Learning Unitaries with gradient descent optimization

Reevu Maity (Oxford)

In progress with Bobak Kiani (MIT), Zi-Wen Liu (Perimeter), Seth Lloyd (MIT) & Milad Marvian(MIT)

It from Qubit 2019

June 13

Outline

- We consider classical optimization algorithms to learn / simulate parametrized unitary transformations generated by two Hamiltonians applied in alternation (QAOA).

- Gradient Descent algorithms are first order classical methods widely studied in the machine learning community for convex optimization.

- Recently, it was shown that QAOA can be used for universal quantum computation.

S. Lloyd (2018)

Any unitary can be simulated with $2d^2$ parameters via the alternating operator method.

S. Lloyd & RM (2019)

- Aim : Study the learnability / simulability of unitaries under the alternating operator / QAOA formalism with gradient descent.

Problem

QAOA Unitary : $\mathcal{U}(\vec{t},\vec{\tau}) = e^{-iB\tau_N} e^{-iAt_N} \cdots e^{-iB\tau_1} e^{-iAt_1}$

A, B are random matrices of dimension d sampled from the GUE and $2N \leq d^2$.

Learning problem

- Given access to a target unitary $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ and knowledge of A, B, can we simulate $\mathcal{U}$ by a sequence $\mathcal{V}(\vec{t},\vec{\tau}) = e^{-iB\tau_K} e^{-iAt_K} \cdots e^{-iB\tau_1} e^{-iAt_1}$ using gradient descent on all $2K$ parameters such that $\|\mathcal{U}(\vec{t^*},\vec{\tau^*}) - \mathcal{V}(\vec{t},\vec{\tau})\| \leq \epsilon$?

  What is the time complexity = minimum number of parameters $2K$ + total number of gradient descent steps ?

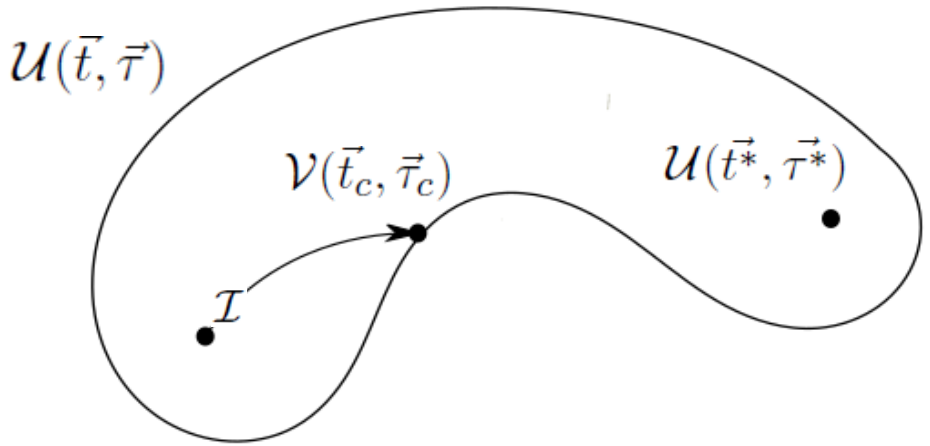
- Suppose $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ is a shallow depth unitary (say, depth-4 with parameters $t_1^*, \tau_1^*, t_2^*, \tau_2^*$), can we find a sequence $\mathcal{V}$ such that $K \leq O(\text{polylog } d)$?

Non-Convex Optimization

QAOA Unitary : $\mathcal{U}(\vec{t},\vec{\tau}) = e^{-iB\tau_N} e^{-iAt_N} \ldots e^{-iB\tau_1} e^{-iAt_1}$

- The space of the set of unitaries $\mathcal{U}(\vec{t},\vec{\tau})$ is in general non-convex.

- Standard gradient descent algorithms do not converge in non-convex spaces.



- Gradient descent usually gets stuck at some local critical point $\vec{t}_c,\vec{\tau}_c$ where $\|\mathcal{U}(\vec{t}^*,\vec{\tau}^*) - \mathcal{V}(\vec{t}_c,\vec{\tau}_c)\| > \epsilon$.

Non-Convex Optimization

- Second order optimization techniques (eg. Newton's method : calculate Hessian and then it's inverse) require,

  a) at least $O(m^2)$ time for a Hessian matrix of dimension m .

  b) fine tuning of hyperparameters.

- Gradient descent methods can be powerful due to their computational efficiency from the above perspectives.

  Require $O(m)$ time to calculate gradients for m parameters, fine tuning not required.

- Can gradient descent optimization enable us to learn paramterized/QAOA unitaries ?

Results so far

QAOA Unitary : $\mathcal{U}(\vec{t},\vec{\tau}) = e^{-iB\tau_N}e^{-iAt_N}\ldots e^{-iB\tau_1}e^{-iAt_1}$

$$\mathcal{V}(\vec{t},\vec{\tau}) = e^{-iB\tau_K}e^{-iAt_K}\ldots e^{-iB\tau_1}e^{-iAt_1}$$

$$\|\mathcal{U}(\vec{t^*},\vec{\tau^*}) - \mathcal{V}(\vec{t},\vec{\tau})\| \leq \epsilon$$

- We find that gradient descent optimization requires at least d^2 parameters in $\mathcal{V}$ to approximate $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ with accuracy ϵ where $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ is sampled from a parameter manifold of dimension $\leq d^2$.

The rate of learning increases when gradient descent is done in overparametrized spaces with dimension $\geq d^2$.

- We propose a greedy algorithm for learning low-depth $\mathcal{U}(\vec{t},\vec{\tau})$ in time $\ll d^2$. However the success probability of efficient learning in non-convex spaces is not ideal.

Gradient Descent

$$\mathcal{U}(\vec{t},\vec{\tau}) = e^{-iB\tau_N}e^{-iAt_N}\ldots e^{-iB\tau_1}e^{-iAt_1}$$

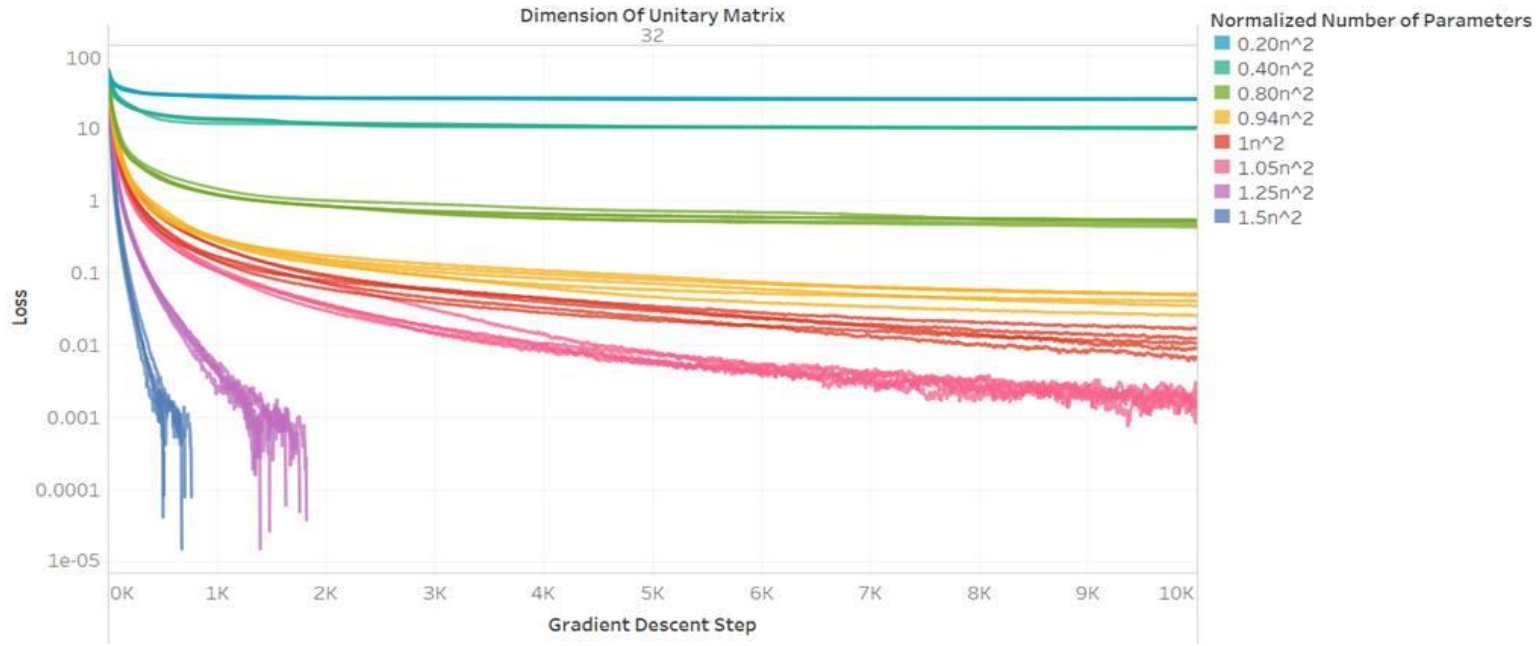
$$\mathcal{V}(\vec{t},\vec{\tau}) = e^{-iB\tau_K}e^{-iAt_K}\ldots e^{-iB\tau_1}e^{-iAt_1}$$

Some basic equations for gradient descent optimization ,

- Loss function : $\mathcal{C}(\vec{t},\vec{\tau})$ = $\|\mathcal{U}(\vec{t^*},\vec{\tau^*}) - \mathcal{V}(\vec{t},\vec{\tau})\|$
- Gradients : $\nabla_t C$, $\nabla_\tau C$
- Learning rate : η , fixed to a certain value during the entire iteration . e.g. η = 0.001
- Parameter update : $t \leftarrow t - \eta\nabla_t C$, $\tau \leftarrow \tau - \eta\nabla_\tau C$

Aim : Optimize $\mathcal{C}(\vec{t},\vec{\tau})$ to a desired accuracy ϵ .

Learning with gradient descent



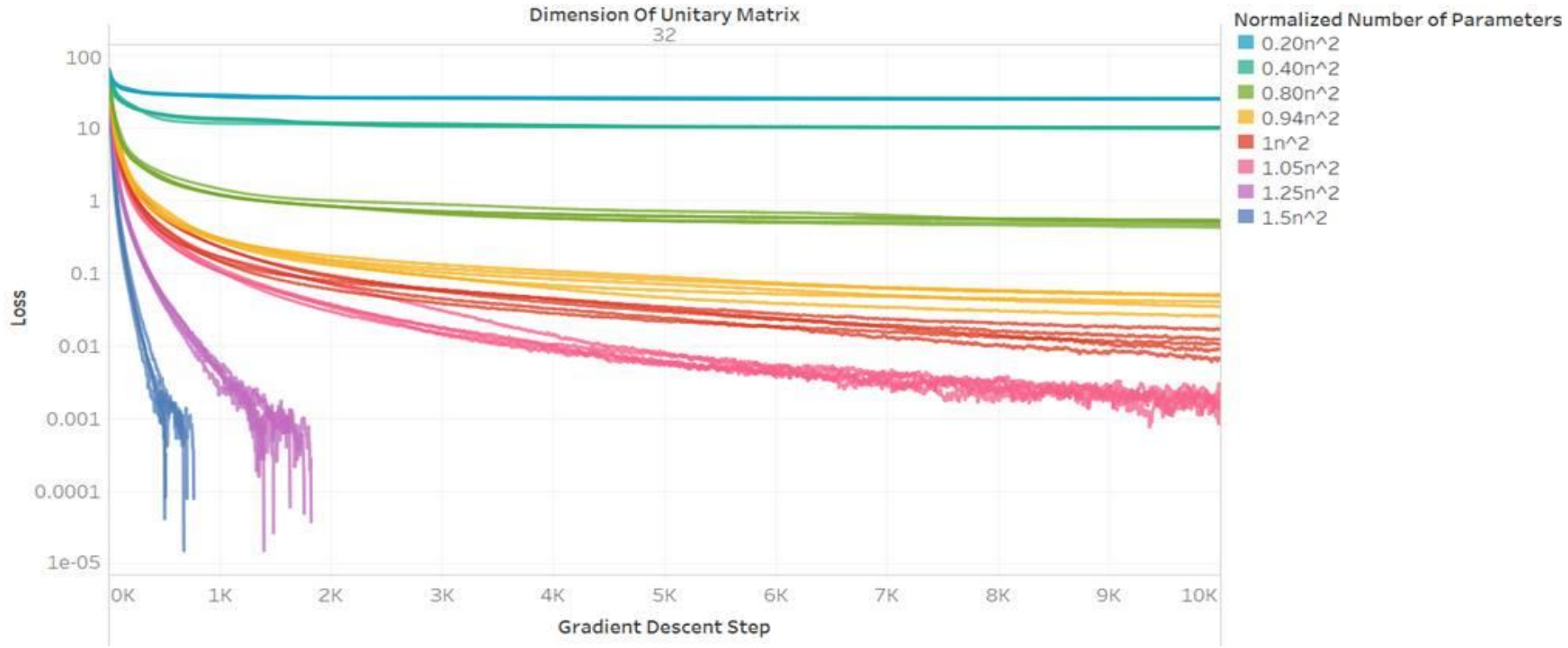
In this work,

Loss Function : $\mathcal{C}(\vec{t},\vec{\tau}) = \|\mathcal{U}(\vec{t^*},\vec{\tau^*}) - \mathcal{V}(\vec{t},\vec{\tau})\|_2^2$.

Aim : Learn $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ to an accuracy $O(10^{-8}.d^2)$.

Simulations for 32 dimension target unitaries $\mathcal{U}(\vec{t},\vec{\tau})$ with $d^2/2$ or 512 parameters while varying the number of learning parameters $2K$ in $\mathcal{V}$.

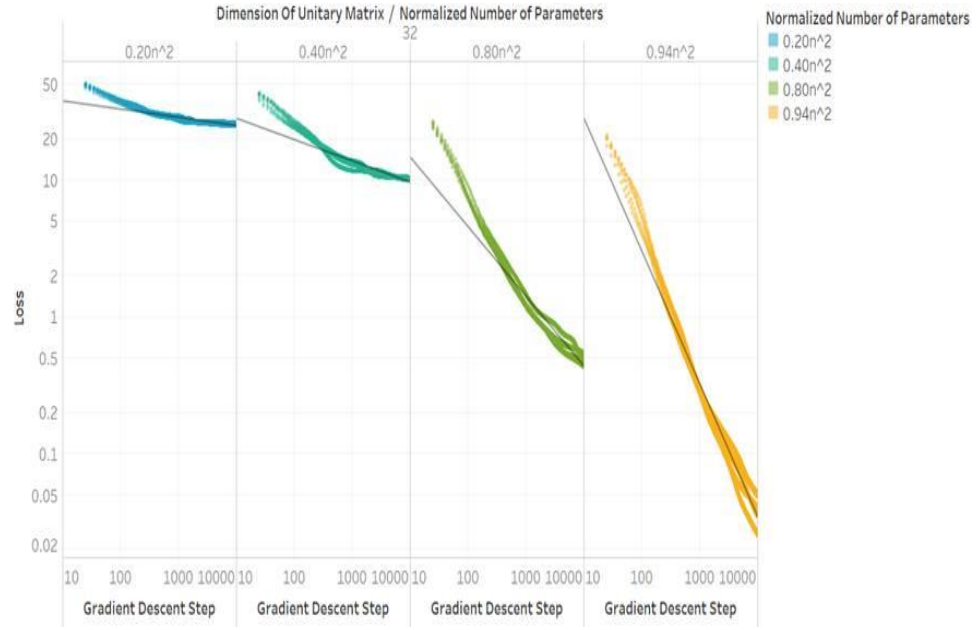
Gradient Descent Numerics



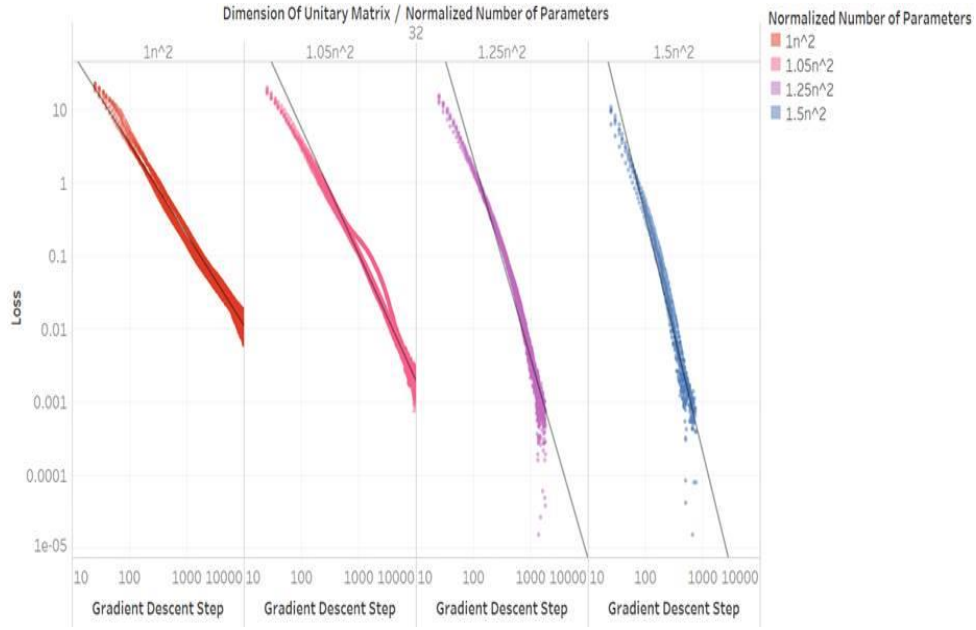
A `transition' occurs when gradient descent is performed in the overparameterized domain, $2K \geq d^2$.

The rate of learning increases as we do gradient descent on more parameters beyond d^2.

Gradient Descent Numerics (Contd.)



Underparametrized



Overparametrized

α = rate of learning

For the first 200 gradient descent steps, Loss $= \kappa\,(\text{no. of grad. descent steps})^{-\alpha}$

The underparametrized models learn $\mathcal{U}(\vec{t^*}, \vec{\tau^*})$ following a power law while the overparametrized models learn faster than the power law.

A Greedy Algorithm for low depth QAOA unitary

Can we learn low depth $\mathcal{U}(\vec{t^*}, \vec{\tau^*})$ with << d^2 parameters ?

A layer = $e^{-iB\tau}e^{-iAt}$

Pseudocode : Given access to $\mathcal{U}(\vec{t^*}, \vec{\tau^*})$ with A, B known .

1. a_0 = Initial Loss = $\|\mathcal{U}(\vec{t^*}, \vec{\tau^*}) - \mathcal{I}\|_2^2$.
2. Add a layer to $\mathcal{I}$ with parameters t_1, τ_1. Cost function = $\|\mathcal{U}(\vec{t^*}, \vec{\tau^*}) - e^{-iB\tau_1}e^{-iAt_1}\|_2^2$.
3. Perform gradient descent on t_1, τ_1 to obtain optimized t_1^1, τ_1^1 .
   a_1 = Updated Loss = $\|\mathcal{U}(\vec{t^*}, \vec{\tau^*}) - e^{-iB\tau_1^1}e^{-iAt_1^1}\|_2^2$ and $a_1 < a_0$.
4. Add a new layer with parameters t_2, τ_2 to the layer in the previous step. Updated Cost function = $\|\mathcal{U}(\vec{t^*}, \vec{\tau^*}) - e^{-iB\tau_2}e^{-iAt_2}e^{-iB\tau_1^1}e^{-iAt_1^1}\|_2^2$
5. Perform gradient descent on $t_1^1, \tau_1^1, t_2, \tau_2$ to obtain optimized $t_1^2, \tau_1^2, t_2^2, \tau_2^2$.
   a_2 = Updated Loss = $\|\mathcal{U}(\vec{t^*}, \vec{\tau^*}) - e^{-iB\tau_2^2}e^{-iAt_2^2}e^{-iB\tau_1^2}e^{-iAt_1^2}\|_2^2$ and $a_2 < a_1$.
6. Repeat the above for n steps till convergence i.e. $a_n < \epsilon$.

Greedy Algorithm Performance

$$\mathcal{U}(\vec{t},\vec{\tau}) = e^{-iB\tau_N} e^{-iAt_N} \cdots e^{-iB\tau_1} e^{-iAt_1}$$

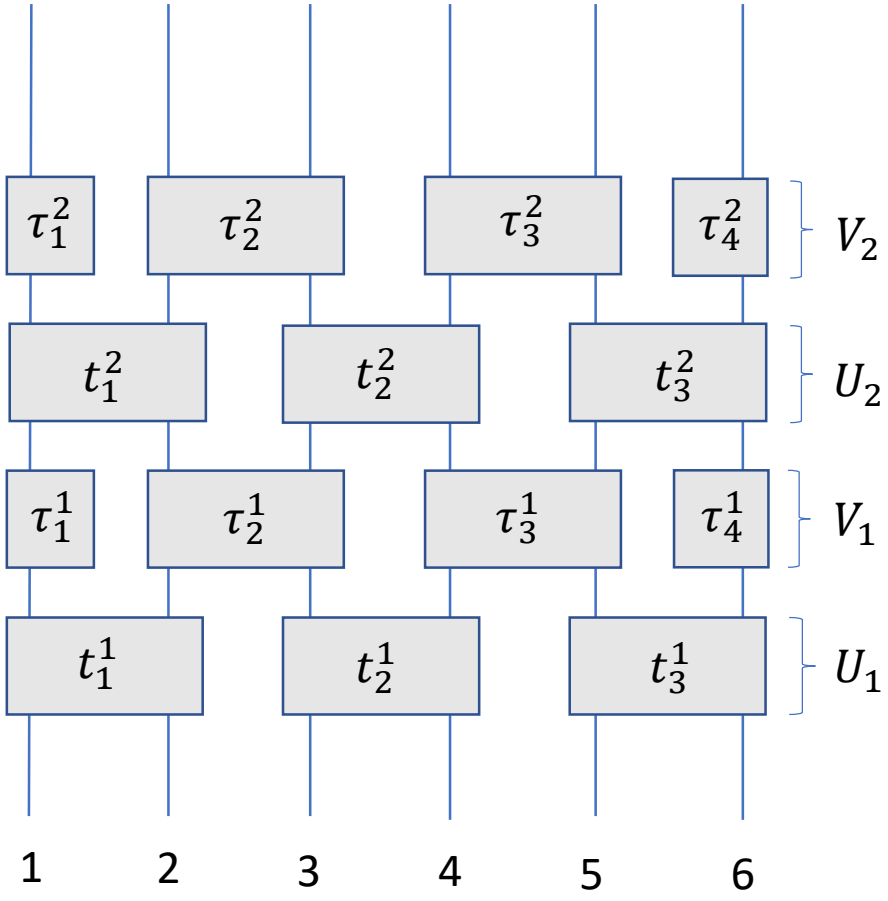
$$\mathcal{V}(\vec{t},\vec{\tau}) = e^{-iB\tau_K} e^{-iAt_K} \cdots e^{-iB\tau_1} e^{-iAt_1}$$

Can we learn low depth $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ with << d^2 parameters ?

- Approximating $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ with depth-4 corresponding to n = 2, 3, 4, 5, 6 qubits .
- Succeeds in finding a sequence $\mathcal{V}$ with at most 20-24 parameters and $\|\mathcal{U}(\vec{t^*},\vec{\tau^*}) - \mathcal{V}(\vec{t},\vec{\tau})\| \leq \epsilon$
- Success probability of learning in non-convex spaces is not ideal , between 0.1 and 0.15 .
- Usually gets stuck at some local critical point or saddle point .

Learning with random local circuits

A general learning setting



Motivation : Study many-body dynamics / MBL .

Goal : Learn / Simulate $\mathcal{U}(\vec{t^*}, \vec{\tau^*})$ with $U_1 V_1 U_2 V_2 \ \ldots U_N V_N$ without assuming knowledge of A, B.

$$U = e^{-iu_{12}t_1} \otimes e^{-iu_{34}t_2} \otimes e^{-iu_{56}t_3}$$

$$V = e^{-iv_1\tau_1} \otimes e^{-iv_{23}\tau_2} \otimes e^{-iv_{45}\tau_3} \otimes e^{-iv_6\tau_4}$$

u , v are random matrices sampled from GUE.

Result : Simulates depth-4 $\mathcal{U}(\vec{t^*}, \vec{\tau^*})$ when gradient descent is done on all d^2 parameters.

Can the local circuit model simulate low depth $\mathcal{U}(\vec{t^*}, \vec{\tau^*})$ with << d^2 parameters ? Can it simulate Haar random unitaries ?

Remarks

QAOA Unitary : $\mathcal{U}(\vec{t},\vec{\tau}) = e^{-iB\tau_N} e^{-iAt_N} \dots e^{-iB\tau_1} e^{-iAt_1}$

- Numerical simulations of the learnability of $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ with at least d^2 parameters by gradient descent.
- A greedy algorithm for simulating short depth $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ with << d^2 parameters. Success probability is not ideal.

In progress

- A rigorous justification of the requirement of more than d^2 parameters for learning $\mathcal{U}(\vec{t^*},\vec{\tau^*})$. Investigate the distribution of critical points in the loss function landscape.
- A local circuit model algorithm that can efficiently simulate low depth $\mathcal{U}(\vec{t^*},\vec{\tau^*})$ with higher success probability than the greedy one.
- Noise resilience of simulating constant depth QAOA unitaries in NISQ devices.