

日本語訳『Qiskit Textbook Machine Learning』勉強会

- Training parameterized quantum circuits

Daiki Murata

Qiskit Advocate / Senior Architect - IBM Consulting



本日のテーマ

Pages

Introduction

Parameterized quantum circuits

Data encoding

Training parameterized quantum circuits **今回**

Supervised learning

Variational classification

Quantum feature maps and kernels

Unsupervised learning

Quantum generative adversarial networks

Project

Quantum Tokyo

第一回

第二回

データの符号化

1. 計算基底符号化

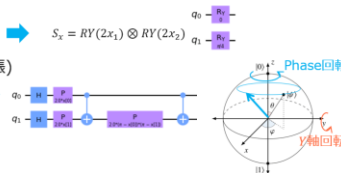
例) データビット $x = \{x^{(1)} = 101, x^{(2)} = 111\}$ $\rightarrow |\mathcal{X}\rangle = \frac{1}{\sqrt{2}}(|101\rangle + |111\rangle)$

2. 振幅符号化

例) $x = \{x^{(1)} = (1.5, 0), x^{(2)} = (-2, 3)\}$ $\rightarrow |\mathcal{X}\rangle = \frac{1}{\sqrt{15.25}}(1.5|00\rangle - 2|10\rangle + 3|11\rangle)$

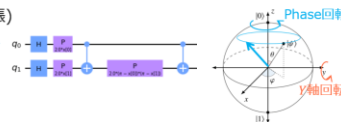
3. 角度符号化

例) データポイント $x = (x_1, x_2)$ $\rightarrow S_x = RY(2x_1) \otimes RY(2x_2)$



4. 任意の符号化(角度符号化の拡張)

例) データポイント $x = (x_1, x_2)$ \rightarrow



パラメータ化された量子回路の特徴づけ

Textbookの参考文献1 *Sim et al. (2019)*では機械学習モデルとして最適なパラメータ化された量子回路を選択するために以下の指標を提唱しています。

- **Expressibility** (表現能力)
- **Entangling Capability** (エンタングリング能力)

回路の定量化

①エンコード ②量子操作

$$|0\rangle^{\otimes n} \xrightarrow{U_{\text{encode}}} U(\theta) \xrightarrow{\quad} |\psi_{\text{out}}\rangle$$

量子コンピューター

⑥重み θ の更新

④コスト関数の計算

⑤最適化計算

古典コンピューター

③測定

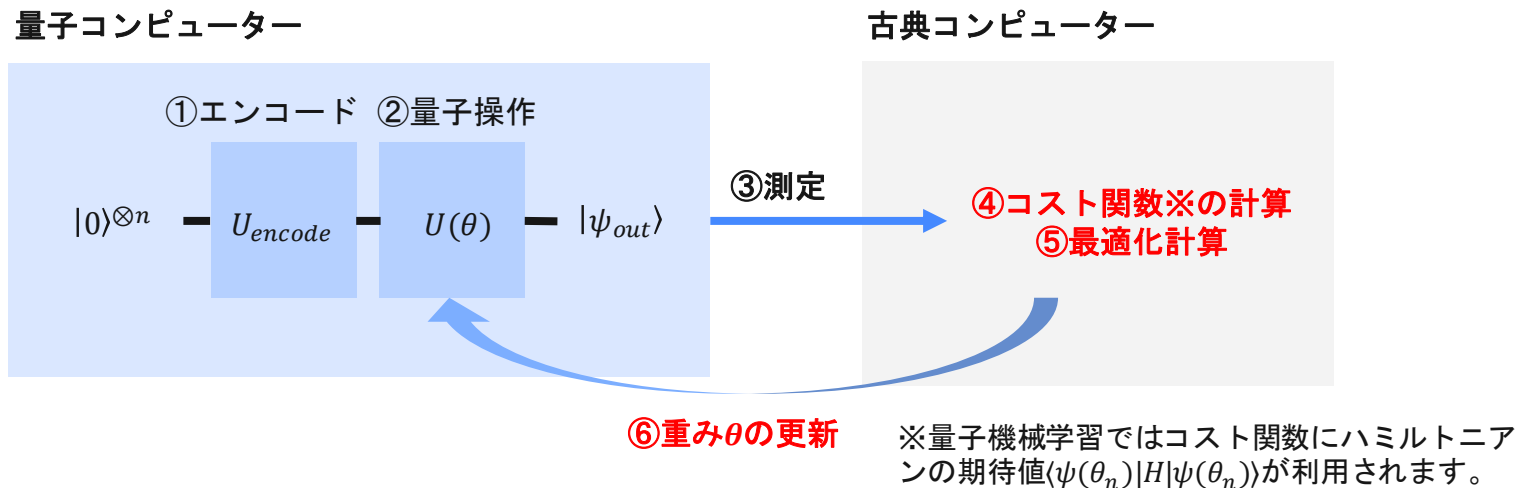
Qt

パラメータ化された回路の学習

量子機械学習の学習プロセスは古典モデルと似ています。

コスト(損失)関数 $f(\theta)$ の最適化 → 最適解 θ^* でパラメータを更新

モデルの学習において、コスト関数の適切かつ効率的な最適化が求められています。



最適化手法

最適化の手法は大きく2つに分類することができます。

- 勾配情報を使う手法
- 勾配情報を使わない手法

今回はオーソドックスな勾配情報使った手法について見ていきましょう。

最適化手法

勾配ベースの手法

有限差分法
解析的勾配法
自然勾配法
SPSA
etc...

勾配フリーな手法

one-shot法
Nelder-Mead法
etc...

普通の(vanilla)勾配

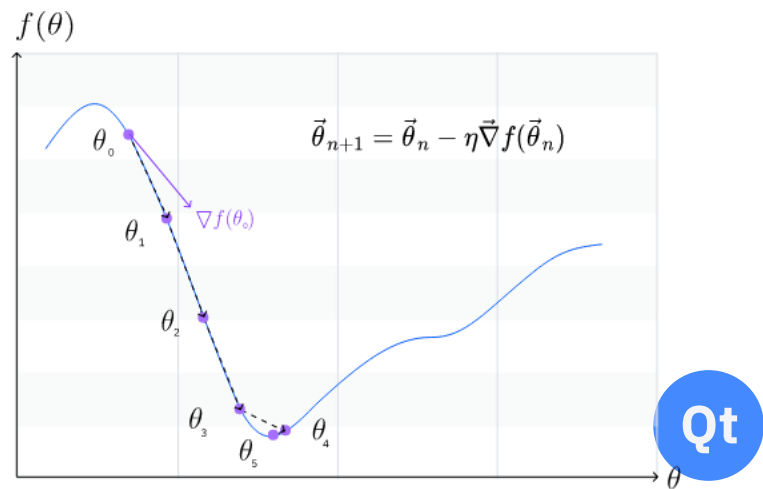
勾配とは関数 f を最も増加させる向きを定義したものです。

$$\nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}$$

最適化の基本的な手法である最急降下法は、勾配方向にパラメータの更新を繰り返し、コスト関数の(局所的)最小値を探索します。

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta \nabla f(\boldsymbol{\theta}_n)$$

$\eta > 0$: 学習率(ハイパーパラメータ)

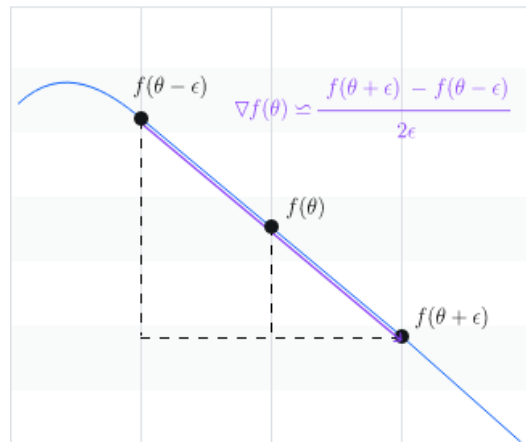


有限差分勾配

とはいえいつでも勾配が計算できるとも限りません。(勾配が計算できない、コスト関数がブラックボックス)

そんな場合は有限差分法による近似が用いられます。

$$\begin{aligned}\frac{df(x)}{dx} &= \lim_{n \rightarrow \infty} \frac{f(x + \epsilon) - f(x - \epsilon)}{2\epsilon} \\ &= \frac{f(x + \epsilon) - f(x - \epsilon)}{2\epsilon} + O(\epsilon^2) + O\left(\frac{d}{\epsilon}\right)\end{aligned}$$

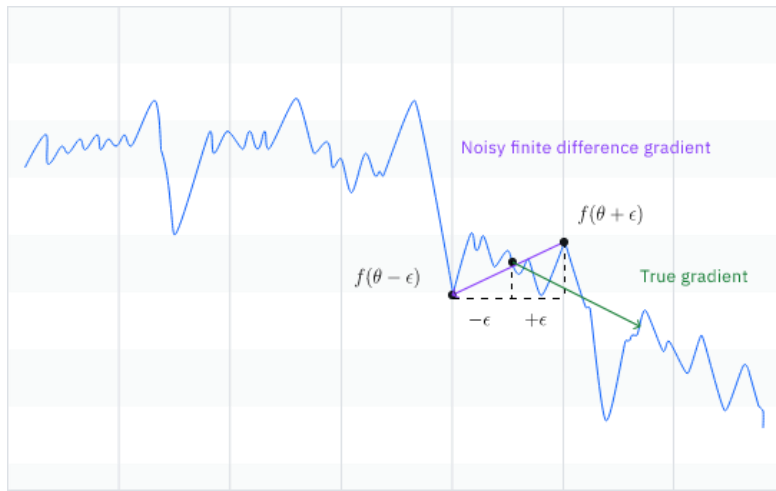


有限差分勾配の課題

有限差分法の近似は2点間の関数の値の差が推定誤差の2倍よりも大きい範囲で有効とされています。

$$2d \leq \epsilon \frac{df(x)}{dx}$$

最適化したい関数がノイジー(多くの鞍点を持つ)な場合、実行が難しいとされています。



解析的勾配(パラメータシフト法)

近似ではなく解析的に勾配を求める(偏微分計算)方法がパラメータシフト法です。

有限差分との違い

- 微分計算の評価に補助ビットを持ち出さずリソースを節約できる
- パラメータを大きくシフトさせて差を取るので誤差に対して堅牢
- 結果が厳密

$$\frac{\partial f}{\partial \theta_i} = \frac{f\left(\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_i\right) - f\left(\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_i\right)}{2}$$

パラメータシフトの導出イメージ

ansatzのパラメータが1量子ビットの回転ゲートの形($e^{\frac{iX\theta}{2}}$)で表現されるとします。

$$\langle H(\theta) \rangle = \langle 0 | e^{i\theta X/2} H e^{-i\theta X/2} | 0 \rangle$$

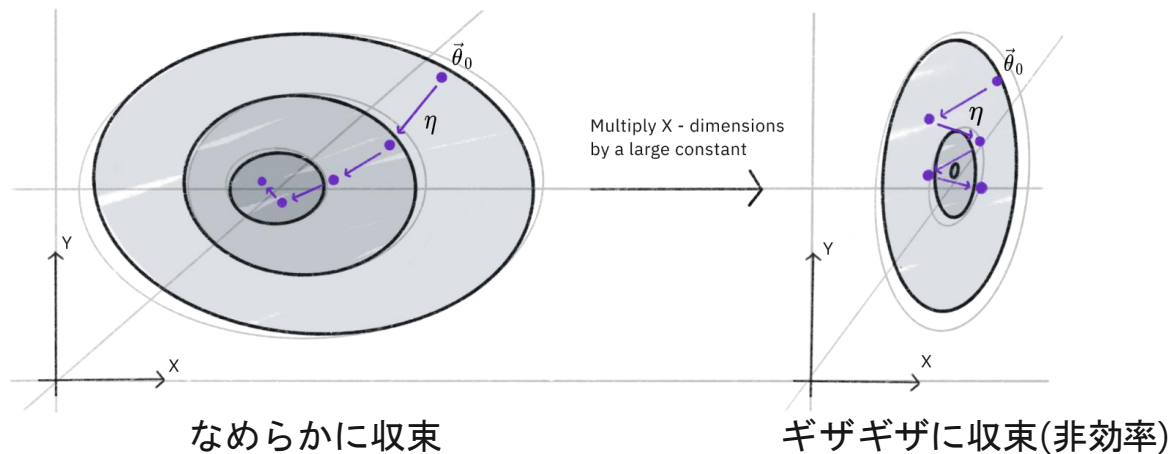
微分を取ると、

$$\begin{aligned} \frac{d\langle H(\theta) \rangle}{d\theta} &= \frac{i}{2} \left(\langle 0 | X e^{i\theta X/2} H e^{-i\theta X/2} | 0 \rangle - \langle 0 | e^{i\theta X/2} H e^{-i\theta X/2} X | 0 \rangle \right) \\ &= \frac{1}{2} \left[\left\langle H \left(\theta + \frac{\pi}{2} \right) \right\rangle - \left\langle H \left(\theta - \frac{\pi}{2} \right) \right\rangle \right] \quad \because e^{i\theta X/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} X \end{aligned}$$

自然勾配

最急降下法もいいけれど...必ずしもベストとは限りません

例)



モデル毎に最適な手法があるのでは・・・？

自然勾配

ベストな最適化手法はパラメータ空間のジオメトリと関連しています。
(最急降下法→ l_2 空間)

$$\begin{aligned}\theta_{n+1} &= \theta_n - \eta \nabla f(\theta_n) \\ &= \arg \min \left[\langle \theta - \theta_n, \nabla f(\theta_n) \rangle + \frac{1}{2\eta} \|\theta - \theta_n\|_2^2 \right]\end{aligned}$$

しかし l_2 空間は複雑なパラメータ空間にはうまくマッチしません。。。

量子自然勾配(Quantum Natural Gradient)は量子状態の空間において最急降下の方
向を決定する手法です。

$$\begin{aligned}\theta_{n+1} &= \arg \min \left[\langle \theta - \theta_n, \nabla f(\theta_n) \rangle + \frac{1}{2\eta} \|\theta - \theta_n\|_{g(\theta_n)}^2 \right] \\ &= \theta_n - \eta g^{-1}(\theta) \nabla f(\theta_n)\end{aligned}$$

とってもラフなイメージ①

l_2 空間(ユークリッド空間)：距離は内積で定義される。

$$d = \sum \theta_i \theta_j$$

量子状態の空間でも内積を使って距離を定義できると仮定してみます。
この下で関数 $f(\theta)$ の変化が最大となる方向を探しましょう。

$$d = \sum g_{ij} \theta_i \theta_j \cdots (1)$$

(1)を満たしながらパラメータを変化させるとき、コスト関数の変化は

$$\Delta f = \eta \frac{\partial f}{\partial \theta} \cdot a$$

Δf を最大化する a が見つかればその方向に向かって進めば効率的に収束することになります。

とってもラフなイメージ②

a の大きさは方向によらず一定である必要があるので、この束縛条件下で Δf を最大化する条件をLagrangeの未定乗数法を用いて考えます。

$$\delta \left(\frac{\partial f}{\partial \theta} \cdot a - \sum_i \sum_j \frac{\lambda}{2} g_{ij} a_i a_j \right) = 0$$
$$\therefore a \propto g^{-1} \frac{\partial f}{\partial \theta}$$

最急降下は a と反対方向であればよいので

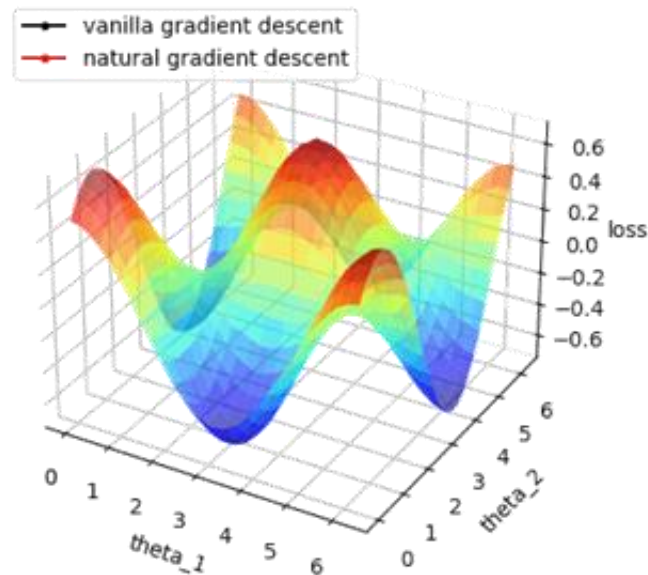
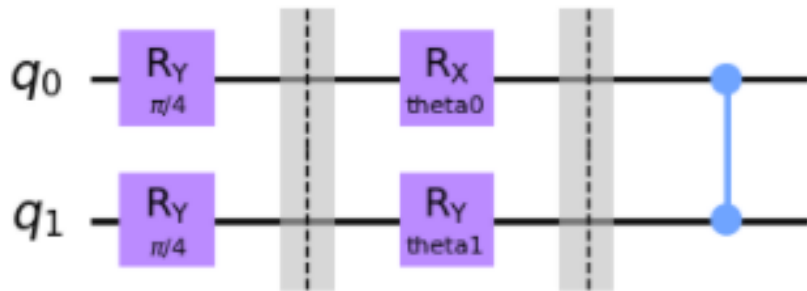
$$\theta_{n+1} = \theta_n - \eta g^{-1} \frac{\partial f}{\partial \theta}$$

Qiskitを使った実装例

2量子ビットのパラメータ化された回路を準備します。

コスト関数はハミルトニアン期待値 $\langle \psi(\theta) | X_1 | \psi(\theta) \rangle$ としたとき、

最急降下法と自然勾配を用いた勾配降下の収束の様子を確認してみましょう。



SPSA(同時摂動確率近似)

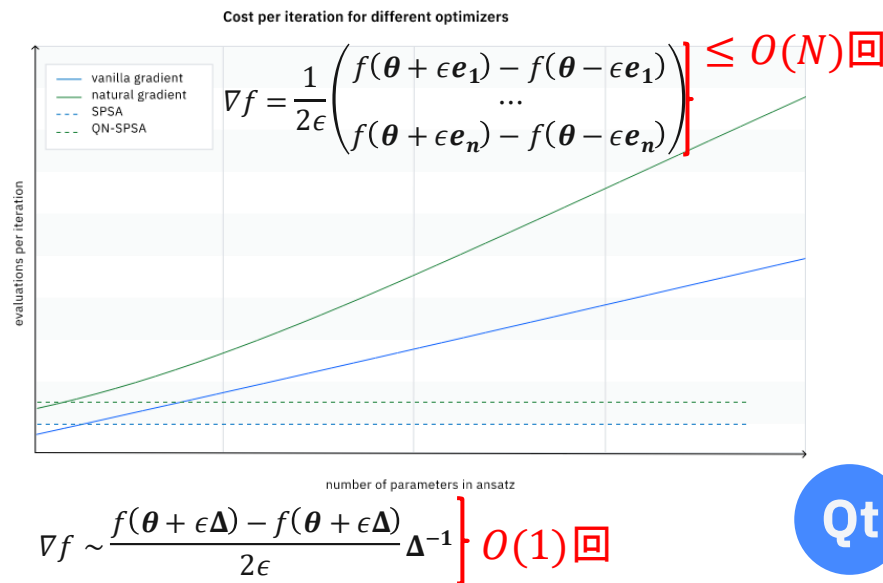
これまでの手法はパラメータの数だけ関数の評価が必要なので計算コスト高です。

SPSAは最適化の繰り返し毎にランダムに選んだ方向に対する差分を計算することで勾配を近似する方法です。

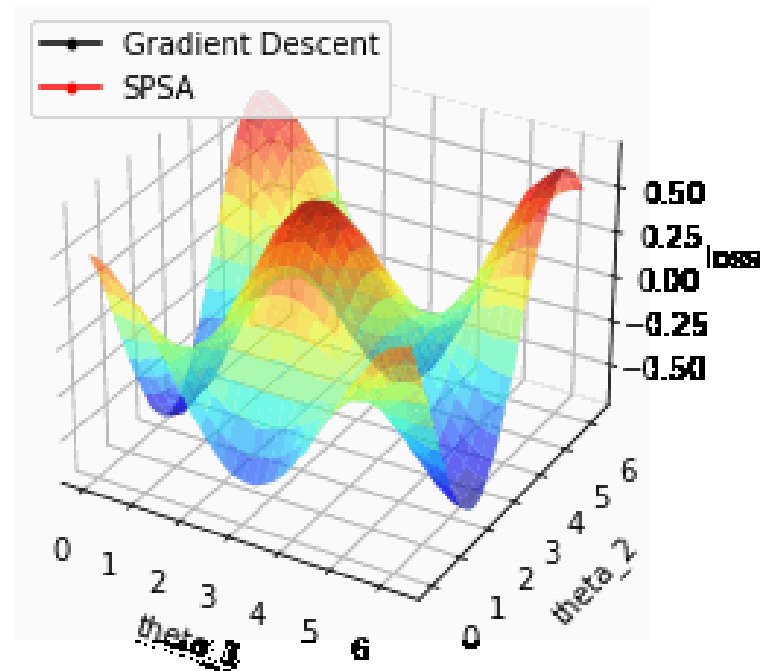
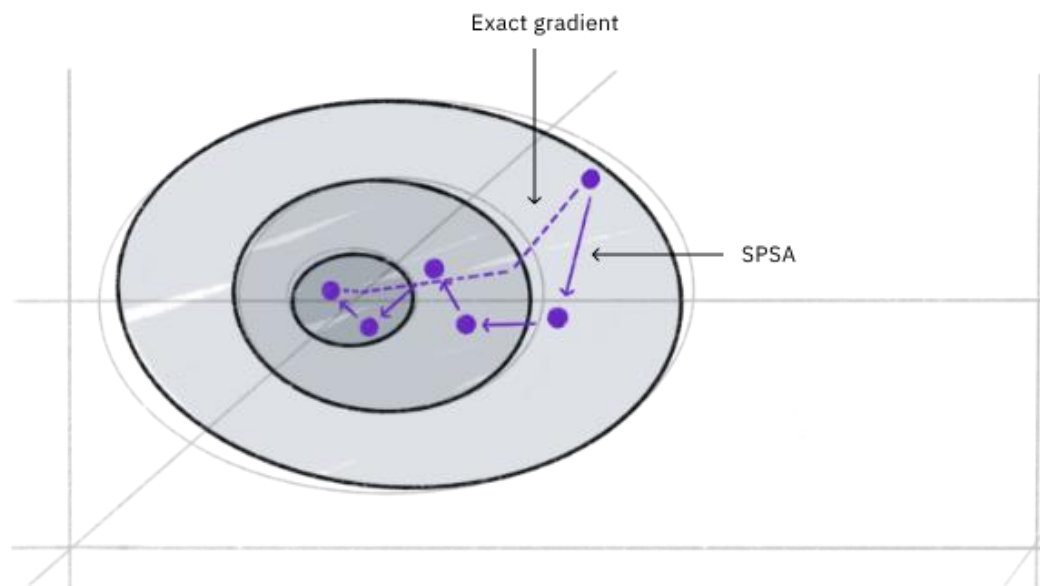
【メリット】

- ・ 計算コストが軽い
- ・ ノイズ(HW由来のエラー)に対して堅牢

$$\nabla f \sim \frac{f(\boldsymbol{\theta} + \epsilon \Delta) - f(\boldsymbol{\theta} - \epsilon \Delta)}{2\epsilon} \Delta^{-1}$$



Qiskitを使った実装例



Barren-plateau問題(不毛な台地)

ランダムにパラメータ化された回路では、回路のサイズに対して勾配が指数関数的に消失してしまう(ヒルベルト空間の次元の項が現れる)現象が起きます。



barren-plateau

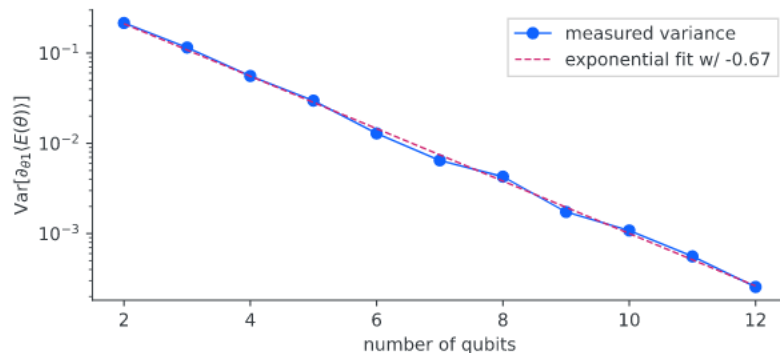
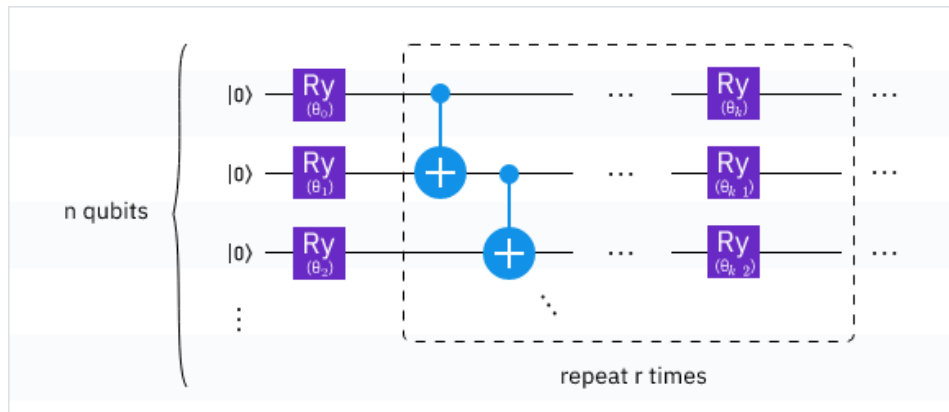
勾配が小さすぎてパラメータの更新が出来ない

※古典機械学習にも「勾配消失問題」という同様の現象が見られます。

Note: the vanishing gradient problem is dynamical when going to deeper and deeper neural networks, while the laziness is static and appears everywhere. (<https://arxiv.org/abs/2206.09313>)

Qiskitによる実装

n量子ビットのRealAmplitudes回路(繰り返しrはnと等しいとする)において、コスト関数の勾配の分散が量子ビットに対して指数関数的に減衰しています。



$$E(\theta) = \langle \psi | Z^{\otimes n} | \psi \rangle$$

Barren-plateauを回避する方法

最近の研究ではコスト関数の性質(グローバルかローカルか)によって、
勾配の減衰速度が遅くなる(Barren-plateauが起きない)ことが確認されています。

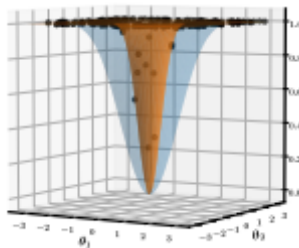
例: ユニタリ行列 $U(\theta) = \bigotimes_j e^{-i\theta_j X_j/2}$ で表される量子回路の場合

グローバル演算子 $O_G = \mathbf{1} - |0\rangle\langle 0|$

↓

コスト関数: $C_G = 1 - \prod_j \cos^2 \frac{\theta_j}{2}$

勾配の分散: $Var \left[\frac{\partial C_G}{\partial \theta_j} \right] = \frac{1}{8} \left(\frac{3}{8} \right)^{n-1}$

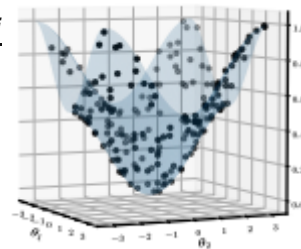


ローカル演算子 $O_L = \mathbf{1} - \frac{1}{n} \sum_j |0\rangle\langle 0|_j \otimes \mathbf{1}_j$

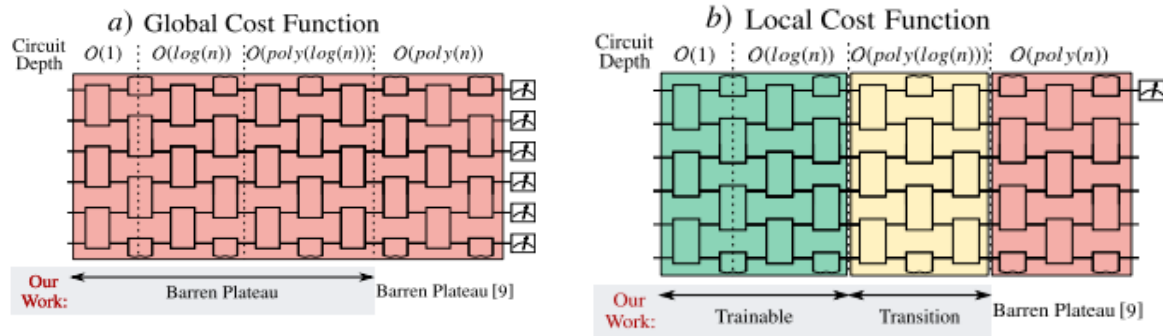
↓

コスト関数: $C_L = 1 - \frac{1}{n} \sum_j \cos^2 \frac{\theta_j}{2}$

勾配の分散: $Var \left[\frac{\partial C_L}{\partial \theta_j} \right] = \frac{1}{8n^2}$



Barren-plateauを回避する方法



	グローバル コスト関数	ローカル コスト関数
繰り返し数 (小)	× (①)	○ (④)
繰り返し数 (線形)	× (②)	× (③)

