ECE 8101: Nonconvex Optimization for Machine Learning

Lecture Note 4-2: Variance-Reduced Zeroth-Order Methods

Jia (Kevin) Liu

Associate Professor
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH, USA

Autumn 2024

Outline

In this lecture:

- Motivation of Variance-Reduced Zeroth-Order Methods
- Representative Algorithms
- Convergence Results

Finite-Sum Minimization with VR Zeroth-Order Methods

- Consider ZO methods for special case of $\min f(\mathbf{x})$: finite-sum minimization

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x})$$

  - We have studied finite-sum minimization with VR first-order methods

- Need for solving finite-sum minimization problem with ZO methods:
  - Reinforcement learning (e.g., [Fazel et al., ICML'18])
  - Non-stationary online optimization problems [Zhang et al., arXiv:2010.07378]

- We have seen that SGD-type ZO methods with noisy $\hat{f}$ have sample complexity $O(d\epsilon^{-4})$ in the last lecture

Can we do better (at least for finite-sum minimization)?

Variance Reduction in First-Order Methods

- SAG (biased) → high mem complexity. convergence rate $O(\frac{1}{k})$, sample compl. $O(N\varepsilon^{-2})$.
- SVRG → double-loop. low memory complexity. conv. rate $O(\frac{1}{k})$. samp. comp. $O(N^{\frac{2}{3}}\varepsilon^{-2})$.
- SAGA (unbiased) → high mem complexity.
- SARAH
- SPIDER/SpiderBoost [NeurIPS'19]

  SARAH, SPIDER/SpiderBoost: recursive update. conv rate $O(\frac{1}{k})$, sample comp. $O(N^{\frac{1}{2}}\varepsilon^{-2})$ (inst $N = O(\varepsilon^{-2})$).

  spider's LR: $O(\varepsilon/L)$. spiderBoost $O(1/L)$
- PAGE probabilistic ("single-loop")

We will develop their ZO counterparts

ZO-SVRG [Liu et al., NeurIPS'18]

- A zeroth-order version of SVRG
- Consider a non-convex finite-sum problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x})$$

  - $f_i \in C_L^{1,1}$ $(\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \le L\|\mathbf{x}-\mathbf{y}\|_2,\ \forall \mathbf{x},\mathbf{y}\in\mathbb{R}^d,\ \forall i \in \{1,\dots,N\})$
  - Bounded variance of stochastic gradient: $\frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|_2^2 \le \sigma^2$
- The following gradient estimations are used in [Liu, et al., NeurIPS'18]:

SSG.

RandGradEst: $\hat{\nabla} f_i(\mathbf{x}) = \frac{d}{\mu}[f_i(\mathbf{x}+\mu\mathbf{u}_i) - f_i(\mathbf{x})]\mathbf{u}_i$ 2pt

Avg-RandGradEst: $\hat{\nabla} f_i(\mathbf{x}) = \frac{d}{\mu q}\sum_{j=1}^{q}[f_i(\mathbf{x}+\mu\mathbf{u}_{i,j}) - f_i(\mathbf{x})]\mathbf{u}_{i,j}$ (q+1)-pt.

CFD. "deterministic".

CoordGradEst: $\hat{\nabla} f_i(\mathbf{x}) = \frac{1}{2\mu}\sum_{j=1}^{d}[f_i(\mathbf{x}+\mu_j\mathbf{e}_j) - f_i(\mathbf{x}-\mu_j\mathbf{e}_j)]\mathbf{e}_j$

ZO-SVRG [Liu et al., NeurIPS'18]

The ZO-SVRG Algorithm

- **Required:** Step-sizes $\{\eta_s^t\}$, epoch length T, starting point $\mathbf{x}_0 \in \mathbb{R}^d$, smoothing parameter μ, number of iterations $K = S \cdot T$, $\phi_0 = \mathbf{x}_0^0$
- **for** $s = 0, 1, 2, \ldots, S-1$

  Compute ZO full gradient estimate $\hat{\nabla} f(\phi_s)$

  for $t = 0, 1, 2, \ldots, T-1$ **then**

  Uniformly randomly pick $I_t \subset \{1, \ldots, N\}$ with $|I_t| = B$ with replacement. Compute:

$$\mathbf{v}_s^t = \frac{1}{B} \sum_{i \in I_t} [\hat{\nabla} f_i(\mathbf{x}_s^t) - \hat{\nabla} f_i(\phi_s)] + \hat{\nabla} f(\phi_s)$$

$$\mathbf{x}_s^{t+1} = \mathbf{x}_s^t - \eta_s^t \mathbf{v}_s^t$$

  all use set of $\{u_i\}$

  end for

  Let $\phi_{s+1} = \mathbf{x}_{s+1}^0 = \mathbf{x}_s^t$

  end for

  Output: $\mathbf{x}_\xi$, where ξ is picked uniformly at random from $\{0, \ldots, K-1\}$

ZO-SVRG [Liu et al., NeurIPS'18]

- Compared to FO-SVRG, the only difference is:

$$\text{FO-SVRG: } \mathbf{x}_s^{t+1} = \mathbf{x}_s^t - \eta_s^t \mathbf{v}_s^t, \; \mathbf{v}_s^t = \nabla f_{I_t}(\mathbf{x}_s^t) - \nabla f_{I_t}(\mathbf{x}_s^0) + \nabla f(\mathbf{x}_s^0)$$

$$\text{ZO-SVRG: } \mathbf{x}_s^{t+1} = \mathbf{x}_s^t - \eta_s^t \hat{\mathbf{v}}_s^t, \; \hat{\mathbf{v}}_s^t = \hat{\nabla} f_{I_t}(\mathbf{x}_s^t) - \hat{\nabla} f_{I_t}(\mathbf{x}_s^0) + \hat{\nabla} f(\mathbf{x}_s^0)$$

  where $\hat{\nabla} f_I(\mathbf{x}) = \frac{1}{B}\sum_{i \in I} \hat{\nabla} f_i(\mathbf{x})$

- Key Problem: $\hat{\nabla} f(\mathbf{x}_s^0)$ is no longer unbiased ZO gradient estimate
- Under stated assumptions, ZO-SVRG after $K = ST$ steps achieves:

$$\text{RandGradEst: } \mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] = O\left(\frac{d}{T} + \frac{1}{B}\right)$$

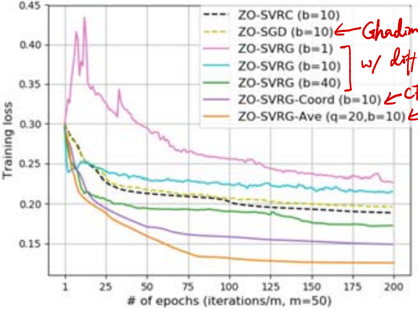
$$\text{Avg-RandGradEst: } \mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] = O\left(\frac{d}{T} + \frac{1}{B \min\{d, q\}}\right)$$

$$\text{CoordGradEst: } \mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] = O\left(\frac{d}{T}\right)$$

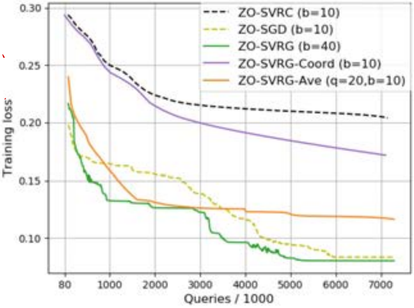
- Insight: CoordGradEst (i.e., deterministic gradient estimation) achieves same convergence rate as FO-SVRG

ZO-SVRG [Liu et al., NeurIPS'18]

- Blackbox classification problem motivated by material science:
  - A nonlinear least square problem $f_i(\mathbf{x}) = (y_i - \phi(\mathbf{x}; \mathbf{a}_i))^2$ for $i \in [N]$, where $\phi(\mathbf{x}, \mathbf{a}_i)$ is a blackbox function that only returns function value
  - $N = 1,000$ crystalline materials/compounds extracted from Open Quantum Materials Database; each compound has $d = 145$ chemical features



(a) Training loss versus iterations



(b) Training loss versus function queries

SpiderSZO [Fang et al., NeurIPS'18]

- **Required:** $n_0 = [1, \frac{30(2d+9)\sigma}{\epsilon}]$, Lipschitz constant L, epoch T, initial $\mathbf{x}_0 \in \mathbb{R}^d$, outer and inner batch-sizes B_1 and B_2, num. of iterations $K = ST$.
- **for** $k = 0, 1, 2, \ldots, K-1$

  if $\mod (k, T) = 0$ **then**

  Uniformly randomly pick $I_k \subset \{1, \ldots, N\}$ with $|I_k| = B_1$ with replacement. Compute:

$$\mathbf{v}_k = \sum_{j=1}^{d} \left(\frac{1}{B_1} \sum_{i \in I_k} \frac{[f_i(\mathbf{x}_k + \mu \mathbf{e}_j) - f_i(\mathbf{x}_k)]}{\mu} \right) \mathbf{e}_j$$

$B_1 \gg B_2$. FFD. approx. "Full Grad".

  else

  Create set of pairs $I_k = \{(i, \mathbf{u}_i)\}$ w/ $|I_k| = B_2$, where $i \sim \mathcal{U}[N]$ (with replacement) and indep. $\mathbf{u}_i \sim \mathcal{N}(0, \mathbf{I}_d)$. Compute:

$$\mathbf{v}_k = \frac{1}{B_2} \sum_{(i, \mathbf{u}_i) \in I_k} \left(\frac{f_i(\mathbf{x}_k + \mu \mathbf{u}_i) - f_i(\mathbf{x}_k)}{\mu} \mathbf{u}_i - \frac{f_i(\mathbf{x}_{k-1} + \mu \mathbf{u}_i) - f_i(\mathbf{x}_{k-1})}{\mu} \mathbf{u}_i \right) + \mathbf{v}_{k-1}$$

GSG. prev. iter.

  end if

  Let $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{v}_k$, where $\eta_k = \min(\frac{\epsilon}{L n_0 \|\mathbf{v}_k\|}, \frac{1}{2 L n_0})$

  end for

  Output: $\mathbf{x}_\xi$, where ξ is picked uniformly at random from $\{0, \ldots, K-1\}$

SpiderSZO [Fang et al., NeurIPS'18]

- Learning rate $\eta_k = \min(\frac{\epsilon}{Ln_0\|\mathbf{v}_k\|}, \frac{1}{2Ln_0})$: = O(ε)
  - Follows from normalized gradient descent (NGD) [Nesterov, Book'04]
  - Inversely proportional to norm of "gradient"

similar to FO-SPIDER.

Theorem 1 ([Fang et al., NeurIPS'18])

After $K = O(\epsilon^{-2})$ iterations, with $O(d\min\{N^{1/2}\epsilon^{-2}, \epsilon^{-3}\})$ incremental zeroth-order oracle (IZO, i.e., returning the value of $f_i(\mathbf{x})$ given $\mathbf{x}$ and i) calls, SpiderSZO ensures that:

$$\mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2] \leq 6\epsilon.$$

- This result is better than the sample complexity of [Nesterov and Spokoiny, FCM'17] by a factor of $N^{1/2}$

Improved ZO-SVRG and ZO-SPIDER [Ji et al., ICML'19]

- A tighter analysis for ZO-SVRG in [Ji et al., ICML'19]:
  - ZO-SVRG-Coord has a better convergence rate $\mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] = O(1/K)$
  - d times better than the previous analysis in [Liu et al., NeurIPS'18]
  - To achieve an ϵ-stationary point (i.e., $\mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] \leq \epsilon^2$), ZO-SVRG-Coord's function query complexity is $O(\min\{N^{2/3} d\epsilon^{-2}, d\epsilon^{-10/3}\})$

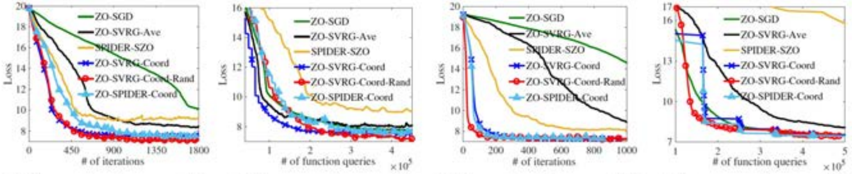
- Proof Sketch:
  1. Consider an intermediate variant of ZO-SVRG-Coord and ZO-SVRG-Ave called ZO-SVRG-Coord-Rand that uses CFD and SSG for the $\hat{\nabla} f(\phi_s)$ and $\hat{\nabla} f_i(\mathbf{x}_s^t) - \hat{\nabla} f_i(\phi_s)$ parts in $\mathbf{v}_s^t = \frac{1}{B}\sum_{i \in I_t}[\hat{\nabla} f_i(\mathbf{x}_s^t) - \hat{\nabla} f_i(\phi_s)] + \hat{\nabla} f(\phi_s)$, respectively, as opposed to [Liu et al., NeurIPS'18] that used only one type of gradient estimation at once.
  2. [Ji et al., ICML'19] showed that, although the replacement of SSG with CFD requires d more oracle calls, it achieves more accurate gradient estimation, which yields a convergence rate $\mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] = O(1/K)$. So, the convergence rate stays the same for ZO-SVRG-Coord.

Improved ZO-SVRG and ZO-SPIDER [Ji et al., ICML'19]

- A new variant of ZO-SPIDER in [Ji et al., ICML'19]: ZO-SPIDER-Coord:
  - Similar to ZO-SVRG-Coord: Use CFD instead of GSG in SpiderSZO
  - Show that ZO-SPIDER-Coord has the same convergence rate as SpiderSZO, but with a bigger size-size $\eta_k = 1/4L$ and doesn't depend on ϵ (using similar idea as in SpiderBoost)
  - With appropriate choices of learning rate, sampling radius parameters, outer batch size, ZO-SPIDER-Coord achieves a convergence rate $O(\sqrt{B_1}/K)$
  - To achieve an ϵ-stationary point (i.e., $\mathbb{E}[\|\nabla f(\mathbf{x}_\xi)\|_2^2] \leq \epsilon^2$), ZO-SVRG-Coord's function query complexity is $O(\min\{N^{1/2}d\epsilon^{-2}, d\epsilon^{-3}\})$

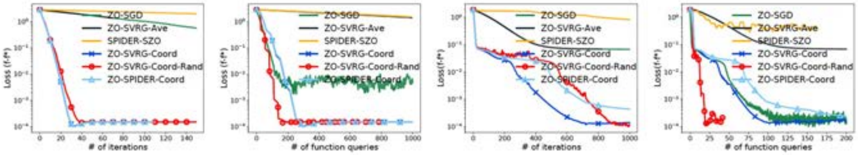
Improved ZO-SVRG and ZO-SPIDER [Ji et al., ICML'19]

- Numerical result comparisons:
  - Generation of black-box adversarial examples (DNN for MNIST handwritten digit classification, use the blackbox attacking loss in [Liu et al. NeurIPS'18])
  - Nonconvex logistic regression on LIBSVM [Chang and Lin, ACM TIST'11]



(a) Loss vs iterations: $n = 10$ (b) Loss vs queries: $n = 10$ (c) Loss vs iterations: $n = 100$ (d) Loss vs queries: $n = 100$

Figure 1. Comparison of different zeroth-order algorithms for generating black-box adversarial examples for digit "1" class



(a) Loss vs iterations: german (b) Loss vs queries: german (c) Loss vs iterations: ijcnn1 (d) Loss vs queries: ijcnn1

Figure 2. Comparison of different zeroth-order algorithms for logistic regression problem with a nonconvex regularizer

Next Class

Complex-Structured Learning