Near-Optimal Quantum Algorithm for Minimizing the Maximal Loss

Hao Wang*

Peking University

Joint work with Chenyi Zhang* and Tongyang Li

arXiv:2402.12745

ICLR 2024

*: Equal Contribution

Minimizing the Maximal Loss

Problem: Find $x_\star$ such that $F_{\max}(x_\star) - \inf_{x \in \mathbb{R}^d} F_{\max}(x) \le \epsilon$

where $F_{\max}(x) := \max_{i \in [N]} f_i(x)$ with each $f_i : \mathbb{R}^d \to \mathbb{R}$ convex and L-Lipshitz

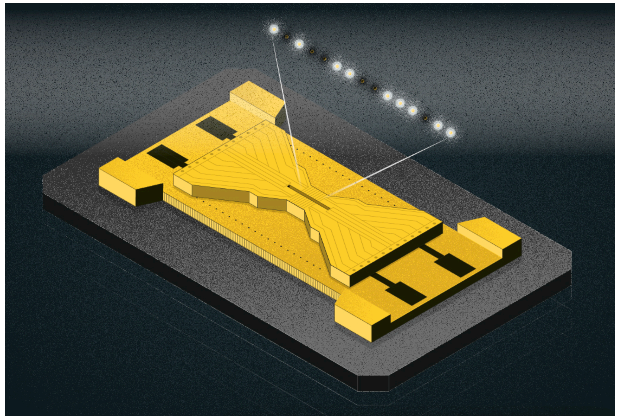
Wide applications in machine learning and optimization.

For example, in Support Vector Machines (SVM), f_i's are loss functions representing the negative margin of the i-th datapoint.

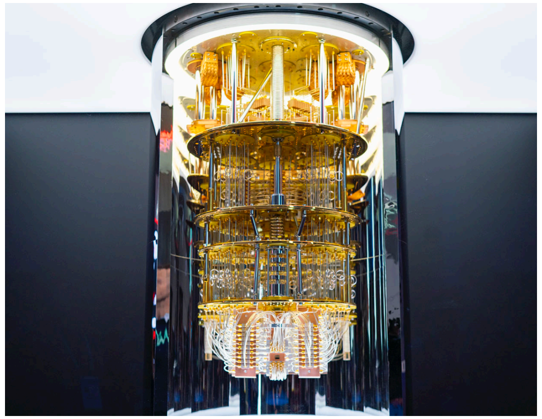
Also related to robust optimization.

Quantum computing

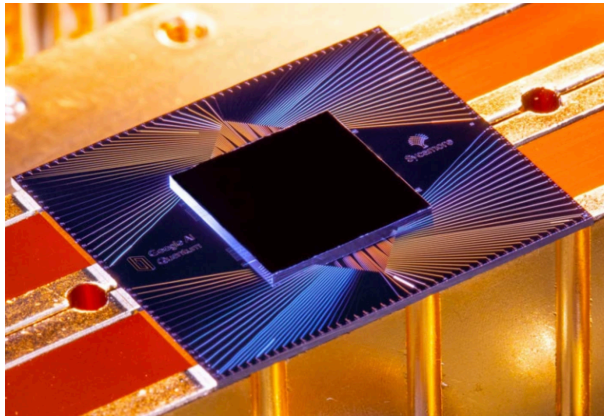
Rapidly advancing technology, having 1000+ qubits now.



IonQ

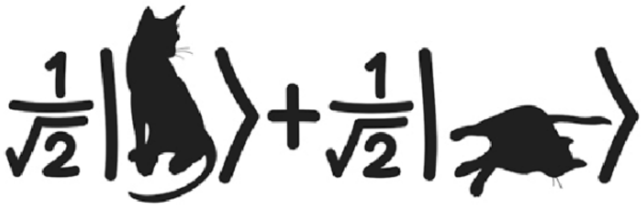


IBM



Google

Can *coherently* access & process data in *quantum superpositions*, like a Schr
ödinger’s cat:



Quantum computing for optimization

Assume *quantum access* to the function value of f ,

$$O_f : |x\rangle |0\rangle \to |x\rangle |f(x)\rangle$$

where $|\cdot\rangle$ denotes registers storing quantum states that allow queries in *quantum superpositions:*

$$O_f(\sum c_i |x_i\rangle |0\rangle) = \sum c_i |x_i\rangle |f(x_i)\rangle$$

[Jordan, 2004]: Can compute the gradient of f using only one query to O_f .

⇕

$\Theta(d)$ function value queries in the classical setting

Classical results on minimizing the maximal Loss

Intuitively, using a subgradient method could output a solution in $O(\epsilon^{-2})$ iterations. In total, each iteration would cost N queries to the oracle, which leads to an overall query complexity of $O(N\epsilon^{-2})$.

[Carmon et al., 2021]. Assuming access to the gradient, using ball oracle optimization scheme improves the complexity to $\tilde{O}(N\epsilon^{-2/3} + \epsilon^{-8/3})$. This complexity is also a classical lower bound on this problem.

Can quantum do better?

Quantum algorithm for minimizing the maximal loss

[This work] In this work, we prove that

1. Assuming access to a quantum **zeroth-order** oracle,

$$O_f \left|i\right\rangle \left|x\right\rangle \left|y\right\rangle \to \left|i\right\rangle \left|x\right\rangle \left|y + f_i(x)\right\rangle$$

There exists a quantum algorithm that achieves a time complexity of

$$\tilde{O}(\sqrt{N}\epsilon^{-5/3} + \epsilon^{-8/3})$$

2. Any quantum algorithm granted access to a quantum first-order oracle must take $\tilde{\Omega}(\sqrt{N}\epsilon^{-2/3})$ queries. In other words, the dependence of our algorithm on N is **optimal** up to poly-logarithmic factors.

Quantum algorithms can achieve a quadratic speedup!

Algorithm design

Our algorithm is based on [Carmon et al., 2021].

The design of our algorithm is based on two observations:

1. Only a zeroth-order quantum oracle is needed to obtain gradient information using Jordan's algorithm. [Jordan, 2004]

2. In [Carmon et al., 2021], the regularized ball optimization oracle is implemented using a modified version of stochastic gradient descent, in which the main bottleneck is the sampling from N loss functions.

Moreover, during each call to the ball optimization oracle, multiple samples are needed. This can be accelerated using quantum Gibbs sampling [Hamoudi, 2022], providing a quadratic speedup.

Pseudocode

Algorithm 1: Quantum-Epoch-SGD-Proj on the exponentiated softmax

Prepare T identical states $|\psi\rangle = \sum_i e^{f_i(\bar{x})/2\epsilon'} / \sqrt{\sum_{i\in[N]} e^{f_i(\bar{x})/\epsilon'}} \, |i\rangle$ using Quantum Gibbs Sampling [Hamoudi, 2022]

Initialize $x_1^1 \in \mathbb{B}_{r_\epsilon}(\bar{x})$ arbitrarily, set $k = 1$

while $\sum_{i\in[k]} T_i \leq T$ **do**

  for $t = 1, \ldots, T_k$ **do**

    Sample $i \in [N]$ by measuring the next unmeasured $|\psi\rangle$

    Call Jordan's algorithm [Jordan, 2005] to compute $\nabla f_i^\lambda(x) = \nabla f_i(x) + \lambda(x - \bar{x})$

    $\hat{g}_t = e^{(f_i^\lambda(x) - f_i^\lambda(\bar{x}))/\epsilon'} \nabla f_i^\lambda(x)$

    Update $x_{t+1}^k \leftarrow \Pi_{B_r(\bar{x}) \cap B_{D_k}(x_1^k)}(x_t^k - \eta_k \hat{g}_t)$

  Let $x_1^{k+1} \leftarrow \frac{1}{T_k} \sum_{t\in[T_k]} x_t^k$

  Update parameters $T_{k+1} \leftarrow 2T_k$, $\eta_{k+1} \leftarrow \eta_k/2$, $D_{k+1} \leftarrow D_k/\sqrt{2}$, $k \leftarrow k+1$

return x_1^k

Lower Bound Proof Sketch

Classical proof strategy. Hard instance with an "N-element chain structure":

1. Information along the chain can only be revealed sequentially
2. Finding a stationary point $\Longleftrightarrow$ learning the entire chain
3. A correct loss function needs to be found in order to make progress

Any classical algorithm needs to go through the chain *sequentially* and checking all the loss functions to find the correct one.

[This work]. "chain structure" is also hard for quantum. However, we could show that a quantum algorithm only needs $\tilde{\Omega}(\sqrt{N})$ queries instead of $\Omega(N)$ queries due to Grover's search.

Thank you!

More information:

arXiv:2402.12745