Chapter 9
QML: Way Forward

Woojoo Shim

Korea Institute for Advanced Study

June 10, 2022

Table of Contents

QML: Way Forward

- Quantum machine learning (QML) is a **cross-disciplinary subject** made up of two of the most exciting research areas: quantum computing and classical machine learning.

Q: Classical-challenging, Quantum-easy problem?

- Microsoft, Google, IBM, Rigetti, Xanadu, D-Wave, IonQ, Intel,...

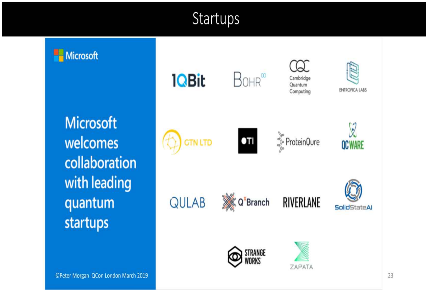


Figure: https://www.infoq.com/presentations/quantum-simulate-chemistry/

Quantum Computing for Chemistry

- Computational chemistry would be one of the first domains to significantly benefit from the development of quantum devices.

– calculate accurate microscopic properties (e.g. energies, forces, and electrostatic multipoles of specific configurations),

– efficient sampling of potential energy surfaces (to obtain corresponding macroscopic properties)

- Hartree-Fock method: mean-field approximation of many-electron system.

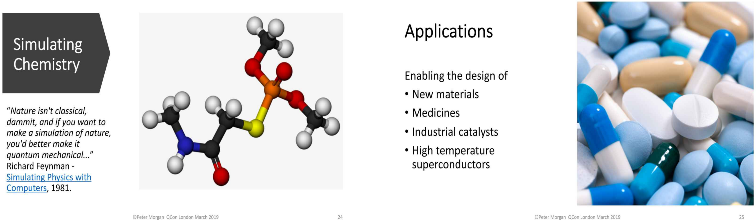


Figure: https://www.infoq.com/presentations/quantum-simulate-chemistry/

OpenFermion

- OpenFermion is an open source library written largely in Python for compiling and analyzing quantum algorithms to simulate bosonic/fermionic systems, including quantum chemistry. Among other functionalities, this version features data structures and tools for obtaining and manipulating representations of bosonic/fermionic and qubit Hamiltonians.

https://github.com/quantumlib/OpenFermion

Hamiltonian Simulation (Quantum Simulation)

Goal (Hamiltonian Simulation)

Approximate $U(t) = e^{-\mathrm{i}Ht}$, where H is known but $U(t)$ is not explicitly known.

— analog simulation: find quantum systems that "naturally" simulate Hamiltonians.

— digital simulation: decompose the time evolution into quantum gates.

Trotter Suzuki simulations

If the time independent hamiltonain H is given by a sum of elementary Hamiltonians H_j, we have

$$e^{-\mathrm{i}Ht} = \left(\prod_j e^{-\mathrm{i}H_j\frac{t}{n}}\right)^n + O\left(\sum_{j,k} n\left\|[\frac{t}{n}H_j, \frac{t}{n}H_k]\right\|\right)$$

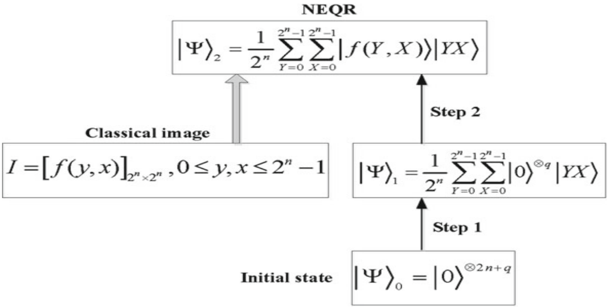
Quantum Image Processing (QIMP)

- Representations of the image on quantum computers: qubit lattice, Real Ket, grid qubit, quantum lattice, FRQI, MCQI, NEQR...

1. FRQI: encode colors(grayscale) throught angles

$$|I\rangle = \frac{1}{2^n} \sum_{j=0}^{2^{2n}-1} |q\rangle_j \otimes |j\rangle, \quad |q\rangle_j = \cos\theta_j|0\rangle + \sin\theta_j|1\rangle.$$

2. MCQI: use three channels to represent RGB.
3. NEQR: gray range $0 \leq f(y,x) \leq 2^q - 1$.

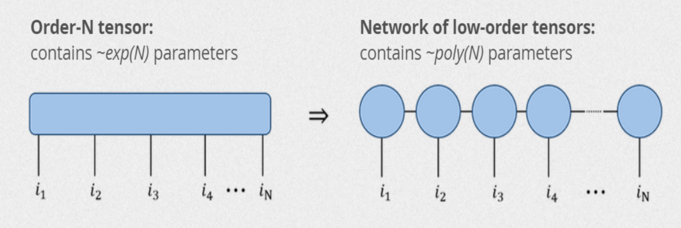


Tensor Networks

- Tensor network: factorization of huge tensor into contracted product of smaller tensors

– exponential reduction in memory needed
– exponential speedup of computations

...



- Google, Company X and Perimeter Institute for Theoretical Physics created the TensorNetwork library

https://www.youtube.com/watch?v=rPZFJSt3dIg

Markov chain and classical random walks

• (Discrete-time) Markov chain: sequence of random variables $X_1, \cdots, X_N, \cdots$ with the Markov property

$$\mathbb{P}(X_{n+1} = x | X_1 = x_1, \cdots, X_n = x_n) = \mathbb{P}(X_{n+1} = x | X_n = x_n).$$

• Time-homogeneous Markov chain:

$$\mathbb{P}(X_{n+1} = j | X_n = i) = \mathbb{P}(X_n = j | X_{n-1} = i) =: m_{ij}, \quad \sum_j m_{ij} = 1.$$

• Probability distribution of a location of walker at time t over a graph G can be described by a stochastic (row) vector $\mathbf{p}(t)$, where

$$\mathbf{p}(t+1) - \mathbf{p}(t) = \mathbf{p}(t)(M - I)$$

• Random walk: G connected, symmetric, no loop, each m_{ij} is either 0 or $\frac{1}{d_i}$.
- in fact, d_i becomes independent of i.

Markov chain and classical random walks

- Continuous-time analogue: denote $M - I = \frac{L}{d}$ and obtain

$$\dot{\mathbf{p}}(t) = \frac{1}{d}\mathbf{p}(t)L, \quad L_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & \text{otherwise} \\ -d & i = j \end{cases},$$

and by considering a proper time-rescaling, we have

$$\dot{\mathbf{p}}(t) = \mathbf{p}(t)L \iff \frac{dp_j(t)}{dt} = \sum_{i:(i,j)\in E} p_i(t)L_{ij}, \quad \forall\ j.$$

- If $G = \mathbb{Z}^k$ with each adjacent points in lattice are connected, then $L \sim \nabla^2$ and $\mathbf{p}$ becomes Gaussian.

Quantum walk

- Quantum walk: Schrödinger equation expressed in Laplacian L,

$$\mathrm{i}\frac{d\psi_j(t)}{dt} = \sum_{i:(i,j)\in E} \psi_i(t) L_{ij}$$

- Eigenstates of the Laplacian operator L: for every $p \in [-\pi, \pi]$,

$$\begin{aligned} |p\rangle &= \sum_x e^{ipx}|x\rangle, \\ L|p\rangle &= \sum_x e^{ipx}|x\rangle + 1 + e^{ipx}|x\rangle - 1 - 2e^{ipx}|x\rangle \\ &= \sum_x (e^{ip} + e^{-ip} - 2)e^{ipx}|x\rangle \\ &= 2(\cos p - 1)|p\rangle. \end{aligned}$$

Quantum walk

- Hence the probability distribution $p(x,t)$ at time t, with initial $|\psi(0)\rangle = |0\rangle$ is given by:

$$\begin{aligned}
|\langle x|e^{-\mathrm{i}Lt}|0\rangle|^2 &= \left|\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-2\mathrm{i}t(\cos p-1)}\langle x|p\rangle\, dp\right|^2 \\
&= \left|\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-2\mathrm{i}t(\cos p-1)} e^{\mathrm{i}px}\, dp\right| \\
&= \left|\frac{1}{2\pi\mathrm{i}^x}\int_{-\pi}^{\pi} e^{\mathrm{i}(xp-2t\sin p)}\, dp\right|^2 \\
&= |J_x(2t)|^2.
\end{aligned}$$

Ensembles: AdaBoost

- Suppose there are 100 gamblers whose winning probabilities are slightly better than random guess.
- Now, the question is " is it possible to combine 100 predictions of the 100 experts to make a better prediction?"
- Surprisingly it is possible, which is called weak learnability.
- AdaBoost is the first algorithm to implement the idea of weak learnability.
- AdaBoost increases the weights for misclassified observations and decreases the weights for correctly classified observations.

Algorithm

1. Start with weights $w_i = \frac{1}{n}, i = 1, \cdots, n$.
2. Repeat for $m = 1, \cdots, M$;
    1. Fit the classifier $f_m(x) \in \{-1, 1\}$ using weights w_i
    2. Compute err_m by
    $$\mathrm{err}_m := \frac{\sum_{i=1}^{n} w_i I(y_i \neq f_m(x_i))}{\sum_{i=1}^{n} w_i}.$$
    3. Set $c_m := \log\left(\frac{1-\mathrm{err}_m}{\mathrm{err}_m}\right)$
    4. Update w_i by $w_i \exp(c_m I(y_i \neq f_m(x_i)))$.
3. Output the classifier $\mathrm{sign}(\sum_{m=1}^{M} c_m f_m(x))$

QBoost

QBoost was developed by researchers at Google and D-Wave labs before QML became popular. It tries to implement an ensemble of K binary classifiers $f_k(x),\ k = 1, \cdots, K$, that are combined by a weighted sum of the form

$$f(x) = \operatorname{sgn}\left(\sum_{k=1}^{K} w_k f_k(x)\right)$$

with $x \in \mathbb{R}^N$ and $f(x) \in \{-1, 1\}$. We choose weights w_k that minimize a least-squares loss function, and add a regularization term to prevent overfitting:

$$\operatorname{argmin}_w \left[\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K} w_k f_k(x_i) - y_i\right)^2 + \lambda \|w\|_0\right].$$

Therefore, the optimization reduces to

$$\sum_{k,k'=1}^{K} w_k w_{k'}\left(\sum_{i=1}^{M} f_k(x_i) f_{k'}(x_i)\right) + \sum_{k=1}^{K} w_k\left(\lambda - 2\sum_{i=1}^{M} f_k(x_i) y_i\right).$$

The coefficients serve as the interaction and field strengths of the Ising model.

감사합니다