Yes, we can!

0.25 in one qubit

No, (1) data needs in the range of [0,1]!
(2) same complexity *O(1)* as classical

1-to-1 mapping! (Angle Encoding)

What Data Can Be Encoded to Quantum Computers, and how?

- Can we encode an arbitrary number into quantum computer? Is it efficient?
- Yes / No

Yes, we can!

Number: 12

Number: 1.5

Number: 0.75

No, because it uses too many qubits!

This encoding is similar to classical bits, where each qubit is regarded as a binary number!

1-to-N mapping! (Boolean Function)

- Can we take use of superposition of qubits to encode data? Is this solution perfect?
- Yes / No

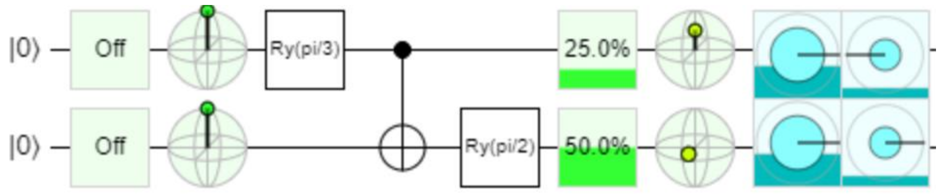
Yes, we can!

0.25 in one qubit

No, (1) data needs in the range of [0,1]!
(2) same complexity *O(1)* as classical

1-to-1 mapping! (Angle Encoding)

- **Can we take use of entanglement of qubits to encode data? Is this solution perfect?**
- **Yes / No**



Yes, 4 # in 2 qubits

$$\begin{bmatrix}+0.61 & -0.35\\ +0.61 & +0.35\end{bmatrix}$$

No, (1) sum of the square of data need to be 1
(2) may have high cost to encode data

N-to-logN mapping! (Amplitude Encoding)

Encoding: 1-to-N *v.s.* 1-to-1 *v.s.* N-to-*log*N

Data Encoding	# of Qubit (C v.s. Q)	Data Limitation	Encoding Complexity
1-to-N	O(1) v.s. O(N)	**Almost No!**	**Low**
1-to-1	O(1) v.s. O(1)	[0,+1]	**Low**
N-to-logN	**O(N) v.s. O(*log*N)**	[-1,+1] and $\sum x^2 = 1$	High

Agenda – Session 1: Introduction

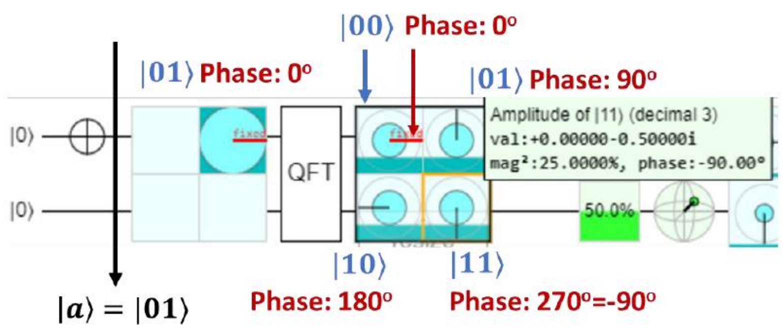
- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
- **Call for paper at “Electronics”**
- **Q&A**

Quantum Fourier Transform (1-to-N)

Problem to be solved: Encoding the binary number represented by states to phase.

QFT, inspired by the discrete Fourier transform, is the **linear operator** defined over an orthonormal basis $|0\rangle, \cdots |N-1\rangle$ of a **N-dimensional complex vector space**, as

$$|\boldsymbol{a}\rangle \xrightarrow{QFT} |a'\rangle = \frac{1}{\sqrt{\boldsymbol{N}}} \sum_{k=0}^{N-1} e^{\frac{2\pi i}{N}\boldsymbol{ak}} |\boldsymbol{k}\rangle$$



Input:

|a⟩ = |01⟩ → **a**=1

$\mathbf{N = 2^2 = 4}$

$$QFT * |a\rangle = \frac{1}{\sqrt{N}} \times (x_0 \cdot |00\rangle + x_1 \cdot |01\rangle + x_2 \cdot |10\rangle + x_3 \cdot |11\rangle)$$

$$x_0 = e^{\frac{2\pi i}{N}ak} = e^{\frac{2\pi * a * k}{N}i} = e^{\frac{2\pi * 1 * 0}{4}i} = e^0$$

(note: k=0 since we consider |k⟩ =|00⟩)

$$x_1 = e^{\frac{2\pi i}{N}ak} = e^{\frac{2\pi * a * k}{N}i} = e^{\frac{2\pi * 1 * 1}{4}i} = e^{\frac{1\pi}{2}i}$$

(note: k=1 since we consider |k⟩ =|01⟩)

$$x_2 = e^{\frac{2\pi i}{N}ak} = e^{\frac{2\pi * a * k}{N}i} = e^{\frac{2\pi * 1 * 2}{4}i} = e^{\pi i}$$

$$x_3 = e^{\frac{2\pi i}{N}ak} = e^{\frac{2\pi * a * k}{N}i} = e^{\frac{2\pi * 1 * 3}{4}i} = e^{\frac{3\pi}{2}i}$$

$$=> \quad QFT * |a\rangle = \frac{1}{2} \times \left(e^0 \cdot |00\rangle + e^{\frac{\pi}{2}i} \cdot |01\rangle + e^{\pi i} \cdot |10\rangle + e^{\frac{3\pi}{2}i} \cdot |11\rangle\right)$$

Quantum Phase Estimation (1-to-N for Output)

Problem to be solved: Extract the phase to binary number represented by states.

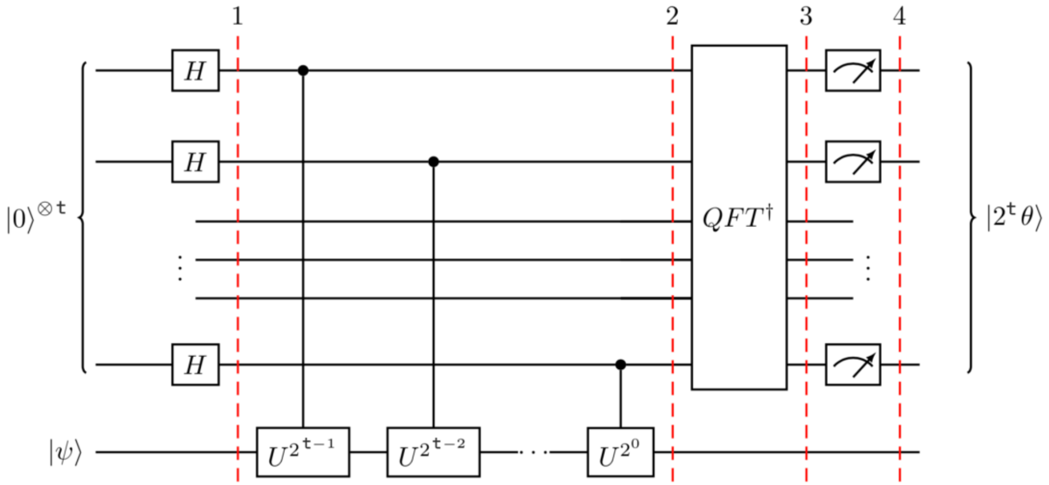
Given a unitary operator *U*, the algorithm estimates θ in $U|\Psi\rangle = e^{2\pi i\theta}|\Psi\rangle$. Here, $|\Psi\rangle$ is an eigenvector and $e^{2\pi i\theta}$ is the corresponding eigenvalue. Since *U* is unitary, all of its eigenvalues have a norm of 1.

Why: Difficulty in Measuring the Phase.

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ → $|0\rangle$ with prob. $\frac{1}{2}$; $|1\rangle$ with prob. $\frac{1}{2}$

$\frac{1}{\sqrt{2}}e^{i\pi/2}(|0\rangle + |1\rangle)$ → $|0\rangle$ with prob. $\frac{1}{2}$; $|1\rangle$ with prob. $\frac{1}{2}$

How?



Given:

- We have the implementation of operator U.
- U^t = Repeat the operator U for t times

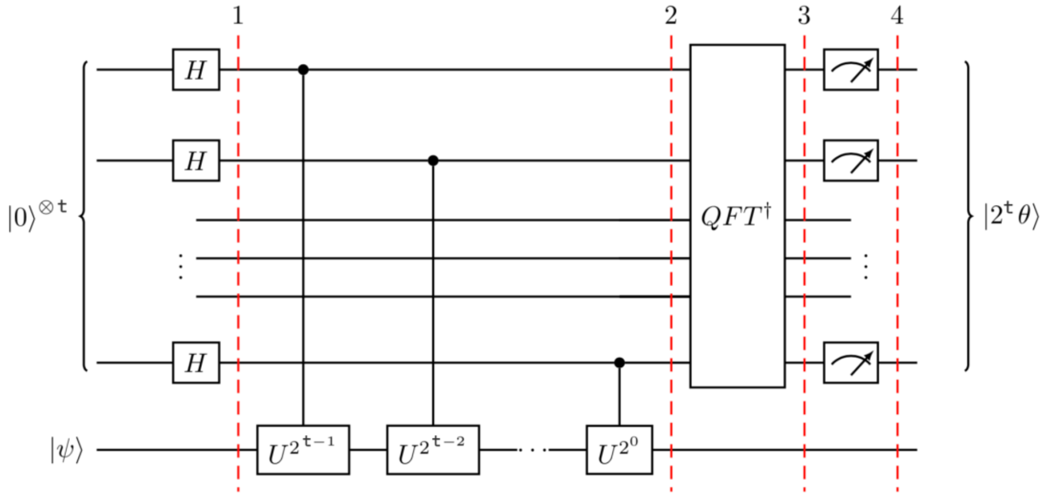
Quantum Phase Estimation: Example

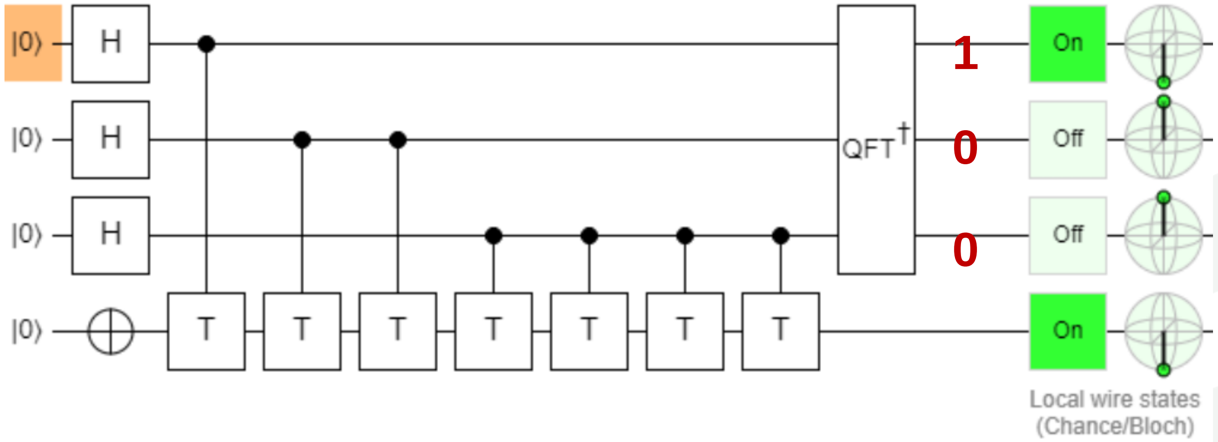
$|\Phi\rangle$

$$\boxed{T}\,\boxed{|1\rangle} = \begin{bmatrix}1 & 0\\ 0 & e^{\frac{i\pi}{4}}\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = e^{\frac{i\pi}{4}}\,|1\rangle$$

Given $T\,|1\rangle = e^{2i\pi\theta}|1\rangle$

$\theta = \frac{1}{8}$





https://algassert.com/quirk#circuit={%22cols%22:[[%22H%22,%22H%22,%22H%22,%22X%22],[%22%E2%80%A2%22,1,1,%22Z^%C2%BC%22],[1,%22%E2%80%A2%22,1,%22Z^%C2%BC%22],[1,%22%E2%80%A2%22,1,%22Z^%C2%BC%22],[1,1,%22%E2%80%A2%22,%22Z^%C2%BC%22],[1,1,%22%E2%80%A2%22,%22Z^%C2%BC%22],[1,1,%22%E2%80%A2%22,%22Z^%C2%BC%22],[1,1,%22%E2%80%A2%22,%22Z^%C2%BC%22],[%22QFT%E2%80%A03%22]]}

0.001 = 1/8

HHL

Problem to solve:

The problem can be defined as, given a matrix **A** $\in \mathbb{C}^{N\times N}$ and a vector $\vec{\boldsymbol{b}} \in \mathbb{C}^{N}$, find $\vec{\boldsymbol{x}} \in \mathbb{C}^{N}$ satisfying **A** $\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}}$

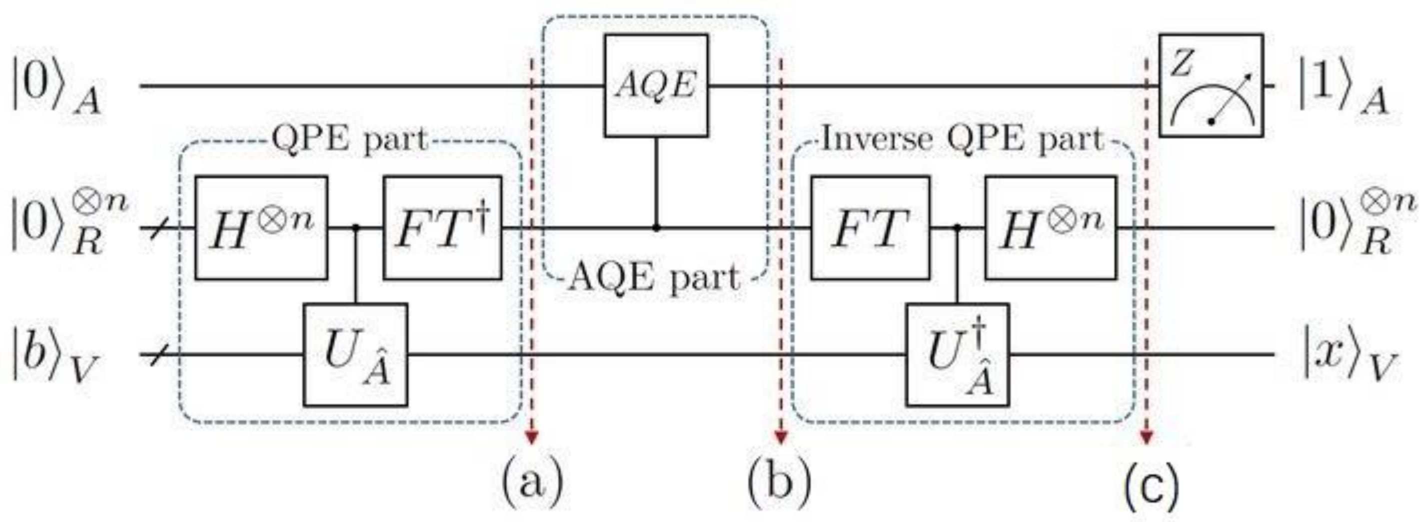
Why: Classical has complexity of $O(N\kappa)$, can quantum reduce the complexity to $O(log(N)\kappa^2)$. N is the number of variables in the linear system. κ is a low condition number.

How?

Given:

- Hermitian matrix A
- b

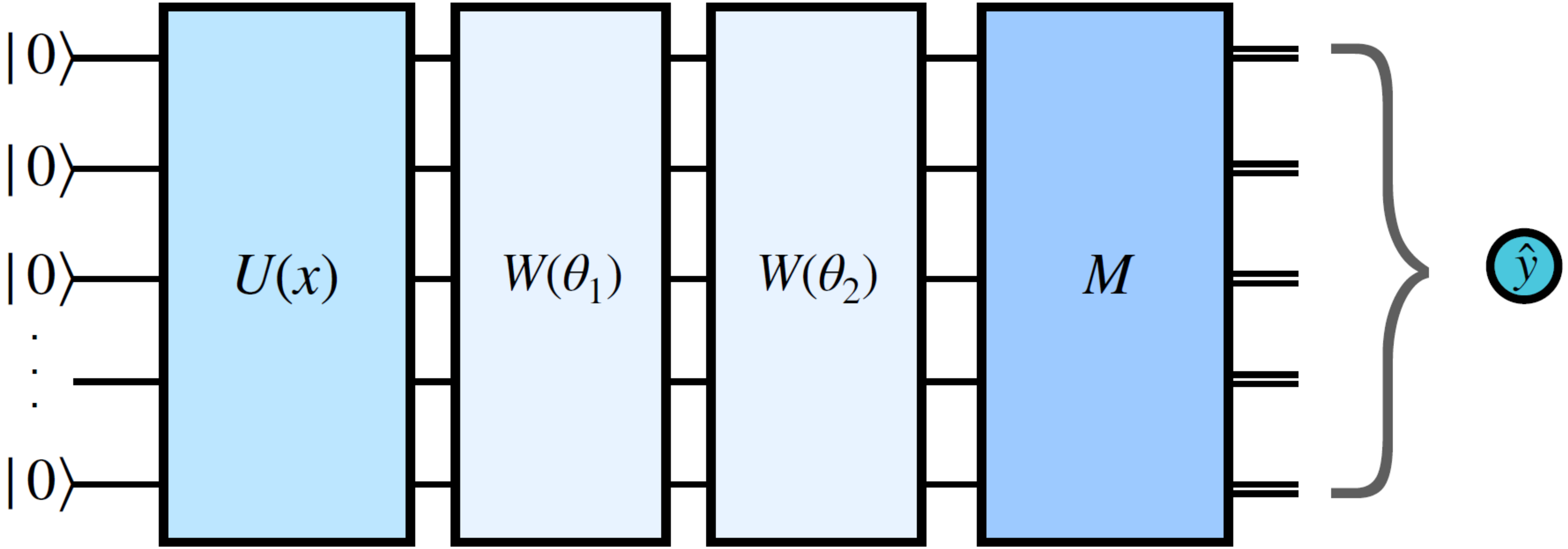
Ancilla quantum encoding (AQE)



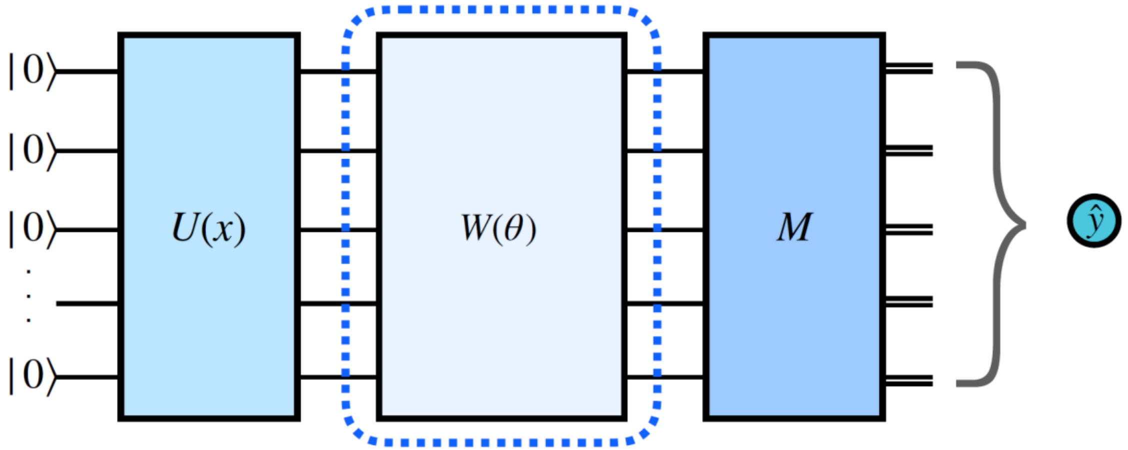
Agenda – Session 1: Introduction

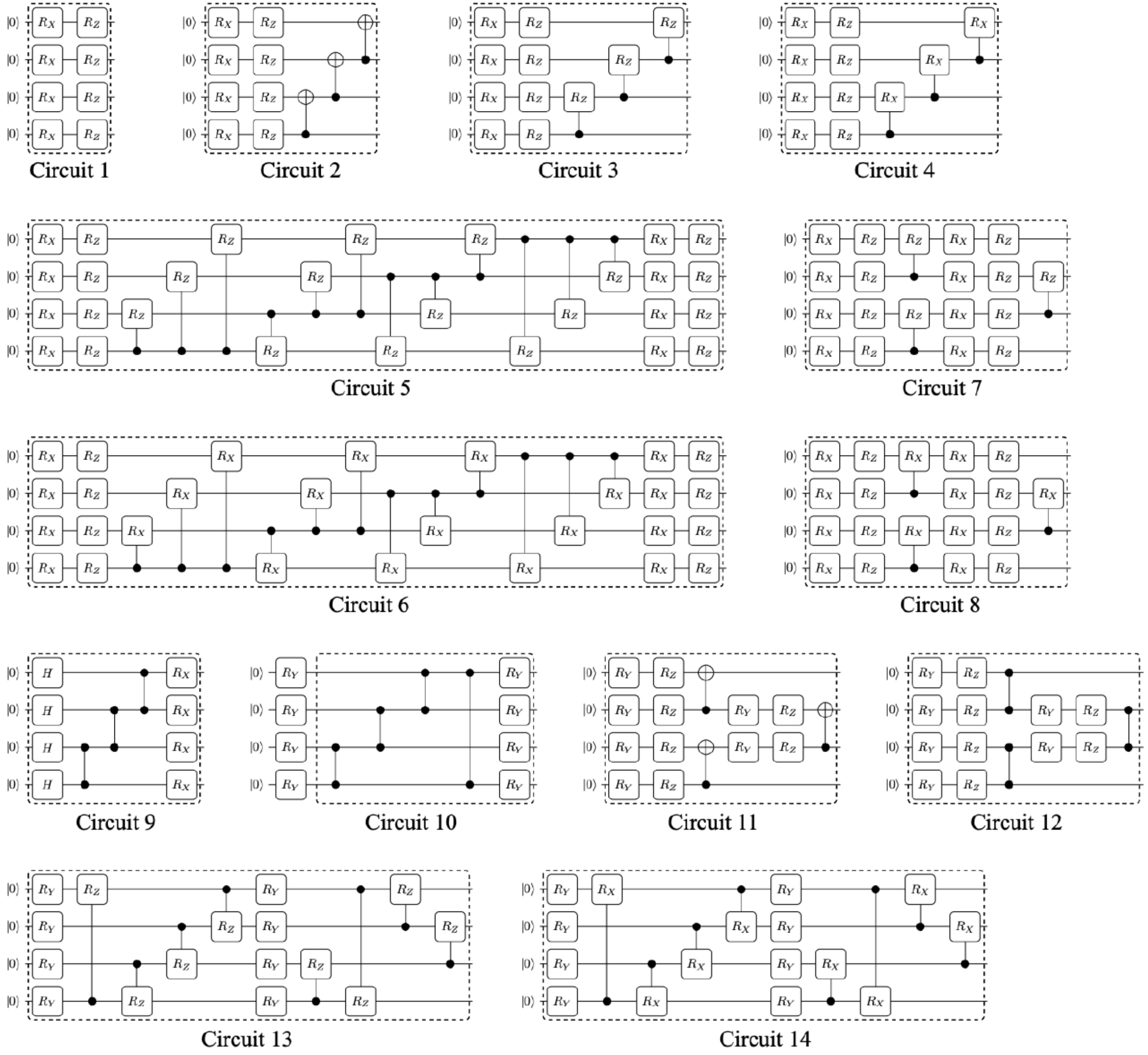
- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
- **Call for paper at "Electronics"**
- **Q&A**

Variational Classifier (1-to-N or 1-to-1 or N-to-*log*N)



Variational Classifier

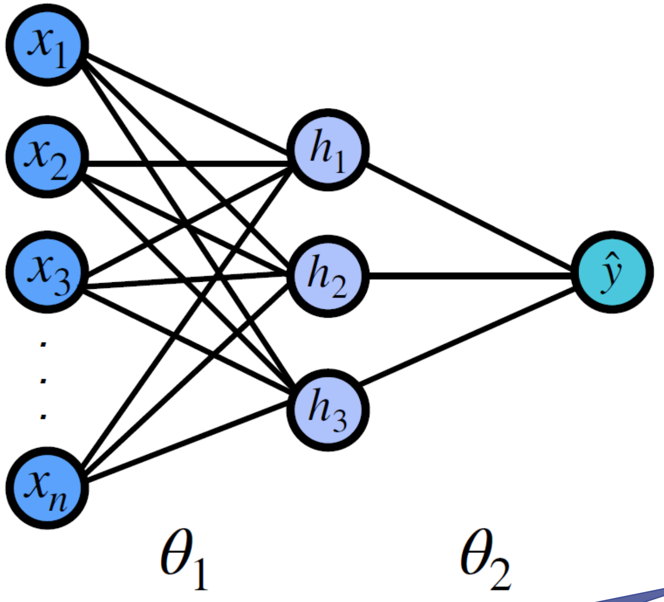




Sim et al. "Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms." Advanced Quantum Technologies 2.12 (2019): 1900070.

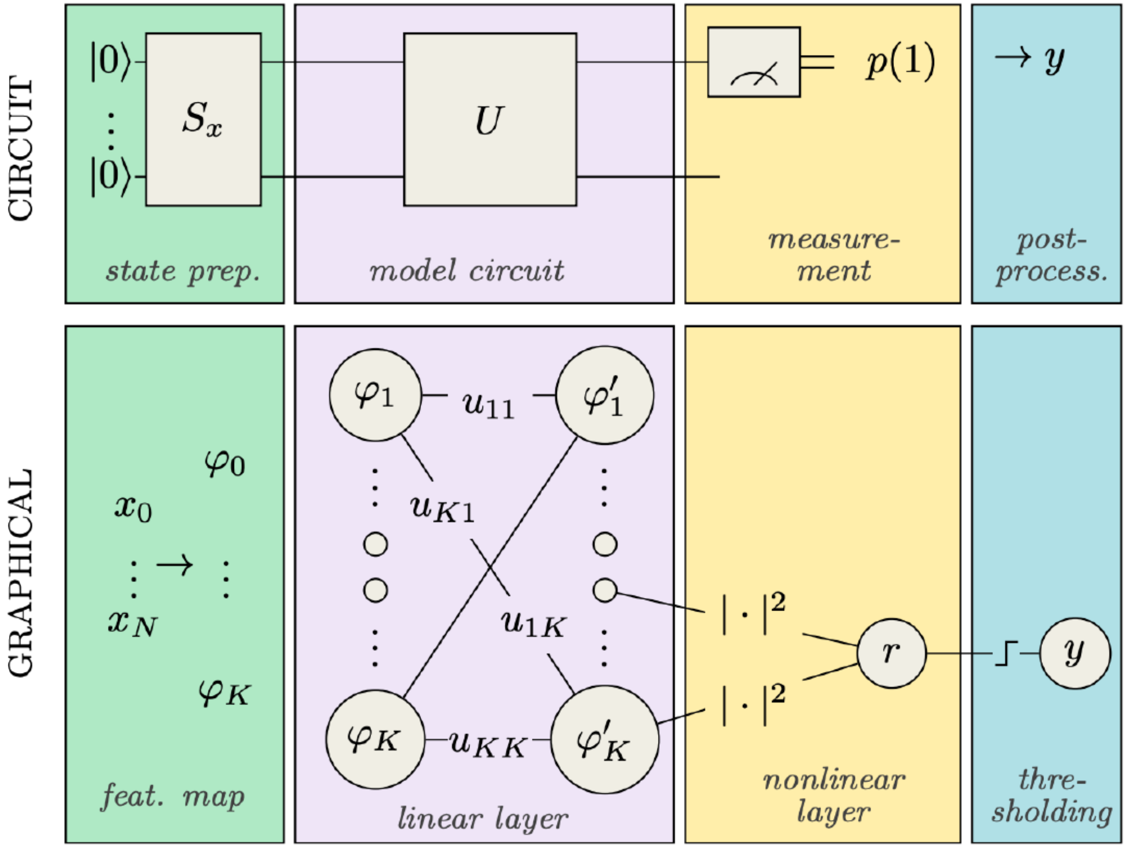
VQC-based Quantum Neural Network

VQC is designed motivated by NN



Weights in VQC has strong correlation

No non-linearity in-between layers

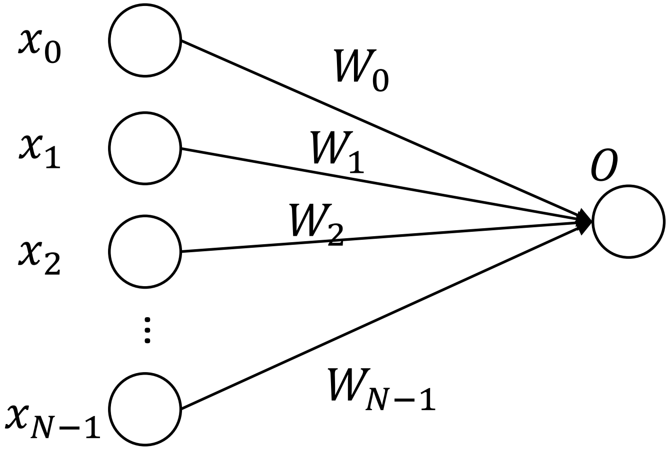


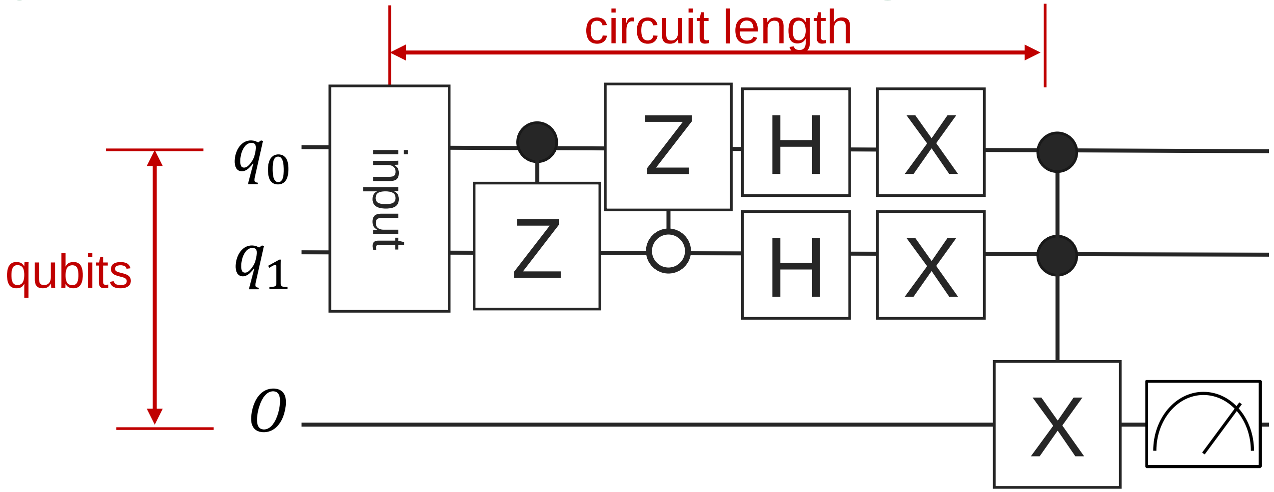
Schuld et al. "Circuit-centric quantum classifiers." Physical Review A 101.3 (2020): 032308.

Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
- **Call for paper at "Electronics"**
- **Q&A**

What's the complexity? Quantum Advantage?





- **Classical computer with 1 MAC**

  $Time$: $O(N)$

  $Space$ *(Comp. Res.)*: $O(1)$

  $\boldsymbol{Time \times Space}$: $\boldsymbol{O(N)}$

- **Classical computer with N MAC**

  $Time$: $O(1)$

  $Space$ *(Comp. Res.)*: $O(N)$

  $\boldsymbol{Time \times Space}$: $\boldsymbol{O(N)}$

- **Time-Space Complexity in Quantum computer**

  $Time$: Circuit Length

  $Space$ *(Comp. Res.)*: Qubits

  $\boldsymbol{Time \times Space\ (T - S)}$: $\boldsymbol{Qubits \times Circuit\ Length}$

- **Given that $T - S$ complexity on classical computer is $\boldsymbol{O(N)}$, Quantum Advantage is achieved if $T - S$ complexity on Quantum can be $\boldsymbol{O(ploylogN)}$ or lower. ------- Exponential Speedup!**

Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
    - Computing using Boolean function (1-to-N)
    - Computing in angles (1-to-1)
    - Computing in amplitude (N-to-*log*N)
  - Applications
- **Call for paper at "Electronics"**
- **Q&A**

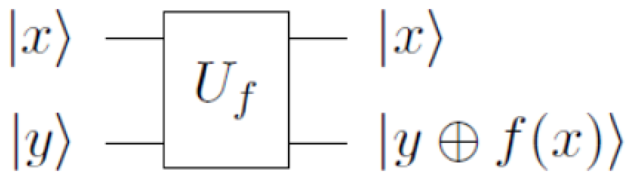
Apply Boolean Function to Realize Any Non-Linear (1-to-N)

Problem to be solved: $U_f : |x\rangle |0\rangle^{\otimes m} \mapsto |x\rangle |f(x)\rangle$

- f can be any non-linear function, say ReLU
- X is a $0.x_1x_2x_3\ldots x_n$ binary format to hold the intermediate data

$$|x,y\rangle \xrightarrow{f} |x, y \oplus f(x)\rangle$$

- X- input register
- Y- output register

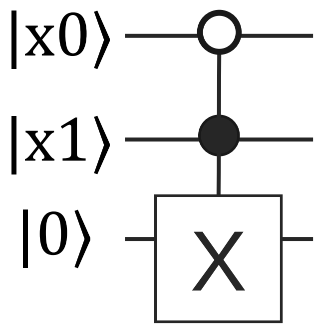


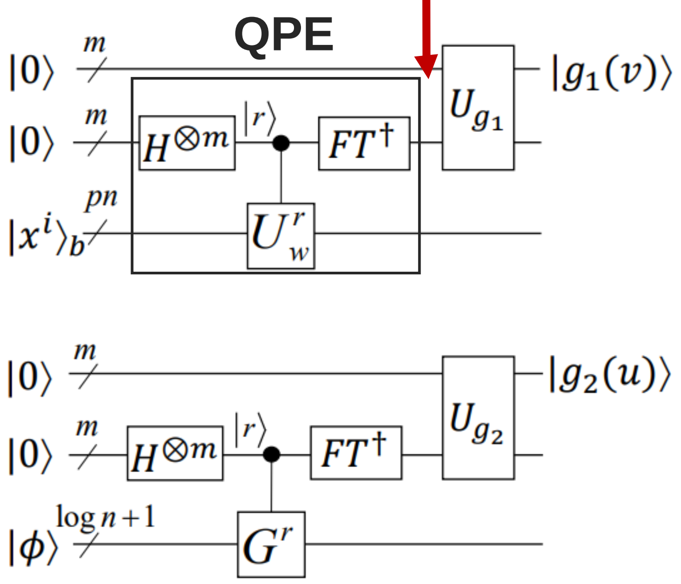
- *f(x)* Boolean function

$$|x\rangle|y\rangle \xrightarrow{init} |x\rangle|0\rangle \xrightarrow{U_f} |x\rangle|ReLU(x)\rangle$$

x	ReLU(x)
00	0
01	1
10	0
11	0

Sign

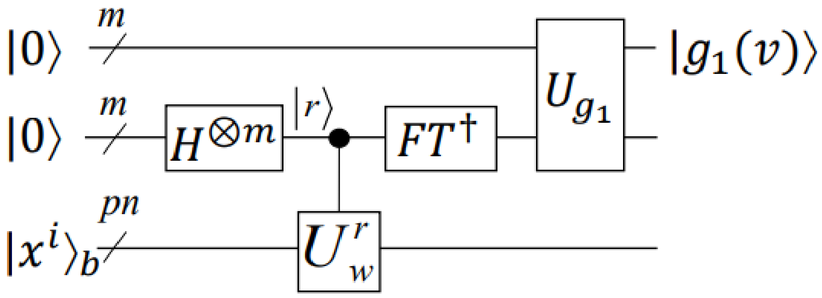




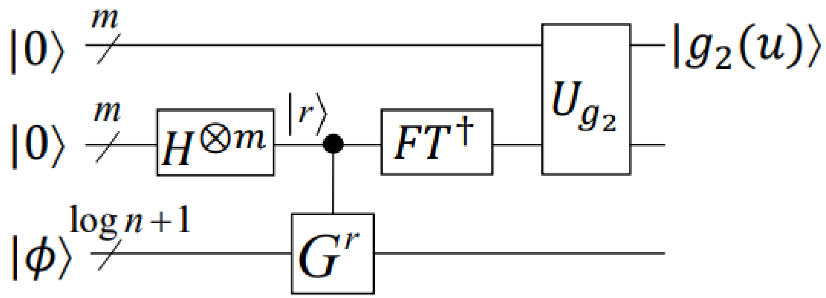
[ref 1] Shilu Yan, et al. , Nonlinear quantum neuron: A fundamental building block for quantum neural networks.
[ref 2] F. M. de Paula Neto , et al. , Implementing Any Nonlinear Quantum Neuron, IEEE TNNLS

Apply Boolean Function to Realize Any Non-Linear (1-to-N)

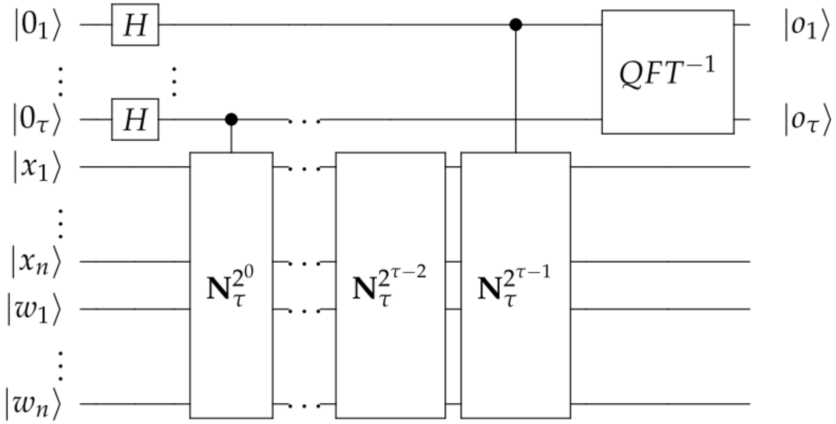
(a) angle encoding



(b) amplitude encoding



(c) [ref]



Comparison of different quantum neurons.

Quantum neurons	Input features	Number of qubits	Gate complexity
(a)	Binary	$pn + 2m$	$O(pn + m2^m)$
(b)	Continuous	$\log n + 2m + 1$	$O(mn^2 + m2^m)$
(c)	Binary	$2pn + m$	$O(m2^{2pn} + m^2)$

n: input data number
p: precision for input
m: precision for output
No Quantum Advantage

[ref] F. M. de Paula Neto , et al. , Implementing Any Nonlinear Quantum Neuron, IEEE TNNLS

Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
    - Computing using Boolean function (1-to-N)
    - Computing in angles (1-to-1)
    - Computing in amplitude (N-to-*log*N)
  - Applications
- **Call for paper at "Electronics"**
- **Q&A**

Quantum Neuron (1-to-1): Linear Part

Problem to be solved (Linear Function):

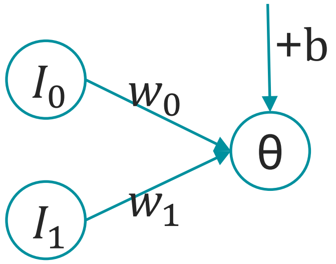
Classical: Given I_0 , I_1 , …, I_n , b , w_0 , w_1 , … w_n ,

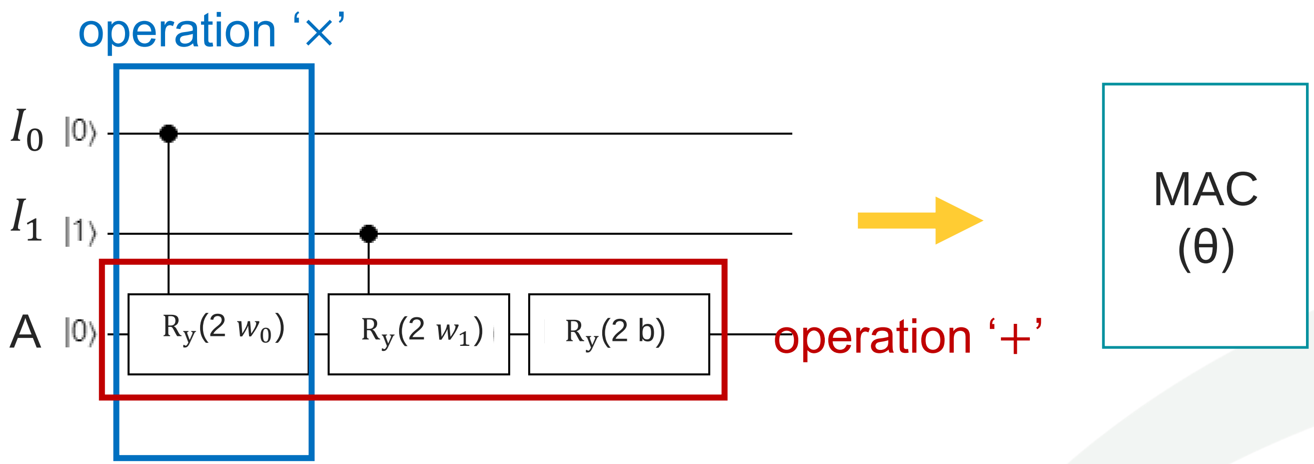
Output: θ = $\sum_{\mathrm{i}=0}^{\mathrm{n}} \mathrm{I_i} \times w_i + \mathrm{b}$

Quantum: Given p_0 , p_1 , …, p_n, $\mathrm{R_y}(2\mathrm{b})$, $\mathrm{R_y}(2w_0)$, $\mathrm{R_y}(2w_1)$,…, $\mathrm{R_y}(2w_n)$

Output: $\mathrm{R_y}(2\theta)$ ---- **θ is computed on angle.**

Example:





Idea: For I_j with Prob{I_j = |**1**⟩ } = p_j

If p_j = 1, then rotate the output qubit “***A***” by 2w_j; otherwise, don’t rotate ***A***

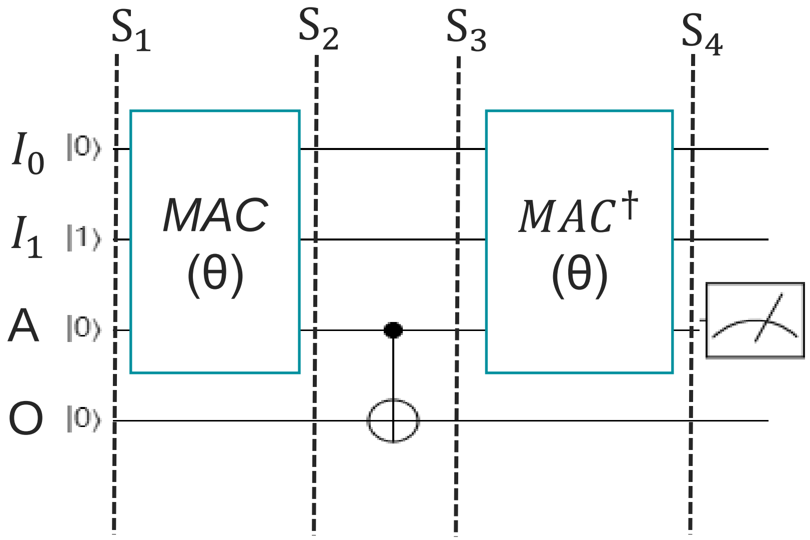
[ref] Cao, Yudong, et al. "Quantum neuron: an elementary building block for machine learning on quantum computers." *arXiv preprint arXiv:1711.11240* (2017).

Quantum Neuron (1-to-1): Non-Linear Part

Problem to be solved (Non-Linear Function):

Classic: Given θ,
Output: $\mathbf{q}(\theta) = \boldsymbol{arctan(tan^2\theta)}$

Quantum: Given $\mathrm{R_y}(2\theta)$
Output: $\mathrm{O} = \mathrm{R_y}(2\mathbf{q}(\theta))$



S1: $|0A\rangle = |00\rangle$

S2: $\mathrm{I} \otimes \mathrm{R_y}(2\theta) \times |00\rangle = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) & -\sin(\theta) \\ 0 & 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ 0 \end{bmatrix}$

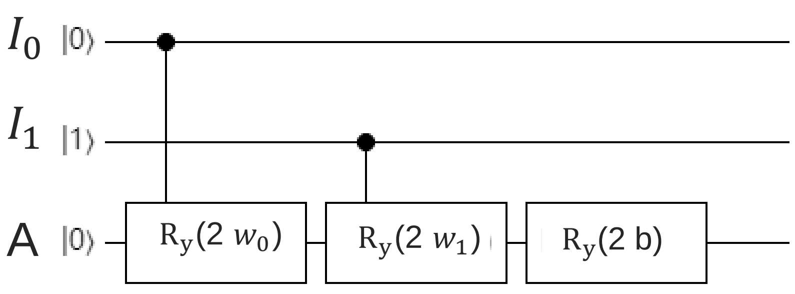
S3: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ 0 \\ 0 \\ \sin(\theta) \end{bmatrix}$

S4: $\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) & -\sin(\theta) \\ 0 & 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \times \begin{bmatrix} \cos(\theta) \\ 0 \\ 0 \\ \sin(\theta) \end{bmatrix} = \begin{bmatrix} \cos^2(\theta) \\ -\sin(\theta)\cos(\theta) \\ \sin^2(\theta) \\ \sin(\theta)\cos(\theta) \end{bmatrix} \begin{matrix} \boldsymbol{|OA\rangle} \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$

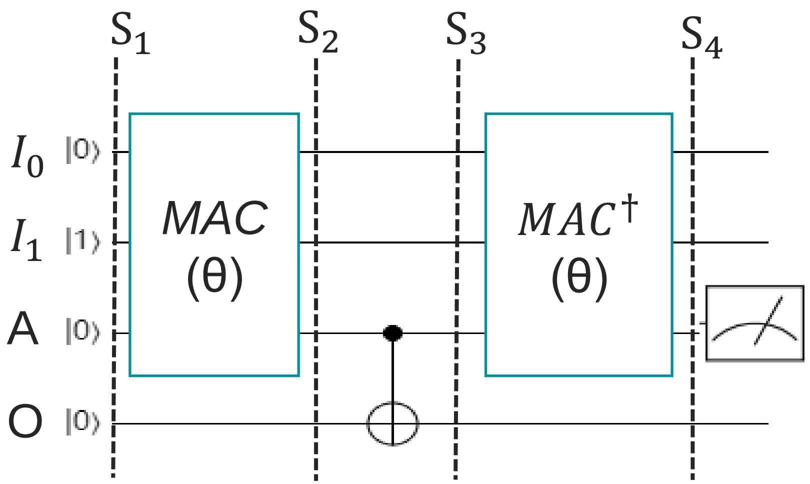
Measure A.
$|A\rangle = |0\rangle$: success
$|A\rangle = |1\rangle$: recover and repeat

$\begin{bmatrix} \cos^2(\theta) \\ -\sin(\theta)\cos(\theta) \\ \sin^2(\theta) \\ \sin(\theta)\cos(\theta) \end{bmatrix} \longrightarrow \begin{bmatrix} \cos^2(\theta) \\ 0 \\ \sin^2(\theta) \\ 0 \end{bmatrix} \rightarrow tan^2\ \theta$

Quantum Neuron (1-to-1): Complexity



(a) Linear computation



(b) Non-linear computation

Input features	Number of Qubits	Number of Gates
Linear	$O(n)$	$O(n)$
Non-linear	$O(1)$	$O(m \cdot n)$

n: input data number
m: repeat number
No Quantum Advantage

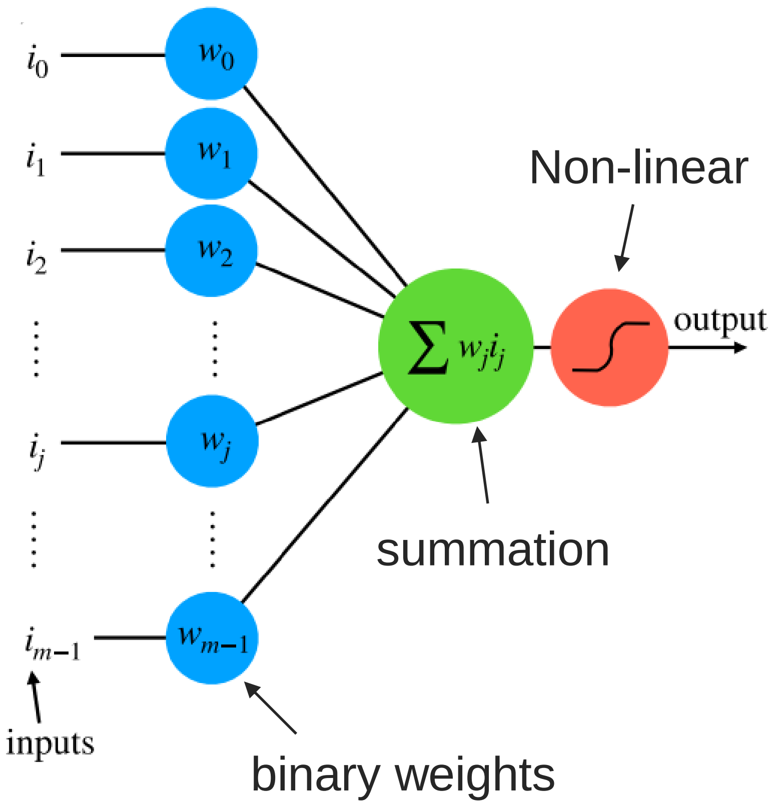
[ref] Cao, Yudong, et al. "Quantum neuron: an elementary building block for machine learning on quantum computers." *arXiv preprint arXiv:1711.11240* (2017).

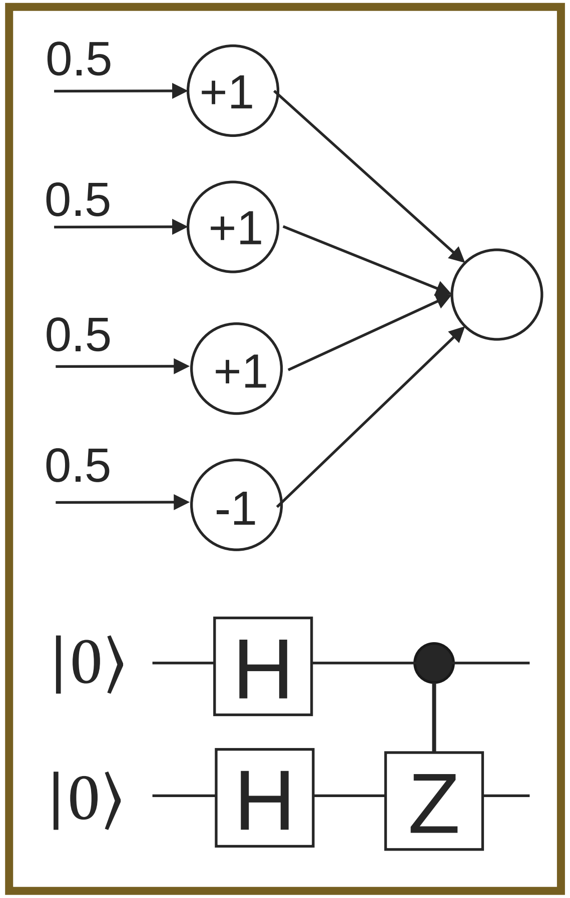
Agenda – Session 1: Introduction

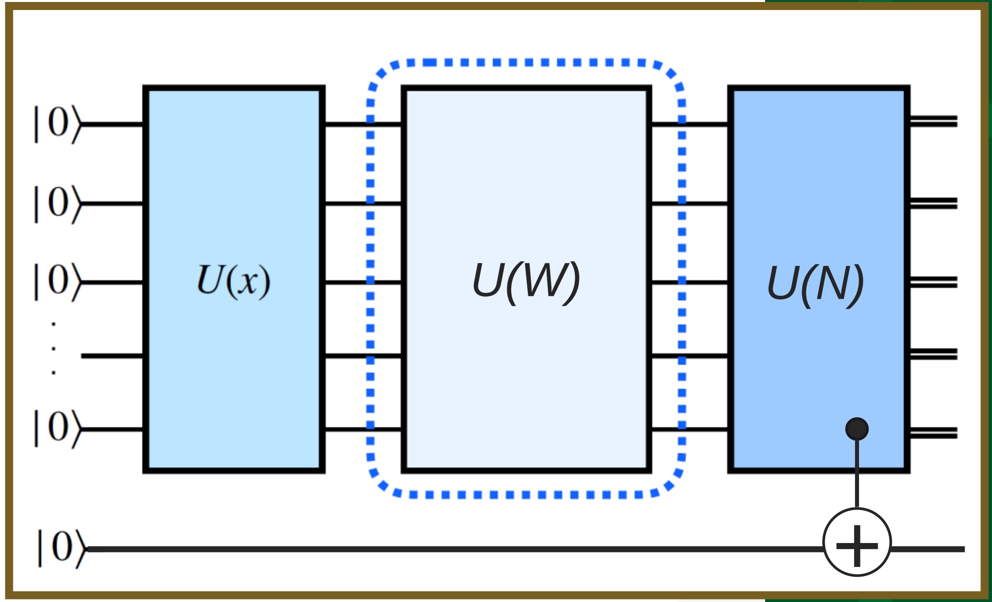
- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
    - Computing using Boolean function (1-to-N)
    - Computing in angles (1-to-1)
    - Computing in amplitude (N-to-*log*N)
  - Applications
- **Call for paper at “Electronics”**
- **Q&A**

Sign Flip on Amplitude for Binary NN (N-to-*log*N)

Problem to be solved (Binary Neural Network):

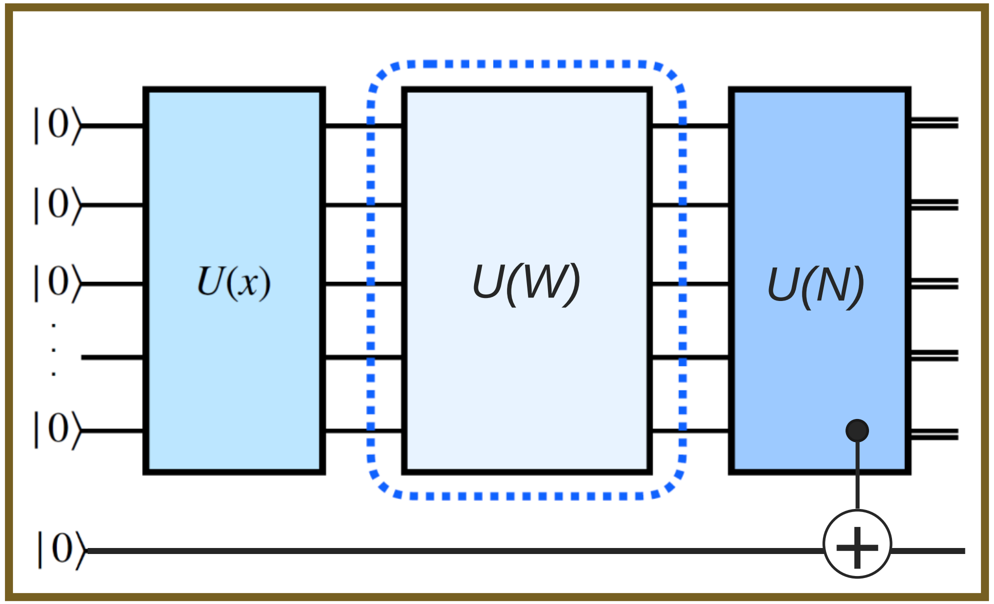






[Ref1] Tacchino, Francesco, et al. "An artificial neuron implemented on an actual quantum processor." *npj Quantum Information* 5.1 (2019): 1-8.
[Ref2] Jiang, Weiwen, Jinjun Xiong, and Yiyu Shi. "When Machine Learning Meets Quantum Computers: A Case Study." *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2021.

Sign Flip on Amplitude (N-to-*log*N): Complexity



Input features	Number of Qubits	Number of Gates
U(x)	$O(log\text{N})$	$\boldsymbol{O}(?)$
U(W)	$O(log\text{N})$	$\boldsymbol{O}(?)$
U(N)	$O(1)$	$O(logN)$

n: input data number
Potential Quantum Advantage

Halfway Takeaway

- **3 typical data encoding without losing information**
  - 1-to-N encoding (Boolean Function)
  - 1-to-1 encoding (Angle Encoding)
  - N-to-*log*N encoding (Amplitude Encoding)

- **Variational Quantum Circuit**
  - Designed based on neural network, but no classical correspondence
  - Can integrate real-number weights in the circuit
  - Non-linearity is difficult to be integrated, leading a 1-layer neural network

Halfway Takeaway

- **Quantum-based Neural Network Accelerator**
  - Boolean function-based design
    - High flexibility
    - High cost **(no quantum advantage)**
  - Angle-based design
    - Work for specific functions
    - Neutral cost **(still hard to have quantum advantage)**
  - Amplitude-based design
    - More limitations
    - Lower cost **(potential of quantum advantage)**

Comparison of different quantum neurons.

Quantum neurons	Input features	Number of qubits	Gate complexity
(a)	Binary	$\boxed{pn} + 2m$	$O(pn + m2^m)$
(b)	Continuous	$\log n + 2m + 1$	$O(\boxed{mn^2} + m2^m)$
(c)	Binary	$\boxed{2pn} + m$	$O(m2^{2pn} + m^2)$

Input features	Number of Qubits	Number of Gates
Linear	$O(n)$	$O(n)$
Non-linear	$O(1)$	$O(m \cdot n)$

Data Encoding

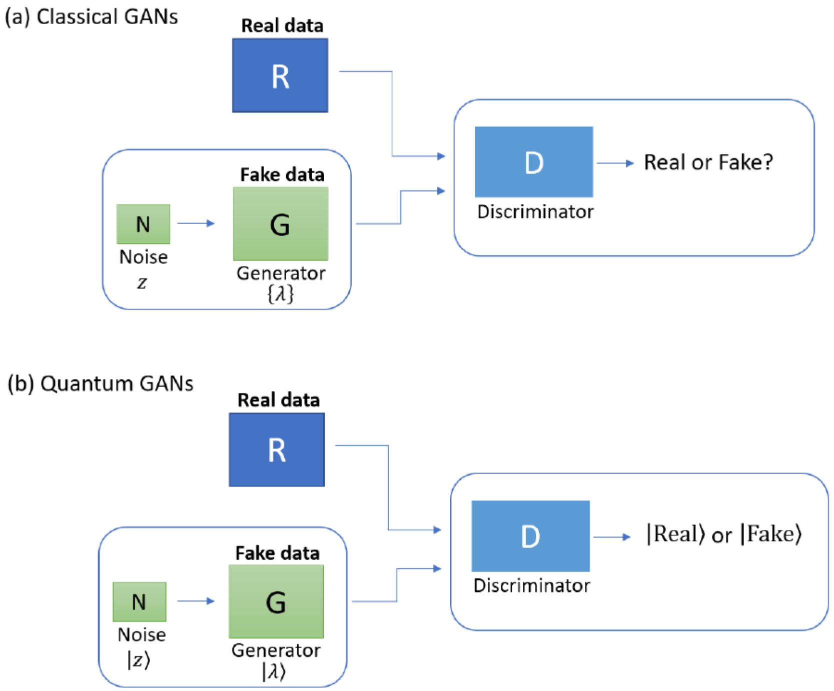
Weight Embedding

Input features	Number of Qubits	Number of Gates
U(x)	$O(logN)$	$\mathbf{O}(?)$
U(W)	$O(logN)$	$\mathbf{O}(?)$
U(N)	$O(1)$	$O(logN)$

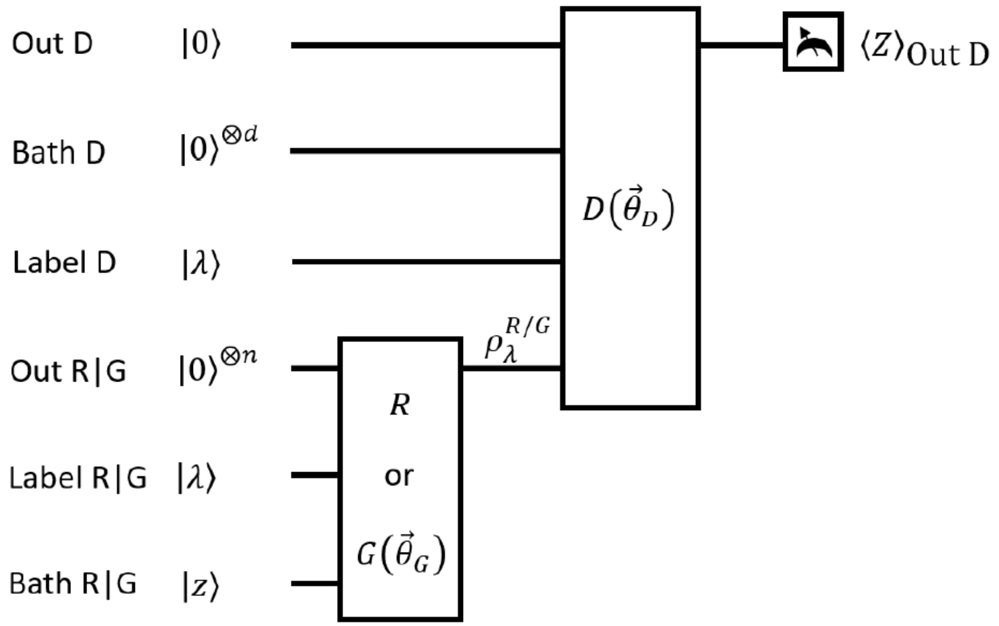
Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
    - QGAN
    - QRNN
    - QLSTM
- **Call for paper at “Electronics”**
- **Q&A**

QGAN'2018: Framework



QuGANs:



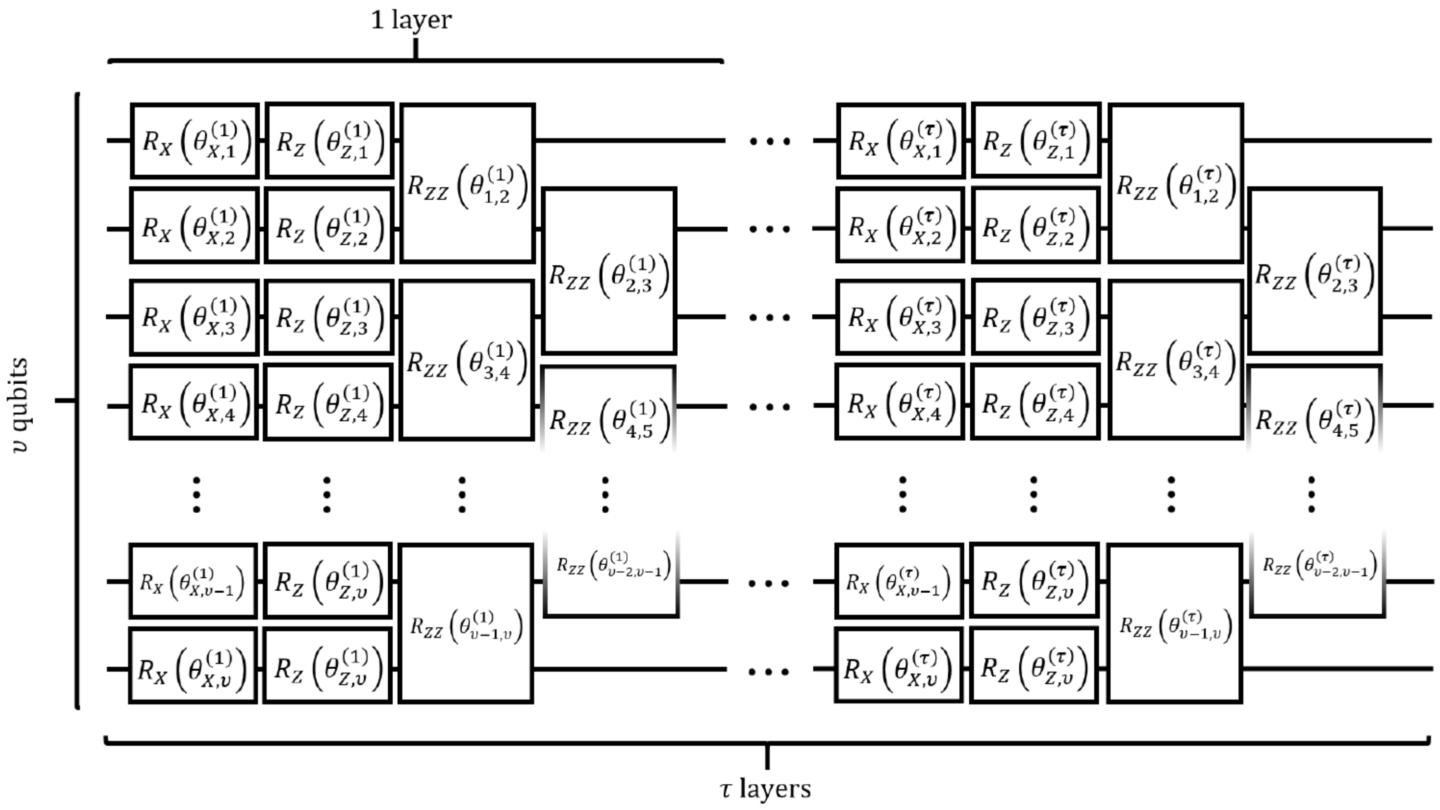
(a): A **discriminator** must determine **whether** the samples it is given are produced by **a real source R** or a generator **G(z)** **equipped with a source of noise z**.

(b): discriminator -> **quantum discriminator**
G(z) -> **G(**$|Z\rangle$)
output(Real or Fake) -> output($|\boldsymbol{real}\rangle$ **or** $|\boldsymbol{fake}\rangle$)

- **R** or the parametrized generator **G(**$\overrightarrow{\boldsymbol{\theta}_G}$**)** is applied on an initial state $|\boldsymbol{0}, \boldsymbol{\lambda}, \boldsymbol{z}\rangle$ defined on the **Label R|G, Out R|G and Bath R|G**.
- The discriminator D($\overrightarrow{\boldsymbol{\theta}_D}$) uses the information $\boldsymbol{\rho}_{\boldsymbol{\lambda}}^{\boldsymbol{R/G}}$ and an initial resource state $|\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{\lambda}\rangle$ defined on the **Out D** ,**Bath D** and **Label D** registers.
- D outputs its answer $|\boldsymbol{real}\rangle$ **or** $|\boldsymbol{fake}\rangle$ in the **Out D** register.
- The expectation value $\langle \boldsymbol{Z}\rangle_{\boldsymbol{Out\ D}}$ is proportional to the probability that D outputs $|\boldsymbol{real}\rangle$ **.**

[ref] Pierre-Luc, et al.2018.Quantum generative adversarial networks . PHYSICAL REVIEW A 98,012324

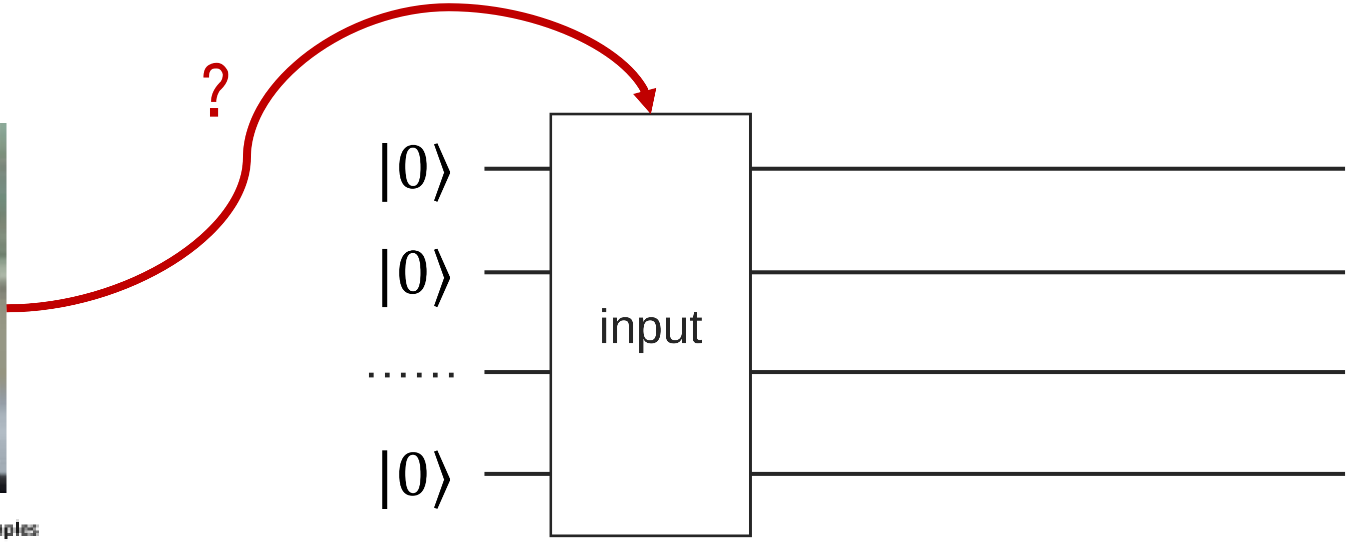
QGAN’2018: Applied VQC

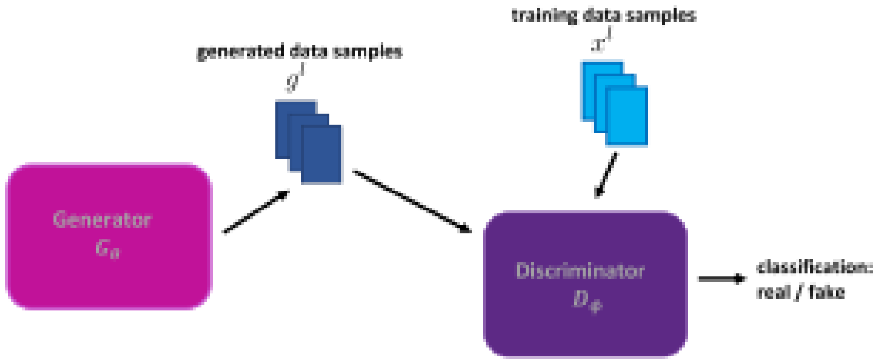


[ref] Pierre-Luc, et al.2018.Quantum generative adversarial networks . PHYSICAL REVIEW A 98,012324

An Application of QGAN’2019

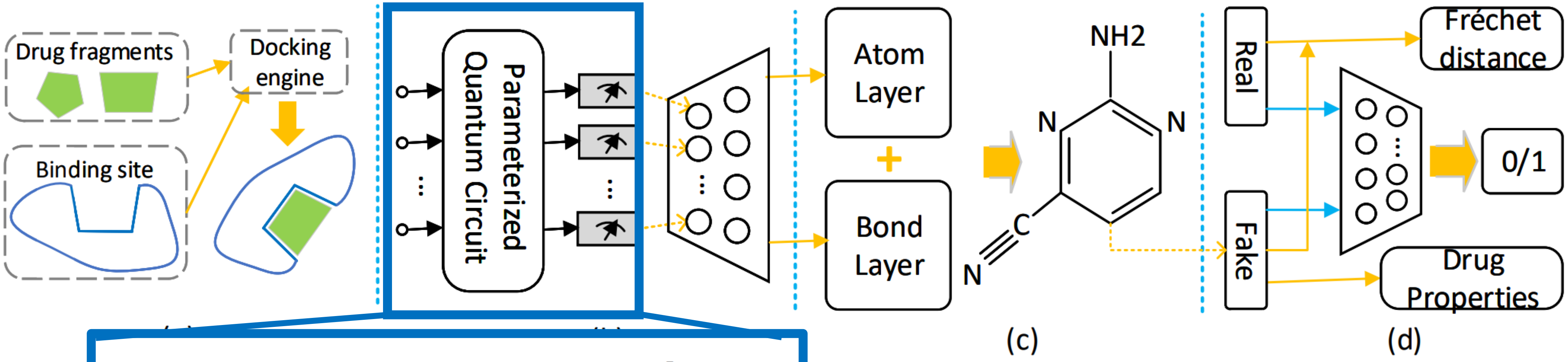
- Quantum State Preparation



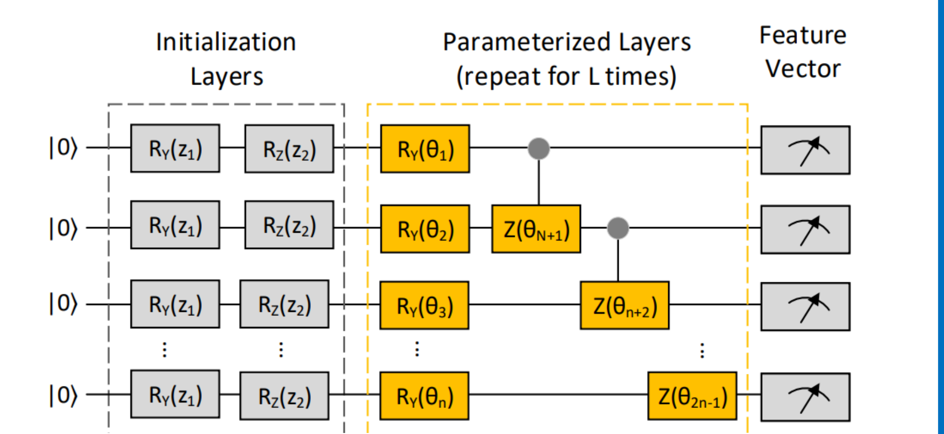


[ref] Christa Zoufal , et al. , Quantum Generative Adversarial Networks for learning and loading random distributions. npj| Quantum Information

Another Application of QGAN'2021



(a) Only ge… s are considered as valid;
(b) (b) qua… easuring the expectation values) and classical stage (neural network with last-layer out-fea…
(c) applica… s (one example synthetic molecule is given);
(d) a batch… synthetic molecules generated from (c) are fed into classical discriminator for real/syn… ic molecules are evaluated using RDKit package. The prediction losses from discrim… it for updating all parameters simultaneously in each training epoch.



- Parameterized quantum circuit to obtain feature vector of N dimensions.
- The circuit is composed of initialization layers, repeatable parameterized layers and measurement layer .
- Two CNOT gates for each ZZ interaction for creating entanglement are not shown here.

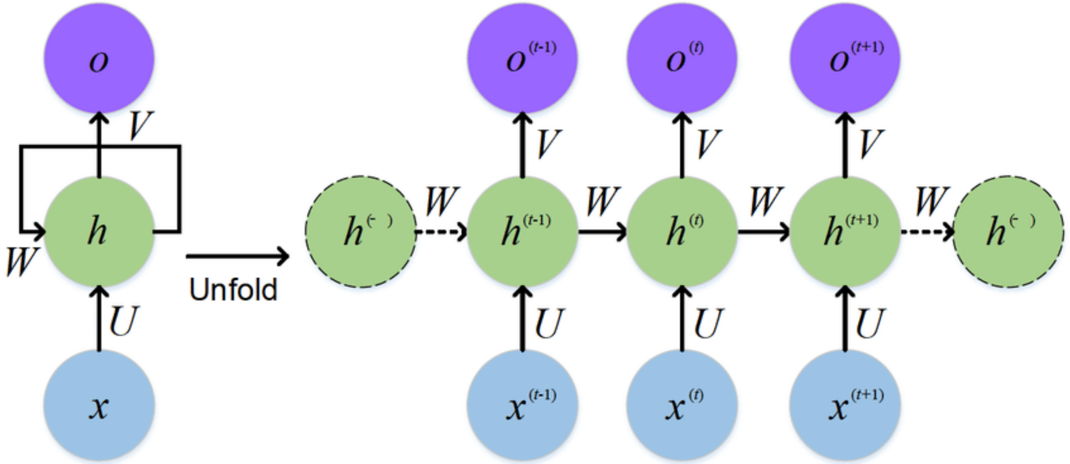
[ref] Junde Li, et al. , Quantum Generative Models for Small Molecule Drug Discovery. arXiv@2021

Agenda – Session 1: Introduction

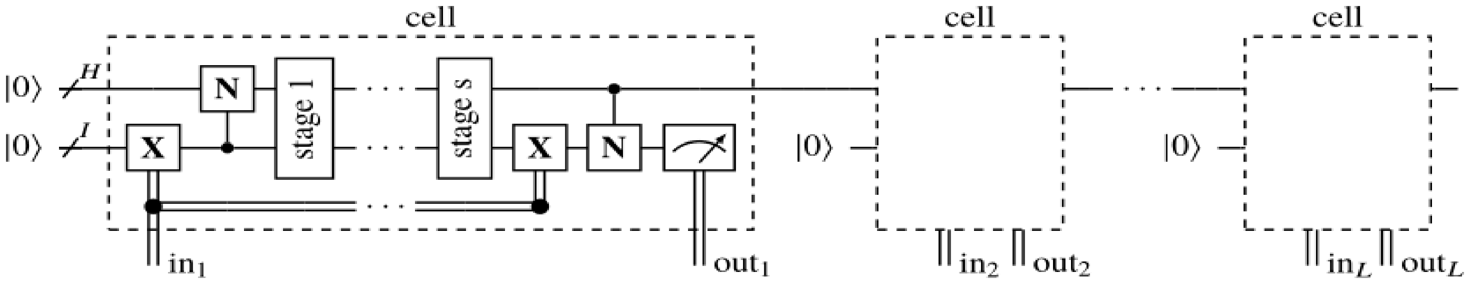
- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
    - QGAN
    - QRNN
    - QLSTM
- **Call for paper at "Electronics"**
- **Q&A**

QRNN --- Based on Quantum Neural (1-to-1)

Classical RNN



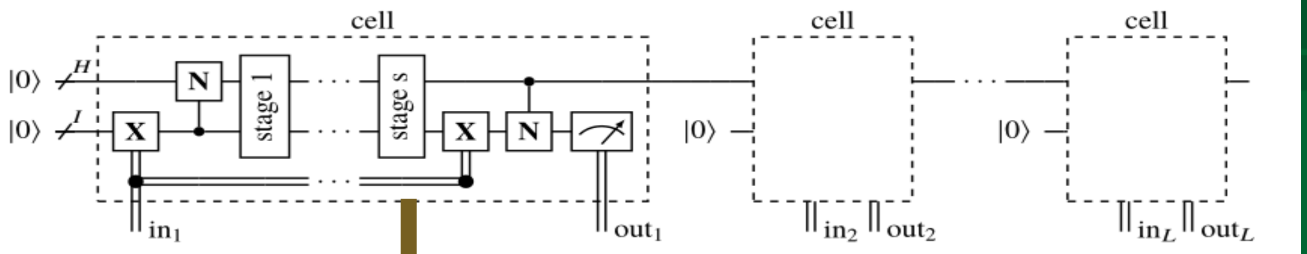
Quantum RNN



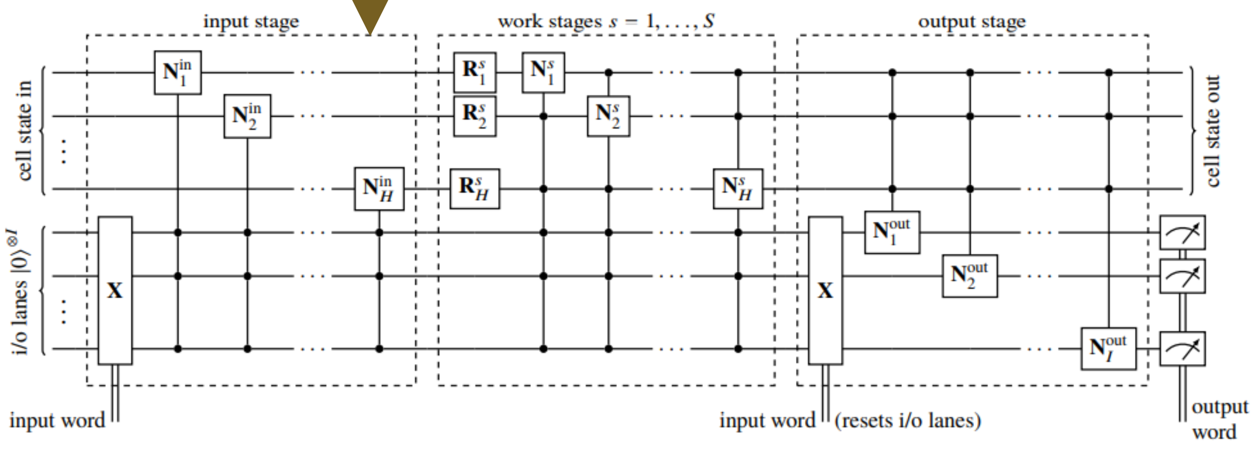
[ref] Johannes Bausch, et al. , Recurrent Quantum Neural Networks Johannes . arXiv @ Jun. 2020

QRNN

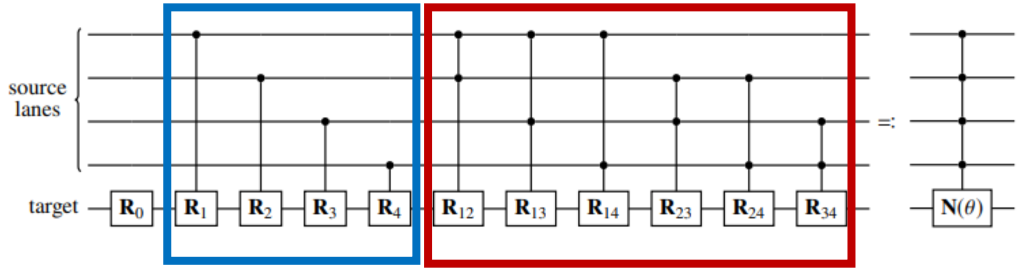
QRNN



QRNN Cell



Quantum Neuron (1-to-1)



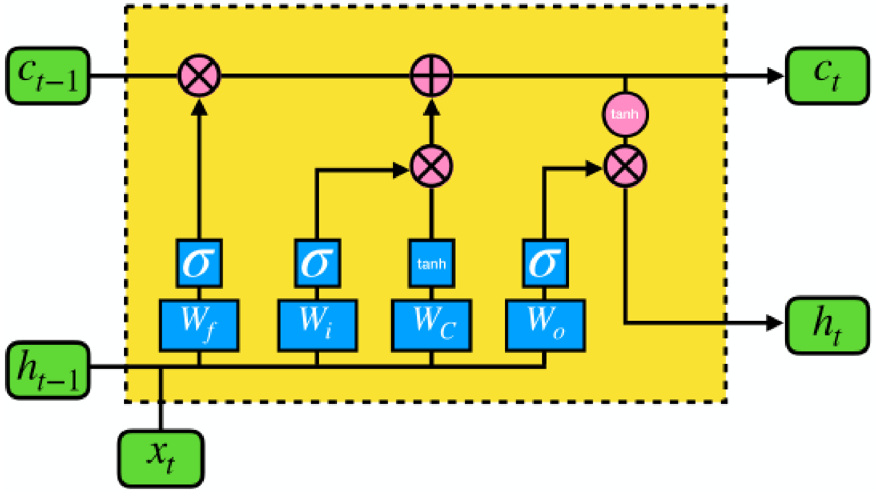
$$\eta' = \theta_0 + \boxed{\sum_{i=1}^{n} \theta_i x_i} + \boxed{\sum_{i=1}^{n} \sum_{j=1}^{n} \theta_{ij} x_i x_j} + \ldots = \sum_{\substack{I \subseteq [n] \\ |I| \leq d}} \theta_I \prod_{i \in I} x_i,$$

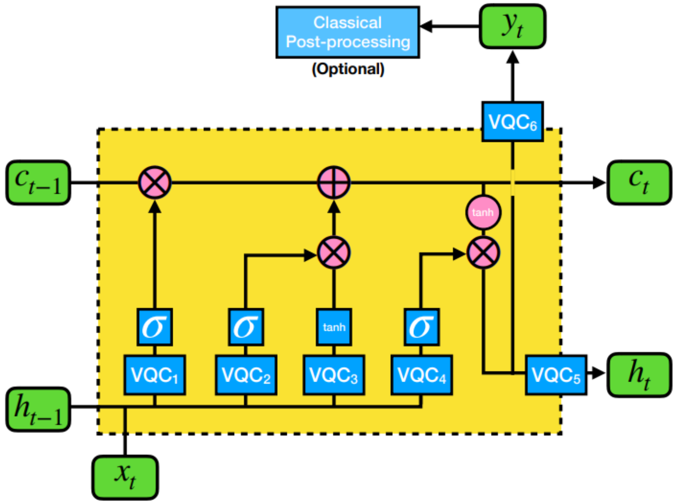
[ref] Johannes Bausch, et al. , Recurrent Quantum Neural Networks Johannes . arXiv @ Jun. 2020

Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
    - QGAN
    - QRNN
    - QLSTM
- **Call for paper at “Electronics”**
- **Q&A**

LSTM v.s. QLSTM -- Based on VQC





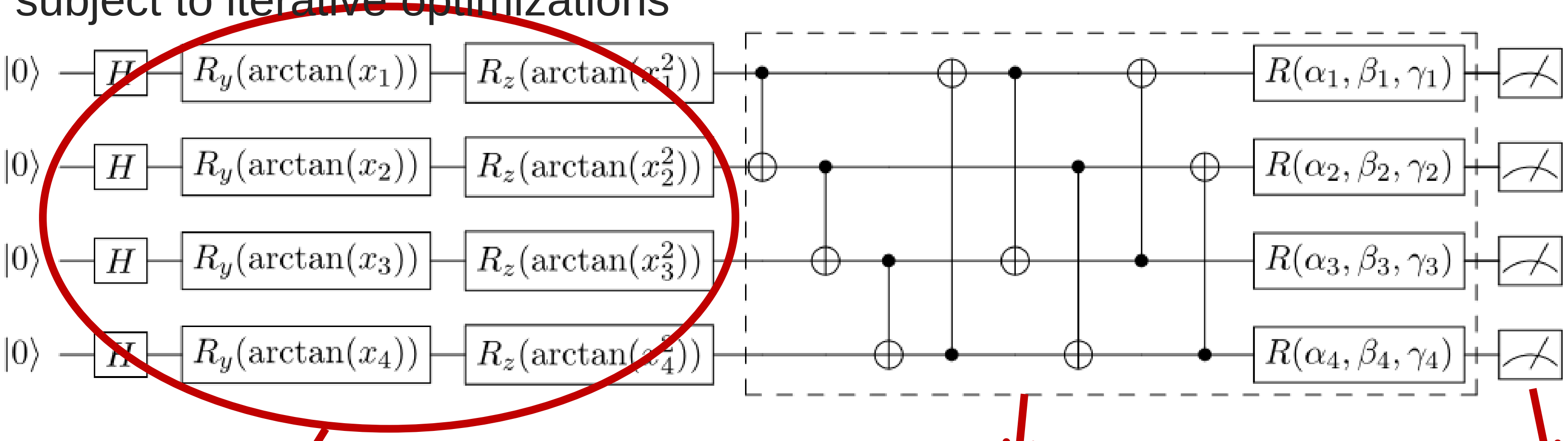
$$f_t = \sigma\left(W_f \cdot v_t + b_f\right)$$
$$i_t = \sigma(W_i \cdot v_t + b_i)$$
$$\tilde{C}_t = \tan h(W_C \cdot v_t + b_C)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot v_t + b_o)$$
$$h_t = o_t * \tan h(c_t)$$

$$f_t = \sigma\left(VQC_1(v_t)\right)$$
$$i_t = \sigma\left(VQC_2(v_t)\right)$$
$$\tilde{C}_t = \tan h\left(VQC_3(v_t)\right)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(VQC_4(v_t)\right)$$
$$h_t = VQC_5(o_t * \tan h(c_t))$$
$$y_t = VQC_6(o_t * \tan h(c_t))$$

[ref] Samuel Yen-Chi, et al. , Quantum Long Short-Term Memory.

VQC Used in Q-LSTM

VQCs are a kind of quantum circuits that have tunable parameters subject to iterative optimizations



Data Encoding Layer: classical input vector encoded into quantum state.

Variational Layer: Generate multi quantum entanglement.

Quantum Measurement Layer：Consider the expectation values of every qubit by measuring in the computational basis

[ref] Samuel Yen-Chi, et al. , Quantum Long Short-Term Memory.

Agenda – Session 1: Introduction

- Roadmap of Quantum Machine Learning

Special Issue "Quantum Machine Learning: Theory, Methods and Applications"

IMPACT FACTOR 2.397

CITESCORE 2.7 SCOPUS

- Special Issue Editors
- Special Issue Information
- Keywords
- Published Papers

A special issue of *Electronics* (ISSN 2079-9292). This special issue belongs to the section "Quantum Electronics".

Deadline for manuscript submissions: **20 April 2022**.

Special Issue Editors

Dr. Weiwen Jiang E-Mail Website
Guest Editor
Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
Interests: Edge Computing; Quantum Computing

Dr. Ying Mao E-Mail Website
Guest Editor
Department of Computer and Information Science, Fordham University, New York, NY 10458, USA
Interests: Quantum Machine Learning; Quantum Networks; Quantum Cloud

Dr. Samuel Yen-Chi Chen E-Mail Website
Guest Editor
Computational Science Initiative, Brookhaven National Laboratory, New York, NY 11973-5000, USA
Interests: Quantum computing; Quantum machine learning; Quantum optimal control; Quantum error correction

- Call for paper at "Electronics"
- Conclusion
- Q&A

electronics
An open access journal by MDPI

Special Issue:

Quantum Machine Learning Theory, Methods and Applications

Guest Editor:

Dr. Weiwen Jiang
Department of Electrical and Computer Engineering, George Mason University, USA

Dr. Ying Mao
Department of Computer and Information Science, Fordham University, New York, USA

Dr. Samuel Yen-Chi Chen
Computational Science Initiative, Brookhaven National Laboratory, New York, USA

Deadline for manuscript submissions:
20 April 2022

Topics are welcome to contribute:

- Quantum machine learning
- Quantum neural network
- Quantum supervised learning
- Quantum unsupervised learning
- Quantum reinforcement learning
- Quantum learning theory
- Variational quantum circuits
- Noisy intermediate-scale quantum devices (NISQ)

https://www.mdpi.com/journal/electronics/special_issues/quantum_machine_learning

Agenda – Session 1: Introduction

- **Roadmap of Quantum Machine Learning**
  - Data Encoding
  - HHL Algorithm
  - Variational Quantum Circuit
  - Quantum-based Neural Network Accelerator
  - Applications
- **Call for paper at “Electronics”**
- **Conclusion**
- **Q&A**

Takeaway

- Quantum Computing
  - # of qubits grows rapidly
  - Q-Circuit design is similar to classical ones, using quantum gates
- Machine Learning meets Quantum Computing
  - Potential to solve computation-bound / memory-wall in classical
  - What is quantum neural network? VQC v.s. Q-Based Accelerator
- What is the fair metric for comparison?
  - Time-space complexity
- What we want to achieve?
  - Quantum advantage for real-world applications in near-term Q!

Q&A

https://github.com/JQub/QuantumFlow_Tutorial (Source Code of All Hands-On in Tutorial)

https://github.com/JQub/qfnn (Source Code of QFNN API & Place to post Issues)

https://pypi.org/project/qfnn/ (Package of QFNN on PYPI)

https://libraries.io/pypi/qfnn/ (QFNN on Libraries.io)

https://jqub.ece.gmu.edu (JQub Website)

https://jqub.ece.gmu.edu/categories/QF (QuantumFlow Website for news and slides)

https://jqub.ece.gmu.edu/categories/QF/qfnn/ (QFNN Documents)

https://www.nature.com/articles/s41467-020-20729-5 (QuantumFlow Paper)

https://arxiv.org/pdf/2012.10360.pdf (Paper on How to Correct Map NN to Q)

https://arxiv.org/pdf/2109.03806.pdf (QF-Mixer)

https://arxiv.org/pdf/2109.03430.pdf (QF-RobustNN) wjiang8@gmu.edu

George Mason University

4400 University Drive
Fairfax, Virginia 22030

Tel: (703)993-1000