University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

ZEROTH-ORDER OPTIMIZATION WITH TRAJECTORY INFORMED DERIVATIVE ESTIMATION

Peiyao Xiao

02-15-2023

Content

- Introduction to Zeroth-order optimization
- Existing methods to solve the problems
- Creativeness of this paper
- Experiments results
- Pros and cons
- Conclusion

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

INTRODUCTION

What is zeroth-order optimization

+

Existing methods

Zeroth-order (ZO) optimization

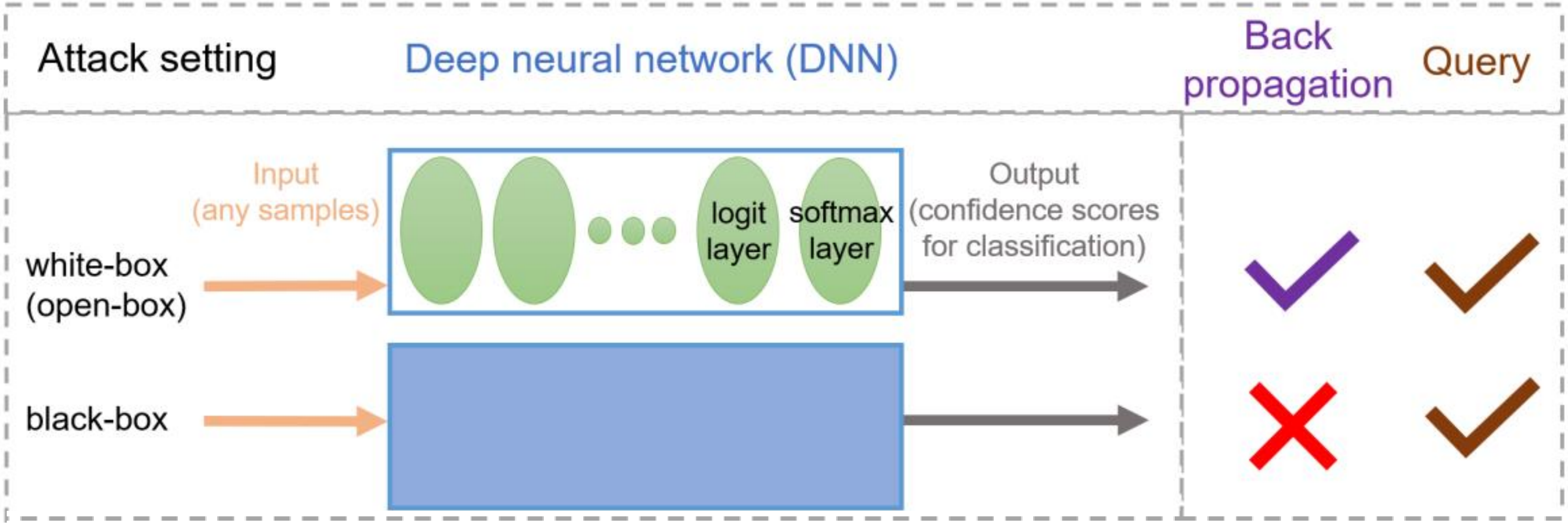
- Many machine learning (ML) and deep learning (DL) applications involve tackling complex optimization problems that are difficult to solve analytically.
- Often the objective function itself may not be in analytical closed form, only permitting function evaluations but not gradient evaluations.
- Optimization corresponding to these types of problems falls into the category of zeroth-order (ZO) optimization

$$\hat{\nabla} f(x) \approx \nabla f(x)$$

$\hat{\nabla} f(x)$ is the estimated gradient
$\nabla f(x)$ is the true gradient

arXiv:2006.06224

Black-box attack



- Deep neural networks are vulnerable. Attack them to lead wrong result
- Assume we can only know (input, output) pairs, which is called queries

arXiv:1708.03999

Existing methods to solve the problems

Finite difference (FD)

unit length vector

$$\hat{\nabla} f(x) = \frac{1}{d} \sum_{i=1}^{d} \frac{f(x+\beta \mathbf{u}_i) - f(x)}{\beta} \mathbf{u}_i$$

dimension

smoothing parameter

$$\hat{\nabla} f_i(x) = \frac{f(x+h\mathbf{e}_i) - f(x-h\mathbf{e}_i)}{2h}$$

- Cropped ImageNet dataset: d = 256 × 256 × 3 = 196, 608
- Too many queries!!

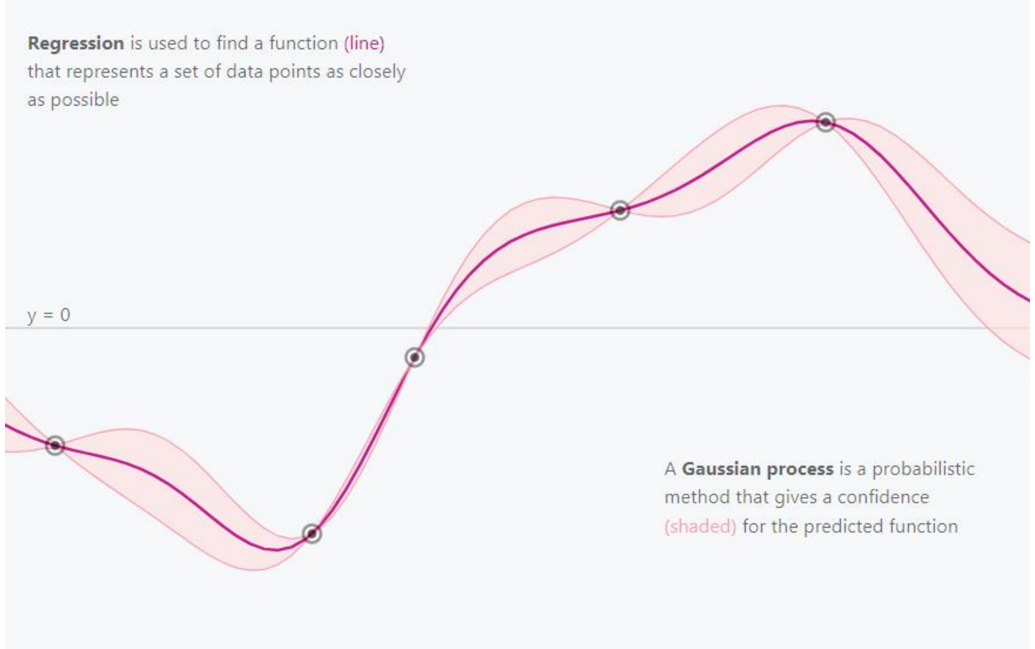
Gaussian process (GP)

- Objective function is sampled from a GP
- The derivative at any input in the domain follows a Gaussian distribution

Brief GP introduction

GP collects infinite Gaussian distribution

- A GP provides a distribution, rather than a single point
- GP projection on an input → Gaussian distribution
- Derivatives of a Gaussian distribution→ Gaussian distribution
- Conditioning, still a Gaussian distribution
- Gaussian distribution depends on mean and variance



Brief GP introduction

A common assumption: f is sampled from *Gaussian process* (GP)

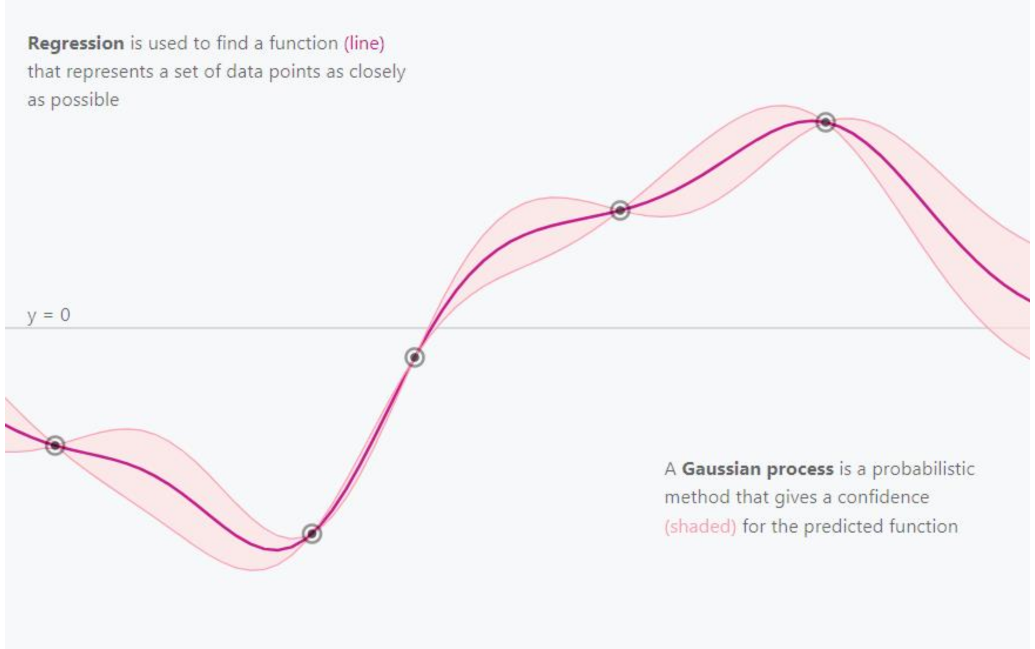
$$f \sim \mathcal{GP}\left(\mu(\cdot), \sigma^2(\cdot,\cdot)\right)$$

$$y(x) = f(x) + \zeta, \quad \zeta(0, \sigma^2)$$

In every iteration t, conditioning on all data before $\{(\boldsymbol{x}_\tau, y_\tau)\}_{\tau=1}^{t-1}$

f follows the posterior GP

$$f \sim \mathcal{GP}\left(\mu_{t-1}(\cdot), \sigma_{t-1}^2(\cdot,\cdot)\right)$$



https://distill.pub/2019/visual-exploration-gaussian-processes/

Brief GP introduction

In every iteration t, conditioning on all data before

f follows the posterior GP

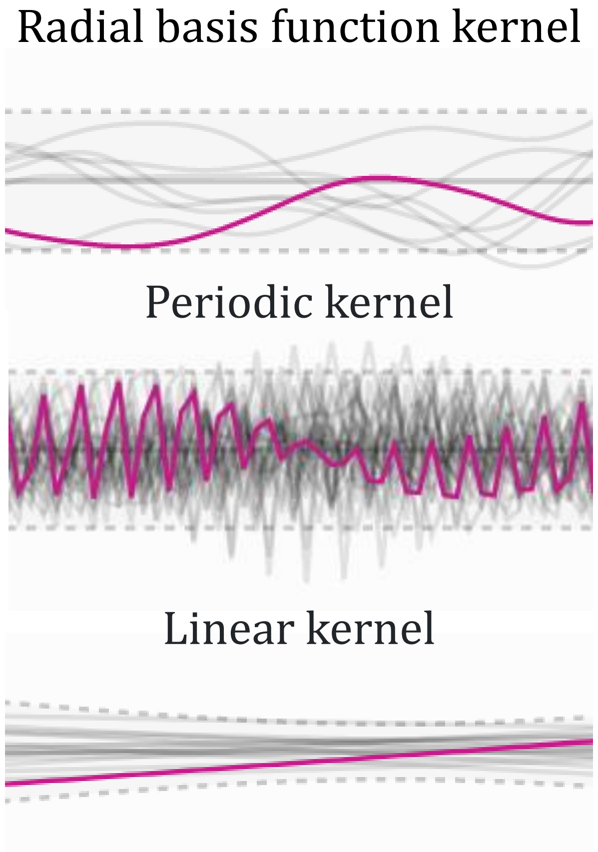
$$f \sim \mathcal{GP}\left(\mu_{t-1}(\cdot), \sigma_{t-1}^2(\cdot,\cdot)\right)$$

$$\mu_{t-1}(\boldsymbol{x}) \triangleq \boldsymbol{k}_{t-1}(\boldsymbol{x})^\top \left(\mathbf{K}_{t-1} + \sigma^2 \mathbf{I}\right)^{-1} \boldsymbol{y}_{t-1}$$

$$\sigma_{t-1}^2(\boldsymbol{x}, \boldsymbol{x}') \triangleq k\left(\boldsymbol{x}, \boldsymbol{x}'\right) - \boldsymbol{k}_{t-1}(\boldsymbol{x})^\top \left(\mathbf{K}_{t-1} + \sigma^2 \mathbf{I}\right)^{-1} \boldsymbol{k}_{t-1}\left(\boldsymbol{x}'\right)$$

Posterior distribution at x is Gaussian with mean $\mu_{t-1}(\boldsymbol{x})$ and variance $\sigma_{t-1}^2(\boldsymbol{x})$

$$\sigma_{t-1}^2(\boldsymbol{x}) \triangleq \sigma_{t-1}^2(\boldsymbol{x}, \boldsymbol{x})$$



Learning materials

- Gaussian process lecture

  https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html

- A Visual Exploration of Gaussian Processes

  https://distill.pub/2019/visual-exploration-gaussian-processes/

- Gaussian processes (3/3) - exploring kernels

  https://peterroelants.github.io/posts/gaussian-process-kernels/

KEY IDEAS

Trajectory-informed Derivative Estimation

+

Dynamic Virtual Updates

Trajectory-informed Derivative Estimation

Algorithm 2: ZORD (Ours)

Input: In addition to the parameters in Algo. 1, set the steps of virtual updates $\{V_t\}_{t=1}^{T}$

for iteration $t = 1, \ldots, T$ **do**

  $\boldsymbol{x}_{t,0} \leftarrow \boldsymbol{x}_{t-1}$

  for iteration $\tau = 1, \ldots, V_t$ **do**

    $\boldsymbol{x}_{t,\tau} \leftarrow \mathcal{P}_{\mathcal{X}}\left(\boldsymbol{x}_{t,\tau-1} - \eta_{t,\tau-1} \nabla \mu_{t-1}(\boldsymbol{x}_{t,\tau-1})\right)$

  end for

  Query $\boldsymbol{x}_t = \boldsymbol{x}_{t,\tau}$ to yield $y(\boldsymbol{x}_t)$

  Update (4) using optimization trajectory

end for

Return $\arg\min_{\boldsymbol{x}_{1:T}} y(\boldsymbol{x})$

$$f \sim \mathcal{GP}\left(\mu(\cdot), \sigma^2(\cdot,\cdot)\right)$$

↓

$$\nabla f \sim \mathcal{GP}\left(\nabla \mu(\cdot), \partial \sigma^2(\cdot,\cdot)\right)$$

↓

$$\nabla f(\boldsymbol{x}) \approx \nabla \mu_{t-1}(\boldsymbol{x})$$

$\mathcal{P}_{\mathcal{X}}(\boldsymbol{x}) \triangleq \arg\min_{\boldsymbol{z} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 / 2$ Projection finds the nearest point

Trajectory-informed Derivative Estimation

Estimate the derivative at any input x using the posterior mean

$$\nabla f(\boldsymbol{x}) \approx \nabla \mu_{t-1}(\boldsymbol{x}) \qquad \nabla \mu_{t-1}(\boldsymbol{x}) \triangleq \partial_{\boldsymbol{z}} \boldsymbol{k}_{t-1}(\boldsymbol{z})^\top \left(\mathbf{K}_{t-1} + \sigma^2 \mathbf{I}\right)^{-1} \boldsymbol{y}_{t-1}\Big|_{\boldsymbol{z}=\boldsymbol{x}}$$

Employ the posterior covariance matrix to obtain a principled measure of uncertainty

$$\partial \sigma^2_{t-1}(\boldsymbol{x})$$

Only makes use of the naturally available optimization trajectory D_{t-1} and does not need any additional query

Dynamic Virtual Updates

Algorithm 2: ZoRD (Ours)

Input: In addition to the parameters in Algo. 1, set the steps of virtual updates $\{V_t\}_{t=1}^{T}$
for iteration $t = 1, \ldots, T$ **do**
  $\boldsymbol{x}_{t,0} \leftarrow \boldsymbol{x}_{t-1}$
  for iteration $\tau = 1, \ldots, V_t$ **do**
    $\boldsymbol{x}_{t,\tau} \leftarrow \mathcal{P}_{\mathcal{X}}\left(\boldsymbol{x}_{t,\tau-1} - \eta_{t,\tau-1} \nabla \mu_{t-1}(\boldsymbol{x}_{t,\tau-1})\right)$
  end for
  Query $\boldsymbol{x}_t = \boldsymbol{x}_{t,\tau}$ to yield $y(\boldsymbol{x}_t)$
  Update (4) using optimization trajectory
end for
Return $\arg\min_{\boldsymbol{x}_{1:T}} y(\boldsymbol{x})$

$\mathcal{P}_{\mathcal{X}}(\boldsymbol{x}) \triangleq \arg\min_{\boldsymbol{z} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 / 2$ Projection finds the nearest point

Dynamic Virtual Updates

Update V_t times without queries, more query efficient

$$\boldsymbol{x}_{t,\tau} = \mathcal{P}_{\mathcal{X}}\left(\boldsymbol{x}_{t,\tau-1} - \eta_{t,\tau-1}\nabla\mu_{t-1}(\boldsymbol{x}_{t,\tau-1})\right) \quad \forall\tau = 1, \cdots, V_t$$

Trade off

- Large V_t → lead to usage of inaccurate derivative estimation
- Small V_t → may not fully exploit the benefit of derivative estimation

Theoretical analysis

Theorem 1 (Derivative Estimation Error). *Let* $\delta \in (0,1)$ *and* $\beta \triangleq \sqrt{d + 2(\sqrt{d}+1)\ln(1/\delta)}$. *For any* $\boldsymbol{x} \in \mathcal{X}$ *and any* $t \geq 1$, *the following holds with probability of at least* $1-\delta$,

$$\|\nabla f(\boldsymbol{x}) - \nabla \mu_t(\boldsymbol{x})\|_2 \leq \beta \left\|\partial \sigma_t^2(\boldsymbol{x})\right\|_2 \ .$$

Gap between the true gradient and estimated gradient, bounded by uncertainty

Theoretical analysis

Theorem 2 (Non-Increasing Error). *For any $\boldsymbol{x} \in \mathcal{X}$ and any $t \geq 1$, we have that*

$$\left\|\partial\sigma_t^2(\boldsymbol{x})\right\|_2 \leq \left\|\partial\sigma_{t-1}^2(\boldsymbol{x})\right\|_2 .$$

Let $\delta \in (0,1)$. Define $r \triangleq \max_{\boldsymbol{x}\in\mathcal{X},t\geq 1} \left\|\partial\sigma_t^2(\boldsymbol{x})\right\|_2 / \left\|\partial\sigma_{t-1}^2(\boldsymbol{x})\right\|_2$, given the β in Thm. 1 , we then have that $r \in [1/(1+1/\sigma^2), 1]$, and that with probability of at least $1-\delta$,

$$\|\nabla f(\boldsymbol{x}) - \nabla\mu_t(\boldsymbol{x})\|_2 \leq \beta \left\|\partial\sigma_t^2(\boldsymbol{x})\right\|_2 \leq \kappa\beta r^t .$$

$$\|\partial_{\boldsymbol{z}}\partial_{\boldsymbol{z}'}k(\boldsymbol{z},\boldsymbol{z}')|_{\boldsymbol{z}=\boldsymbol{z}'=\boldsymbol{x}}\|_2 \leq \kappa \text{ , } \forall \boldsymbol{x} \in \mathcal{X} \text{ for some } \kappa > 0$$

Uncertainty is non-increasing

The gap can be exponential decay if r < 1

Theoretical analysis

The convergence of our ZoRD is formally guaranteed by Thm. 3 below (proof in Appx. B.4).

Theorem 3 (Convergence of ZoRD). *Let* $\delta \in (0,1)$. *Suppose our ZoRD (Algo. 2) is run with* $V_t = V$ *and* $\eta_{t,\tau} = \eta \leq 1/L_s$ *for any* t *and* τ. *Then with probability of at least* $1-\delta$, *when* $r < 1$,

$$\min_{t \leq T} \frac{1}{V} \sum_{\tau=0}^{V-1} \|G_{t,\tau}\|_2^2 \leq \underbrace{\frac{2[f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*)]/\eta}{TV}}_{①} + \underbrace{\frac{2\alpha^2 r^2}{T(1-r^2)} + \frac{(2L_c + 1/\eta)\alpha r}{T(1-r)}}_{②}$$

where $\alpha \triangleq \kappa\sqrt{d + 2(\sqrt{d}+1)\ln(VT/\delta)}$. *When* $r = 1$, *we instead have* ② $= 2\alpha^2 + (2L_c + 1/\eta)\alpha$.

$$G_{t,\tau} \triangleq \left(\boldsymbol{x}_{t,\tau} - \mathcal{P}_{\mathcal{X}}(\boldsymbol{x}_{t,\tau} - \eta_{t,\tau}\nabla f(\boldsymbol{x}_{t,\tau}))\right)/\eta_{t,\tau} \ .$$

r < 1, converge at a rate of O (1/T), r = 1, O(1/T + C)

Query complexity O(T) instead of O(nT)

Experiment

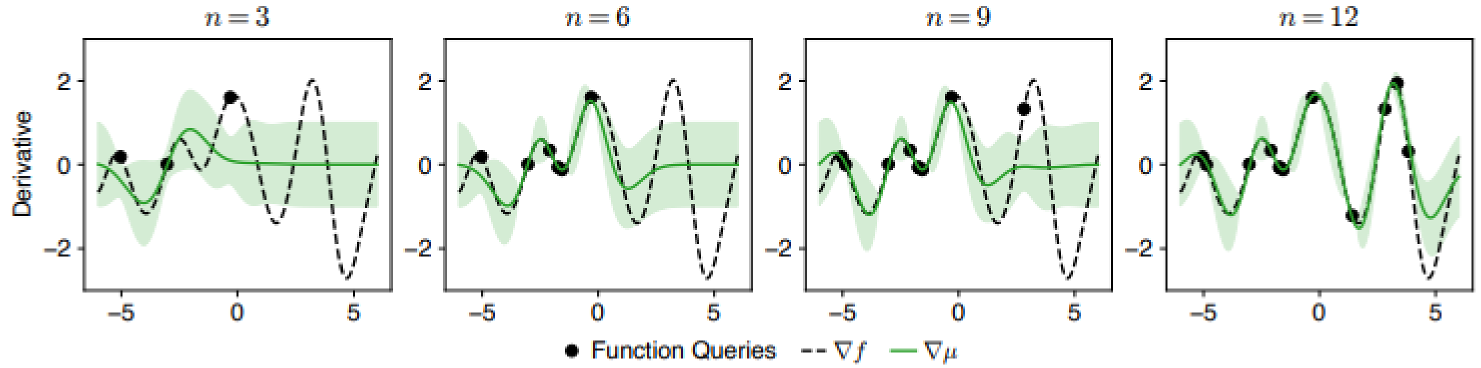


Figure 1: Our derived GP for derivative estimation (4) with different number n of queries. Green curve and its confidence interval denote the mean $\nabla\mu(\boldsymbol{x})$ and standard deviation of the derived GP.

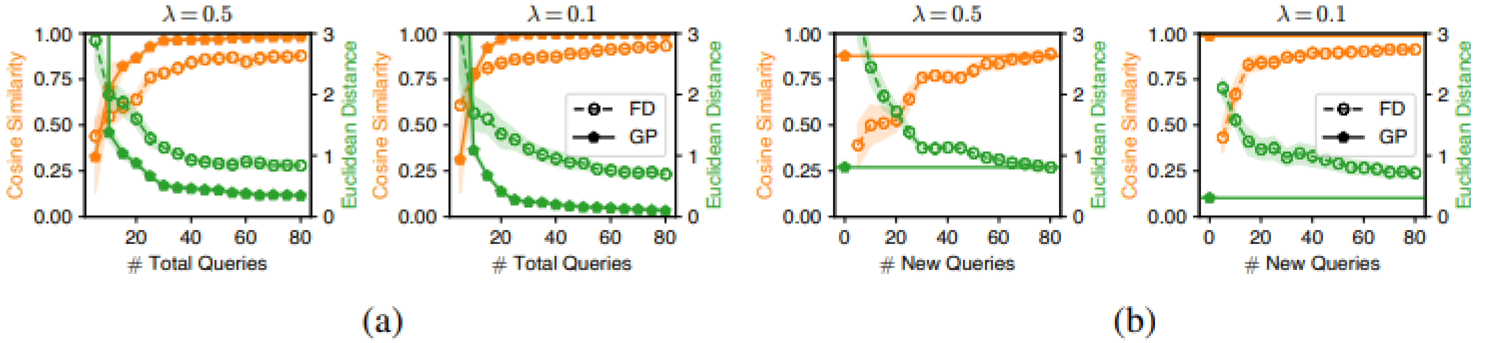


Figure 2: Comparison of the derivative estimation errors of our derived GP-based estimator (6) (GP) and the FD estimator, measured by cosine similarity (larger is better) and Euclidean distance (smaller is better). Each curve is the mean ± standard error from five independent runs.

GD provides a good estimation of true gradient

GP is four times query efficient than FD

Experiment—Black box attack

Table 1: Comparison of the number of required queries to achieve a successful black-box adversarial attack. Every entry represents mean ± standard deviation from five independent runs.

Dataset	Metric	GLD	RGF	PRGF	TuRBO-1	TuRBO-10	ZoRD
MNIST	# Queries	1780±222	1192±260	1236±145	654±70	747±60	**248±50**
	Speedup	7.2×	4.8×	5.0×	2.6×	3.0×	**1.0**×
CIFAR-10	# Queries	964±175	3622±1155	4133±1525	638±108	708±105	**384±59**
	Speedup	2.5×	9.4×	10.8×	1.7×	1.8×	**1.0**×

Queries efficient in both theoretical and experimental levels

Pros & Cons

Pros

- A good estimation of gradient, with proofs
- Query much more efficient
  - Trajectory-informed Derivative Estimation
  - Dynamic Virtual Updates

Cons

- Variance matrix inverse, high cost
- Did not discus the case when r = 1, just assume r < 1
- In real experiments, the choice of kernel function needs experience

$$\nabla \mu_{t-1}(\boldsymbol{x}) \triangleq \partial_{\boldsymbol{z}} \boldsymbol{k}_{t-1}(\boldsymbol{z})^{\top} \boxed{\left(\mathbf{K}_{t-1} + \sigma^2 \mathbf{I}\right)^{-1}} \boldsymbol{y}_{t-1} \Big|_{\boldsymbol{z}=\boldsymbol{x}}$$

Conclusion

- Two methods for ZO optimization, but there are more
- Two important ideas
- Query efficient
- High cost in matrix inverse, not complete proofs

Main References

- Liu, Sijia, et al. "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications." *IEEE Signal Processing Magazine* 37.5 (2020): 43-54.
- Chen, Pin-Yu, et al. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models." *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 2017.
- Nesterov, Yurii, and Vladimir Spokoiny. "Random gradient-free minimization of convex functions." *Foundations of Computational Mathematics* 17 (2017): 527-566.
- https://distill.pub/2019/visual-exploration-gaussian-processes/
- https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html
- https://peterroelants.github.io/posts/gaussian-process-kernels/

Thank you!
Q&A