LAMDA
Learning And Mining from DatA

Stochastic, Non-Convex and Non-Smooth Optimization via Online-to-Non-convex Conversion

Presented by Yan-Feng Xie

2024.09.21

Outline

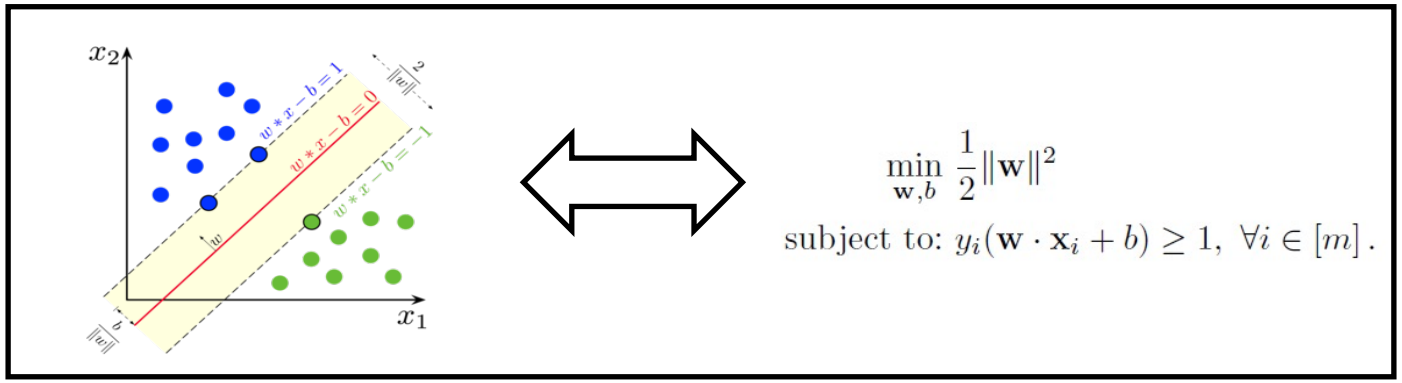
- Background
- Non-smooth optimization: Locally linearized
- Standard O2NC
- Exp-O2NC

Outline

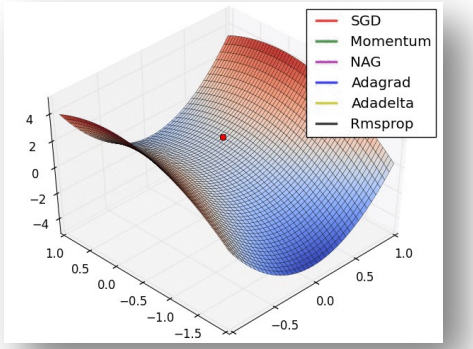
- Background
- Non-smooth optimization: Locally linearized
- Standard O2NC
- Exp-O2NC

Motivation

- Optimization is ***essential*** for training machine learning models



- However, classic theories fall short for **neural network (NN) optimization**
  - ❑ NN is ***non-convex***
  - ❑ feedback is ***stochastic*** (training with mini-batch)



Formalization

- Study ***non-convex*** and ***stochastic*** optimization

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$$

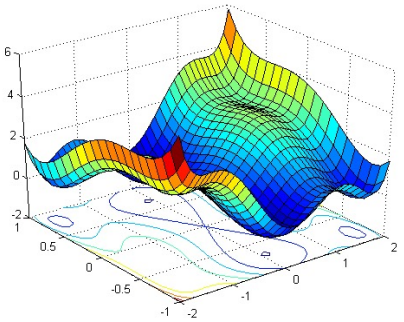
  - $F(\mathbf{x})$ is non-convex

  - $F(\mathbf{x}) = \mathbb{E}_\xi[f(\mathbf{x};\xi)]; \nabla F(\mathbf{x}) = \mathbb{E}_\xi[\nabla f(\mathbf{x};\xi)]$

- Convergence rate to ***stationary point*** s.t. $\|\nabla F(\mathbf{x}_\star)\| = 0$

  - slightly relax to ϵ-approximation point

$$\|\nabla F(\mathbf{x}_\star)\| \leq \epsilon$$



Results under Smoothness

- (Global) Smoothness: $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$

  - convergence rate: $\mathcal{O}\left(\frac{1}{N^{\frac{1}{4}}}\right)$ - algorithm: SGD

- Second-order Smoothness: $\|\nabla^2 F(\mathbf{x}) - \nabla^2 F(\mathbf{y})\| \leq \rho\|\mathbf{x} - \mathbf{y}\|$

  - convergence rate: $\mathcal{O}\left(\frac{1}{N^{\frac{1}{3.5}}}\right)$ - algorithm: SGD

- Mean-squared Smoothness: $\mathbb{E}_\xi[\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{y};\xi)\|^2] \leq L\|\mathbf{x} - \mathbf{y}\|^2$

  - convergence rate: $\mathcal{O}\left(\frac{1}{N^{\frac{1}{3}}}\right)$ - algorithm: SPIDER (SGD + variance-reduction + normalized)

above results are minimax optimal

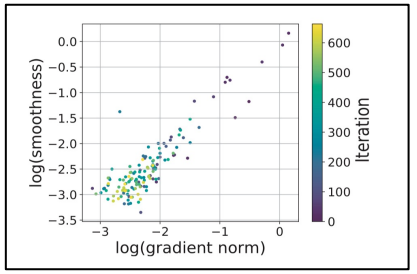
Smoothness

- Assumed (global) smoothness is ***unrealistic*** in NN opt.

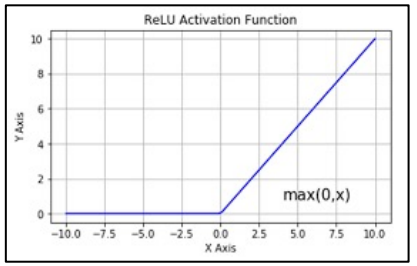
  - Local smoothness: $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq \ell\left(\|\nabla F(\mathbf{x})\|\right) \cdot \|\mathbf{x} - \mathbf{y}\|$

  - Non-smoothness: Do ***not*** assume smoothness at all!

    our topic today!



LSTM training



ReLU

mean-squared | second-order | smooth | generalized smooth | non-smooth

Strong → Weak

$\mathcal{O}(\frac{1}{N^{1/3}})$ $\mathcal{O}(\frac{1}{N^{1/3.5}})$ $\mathcal{O}(\frac{1}{N^{1/4}})$ $\mathcal{O}(\frac{1}{N^{1/4}})$ $\mathcal{O}(\frac{1}{(N\delta)^{1/3}})$

weaker measure

Outline

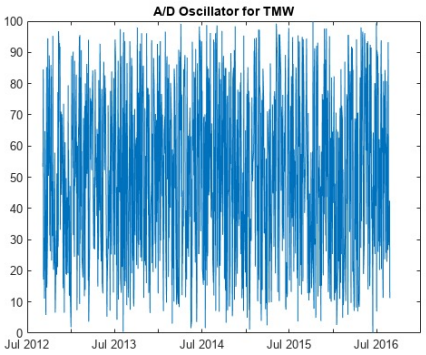
- Background
- Non-smooth optimization: Locally linearized
- Standard O2NC
- Exp-O2NC

Step into Non-smoothness

Learning And Mining from DatA

- Optimization without any assumptions can be **NP-hard**!
  - *e.g.* 0-1 optimization, combinatorial optimization

we expect linearity at least in local!



oscillation

- Let see how previous methods develop:

$$F(\mathbf{y}) = F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top \nabla^2 F(\xi)(\mathbf{y} - \mathbf{x})$$

$$\underbrace{\leq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2}_{U_{\mathbf{x}}(\mathbf{y})\ \text{(uniformly for any } \mathbf{y})}$$

$$\mathbf{x}_+ = \arg\min_{\mathbf{y}} U_{\mathbf{x}}(\mathbf{y})$$

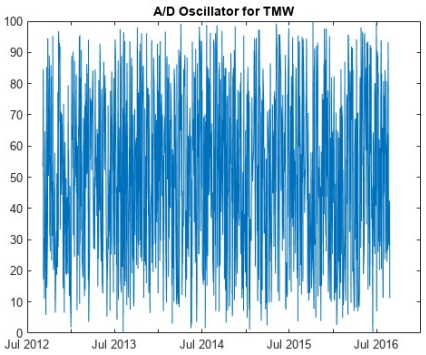
$$\underbrace{F(\mathbf{x}_+) \leq U_{\mathbf{x}}(\mathbf{x}_+) \leq U_{\mathbf{x}}(\mathbf{x}) = F(\mathbf{x})}_{F(\mathbf{x}_+) \leq F(\mathbf{x})}$$

Step into Non-smoothness

LAMDA
Learning And Mining from DatA

- Optimization without any assumptions can be **NP-hard**!
  - *e.g.* 0-1 optimization, combinatorial optimization

we expect linearity at least in local!



oscillation

- For non-smooth setting, we do ***not*** have ***second-order*** estimation!

Analogously: ***assume we have an estimator***

$$F(\mathbf{y}) = \underbrace{F(\mathbf{x}) + \langle \nabla F(\xi), \mathbf{y} - \mathbf{x} \rangle}_{U_{\mathbf{x}}(\mathbf{y})}$$

match our expectation

$U_{\mathbf{x}}(\mathbf{y})$ (uniformly for any $\mathbf{y}$?)

troublesome

$$\mathbf{x}_+ = \arg\min_{\mathbf{y}} U_{\mathbf{x}}(\mathbf{y})$$

ξ lies between $\mathbf{y}, \mathbf{x}$, while $\mathbf{y}$ is ***to be determined***!

Can we provide $\mathbf{y}$ before receive gradient?

Optimal, Stochastic, Non-smooth, Non-convex Optimization through Online-to-Non-convex Conversion, ICML'23

Components

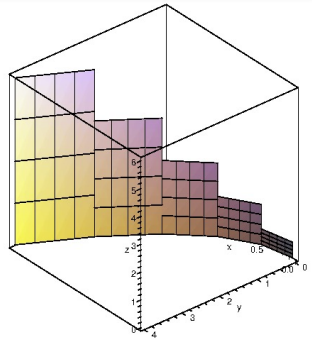
- **Challenge I**: Gradient estimator

well-behaved assumption: $F(\mathbf{y}) = F(\mathbf{x}) + \int_0^1 \langle \nabla F(\mathbf{x} + t(\mathbf{y} - \mathbf{x})), \mathbf{y} - \mathbf{x} \rangle dt$

- $\nabla F(\mathbf{x})$ is *integrable* along any line, thus, $\nabla F(\mathbf{x})$ *changes slowly* in local

- sufficient condition: *locally lipschitz, differentiable*

match our expectation



- estimator: $g_{\mathbf{x},\mathbf{y}} = \nabla f(\mathbf{x} + s(\mathbf{y} - \mathbf{x}); \xi),\ s \sim \text{Uni}[0, 1]$

$$\mathbb{E}[\langle g_{\mathbf{x},\mathbf{y}}, \mathbf{y} - \mathbf{x} \rangle] = \int_0^1 \int_{\xi \in \Omega} \langle \nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}); \xi), \mathbf{y} - \mathbf{x} \rangle d\xi dt$$

$$= \int_0^1 \langle \nabla F(\mathbf{x} + t(\mathbf{y} - \mathbf{x})), \mathbf{y} - \mathbf{x} \rangle dt = F(\mathbf{y}) - F(\mathbf{x})$$

Components

- **Challenge I**: Gradient estimator

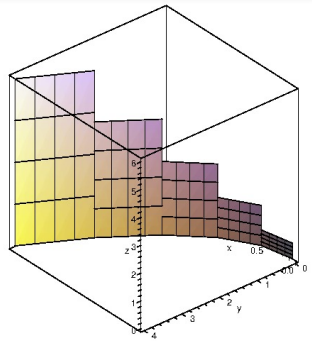
well-behaved assumption: $F(\mathbf{y}) = F(\mathbf{x}) + \int_0^1 \langle \nabla F(\mathbf{x} + t(\mathbf{y} - \mathbf{x})), \mathbf{y} - \mathbf{x} \rangle dt$

- $\nabla F(\mathbf{x})$ is *integrable* along any line, thus, $\nabla F(\mathbf{x})$ *changes slowly* in local

- sufficient condition: *locally lipschitz, differentiable*

match our expectation

- estimator: $g_{\mathbf{x},\mathbf{y}} = \nabla f(\mathbf{x} + s(\mathbf{y} - \mathbf{x}); \xi),\ s \sim \text{Uni}[0, 1]$

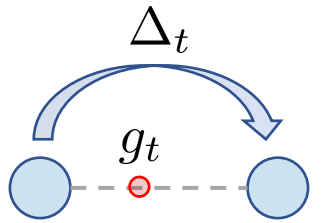


- **Challenge II**: Update and estimation conflict

- Online Learning *naturally* resolves it!

- online learner provide update Δ_t
- $\mathbf{x}_t = \mathbf{x}_{t-1} + \Delta_t$
- design $g_t = \nabla f(\mathbf{x}_t + s\Delta_t; \xi_t), s \sim U[0, 1]$
- send g_t to online learner

online learner



submitted to env. $\mathbf{x}_{t-1}$ $\mathbf{x}_t$

Outline

- Background
- Non-smooth optimization: Locally linearized
- Standard O2NC
- Exp-O2NC

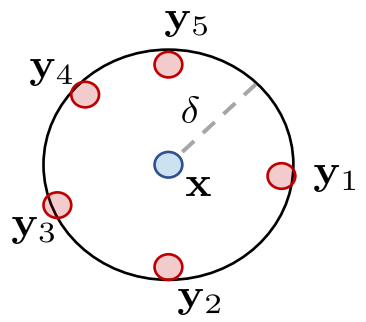
O2NC Framework

- $(\delta, \epsilon)-$Goldstein stationary point

If there exists $\mathbf{y}$ distributed around $\mathbf{x}$, with $\|\mathbf{y}-\mathbf{x}\| \leq \delta$ and $\mathbb{E}[\mathbf{y}] = \mathbf{x}$, such that

$$\|\mathbb{E}_{\mathbf{y}}[\nabla F(\mathbf{y})]\| \leq \epsilon$$

then we call $\mathbf{x}$ is a (δ, ϵ)-Goldstein stationary point



- ***E.g.***, if we find T points $\{\mathbf{y}_i\}_{i=1}^T$ such:

(i) $\|\frac{1}{T}\sum_i \nabla F(\mathbf{y}_i)\| \leq \epsilon$; (ii) $\|\mathbf{y}_i - \bar{\mathbf{y}}\| \leq \delta \ \ \forall i \in [T]$;

then $\bar{\mathbf{y}}$ is a Goldstein stationary point.

How to optimize this measure within online learning scheme?

O2NC Framework

Algorithm 1 Online-to-Non-Convex Conversion

Input: Initial point $\mathbf{x}_0$, $K \in \mathbb{N}$, $T \in \mathbb{N}$, online learning algorithm $\mathcal{A}$.
Set $M = K \cdot T$
for $n = 1 \dots M$ **do**
  Get $\mathbf{\Delta}_n$ from $\mathcal{A}$
  Set $\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{\Delta}_n$
  Generate $s_n \in [0, 1]$ // usually uniformly random, see Theorem statements for precise settings.
  Set $\mathbf{w}_n = \mathbf{x}_{n-1} + s_n \mathbf{\Delta}_n$
  Sample random $\mathbf{z}_n$
  Generate gradient $\mathbf{g}_n = \textsc{Grad}(\mathbf{w}_n, \mathbf{z}_n)$
  Send $\mathbf{g}_n$ to $\mathcal{A}$ as gradient
end for
Set $\mathbf{w}_t^k = \mathbf{w}_{(k-1)T+t}$ for $k = 1, \dots, K$ and $t = 1, \dots, T$
Set $\overline{\mathbf{w}}^k = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}_t^k$ for $k = 1, \dots, K$
Return $\{\overline{\mathbf{w}}^1, \dots, \overline{\mathbf{w}}^K\}$

- How to find a series of points **y ?**
  - Split into ***K*** epochs, each epoch contains ***T*** iterations, M = K * T
- How to ensure they are closed?
  - Project Δ_n into bounded domain with radius D small enough

  Can we employ unbounded OGD?
- Why use mid point $\mathbf{w}_n$ as final submit?
  - To ensure $\mathbb{E}[\nabla f(\mathbf{w}_n; \xi_n)] = \mathbb{E}[\nabla F(\mathbf{w}_n)]$

Analysis of O2NC

Lemma 1 *Denoted by* $\nabla_n = \int_0^1 F(\mathbf{x}_{n-1} + t\Delta_n)dt$, *O2NC ensures:*

$$F(\mathbf{x}_M) = F(\mathbf{x}_0) + \sum_{n=1}^{M} \langle \mathbf{g}_n, \Delta_n - \mathbf{u}_n \rangle + \sum_{n=1}^{M} \langle \nabla_n - \mathbf{g}_n, \Delta_n \rangle + \sum_{n=1}^{M} \langle \mathbf{g}_n, \mathbf{u}_n \rangle$$

where $\mathbf{u}_n$ is comparators to be determined

Proof: By well-behavedness: $F(\mathbf{x}_n) = F(\mathbf{x}_{n-1}) + \langle \nabla_n, \Delta_n \rangle$

Summing over n: $F(\mathbf{x}_M) = F(\mathbf{x}_0) + \sum_{n=1}^{M} \langle \nabla_n, \Delta_n \rangle$

$$= F(\mathbf{x}_0) + \sum_{n=1}^{M} \langle \mathbf{g}_n, \Delta_n - \mathbf{u}_n \rangle + \sum_{n=1}^{M} \langle \nabla_n - \mathbf{g}_n, \Delta_n \rangle + \sum_{n=1}^{M} \langle \mathbf{g}_n, \mathbf{u}_n \rangle$$

Analysis of O2NC

Lemma 1 *Denoted by* $\nabla_n = \int_0^1 F(\mathbf{x}_{n-1} + t\Delta_n)dt$, *O2NC ensures:*

$$F(\mathbf{x}_M) = F(\mathbf{x}_0) + \sum_{n=1}^{M}\langle \mathbf{g}_n, \Delta_n - \mathbf{u}_n\rangle + \sum_{n=1}^{M}\langle \nabla_n - \mathbf{g}_n, \Delta_n\rangle + \sum_{n=1}^{M}\langle \mathbf{g}_n, \mathbf{u}_n\rangle$$

where $\mathbf{u}_n$ is comparators to be determined

Lemma 2 *Taking expectation over* s_n, ξ_n, *O2NC ensures:*

$$-\mathbb{E}\left[\sum_{n=1}^{M}\langle \nabla F(\mathbf{w}_n), \mathbf{u}_n\rangle\right] \leq F(\mathbf{x}_0) - F_\star + \mathbb{E}\left[\textsc{Reg}(\mathbf{u}_1, \ldots, \mathbf{u}_M)\right] + \mathbb{E}\left[\sum_{n=1}^{M}\langle \mathbf{g}_n - \nabla F(\mathbf{w}_n), \mathbf{u}_n\rangle\right]$$

Proof: $\mathbb{E}_{s_n}\left[\mathbb{E}_{\xi_n}[\mathbf{g}_n]\right] = \mathbb{E}_{s_n}\left[\nabla F(\mathbf{x}_{n-1} + s_n\Delta_n)\right] = \nabla_n$, and Δ_n is independent of s_n and ξ_n

Guarantees

Theorem 1 *Split M into K epochs with $M = K \cdot T$, denote by $\mathbf{w}_t^k = \mathbf{w}_{(k-1)\cdot T+t}$ and $\mathbf{u}_t^k = \mathbf{u}_{(k-1)\cdot T+t}$, employ online algorithm with $\mathcal{O}(\sqrt{T(1+P_T)})$ dynamic regret as online learner, assume $\mathbb{E}[\|\mathbf{g}_n\|^2] \leq G^2$, and set,*

$$\mathbf{u}_1^k = \cdots = \mathbf{u}_T^k = -D \cdot \frac{\sum_{t=1}^T \nabla F(\mathbf{w}_t^k)}{\|\sum_{t=1}^T \nabla F(\mathbf{w}_t^k)\|},$$

we have:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{T}\|\frac{1}{T}\sum_{t=1}^{T}\nabla F(\mathbf{w}_t^k)\|\right] \lesssim \frac{F(\mathbf{x}_0)-F_\star}{DM} + \frac{GD\sqrt{KT}}{DM} + \frac{GDK\sqrt{T}}{DM}$$

*the **K** is dominated by variance term*

Proof

Lemma 2 *Taking expectation over s_n, ξ_n, O2NC ensures:*

$$-\mathbb{E}\left[\sum_{n=1}^{M}\langle\nabla F(\mathbf{w}_n),\mathbf{u}_n\rangle\right] \leq F(\mathbf{x}_0)-F_\star+\mathbb{E}\left[\textsc{Reg}(\mathbf{u}_1,\ldots,\mathbf{u}_M)\right]+\mathbb{E}\left[\sum_{n=1}^{M}\langle\mathbf{g}_n-\nabla F(\mathbf{w}_n),\mathbf{u}_n\rangle\right]$$

For the second term in LHS, directly, it can be bounded by $\mathcal{O}(GD\sqrt{KT})$

For the third term in LHS:

$$\mathbb{E}\left[\sum_{n=1}^{M}\langle\mathbf{g}_n-\nabla F(\mathbf{w}_n),\mathbf{u}_n\rangle\right]=\mathbb{E}\left[\sum_{k=1}^{K}\sum_{t=1}^{T}\langle\mathbf{g}_t^k-\nabla F(\mathbf{w}_t^k),\mathbf{u}_k\rangle\right]\leq\mathbb{E}\left[D\sum_{k=1}^{K}\left\|\sum_{t=1}^{T}\mathbf{g}_t^k-\nabla F(\mathbf{w}_t^k)\right\|\right]$$

$$\leq DK\sqrt{\mathbb{E}\left[\sum_{n=1}^{M}\|\mathbf{g}_n-\nabla F(\mathbf{w}_n)\|^2\right]} \quad \text{(Cauchy \& Independent)}$$

$$\leq GDK\sqrt{M} \qquad (\mathrm{Var}(\mathbf{g}_n)\leq\mathbb{E}[\|\mathbf{g}_n\|^2])$$

Tuning

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{T}\|\frac{1}{T}\sum_{t=1}^{T}\nabla F(\mathbf{w}_t^k)\|\right] \lesssim \frac{F(\mathbf{x}_0)-F_\star}{DM}+\frac{GK\sqrt{T}}{M}$$

To ensure $\|\bar{\mathbf{w}}^k - \mathbf{w}_t^k\| \leq \delta$, where $\bar{\mathbf{w}}^k = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t^k$, set:

$$D = \frac{\delta}{T}, \text{ i.e., } \|\Delta_n\| \leq \frac{\delta}{T}$$

Notice that $K = \frac{M}{T}$, LHS becomes

$$\frac{T(F(\mathbf{x}_0)-F_\star)}{\delta M}+\frac{G}{\sqrt{T}}$$

Best tuning $T = \mathcal{O}((\frac{\delta M}{F(\mathbf{x}_0)-F_\star})^{2/3})$ leads to $\mathcal{O}(1/(\delta M)^{\frac{1}{3}})$ convergence rate

Summary

- Concrete algorithm:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \Delta_n$$

$$\mathbf{g}_n = \nabla f(\mathbf{x}_{n-1} + s_n \Delta_n; \xi_n)$$ *not natural*

$$\Delta_{n+1} = \mathrm{Clip}_D(\Delta_n - \eta \mathbf{g}_n)$$ *clipping is essential*

- Exploit negative terms?

- For deterministic and smooth setting, $\mathbf{g}_t = \nabla F(\mathbf{x}_t)$

$$\sum \|\nabla F(\mathbf{x}_n) - \nabla F(\mathbf{x}_{n-1})\|^2 - \sum \|\Delta_n - \Delta_{n-1}\|^2 \leq L^2 \sum \|\Delta_n\|^2 - \sum \|\Delta_n - \Delta_{n-1}\|^2$$

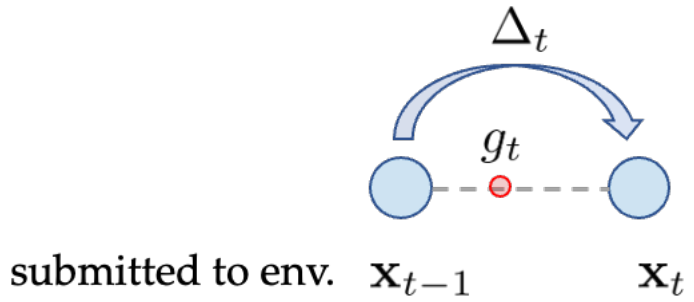
Outline

- Background
- Non-smooth optimization: Locally linearized
- Standard O2NC
- Exp-O2NC

Motivation

- Gradient evaluated at mid-point between $\mathbf{x}_n$ and $\mathbf{x}_{n-1}$

- online learner provide update Δ_t
- $\mathbf{x}_t = \mathbf{x}_{t-1} + \Delta_t$
- design $g_t = \nabla f(\mathbf{x}_t + s\Delta_t; \xi_t), s \sim U[0,1]$
- send g_t to online learner



Can we estimate at submitted point?

- Clipping for update Δ_n is required

  - This is to ensure $\|\bar{\mathbf{w}}^k - \mathbf{w}_t^k\| \leq \delta$, by requiring $\|\Delta_n\| \leq \frac{\delta}{T}$

How to support larger step update?

Random Scaling and Momentum for Non-smooth Non-convex Optimization, ICML'24

Gradient Estimators

- Close look at "backward" gradient estimation:

$$F(\mathbf{x}_t+\Delta_t)-F(\mathbf{x}_t)=\int_0^1\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle ds=\mathbb{E}_s[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]$$

only randomness

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim U[0,1]$)?

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_s[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)] && \textit{the decision is randomized!}\\ &=\int_0^1\int_0^1\langle\nabla F(\mathbf{x}_t+ts\Delta_t),s\Delta_t\rangle p(s)dtds && (p(s)=1)\\ &=\int_0^1\left(\int_0^1\langle\nabla F(\mathbf{x}_t+ts\Delta_t),s\Delta_t\rangle dt\right)p(s)ds\\ &=\int_0^1\left(\int_0^s\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\right)p(s)ds && (d\tau=sdt)\\ &=\int_0^1\left(\int_0^1\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle\cdot\mathbf{1}\{\tau\le s\}d\tau\right)p(s)ds\end{aligned}$$

Gradient Estimators

- Close look at "backward" gradient estimation:

$$F(\mathbf{x}_t+\Delta_t)-F(\mathbf{x}_t)=\int_0^1\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle ds=\mathbb{E}_s[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]$$

only randomness

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim U[0,1]$)?

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_s[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]\\&=\int_0^1\left(\int_0^1\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle\cdot\mathbf{1}\{\tau\le s\}d\tau\right)p(s)ds\\&=\int_0^1\left(\int_0^1\mathbf{1}\{\tau\le s\}p(s)ds\right)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\\&=\int_0^1(1-\tau)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\\&=\mathbb{E}_{\tau\sim U[0,1]}[(1-\tau)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle]\end{aligned}$$

the decision is randomized!

Gradient Estimators

- Close look at "backward" gradient estimation:

$$F(\mathbf{x}_t+\Delta_t)-F(\mathbf{x}_t)=\int_0^1\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle ds=\mathbb{E}_s[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]$$

only randomness

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim U[0,1]$)?

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_s[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]\\&=\int_0^1\left(\int_0^1\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle\cdot\mathbf{1}\{\tau\le s\}d\tau\right)p(s)ds\\&=\int_0^1\left(\int_0^1\mathbf{1}\{\tau\le s\}p(s)ds\right)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\\&=\int_0^1(1-\tau)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\\&=\mathbb{E}_{s\sim U[0,1]}[(1-s)\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]\end{aligned}$$

the decision is randomized!

change symbol to s

$(g_t=(1-s)\cdot\nabla f(\mathbf{x}_{t+1}))$

$(\mathbb{E}[g_t]\neq\nabla F(\mathbf{x}_{t+1}))$

Gradient Estimators

- Close look at "backward" gradient estimation:

$$F(\mathbf{x}_t+\Delta_t)-F(\mathbf{x}_t)=\int_0^1\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle ds=\mathbb{E}_s[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]$$

only randomness

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim U[0,1]$)?

$$\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]=\mathbb{E}_s[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]$$

the decision is randomized!

$$=\int_0^1\left(\int_0^1\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle\cdot\mathbf{1}\{\tau\le s\}d\tau\right)p(s)ds$$

$$=\int_0^1\left(\int_0^1\mathbf{1}\{\tau\le s\}p(s)ds\right)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau$$

we are expected $\int_\tau^\infty p(s)ds=p(\tau)$ ⇒ *exponential distribution*

Gradient Estimators

- Close look at "backward" gradient estimation:

$$F(\mathbf{x}_t+\Delta_t)-F(\mathbf{x}_t)=\int_0^1\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle ds=\mathbb{E}_s[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]$$

only randomness

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim U[0,1]$)?

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_s[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]\\&=\mathbb{E}_{s\sim U[0,1]}[(1-s)\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]\end{aligned}$$

the decision is randomized!

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim \mathrm{Exp}[\lambda]$)? $(p(s)=\lambda\exp(-\lambda s))$

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_{s\sim\mathrm{Exp}(\lambda)}[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]\\&=\int_0^\infty\left(\int_0^\infty\mathbf{1}\{\tau\le s\}p(s)ds\right)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\\&=\int_0^\infty\frac{p(\tau)}{\lambda}\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\end{aligned}$$

Gradient Estimators

- Close look at "backward" gradient estimation:

$$F(\mathbf{x}_t+\Delta_t)-F(\mathbf{x}_t)=\int_0^1\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle ds=\mathbb{E}_s[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]$$

only randomness

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim U[0,1]$)?

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_s[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]\\&=\mathbb{E}_{s\sim U[0,1]}[(1-s)\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]\end{aligned}$$

the decision is randomized!

- What if we take gradient at $x_t+s\Delta_t$ ($s\sim \mathrm{Exp}[\lambda]$)? $(p(s)=\lambda\exp(-\lambda s))$

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})-F(\mathbf{x}_t)]&=\mathbb{E}_{s\sim\mathrm{Exp}(\lambda)}[F(\mathbf{x}_t+s\Delta_t)-F(\mathbf{x}_t)]\\&=\int_0^\infty\left(\int_0^\infty\mathbf{1}\{\tau\le s\}p(s)ds\right)\langle\nabla F(\mathbf{x}_t+\tau\Delta_t),\Delta_t\rangle d\tau\\&=\tfrac{1}{\lambda}\mathbb{E}_{s\sim\mathrm{Exp}(\lambda)}[\langle\nabla F(\mathbf{x}_t+s\Delta_t),\Delta_t\rangle]\qquad(g_t=\tfrac{1}{\lambda}\cdot\nabla F(\mathbf{x}_{t+1}))\end{aligned}$$

Relaxed Measure

- Consider a ***softer*** distance restriction

If there exists $\mathbf{y}$ distributed around $\mathbf{x}$, with $\mathbb{E}[\mathbf{y}] = \mathbf{x}$, such that

$$\|\mathbb{E}_{\mathbf{y}}[\nabla F(\mathbf{y})]\| + c \cdot \mathbb{E}\left[\|\mathbf{y} - \mathbf{x}\|^2\right] \leq \epsilon$$

then we call $\mathbf{x}$ is a (c, ϵ)-stationary point

- by choosing $c = \epsilon/\delta^2$, a (c, ϵ)-stationary point is also a (δ, ϵ)-Goldstein stationary point, ***except for*** $\|\mathbf{y} - \mathbf{x}\| \leq \delta$ is ***relaxed to*** $\mathbb{E}\|\mathbf{y} - \mathbf{x}\|^2 \leq \delta^2$

Lemma 1 *Suppose $F(\mathbf{x})$ is G-Lipschitz. Then a (c, ϵ)-stationary point is also a (δ, ϵ')-Goldstein point where $\epsilon' = (1 + \frac{2G}{c\delta^2})\epsilon$*

Exponentiated O2NC

- Exponetially distributed points:

given $\{\mathbf{x}_n\}_{n=1}^N$, random variable $\mathbf{y}$ is chosen so that $\mathbb{P}(\mathbf{y} = \mathbf{x}_n) \propto \beta^{N-n}$

- $\bar{\mathbf{x}}_N = \sum_{n=1}^N p_n \mathbf{x}_n$, where $p_n = \beta^{N-n} \frac{1-\beta}{1-\beta^N}$

- $\bar{\mathbf{x}}_N$ is more ***closed*** to $\mathbf{x}_N$, while $\mathbf{y}$ is more ***likely*** to be $\mathbf{x}_N$!

  thus $\mathbb{E}[\|\mathbf{y} - \bar{\mathbf{x}}_N\|]^2$ can be controlled

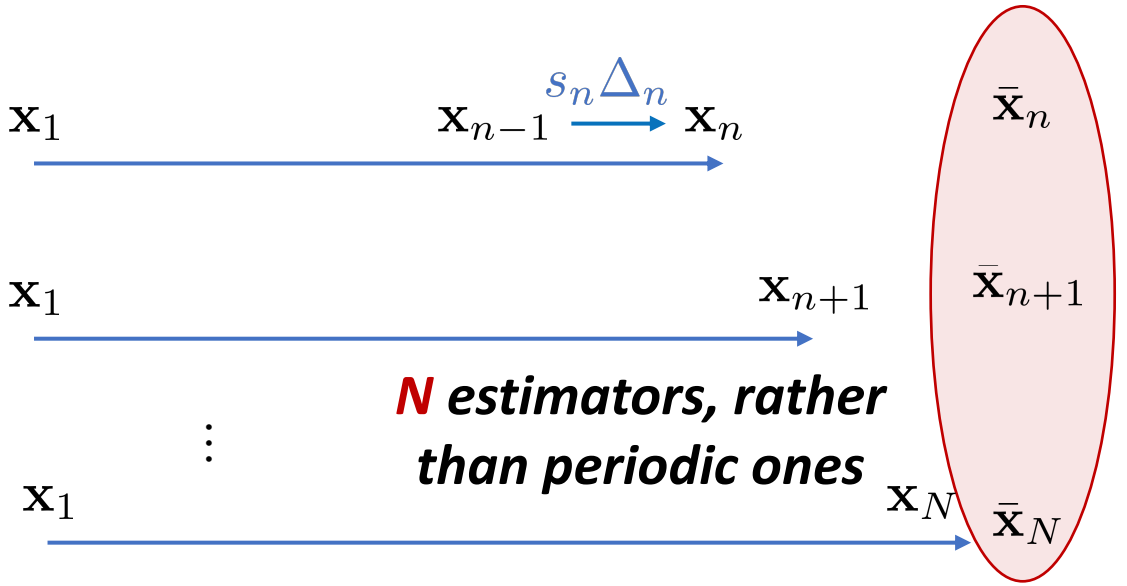
In my understanding, recall $\|\mathbf{x}_i - \mathbf{x}_j\| = \|\sum_{k=i+1}^j s_k \Delta_k\|$

$$\mathbb{E}\|\mathbf{y} - \bar{\mathbf{x}}_N\|^2 \approx \underbrace{\sum_{n=1}^N n\beta^n}_{\text{constant}} \cdot \underbrace{\|\Delta_n\|^2}_{\text{regularizer}} + \text{others}$$

Exponentiated O2NC

Algorithm 2 Exponentiated O2NC

1: **Input:** OCO algorithm $\mathcal{A}$, initial state $\boldsymbol{x}_0$, parameters $N \in \mathbb{N}, \beta \in (0,1)$, regularizers $\mathcal{R}_n(\Delta)$.
2: **for** $n \leftarrow 1, 2, \ldots, N$ **do**
3: Receive Δ_n from $\mathcal{A}$.
4: Update $\boldsymbol{x}_n \leftarrow \boldsymbol{x}_{n-1} + s_n \Delta_n$, where $s_n \sim \mathrm{Exp}(1)$ i.i.d.
5: Compute $\boldsymbol{g}_n \leftarrow \nabla f(\boldsymbol{x}_n, z_n)$.
6: Send loss $\ell_n(\Delta) = \langle \beta^{-n} \boldsymbol{g}_n, \Delta \rangle + \mathcal{R}_n(\Delta)$ to $\mathcal{A}$.
// For output only (does *not* affect training):
7: Update $\overline{\boldsymbol{x}}_n = \frac{\beta - \beta^n}{1-\beta^n} \overline{\boldsymbol{x}}_{n-1} + \frac{1-\beta}{1-\beta^n} \boldsymbol{x}_n$.
Equivalently, $\overline{\boldsymbol{x}}_n = \sum_{t=1}^{n} \beta^{n-t} \boldsymbol{x}_t \cdot \frac{1-\beta}{1-\beta^n}$.
8: **end for**
9: Output $\overline{\boldsymbol{x}} \sim \mathrm{Unif}(\{\overline{\boldsymbol{x}}_n : n \in [N]\})$.



Invent Exp-O2NC

$$\left\| \sum_{n=1}^{N} p_n \nabla F(\mathbf{x}_n) \right\|, \quad \text{where } p_n \propto \beta^{N-n}$$

O2NC constructs this term from: $\mathbb{E}\left[\sum_{n=1}^{N} \langle \nabla F(\mathbf{x}_n), \mathbf{u} \rangle\right]$

there is missing something: $\mathbb{E}\left[\sum_{n=1}^{N} \langle p_n \nabla F(\mathbf{x}_n), \mathbf{u} \rangle\right]$

⇨ suggest input for OL: $\mathbf{g}_n = \beta^{-n} \nabla f(\mathbf{x}_n)$

Analysis of Exp-O2NC

- Start by well-behavedness:

$$\mathbb{E}\left[F(\mathbf{x}_n) - F(\mathbf{x}_{n-1})\right] = \mathbb{E}\left[\langle \nabla F(\mathbf{x}_n), \Delta_n \rangle\right] = \mathbb{E}\left[\langle \nabla F(\mathbf{x}_n) - \mathbf{g}_n, \Delta_n \rangle + \langle \mathbf{g}_n, \Delta_n - \mathbf{u}_n \rangle + \langle \mathbf{g}_n, \mathbf{u}_n \rangle\right]$$

$$= \mathbb{E}\left[\langle \mathbf{g}_n, \Delta_n - \mathbf{u}_n \rangle + \langle \mathbf{g}_n - \nabla F(\mathbf{x}_n), \mathbf{u}_n \rangle + \langle \nabla F(\mathbf{x}_n), \mathbf{u}_n \rangle\right]$$

missing weight?

$$\beta^{N-n}\mathbb{E}\left[F(\mathbf{x}_n) - F(\mathbf{x}_{n-1})\right] = \beta^N \cdot \mathbb{E}\left[\langle \beta^{-n}\mathbf{g}_n, \Delta_n - \mathbf{u}_n \rangle + \beta^{-n}\langle \mathbf{g}_n - \nabla F(\mathbf{x}_n), \mathbf{u}_n \rangle + \langle \beta^{-n}\nabla F(\mathbf{x}_n), \mathbf{u}_n \rangle\right]$$

- So far so good, but ***recall*** our goal: $\|\mathbb{E}_{\mathbf{y}}[\nabla F(\mathbf{y})]\| + c \cdot \mathbb{E}\left[\|\mathbf{y} - \mathbf{x}\|^2\right] \leq \epsilon$
  - we should slightly modify the loss function: $\ell_n(\Delta) = \langle \beta^{-n}\mathbf{g}_n, \Delta \rangle + \mathcal{R}_n(\Delta)$

    where we set $\mathcal{R}_n(\Delta) = \frac{\mu_n}{2}\|\Delta\|^2$ eventually

Analysis of Exp-O2NC

$$\beta^{N-n}\mathbb{E}\left[F(\mathbf{x}_n) - F(\mathbf{x}_{n-1})\right] = \beta^N \cdot \mathbb{E}\left[\langle \beta^{-n}\mathbf{g}_n, \Delta_n - \mathbf{u}_n\rangle + \beta^{-n}\langle \mathbf{g}_n - \nabla F(\mathbf{x}_n), \mathbf{u}_n\rangle + \langle \beta^{-n}\nabla F(\mathbf{x}_n), \mathbf{u}_n\rangle\right]$$

For the regret term:

$$\beta^N \cdot \mathbb{E}\left[\langle \beta^{-n}\mathbf{g}_n, \Delta_n - \mathbf{u}_n\rangle\right] = \beta^N \cdot \mathbb{E}\underbrace{\left[\langle \beta^{-n}\mathbf{g}_n, \Delta_n - \mathbf{u}_n\rangle + \mathcal{R}_n(\Delta_n) - \mathcal{R}_n(\mathbf{u}_n)\right]}_{\text{instantaneous regret for OMD with composite loss}} + \beta^N\mathbb{E}\left[\mathcal{R}_n(\mathbf{u}_n) - \mathcal{R}_n(\Delta_n)\right]$$

Notice that we ***only*** consider the sequence from 1 to N; there are N sequences, e.g., $[1, 2], \ldots, [1, N-1], [1, N]$. Sum them up:

$$\mathbb{E}\sum_{n=1}^{N}\sum_{t=1}^{n}\beta^n(-\mathcal{R}_t(\Delta_t) + \mathcal{R}_t(\mathbf{u}_t)) \approx \mathbb{E}\sum_{t=1}^{N}\beta^n(-\frac{\mu_t}{2}\|\Delta_t\|^2 + \frac{\mu_t}{2}D^2)$$

Analysis of Exp-O2NC

$$\mathbb{E}\sum_{n=1}^{N}\sum_{t=1}^{n}\beta^n(-\mathcal{R}_t(\Delta_t)+\mathcal{R}_t(\mathbf{u}_t)) \approx \mathbb{E}\sum_{t=1}^{N}\beta^n(-\frac{\mu_t}{2}\|\Delta_t\|^2+\frac{\mu_t}{2}D^2)$$

Lemma 3.2. *For any $\beta \in (0,1)$,*

$$\mathbb{E}_s\sum_{n=1}^{N}\mathbb{E}_{\boldsymbol{y}_n}\|\boldsymbol{y}_n-\overline{\boldsymbol{x}}_n\|^2 \le \sum_{n=1}^{N}\frac{12}{(1-\beta)^2}\|\Delta_n\|^2.$$

therefore:

$$\mathbb{E}\sum_{n=1}^{N}\sum_{t=1}^{n}\beta^n(-\mathcal{R}_t(\Delta_t)+\mathcal{R}_t(\mathbf{u}_t)) \approx \mathbb{E}\sum_{t=1}^{N}\beta^n(-\frac{\mu_t}{2}\|\Delta_t\|^2+\frac{\mu_t}{2}D^2) \lesssim -\mathbb{E}\sum_{n=1}^{N}\|\mathbf{y}_n-\bar{\mathbf{x}}_n\|^2+D^3$$

Recall $\|\mathbf{x}_i-\mathbf{x}_j\| = \|\sum_{k=i}^{j-1}s_k\Delta_k\|$, further, $\quad \mathbb{E}\|\mathbf{y}-\bar{\mathbf{x}}_N\|^2 \approx \sum_{n=1}^{N} n\beta^n\cdot\|\Delta_n\|^2 + \text{others}$

Insight from Exp-O2NC

- Exp-O2NC allows OL to predict under ***unconstrained*** domain:

$$\Delta_{n+1} = \arg\min_{\Delta \in \mathbb{R}^d} \underbrace{\langle \widetilde{\mathbf{g}}_t, \Delta \rangle + \frac{\mu_n}{2}\|\Delta\|^2}_{\text{composite loss}} + \underbrace{\frac{1}{2\eta_t}\|\Delta - \Delta_n\|^2}_{\text{Bregman divergence}} + \underbrace{(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t})\|\Delta\|^2}_{\text{stabilizer}}$$

- this algorithm ensures:

$$\textsc{Reg}_T(\mathbf{u}) \leq (\frac{2}{\eta_{N+1}} + \frac{\mu_{N+1}}{2})\|\mathbf{u}\|^2 + \sum_{n=1}^{N} \eta_t \|\widetilde{\mathbf{g}}_t\|^2$$

- Insight:

$$\Delta_{t+1} = \alpha\Delta_t - \beta\mathbf{g}_t \iff -\frac{1-\alpha}{\beta}\Delta_{t+1} = \alpha(-\frac{1-\alpha}{\beta}\Delta_t) + (1-\alpha)\cdot\mathbf{g}_t$$

$$m_t = -\tfrac{1-\alpha}{\beta}\Delta_t \iff m_{t+1} = \alpha m_t + (1-\alpha)\mathbf{g}_t$$

Insight from Exp-O2NC

- Though, the result seems fascinating..., indeed:

$$\Delta_{t+1} = \alpha\Delta_t - \beta\mathbf{g}_t \iff -\frac{1-\alpha}{\beta}\Delta_{t+1} = \alpha(-\frac{1-\alpha}{\beta}\Delta_t) + (1-\alpha)\cdot\mathbf{g}_t$$

$$m_t = -\tfrac{1-\alpha}{\beta}\Delta_t$$

$$\iff$$

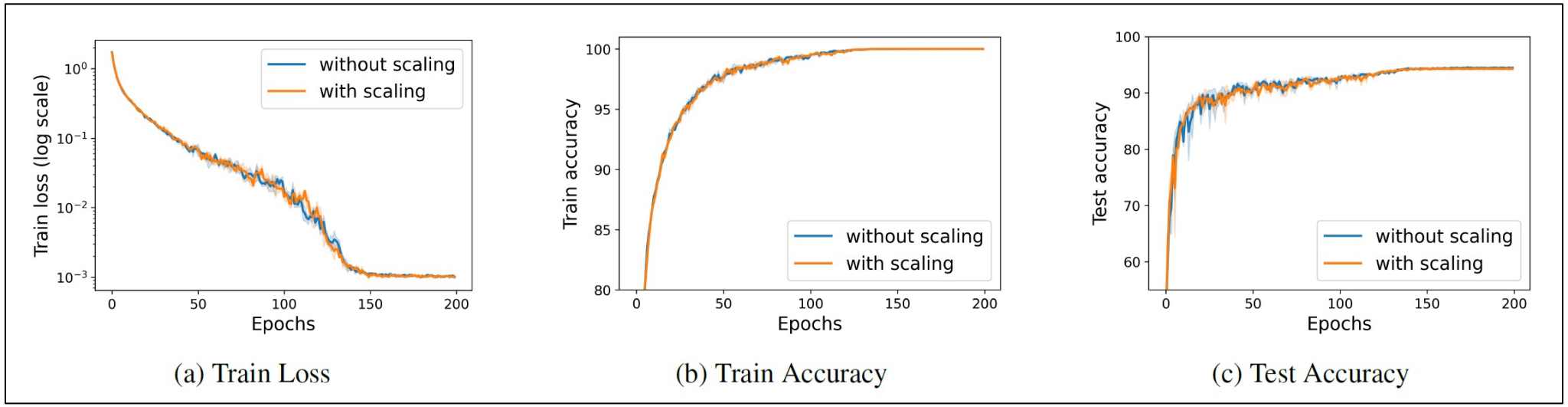
- For update:

$$m_{t+1} = \alpha m_t + (1-\alpha)\mathbf{g}_t$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + s_t\Delta_t = \mathbf{x}_t - s_t\gamma\mathbf{m}_t$$

- The O2NC reduction plays the key role; it is important to finally obtain static tuning rate.

Experiments



(a) Train Loss (b) Train Accuracy (c) Test Accuracy

verify the impact of random scaling only

Summary

- Hot plug technique to correct the querying position
- Soften measure to avoid deterministic requirements on distance
- Exponentially weighted to control the distance

Take-Home Message

- Non-smooth optimization: ***locally linearized + multi-round average***
- O2NC framework: use online algorithm to determine ***update step***
- ***Improvement*** for ICML'24: allow online algorithm to predict in unconstrained domain

Thanks!