Quantum Differentiable Programming with PennyLane

XANADU

Module:
Analytic quantum gradients

Schuld, Maria, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. "Evaluating analytic gradients on quantum hardware." *Physical Review A* 99, no. 3 (2019): 032331.

XANADU

Try-it-yourself: Quantum gradient

Given the quantum model

$$f(\theta) = \langle\psi(\theta)|\sigma_z|\psi(\theta)\rangle$$

with

$$|\psi(\theta)\rangle = R_x(\theta)|0\rangle, \quad R_x(\theta) = e^{-i\theta\sigma_x}$$

and

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

How can we estimate $\frac{\partial f}{\partial \theta}$ with a quantum computer?

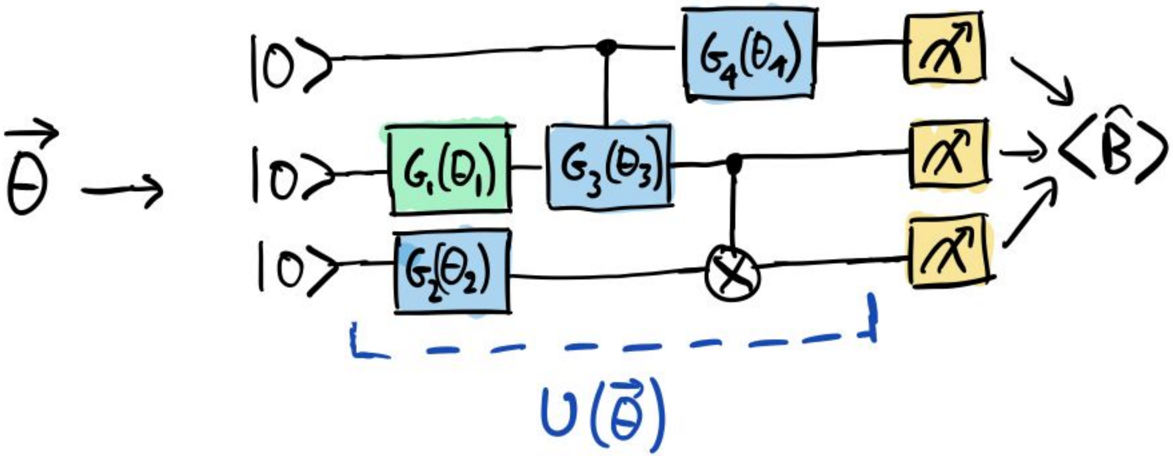
Try-it-yourself: Quantum gradient

$$\frac{\partial f}{\partial \theta} = \frac{\partial}{\partial \theta} \langle 0 | e^{i\theta\sigma_x} \sigma_z e^{-i\theta\sigma_x} | 0 \rangle$$

$$= \langle 0 | (i\sigma_x) e^{i\theta\sigma_x} \sigma_z e^{-i\theta\sigma_x} | 0 \rangle - \langle 0 | e^{i\theta\sigma_x} \sigma_z (-i\sigma_x) e^{-i\theta\sigma_x} | 0 \rangle$$

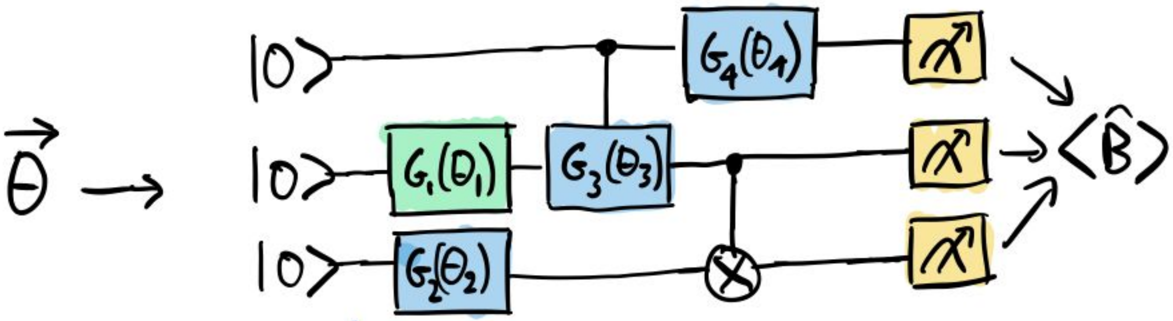
$$= i \left(\langle 0 | \sigma_x e^{i\theta\sigma_x} \sigma_z e^{-i\theta\sigma_x} | 0 \rangle + \langle 0 | e^{i\theta\sigma_x} \sigma_z \sigma_x e^{-i\theta\sigma_x} | 0 \rangle \right)$$

Analytic quantum gradients



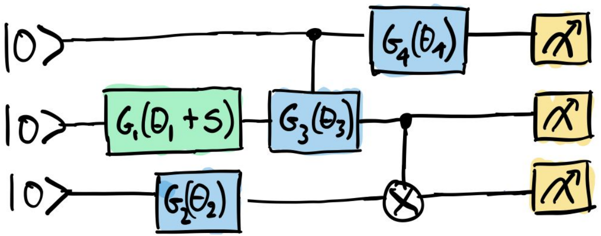
$$f(\vec{\theta}) = \langle 0 | U^\dagger(\vec{\theta}) \hat{B} U(\theta) | 0 \rangle$$

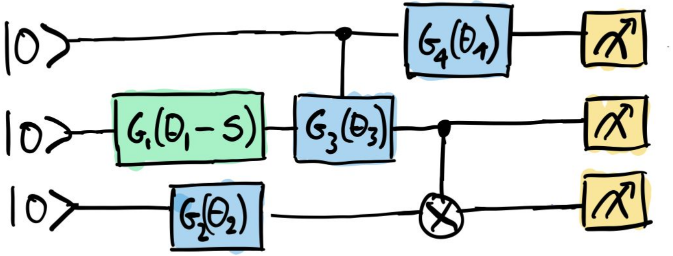
Analytic quantum gradients



$$\frac{\partial f}{\partial \theta_i} = \frac{1}{2\sin(s)}\left[f(\theta_i + s) - f(\theta_i - s)\right]$$

Analytic quantum gradients





$$\frac{\partial f}{\partial \theta_i} = \frac{1}{2\sin(s)}\left[f(\theta_i + s) - f(\theta_i - s)\right]$$

This is not finite difference!

$$\partial_\theta f(\theta) = c[f(\theta + s) - f(\theta - s)]$$

- Exact
- Shift is specific to each gate – in general, we use a **large** shift

$$\partial_\theta f(\theta) = \frac{f(\theta + \Delta\theta) - f(\theta - \Delta\theta)}{2\Delta\theta}$$

- Only an **approximation**
- Requires that shift is small
- Known to give rise to numerical issues
- For near-term devices, small shifts could lead to the resulting difference being swamped by noise

The parameter-shift rule

$$f(\theta) = \sin\theta \Rightarrow \partial_\theta f(\theta) = \cos\theta$$

$$\cos\theta = \frac{\sin(\theta+\pi/4)-\sin(\theta-\pi/4)}{\sqrt{2}}$$

$$\partial_\theta f = \frac{1}{\sqrt{2}}[f(\theta+\pi/4) - f(\theta-\pi/4)]$$

The parameter-shift rule

$$U(\theta) = e^{-i\,K_i\theta/2} \quad \text{where} \quad K_i^2 = I$$

$$U(\theta) = e^{-i\,K_i\frac{\theta}{2}} = I\cos\left(\frac{\theta}{2}\right) - iK\sin\left(\frac{\theta}{2}\right) \quad \Rightarrow \quad \nabla_\theta U(\theta) = -\frac{i}{2}KU(\theta) = -\frac{i}{2}U(\theta)K$$

$$f(\theta) = \langle\psi|U^\dagger(\theta)BU(\theta)|\psi\rangle \quad \Rightarrow \quad \nabla_\theta f(\theta) = \frac{i}{2}\langle\psi|U^\dagger(\theta)[K,B]U(\theta)|\psi\rangle$$

Substitute in $[K,B] = -\frac{i}{\sin s}\left[U^\dagger(s)KU(s) - U^\dagger(-s)KU(-s)\right]$

$$\boxed{\nabla_\theta f(\theta) = \frac{1}{2\sin s}[f(\theta+s) - f(\theta-s)], \qquad s \in [0,2\pi)}$$

The parameter-shift rule

$$U(\theta) = e^{-iK_i\theta/2} \quad \text{where} \quad K_i^2 \neq I$$

Controlled rotation operations: $CR = e^{-i(I\otimes K_i)\theta/2} = \begin{bmatrix} I & 0 \\ 0 & R_i(\theta) \end{bmatrix}$

$$\nabla_\theta f(\theta) = c_1[f(\theta+\alpha) - f(\theta-\alpha)] - c_2[f(\theta+\beta) - f(\theta-\beta)]$$

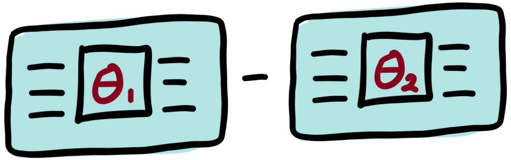
where $c_1 \sin\frac{\alpha}{2} - c_2 \sin\frac{\beta}{2} = \frac{1}{4}$ and $c_1 \sin(\alpha) - c_2 \sin\beta = \frac{1}{2}$

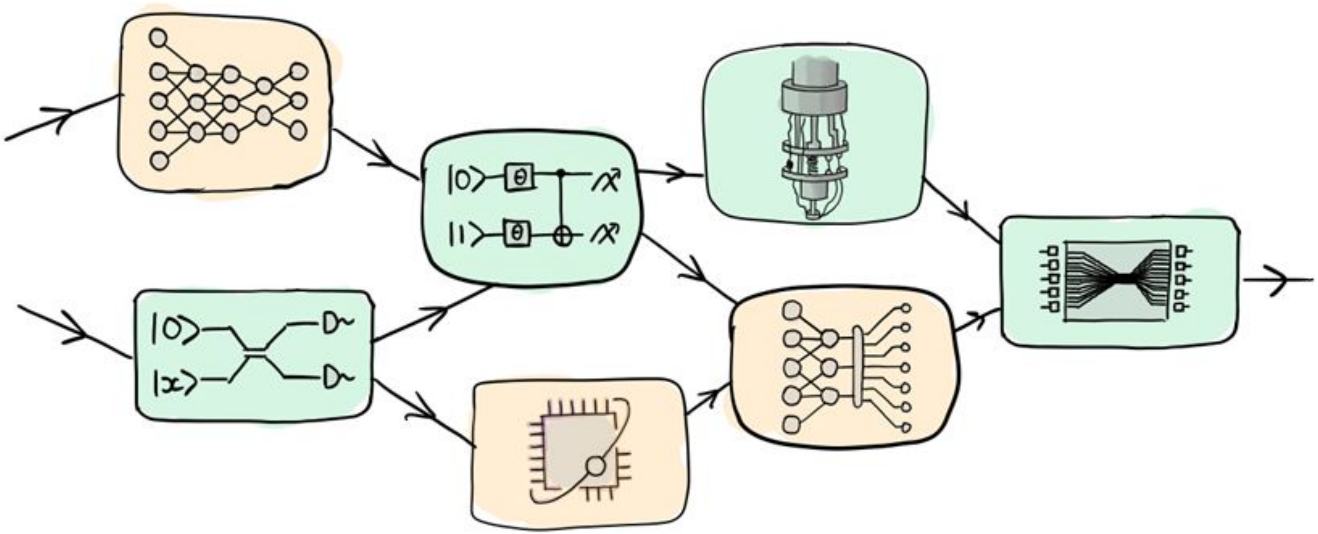
The parameter-shift rule

The parameter-shift rule has been explored and extended in-depth since first detailed

- Quantum hardware benchmarking
  https://arxiv.org/abs/2008.06517
- Extended to higher derivatives
  https://arxiv.org/abs/2008.06517
- Extended to additional gates, including noisy channels
  https://arxiv.org/abs/2005.10299
  https://johannesjakobmeyer.com/blog/004-noisy-parameter-shift/
  https://pennylane.ai/qml/demos/tutorial_stochastic_parameter_shift.html
- Convergence proofs and criteria
  https://arxiv.org/abs/1910.01155
  https://pennylane.ai/qml/demos/tutorial_doubly_stochastic.html

$$\nabla_\theta f = f(\theta_1) - f(\theta_2)$$



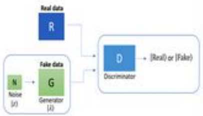


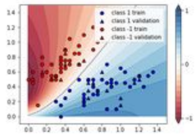
PENNYLANE

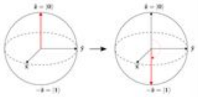
pennylane.ai/qml @pennylaneai


State preparation with Rigetti Forest + PyTorch

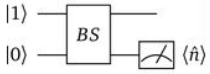

3-qubit Ising model in PyTorch

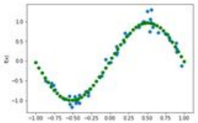

Quantum Generative Adversarial Networks with Cirq + TensorFlow

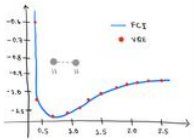

Variational classifier

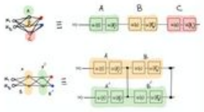

Basic tutorial: qubit rotation

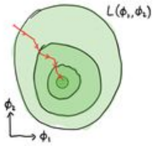

Gaussian transformation

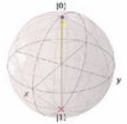

Plugins and Hybrid computation

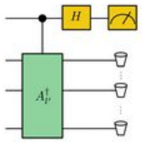

Function fitting with a quantum neural network

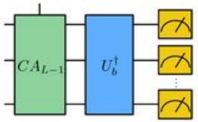

A brief overview of VQE

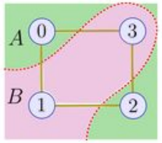

Data-reuploading classifer

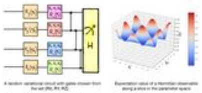

Quantum natural gradient

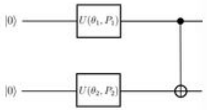

PyTorch and noisy devices

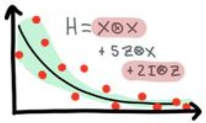

Variational Quantum Linear Solver


Coherent Variational Quantum Linear Solver


QAOA for MaxCut

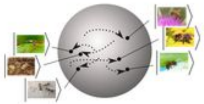

Barren plateaus in quantum neural networks


Quantum circuit structure learning

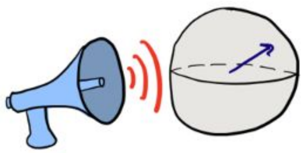

Doubly stochastic gradient descent


Advanced Usage


Quantum transfer learning

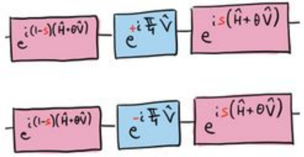

Quantum embeddings and metric learning

Check out the demos!



Optimizing noisy circuits with Cirq

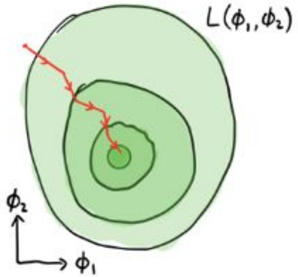
Quantum gradients with backpropagation



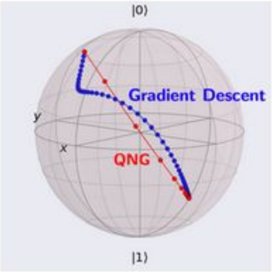
The Stochastic Parameter-Shift Rule

pennylane.ai/qml **@pennylaneai**

Check out the demos!



Quantum natural gradient



Accelerating VQEs with quantum natural gradient

pennylane.ai/qml **@pennylaneai**

Thank you!

pennylane.ai/qml
@pennylaneai
@3rdquantization

josh@xanadu.ai

XANADU

www.xanadu.ai