New Variance Reduction Algorithms for Nonconvex Finite-Sum Optimization

Quanquan Gu

Nov 8, 2018

Computer Science Department
UCLA

This is a joint work with Dongruo Zhou and Pan Xu

Outline

Outline

Problem Setup

- **General Finite-sum Optimization problem:**

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}), \ f_i \text{ can be nonconvex.}$$

- NP-hard to solve in general. (Hillar & Lim, 2013).

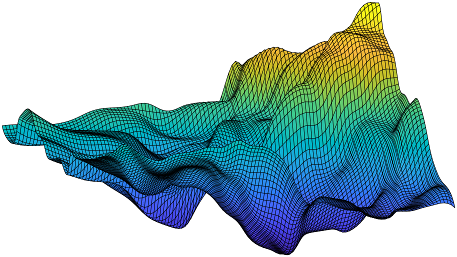


Figure: The landscape view of a nonconvex function.

Some Finite-sum function Examples

- Very common in machine learning.
- Non-convex regularized logistic regression (Reddi et al., 2016b).

$$f_i(\mathbf{x}) = y_i \log \phi(\mathbf{z}_i^\top \mathbf{x}) + (1 - y_i) \log[1 - \phi(\mathbf{z}_i^\top \mathbf{x}))] + \lambda \sum_{j=1}^{d} \frac{\mathbf{x}_{(j)}^2}{1 + \mathbf{x}_{(j)}^2},$$

  where $\phi(\mathbf{x})$ is the sigmoid function.
- Two layer neural network,

$$f_i(\mathbf{W}) = \left[y_i - \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\mathbf{w}_r^\top \mathbf{x}_i) \right]^2,$$

  where $\sigma(x) = \max\{x, 0\}$ is ReLU function.

Outline

First-order stationary points

- We aim at finding ϵ-first order stationary points $\widetilde{\mathbf{x}}$ of $F(\mathbf{x})$, where

$$\|\nabla F(\widetilde{\mathbf{x}})\|_2 \leq \epsilon. \tag{2.1}$$

- We use the number of stochastic gradient computations to measure the algorithm performance.

Gradient Descent & Stochastic Gradient Descent

Algorithm 1 GD

Input: $\mathbf{x}_0$, η, S
for $s = 1, \ldots, S$ **do**
    $\mathbf{x}_s \leftarrow \mathbf{x}_{s-1} - \eta \cdot \nabla F(\mathbf{x}_{s-1})$.
end for
Output: Uniformly choose $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_s\}$.

Algorithm 2 SGD

Input: $\mathbf{x}_0$, η, B, S.
for $s = 1, \ldots, S$ **do**
    Uniformly choose index set $\mathcal{I}_B \subset [n], |\mathcal{I}_B| = B$.
    $\mathbf{x}_s \leftarrow \mathbf{x}_{s-1} - \eta \cdot \nabla f_{\mathcal{I}_B}(\mathbf{x}_{s-1})$.
end for
Output: Uniformly choose $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_s\}$.

Gradient Descent & Stochastic Gradient Descent

- To converge to an ϵ-first order stationary point,
  - GD: $O(n\epsilon^{-2})$ stochastic gradient computations;
  - SGD: $O(\epsilon^{-4})$ stochastic gradient computations.
- GD: more computations per iteration, less iterations.
  SGD: less computations per iteration, more iterations.
- Can we combine them?

Idea of Variance Reduction

- For SGD,
  - $\mathbf{v} = \nabla f_{\mathcal{I}_B}(\mathbf{x})$,
  - $\mathbb{E}\mathbf{v} = \nabla F(\mathbf{x})$,
  - $\mathbb{E}\|\mathbf{v} - \nabla F(\mathbf{x})\|_2^2 \leq \sigma^2$,

  where the variance of $\mathbf{v}$ remains constant!
- Using reference point $\mathbf{x}_0$ and reference gradient $\mathbf{g}$ to reduce the variance of gradient estimator.
  - $\mathbf{v} = \nabla f_{\mathcal{I}_b}(\mathbf{x}) - \nabla f_{\mathcal{I}_b}(\mathbf{x}_0) + \mathbf{g} = \nabla f_{\mathcal{I}_b}(\mathbf{x}) - \nabla f_{\mathcal{I}_b}(\mathbf{x}_0) + \nabla F(\mathbf{x}_0)$,
  - $\mathbb{E}\mathbf{v} = \nabla F(\mathbf{x})$,
  - $\mathbb{E}\|\mathbf{v} - \nabla F(\mathbf{x})\|_2^2 \leq O(\|\mathbf{x} - \mathbf{x}_0\|_2^2)$

  where the variance of gradient estimator is decreasing!

Stochastic Variance-Reduced Gradient (SVRG)[1]

Algorithm 3 SVRG (Outer loop)

1: **Input:** $\widetilde{\mathbf{x}}_0$, η, B, b, S, T.
2: **for** $s = 1, \ldots, S$ **do**
3: $\quad \widetilde{\mathbf{x}}_s \leftarrow$ SVRG-Epoch$(\widetilde{\mathbf{x}}_{s-1}, \eta, B, b, T)$.
4: **end for**
5: **Output:** Uniformly choose $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_s\}$.

Algorithm 4 SVRG-Epoch

1: **Input:** $\mathbf{x}_0$, η, b, B, S, T.
2: $\mathbf{g} \leftarrow \nabla F(\mathbf{x}_0)$.
3: **for** $t = 1, \ldots, T$ **do**
4: $\quad$ Randomly pick $\mathcal{I}_b$ with size b.
5: $\quad \mathbf{v} \leftarrow \nabla f_{\mathcal{I}_b}(\mathbf{z}_{t-1}) - \nabla f_{\mathcal{I}_b}(\mathbf{z}_0) + \mathbf{g}$
6: $\quad \mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \cdot \mathbf{v}$.
7: **end for**
8: **Output:** Uniformly choose $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_t\}$.

[1] Johnson, Rie, and Tong Zhang. "Accelerating stochastic gradient descent using predictive variance reduction." Advances in neural information processing systems. 2013.

Stochastically Controlled Stochastic Gradient (SCSG)[1]

Algorithm 5 SCSG (Outer loop)

Input: $\widetilde{\mathbf{x}}_0$, η, B, b, S, T.
for $s = 1, \ldots, S$ **do**
  $\widetilde{\mathbf{x}}_s \leftarrow$ SCSG-Epoch$(\widetilde{\mathbf{z}}_{s-1}, \eta, B, b, T)$.
end for
Output: Uniformly choose $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_s\}$.

Algorithm 6 SCSG-Epoch

Input: $\mathbf{x}_0$, η, b, B, S, T.
Randomly pick $\mathcal{I}_B$ with size B.
$\mathbf{g} \leftarrow \nabla f_{\mathcal{I}_B}(\mathbf{x}_0)$.
for $t = 1, \ldots, T$ **do**
  Randomly pick $\mathcal{I}_b$ with size b.
  $\mathbf{v} \leftarrow \nabla f_{\mathcal{I}_b}(\mathbf{x}_{t-1}) - \nabla f_{\mathcal{I}_b}(\mathbf{x}_0) + \mathbf{g}$
  $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta \cdot \mathbf{v}$.
end for
Output: Uniformly choose $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_t\}$.

[1] Lei, Lihua, et al. "Non-convex finite-sum optimization via scsg methods." Advances in Neural Information Processing Systems. 2017.

Gradient Complexity Comparison

- To converge to an ϵ-first order stationary point,

Algorithm	Stochastic gradient computations
GD	$O(n\epsilon^{-2})$
SGD	$O(\epsilon^{-4})$
SVRG (Allen-Zhu et al., 2016) (Reddi et al., 2016)	$O(n^{2/3}\epsilon^{-2})$
SCSG (Lei et al., 2017)	$O(n^{2/3}\epsilon^{-2} \wedge \epsilon^{-10/3})$

- Strictly improves upon GD & SGD!

How to Improve it?

- For SVRG ,
  $\mathbf{v} = \nabla f_{\mathcal{I}_b}(\mathbf{x}) - \nabla f_{\mathcal{I}_b}(\mathbf{x}_0) + \nabla F(\mathbf{x}_0) = \mathbf{g}^{(1)} + \mathbf{g}^{(0)}$,
  $\mathbf{g}^{(1)} = \nabla f_{\mathcal{I}_b}(\mathbf{x}) - \nabla f_{\mathcal{I}_b}(\mathbf{x}_0), \mathbf{g}^{(0)} = \nabla F(\mathbf{x}_0)$.
- Only use two reference points $(\mathbf{x}, \mathbf{x}_0)$ and two reference gradients $(\mathbf{g}^{(1)}, \mathbf{g}^{(0)})$ ......
- Using more than two reference points and reference gradients!
- $\mathbf{v} = \mathbf{g}^{(K)} + \cdots + \mathbf{g}^{(1)} + \mathbf{g}^{(0)}$,
  $\mathbf{g}^{(l)} = \nabla f_{\mathcal{I}_l}(\mathbf{x}^{(l)}) - \nabla f_{\mathcal{I}_l}(\mathbf{x}^{(l-1)}), 1 \le l \le K$,
  $\mathbf{g}^{(0)} = \nabla F(\mathbf{x}^{(0)})$.
- $\mathbf{x}_t = \mathbf{x}_{t-1} - \eta \mathbf{v}$.

Stochastic Nested Variance Reduced Gradient Descent(SNVRG)[1]

Algorithm 7 SNVRG (Outer loop)

Input: $\mathbf{z}_0, \eta, B, S, K, \{B_l\}, \{T_l\}$.
for $s = 1, \ldots, S$ **do**
  $[\mathbf{y}_s, \mathbf{z}_s] \leftarrow$ SNVRG-Epoch $(\mathbf{z}_{s-1}, \eta, B, K, \{B_l\}, \{T_l\})$.
end for
Output: Uniformly choose $\mathbf{y}_{\text{out}}$ from $\{\mathbf{y}_s\}$.

Algorithm 8 SNVRG-Epoch

Input: $\mathbf{x}_0, \eta, B, K, \{B_l\}, \{T_l\}$.
Randomly pick $\mathcal{I}_B$ with size B.
$\mathbf{g}_0^{(0)} \leftarrow \nabla f_{\mathcal{I}_B}(\mathbf{x}_0)$, $\mathbf{x}_0^{(0)} \leftarrow \mathbf{x}_0$
$\mathbf{g}_0^{(l)} \leftarrow 0, \mathbf{x}_0^{(l)} \leftarrow \mathbf{x}_0, l \in [K]$
$\mathbf{v}_0 \leftarrow \sum_{l=0}^{K} \mathbf{g}_0^{(l)}, \mathbf{x}_1 \leftarrow \mathbf{x}_0 - \eta \cdot \mathbf{v}_0$
for $t = 1, \ldots, \prod_{l=1}^{K} T_l - 1$ **do**
  Update $\{\mathbf{x}_t^{(l)}\}$ and $\{\mathbf{g}_t^{(l)}\}$
  $\mathbf{v}_t \leftarrow \sum_{l=1}^{K} \mathbf{g}_t^{(l)} + \mathbf{g}_t^{(0)}$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \cdot \mathbf{v}_t$
end for
Output: $[\mathbf{x}_{\text{out}}, \mathbf{x}_{\prod_{l=1}^{K} T_l}]$, $\mathbf{x}_{\text{out}}$ from $\{\mathbf{x}_{0 \leq t < \prod_{l=1}^{K} T_l}\}$,

Update rules

- Update parameters: batch size parameters $\{B_l\}$, loop length parameters $\{T_l\}$.
- Let r be the smallest number where t can be divided by $\prod_{l=r+1}^{K} T_l$.
- Update rules for reference points $\{\mathbf{x}_t^{(l)}\}$:
  - $\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(r-1)}$ remain the same as $\mathbf{x}_{t-1}^{(1)}, \ldots, \mathbf{x}_{t-1}^{(r-1)}$
  - Set $\mathbf{x}_t^{(r)}, \ldots, \mathbf{x}_t^{(K)} \leftarrow \mathbf{x}_t$.
- Update rules for reference gradients $\{\mathbf{g}_t^{(l)}\}$:
  - We do not need to upgrade reference gradients unless they have changed!
  - $\mathbf{g}_t^{(1)}, \ldots, \mathbf{g}_t^{(r-1)}$ remain the same as $\mathbf{g}_{t-1}^{(1)}, \ldots, \mathbf{g}_{t-1}^{(r-1)}$
  - For $r \leq l \leq K$,
    randomly pick up $\mathcal{I}$ with size B_l, set $\mathbf{g}_t^{(l)} \leftarrow \nabla f_{\mathcal{I}}(\mathbf{x}_t^{(l)}) - \nabla f_{\mathcal{I}}(\mathbf{x}_t^{(l-1)})$.

Step 0

$K = 2$, $T_1 = 2$, $T_2 = 3$ as an example.

Reference points:

$$\mathbf{x}_0^{(0)} \leftarrow \mathbf{x}_0, \mathbf{x}_0^{(1)} \leftarrow \mathbf{x}_0, \mathbf{x}_0^{(2)} \leftarrow \mathbf{x}_0,$$

Reference gradients:

$$\mathbf{g}_0^{(0)} \leftarrow \nabla f_{\mathcal{I}_0}(\mathbf{x}_0^{(0)}),$$
$$\mathbf{g}_0^{(1)} \leftarrow \nabla f_{\mathcal{I}_1}(\mathbf{x}_0^{(1)}) - \nabla f_{\mathcal{I}_1}(\mathbf{x}_0^{(0)}),$$
$$\mathbf{g}_0^{(2)} \leftarrow \nabla f_{\mathcal{I}_2}(\mathbf{x}_0^{(2)}) - \nabla f_{\mathcal{I}_2}(\mathbf{x}_0^{(1)}),$$

Updating rule:

$$\mathbf{x}_1 \leftarrow \mathbf{x}_0 - \eta(\mathbf{g}_0^{(0)} + \mathbf{g}_0^{(1)} + \mathbf{g}_0^{(2)}).$$

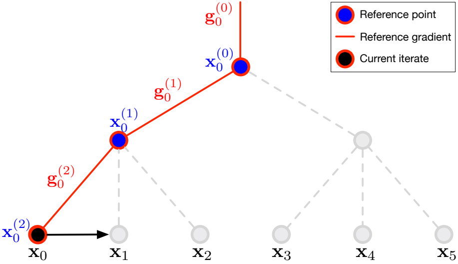


Figure: Iterate $t = 0$.

Step 1

Reference points:

$$\mathbf{x}_1^{(0)} \leftarrow \mathbf{x}_0^{(0)}, \mathbf{x}_1^{(1)} \leftarrow \mathbf{x}_0^{(1)}, \mathbf{x}_1^{(2)} \leftarrow \mathbf{x}_1,$$

Reference gradients:

$$\mathbf{g}_1^{(0)} \leftarrow \mathbf{g}_0^{(0)},$$
$$\mathbf{g}_1^{(1)} \leftarrow \mathbf{g}_0^{(1)},$$
$$\mathbf{g}_1^{(2)} \leftarrow \nabla f_{\mathcal{I}_2}(\mathbf{x}_1^{(2)}) - \nabla f_{\mathcal{I}_2}(\mathbf{x}_1^{(1)}),$$

Updating rule:

$$\mathbf{x}_2 \leftarrow \mathbf{x}_1 - \eta(\mathbf{g}_1^{(0)} + \mathbf{g}_1^{(1)} + \mathbf{g}_1^{(2)}).$$

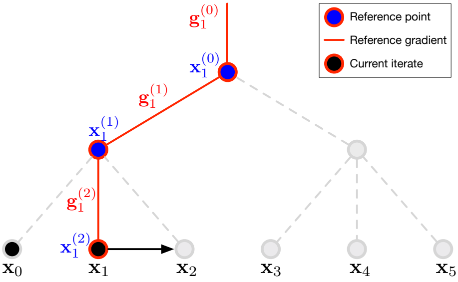


Figure: Iterate $t = 1$.

Step 2

Reference points:

$$\mathbf{x}_2^{(0)} \leftarrow \mathbf{x}_1^{(0)}, \mathbf{x}_2^{(1)} \leftarrow \mathbf{x}_1^{(1)}, \mathbf{x}_2^{(2)} \leftarrow \mathbf{x}_2,$$

Reference gradients:

$$\mathbf{g}_2^{(0)} \leftarrow \mathbf{g}_1^{(0)},$$
$$\mathbf{g}_2^{(1)} \leftarrow \mathbf{g}_1^{(1)},$$
$$\mathbf{g}_2^{(2)} \leftarrow \nabla f_{\mathcal{I}_2}(\mathbf{x}_2^{(2)}) - \nabla f_{\mathcal{I}_2}(\mathbf{x}_2^{(1)}),$$

Updating rule:

$$\mathbf{x}_3 \leftarrow \mathbf{x}_2 - \eta(\mathbf{g}_2^{(0)} + \mathbf{g}_2^{(1)} + \mathbf{g}_2^{(2)}).$$

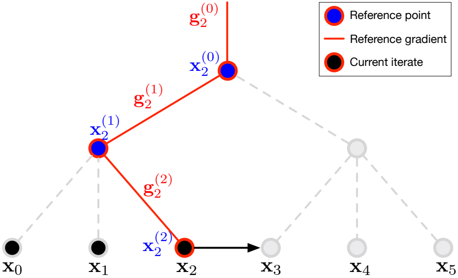


Figure: Iterate $t = 2$.

Step 3

Reference points:

$$\mathbf{x}_3^{(0)} \leftarrow \mathbf{x}_2^{(0)}, \mathbf{x}_3^{(1)} \leftarrow \mathbf{x}_3, \mathbf{x}_3^{(2)} \leftarrow \mathbf{x}_3,$$

Reference gradients:

$$\mathbf{g}_3^{(0)} \leftarrow \mathbf{g}_2^{(0)},$$
$$\mathbf{g}_3^{(1)} \leftarrow \nabla f_{\mathcal{I}_1}(\mathbf{x}_3^{(1)}) - \nabla f_{\mathcal{I}_1}(\mathbf{x}_3^{(0)}),$$
$$\mathbf{g}_3^{(2)} \leftarrow \nabla f_{\mathcal{I}_2}(\mathbf{x}_3^{(2)}) - \nabla f_{\mathcal{I}_2}(\mathbf{x}_3^{(1)}),$$

Updating rule:

$$\mathbf{x}_4 \leftarrow \mathbf{x}_3 - \eta(\mathbf{g}_3^{(0)} + \mathbf{g}_3^{(1)} + \mathbf{g}_3^{(2)}).$$

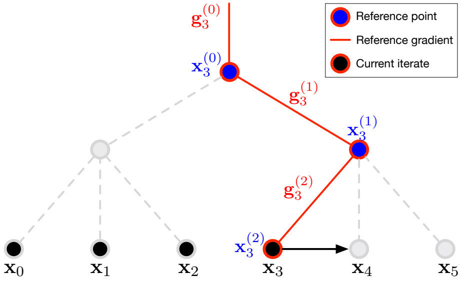


Figure: Iterate $t = 3$.

Step 4

Reference points:

$$\mathbf{x}_4^{(0)} \leftarrow \mathbf{x}_2^{(0)}, \mathbf{x}_3^{(1)} \leftarrow \mathbf{x}_3, \mathbf{x}_3^{(2)} \leftarrow \mathbf{x}_3,$$

Reference gradients:

$$\mathbf{g}_3^{(0)} \leftarrow \mathbf{g}_2^{(0)},$$
$$\mathbf{g}_3^{(1)} \leftarrow \nabla f_{\mathcal{I}_1}(\mathbf{x}_3^{(1)}) - \nabla f_{\mathcal{I}_1}(\mathbf{x}_3^{(0)}),$$
$$\mathbf{g}_3^{(2)} \leftarrow \nabla f_{\mathcal{I}_2}(\mathbf{x}_3^{(2)}) - \nabla f_{\mathcal{I}_2}(\mathbf{x}_3^{(1)}),$$

Updating rule:

$$\mathbf{x}_4 \leftarrow \mathbf{x}_3 - \eta(\mathbf{g}_3^{(0)} + \mathbf{g}_3^{(1)} + \mathbf{g}_3^{(2)}).$$

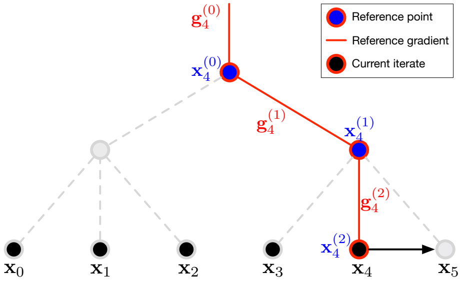


Figure: Iterate $t = 4$.

Step 5

Reference points:

$$\mathbf{x}_5^{(0)} \leftarrow \mathbf{x}_4^{(0)}, \mathbf{x}_5^{(1)} \leftarrow \mathbf{x}_4^{(1)}, \mathbf{x}_5^{(2)} \leftarrow \mathbf{x}_5,$$

Reference gradients:

$$\mathbf{g}_5^{(0)} \leftarrow \mathbf{g}_4^{(0)},$$
$$\mathbf{g}_5^{(1)} \leftarrow \mathbf{g}_4^{(1)},$$
$$\mathbf{g}_5^{(2)} \leftarrow \nabla f_{\mathcal{I}_2}(\mathbf{x}_5^{(2)}) - \nabla f_{\mathcal{I}_2}(\mathbf{x}_5^{(1)}),$$

Updating rule:

$$\mathbf{x}_6 \leftarrow \mathbf{x}_5 - \eta(\mathbf{g}_5^{(0)} + \mathbf{g}_5^{(1)} + \mathbf{g}_5^{(2)}).$$

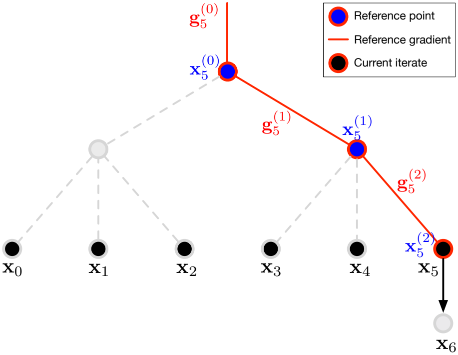


Figure: Iterate $t = 5$.

Theoretical Results

- **Assumptions:**
  - (Optimal Gap) $F(\mathbf{x}_0) - \inf_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) \le \Delta_F$
  - (Smoothness) For each i, f_i is L-smooth, where $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \le L\|\mathbf{x} - \mathbf{y}\|_2$.
  - (Variance Bounded) For any $\mathbf{x}$, $\mathbb{E}_i\|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|_2^2 \le \sigma^2$

Theorem (Complexity analysis for SNVRG)

With specific parameter choices, SNVRG will find an ϵ-first order stationary point within

$$\widetilde{O}\left(\left[\frac{\sigma^2}{\epsilon^2} \wedge n + \frac{L\Delta_F}{\epsilon^2}\left[\frac{\sigma^2}{\epsilon^2} \wedge n\right]^{1/2}\right]\right)$$

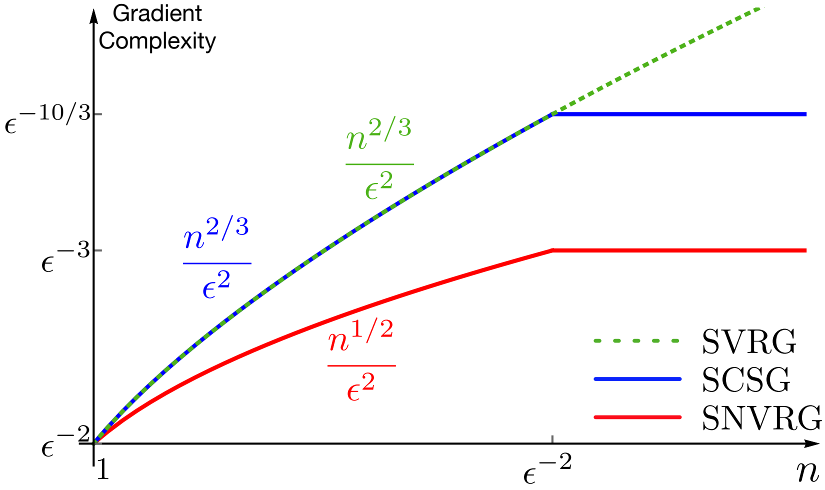
stochastic gradient computations.

Gradient Complexity Comparison

Algorithm	Stochastic gradient computations
GD	$O(n\epsilon^{-2})$
SGD	$O(\epsilon^{-4})$
SVRG (Allen-Zhu et al., 2016) (Reddi et al., 2016)	$O(n^{2/3}\epsilon^{-2})$
SCSG (Lei et al., 2017)	$O(n^{2/3}\epsilon^{-2} \wedge \epsilon^{-10/3})$
SNVRG (this paper)	$\widetilde{O}(n^{1/2}\epsilon^{-2} \wedge \epsilon^{-3})$

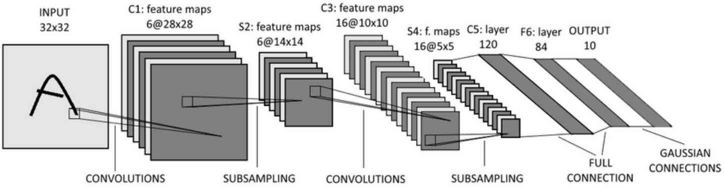
- SNVRG strictly better than SCSG by a factor $\Omega(n^{1/6} \wedge \epsilon^{-1/3})$.

Gradient Complexity Comparison

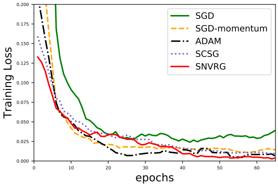


Experimental Results

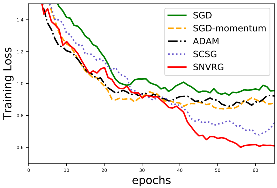
- **Baseline Algorithms**
  - SGD (**SGD**)
  - SGD with momentum (**SGD-momentum**)
  - ADAM (**ADAM**) (Kingma et al., 2014)
  - SCSG (**SCSG**) (Lei et al., 2017)
  - SNVRG with $K = 2$ (**SNVRG**)
- **Benchmark Optimization Problems**
  LeNet-5 on
  - MNIST (LeCun et al., 1998a)
  - CIFAR10 (Krizhevsky, 2009)
  - SVHN (Netzer et al., 2011)



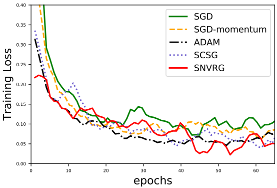
Experimental Results



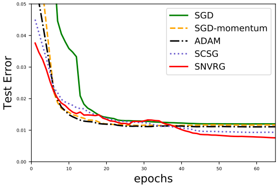
(a) MNIST



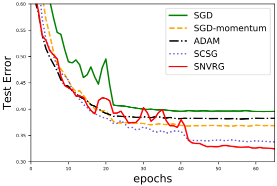
(b) CIFAR10



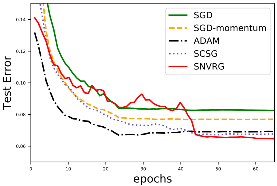
(c) SVHN



(d) MNIST



(e) CIFAR10

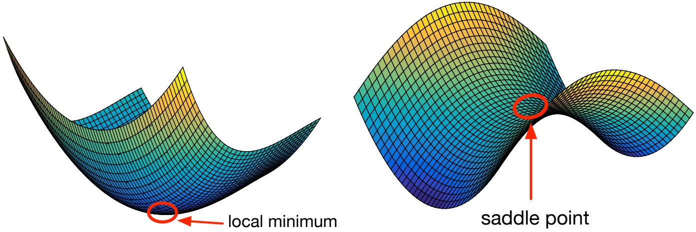


(f) SVHN

Outline

Second-order stationary points

- We can find first-order stationary point via SNVRG, however......
- Is first-order stationary point enough?



(g) Local minimum (h) Saddle point

- **Goal:** To find an (ϵ_g, ϵ_H)-second-order staionary point $\mathbf{x}$ such that $\|\nabla F(\mathbf{x})\| \leq \epsilon_g$, $\lambda_{\min}(\nabla^2 F(\mathbf{x})) \geq -\epsilon_H$

Cubic Regularization of Newton Method[1]

- **Cubic Regularization**
  Starting from $\mathbf{x}_0$, iteratively execute

$$m_t(\mathbf{h}) = \langle \nabla F(\mathbf{x}_t), \mathbf{h} \rangle + \frac{1}{2} \langle \nabla^2 F(\mathbf{x}_t)\mathbf{h}, \mathbf{h} \rangle + \frac{M}{6} \|\mathbf{h}\|_2^3$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \operatorname*{argmin}_{\mathbf{h} \in \mathbb{R}^d} m_t(\mathbf{h})$$

- Minimize cubic regularized subproblem $m_t(\mathbf{h})$ in each iteration.

Theorem (Informal)

Under certain conditions, $\mathbf{x}_t$ converges to an $(\epsilon, \sqrt{\epsilon})$-second-order staionary point within $O(1/\epsilon^{3/2})$ number of iterations.

- Expensive to compute $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$ when n is large .

[1] Nesterov, Yurii, and Boris T. Polyak. "Cubic regularization of Newton method and its global performance." Mathematical Programming 108.1 (2006): 177-205.

Stochastic Variance-Reduced Cubic Regularization (SVRC)[1]

- **The Proposed SVRC**
  - Operates with epochs.
  - At t-th iteration of $(s+1)$-th epoch, we have

$$m_t^{s+1}(\mathbf{h}) = \langle \mathbf{v}_t^{s+1}, \mathbf{h} \rangle + \frac{1}{2} \langle \mathbf{U}_t^{s+1} \mathbf{h}, \mathbf{h} \rangle + \frac{M_{s+1,t}}{6} \|\mathbf{h}\|_2^3$$
$$\mathbf{x}_{t+1}^{s+1} = \mathbf{x}_t^{s+1} + \operatorname*{argmin}_{\mathbf{h} \in \mathbb{R}^d} m_t^{s+1}(\mathbf{h})$$

- Penalty parameters M_t^{s+1}
- semi-stochastic gradient $\mathbf{v}_t^{s+1} \approx \nabla F(\mathbf{x}_t^{s+1})$
- semi-stochastic Hessian $\mathbf{U}_t^{s+1} \approx \nabla^2 F(\mathbf{x}_t^{s+1})$.

[1] Dongruo Zhou, Pan Xu, Quanquan Gu ; Proceedings of the 35th International Conference on Machine Learning, PMLR 80:5990-5999, 2018

Semi-stochastic gradient and Hessian

- Reference point $\widehat{\mathbf{x}}^s = \mathbf{x}_0^{s+1}$, reference gradient $\mathbf{g}^s = \nabla f(\widehat{\mathbf{x}}^s)$ and reference Hessian $\mathbf{H}^s = \nabla^2 f(\widehat{\mathbf{x}}^s)$.

$$
\begin{aligned}
\mathbf{v}_t^{s+1} &= \frac{1}{b_g}\sum_{i_t\in I_g}\nabla f_{i_t}(\mathbf{x}_t^{s+1}) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s \\
&\quad - \bigg(\frac{1}{b_g}\sum_{i_t\in I_g}\nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\bigg)(\mathbf{x}_t^{s+1} - \widehat{\mathbf{x}}^s), \\
\mathbf{U}_t^{s+1} &= \frac{1}{b_h}\bigg(\sum_{j_t\in I_h}\nabla^2 f_{j_t}(\mathbf{x}_t^{s+1}) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\bigg) + \mathbf{H}^s,
\end{aligned}
$$

- I_g and I_h are index sets, $|I_g| = b_g$, $|I_h| = b_h$, $b_g, b_h \ll n$.
- Unlike subsampled gradient and Hessian which have unchanged variances, the variances of $\mathbf{v}_t^{s+1}$ and $\mathbf{U}_t^{s+1}$ are reduced.

Oracle Definition

- **Second Order Oracle (SO)**
  Given an index i and a point $\mathbf{x}$, one second-order oracle (SO) call returns such a triple:

$$[f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x})]$$

- **Cubic Subproblem Oracle(CSO)**
  Given a vector $\mathbf{g}$, a Hessian matrix $\mathbf{H}$ and a positive constant θ, one Cubic Subproblem Oracle (CSO) call returns $\mathbf{h}_{\text{sol}}$, where $\mathbf{h}_{\text{sol}}$ can be solved exactly as follows

$$\mathbf{h}_{\text{sol}} = \operatorname*{argmin}_{\mathbf{h} \in \mathbb{R}^d} \langle \mathbf{g}, \mathbf{h} \rangle + \frac{1}{2}\langle \mathbf{h}, \mathbf{H}\mathbf{h} \rangle + \frac{\theta}{6}\|\mathbf{h}\|_2^3.$$

Convergence Analysis

- **Assumptions:**
  - (Optimal Gap) $F(\mathbf{x}_0) - \inf_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \leq \Delta_F$
  - (Hessian Lipschitz) For each i, f_i is ρ-Hessian Lipschitz, where $\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\|_2 \leq \rho\|\mathbf{x}-\mathbf{y}\|_2$.

Theorem (Complexity analysis for SVRC)

With specific parameter choices, SVRC will find an $(\epsilon, \sqrt{\rho\epsilon})$-second-order staionary point within SO complexity

$$O\left(n + \frac{\Delta_F\sqrt{\rho}n^{4/5}}{\epsilon^{3/2}}\right),$$

and CSO complexity

$$O\left(\frac{\Delta_F\sqrt{\rho}}{\epsilon^{3/2}}\right).$$

Complexity Comparision

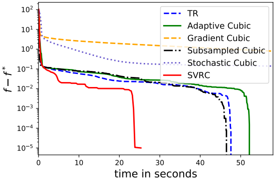
Algorithm	SO calls	CSO calls	Smoothness
Cubic regularization (Nesterov & Polyak, 2006)	$O(n\epsilon^{-3/2})$	$O(\epsilon^{-3/2})$	No
Subsampled cubic (Kohler & Lucchi, 2017)	$\widetilde{O}(\epsilon^{-7/2} + \epsilon^{-5/2})$	$O(\epsilon^{-3/2})$	Yes
Subsampled cubic (Xu et al., 2017)	$\widetilde{O}(n\epsilon^{-3/2} + \epsilon^{-5/2})$	$O(\epsilon^{-3/2})$	Yes
SVRC (this paper)	$\widetilde{O}(n^{4/5}\epsilon^{-3/2})$	$O(\epsilon^{-3/2})$	No

- SVRC does not need gradient Lipschitz assumption (smoothness).
- SVRC is strictly better than cubic regularization by a factor $\Omega(n^{1/5})$, and better than subsampled cubic when $\epsilon \ll n^{-2/5}$.

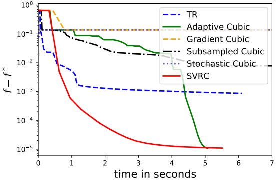
Experiment

- **Baseline Algorithms**
  - Adaptive cubic regularization (**Adaptive Cubic**) (Cartis et al., 2011)
  - subsampled cubic regularization (**Subsampled Cubic**) (Kohler & Lucchi, 2017)
  - Stochastic cubic regularization (**Stochastic Cubic**) (Tripuraneni et al., 2017)
  - Gradient cubic regularization (**Gradient Cubic**) (Carmon & Duchi, 2016)
  - Trust region Newton method (**TR**) (Conn et al., 2000)
- **Benchmark Optimization Problems**
  - Nonconvex Regularized logistic regression (Reddi et al., 2016b)
  - Nonlinear least square (Xu et al., 2017a)
  - Robust linear regression (Barron, 2017)

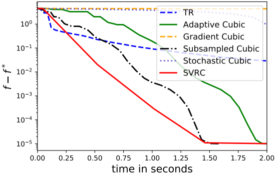
Experimental Results



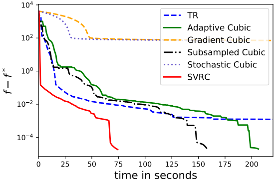
(i) a9a



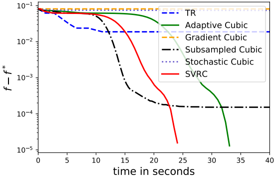
(j) a9a



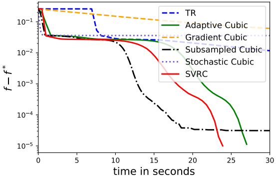
(k) a9a



(l) covtype



(m) covtype



(n) covtype

Outline

Summary

- To find first-order stationary points, SNVRG is near optimal.
- To find second-order stationary points, optimality is still an open problem!

Questions

Any Questions?