Variational Quantum Eigensolvers (Part 2)

Applied Quantum Algorithms

Casper Gyurik

Leiden Institute of Advanced Computer Science

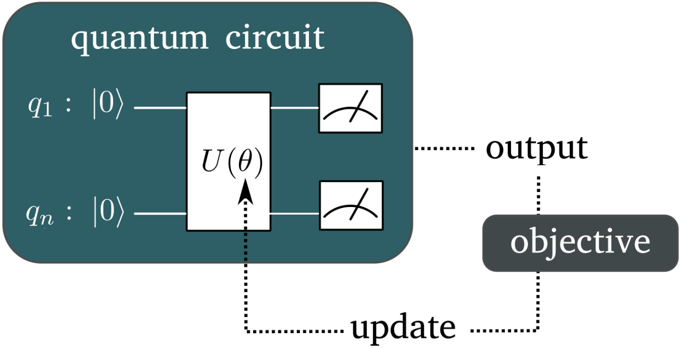
February 21, 2020

Universiteit Leiden
The Netherlands

Table of contents

Variational Quantum Algorithms

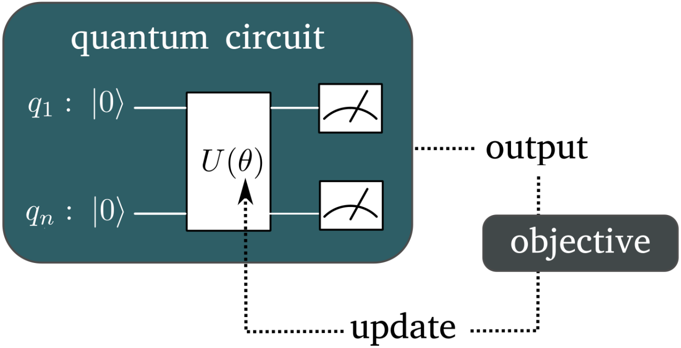
Recap: an overview



Source: arXiv:1811.04968

Variational Quantum Algorithms

Recap: an overview

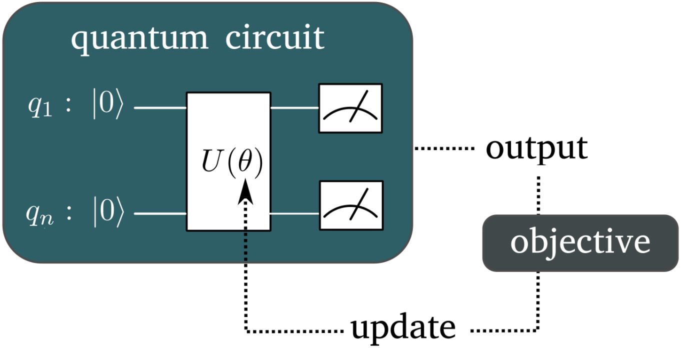


Source: arXiv:1811.04968

- Current step: extracting (usefull) classical information.

Variational Quantum Algorithms

Recap: an overview



Source: arXiv:1811.04968

- Current step: extracting (usefull) classical information.
- Next step: update parameters of the quantum circuit.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.
- Mathematically speaking, observables are Hermitian matrices ($O^\dagger = O$).

Pauli strings and the real vector space Herm$\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.
- Mathematically speaking, observables are Hermitian matrices ($O^\dagger = O$).
- Set of N-qubit Hermitian matrices denoted Herm$(\mathbb{C}^{2^N})$.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.
- Mathematically speaking, observables are Hermitian matrices ($O^\dagger = O$).
- Set of N-qubit Hermitian matrices denoted Herm($\mathbb{C}^{2^N}$).

Let us look at the structure of Herm($\mathbb{C}^{2^N}$).

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.
- Mathematically speaking, observables are Hermitian matrices ($O^\dagger = O$).
- Set of N-qubit Hermitian matrices denoted Herm($\mathbb{C}^{2^N}$).

Let us look at the structure of Herm($\mathbb{C}^{2^N}$).

- Herm($\mathbb{C}^{2^N}$) is a real vector space with inner product

$$< A, B >:= \mathrm{Tr}(AB^\dagger) \qquad \text{(Hilbert-Schmidt inner product).}$$

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.
- Mathematically speaking, observables are Hermitian matrices ($O^\dagger = O$).
- Set of N-qubit Hermitian matrices denoted Herm($\mathbb{C}^{2^N}$).

Let us look at the structure of Herm($\mathbb{C}^{2^N}$).

- Herm($\mathbb{C}^{2^N}$) is a real vector space with inner product

$$< A, B >:= \mathrm{Tr}(AB^\dagger) \qquad \text{(Hilbert-Schmidt inner product).}$$

- Interesting subset are the Pauli strings $\{I, X, Y, Z\}^{\otimes N}$.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

How can we extract classical information?

- We can measure observables O.
- Mathematically speaking, observables are Hermitian matrices ($O^\dagger = O$).
- Set of N-qubit Hermitian matrices denoted Herm($\mathbb{C}^{2^N}$).

Let us look at the structure of Herm($\mathbb{C}^{2^N}$).

- Herm($\mathbb{C}^{2^N}$) is a real vector space with inner product

$$< A, B > := \mathrm{Tr}(AB^\dagger) \qquad \text{(Hilbert-Schmidt inner product).}$$

- Interesting subset are the Pauli strings $\{I, X, Y, Z\}^{\otimes N}$.

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ are a basis for the real vector space Herm($\mathbb{C}^{2^N}$).

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ are a basis for the real vector space Herm$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{C}}(\mathsf{Mat}_{2^N \times 2^N}(\mathbb{C})) = 4^N$.

□

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ *are a basis for the real vector space Herm*$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{C}}(\mathsf{Mat}_{2^N \times 2^N}(\mathbb{C})) = 4^N$.
- $\dim_{\mathbb{R}}(\mathbb{C}) = 2$.

□

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ *are a basis for the real vector space Herm*$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{C}}(\mathsf{Mat}_{2^N \times 2^N}(\mathbb{C})) = 4^N$.
- $\dim_{\mathbb{R}}(\mathbb{C}) = 2$.
- $\implies \dim_{\mathbb{R}}(\mathsf{Mat}_{2^N \times 2^N}(\mathbb{C})) = 2 \cdot 4^N$.

□

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^N}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ *are a basis for the real vector space Herm*$(\mathbb{C}^{2^N})$*.*

Proof.

- $\dim_{\mathbb{C}}(\mathsf{Mat}_{2^N \times 2^N}(\mathbb{C})) = 4^N$.
- $\dim_{\mathbb{R}}(\mathbb{C}) = 2$.
- $\implies \dim_{\mathbb{R}}(\mathsf{Mat}_{2^N \times 2^N}(\mathbb{C})) = 2 \cdot 4^N$.
- $\left(A \in \mathsf{Herm}(\mathbb{C}^{2^N}) \iff A^\dagger = A\right) \Rightarrow \dim_{\mathbb{R}}(\mathsf{Herm}(\mathbb{C}^{2^N}) = 2 \cdot 4^N/2 = 4^N$.

□

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^n}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ *are a basis for the real vector space Herm*$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{R}}(\mathsf{Herm}(\mathbb{C}^{2^N}) = 4^N$.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^n}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ are a basis for the real vector space Herm$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{R}}(\text{Herm}(\mathbb{C}^{2^N}) = 4^N$.
- $\#\{I, X, Y, Z\}^{\otimes N} = 4^N$.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^n}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ are a basis for the real vector space Herm$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{R}}(\text{Herm}(\mathbb{C}^{2^N}) = 4^N$.
- $\#\{I, X, Y, Z\}^{\otimes N} = 4^N$.
- Pauli strings are l.i., since orthogonal w.r.t. H-S inner product.

Pauli strings and the real vector space Herm $\left(\mathbb{C}^{2^n}\right)$

an intermezzo

Lemma

Pauli strings $\{I, X, Y, Z\}^{\otimes N}$ are a basis for the real vector space Herm$(\mathbb{C}^{2^N})$.

Proof.

- $\dim_{\mathbb{R}}(\text{Herm}(\mathbb{C}^{2^N}) = 4^N$.
- $\#\{I, X, Y, Z\}^{\otimes N} = 4^N$.
- Pauli strings are l.i., since orthogonal w.r.t. H-S inner product.

$\implies$ $\{I, X, Y, Z\}^{\otimes N}$ is a basis for the real vector space Herm$(\mathbb{C}^{2^N})$ □

A standard VQE cost function

what to optimize?

A standard VQE cost function is of the form

$$f_O(\theta) = \langle\psi(\theta)|\, O \,|\psi(\theta)\rangle\,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

A standard VQE cost function

what to optimize?

A standard VQE cost function is of the form

$$f_O(\theta) = \langle\psi(\theta)|\, O\, |\psi(\theta)\rangle\,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let us go over some examples.

A standard VQE cost function

what to optimize?

A standard VQE cost function is of the form

$$f_O(\theta) = \langle\psi(\theta)|\, O \,|\psi(\theta)\rangle\,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let us go over some examples.

Estimating the ground state energy of a Hamiltonian

Physics interested in computing the ground state energy of a physical system.

A standard VQE cost function

what to optimize?

A standard VQE cost function is of the form

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let us go over some examples.

Estimating the ground state energy of a Hamiltonian

Physics interested in computing the ground state energy of a physical system.
Mathematically speaking: compute the smallest eigenvalue of $H \in \mathsf{Herm}(\mathbb{C}^{2^N})$.

A standard VQE cost function

what to optimize?

A standard VQE cost function is of the form

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let us go over some examples.

Estimating the ground state energy of a Hamiltonian

Physics interested in computing the ground state energy of a physical system.
Mathematically speaking: compute the smallest eigenvalue of $H \in \mathsf{Herm}(\mathbb{C}^{2^N})$.
The corresponding cost function is simply

$$f_H(\theta) = \langle\psi(\theta)| H |\psi(\theta)\rangle,$$

because we know that $\lambda_0(H) = \min_{|\phi\rangle} \langle\phi| H |\phi\rangle$.

VQE cost functions examples

Approximating the MaxCut of a graph

Let $G = (V, E)$ be a graph.

- The MaxCut of G is the maximal size of a cut-set of a cut of G.

VQE cost functions examples

Approximating the MaxCut of a graph

Let $G = (V, E)$ be a graph.

- The MaxCut of G is the maximal size of a cut-set of a cut of G.
- A cut is a partition of V into two disjoint subsets (S, T).

VQE cost functions examples

Approximating the MaxCut of a graph

Let $G = (V, E)$ be a graph.

- The MaxCut of G is the maximal size of a cut-set of a cut of G.
- A cut is a partition of V into two disjoint subsets (S, T).
- The cut-set of a cut is the set of edges "crossing the cut", i.e.,

$$\text{cut-set of } (S,T) \; = \{(s,t) \in E | s \in S \text{ and } t \in T\}.$$

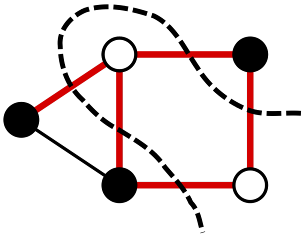


Figure: An example of a MaxCut.

VQE cost functions examples

Approximating the MaxCut of a graph

Turns out: MaxCut(G) can be written as $\lambda_0(O)$ for some $O \in \mathrm{Herm}(\mathbb{C}^{2^N})$.

VQE cost functions examples

Approximating the MaxCut of a graph

Turns out: MaxCut(G) can be written as $\lambda_0(O)$ for some $O \in$ Herm$(\mathbb{C}^{2^N})$.

- That is, let $V = \{1, \ldots, N\}$ and consider the Hermitian operator

$$O_{cut} = \frac{1}{2} \sum_{(i,j) \in E} I + Z_i Z_j.$$

VQE cost functions examples

Approximating the MaxCut of a graph

Turns out: MaxCut(G) can be written as $\lambda_0(O)$ for some $O \in$ Herm$(\mathbb{C}^{2^N})$.

- That is, let $V = \{1, \ldots, N\}$ and consider the Hermitian operator

$$O_{cut} = \frac{1}{2} \sum_{(i,j) \in E} I + Z_i Z_j.$$

- Then it turns out that: $\lambda_0(O_{cut}) = |E| - \text{MaxCut}(G)$.

VQE cost functions examples

Approximating the MaxCut of a graph

Turns out: MaxCut(G) can be written as $\lambda_0(O)$ for some $O \in$ Herm$(\mathbb{C}^{2^N})$.

- That is, let $V = \{1, \ldots, N\}$ and consider the Hermitian operator

$$O_{cut} = \frac{1}{2} \sum_{(i,j) \in E} I + Z_i Z_j.$$

- Then it turns out that: $\lambda_0(O_{cut}) = |E| -$ MaxCut(G).

Thus, to approximate MaxCut(G) we use the cost function

$$f_{O_{cut}}(\theta) = \langle \psi(\theta) | O_{cut} | \psi(\theta) \rangle .$$

VQE cost functions examples

Generative modeling, i.e., learning a distribution

Generative modeling: branch of ML that tries to learn a distribution p.

- That is, based on old samples from p, train your computer to generate new samples from a distribution that is close to p.

VQE cost functions examples

Generative modeling, i.e., learning a distribution

Generative modeling: branch of ML that tries to learn a distribution p.

- That is, based on old samples from p, train your computer to generate new samples from a distribution that is close to p.

We can tackle this using a variational quantum algorithm.

VQE cost functions examples

Generative modeling, i.e., learning a distribution

Generative modeling: branch of ML that tries to learn a distribution p.

- That is, based on old samples from p, train your computer to generate new samples from a distribution that is close to p.

We can tackle this using a variational quantum algorithm.

- Quantum state $|\psi(\theta)\rangle$ together with observale O generate distribution q_θ.

VQE cost functions examples

Generative modeling, i.e., learning a distribution

Generative modeling: branch of ML that tries to learn a distribution p.

- That is, based on old samples from p, train your computer to generate new samples from a distribution that is close to p.

We can tackle this using a variational quantum algorithm.

- Quantum state $|\psi(\theta)\rangle$ together with observale O generate distribution q_θ.
- **Goal:** Find θ such that q_θ is close to p.

VQE cost functions examples

Generative modeling, i.e., learning a distribution

Generative modeling: branch of ML that tries to learn a distribution p.

- That is, based on old samples from p, train your computer to generate new samples from a distribution that is close to p.

We can tackle this using a variational quantum algorithm.

- Quantum state $|\psi(\theta)\rangle$ together with observale O generate distribution q_θ.
- **Goal:** Find θ such that q_θ is close to p.

KL divergence: measure how one distribution is different from a second

$$f_{D_{KL}}(\vec{\theta}) = D_{KL}(p, q_\theta) = \sum_x p(x) \log\left(\frac{p(x)}{q_\theta(x)}\right).$$

Evaluating our cost function

We only care about Pauli strings

Let us investigate what kind of observables we have to measure.

Evaluating our cost function

We only care about Pauli strings

Let us investigate what kind of observables we have to measure.

- The goal is to esimate the VQE cost function:

$$f_O(\theta) = \langle \psi(\theta) | \, O \, | \psi(\theta) \rangle \,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Evaluating our cost function

We only care about Pauli strings

Let us investigate what kind of observables we have to measure.

- The goal is to esimate the VQE cost function:

$$f_O(\theta) = \langle\psi(\theta)|\, O\, |\psi(\theta)\rangle\,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

- From the Lemma at the beginning, we learn that we can decompose

$$O = \sum_{i=1}^{4^N} h_i P_i,$$

where $P_i \in \{I, X, Y, Z\}^{\otimes N}$ and $h_i \in \mathbb{R}$.

Evaluating our cost function

We only care about Pauli strings

Let us investigate what kind of observables we have to measure.

- The goal is to esimate the VQE cost function:

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

- From the Lemma at the beginning, we learn that we can decompose

$$O = \sum_{i=1}^{4^N} h_i P_i,$$

where $P_i \in \{I, X, Y, Z\}^{\otimes N}$ and $h_i \in \mathbb{R}$.

- By linearity we find that

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle = \langle\psi(\theta)| \sum_{i=1}^{4^N} h_i P_i |\psi(\theta)\rangle = \sum_{i=1}^{4^N} h_i \langle\psi(\theta)| P_i |\psi(\theta)\rangle.$$

Evaluating our cost function

We only care about Pauli strings

Let us investigate what kind of observables we have to measure.

- The goal is to esimate the VQE cost function:

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle , \quad O \in \mathrm{Herm}(\mathbb{C}^{2^N}).$$

- From the Lemma at the beginning, we learn that we can decompose

$$O = \sum_{i=1}^{4^N} h_i P_i,$$

  where $P_i \in \{I, X, Y, Z\}^{\otimes N}$ and $h_i \in \mathbb{R}$.

- By linearity we find that

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle = \langle\psi(\theta)| \sum_{i=1}^{4^N} h_i P_i |\psi(\theta)\rangle = \sum_{i=1}^{4^N} h_i \langle\psi(\theta)| P_i |\psi(\theta)\rangle .$$

- Thus, we only have to estimate $\langle\psi(\theta)| P_i |\psi(\theta)\rangle$ for the Pauli strings.

Evaluating our cost function

We only care about Pauli strings

Let us investigate what kind of observables we have to measure.

- The goal is to esimate the VQE cost function:

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

- From the Lemma at the beginning, we learn that we can decompose

$$O = \sum_{i=1}^{4^N} h_i P_i,$$

  where $P_i \in \{I, X, Y, Z\}^{\otimes N}$ and $h_i \in \mathbb{R}$.

- By linearity we find that

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle = \langle\psi(\theta)| \sum_{i=1}^{4^N} h_i P_i |\psi(\theta)\rangle = \sum_{i=1}^{4^N} h_i \langle\psi(\theta)| P_i |\psi(\theta)\rangle.$$

- Thus, we only have to estimate $\langle\psi(\theta)| P_i |\psi(\theta)\rangle$ for the Pauli strings.

Remark: for relevant problems usually only poly-many strings.

- Area of research to bring this down further (e.g., using commuting strings).

Evaluating our cost function

We are going to have to estimate

How do we actually evaluate our cost function?

$$f_O(\theta) = \langle\psi(\theta)|\, O \,|\psi(\theta)\rangle\,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Evaluating our cost function

We are going to have to estimate

How do we actually evaluate our cost function?

$$f_O(\theta) = \langle\psi(\theta)| O |\psi(\theta)\rangle, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let $O = \sum_{i=1}^{2^N} \lambda_i |\varphi_i\rangle\langle\varphi_i|$ denote the spectral decomposition.

Evaluating our cost function

We are going to have to estimate

How do we actually evaluate our cost function?

$$f_O(\theta) = \langle\psi(\theta)|\, O\, |\psi(\theta)\rangle\,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let $O = \sum_{i=1}^{2^N} \lambda_i \,|\varphi_i\rangle\langle\varphi_i|$ denote the spectral decomposition.

- Note that $f_O(\theta) = \mathbb{E}[X]$, where $X \in_R \{\lambda_i\}$ with

$$\mathbb{P}(X = \lambda_i) = |\,\langle\varphi_i \mid \psi(\theta)\rangle\,|^2$$

Evaluating our cost function

We are going to have to estimate

How do we actually evaluate our cost function?

$$f_O(\theta) = \langle\psi(\theta)|\, O \,|\psi(\theta)\rangle\,, \quad O \in \mathrm{Herm}(\mathbb{C}^{2^N}).$$

Let $O = \sum_{i=1}^{2^N} \lambda_i \,|\varphi_i\rangle\,\langle\varphi_i|$ denote the spectral decomposition.

- Note that $f_O(\theta) = \mathbb{E}[X]$, where $X \in_R \{\lambda_i\}$ with

$$\mathbb{P}(X = \lambda_i) = |\,\langle\varphi_i \mid \psi(\theta)\rangle\,|^2$$

- As we can only sample from X, the best we can do is estimate $\mathbb{E}[X]$.

Evaluating our cost function

We are going to have to estimate

How do we actually evaluate our cost function?

$$f_O(\theta) = \langle \psi(\theta)| \, O \, |\psi(\theta)\rangle \,, \quad O \in \mathsf{Herm}(\mathbb{C}^{2^N}).$$

Let $O = \sum_{i=1}^{2^N} \lambda_i \, |\varphi_i\rangle \langle \varphi_i|$ denote the spectral decomposition.

- Note that $f_O(\theta) = \mathbb{E}[X]$, where $X \in_R \{\lambda_i\}$ with

$$\mathbb{P}(X = \lambda_i) = |\, \langle \varphi_i \mid \psi(\theta) \rangle \,|^2$$

- As we can only sample from X, the best we can do is estimate $\mathbb{E}[X]$.
- How many samples from X do we need? Let's use Chebyshev's Inequality!

Evaluating our cost function

We are going to have to estimate

Let $O = \sum_{i=1}^{2^N} \lambda_i \left|\varphi_i\right\rangle \left\langle\varphi_i\right|$ denote the spectral decomposition.

Goal: Estimate $E = \mathbb{E}[X]$ to within additive precision ϵ, where $X \in_R \{\lambda_i\}$ whose probabilities are given by $\mathbb{P}(X = \lambda_i) = |\left\langle\varphi_i \mid \psi(\theta)\right\rangle|^2$.

Evaluating our cost function

We are going to have to estimate

Let $O = \sum_{i=1}^{2^N} \lambda_i |\varphi_i\rangle \langle\varphi_i|$ denote the spectral decomposition.

Goal: Estimate $E = \mathbb{E}[X]$ to within additive precision ϵ, where $X \in_R \{\lambda_i\}$ whose probabilities are given by $\mathbb{P}(X = \lambda_i) = |\langle\varphi_i \mid \psi(\theta)\rangle|^2$.

- Take M samples, denoted $X_1, \ldots, X_M$, and we compute $\frac{1}{m} \sum_{i=1}^{M} X_i$.

Evaluating our cost function

We are going to have to estimate

Let $O = \sum_{i=1}^{2^N} \lambda_i \left|\varphi_i\right\rangle \left\langle\varphi_i\right|$ denote the spectral decomposition.

Goal: Estimate $E = \mathbb{E}[X]$ to within additive precision ϵ, where $X \in_R \{\lambda_i\}$ whose probabilities are given by $\mathbb{P}(X = \lambda_i) = |\left\langle\varphi_i \mid \psi(\theta)\right\rangle|^2$.

- Take M samples, denoted $X_1, \ldots, X_M$, and we compute $\frac{1}{m}\sum_{i=1}^{M} X_i$.
- Chebyshev's inequality:

$$\mathbb{P}\left(\left|\frac{\sum_{i=1}^{M} X_i}{M} - E\right| \geq \epsilon\right) \leq \frac{\sigma^2}{M\epsilon^2},$$

where σ denotes the variance of X.

Evaluating our cost function

We are going to have to estimate

Let $O = \sum_{i=1}^{2^N} \lambda_i |\varphi_i\rangle \langle\varphi_i|$ denote the spectral decomposition.

Goal: Estimate $E = \mathbb{E}[X]$ to within additive precision ϵ, where $X \in_R \{\lambda_i\}$ whose probabilities are given by $\mathbb{P}(X = \lambda_i) = |\langle\varphi_i \mid \psi(\theta)\rangle|^2$.

- Take M samples, denoted $X_1, \ldots, X_M$, and we compute $\frac{1}{m}\sum_{i=1}^{M} X_i$.
- Chebyshev's inequality:

$$\mathbb{P}\left(\left|\frac{\sum_{i=1}^{M} X_i}{M} - E\right| \geq \epsilon\right) \leq \frac{\sigma^2}{M\epsilon^2},$$

  where σ denotes the variance of X.
- Therefore, if we take $M = \frac{\sigma^2}{0.01\epsilon^2}$, then we find that

$$\mathbb{P}\left(\left|\frac{\sum_{i=1}^{M} X_i}{M} - E\right| \geq \epsilon\right) \leq 0.01.$$

Evaluating our cost function

We are going to have to estimate

Let $O = \sum_{i=1}^{2^N} \lambda_i \left|\varphi_i\right\rangle \left\langle\varphi_i\right|$ denote the spectral decomposition.

Goal: Estimate $E = \mathbb{E}[X]$ to within additive precision ϵ, where $X \in_R \{\lambda_i\}$ whose probabilities are given by $\mathbb{P}(X = \lambda_i) = |\left\langle\varphi_i \mid \psi(\theta)\right\rangle|^2$.

- Take M samples, denoted $X_1, \ldots, X_M$, and we compute $\frac{1}{m}\sum_{i=1}^{M} X_i$.
- Chebyshev's inequality:

$$\mathbb{P}\left(\left|\frac{\sum_{i=1}^{M} X_i}{M} - E\right| \geq \epsilon\right) \leq \frac{\sigma^2}{M\epsilon^2},$$

  where σ denotes the variance of X.
- Therefore, if we take $M = \frac{\sigma^2}{0.01\epsilon^2}$, then we find that

$$\mathbb{P}\left(\left|\frac{\sum_{i=1}^{M} X_i}{M} - E\right| \geq \epsilon\right) \leq 0.01.$$

- Variance is usually bounded, thus we need to do M measurements where

$$M \sim \frac{1}{\epsilon^2}.$$

Optimizing the VQE cost function

finding the right parameters

Now that we know how to evaluate our cost function, let us try to optimize it.

Optimizing the VQE cost function

finding the right parameters

Now that we know how to evaluate our cost function, let us try to optimize it.

To this end, we will employ classical optimization routines.

Optimizing the VQE cost function

finding the right parameters

Now that we know how to evaluate our cost function, let us try to optimize it.

To this end, we will employ classical optimization routines.

Broadly speaking, these optimization routines fall into two categories.

1. Gradient-based optimization.
2. Gradient-free optimization.

Optimizing the VQE cost function

finding the right parameters

Now that we know how to evaluate our cost function, let us try to optimize it.

To this end, we will employ classical optimization routines.

Broadly speaking, these optimization routines fall into two categories.

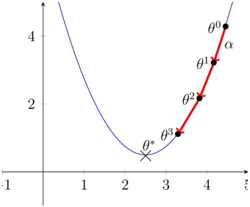
1. Gradient-based optimization.
2. Gradient-free optimization.

Let us go over them both, including their pros and cons.

Optimizing the VQE cost function

gradient-based methods

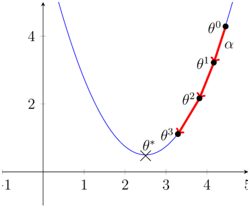
As the name suggests, gradient-based methods employ the gradient of the cost.



Optimizing the VQE cost function

gradient-based methods

As the name suggests, gradient-based methods employ the gradient of the cost.



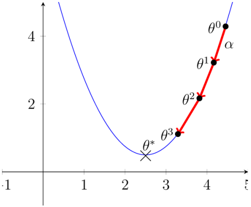
Pros:

- Works incredibly well when your cost function landscape is nicely smooth.
- Convergence properties are very well established.

Optimizing the VQE cost function

gradient-based methods

As the name suggests, gradient-based methods employ the gradient of the cost.



Pros:

- Works incredibly well when your cost function landscape is nicely smooth.
- Convergence properties are very well established.

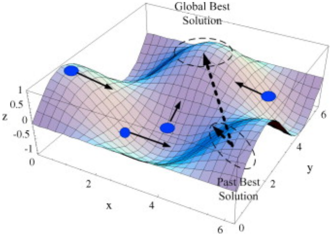
Cons:

- Unreliable under noisy gradient evaluations.
- Suffers from vanishing and exploding gradients (i.e., when your cost function landscape is barren or rigid).
- Can take very long if gradient evaluation is expensive.

Optimizing the VQE cost function

gradient-based methods

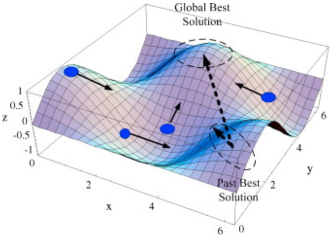
As the name suggests, gradient-free methods don't use the gradient of the cost. They usually rely on a large number of function evaluations.



Optimizing the VQE cost function

gradient-based methods

As the name suggests, gradient-free methods don't use the gradient of the cost. They usually rely on a large number of function evaluations.



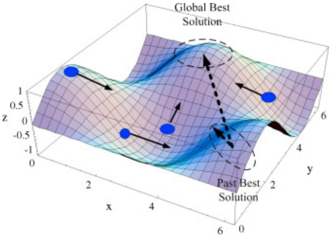
Pros:

- Works decently well even when your landscape is barren or rigid.
- Does not require the ability to (efficiently) evaluate the gradient.

Optimizing the VQE cost function

gradient-based methods

As the name suggests, gradient-free methods don't use the gradient of the cost. They usually rely on a large number of function evaluations.



Pros:

- Works decently well even when your landscape is barren or rigid.
- Does not require the ability to (efficiently) evaluate the gradient.

Cons:

- Does not converge as quick as gradient-based methods when the landscape is smooth.
- Requires a lot of function evaluations in general.

Estimating the gradient of the VQE cost function

To use gradient-based methods, we need gradients

How do we estimate the gradient of $f_O(\theta)$? Here are two possible ways.

Estimating the gradient of the VQE cost function

To use gradient-based methods, we need gradients

How do we estimate the gradient of $f_O(\theta)$? Here are two possible ways.

- The "parameter-shift rule", which states that for a large family of ansatzes

$$\frac{\partial f_O(\theta)}{\partial \theta_j} = \frac{f_O(\vec{\theta} + \frac{\pi}{2} e_j) + f_O(\vec{\theta} - \frac{\pi}{2} e_j)}{2},$$

  where e_j is the j-th standard basis vector.

Estimating the gradient of the VQE cost function

To use gradient-based methods, we need gradients

How do we estimate the gradient of $f_O(\theta)$? Here are two possible ways.

- The "parameter-shift rule", which states that for a large family of ansatzes

$$\frac{\partial f_O(\theta)}{\partial \theta_j} = \frac{f_O(\vec{\theta} + \frac{\pi}{2} e_j) + f_O(\vec{\theta} - \frac{\pi}{2} e_j)}{2},$$

  where e_j is the j-th standard basis vector.
  Pros: Exact formula for the gradient.
  Cons: Requires $2N_{\text{param}}$ evaluations of f_O.

Estimating the gradient of the VQE cost function

To use gradient-based methods, we need gradients

How do we estimate the gradient of $f_O(\theta)$? Here are two possible ways.

- The "parameter-shift rule", which states that for a large family of ansatzes

$$\frac{\partial f_O(\theta)}{\partial \theta_j} = \frac{f_O(\vec{\theta} + \frac{\pi}{2} e_j) + f_O(\vec{\theta} - \frac{\pi}{2} e_j)}{2},$$

  where e_j is the j-th standard basis vector.
  Pros: Exact formula for the gradient.
  Cons: Requires $2N_{\text{param}}$ evaluations of f_O.

- Stochastic pertubation techniques that use the approximation

$$\frac{\partial f_O(\theta)}{\partial \theta} \approx \frac{f_O(\theta + c\Delta) + f_O(\theta - c\Delta)}{2c\Delta_j},$$

  where $\Delta \in_R \{- \pm 1\}^{N_{\text{param}}}$ Rademacher random and $c > 0$ very small.

Estimating the gradient of the VQE cost function

To use gradient-based methods, we need gradients

How do we estimate the gradient of $f_O(\theta)$? Here are two possible ways.

- The "parameter-shift rule", which states that for a large family of ansatzes

$$\frac{\partial f_O(\theta)}{\partial \theta_j} = \frac{f_O(\vec{\theta} + \frac{\pi}{2} e_j) + f_O(\vec{\theta} - \frac{\pi}{2} e_j)}{2},$$

  where e_j is the j-th standard basis vector.
  Pros: Exact formula for the gradient.
  Cons: Requires $2N_{\text{param}}$ evaluations of f_O.

- Stochastic pertubation techniques that use the approximation

$$\frac{\partial f_O(\theta)}{\partial \theta} \approx \frac{f_O(\theta + c\Delta) + f_O(\theta - c\Delta)}{2c\Delta_j},$$

  where $\Delta \in_R \{- \pm 1\}^{N_{\text{param}}}$ Rademacher random and $c > 0$ very small.
  Pros: Only requires 2 evaluations of f_O.
  Cons: At best a finite difference *approximation* of the gradient.

Estimating the gradient of the VQE cost function

To use gradient-based methods, we need gradients

How do we estimate the gradient of $f_O(\theta)$? Here are two possible ways.

- The "parameter-shift rule", which states that for a large family of ansatzes

$$\frac{\partial f_O(\theta)}{\partial \theta_j} = \frac{f_O(\vec{\theta} + \frac{\pi}{2} e_j) + f_O(\vec{\theta} - \frac{\pi}{2} e_j)}{2},$$

  where e_j is the j-th standard basis vector.
  Pros: Exact formula for the gradient.
  Cons: Requires $2N_{\text{param}}$ evaluations of f_O.

- Stochastic pertubation techniques that use the approximation

$$\frac{\partial f_O(\theta)}{\partial \theta} \approx \frac{f_O(\theta + c\Delta) + f_O(\theta - c\Delta)}{2c\Delta_j},$$

  where $\Delta \in_R \{- \pm 1\}^{N_{\text{param}}}$ Rademacher random and $c > 0$ very small.
  Pros: Only requires 2 evaluations of f_O.
  Cons: At best a finite difference *approximation* of the gradient.

- **Biggest difference:** analytic vs numerical approximation.

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

- We know that VQE cost function landscapes contain "barren plateaus".

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

- We know that VQE cost function landscapes contain "barren plateaus".

- Raises question of parameter initialization: starting at a random point will most likely cause you to end up in one if these barren plateaus.

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

- We know that VQE cost function landscapes contain “barren plateaus”.

- Raises question of parameter initialization: starting at a random point will most likely cause you to end up in one if these barren plateaus.
  - Turns out to be hard, as the structure of the landscapes are largely unknown.

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

- We know that VQE cost function landscapes contain "barren plateaus".

- Raises question of parameter initialization: starting at a random point will most likely cause you to end up in one if these barren plateaus.
  - Turns out to be hard, as the structure of the landscapes are largely unknown.
- Number/size of barren plateaus are very dependant on O and ansatz.

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

- We know that VQE cost function landscapes contain "barren plateaus".

- Raises question of parameter initialization: starting at a random point will most likely cause you to end up in one if these barren plateaus.
  - Turns out to be hard, as the structure of the landscapes are largely unknown.
- Number/size of barren plateaus are very dependant on O and ansatz.
- Experiments are a good way to get a feeling for these landscapes.

The landscape of a VQE cost function

Beware: barren plateaus

What do we know about the landscape of VQE cost functions?

- We know that VQE cost function landscapes contain “barren plateaus”.

- Raises question of parameter initialization: starting at a random point will most likely cause you to end up in one if these barren plateaus.
  - Turns out to be hard, as the structure of the landscapes are largely unknown.
- Number/size of barren plateaus are very dependant on O and ansatz.
- Experiments are a good way to get a feeling for these landscapes.
  - Many interesting student projects!

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

- As we are currently in the NISQ era, our quantum circuits will be noisy.

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

- As we are currently in the NISQ era, our quantum circuits will be noisy.
- A consequence of this is that evaluation/estimations of $f_O(\theta)$ will be noisy.

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

- As we are currently in the NISQ era, our quantum circuits will be noisy.
- A consequence of this is that evaluation/estimations of $f_O(\theta)$ will be noisy.

Variational quantum algorithms have shown to be quite resillient to this noise.

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

- As we are currently in the NISQ era, our quantum circuits will be noisy.
- A consequence of this is that evaluation/estimations of $f_O(\theta)$ will be noisy.

Variational quantum algorithms have shown to be quite resillient to this noise.

- Part of the reason: optimization routines can deal with this noise.

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

- As we are currently in the NISQ era, our quantum circuits will be noisy.
- A consequence of this is that evaluation/estimations of $f_O(\theta)$ will be noisy.

Variational quantum algorithms have shown to be quite resillient to this noise.

- Part of the reason: optimization routines can deal with this noise.

Open question: what optimization routines are best in this noisy setting?

Optimization of a noisy function

Getting back to the NISQ era

What are the challenges for the classical optimization routine in the NISQ era?

- As we are currently in the NISQ era, our quantum circuits will be noisy.
- A consequence of this is that evaluation/estimations of $f_O(\theta)$ will be noisy.

Variational quantum algorithms have shown to be quite resillient to this noise.

- Part of the reason: optimization routines can deal with this noise.

Open question: what optimization routines are best in this noisy setting?

- Many interesting student projects!