SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path Integrated Differential Estimator

Chris Junchi Li

Tencent & Princeton University

Joint work with Cong Fang (Peking U), Zhouchen Lin (Peking U), and Tong Zhang (Tencent AI Lab)

November 2018

Outline

Introduction

We study the nonconvex optimization problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \qquad f(x) \equiv \mathbb{E}\left[F(x;\zeta)\right] \tag{*}$$

- Non-convex optimization problem of form (*) contains many large-scale learning tasks (Bottou, 2010; Bubeck et al., 2015; Bottou et al., 2016)
- E.g. PCA, ICA, estimation of graphical models, training deep neural networks (Goodfellow et al., 2016)

When ζ takes only finite values, (*) can be further reduced to

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \qquad f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{**}$$

Study both finite-sum case (n is finite) and online case (n is ∞)

Motivation: Speed Up Machine Learning Algorithms

To find an ε-first-order stationary point (ε-FSP) that satisfies $\|\nabla f(x)\| \leq \varepsilon$ for nonconvex function $f(x)$, the gradient costs are

- GD / NGD / SGD: $n\varepsilon^{-2} \wedge \varepsilon^{-4}$ (Nesterov, 2004)
- SVRG / SCSG: $(n + n^{2/3}\varepsilon^{-2}) \wedge \varepsilon^{-3.333}$ (Allen-Zhu & Hazan, 2016, Reddi et al., 2016, Lei et al. 2017)

where $a \wedge b := \min(a, b)$.

Question: can the gradient cost be further reduced?

Outline

Stochastic Path-Integrated Differential Estimator: Core Idea

Observe a sequence $\widehat{x}_{0:K} = \{\widehat{x}_0, \ldots, \widehat{x}_K\}$ (often moves continuously), goal is to dynamically track, for an arbitrary deterministic vector quantity $Q(x)$, $Q(\widehat{x}^k)$ for $k = 0, 1, \ldots, K$

- Initial estimate $\widetilde{Q}(\widehat{x}^0) \approx Q(\widehat{x}^0)$
- Unbiased estimate $\xi_k(\widehat{x}_{0:k})$ of $Q(\widehat{x}^k) - Q(\widehat{x}^{k-1})$ such that for each $k = 1, \ldots, K$

$$\mathbb{E}\left[\xi_k(\widehat{x}_{0:k}) \mid \widehat{x}_{0:k}\right] = Q(\widehat{x}^k) - Q(\widehat{x}^{k-1})$$

- Integrate (in the discrete sense) the stochastic differential estimate as

$$\widetilde{Q}(\widehat{x}_{0:K}) := \widetilde{Q}(\widehat{x}^0) + \sum_{k=1}^{K} \xi_k(\widehat{x}_{0:k}) \tag{2.1}$$

- Call estimator $\widetilde{Q}(\widehat{x}_{0:K})$ the Stochastic Path-Integrated Differential EstimatoR, or SPIDER for brevity
- In this talk our $Q(x)$ is picked as $\nabla f(x)$ (or $f(x)$)

To bound the second moment of the error of our estimator $\|\widetilde{Q}(\widehat{x}_{0:K}) - Q(\widehat{x}^K)\|$

Proposition 1

$$\begin{aligned} &\mathbb{E}\|\widetilde{Q}(\widehat{x}_{0:K}) - Q(\widehat{x}^K)\|^2 \\ &= \mathbb{E}\|\widetilde{Q}(\widehat{x}^0) - Q(\widehat{x}^0)\|^2 + \sum_{k=1}^{K} \mathbb{E}\|\xi_k(\widehat{x}_{0:k}) - (Q(\widehat{x}^k) - Q(\widehat{x}^{k-1}))\|^2 \end{aligned} \tag{2.2}$$

Proposition 1 can be easily concluded using the property of square-integrable martingales

To obtain the high-probability bound of our estimator $\|\widetilde{Q}(\widehat{x}_{0:K}) - Q(\widehat{x}^K)\|$

Proposition 2

Suppose $\mathbb{E}\exp\left(\|\widetilde{Q}(\widehat{x}^0) - Q(\widehat{x}^0)\|^2/b_0^2\right) \le 2$ and, for all $k = 0, 1, \ldots, K$,

$$\mathbb{E}\exp\left(\|\xi_k(\widehat{x}_{0:k}) - (Q(\widehat{x}^k) - Q(\widehat{x}^{k-1}))\|^2/b_k^2\right) \le 2 \tag{2.3}$$

Then if the Lipschitz condition holds $\|\widetilde{Q}(x) - \widetilde{Q}(y)\| \le L_Q\|x - y\|$, for any $\gamma > 0$ we have with probability at least $1 - 2\gamma K$ the following

$$\left\|\widetilde{Q}(\widehat{x}_{0:k}) - Q(\widehat{x}^k)\right\| \le 2L_Q\sqrt{\sum_{j=0}^{k} b_j^2 \cdot \log\frac{1}{\gamma}}, \quad \text{for all } k = 0, 1, \ldots, K \tag{2.4}$$

Azuma-Hoeffding-type concentration inequality for vector-valued martingales.
Proof needs a dimension-reduction lemma (Kallenberg & Sztencel, 1991)

Also see (Juditsky & Nemirovski, Technical report 2008 hal-00318071)

Outline

Finite-Sum Case, Expectation

Algorithm 1 SPIDER-SFO: Input x^0, $q = n$ (finite-sum case, in expectation)

for $k = 0$ to $K - 1$ **do**
  if $\mod(k, q) = 0$ **then**
    Compute the full gradient $v^k = \nabla f(x^k)$
  else
    Draw $i \sim [n]$ uniformly at random, and let $v^k = \nabla f_i(x^k) - \nabla f_i(x^{k-1}) + v^{k-1}$
  end if
  $x^{k+1} = x^k - \eta^k v^k$ where $\eta^k = \min\left(\frac{\epsilon}{\|v^k\|}, \frac{1}{2}\right) \cdot \frac{1}{Ln^{1/2}}$.
end for
Return $\tilde{x}$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

Theorem (First-order Stationary Point, finite-sum setting)

Algorithm 1 outputs in $K_0 = (4L\Delta n^{1/2})\epsilon^{-2} + 1$ *iterations* $\mathbb{E}\left[\|\nabla f(\widetilde{x})\|\right] \leq 5\epsilon$. *The gradient cost upper bound is*

$$n + (12L\Delta n^{1/2})\epsilon^{-2} = \widetilde{\mathcal{O}}(n + n^{1/2}\epsilon^{-2})$$

Essentially the same convergence rate results were also achieved independently by the so-called SNVRG algorithm (Zhou, Xu, Gu, NIPS 2018)

- The recursive update equation of v^k

$$v^k = \nabla f_i(x^k) - \nabla f_i(x^{k-1}) + v^{k-1}$$

is a SPIDER for $\nabla f(x^k)$ [Seeing $\xi_k(x_{0:k}) = \nabla f_i(x^k) - \nabla f_i(x^{k-1})$ as in $\widetilde{Q}(\widehat{x}_{0:K}) := \widetilde{Q}(\widehat{x}^0) + \sum_{k=1}^K \xi_k(\widehat{x}_{0:k})$ of SPIDER]

- Compared with SVRG (Johnson & Zhang 2013) update rule where $v^0 = \nabla f(x^0) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^0)$ and

$$v^k = \nabla f_i(x^k) - \nabla f_i(x^0) + v^0$$

Satisfying unbiasedness $\mathbb{E}[v^k \mid \mathcal{F}_{k-1}] = \nabla f(x^k)$

- SPIDER-SFO drops unbiasedness in exchange for a sharper estimate!

- No storage requirement compared to SAG (Schmidt et al. 2017) / SAGA (Defazio et al. 2014) and many other variance reduction methods

Assumption

(i) $\Delta := f(x^0) - f(x^*) < \infty$ where x^* is a global minimizer of $f(x)$

(ii) The component function $f_i(x)$ has an averaged L-Lipschitz gradient

$$\mathbb{E}\|\nabla f_i(x) - \nabla f_i(y)\|^2 \leq L^2\|x - y\|^2$$

Under the above averaged L-Lipschitz gradient assumption, SPIDER-SFO matches the lower bound (Carmon, Duchi, Hinder, Sidford, Submitted to Mathematical Programming 2017+), modified to the finite-sum setting:

Theorem (Lower bound for SFO for the finite-sum setting)

For any $L > 0$, $\Delta > 0$, and $2 \leq n \leq O\left(\Delta^2 L^2 \cdot \epsilon^{-4}\right)$, for any algorithm $\mathcal{F}$ satisfying

$$x^k = \mathbf{F}^{k-1}\left(\xi, \nabla f_{\mathcal{F}_{i_0}}(x^0), \nabla f_{\mathcal{F}_{i_1}}(x^1), \ldots, \nabla f_{\mathcal{F}_{i_{k-1}}}(x^{k-1})\right)$$

there exists a dimension $d = \widetilde{\mathcal{O}}\left(\Delta^2 L^2 \cdot n\epsilon^{-4}\right)$, and a function f satisfies the above Assumption, such that in order to find a point $\widetilde{x}$ for which $\|\nabla f(\widetilde{x})\| \leq \epsilon$, $\mathcal{F}$ must cost at least $\Omega\left(L\Delta \cdot n^{1/2}\epsilon^{-2}\right)$ stochastic gradient accesses.

Proof of Theorem in Finite-Sum Case

Theorem (First-order Stationary Point, finite-sum setting)

Algorithm 1 outputs in $K = (4L\Delta n^{1/2})\epsilon^{-2}$ *iterations* $\mathbb{E}\left[\|\nabla f(\widetilde{x})\|\right] \leq 5\epsilon$ *with probability* $\geq 1-\delta$. *The gradient cost upper bound is*

$$n + (12L\Delta n^{1/2})\epsilon^{-2} = \widetilde{\mathcal{O}}(n + n^{1/2}\epsilon^{-2})$$

Applying Proposition 1 gives us the following lemma, which indicates that SPIDER maintains an error of $\|v^k - \nabla f(x^k)\| = \mathcal{O}(\epsilon)$ in second moment:

Lemma

Let $k_0 = \lfloor k/q \rfloor \cdot q$. *Then under the Assumption, Algorithm 1 satisfies*

$$\mathbb{E}\left[\|v^k - \nabla f(x^k)\|^2 \mid x_{0:k_0}\right] \leq \epsilon^2$$

Proof of Theorem in Finite-Sum Case

Standard descent analysis for our algorithm gives a second lemma

Lemma

$$\mathbb{E}\left[f(x^{k+1}) - f(x^k)\right] \leq -\frac{\epsilon}{4Ln^{1/2}}\mathbb{E}\left\|v^k\right\| + \frac{3\epsilon^2}{4n^{1/2}L}$$

Telescoping from $k = 0$ to $K - 1$ in the last lemma

$$\frac{\epsilon}{4Ln^{1/2}}\sum_{k=0}^{K-1}\mathbb{E}\|v^k\| \leq f(x^0) - \mathbb{E}f(x^K) + \frac{3K\epsilon^2}{4Ln^{1/2}} \leq \Delta + \frac{3K\epsilon^2}{4Ln^{1/2}} \tag{3.1}$$

Multiplying ϵ/Δ in the last display, and using $K = \frac{4L\Delta n^{1/2}}{\epsilon^2}$, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|v^k\| \leq 4\epsilon \tag{3.2}$$

Since $\widetilde{x}$ is chosen uniformly at random from $\{0, \ldots, K-1\}$, we have

$$\mathbb{E}\|\nabla f(\widetilde{x})\| = \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(x^k)\| \overset{Lemma1}{\leq} \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|v^k\| + \epsilon \overset{(3.2)}{\leq} 5\epsilon \tag{3.3}$$

Proof of Theorem in Finite-Sum Case

The gradient cost analysis is computed as:

$$\begin{aligned}\left\lceil \frac{\Delta}{\epsilon^2/(4Ln^{1/2})} \cdot \frac{1}{n} \right\rceil n + 2 \cdot \frac{\Delta}{\epsilon^2/(4Ln^{1/2})} &\leq 3 \cdot \frac{\Delta}{\epsilon^2/(4Ln^{1/2})} + n \\ &= \frac{12(L\Delta) \cdot n^{1/2}}{\epsilon^2} + n \qquad (3.4)\end{aligned}$$

This concludes a gradient cost of $n + 12(L\Delta) \cdot n^{1/2}\epsilon^{-2}$.

Outline

Online Case, High-Probability

Algorithm 2 SPIDER-SFO: Input x^0, $\mathcal{S}_1 = 2\sigma^2/\epsilon^2$, $q = 2\sigma^2/\epsilon^2$ (online case, in high probability)

for $k = 0$ to $K-1$ **do**
  if $\mod(k, q) = 0$ **then**
    Draw $\mathcal{S}_1$ samples and let $v^k = \nabla f_{\mathcal{S}_1}(x^k)$
  else
    Draw $i \sim [n]$ uniformly at random, and let $v^k = \nabla f_i(x^k) - \nabla f_i(x^{k-1}) + v^{k-1}$
  end if
  if $\|v^k\| \leq 2\epsilon$ **then**
    return x^k
  else
    $x^{k+1} = x^k - \eta \cdot (v^k/\|v^k\|)$ where $\eta = \epsilon^2/(2L\sigma)$.
  end if
end for
return x^K ◇ however, this line is *not* reached with high probability

Theorem (First-order Stationary Point, online setting)

Algorithm 2 outputs in $\mathcal{K} \leq K_0 = \widetilde{\mathcal{O}}\left((L\Delta\sigma)\epsilon^{-3} + 1\right)$ *iterations* $\|\nabla f(x^{\mathcal{K}})\| \leq \epsilon$. *The gradient cost upper bound is*

$$\left(4\sigma^2\epsilon^{-2} + 64\sqrt{2}L\Delta\sigma \cdot \epsilon^{-3}\right) \cdot \log(\mathit{Factor}) = \widetilde{\mathcal{O}}(\epsilon^{-3})$$

Assumption

(i) $\Delta := f(x^0) - f(x^*) < \infty$

(ii') Each component function $f_i(x)$ has L-Lipschitz continuous gradient

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$$

(iii') (For online case only) the stochastic gradients are subgaussian vectors with variance proxy $\sigma^2 < \infty$, i.e.

$$\mathbb{E} \exp\left(\|\nabla f_i(x) - \nabla f(x)\|^2 / \sigma^2\right) \leq 2$$

- Idea is similar to the finite-sum setting, except we treat $n \approx 2\sigma^2/\varepsilon^2$ so

$$\mathbb{E}\left[\|v^k - \nabla f(x^k)\|^2 \mid x_{0:k_0}\right] \leq \epsilon^2$$

still holds

- Focus on high-probability result (seems to be the first time in non-convex literature)

Vector Martingale Concentration Inequality

Proposition

Let $\varepsilon_1, \ldots, \varepsilon_K \in \mathbb{R}^d$ be a vector-valued martingale difference sequence with respect to $\mathcal{F}_k$, i.e., for each $k = 1, \ldots, K$, $\mathbb{E}[\varepsilon_k \mid \mathcal{F}_{k-1}] = 0$ and $\mathbb{E}\exp(\|\varepsilon_k\|^2 / B_k^2 \mid \mathcal{F}_{k-1}) \leq 2$. We have for an arbitrary real positive λ

$$\mathbb{P}\left(\left\|\sum_{k=1}^{K} \varepsilon_k\right\| \geq \lambda\right) \leq 4\exp\left(-\frac{\lambda^2}{4\sum_{k=1}^{K} B_k^2}\right) \tag{4.1}$$

Sufficient to prove it for $d = 2$, in which case it is Azuma-Hoeffding!
(Kallenberg & Sztencel 1991, Lee & Peres 2015)

Lemma (Dimension reduction lemma for $\mathbb{R}^d$ or Hilbert space)

Let $(N_k, k = 0, 1, \ldots, K)$ be a discrete-time, $\mathbb{R}^d$-valued martingale. Then there exists a discrete-time, $\mathbb{R}^2$-valued martingale $(M_k, k = 0, 1, \ldots, K)$ so that

$$\|M_k\| = \|N_k\| \quad \text{for each } k = 0, 1, \ldots K \ ,$$

and

$$\|M_{k+1} - M_k\| = \|N_{k+1} - N_k\| \quad \text{for each } k = 0, 1, \ldots K-1 \ .$$

Set $\mathcal{K}$ as the time when Algorithm 2 terminates (random stopping time), i.e.

$$\mathcal{K} = \inf\{k \geq 0 : \|v^k\| < 2\varepsilon\}$$

Let $n_0 = \widetilde{\mathcal{O}}(\sigma/\varepsilon)$

Lemma

Under the Assumption, we have for fixed $K_0 = 4L\Delta n_0/\epsilon^2 + 1$

$$\mathcal{H}_{K_0} = \left(\|v^k - \nabla f(x^k)\| \leq \epsilon, \quad \forall k \leq \min(\mathcal{K}, K_0)\right).$$

occurs with probability at least $1 - \delta$.

Lemma

We have under the Assumption that on $\mathcal{H}_{K_0} \cap (\mathcal{K} > K_0)$, *for all* $0 \leq k \leq K_0$

$$f(x^{k+1}) - f(x^k) \leq -\frac{\epsilon^2}{4Ln_0} \tag{4.2}$$

and hence

$$f(x^{K_0+1}) - f(x^0) \leq -\frac{\epsilon^2}{4Ln_0} \cdot K_0$$

Proof of Theorem 2

We only want to prove $(\mathcal{K} \leq K_0) \supseteq \mathcal{H}_{K_0}$, so if $\mathcal{H}_{K_0}$ occurs, we have $\mathcal{K} \leq K_0$, and $\|v^{\mathcal{K}}\| \leq 2\varepsilon$.

Because $\|v^{\mathcal{K}} - \nabla f(x^{\mathcal{K}})\| \leq \epsilon$ occurs in $\mathcal{H}_{K_0}$, so $\|\nabla f(x^{\mathcal{K}})\| \leq 3\varepsilon$.

If $(\mathcal{K} > K_0)$ and $\mathcal{H}_{K_0}$ occur, plugging in $K_0 = \frac{4L\Delta n_0}{\epsilon^2} + 1$, then from Lemma 4.6 at each iteration the function value descends by at least $\epsilon^2/(4Ln_0)$, We thus have

$$-\Delta \leq f(x^*) - f(x^0) \leq f(x^{K_0}) - f(x^0) \leq -\left(\Delta + \frac{\epsilon^2}{4Ln_0}\right),$$

contradicting the fact that $-\Delta > -\left(\Delta + \frac{\epsilon^2}{4Ln_0}\right)$.

This show $(\mathcal{K} \leq K_0) \supseteq \mathcal{H}_{K_0}$.

From Lemma 4.5, with probability $1 - \delta$, $\mathcal{H}_{K_0}$ occurs, and then $\|v^{\mathcal{K}}\| \leq 2\varepsilon$ and $\|\nabla f(x^{\mathcal{K}})\| \leq 3\varepsilon$.

To compute the gradient cost, note that with probability $1-\delta$, there are most $K_0 = \frac{\Delta}{\epsilon^2/(4Ln_0)} + 1$ iterations.

For the first $\frac{\Delta}{\epsilon^2/(4Ln_0)}$ iteration, note that in each q iterations we access for one time of $\widetilde{\mathcal{O}}(\frac{8\sigma^2}{\epsilon^2})$ stochastic gradients and for q times of one stochastic gradient.

Hence with probability $1-\delta$, the total gradient costs are bounded by

$$\begin{aligned} & \left\lceil \frac{\Delta}{\epsilon^2/(4Ln_0)} \cdot \frac{1}{q} \right\rceil \widetilde{\mathcal{O}}(\frac{8\sigma^2}{\epsilon^2}) + \frac{\Delta}{\epsilon^2/(4Ln_0)} + \widetilde{\mathcal{O}}(\frac{8\sigma^2}{\epsilon^2}) \\ \leq \quad & \left(4\sigma^2\epsilon^{-2} + 64\sqrt{2}L\Delta\sigma \cdot \epsilon^{-3}\right) \cdot \log(\mathit{Factor}) \end{aligned} \tag{4.3}$$

Outline

SARAH

An idea of resemblance was earlier presented the *StochAstic Recursive grAdient algoritHm* (SARAH) (Nguyen et al., 2017a,b)

Algorithm 1: SPIDER-SFO online

Input: ϵ, $\eta = \epsilon/L$, q, K
for $k = 0, 1, \ldots, K-1$ **do**
  if $\mod(k, q) = 0$ **then**
    Draw S samples and let $v^k = (1/S)\sum_{i \in \mathcal{S}} \nabla f_i(x^k)$
  else
    Draw $i \sim [n]$ randomly, set $v^k = \nabla f_i(x^k) - \nabla f_i(x^{k-1}) + v^{k-1}$
  end if
  $x^{k+1} = x^k - \eta v^k / \|v^k\|$
end for
Return x^k whenever $\|v^k\| \leq \varepsilon$

Algorithm 2: iSARAH

Input: $\eta = 1/(2L)$, q, K
for $k = 0, 1, \ldots, K-1$ **do**
  if $\mod(k, q) = 0$ **then**
    Draw S samples and let $v^k = (1/S)\sum_{i \in \mathcal{S}} \nabla f_i(x^k)$
  else
    Draw $i \sim [n]$ randomly, set $v^k = \nabla f_i(x^k) - \nabla f_i(x^{k-1}) + v^{k-1}$
  end if
  $x^{k+1} = x^k - \eta v^k$
end for
Return $\widetilde{x}$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

https://lamnguyen-mltd.github.io/files/INFORMS2017_LamNguyen.pdf

SARAH vs SPIDER-SFO

Both adopt the recursive stochastic gradient update framework, and our SPIDER-SFO can be viewed as a variant of SARAH with normalization. Difference from the works Nguyen et al. (2017a,b) in two aspects:

(i) Our analysis techniques are totally different from the version of SARAH proposed by Nguyen et al. (2017a,b). SARAH proposed can be seen as a variant of gradient descent, while ours hybrids the SPIDER with normalized gradient descent

(ii) Nguyen et al. (2017a,b) adopt a large stepsize setting (in fact their goal was to design a memory-saving variant of SAGA), while our algorithms adopt a small stepsize that is proportional to ε

(iii) Our proposed SPIDER technique is a much more general variance-reduced estimation method for many quantities (not limited to gradients) and can be flexibly applied to numerous problems, e.g. stochastic zeroth-order method

It remains open whether a modification of SARAH can achieve a desirable gradient cost that matches (or surpasses) the cost of SPIDER-SFO

Recently solved by SpiderBoost arXiv:1810.10690

Outline

Finding Second-order Stationary Point

To find an ε-second-order stationary point (ε-SSP)

$$\|\nabla f(x)\| \leq \varepsilon, \qquad \lambda_{\min}\left(\nabla^2 f(x)\right) \geq -\sqrt{\rho\varepsilon}$$

under the assumptions

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$$

and

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq \rho\|x - y\|$$

- Cubic Regularization finds an $(\varepsilon, \sqrt{\rho\varepsilon})$-second-order stationary point in $\mathcal{O}(\varepsilon^{-1.5})$ iterations, but each iteration is costly (Nesterov & Polyak, 2006)
- Trust Region method share the $\mathcal{O}(\varepsilon^{-1.5})$ rate but has similar per-iteration computation cost (Curtis, Robinson & Samadi, 2014) (Cartis-Gould-Toint, 2012)

Finding Second-order Stationary Point

Stochastic Gradient Descent (SGD) (Ge et al., 2015) can escape from all saddle points and achieve an ε-approximate second-order stationary point in $\widetilde{\mathcal{O}}(poly(d)\varepsilon^{-4})$ stochastic gradients

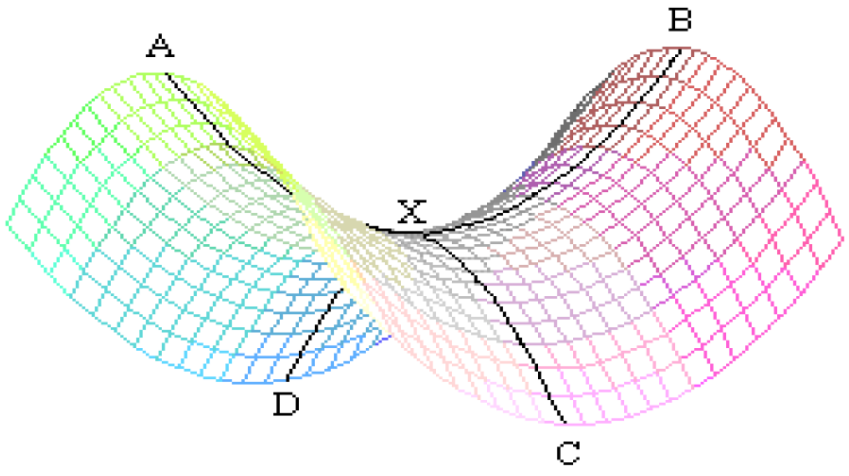
- Update using noisy gradient (injecting the stochastic gradient with isotropic noise)
- Strict-saddle assumption

Perturbed Gradient Descent (PGD) (Jin et al., 2017a) can escape from all saddle points and achieve an ε-approximate second-order stationary point in $\widetilde{\mathcal{O}}(\varepsilon^{-2})$ full gradients

- Update using full gradient and perturb the iteration periodically using ball-shape noise

Noise-perturbed variant of NGD (Levy, 2016) achieves $poly(d)\varepsilon^{-3}$ full gradient

Finding Second-order Stationary Point



- Noise is important!
  (Lee et al. 2016 COLT) Gradient Descent Only Converges to Local Minimizers
  (Du et al. 2017 NIPS) Gradient Descent Can Take Exponential Time to Escape Saddle Points

Finding Second-order Stationary Point

Recent breakthrough in 2016/2017, which allows stochastic Hessian-product accesses. Since $[\nabla^2 f_i(x)]v \approx [\nabla f_i(x+qv) - \nabla f_i(x)]/q$ we assume

Computing $[\nabla^2 f_i(x)]v$ costs similar time as computing $\nabla f_i(x)$

- FastCubic (Agarwal et al., 2017) and its online variant Stochastic Cubic Regularization (Tripuraneni et al., 2017) covert cubic regularization method by Nesterov and Polyak for finding SSP and achieve a gradient cost of $\widetilde{\mathcal{O}}(\min(n\varepsilon^{-1.5} + n^{3/4}\varepsilon^{-1.75}, \varepsilon^{-3.5}))$
- CDHS (Carmon et al., 2016) and its online variant Natasha2 (Allen Zhu, 2017) utilize Negative-Curvature Search method to avoid saddle points and achieve a gradient cost of $\widetilde{\mathcal{O}}(\min(n\varepsilon^{-1.5} + n^{3/4}\varepsilon^{-1.75}, \varepsilon^{-3.5}))$
- See AGD Jin et al. (2017b) and Reddi et al. (2018) for similar rates

NEON for Finding SSP

- In late 2017, two groups Xu et al. (2017); Allen-Zhu & Li (2017b) proposed a Negative-Curvature-Search method called NEgative-curvature-Originated-from-Noise (NEON for short) using stochastic gradients only

- NEON is simply a gap-free variant of Oja's iteration for principal component estimation (Oja, 1982), and its global convergence rate was proved to be optimal (**L.**-Wang-Liu-Zhang, 2018 Math. Prog.; Jain et al., 2016, Allen-Zhu & Li 2017a)

- Using such NEON method, one can convert a series of optimization algorithms that finds FSP (GD/SGD, SVRG/SCSG, CDHS/Natasha2, etc.) to the one that finds SSP using only stochastic gradients

NEON for Finding SSP

- Oja's method for Negative-Curvature Search

$$w^+ = w - \eta[\nabla^2 f_i(x_0)]w \qquad \text{(Oja)}$$

Oja's online PCA is *statistically optimal* and *globally convergent*

- NEON method uses stochastic gradients

$$x^+ = x - \eta[\nabla f_i(x) - \nabla f_i(x_0)] \qquad \text{(Neon)}$$

NEON for Finding SSP

To find an SSP, we can fuse our SPIDER-SFO with a Negative-Curvature-Search (NC-Search) iteration that solves the following task:

$$\boxed{\text{given a point } x \in \mathbb{R}^d, \text{ decide if } \lambda_{\min}(\nabla^2 f(x)) \geq -\delta}$$

OR

$$\boxed{\text{find a unit vector } w_1 \text{ such that } w_1^\top \nabla^2 f(x) w_1 \leq -\delta/2}$$

gradient cost is $\widetilde{\mathcal{O}}(\delta^{-2})$

When w_1 is found, one can set $w_2 = \pm(\delta/\rho)w_1$ where $\pm$ is a random sign. Then under our smoothness Assumption, Taylor's expansion gives

$$f(x + w_2) \leq f(x) + [\nabla f(x)]^\top w_2 + \frac{1}{2} w_2^\top [\nabla^2 f(x)] w_2 + \frac{\rho}{6}\|w_2\|^3$$

Taking expectation, one has

$$\mathbb{E} f(x + w_2) \leq f(x) - \delta^3/(2\rho^2) + \delta^3/(6\rho^2) = f(x) - \delta^3/(3\rho^2)$$

This indicates that when we find a direction of negative curvature or Hessian, updating $x \leftarrow x + w_2$ decreases the function value by $\Omega(\delta^3)$ in expectation

NEON for Finding SSP

Setting $\delta = \sqrt{\rho\varepsilon}$, our SPIDER-SFO$^+$ algorithm that marries SPIDER-SFO with NC-Search is described as

SPIDER-SFO$^+$ Procedure

Step 1. Run NEON2 (Allen-Zhu & Li, 2017) for NC-Search iteration to find an $\mathcal{O}(\delta)$-approximate negative Hessian direction w_1 using stochastic gradients

Step 2. If NC-Search find a w_1, update $x \leftarrow x \pm (\delta/\rho)w_1$ in $\delta/(\rho\eta)$ mini-steps, and simultaneously use SPIDER v^k to maintain an estimate of $\nabla f(x)$. Then Goto Step 1.

Step 3. If such w_1 *cannot* be found, run SPIDER-SFO for $\delta/(\rho\eta)$ steps directly using the SPIDER v^k (without restart) in Step 2. Then Goto Step 1.

Step 4. During Step 3, if we find $\|v^k\| \leq 2\varepsilon$, STOP and return x^k

Theorem (Second-Order Stationary Point, Online)

For the online case, Algorithm SPIDER-*SFO*$^+$ *outputs an* x^k *satisfying with high probability*

$$\|\nabla f(x^k)\| \leq \varepsilon \quad \textit{and} \quad \lambda_{\min}(\nabla^2 f(x^k)) \geq -\sqrt{\rho\varepsilon} \tag{5.1}$$

at a gradient cost of

$$\widetilde{\mathcal{O}}\left(\frac{\Delta L\sigma}{\epsilon^3} + \frac{\Delta L^2}{\rho^{0.5}\varepsilon^{2.5}} + \frac{\sigma^2}{\epsilon^2} + \frac{L^2}{\rho\varepsilon}\right) = \widetilde{\mathcal{O}}(\varepsilon^{-3})$$

Analogously for finite-sum case, SPIDER-SFO$^+$ has a gradient cost of $\widetilde{\mathcal{O}}\left(n + n^{1/2}\epsilon^{-2} + \epsilon^{-2.5}\right)$ which is $\widetilde{\mathcal{O}}(n^{1/2}\varepsilon^{-2})$ when $\varepsilon^{-1} \ll n \ll \varepsilon^{-4}$

Assumption

(i) $\Delta := f(x^0) - f(x^*) < \infty$

(ii') Each component function $f_i(x)$ has L-Lipschitz continuous gradient

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$$

(ii'') Each component function $f_i(x)$ has ρ-Lipschitz continuous Hessian

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq \rho\|x - y\|$$

(iii') (For online case only) the stochastic gradients are subgaussian vectors with variance proxy $\sigma^2 < \infty$, i.e.

$$\mathbb{E}\exp\left(\|\nabla f_i(x) - \nabla f(x)\|^2 / \sigma^2\right) \leq 2$$

SPIDER-SFO$^+$ is NOT NEON+ SPIDER-SFO

Note if we simply apply NEON+ SPIDER-SFO it achieves a gradient cost of $\widetilde{\mathcal{O}}\big(\min(\epsilon^{-3.5}, n^{1/2}\epsilon^{-2.5})\big)$

- The dominate term NEON+ SPIDER-SFO is the *coupling term*

$$\text{for the online case} \qquad \epsilon^{-2}(\sqrt{\epsilon})^{-3} = \epsilon^{-3.5}$$

$$\text{for the finite-sum case} \qquad n^{1/2}\epsilon^{-1}(\sqrt{\epsilon})^{-3} = n^{1/2}\epsilon^{-2.5}$$

Comparable results fails to break the $\mathcal{O}(\epsilon^{-3.5})$ barrier

- SPIDER-SFO$^+$ sharpens the hard-cookie coupling term by cutting one large NEON step into many mini-steps, and maintain the SPIDER estimate

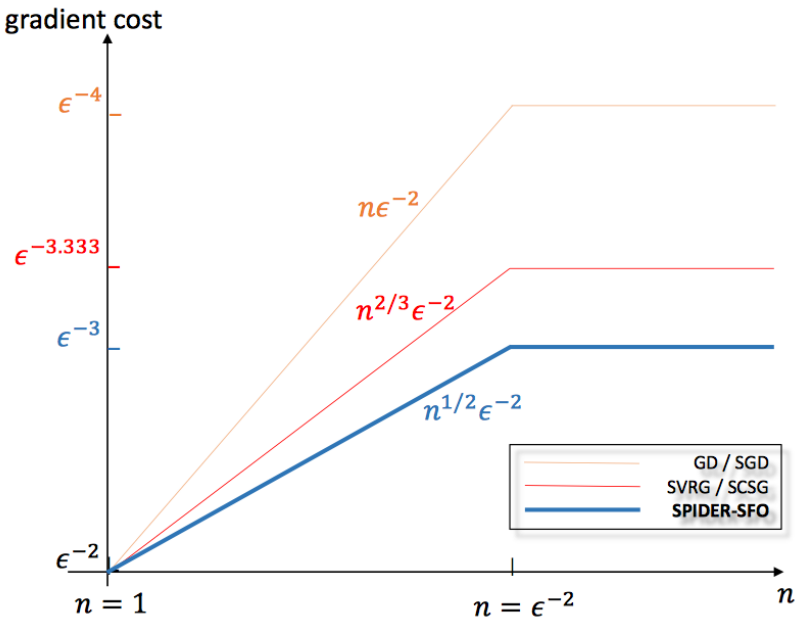
$$\text{for the online case} \qquad \epsilon^{-2}(\sqrt{\epsilon})^{-2} = \epsilon^{-3}$$

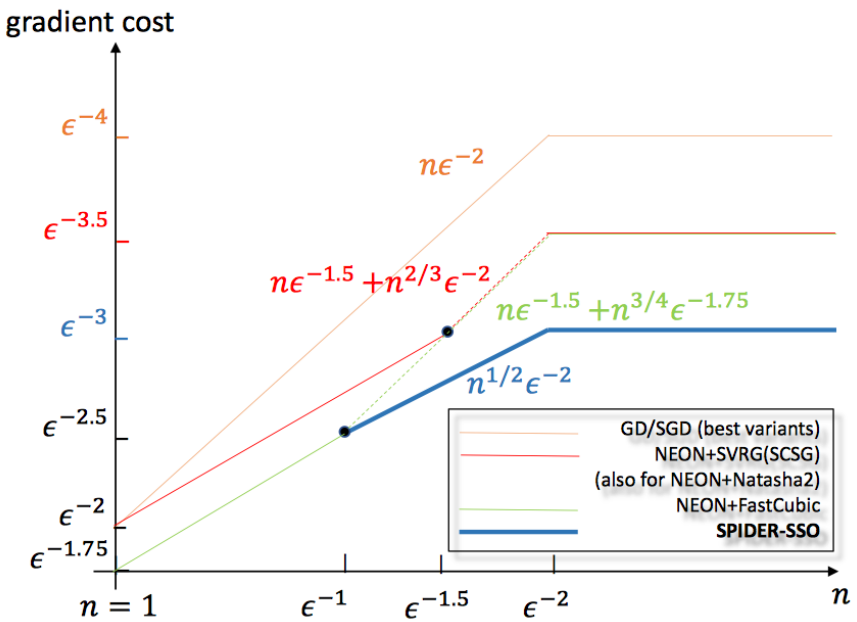
$$\text{for the finite-sum case} \qquad n^{1/2}\epsilon^{-1}(\sqrt{\epsilon})^{-2} = n^{1/2}\epsilon^{-2}$$

- For the finite-sum case, only in the regime $n \gg \varepsilon^{-1}$ can SPIDER-SFO$^+$ outperforms the state-of-the-art.Acceleration in low-n regime is possible

$$\text{State-of-the-art} \qquad \left(n\varepsilon^{-1.5} + n^{3/4}\varepsilon^{-1.75}\right) \wedge \left(n + n^{1/2}\varepsilon^{-2} + \varepsilon^{-2.5}\right)$$

	Algorithm		Online	Finite-Sum
Stationary Point	GD / SGD	(Nesterov, 2004)	ϵ^{-4}	$n\epsilon^{-2}$
	SVRG / SCSG	(Allen-Zhu & Hazan, 2016) (Reddi et al., 2016) (Lei et al., 2017)	$\epsilon^{-3.333}$	$n + n^{2/3}\epsilon^{-2}$
	SPIDER-SFO	(this work)	ϵ^{-3}	$n + n^{1/2}\epsilon^{-2}$ Δ
Local Minimizer (Hessian-Lipschitz Required)	Perturbed GD / SGD	(Ge et al., 2015) (Jin et al., 2017a)	$poly(d)\epsilon^{-4}$	$n\epsilon^{-2}$
	NEON+GD / NEON+SGD	(Xu et al., 2017) (Allen-Zhu & Li, 2017)	ϵ^{-4}	$n\epsilon^{-2}$
	AGD	(Jin et al., 2017b)	N/A	$n\epsilon^{-1.75}$
	NEON+SVRG / NEON+SCSG	(Allen-Zhu & Hazan, 2016) (Reddi et al., 2016) (Lei et al., 2017)	$\epsilon^{-3.5}$ $(\epsilon^{-3.333})$	$n\epsilon^{-1.5} + n^{2/3}\epsilon^{-2}$
	NEON+FastCubic/CDHS	(Agarwal et al., 2017) (Carmon et al., 2016) (Tripuraneni et al., 2017)	$\epsilon^{-3.5}$	$n\epsilon^{-1.5} + n^{3/4}\epsilon^{-1.75}$
	NEON+Natasha2	(Allen-Zhu, 2017) (Xu et al., 2017) (Allen-Zhu & Li, 2017)	$\epsilon^{-3.5}$ $(\epsilon^{-3.25})$	$n\epsilon^{-1.5} + n^{2/3}\epsilon^{-2}$
	SPIDER-SSO	(this work)	ϵ^{-3}	$n^{1/2}\epsilon^{-2}$ Θ


gradient cost
ϵ^{-4}
$\epsilon^{-3.333}$
ϵ^{-3}
ϵ^{-2}
$n\epsilon^{-2}$
$n^{2/3}\epsilon^{-2}$
$n^{1/2}\epsilon^{-2}$
GD / SGD
SVRG / SCSG
SPIDER-SFO
$n = 1$
$n = \epsilon^{-2}$
n


gradient cost
ϵ^{-4}
$\epsilon^{-3.5}$
ϵ^{-3}
$\epsilon^{-2.5}$
ϵ^{-2}
$\epsilon^{-1.75}$
$n\epsilon^{-2}$
$n\epsilon^{-1.5} + n^{2/3}\epsilon^{-2}$
$n\epsilon^{-1.5} + n^{3/4}\epsilon^{-1.75}$
$n^{1/2}\epsilon^{-2}$
GD/SGD (best variants)
NEON+SVRG(SCSG)
(also for NEON+Natasha2)
NEON+FastCubic
SPIDER-SSO
$n = 1$
ϵ^{-1}
$\epsilon^{-1.5}$
ϵ^{-2}
n

Outline

Summary of Our Contribution

(i) Proposed SPIDER technique for tracking:

- Avoidance of excessive access of oracles and reduction of time complexity
- Potential application in many stochastic estimation problems

(ii) Proposed SPIDER-SFO and SPIDER-SFO$^+$ algorithms for first-order non-convex optimization

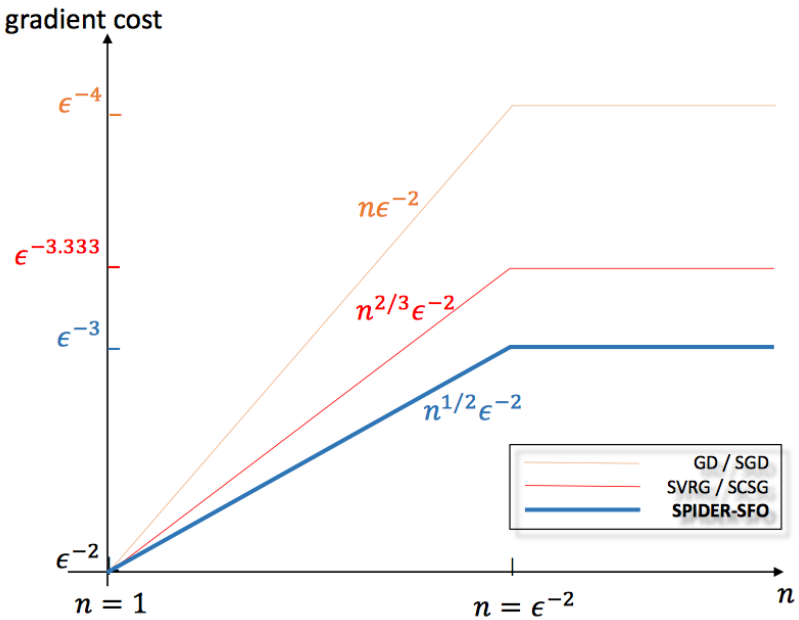
- Achieves $\widetilde{\mathcal{O}}(\varepsilon^{-3})$ rate for finding ε-FSP and $(\varepsilon, \sqrt{\rho\varepsilon})$-SSP
- Convergence rate in both high-probability and expectation
- Proved that SPIDER-SFO matches the lower bound in the finite-sum case [Carmon et al 2017]

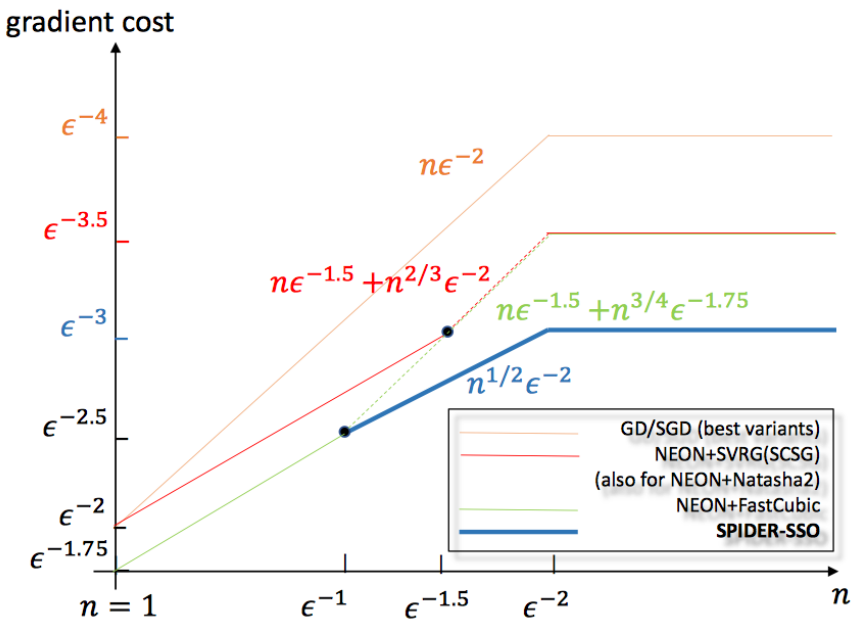
(iii) Proposed SPIDER-SZO algorithm for zero-order non-convex optimization

- Achieves an improved rate of $\mathcal{O}(d\varepsilon^{-3})$

- More done: SFO mini-batches in inner loops
- To be done: proximal case SpiderBoost, lower bounds for the online case

	Algorithm		Online	Finite-Sum
Stationary Point	GD / SGD	(Nesterov, 2004)	ϵ^{-4}	$n\epsilon^{-2}$
	SVRG / SCSG	(Allen-Zhu & Hazan, 2016) (Reddi et al., 2016) (Lei et al., 2017)	$\epsilon^{-3.333}$	$n + n^{2/3}\epsilon^{-2}$
	SPIDER-SFO	(this work)	ϵ^{-3}	$n + n^{1/2}\epsilon^{-2}$ Δ
Local Minimizer (Hessian-Lipschitz Required)	Perturbed GD / SGD	(Ge et al., 2015) (Jin et al., 2017a)	$poly(d)\epsilon^{-4}$	$n\epsilon^{-2}$
	NEON+GD / NEON+SGD	(Xu et al., 2017) (Allen-Zhu & Li, 2017)	ϵ^{-4}	$n\epsilon^{-2}$
	AGD	(Jin et al., 2017b)	N/A	$n\epsilon^{-1.75}$
	NEON+SVRG / NEON+SCSG	(Allen-Zhu & Hazan, 2016) (Reddi et al., 2016) (Lei et al., 2017)	$\epsilon^{-3.5}$ $(\epsilon^{-3.333})$	$n\epsilon^{-1.5} + n^{2/3}\epsilon^{-2}$
	NEON+FastCubic/CDHS	(Agarwal et al., 2017) (Carmon et al., 2016) (Tripuraneni et al., 2017)	$\epsilon^{-3.5}$	$n\epsilon^{-1.5} + n^{3/4}\epsilon^{-1.75}$
	NEON+Natasha2	(Allen-Zhu, 2017) (Xu et al., 2017) (Allen-Zhu & Li, 2017)	$\epsilon^{-3.5}$ $(\epsilon^{-3.25})$	$n\epsilon^{-1.5} + n^{2/3}\epsilon^{-2}$
	SPIDER-SSO	(this work)	ϵ^{-3}	$n^{1/2}\epsilon^{-2}$ Θ


gradient cost
ϵ^{-4}
$\epsilon^{-3.333}$
ϵ^{-3}
ϵ^{-2}
$n\epsilon^{-2}$
$n^{2/3}\epsilon^{-2}$
$n^{1/2}\epsilon^{-2}$
GD / SGD
SVRG / SCSG
SPIDER-SFO
$n = 1$
$n = \epsilon^{-2}$
n


gradient cost
ϵ^{-4}
$\epsilon^{-3.5}$
ϵ^{-3}
$\epsilon^{-2.5}$
ϵ^{-2}
$\epsilon^{-1.75}$
$n\epsilon^{-2}$
$n\epsilon^{-1.5} + n^{2/3}\epsilon^{-2}$
$n\epsilon^{-1.5} + n^{3/4}\epsilon^{-1.75}$
$n^{1/2}\epsilon^{-2}$
GD/SGD (best variants)
NEON+SVRG(SCSG)
(also for NEON+Natasha2)
NEON+FastCubic
SPIDER-SSO
$n = 1$
ϵ^{-1}
$\epsilon^{-1.5}$
ϵ^{-2}
n

Thanks for Your Attention!

- Z. Allen-Zhu & Y. Li (2018). Neon2: Finding local minima via first-order oracles, NIPS 2018
- Yi Xu and Tianbao Yang (2018). First-order Stochastic Algorithms for Escaping From Saddle Points in Almost Linear Time, NIPS 2018
- Lam M. Nguyen, Liu, Scheinberg, Takac (2017a). SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient, ICML 2017
- Lam M. Nguyen, Liu, Scheinberg, Takac (2017b). Stochastic Recursive Gradient Algorithm for Nonconvex Optimization
- Zhou, Xu, Gu (2018a). Stochastic Nested Variance Reduction for Nonconvex Optimization
- Zhou, Xu, Gu (2018b). Finding Local Minima via Stochastic Nested Variance Reduction

Other Methods: Perturbed GD/AGD, NEON, etc

- Rong Ge, Furong Huang, Chi Jin & Yang Yuan (2015), COLT
- Jason D. Lee, Simchowitz, Jordan & Recht (2016), COLT
- Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., & Jordan, M. I. (2017). How to escape saddle points efficiently. ICML
- Jin, C., Netrapalli, P., & Jordan, M. I. (2018). Accelerated gradient descent escapes saddle points faster than gradient descent. COLT

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., & Ma, T; Carmon, Duchi, Hinder, Sidford; Reddi et al. (2018); Tripuraneni et al. (2017), etc.