Advanced QML algorithms for state preparation

Christa Zoufal
Quantum Applications Researcher

Qiskit

Motivation

State preparation

Loading data into ***quantum states*** can be difficult.

In the worst case, the data loading is ***exponentially*** expensive and thereby ***impairs*** potential advantage of algorithms such as Quantum Amplitude Estimation.

Approximative data loading schemes can be implemented by using a ***parameterized quantum circuit*** and machine learning techniques.

Quantum Generative Adversarial Networks

Qiskit

npj | Quantum Information

www.nature.com/npjqi

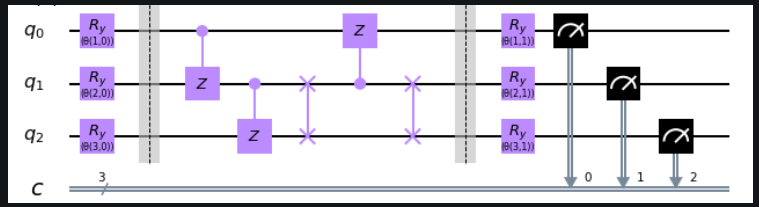
ARTICLE OPEN

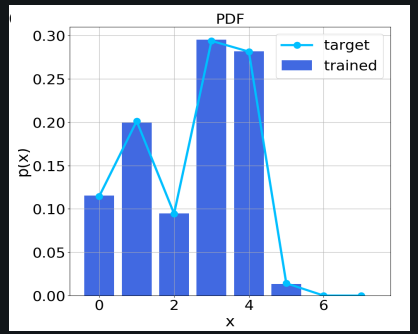
Quantum Generative Adversarial Networks for learning and loading random distributions

Christa Zoufal[1,2*], Aurélien Lucchi[2] and Stefan Woerner[1]

Idea:
Encode probability distribution into a parameterized quantum circuit

Implementation:
Use GAN to learn a probability distribution underlying given data samples

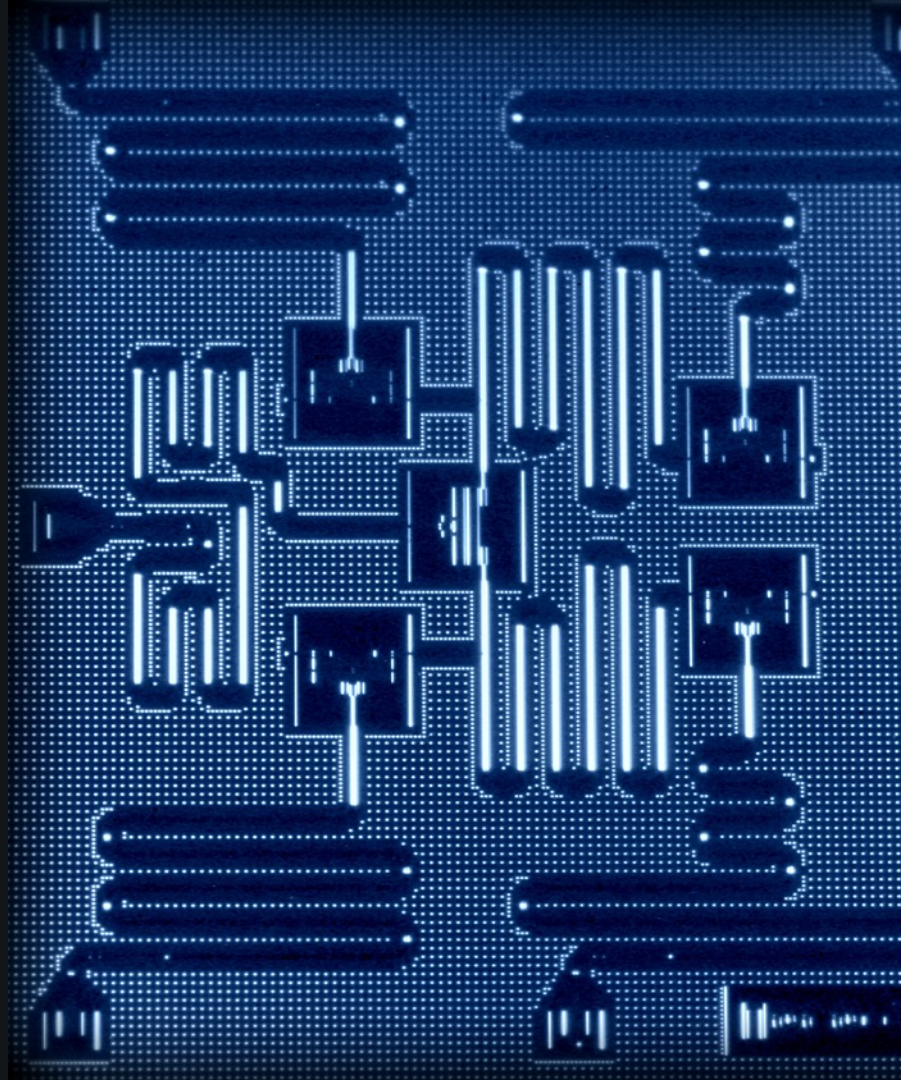




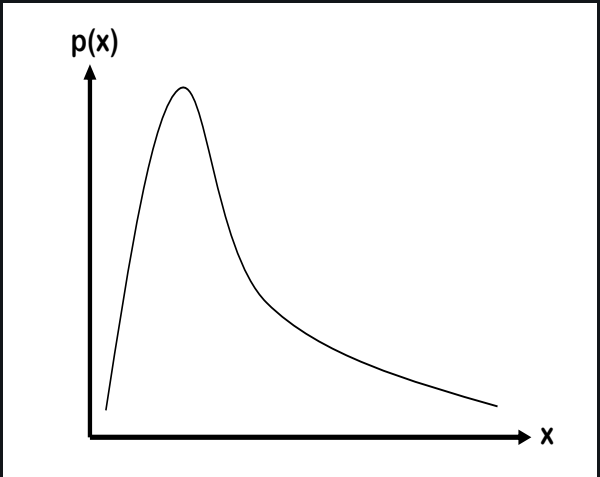
Christa Zoufal, ouf@zurich.ibm.com

Outlook

- What are (Quantum) Generative Adversarial Networks?
- How do we realize Quantum Generative Adversarial Networks?
- Numerical examples

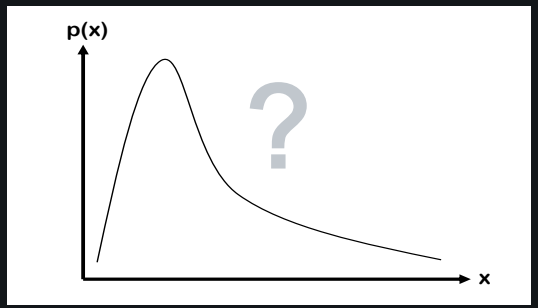


Distribution Loading



Christa Zoufal, ouf@zurich.ibm.com

Direct Approach



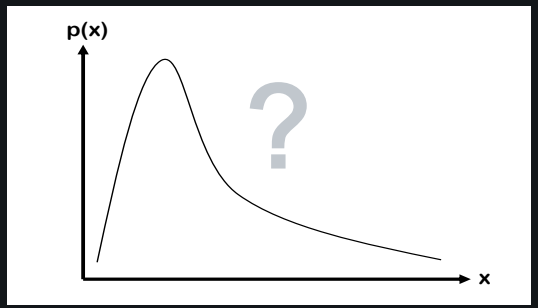
What about generic distributions?

→best known $O(2^n)$

Christa Zoufal, ouf@zurich.ibm.com

Data Loading with qGANs

Qiskit



Avoid exponential overhead

Christa Zoufal, ouf@zurich.ibm.com

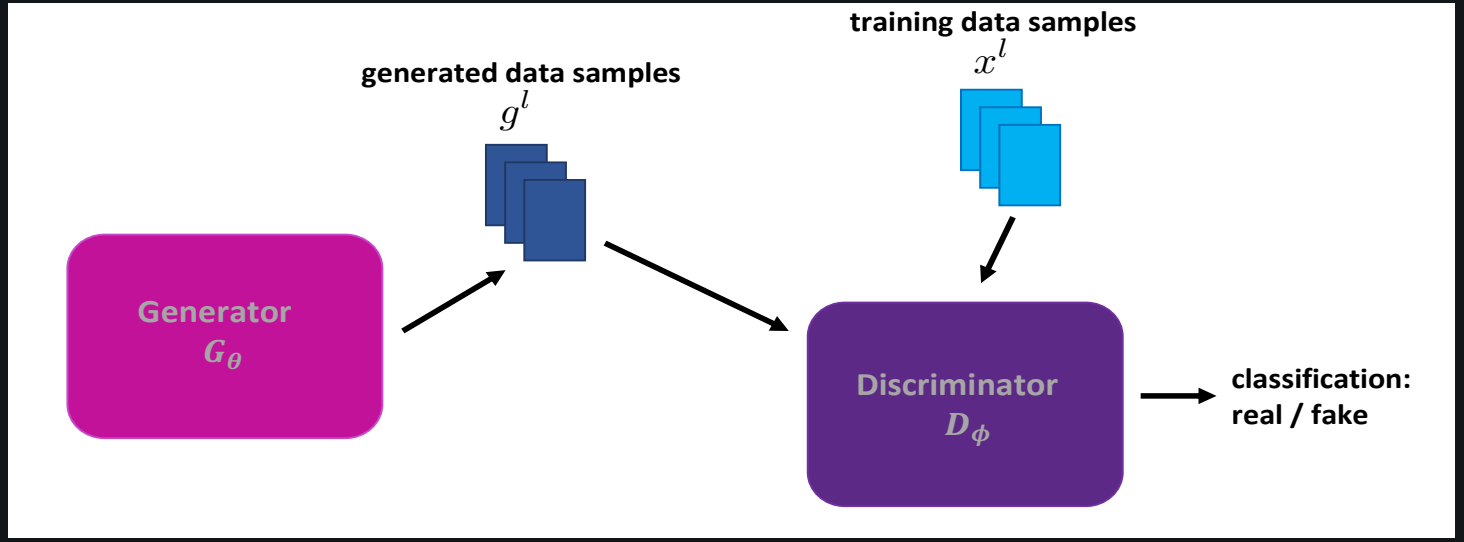
Generative Adversarial Networks

Generated cat images [1]

[1] Mao X., Li Q. (2021) Generative Adversarial Networks for Image Generation. Springer, Singapore.

Christa Zoufal, ouf@zurich.ibm.com

GAN Architecture



Discriminator Network: Distinguish between training data x_i and generated data $G_\theta(z_i)$

Generator Network: Create data $G(z_i)$ that the discriminator cannot distinguish from the **real** data x_i

Christa Zoufal, ouf@zurich.ibm.com

Training the discriminator

Qiskit

$$L_D(\phi) = \frac{1}{m}\sum_{i=1}^{m}\left[\log\left(D_\phi\left(x^{(i)}\right)\right) + \log\left(1 - D_\phi\left(G_\theta\left(z^{(i)}\right)\right)\right)\right]$$

Number of data samples in batch

Training data samples

Random input (prior distribution)

Maximize $L_D(\phi)$ w.r.t ϕ

Goal: Discrimination and labelling of **training data samples** and **generated data samples**

Christa Zoufal, ouf@zurich.ibm.com

Training the generator

Qiskit

$$L_G(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\log\left(D_\phi\left(G_\theta\left(z^{(i)}\right)\right)\right)$$

Number of data samples in batch

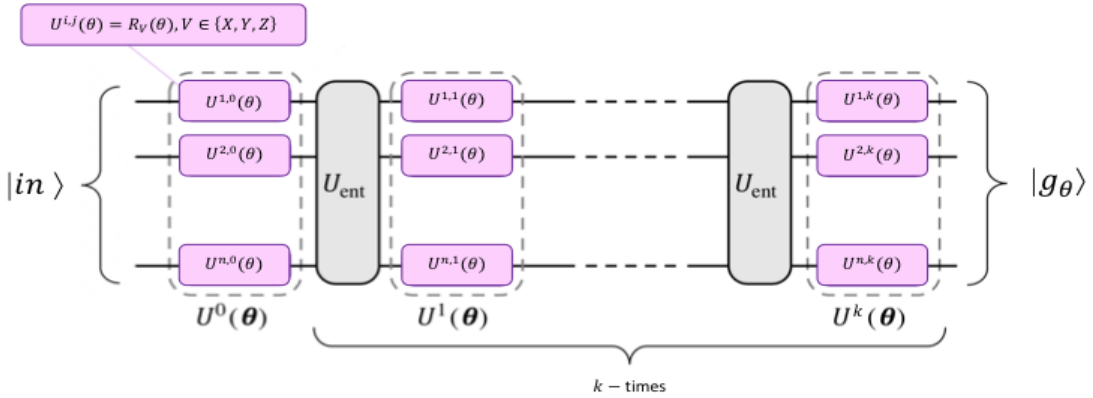
Random input (prior distribution)

Maximize $L_G(\theta)$ w.r.t. θ

Target: Get the discriminator to label **generated data samples** as **training data samples**

Christa Zoufal, ouf@zurich.ibm.com

Quantum Generator



$$G_\theta |in\rangle = |g_\theta\rangle \xrightarrow{\text{Measurement: quantum → classical}} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \xrightarrow{\text{Mapping to the training data feature space}} g^l$$

Christa Zoufal, ouf@zurich.ibm.com

Training

Qiskit

$$L_D(\phi, \theta) = \frac{1}{m} \sum_{l=1}^{m} \left[\log D_\phi \left(x^l \right) + \log \left(1 - D_\phi \left(g^l \right) \right) \right]$$

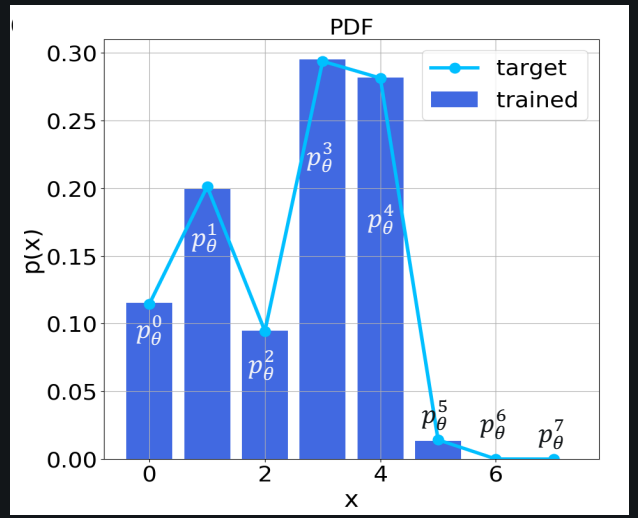
$$G_\theta |in\rangle = |g_\theta\rangle \xrightarrow{\text{measure}} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \xrightarrow{\text{map}} g^l$$

$$L_G(\phi, \theta) = -\frac{1}{m} \sum_{l=1}^{m} \left[\log \left(D_\phi \left(g^l \right) \right) \right]$$

Christa Zoufal, ouf@zurich.ibm.com

Quantum Generator

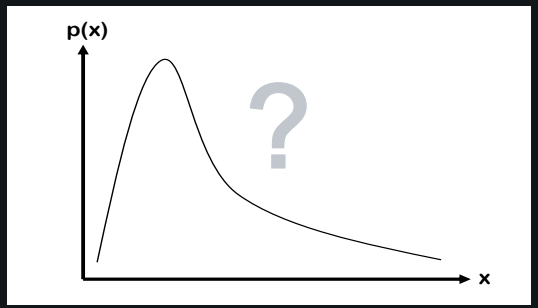
$$|g_\theta\rangle = \sum_{i=0}^{2^n-1} \sqrt{p_\theta^i}\,|g^i\rangle$$



Christa Zoufal, ouf@zurich.ibm.com

Data Loading with qGANs

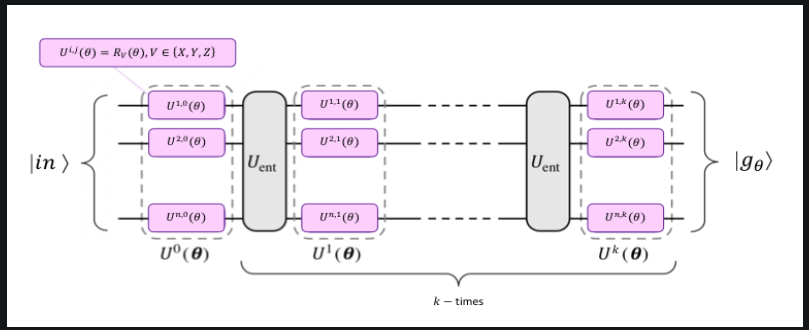
Qiskit

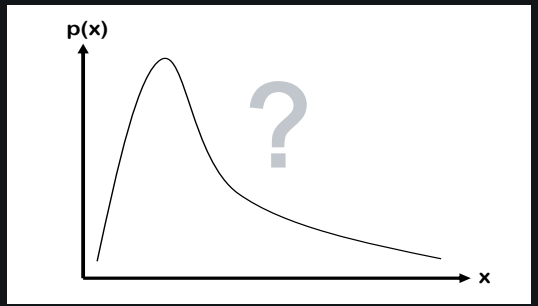


Christa Zoufal, ouf@zurich.ibm.com

Data Loading with qGANs

Qiskit

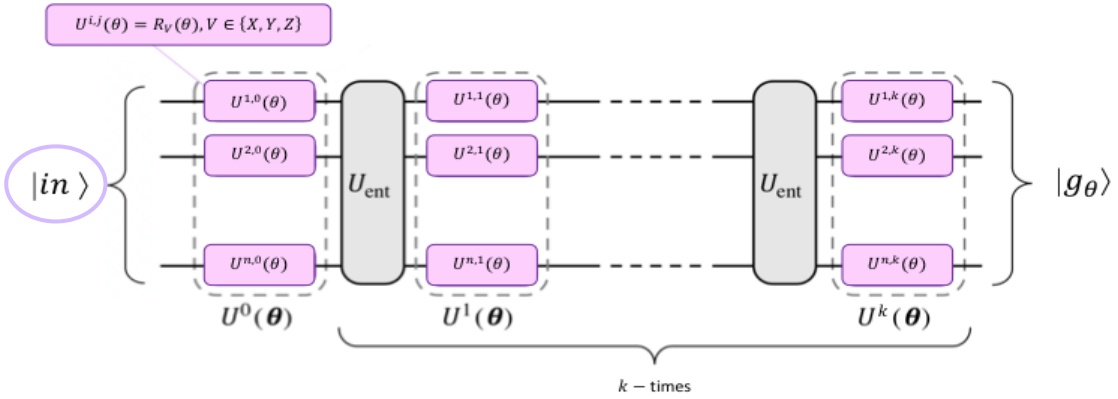




$$|g_\theta\rangle = \sum_{i=0}^{2^n-1} \sqrt{p_\theta^i}\,|g^i\rangle$$

Christa Zoufal, ouf@zurich.ibm.com

Quantum Generator



What is a good initial state $|in\rangle$?

Investigated candidates: random, uniform, normal according to 1^{st} and 2^{nd} momentum of the training data

Christa Zoufal, ouf@zurich.ibm.com

Benchmarking

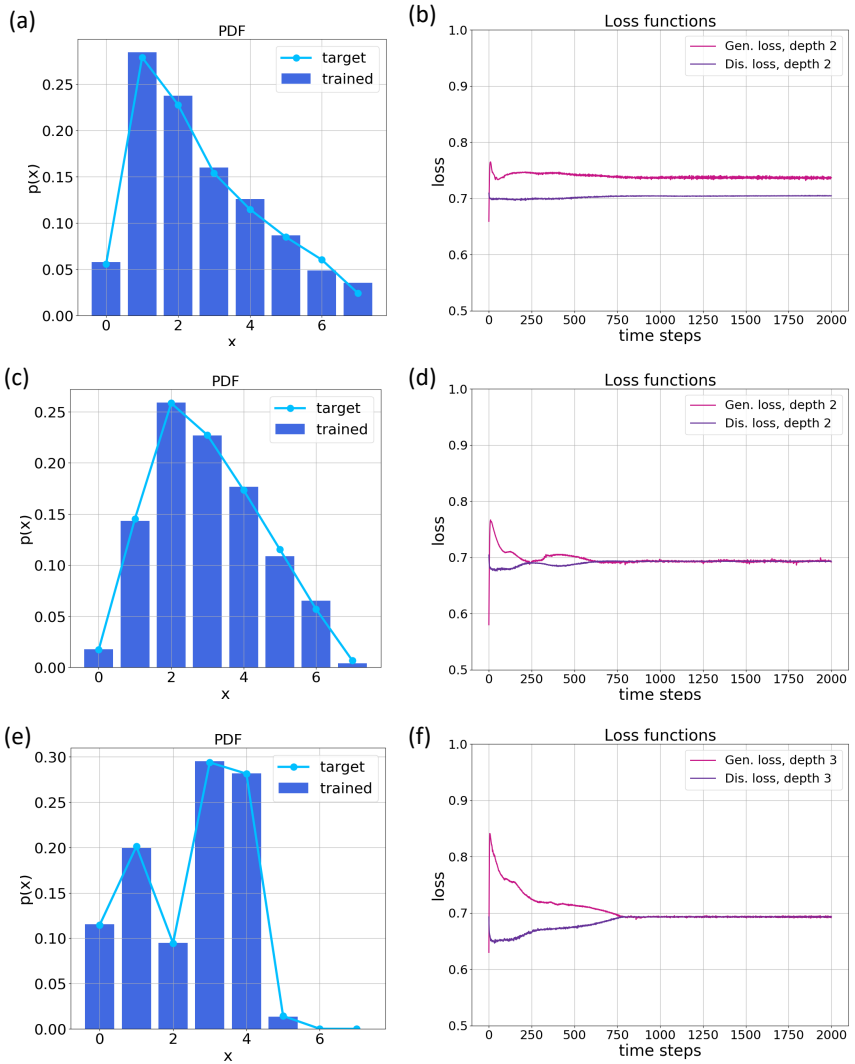
Training Data

Samples from a discretized, & truncated distributions

Generator

Variational form: $RealAmplitudes$

Number qubits: 3



Benchmarking

Qiskit

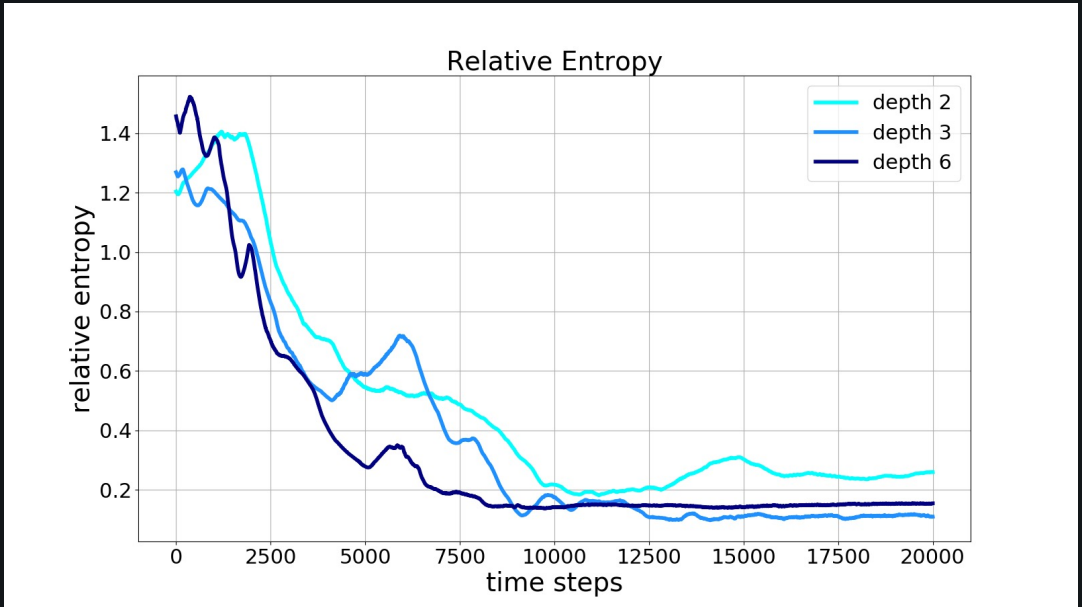
Training Data

First 2 principle components of multivariate, constant maturity treasury rates of US government bonds

Generator

Variational form: $RealAmplitudes$

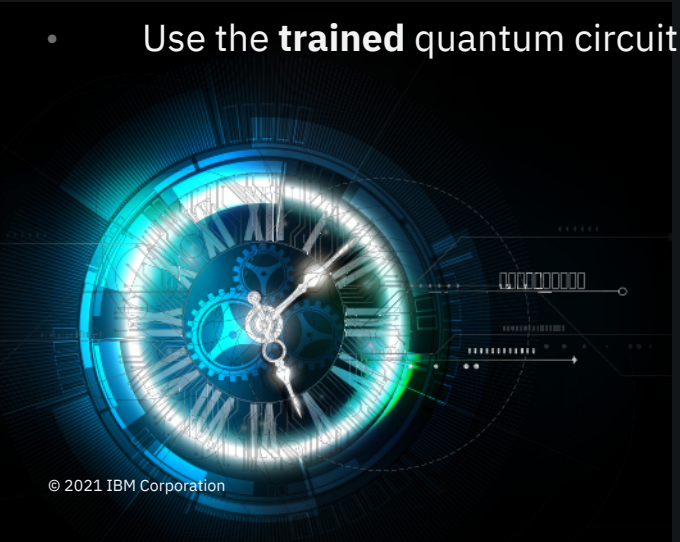
Number qubits: 6



Christa Zoufal, ouf@zurich.ibm.com

Summary

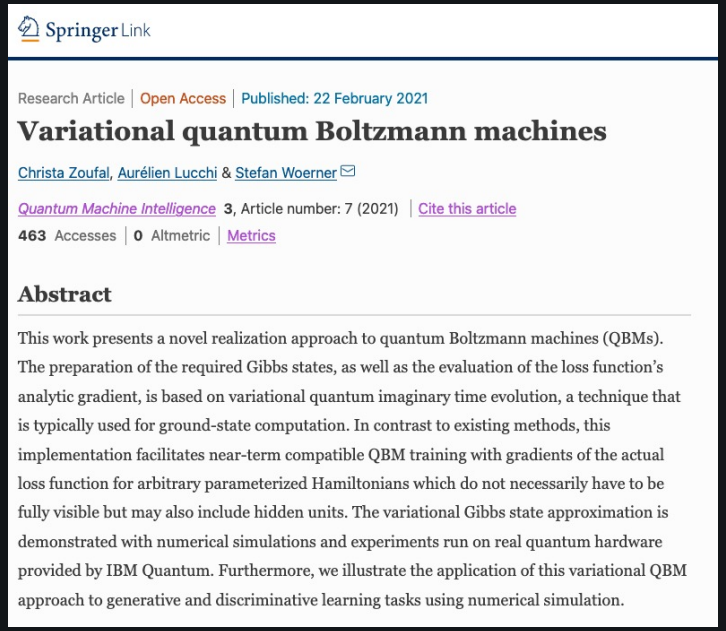
- **Efficient** loading of approximations to generic probability distributions
- Replace classical neural network with **parameterized** quantum circuit
- Use the **trained** quantum circuit for state preparation in quantum algorithms



Christa Zoufal, ouf@zurich.ibm.com

Quantum Boltzmann Machines

Qiskit



Springer Link

Research Article | Open Access | Published: 22 February 2021

Variational quantum Boltzmann machines

Christa Zoufal, Aurélien Lucchi & Stefan Woerner

Quantum Machine Intelligence **3**, Article number: 7 (2021) | Cite this article

463 Accesses | **0** Altmetric | Metrics

Abstract

This work presents a novel realization approach to quantum Boltzmann machines (QBMs). The preparation of the required Gibbs states, as well as the evaluation of the loss function's analytic gradient, is based on variational quantum imaginary time evolution, a technique that is typically used for ground-state computation. In contrast to existing methods, this implementation facilitates near-term compatible QBM training with gradients of the actual loss function for arbitrary parameterized Hamiltonians which do not necessarily have to be fully visible but may also include hidden units. The variational Gibbs state approximation is demonstrated with numerical simulations and experiments run on real quantum hardware provided by IBM Quantum. Furthermore, we illustrate the application of this variational QBM approach to generative and discriminative learning tasks using numerical simulation.

Idea:

Use structural knowledge to define your state preparation problem

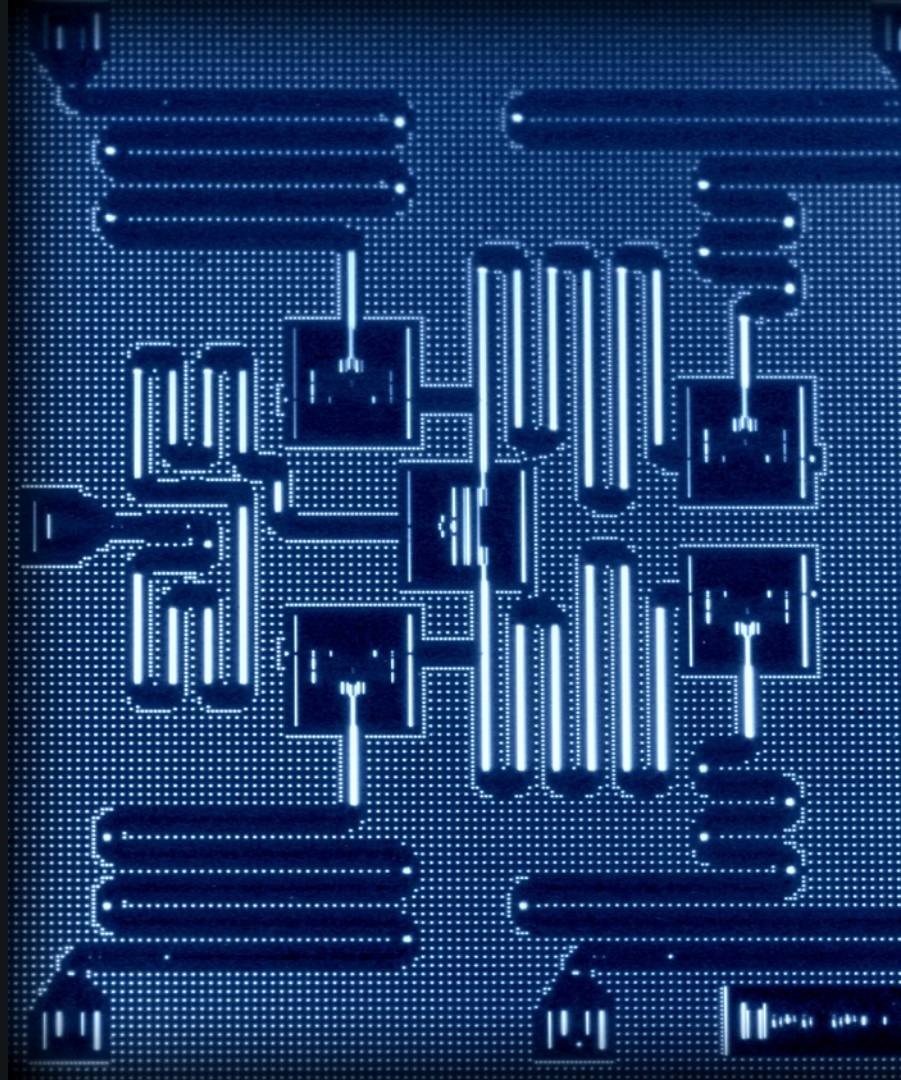
Implementation:

Approximate Gibbs State with a constant depth quantum circuit

Christa Zoufal, ouf@zurich.ibm.com

Outlook

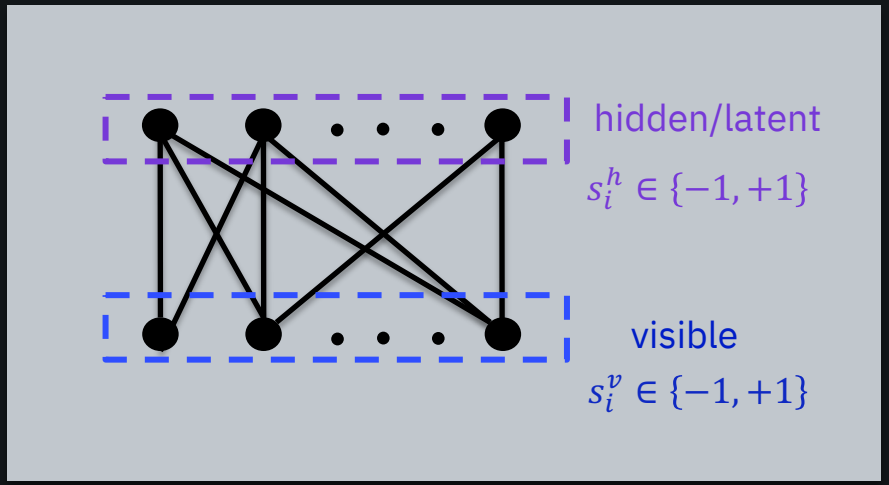
- What are (Quantum) Boltzmann Machines?
- How do we realize Quantum Boltzmann Machines?
- Numerical examples



Boltzmann Machines

Qiskit

Energy-based Machine Learning Models



Energy function defining the connectivity

$$E_\theta = -\left(\sum_{i<j} \theta_{ij} s_i s_j + \sum_i \tilde{\theta}_i s_i\right)$$

for nodes $\{s_i\}$ and parameters $\{\theta_{ij}\}, \{\tilde{\theta}_i\}$

Probability for a configuration of ***visible nodes***

$$P(v) = e^{-E(v)}/Z$$

Normalization factor

$$Z = \sum_u e^{-E(u)}$$

Christa Zoufal, ouf@zurich.ibm.com

Boltzmann Machines

Qiskit

Aim

Train $\{\theta_{ij}\}, \{\tilde{\theta}_i\}$ s.t. $P(v)$ represents the probability distribution underlying given training data (& training labels)

Loss function

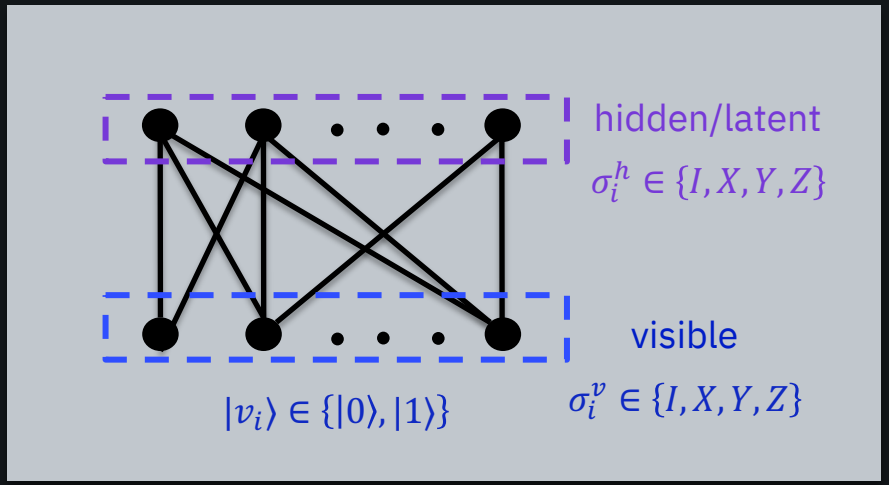
$$\min_{\theta_{ij}, \tilde{\theta}_i} L = -\sum_v P^{data}(v) \log\left(P^{BM}(v)\right)$$

Christa Zoufal, ouf@zurich.ibm.com

Quantum Boltzmann Machines

Qiskit

Let's replace bits with qubits.



Hamiltonian defining the connectivity

$$H_\theta = -\left(\sum_{i<j} \theta_{ij}\sigma_i\sigma_j + \sum_i \tilde{\theta}_i\sigma_i\right)$$

for Pauli operators $\{\sigma_i\}$ and parameters $\{\theta_{ij}\}, \{\tilde{\theta}_i\}$

Probability to measure configuration of ***visible qubits*** $|v\rangle$

$$P^{QBM}(v) = Tr\left[\Lambda_v \rho^{Gibbs}\right]$$

Quantum ***Gibbs state***

$$\rho^{Gibbs} = \frac{e^{-H}}{Z}$$

Normalization factor

$$Z = Tr[e^{-H}]$$

Difficult to evaluate classically for some Hamiltonians → sign problem

Christa Zoufal, ouf@zurich.ibm.com

Quantum Boltzmann Machines

Qiskit

Aim:
Train $\{\theta_{ij}\}, \{\tilde{\theta}_i\}$ s.t. $P^{QBM}(v) = Tr[\Lambda_v \rho^{Gibbs}]$ represents the probability distribution underlying given training data (& training labels)

Loss function:

1. Classical: $\min_{\theta_{ij}, \tilde{\theta}_i} L = -\sum_v P^{data}(v) \log\left(P^{QBM}(v)\right)$ → Match the sampling probability of a quantum state to the sampling probability of a classical data set

2. Quantum: $\min_{\theta_{ij}, \tilde{\theta}_i} L = -Tr[\rho^{data} \log(\rho^{Gibbs})]$ → Encode the classical probabilities into a rank-1 quantum state

Christa Zoufal, ouf@zurich.ibm.com

Quantum Imaginary Time Evolution

Qiskit

QITE: Given an n-qubit Hamiltonian $H = \sum_i \theta_i h_i$

Wick-rotated Schrödinger equation $\frac{d|\psi(\tau)\rangle}{d\tau} = -(H - E_\tau)|\psi(\tau)\rangle$

$|\psi(\tau)\rangle = \frac{e^{-H\tau}}{C(\tau)}|\psi(0)\rangle$

Normalization $C(\tau) = \sqrt{Tr[e^{-2H\tau}|\psi(0)\rangle\langle\psi(0)|]}$

VarQITE: $\left|\tilde{\psi}\left(\vec{\omega}(\tau)\right)\right\rangle = V\left(\vec{\omega}(\tau)\right)|\psi_{in}\rangle$

parametrized Ansatz input state

Idea:
Approximate state evolution with parameter evolution

Christa Zoufal, ouf@zurich.ibm.com

Parameter Evolution

Qiskit

McLachlan's variational principle for ITE [1]

$$\delta \left\| \left(\frac{d}{d\tau} + H - E_\tau \right) |\tilde{\psi}(\vec{\omega}(\tau))\rangle \right\| = 0$$

gives a system of linear equations

$$A\dot{\vec{\omega}}(\tau) = C$$

$$A_{ij} = Re\left(\frac{\partial \langle \tilde{\psi}(\vec{\omega}(\tau))|}{\partial \omega_i} \frac{\partial |\tilde{\psi}(\vec{\omega}(\tau))\rangle}{\partial \omega_j} \right) \qquad C_i = \frac{1}{2} \frac{\partial \langle \tilde{\psi}(\vec{\omega}(\tau))|H|\tilde{\psi}(\vec{\omega}(\tau))\rangle}{\partial \omega_i}$$

➔Evolve $\vec{\omega}\,(\tau)$ e.g. with explicit Euler

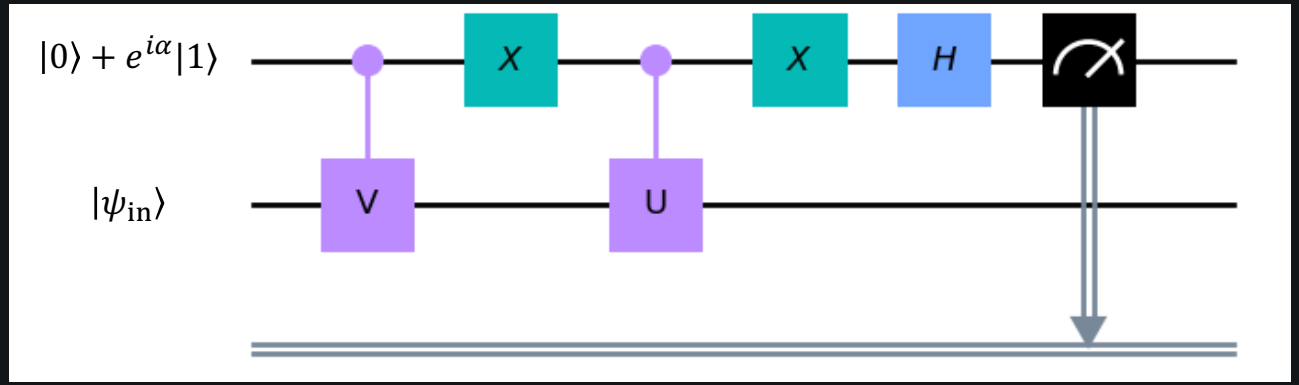
$$\vec{\omega}\,(\tau) \simeq \vec{\omega}\,(\tau - \delta\tau) + \dot{\vec{\omega}}(\tau - \delta\tau)\delta\tau$$

$\mathcal{O}(m(k+m))$
m: # parameters in V
k: # parameters in H

[1] A.D. McLachlan. A variational solution of the time-dependent Schrödinger equation. Molecular Physics, 8(1):39-44, 1964.

Christa Zoufal, ouf@zurich.ibm.com

SLE Components



[1]

$$\text{Re}(e^{i\alpha}\text{Tr}[U^{\dagger}V|\psi_{in}\rangle\langle\psi_{in}|])$$

$$A_{ij} = Re\left(\frac{\partial\langle\tilde{\psi}(\vec{\omega}(\tau))|}{\partial\omega_i}\frac{\partial|\tilde{\psi}(\vec{\omega}(\tau))\rangle}{\partial\omega_j}\right)$$

$$C_i = Re\left(\frac{\partial\langle\tilde{\psi}(\vec{\omega}(\tau))|}{\partial\omega_i}H|\tilde{\psi}(\vec{\omega}(\tau))\rangle\right)$$

[1] Simulating physical phenomena by quantum networks- R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme,Phys. Rev. A 65, 042323 – 2002.

Christa Zoufal, ouf@zurich.ibm.com

VarQITE for Gibbs State Preparation

Qiskit

Target state

$$\rho_{Gibbs} = \frac{e^{-H_A/k_BT}}{Z}$$

Normalization factor $Z = Tr\left[e^{-H_A/k_BT}\right]$

Method

- $\tau = \frac{1}{2k_BT}$
- $H_{AB} = H_A \otimes I_B$
- $\left|\widetilde{\psi}(0)\right\rangle_{AB} = (|00\rangle + |11\rangle)^{\otimes n} \Rightarrow \quad Tr_B\left[\left|\widetilde{\psi}(0)\right\rangle\left\langle\widetilde{\psi}(0)\right|_{AB}\right] = I\,\frac{1}{2^n}$

$\left|\widetilde{\psi}(\tau)\right\rangle = \frac{e^{-H_{AB}\tau}}{C(\tau)}\left|\widetilde{\psi}(\omega(0))\right\rangle$

$\tilde{\rho}_{Gibbs} = Tr_B[\left|\widetilde{\psi}(\tau)\right\rangle\left\langle\widetilde{\psi}(\tau)\right|]$

Christa Zoufal, ouf@zurich.ibm.com

Gibbs State Preparation Example

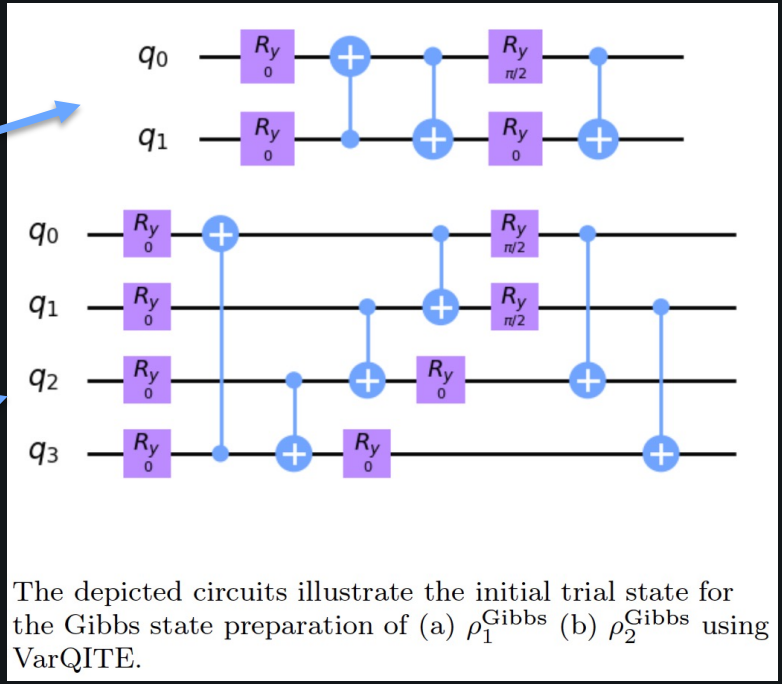
Qiskit

$$H_1 = 1.0Z,$$

$$H_2 = 1.0ZZ - 0.2ZI - 0.2IZ + 0.3XI + 0.3IX.$$

$$\rho_1^{\text{Gibbs}} = \begin{pmatrix} 0.12 & 0. \\ 0. & 0.88 \end{pmatrix},$$

$$\rho_2^{\text{Gibbs}} = \begin{pmatrix} 0.10 & -0.06 & -0.06 & 0.01 \\ -0.06 & 0.43 & 0.02 & -0.05 \\ -0.06 & 0.02 & 0.43 & -0.05 \\ 0.01 & -0.05 & -0.05 & 0.05 \end{pmatrix}.$$

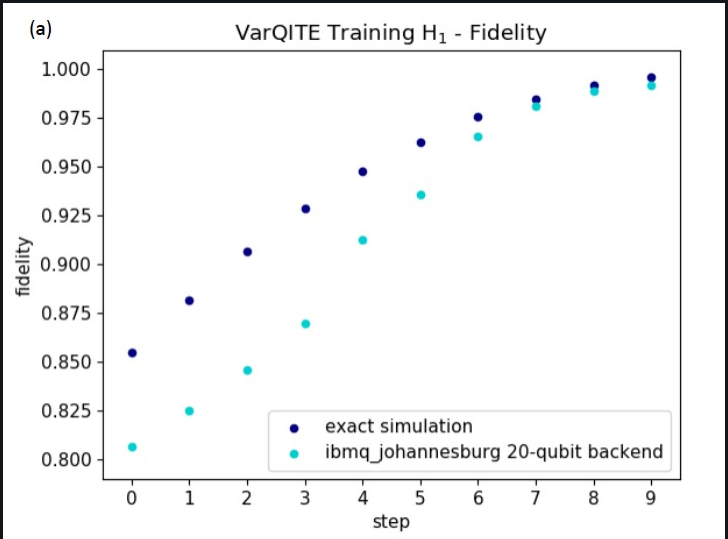


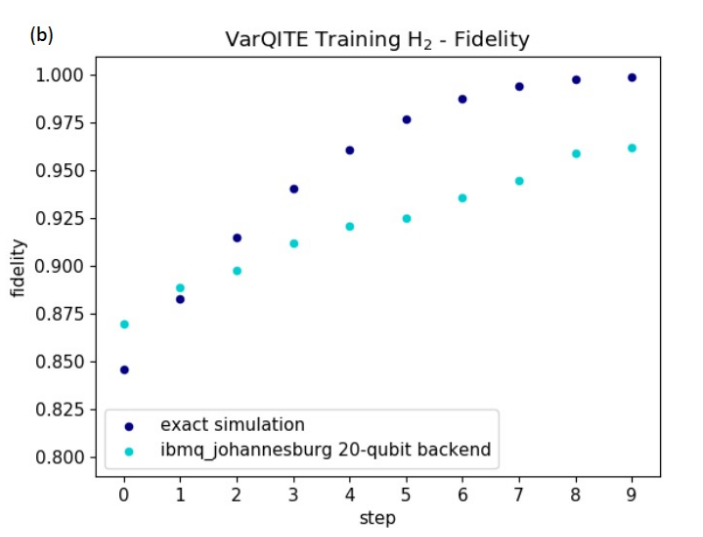
The depicted circuits illustrate the initial trial state for the Gibbs state preparation of (a) ρ_1^{Gibbs} (b) ρ_2^{Gibbs} using VarQITE.

Aim
Evaluate fidelity $\mathrm{F}(\rho_{QBM}, \rho_{target})$ throughout VarQITE

Christa Zoufal, ouf@zurich.ibm.com

Gibbs State Preparation Example





Backend: IBM Q Johannesburg

VarQITE: 10 steps/Gibbs state preparation

Convergence Measure: $\mathrm{F}(\rho_{QBM}, \rho_{target})$

Fidelity between trained and target Gibbs state with VarQITE for (a) ρ_1^{Gibbs} (b) ρ_2^{Gibbs} trained with an ideal simulator and real quantum hardware, i.e., the *ibmq_johannesburg* 20-*qubit* backend. Each simulation used 10 time steps.

Christa Zoufal, ouf@zurich.ibm.com

QBM Training

Qiskit

Loss function

$$L = -\sum_v P^{data}(v)\log\left(P^{QBM}(v)\right) = -\sum_v P^{data}(v)\log\left(Tr\left[\Lambda_v \rho^{Gibbs}\right]\right)$$

Gradient loss function

$$\frac{\partial L}{\partial \theta_i} = -\sum_v P_v^{data}\frac{\frac{\partial Tr\left[\Lambda_v \rho^{Gibbs}\right]}{\partial \theta_i}}{Tr[\Lambda_v \rho^{Gibbs}]}$$

$$\frac{\frac{\partial Tr\left[\Lambda_v \rho^{Gibbs}\right]}{\partial \theta_i}}{Tr[\Lambda_v \rho^{Gibbs}]} = \frac{Tr\left[\Lambda_v \, \partial e^{-H} / \partial \theta_i\right]}{Tr[\Lambda_v e^{-H}]} - \frac{Tr\left[\partial e^{-H} / \partial \theta_i\right]}{Tr[e^{-H}]} \quad -\left\langle \frac{\partial H}{\partial \theta_i} \right\rangle$$

➔ **Golden-Thompson**

Christa Zoufal, ouf@zurich.ibm.com

QBM Training

Qiskit

Loss function

$$L = -\sum_v P^{data}(v)\log\left(P^{QBM}(v)\right) = -\sum_v P^{data}(v)\log\left(Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}(\vec{\omega}(\tau))\right]\right)$$

Gradient loss function

$$\frac{\partial L}{\partial \theta_i} = -\sum_v P_v^{data}\frac{\frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \theta_i}}{Tr[\Lambda_v \tilde{\rho}^{Gibbs}]}$$

➔ **But with VarQITE**

$$\frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \theta_i} = \frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \vec{\omega}(\tau)}\frac{\partial \vec{\omega}(\tau)}{\partial \theta_i}$$

Automatic differentiation

Christa Zoufal, ouf@zurich.ibm.com

QBM Training

Qiskit

Gradient w.r.t. trial state parameters

$$\frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \vec{\omega}(\tau)} = \frac{\partial \langle \psi(\tau)|\Lambda_v|\psi(\tau)\rangle}{\partial \vec{\omega}(\tau)}$$

First order gradient of an expectation value w.r.t. a quantum state

Gradient w.r.t. the Hamiltonian parameters

$$\partial\theta_i \left(A\dot{\vec{\omega}}(\tau)\right) = \partial\theta_i C$$

Compute the derivative of the SLE resulting from McLachlan

$$A\frac{\partial\dot{\vec{\omega}}(\tau)}{\partial\theta_i} = \frac{\partial C}{\partial\theta_i} - \frac{\partial A}{\partial\theta_i}\dot{\vec{\omega}}(\tau)$$

System of linear equations that involves higher order gradients

Evolve $\dot{\vec{\omega}}(\tau)$ with explicit Euler

$$\frac{\partial\vec{\omega}(\tau)}{\partial\theta_i} \simeq \frac{\partial\vec{\omega}(\tau-\delta\tau)}{\partial\theta_i} + \frac{\partial\dot{\vec{\omega}}(\tau-\delta\tau)}{\partial\theta_i}\delta\tau$$

Christa Zoufal, ouf@zurich.ibm.com

QBM Training

Qiskit

Loss function

$$L = -\sum_v P^{data}(v)\log\left(P^{QBM}(v)\right) = -\sum_v P^{data}(v)\log\left(Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}(\vec{\omega}(\tau))\right]\right)$$

Gradient loss function

$$\frac{\partial L}{\partial \theta_i} = -\sum_v P_v^{data} \frac{\frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \theta_i}}{Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}$$

➔ **But with VarQITE**

$$\frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \theta_i} = \frac{\partial Tr\left[\Lambda_v \tilde{\rho}^{Gibbs}\right]}{\partial \vec{\omega}(\tau)} \frac{\partial \vec{\omega}(\tau)}{\partial \theta_i}$$

Automatic differentiation

Christa Zoufal, ouf@zurich.ibm.com

Implementation

Qiskit

$A\dot{\vec{\omega}}(\tau) = C$

Training procedure

$H = \sum_i \theta_i h_i$ $\quad A\frac{\partial\dot{\vec{\omega}}(\tau)}{\partial\theta_i} = \frac{\partial C}{\partial\theta_i} - \frac{\partial A}{\partial\theta_i}\dot{\vec{\omega}}(\tau)$x

Initialize H_θ and choose a set of initial θ

1. Use VarQITE to prepare ρ_ω^{Gibbs}

   & to compute $\frac{\partial\omega}{\partial\theta}$ for H_θ

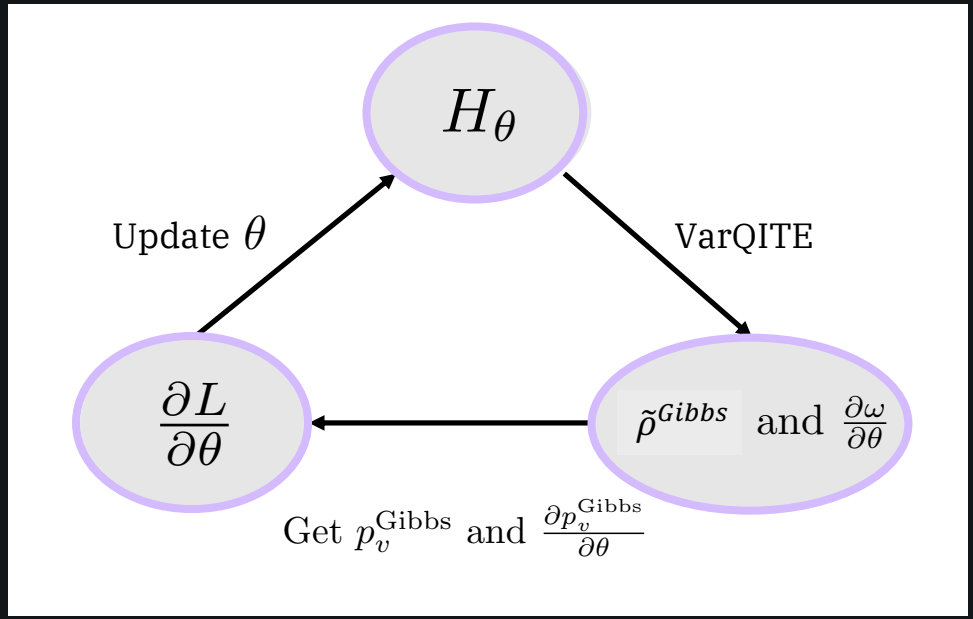
2. Sample to get p_v^{Gibbs}

   & apply the chain rule to get $\frac{\partial p_v^{Gibbs}}{\partial\theta}$

3. Compute the resulting $\frac{\partial L}{\partial\theta}$

4. Update θ with a classical optimizer

Repeat steps 1.-4.



$$\frac{\partial L}{\partial\theta_i} = -\sum_v P_v^{data}\frac{\frac{\partial Tr[\Lambda_v\tilde{\rho}^{Gibbs}]}{\partial\theta_i}}{Tr[\Lambda_v\tilde{\rho}^{Gibbs}]}$$

Christa Zoufal, ouf@zurich.ibm.com

Generative QBM Example

Target

Bell state

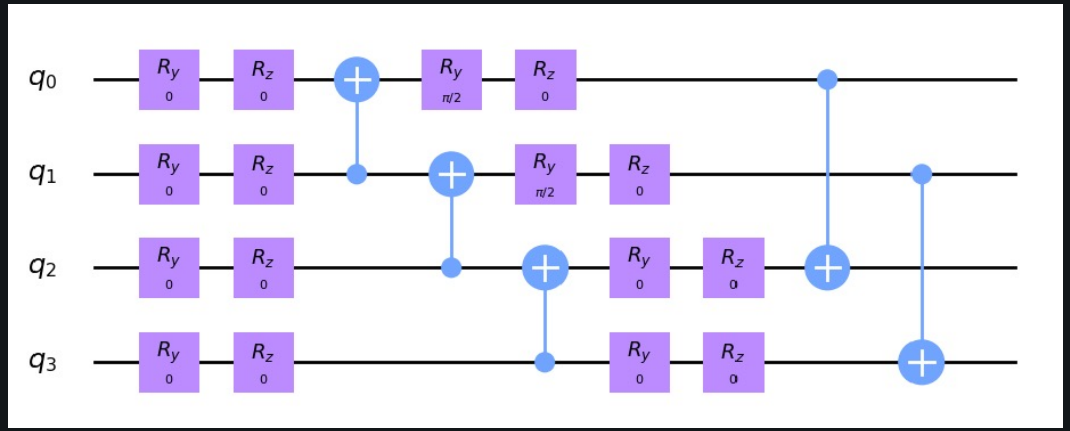
sampling probabilities ground state: $\mathrm{p}^{data} = [0.5, 0, 0, 0.5]$

Specify Model

$H_{QBM} = \alpha \cdot ZZ + \beta \cdot IZ + \gamma \cdot ZI$

$\mathrm{p}_{QBM} = Tr[|i\rangle\langle i|\rho_{QBM}]$

Starting state $|\tilde{\psi}(\vec{\omega}(0))\rangle =$



Aim
Train α, β, γ

Christa Zoufal, ouf@zurich.ibm.com

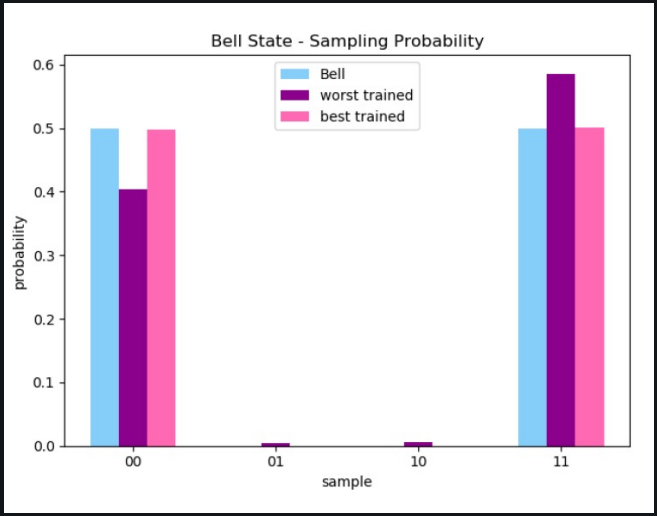
Generative QBM Example

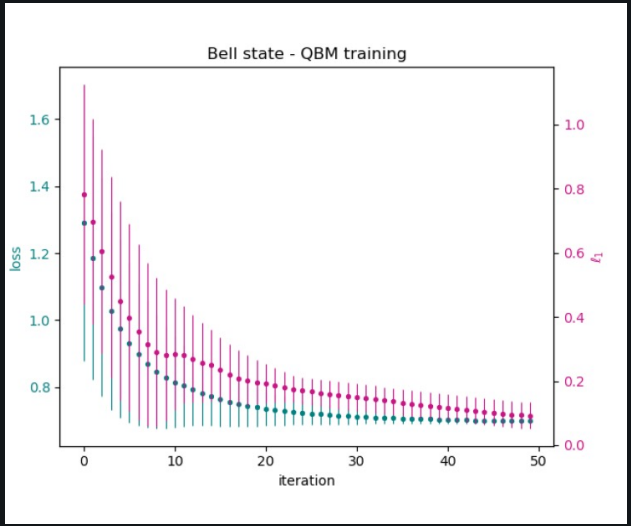
Initialization: $\alpha, \beta, \gamma \sim \mathrm{U}[-1, 1]$

VarQITE: 10 steps/Gibbs preparation

Optimization: Amsgrad

Convergence Measure: loss, $\|p_{ground} - p_{QBM}\|_1$

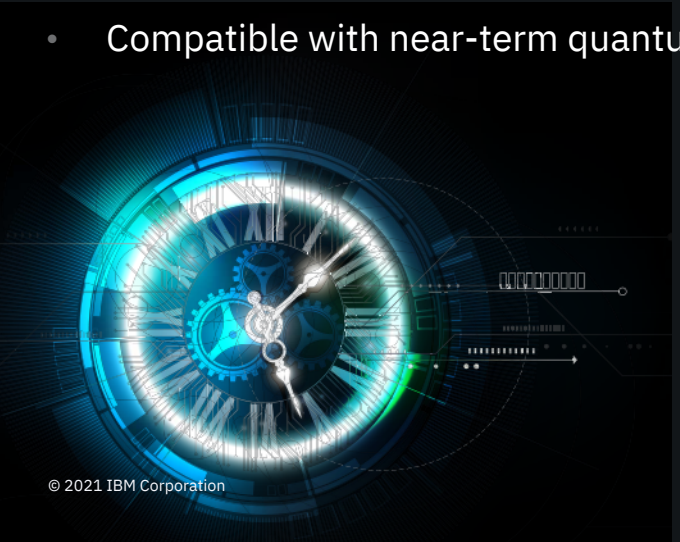




Christa Zoufal, ouf@zurich.ibm.com

Summary

- Explore **data structure** to construct suitable Hamiltonians
- Use **automatic differentiation** for the QBM training
- Compatible with near-term quantum computers due to **constant-depth** quantum circuits



Christa Zoufal, ouf@zurich.ibm.com

Key messages

- To explore quantum algorithms for **data processing** we need to load the data into a quantum state
- Data loading can be **difficult** or even exponentially expensive
- QML can be used to implement approximate but **efficient** data loading schemes

Christa Zoufal, ouf@zurich.ibm.com