Quantum Algorithm

Euntaek Lee

Seoul National University

April 22, 2022

General considerations for quantum algorithm

- How can we embed data to quantum computer?
- How can we design the algorithm?
- How much is the cost of given algorithm?

How can we embed data to quantum computer?

Strategies of Input encoding

1. Basis encoding
2. Amplitude encoding
3. Tensor product encoding
4. Hamiltonian encoding

Basis encoding

Bit string		Basis state
(0011)	$\rightarrow$	$\|0011\rangle$
1 bit	$\rightarrow$	1 qubit

It is possible, but consider the price.

Price of 1 qubit

This 2-qubit quantum computer is $5000.

Amplitude encoding

Classical data		Quantum state
$\boldsymbol{x} = (x_i)_{0 \leq i \leq 2^n - 1}$	$\rightarrow$	$\lvert\psi_x\rangle = \sum_j x_j \lvert j\rangle$
$2^n \times$size(double)	$\rightarrow$	n qubit

For given arbitrary vector $\boldsymbol{x}$, we need to generate a state

$$\lvert\psi_x\rangle = \sum_j x_j \lvert j\rangle .$$

But best known algorithm can do it with time complexity $O\left(\frac{1}{n}2^n\right)$.

Tensor product encoding

Classical data		Quantum state
$\boldsymbol{x} = (x_i)_{0 \leq i \leq 2^n - 1}$	$\rightarrow$	$\bigotimes_{i=1}^{n} R_y(2x_i)\|0\rangle$ $= \bigotimes_{i=1}^{n} (\cos x_1 \|0\rangle + \sin x_1 \|1\rangle)$
$2^n \times$size(double)	$\rightarrow$	n qubit

Where

$$R_y(\theta) = \exp\left(-i\frac{\theta}{2}\sigma_y\right) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}.$$

We need to prepare only n gates.

Hamiltonian encoding

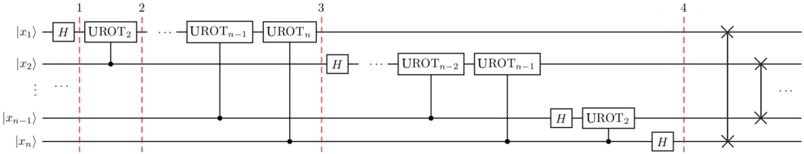
Classical data		Operator
$A = (a_{ij})_{0\leq i,j\leq n}$	$\rightarrow$	$\begin{pmatrix} O & A \\ A^\dagger & O \end{pmatrix}$
$n^2\times$size(double)	$\rightarrow$	1 operator

How much is the cost of given algorithm?

Complexity class

- EQP : A class of problems that can be solved exactly by quantum computers using polynomial time.
- BQP : Problems that can be solved by quantum computers using polynomial time with error probability $\leq 1/3$ on every input.
- QPSPACE : Problems that can be solved by quantum computers using polynomial order of spaces.

Measuring time and space complexity



How can we design the algorithm?

Discrete Fourier transform

- Input : $\boldsymbol{x} = (x_k)_{k=0}^{N-1} \in \mathbb{C}^N$
- Output : $X = (X_k)_{k=0}^{N-1}$ defined by

$$X_k = \sum_{l=0}^{N-1} x_l \exp\left(i \frac{2\pi}{N} kl\right)$$

- Time complexity : $N \log N$
- Space complexity : $N \log N$

Quantum Fourier transform

- Input : n-qubit $\boldsymbol{x} = (x_k)_{k=0}^{2^n-1} = \sum_{j=0}^{2^n-1} x_j \left|j\right\rangle$
- Output : $X = (X_k)_{k=0}^{2^n-1}$ defined by

$$X_k = \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} x_j \exp\left(i\frac{2\pi}{N}jk\right)$$

- Complexity : width $O(n)$, depth $O(n^2)$

Quantum Fourier transform

Algorithm(Modified)

- Input : $|j\rangle = |x_1 x_2 \cdots x_n\rangle$.
- Wanted output :

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2\pi ijk} \exp\left(\frac{2\pi i}{2^n} jk\right) |k\rangle = \frac{1}{2^{n/2}} \bigotimes_{k=1}^{n} \left(|0\rangle + \exp\left(\frac{2\pi i}{2^k} j\right) |1\rangle\right)$$

- Complexity : depth $O(n^2)$, width $O(n)$

Quantum Fourier transform

Algorithm representation

$$|x_1 x_2 \cdots x_n\rangle \xrightarrow{H_1} \frac{1}{2^{1/2}}\left(|0\rangle + \exp\left(\frac{2\pi i}{2}x_1\right)|1\rangle\right)|x_2 \cdots x_n\rangle$$

$$\xrightarrow{\text{UROT}_2} \frac{1}{2^{1/2}}\left(|0\rangle + \exp\left(\frac{2\pi i}{2^2}x_2 + \frac{2\pi i}{2}x_1\right)|1\rangle\right)|x_2 \cdots x_n\rangle$$

$$\cdots$$

$$\xrightarrow{\text{UROT}_n} \frac{1}{2^{1/2}}\left(|0\rangle + \exp\left(\frac{2\pi i}{2^n}x\right)|1\rangle\right)|x_2 \cdots x_n\rangle .$$

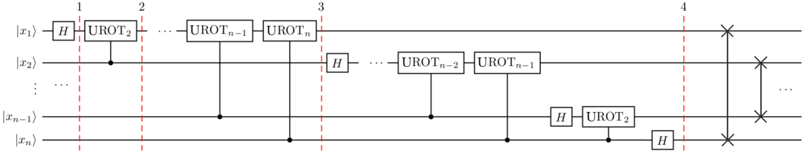
$$\cdots$$

$$\longrightarrow \frac{1}{2^{n/2}}\bigotimes_{k=0}^{n-1}\left(|0\rangle + \exp\left(\frac{2\pi i}{2^{n-k}}x\right)|1\rangle\right)$$

$$\xrightarrow{\text{Rev}} \frac{1}{2^{n/2}}\bigotimes_{k=1}^{n}\left(|0\rangle + \exp\left(\frac{2\pi i}{2^k}j\right)|1\rangle\right)$$

Quantum Fourier transform

Circuit



Quantum Fourier transform

Notes

- We can see width is $O(n)$, depth is $O(n^2)$
- To do inverse QFT, Apply

$$\mathrm{UROT}_k^{-1} = \mathrm{diag}\left(1, \exp\left(-\frac{2\pi i}{2^k}\right)\right)$$

instead of UROT.

Quantum phase estimation

- Input : Given unitary operator U with eigenvalue equation

$$U\left|\phi\right\rangle = e^{2\pi i\varphi}\left|\phi\right\rangle$$

- Output : An estimate of $\varphi \in [0,1)$.
- Complexity : depth $O\left(n^2\right)$, width $O(n)$

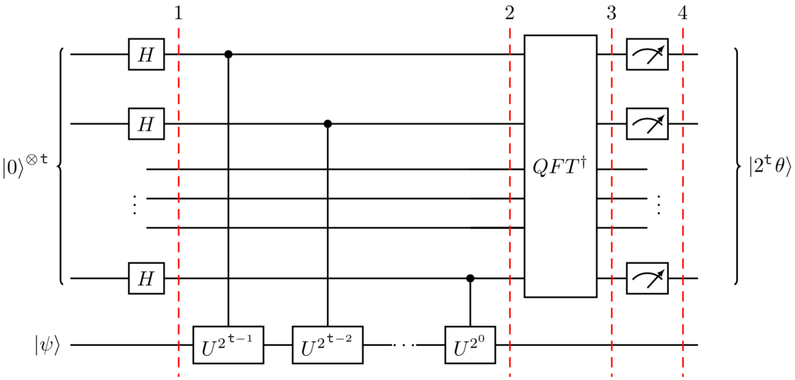
Quantum phase estimation

Algorithm representation

$$
\begin{aligned}
|00\cdots 0\rangle\,|\phi\rangle &\xrightarrow{H^{\otimes n}} \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}|k\rangle\,|\phi\rangle \\
&\xrightarrow{\text{CU}} \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}|k\rangle\,U^k\,|\phi\rangle = \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}e^{2\pi ik\varphi}\,|k\rangle\,|\phi\rangle \\
&\xrightarrow{\text{QFT}^{-1}} \frac{1}{2^n}\sum_{k,l}\exp\left(2\pi ik\left(\varphi-\frac{l}{2^n}\right)\right)|l\rangle\,|\phi\rangle\,. \\
&\xrightarrow{\text{Measure}} P\,(\text{observe nearest integer a to } 2^n\varphi) \geq \frac{4}{\pi^2}
\end{aligned}
$$

Quantum phase estimation

Circuit



Quantum phase estimation

Doing Inverse QFT

$$\frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1} e^{2\pi i k\varphi} |k\rangle |\phi\rangle$$

$$\xrightarrow{\text{QFT}^{-1}} \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1} e^{2\pi i k\varphi} \left(\frac{1}{2^{n/2}} \sum_{l=0}^{2^n-1} e^{-2\pi i k l/2^n} |l\rangle \right) |\phi\rangle$$

$$= \frac{1}{2^n}\sum_{k,l=0}^{2^n-1} e^{2\pi i k(\varphi - l/2^n)} |l\rangle |\phi\rangle$$

Quantum phase estimation

Probability estimation

$a :=$An integer with $2^n\delta := |2^n\varphi - a| \leq \frac{1}{2}$.
In state vector $\frac{1}{2^n}\sum_{k,l=0}^{2^n-1} e^{2\pi ik(\varphi - l/2^n)} |l\rangle$,

$$P\left(\text{observe } a\right) = \left|\frac{1}{2^n}\sum_{k=0}^{2^n-1} e^{2\pi ik\delta}\right|^2 = \frac{1}{2^{2n}}\left|\frac{1-e^{2\pi i2^n\delta}}{1-e^{2\pi i\delta}}\right|^2 = \left|\frac{\sin\left(2^n\pi\delta\right)}{2^n\sin\left(\pi\delta\right)}\right|^2$$

$$\geq \left|\frac{\sin\left(2^n\pi\delta\right)}{2^n\pi\delta}\right|^2$$

$$\geq \left(\frac{2}{\pi}\right)^2.$$

References

- Schuld, M. & Petruccione, F.
  Machine Learning with Quantum Computers, Second Edition.
  Springer, 2021.
- Ganguly, S.
  Quantum Machine Learning:An Applied Approach.
  Apress, 2021.
- The Jupyter Book Community (2021).
  Quantum Fourier Transform, Qiskit.
  https://qiskit.org/textbook/ch-algorithms/quantum-fourier-transform.html
- The Jupyter Book Community (2021).
  Quantum Phase Estimation, Qiskit.
  https://qiskit.org/textbook/ch-algorithms/quantum-phase-estimation.html