Quantum Algorithms for Escaping from Saddle Points

arXiv: 2007.10253

Jiaqi Leng

Joint work with Chenyi Zhang and Tongyang Li

February 2, 2021

Optimization

Problem:

$$f : \mathbb{R}^n \to \mathbb{R}, \quad \arg\min_x f(x)$$

Core topic in math, theoretical computer science, operations research, etc.

Recently: Significant interest from machine learning:

$$\text{Train an ML model} \iff \text{Optimize a loss function}$$

Provable guarantee for solving an optimization problem?

Convex optimization

Convex optimization problems can be solved in polynomial time with classical algorithms, e.g., ellipsoid method, interior point method, etc.

- **Classical:** $\mathrm{poly}(n, \log 1/\epsilon)$, state-of-the-art: $\tilde{O}(n^2)$ query and $\tilde{O}(n^3)$ time [Lee, Sidford, and Vempala (arXiv:1706.07357)];
- **Quantum:** Assume the quantum evaluation oracle

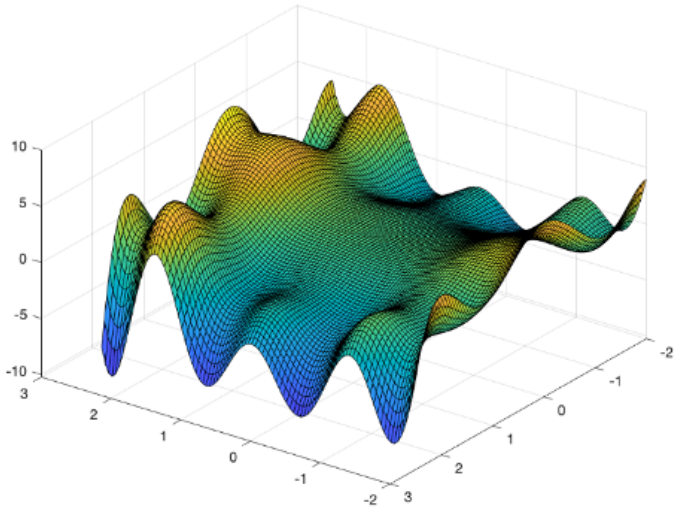
$$O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle,$$

$\tilde{O}(n)$ query and $\tilde{O}(n^3)$ time [Chakrabarti, Childs, Li, and Wu (arXiv:1809.01731); van Apeldoorn, Gilyen, Gribling, de Wolf (arXiv:1809.00643)].

Non-convex optimization in ML

Two concerns about this result in the practice of ML:

- Loss functions of ML models are typically nonconvex.
- Many common cases have large n while can also tolerate reasonably large ϵ.



Speaking of provable guarantee, maybe we want to pursue algorithms with cost

$$\mathrm{poly}(n, \log\, 1/\epsilon) \Rightarrow \mathrm{poly}(\log\, n, 1/\epsilon).$$

Such algorithms are called (almost) **dimension-free** methods.

Non-convex optimization in ML

Common facts about many learning problems:

- Finding the global minima is NP-hard (intractable in practice).
- The major difficulty is not local minima:
  - no spurious local minima;
  - local minima are nearly as good as global minima.
- Saddle points are ubiquitous;
- Saddle points (and local maxima) can give highly suboptimal solutions.

Non-convex optimization in ML

Common facts about many learning problems:

- Finding the global minima is NP-hard (intractable in practice).
- The major difficulty is not local minima:
  - no spurious local minima;
  - local minima are nearly as good as global minima.
- Saddle points are ubiquitous;
- Saddle points (and local maxima) can give highly suboptimal solutions.

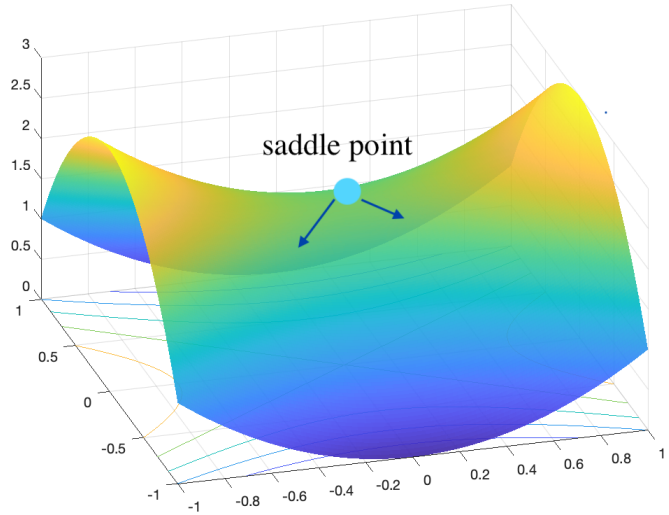
Conclusion: We would want to escape from saddle points, but are satisfied with reaching an ϵ-approx. local minimum x_ϵ:

$$\|\nabla f(x_\epsilon)\| \le \epsilon, \quad \lambda_{\min}(\nabla^2 f(x_\epsilon)) \ge -\sqrt{\rho\epsilon}.$$

Here f is ρ-Hessian Lipschitz: $\|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \le \rho\|\mathbf{x}_1 - \mathbf{x}_2\|$, $\forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$.

Escaping from saddle points

The main idea: perturbed gradient descent (PGD)

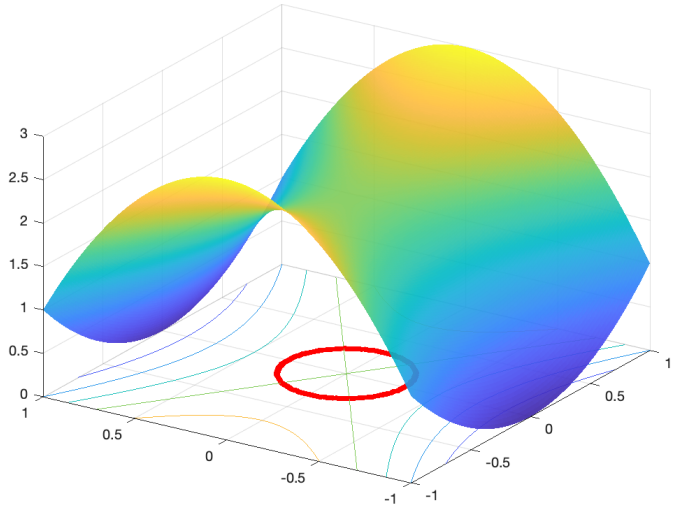


- **Radius of perturbation:** Too large? Too small?
- **Way of perturbation:** What's the most efficient approach?
- **Gradient descent:** GD/SGD? Faster versions?

Escaping from saddle points: classical proposal

Classically, uniform perturbation in a ball + accelerated GD.

Complexity: $\tilde{O}(\log^6 n/\epsilon^{1.75})$ [Jin, Netrapalli, Jordan (arXiv:1711.10456)].

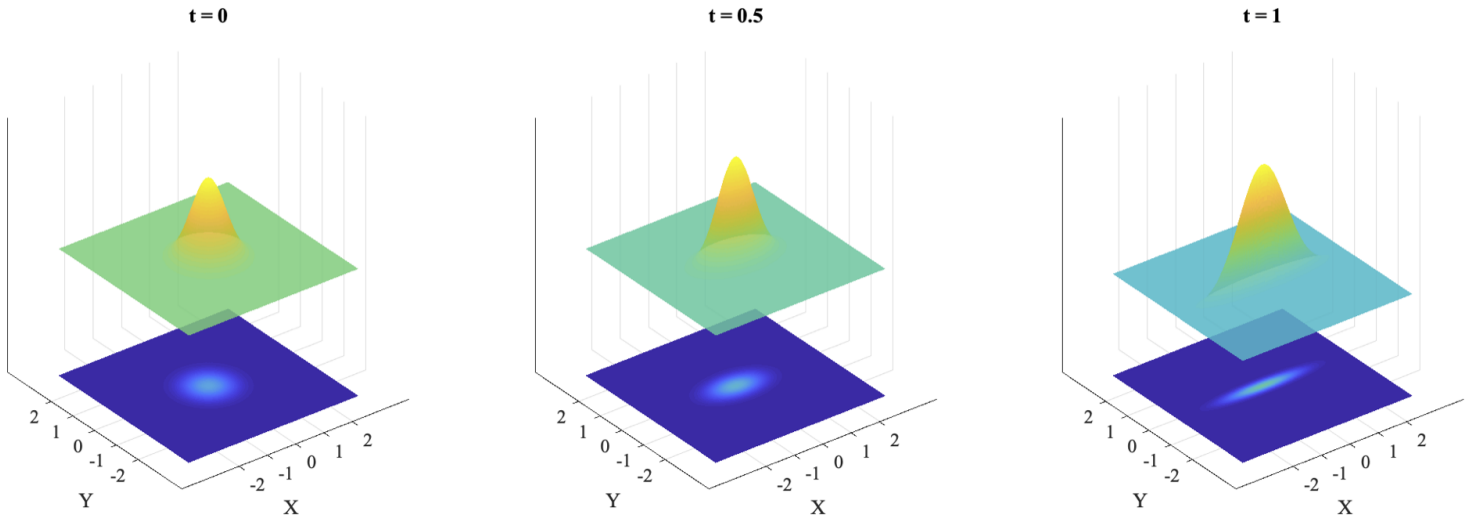


Nesterov's accelerated gradient descent (AGD)

$$y_t \leftarrow x_t + (1-\theta)v_t,\ x_{t+1} \leftarrow y_t - \eta\nabla f(y_t),\ v_{t+1} \leftarrow x_{t+1} - x_t.$$

Escaping from saddle points: quantum proposal

The main observation: Gaussian wave packets disperse along the negative curvature direction of the saddle point.



We give a hybrid quantum-classical algorithm where the perturbation is replaced by the measurement of a **squeezed** Gaussian wave function. Then, we apply gradient descents.

Complexity: $\tilde{O}(\log^2 n/\epsilon^{1.75})$.

How to prepare the squeezed state?

Easy, let it evolve like quantum!

$$i\frac{\partial}{\partial t}\Psi = \Big[-\frac{1}{2}\nabla^2 + f(x)\Big]\Psi.$$

How to prepare the squeezed state?

Easy, let it evolve like quantum!

$$i\frac{\partial}{\partial t}\Psi = \Big[-\frac{1}{2}\nabla^2 + f(x)\Big]\Psi.$$

Near a saddle point, the function is well-approximated by a quadratic function, i.e., $f(x) = \frac{\lambda}{2}x^2$. If the initial wave function is Gaussian (the distribution of the position quadrature follows $\mathcal{N}(0,1)$), and it evolves for $t \geq 0$, the position quadrature still follows a normal distribution $\mathcal{N}\left(0, \sigma^2(t;\lambda)\right)$.

How to prepare the squeezed state?

Easy, let it evolve like quantum!

$$i\frac{\partial}{\partial t}\Psi = \Big[-\frac{1}{2}\nabla^2 + f(x)\Big]\Psi.$$

Near a saddle point, the function is well-approximated by a quadratic function, i.e., $f(x) = \frac{\lambda}{2}x^2$. If the initial wave function is Gaussian (the distribution of the position quadrature follows $\mathcal{N}(0,1)$), and it evolves for $t \geq 0$, the position quadrature still follows a normal distribution $\mathcal{N}\left(0, \sigma^2(t;\lambda)\right)$.

$$\sigma^2(t;\lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2)-(1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2+4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases}$$

Why it works?

$$\sigma^2(t;\lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2)-(1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2+4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases}$$

Fact: Exp. dispersion rate for $\lambda < 0$, quadratic for $\lambda = 0$, and at most constant for $\lambda > 0$.

Why it works?

$$\sigma^2(t;\lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2)-(1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2+4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases}$$

Fact: Exp. dispersion rate for $\lambda < 0$, quadratic for $\lambda = 0$, and at most constant for $\lambda > 0$.

If we are near a saddle, then at least one eigenvalue is quite negative (less than $-\sqrt{\rho\epsilon}$). The quantum wave immediately "leaks" along these negative directions.

Compared with the uniform perturbation in the classical proposal, we make use of the geometry of the landscape!

The cost of preparing squeezed states

The cost of the quantum simulation is cheap, but only for quantum computers!

- Simulating (large) quantum systems is believed to be intractable for classical computers: the cost scales **exponentially** with respect to the dimension n.
- While quantum computers are particularly good for quantum simulation. The cost scales **polynomially** with respect to the dimension n [Childs, Liu, Ostrander (arXiv:2002.07868)].

Overview of our quantum algorithm

- Step 1: Start with an initial guess x_0. Compute the gradient.
- Step 2: When the gradient is large, apply AGD;
- Step 3: When the gradient is small, run quantum simulation with time $O(\log n)$ and measure (with position quadrature);
- Step 4: Apply AGD for $O(\log n)$ iterations, and go to Step 2 or 3 (depending on the norm of the gradient).

Overview of our quantum algorithm

- Step 1: Start with an initial guess x_0. Compute the gradient.
- Step 2: When the gradient is large, apply AGD;
- Step 3: When the gradient is small, run quantum simulation with time $O(\log n)$ and measure (with position quadrature);
- Step 4: Apply AGD for $O(\log n)$ iterations, and go to Step 2 or 3 (depending on the norm of the gradient).

The gradient is computed by the Jordan's algorithm (using the quantum evaluation oracle).

Some previous works [CCLW18, AGGW18] showed how to apply Jordan's algorithm to convex optimization. We generalize it to non-convex optimization.

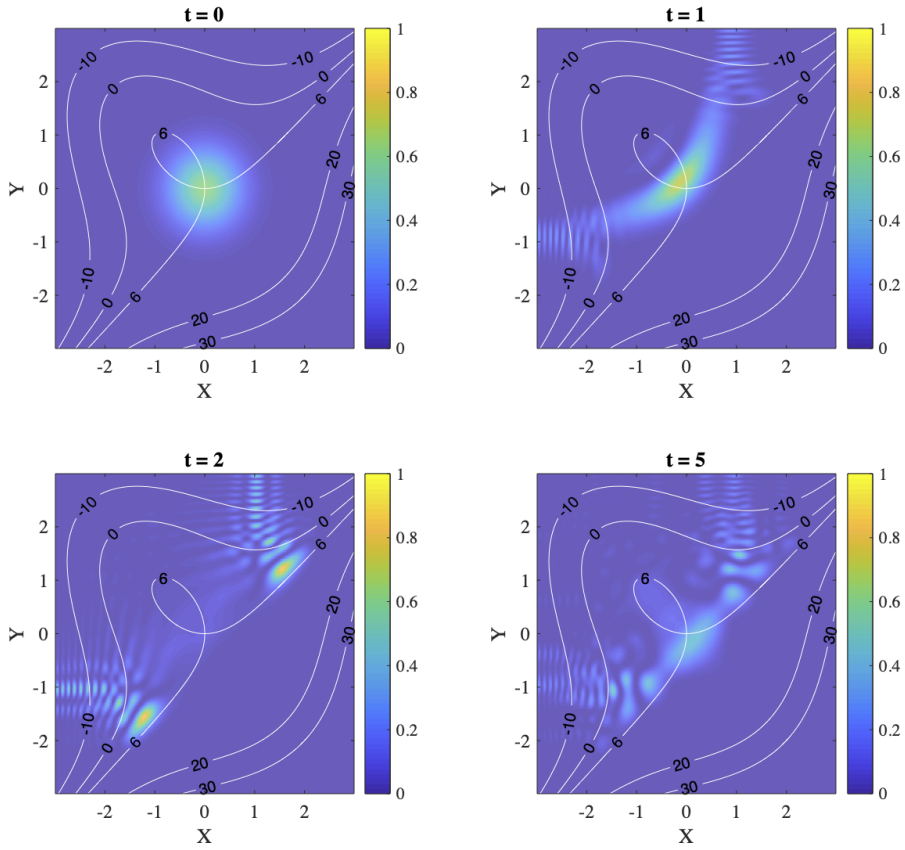
Summary

Reference	Queries	Oracle
Prior arts [JNJ17]	$\tilde{O}(\log^6 n/\epsilon^{1.75})$	Gradient
This work	$\tilde{O}(\log^2 n/\epsilon^{1.75})$	Quantum evaluation

Outcome: Cubic quantum speedup in n, match the classical best-known in ϵ.

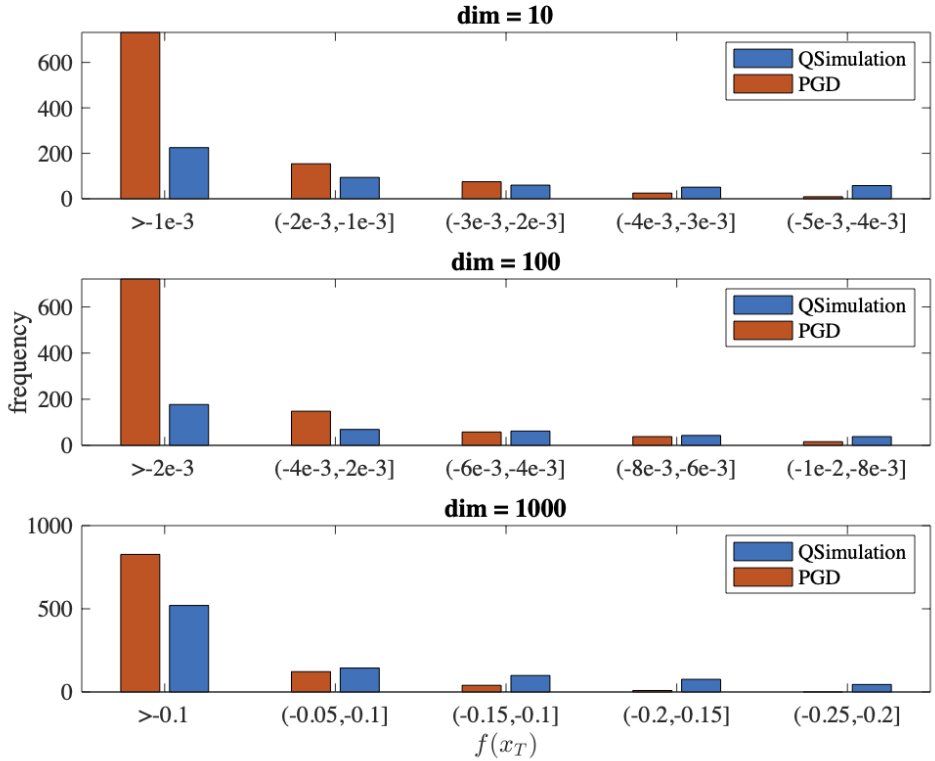
- Achieve quantum speedup by using quantum simulation to escape from saddle points;
- Reduce classical gradients to quantum evaluations by introducing Jordan's algorithm.

Numerical experiment: non-quadratic potential



The quantum evolution of a Gaussian wave packet over the potential function $f(x, y) = x^3 - y^3 - 2xy + 6$. The wave packet is shown as a heat map over the contour of the potential function.

Numerical experiment: dimension dependence



The objective function is of dimension $n = 10^p$ for $p = 1, 2, 3$. Quantum evolution time $t_e = p$. Classical iteration number $\mathscr{T}_c = 50p^2 + 50$, quantum iteration number $\mathscr{T}_q = 30p$. We take $M = 1000$ initial guesses for both methods.

Open questions

- Can we give quantum-inspired classical algorithms for escaping from saddle points? (Dequantizing the quantum algorithm, perhaps?)
- Can quantum algorithms achieve speedup in terms of $1/\epsilon$?
- Beyond local minima, does quantum provide advantage for approaching global minima?

Open questions

- Can we give quantum-inspired classical algorithms for escaping from saddle points? (Dequantizing the quantum algorithm, perhaps?)
- Can quantum algorithms achieve speedup in terms of $1/\epsilon$?
- Beyond local minima, does quantum provide advantage for approaching global minima?

Thanks for your attention!